Machine Learning for Our Multi-agentic World

Tom Yan

December 2025
CMU-ML-25-121

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Zachary Lipton, Carnegie Mellon University (Chair)
Ariel Procaccia, Harvard University (Co-Chair)
Andrej Risteski, Carnegie Mellon University
Kun Zhang, Carnegie Mellon University
Avrim Blum, Toyota Technological Institute at Chicago

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was sponsored by National Science Foundation awards DGE1745016, DGE2140739, IIS1714140 and CCF1733556, United States Army award W911NF1320045 and grants from Highmark and PricewaterhouseCoopers.

Keywords: Multi-agent Learning, Game Theory, Reinforcement Learning Theory, Computational Economics, Strategic Learning, AI Safety

To my family

Abstract

Our world is multi-agentic, and increasingly our use of machine learning is mirroring this. Over the past decade, machine learning models have improved by leaps and bounds. As models grow in capability, they become more widely adopted and used to solve more complex problems. Both such trends result in increasingly numerous and sophisticated *multi-agent* interactions.

This thesis studies several fundamental challenges when designing machine learning models to handle and harness multi-agent interactions, developing new guarantees, algorithms and frameworks to this end.

The first part of this thesis focuses on advancing machine learning from application in static, single-player settings to agentic, two-player games. How can we adapt machine learning models to account for the presence of other agents affected by their output? Towards this goal, we study machine learning in three important classes of games: strategic data collection, prediction and auditing. All three types of games are motivated by the use of machine learning in consequential, societal applications such as training data collection, automated loan approval and industry AI regulation.

Beyond accounting for multi-agent considerations, we may wish to *leverage* multiple agents to solve complex problems. The second part of this thesis focuses on machine learning for and of multi-agent systems. How can we use learning to allow multiple agents to be better together? Towards this aim, in the decentralized multi-agent setting, we study how learning can facilitate multi-agent coordination, examining agent attribution and payment design as two possible coordination mechanisms. In the centralized multi-agent setting, we investigate how learning can realize the advantage of multiple agents, exploring diversity and specialization as two natural avenues for unlocking their utility.

Altogether, this thesis seeks to consolidate our understanding of several fundamental problems in multi-agent learning, a topic that I believe will become steadily more relevant in the years to come.

Acknowledgments

I had a great time in grad school working on the research I was excited about, and a huge part of this was the people who I was fortunate to be around and learn from! I thank each and every one of them. Below is a chronological account of people who I was lucky enough to interact with during this journey.

Starting out, I was fortunate to have been mentored by Ariel. True to his email handle, he is a "pro" researcher and I was lucky to be able to learn from one of the best researchers in EC. Besides being very sharp and meticulous, Ariel is also very hard-working. He is patient (which I am sure I thoroughly tested). And importantly, I think he finds great joy in the work he does. I think he has achieved something very rare, which is to have built and now lead a very impactful community, demonstrating time and time again the wonderful use and great impact of theory. I really admire him for this and find it very inspiring! All in all, it was a privilege for a rookie like me to learn from a pro like him. I am grateful to have had this experience, and hope he is doing really well at Harvard.

Following Ariel, I was also fortunate to have been mentored by Zack. I remember Zack as a very gritty researcher. He is very resourceful, and just seems to have some kind of je ne sais quoi to get things done, one way or another. Like Ariel, he also seems to have boundless energy. I distinctly remember that he would tell me to "go up strong" and be more decisive in the things I do and say. I appreciate that he always kept it real with me in terms of constructive criticism, and that he has provided timely advice to me, all of which turned out to be very useful. For example, he told me that there are multiple facets to being a researcher. It is not only about publishing papers, but also fostering a community and mentoring junior students. All of this was very important to know and helped complement what I learned from Ariel. So I thank him for sharing this with me! Finally, and above all else, I thank Zack for giving me the freedom to do the research I want to do and thus allowing me to produce this (somewhat) coherent body of work. This is not something to be taken for granted! And I am grateful to him for this, and hope he is doing really well at his startup.

Next, I would like to thank the rest of my thesis committee: Avrim, Kun and Andrej. In particular, I thank Avrim for his sharp and incisive questions. I thought it was impressive he got to the heart of the matter with just one question (about my problem formulation) during my proposal. Thinking more about his question gave me a clearer sense for how to better position my paper.

I had fun collaborations during my PhD, and I only wish that I had collaborated more! I am grateful for the collaborations and conversations with the following people: Ariel Procaccia, Ritesh Noothigattu, Christian Kroer, Alex Peysakhovich, Ellen Vitercik, Shantanu Gupta, Rachel Childers, Zack Lipton, Neil Xu and Chicheng Zhang. Having great collaborators is not something to be taken for granted, and I am grateful to each and every one of them for working with me!

In particular, I would like to thank Neil, who was the first junior student I worked with. This was an important step for me to take. I learned that besides the research itself, it is important to keep everyone's spirits up and maintain faith that the project will work out (which I am glad it did). I am thankful for his trust in me to drive the project, for working with me even during his internships and for teaching me a lot about anytime approaches. I hope he is doing very well in NYC.

I would be remiss if I did not give a big shout out to my long-time collaborator, Chicheng. Above all else, I feel a high level of comfort when working with Chicheng, and this really allowed creative research ideas to flow. And even though he thinks of me as a peer, I really think of Chicheng as a mentor. He has taught me a lot about theory and I am thankful that I was able to learn from one of the best theory researchers in the business (especially in active learning and IRL). In sum, I have always had a fun time collaborating and hope that he and I will collaborate again! Also, I hope he is doing very well in Arizona with his family.

Navigating through logistics during one's PhD is no joke, and I am grateful for the MLD staff who make it so easy. Thanks to Laura and Suzanne for helping me out. And of course, special thanks to the wonderful Diane, who truly helped to keep MLD together. I hope she has a great retirement and lots of fun trips!

On a more personal note, I am tremendously grateful for the magical companionship of my friends (and ex-girlfriends), from my undergrad, the wonderful MLD community and the broader CMU SCS community! It is always uplifting and fun to chat about all kinds of topics ranging from AI to current affairs to the NFL. I am grateful to have all of you around, brimming with curiosity to discuss every random thing under the sun or just to talk some smack and laugh over memes (no shortage of these nowadays...). Miss you all dearly!

Lastly, and above all else, I want to deeply thank my family. They have always shown me endless support through the toughest of times. I am grateful to them for letting me be myself and for affording me all these opportunities through their sacrifices. Truly, I am able to go further only because you have come so very far. Love you Mom and Dad!

Contents

List of Figures

List of Tables

Chapter 1

Overview

Our world is inherently multi-agentic, and increasingly our use of machine learning reflects this. As ML models grow in capability, they become more widely adopted and used to solve progressively complex problems. Wider adoption results in increasingly numerous agent-on-agent interactions. Solving increasingly complex problems will require accounting for multi-agent considerations, if not training multi-agent solutions. This thesis studies several fundamental challenges that arise when designing machine learning models to handle and harness multi-agent interactions.

1.1 Adapting ML for Multi-Agent settings

In the first half of this thesis, we study how to adapt machine learning models to account for multi-agent considerations, wherein ML is used in the presence of another agent affected by its output. We focus on three classes of games that model strategic training data collection for automation, strategic prediction on individuals and strategic auditing of black-box industry models.

Strategic Data Collection: Automation is one of the primary uses of ML. When companies are training ML models to automate jobs, labelers are needed to label the training data. However, this in turn introduces a natural conflict of interest. The labelers would be helping to train a model that will go on to render their expertise and jobs redundant. Towards formalizing this conflict of interest, in Chapter 2, we study the data labeling game that results when data labelers have agency and strategize knowing their eventual replacement by the trained model [307].

Strategic Prediction: Machine learning models are now also used by institutions to automate consequential decision making, which includes predictions on individuals e.g. in loan approvals. This constitutes a multi-agent interaction as the individuals will adjust to the ML model, so as to induce their desired prediction in consequential settings. Thus, during prediction, the machine learning model has to account for an agentic data distribution that shifts according to the model itself. Causality is a principled framework for assessing distribution shift (especially in social applications). This thesis devotes three chapters to consolidating our understanding of the causal strategic foundation of ML.

In Chapter 3, we study how to minimally reveal the ML model in order to disincentivize any shift and subsequent causal effect [309]. Next, in Chapter 4, we study which ML models are also good causal incentives (reward models) that can leverage the agency of the data distribution to induce distribution shifts with desirable causal effects [313]. Finally, in Chapter 5, we close the loop and study how to perform finite-sample causal discovery so that we can discover the underlying causal structure, which can then be used for ML model design [311].

Strategic Auditing: As ML models become widely adopted in industries for automation, there is a growing need to audit such models to ensure they are properly regulated. Besides the challenge of statistical estimation, the auditor also needs to account for post-audit manipulation on the part of the strategic auditee. In Chapter 6, we study black-box auditing of machine learning models while accounting for strategic effects [308]. We formalize the task of auditing and the notion of manipulation-proofness, and study various algorithms for statistically efficient audits that cannot be circumvented by post-audit manipulation.

1.2 ML for and of Multi-agent Systems

In the second half of this thesis, we study how machine learning can allow multiple agents to be better together, in both decentralized and centralized settings.

1.2.1 Coordination in Decentralized Multi-agent Systems

In decentralized multi-agent systems, the agents have not been jointly trained. This makes coordination a central challenge. Thus, we investigate how learning can be used to learn mechanisms that facilitate coordination among agents.

Evaluating and Rewarding Agents in MAS: A key challenge in multi-agent systems is that of agent evaluation, useful for equitable credit assignment that ensures cohesion among the agents. What is a principled and "fair" method for attributing and assigning rewards, after multiple agents have collaborated to perform a task? Fortunately, there are solution concepts from Cooperative Game Theory for this purpose. Two such solution concepts with provable guarantees are the Shapley Value and the Core, providing principled ways to attribute marginal and coalitional impact respectively. In Chapter 7 and Chapter 8, we study how ML can be used as a scalable means to operationalize these solution concepts for attribution in multi-agent systems [306, 312].

Outcome-based Payment for Decentralized Coordination in MAS: As more businesses adopt agents to carry out tasks on their behalf, their agents will inevitably interact (much in the same way businesses interact in our present commercial world). All of these agents will thus form a decentralized multi-agent system. In anticipation of this, we seek to understand how an agent can coordinate with other agents that may have differing (business) interests. In present-day commerce, payment is a standard means that different business parties use to better align their business interests. In Chapter 9, we study how one could learn analogous outcome-based payment schemes for helping agents coordinate in the decentralized multi-agent setting [310].

1.2.2 Training in Centralized Multi-agent Systems

Turning to the centralized setting, we investigate how learning can realize the advantages of multi-agent systems over single-agent systems, and produce a system that supersedes the capability of any one agent. Indeed, in the future, we hope to use ML to solve increasingly complex problems. To do so, we may wish to train multiple agents that work together, much in the same way we humans collaborate to solve problems beyond any of our individual means.

Benefits of Diversity in MAS: One key advantage of multi-agent systems is diversity. It is intuitive that combining multiple, diverse agents should yield a stronger agent. A natural method for policy aggregation is Inverse Reinforcement Learning, which can be used to learn from all agents through their trajectories. In Chapter 10, we investigate the effectiveness of policy aggregation in the context of multi-agent Inverse Reinforcement Learning, where we study a canonical multi-agent setup with the population of agents having similar but not identical rewards [216].

Benefits of Specialization in MAS: Another major advantage of multi-agent systems is that of specialization. If the task structure allows for apt task division, an agent can be trained to focus on each smaller sub-task. This in turn realizes a key benefit of multi-agent systems over monolithic, single-agent systems. Each smaller and specialized agent can be more readily supervised, thus enabling scalable oversight. In Chapter 11, we study these benefits of hierarchical multi-agent systems in the context of Hierarchical Reinforcement Learning, developing learning algorithms that leverage this structure for scalable oversight and improved sample efficiency [304].

Part I

Adapting Machine Learning for Multi-Agent settings

Chapter 2

Strategic Data Collection

2.1 Introduction

Over the past few years, the rapid growth of Machine Learning (ML) capabilities has raised the possibility of wide-ranging automation, and consequent worker replacement. Taking a step back from when these ML models are phased in, we ask a basic question on how they first come about:

> Where will the training data for these ML models come from?

In many industries, domain-specific knowledge is required to perform the job. Much of this expertise is proprietary (e.g. trade secrets), and not made publicly available (e.g. on the internet). Thus, in these industries, the answer to our question is paradoxically that: the training data can only come from the workers themselves. At this point, we arrive at a clear conflict of interest.

On the one hand, corporations wish to automate tasks through ML models. On the other hand, the data needed to train these models can only come from the domain experts — the workers in this case, who *know full well* that these models, when trained, will go on to replace them at their jobs. Thus, this raises the possibility that we may see workers actually aim to slow down learning, in order to delay replacement and be compensated for as many labels as possible before then.

We note that the idea of AI job displacement is no longer a rarefied topic, entertained only in academia. The possibility of AI displacement has been written about in recent articles [42], and even surfaced in labor union negotiations. In May 2023, Hollywood screenwriters went on strike to negotiate a better deal. One part of their demands is for there to be limits on companies being able to train ML models on the scripts produced by the writers themselves [292]. Indeed, without this protection, companies can train AI models to emulate and write as well as the writers, eventually replacing them with the trained models. In sum, we believe it is now high time to develop our understanding of the *replacement* aspect of learning, which is what we set out to do in this chapter.

Remark: Before moving on, we point out that the conflict of interest described above is fairly general, and arises *whenever* the labeler wishes to maximize payment from labeling. Consider more broadly the interaction between any data provider (e.g. a data labeling company) and learner (e.g. company needing ML models). The more informative the data labeled by the provider, the faster the learner learns, the fewer the examples the learner needs to query the provider and the lower the provider's subsequent payment. The AI automation setting we describe is one of many such instances where the labeler's objective is at odds with that of the learner: the labelers have the incentive to slow down learning, to maximize their compensation from labeling before the models are fully trained and render their labeling expertise redundant.

In this chapter, we study the learning game that arises when the labeler and learner's objective are at odds. The learner wants to learn quickly, but the labeler wants the learning to progress slowly. Notably, this requires departing from the standard assumption in learning theory that the labeler readily labels any example queried (including the informative examples). We term this game the *Human-AI Substitution game*, since typically the labeler is human and the more the model is trained, the less the learner needs the labeler (to label). To study the rate of learning, we turn to theory to analyze how the labeler can slow down learning.

Our Contributions: In Section 2.2, we formalize the learning game and game value, developing a novel representation of the game state — effective version space (henceforth abbreviated as E-VS). In Section 2.3, we then develop a natural, efficient learning Algorithm 2, which we prove achieves near-optimal minimax query complexity. We also show that other AL algorithms may be inefficient. In Section 2.4, we examine more general settings involving noisy or non-strategic labelers, showing that our algorithm can nevertheless achieve good query complexity. Finally, in Section 2.5, we consider the multi-task setting and analyze when strategic labeling can further enlarge the learner's query complexity beyond the sum of the individual tasks' query complexities.

2.1.1 Active learning with a simple twist

We begin our investigation by adopting the standard active learning setup [135], with the only twist that the labeler aims to maximize the learner's query cost. We focus on perhaps the most fundamental setting: exact learning through membership queries [13, 138]. As we will see, this setup is fairly general, and one may use standard reductions to reduce the PAC and noisy setting to this setting.

Setup of the Learning Game:

- The learner is interested in learning a hypothesis h^* in hypothesis class $\mathcal{H} \subset (\mathcal{X} \to \{+1, -1\})$ over a finite pool of unlabeled data $\mathcal{X}$, collected by the learner.
- The labeler knows h^* and responds using labeling strategy T with response $T(x) \in \{h^*(x), \perp\}$, where $\perp$ denotes abstention. [1]
- The learner repeatedly interacts with the labeler adaptively, and makes label queries on unqueried example x, and incurs cost $\mathbb{1}(T(x) \neq \perp)$ for each such query.[2]

In this chapter, we model the labeler as being able to strategically abstain on queried data, to slow down learning. Being the domain expert with specialized expertise, the labeler is assumed to be able to use this leverage to selectively decide which data points to label. As noted in Section 2.1, some data points are particularly informative, and naturally the labeler would wish to decline labeling these so that more data would need to be labeled. We also add that this strategy of

[1] In Section 2.2.2 and Appendix 2.8 we also study a variant of the game (Protocol 4) where the labeler can choose to reveal binary labels or abstain *adaptively*.

[2] Note that we define the cost for all non-abstention label feedback to be 1 for all x. However, as we show in Appendix 2.9, our algorithm can generalize to handle varying data prices (price for non-abstention label feedback $c(x)$ can be dependent on feature x).

slowing down the transfer of expertise is not a novel conception. It has been well-documented that in apprenticeships, for instance, teachers (master) strategically slow down the training of their apprentices [111].

The interaction finishes when the termination condition is met, or the learner's querying strategy halts. Based on the learner's desired learning outcome, the termination condition is defined as when $h^* \in \mathcal{H}$ is identified, which we formalize in the following section. If the termination condition is met, the labeler gets a payoff of 1 for every *labeled* data provided. If the termination condition is not met, the labeler gets a payoff of 0. In this game, the learner aims to minimize the total payoff needed to learn h^*, while the labeler aims for the opposite and to maximize the total payoff.

Guaranteeing Learning Outcome: Before proceeding, we note that the labeler *can* always satisfy the learner's objective — by using the non-strategic labeling strategy $T(x) = h^*(x)$ as in the standard active learning setup. Since the labeler can realize the learning outcome, we assume that the learner has this guarantee (of the learning outcome) written into the contract; no payment is awarded otherwise. Indeed, if the labeler cannot guarantee the learning outcome, it seems unlikely that the learner would have chosen to contract the labeler in the first place.

Prolonging Learning through Abstention: The key tension in this interaction is that the labeler has to label in order to be paid, but any labeling results in less data that subsequently need to be labeled. With the labeler only allowed to abstain besides labeling, it is natural to ask: can abstention *significantly* enlarge the query complexity? Our investigation is motivated by the affirmative answer below, where we find that abstention can *exponentially* enlarge query complexity in some settings.

Proposition 1 (Abstention induces exponentially higher query complexity)**.** *There exists a hypothesis class $\mathcal{H}$, instance domain $\mathcal{X}$ such that: the query complexity is $O(\log |\mathcal{X}|)$ if the labeler is unable to abstain, and $\Omega(|\mathcal{X}|)$ for any learning algorithm if the labeler is allowed to abstain.*

2.2 The Minimax Learning Game

2.2.1 Representation of the learning game state

To study this learning game, we first develop a useful, succinct representation of the game state, which is a key contribution of our paper and allows us to formalize the termination condition and the protocol. We start by defining the canonical state representation used in conventional AL without abstention, the version space (VS) [207].

Definition 1. *Given a queried dataset S and a set of hypotheses V, define version space $V[S] = \big\{h \in V : \forall (x, y) \in S \wedge y \neq \perp, h(x) = y\big\}$ as the subset of hypotheses in V consistent with S.*

In our setting of learning with strategic abstention, some queried examples in S will not have their binary labels available to the learner, due to the labeler's abstention. And so, we observe that certain hypotheses may be consistent, but *indistinguishable* from other hypotheses, even if all the remaining unqueried data is labeled. This motivates defining a new notion of identifiability of a hypothesis under queried dataset S. Let the set of all queried examples be $S_X = \big\{x : (x, y) \in S\big\}$.

Definition 2. *Given the set of queried examples and their label responses S, and the queried examples S_X, hypothesis $h \in \mathcal{H}$ is said to be identifiable with respect to S if:*

Protocol 1 Human-AI Substitution game interaction protocol

```
Require: Instance domain $\mathcal{X}$, hypothesis class $\mathcal{H}$, queried examples $S_X$, queried dataset $S$
$V \leftarrow \mathcal{H}$, $S_X \leftarrow \emptyset$, $S \leftarrow \emptyset$

Nature chooses some $h^* \in \mathcal{H}$ given to the labeler    ▷ Throughout, labeler maintains that $h^*$ is identifiable: $h^* \in E(V, S_X)$.
while $|E(V, S_X)| \geq 2$ do
    Learner adaptively queries example $x \in \mathcal{X} \setminus S_X$ using learning algorithm $\mathcal{A}$
    Labeler adaptively gives label feedback $y \in \{h^*(x), \perp\}$ using labeling oracle $T$
    Learner updates the VS: $V \leftarrow V[(x, y)]$    ▷ Recall Definition 1
    $S_X \leftarrow S_X \cup \{x\}$, $S \leftarrow S \cup \{(x, y)\}$
if $|E(V, S_X)| = 1$ then
    Learner makes total payment to the labeler: $\sum_{(x_i, y_i) \in S} \mathbb{1}\{y_i \neq \perp\}$
```

$\mathcal{X}$ $\mathcal{H}$	x_1	x_2	x_3
h_1	$+1$	-1	$+1$
h_2	-1	-1	$+1$
h_3	$+1$	$+1$	-1
h_4	-1	$+1$	-1
h_5	$+1$	$+1$	$+1$

Table 2.1: Consider an example hypothesis class $\mathcal{H} = \{h_1, h_2, h_3, h_4, h_5\}$ and instance space $\mathcal{X} = \{x_1, x_2, x_3\}$. The interaction history is $S = \{(x_1, \perp)\}$, and therefore $S_X = \{x_1\}$. Under S, we have that the VS (Definition 1), $V = \mathcal{H}[S] = \{h_1, h_2, h_3, h_4, h_5\}$. We observe that h_1 and h_2 make identical predictions on $\mathcal{X} \setminus S_X = \{x_2, x_3\}$. Likewise, h_3 and h_4 make identical predictions on $\mathcal{X} \setminus S_X$. Therefore, effective version space is actually $E(V, S_X) = \{h_5\}$. If the game reaches this stage, the learner can *already identify* that the target h^* must be h_5.

- *h is consistent with S, $h \in \mathcal{H}[S]$.*
- *for all other consistent $h' \in \mathcal{H}[S]$: $h'(\mathcal{X} \setminus S_X) = h(\mathcal{X} \setminus S_X) \implies h' = h$, where for brevity we denote $h_1(U) = h_2(U) \iff \forall x \in U \,.\, h_1(x) = h_2(x)$.*

In other words, h is identifiable with respect to S if over the remaining examples $\mathcal{X} \setminus S_X$, some labeling strategy (specifically, one that reveals $h(x)$ on every $x \in \mathcal{X} \setminus S_X$) allows h to be distinguished from all other hypotheses in $\mathcal{H}[S]$. With this, we may develop a new representation of the state of the game, effective version space (E-VS). The E-VS is a refinement of VS, and comprises of only identifiable hypotheses given the examples queried. Please see Table 2.1 for an illustration.

Remark: The key insight here is that abstention can in fact *reveal information*. This is despite that abstention is used by the labeler to *prevent releasing* information about h^*. The reason why one can glean information from labeler's abstention is that hypotheses could be rendered unidentifiable by abstention on a data point, and thus be ruled out without needing further queries. We operationalize this insight to develop the effective version space representation, which we formalize below.

Definition 3. *Given a set of classifiers V and a set of examples S_X, define*

$$E(V, S_X) = \{h \in V : \forall h' \in V \setminus \{h\} : h'(\mathcal{X} \setminus S_X) \neq h(\mathcal{X} \setminus S_X)\}$$

as the effective version space *with respect to V and S_X.*

Definition 4. *$h^* \in \mathcal{H}$ is* identified *by queried dataset S if the E-VS, $E(\mathcal{H}[S], S_X) = \{h^*\}$.*

With the identification criterion defined, we now formalize the interaction in Protocol 1. Here, the termination states are defined as either $|E(V, S_X)| = 1$ (a hypothesis is identified and the learning outcome is met), or $E(V, S_X) = \emptyset$ (no hypothesis *can* be identified).

2.2.2 The minimax learning game

In this paper, we analyze the minimax query complexity — that of the worst-case $h^* \in \mathcal{H}$ to learn under Protocol 1. Towards this, we formulate a related minimax learning game (see Protocol 4 in Appendix 2.8), where both the learner queries and the labeler labels *adaptively*, depending on the interaction in previous rounds, with the game's optimal value function defined as follows:

$$\mathrm{Cost}(V, S_X) = \begin{cases} -\infty & E(V, S_X) = \emptyset \\ 0 & |E(V, S_X)| = 1 \\ \min_{x \in \mathcal{X} \setminus S_X} \max_{y \in \mathcal{Y}} \mathbb{1}(y \neq \perp) + \mathrm{Cost}(V[(x, y)], S_X \cup \{x\}) & |E(V, S_X)| \geq 2 \end{cases} \tag{2.1}$$

Compared to the original Protocol 1, Protocol 4 can be viewed as giving the labeler more freedom: the labeler does not need to commit to provide binary labels using a given h^*; it just needs to maintain the invariant that there is some h^* identifiable and consistent with all examples seen. As we will see shortly, the optimal value function Cost of Protocol 4 serves as a useful tool in analyzing the optimal query complexity of Protocol 1.

In the case of non-identifiability, we use a base-case payoff of $-\infty$ to encode that the labeler must ensure identification. As noted in Section 2.1, any optimal labeler will never end up in such a state, because a positive payoff can always be achieved – the strategy $T = h^*$ results in a positive payoff. We now turn to formalizing what an identifiable strategy is.

Definition 5. *Given $h \in \mathcal{H}$, define the set of labeling oracles consistent with h, as:*

$$\mathcal{T}_h = \{T : \mathcal{X} \to \{+1, -1, \perp\} \, | \forall x \in \mathcal{X} \textit{ s.t } T(x) \neq \perp, T(x) = h(x)\}.$$

For subset $S_X \subseteq \mathcal{X}$, let $T(S_X) = \big\{(x, T(x)) : x \in S_X\big\}$ be the labeled (binary or abstention) examples provided by labeling oracle T on the examples S_X.

Definition 6. *A labeling strategy $T \in \mathcal{T}_h$ is an identifiable oracle if the VS, $\mathcal{H}[T(\mathcal{X})] = \{h\}$.*

In the learning game, the labeler's strategy is some labeling oracle, while the learner's strategy corresponds to some deterministic querying algorithm: $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^* \to \mathcal{X}$, where $\mathcal{Y} = \{+1, -1, \perp\}$. Define $\mathrm{Cost}_{\mathcal{A},T}(V, S_X)$ to be value of the learning game under querying strategy $\mathcal{A}$ and labeling strategy T. The key result of this subsection is that the game value $\mathrm{Cost}(\mathcal{H}, \emptyset)$ can serve as a useful measure of minimax query complexity. $\mathrm{Cost}(\mathcal{H}, \emptyset)$ lower bounds the worst-case query complexity of any deterministic learning algorithm in Protocol 1.

Proposition 2. *For any deterministic, exact learning algorithm $\mathcal{A}$,*

$$\max_{h \in \mathcal{H}, T \in \mathcal{T}_h} \mathrm{Cost}_{\mathcal{A},T}(\mathcal{H}, \emptyset) \geq \mathrm{Cost}(\mathcal{H}, \emptyset)$$

This means that for every exact learning algorithm $\mathcal{A}$, there is some worst-case labeling oracle T_h that induces at least $\mathrm{Cost}(\mathcal{H}, \emptyset)$ cost. Please see Appendix 2.8 for all proofs in this section.

2.3 E-VS Bisection Algorithm Analysis

In this section, we design a natural and efficient algorithm based on E-VS bisection, Algorithm 2, which we prove achieves query complexity $O(\mathrm{Cost}(\mathcal{H}, \emptyset) \ln |\mathcal{H}|)$. Proving this guarantee allows us to use the lower bound result, Proposition 2, from the previous section to conclude that Algorithm 2's minimax query complexity is optimal up to log factors. Towards analyzing the algorithm performance (and inspired by a related measure in Hanneke [131] for the conventional non-abstention setting), we first introduce a new complexity measure named global identification cost (GIC), that will allow us to bridge Algorithm 2's performance to Cost.

Definition 7. *Given $\mathcal{H}, \mathcal{X}$, define the global identification cost of $V \subset \mathcal{H}$, instance set S_X as:*

$$\mathrm{GIC}(V, S_X) = \min\{t \in \mathbb{N} : \forall T : \mathcal{X} \setminus S_X \to \{+1, -1, \perp\}, \\ \exists \Sigma \subseteq \mathcal{X} \setminus S_X \text{ s.t. } \sum_{x \in \Sigma} \mathbb{1}(T(x) \neq \perp) \leq t \wedge |E(V[T(\Sigma)], S_X \cup \Sigma)| \leq 1\}.$$

Intuitively, GIC represents the worst-case sample complexity of a clairvoyant querying algorithm that knows ahead of time the labeling oracle that is used by the labeler.

The key lemma behind the analysis of Algorithm 2 is that there always exists a point that significantly bisects the current E-VS, resulting a size reduction of at least a constant $1 - \frac{1}{\mathrm{GIC}(V, S_X)}$ factor. This justifies greedily querying the point that maximally bisects the E-VS.

Lemma 1. *For any V, S_X such that $\mathrm{GIC}(V, S_X)$ is finite, $\exists x \in \mathcal{X} \setminus S_X$ such that:*

$$\max_{y \in \{-1, +1\}} (|E(V[(x, y)], S_X \cup \{x\}))| - 1) \leq (|E(V, S_X)| - 1)(1 - \frac{1}{\mathrm{GIC}(V, S_X)}).$$

To analyze the algorithm's query complexity, we lower bound $\mathrm{Cost}(V, S_X)$ by $\mathrm{GIC}(V, S_X)$.

Lemma 2. *For any $V \subset \mathcal{H}$ and $S_X \subset \mathcal{X}$:* $\mathrm{GIC}(V, S_X) \leq \mathrm{Cost}(V, S_X)$.

With this, we can prove that Algorithm 2: a) has query complexity of $O(\mathrm{Cost}(\mathcal{H}, \emptyset) \ln |\mathcal{H}|)$; b) identifies h^* when the labeler's labeling strategy is identifiable. Please see Appendix 2.9 for all the proofs.

Theorem 1 (Algorithm 2's query complexity guarantee)**.** *If Algorithm 2 interacts with a labeling oracle T, then it incurs total query cost at most* $\mathrm{GIC}(\mathcal{H}, \emptyset) \ln |\mathcal{H}| + 1 \leq \mathrm{Cost}(\mathcal{H}, \emptyset) \ln |\mathcal{H}| + 1$. *Furthermore, if Algorithm 2 interacts with an identifiable oracle T consistent with some $h^* \in \mathcal{H}$, then it identifies h^*.*

2.3.1 Accessing the E-VS

Algorithm 2 may be viewed as the E-VS variant of the well-known, VS bisection algorithm [278], an "aggressive" active learning algorithm that greedily queries the informative point that maximally bisects the VS. The canonical approach for accessing the VS is via sampling, by assuming access to a sampling oracle $\mathcal{O}$. For example, if $\mathcal{H}$ is linear, the VS is a single polytope and one can use a polytope sampler to evaluate and search for the point x that maximally bisects the VS.

E-VS Structure: Maximal E-VS bisection point search is less straightforward by contrast. The following structural lemma shows that there exists a setting of linear hypothesis classes in $\mathbb{R}^d$

Algorithm 2 E-VS Bisection Algorithm

```
Require: Data pool $\mathcal{X}$, hypothesis class $\mathcal{H}$
$V \leftarrow \mathcal{H}$, $S \leftarrow \emptyset$    ▷ VS, queried dataset
while $\left|E(V, S_X)\right| \geq 2$ and $S_X \neq \mathcal{X}$ do
    Query:    ▷ Maximal E-VS bisection point
        $x = \operatorname*{argmin}_{x \in \mathcal{X} \setminus S_X} \max_{y \in \{-1,+1\}} \left|E(V, S_X)[(x, y)]\right|$
    Labeler $T$ provides label response: $y \in \{-1, +1, \perp\}$
    $S \leftarrow S \cup \{(x, y)\}$
    if $y \neq \perp$ then
        $V \leftarrow V[(x, y)]$
return $h$, the unique element in $E(V, S_X)$
```

Algorithm 3 Bisection Point Search Subroutine

```
Require: Unqueried examples $U = \mathcal{X} \setminus S_X$, abstained examples $S_\perp$, Version Space $V$, sampling oracle $\mathcal{O}$
1: for sample $h \sim \mathcal{O}(V)$ do
2:     Construct $Z_1 = \{(x, -h(x)) : x \in S^\perp\}$, $Z_2 = \{(x, h(x)) : x \in \mathcal{X} \setminus S^\perp\}$
3:     Run C-ERM to obtain: $\hat{h} \in \operatorname{argmin}\{\mathrm{err}(h', Z_1) : h' \in \mathcal{H}, \mathrm{err}(h', Z_2) = 0\}$
4:     if $\hat{h} \neq h$ then continue
5:     else    ▷ $h \in E(V, S_X)$ in this case
6:         $r_x^- \leftarrow r_x^- + 1$ if $h(x) = -1$ else $r_x^+ \leftarrow r_x^+ + 1$ for $x \in U$, $n \leftarrow n + 1$
   return $x^* = \operatorname{argmin}_{x \in U} |r_x^+/n - r_x^-/n|$
```

with $\mathcal{X}$ and S such that the E-VS comprises of an *exponential* number of disjoint polytopes. This means that it is computationally intractable to access the E-VS as polytopes, if one is to use the sampling approach as in VS-bisection.

Proposition 3. *There exists an instance space $\mathcal{X} \subset \mathbb{R}^d$, a linear hypothesis class $\mathcal{H}$, and query response S such that the resultant E-VS comprises of an exponential in d number of disjoint polytopes.*

Towards tractable maximal E-VS bisection point search: To overcome this issue, we develop a novel, oracle-efficient method for accessing the E-VS. We observe that a structural property of the E-VS can be used to check membership given access to a constrained empirical risk minimization (C-ERM) oracle [84]. This allows us to design an oracle-efficient subroutine, Algorithm 3 for any general hypothesis class $\mathcal{H}$, which we prove is sound.

Definition 8. *A* constrained-ERM oracle *for hypothesis class $\mathcal{H}$,* C-ERM*, takes as input labeled datasets Z_1 and Z_2, and outputs a classifier: $\hat{h} \in \operatorname{argmin}_{h' \in \mathcal{H}} \{\mathrm{err}(h', Z_1) : \mathrm{err}(h', Z_2) = 0\}$, where for dataset Z, $\mathrm{err}(h', Z) = \sum_{(x,y) \in Z} \mathbb{1}(h'(x) \neq y)$.*

Proposition 4. *Given some $h \in V$ and access to a* C-ERM *oracle, lines 2to 4 in Algorithm 3 verifies whether $h \in E(V, S_X)$, with one call to the oracle.*

2.3.2 Comparing with the VS bisection algorithm

Labeling without identifiability: An advantage of the E-VS algorithm is its robustness to strategic labeling. Theorem 1 states that the E-VS algorithm has provable guarantees, *even when* the labeler does not guarantee identification. By contrast, VS-bisection is not robust this way. To concretely compare the two, we construct a learning setup without identification, wherein Algorithm 2 incurs a much smaller number of samples.

Theorem 2. *There exists a $\mathcal{H}$ and $\mathcal{X}$ such that the number of labeled examples queried by the E-VS bisection algorithm is $O(\log |\mathcal{X}|)$, while the VS bisection algorithm queries $\Omega(|\mathcal{X}|)$ labels.*

Remark: The key observation here is that, by *optimistically* assuming identifiability (even when this is not guaranteed), Algorithm 2 can ensure a small query cost. It does so by using the E-VS cardinality to detect when the labeling strategy is non-identifiable and halt the interaction.

Please refer to Appendix 2.10 for all proofs in these subsections and a comparison with EPI-CAL [146], a natural 'mellow" active learning algorithm that can handle labeler abstentions. Additionally, please see Appendix 2.15 for some toy experiments based on synthetic data.

2.4 Extensions to Other Learning Settings

The prior sections have assumed that the labeler (e.g. data labeling company) is resourcefully providing non-noisy, labeled data that exactly identifies h^*. In this section, we examine a few ways in which the labeler (e.g. a human worker) may be imperfect in labeling, and extend our guarantees to show how the learner may learn in such settings. Indeed, it is possible for the labeler to abstain *non-strategically* simply due to uncertainty (or lack of knowledge) about the label. As we will see, Algorithm 2 will also allow for efficient learning with non-strategic, abstaining labelers.

2.4.1 Approximate Identifiability

A relaxation of the goal of exact learning is PAC learning: learning some $\hat{h}$ such that its error $\Pr_{x\sim\mathcal{D}}(\hat{h}(x) \neq h^*(x)) \leq \epsilon$ on distribution $\mathcal{D}$ supported on $\mathcal{X}$, with probability (w.p.) greater than $1-\delta$. This learning goal can arise when the learner wishes to relax the learning outcome/termination criterion, or wishes to weaken the assumption that the labeler identifies h^*, to only knowing a fairly accurate hypothesis $\hat{h} \in \mathcal{H}$.

Reduction: To study the PAC setting, one may use the standard PAC to exact learning reduction [283]. It is well known that PAC learning can be reduced to to exact learning on a sub-sampled set, $X^m \subseteq \mathcal{X}$, of $m = O(\frac{\mathrm{VC}(\mathcal{H})}{\epsilon}(\ln\frac{1}{\epsilon} + \ln\frac{1}{\delta}))$ i.i.d points from $\mathcal{D}$ (VC($\mathcal{H}$) denotes the VC dimension of $\mathcal{H}$).

Then, X^m partitions $\mathcal{H}$ into *clusters* of equivalent hypotheses. Let the projection of $\mathcal{H}$ on X^m be $\mathcal{H}_{|X^m} = \big\{h(X^m) : h \in \mathcal{H}\big\}$. For $y \in \mathcal{H}_{|X^m}$, a cluster $C(y)$ of equivalent hypotheses may then be defined as $C(y) = \big\{h \in \mathcal{H} : h(X^m) = y\big\}$. The reduction guarantees that, w.p. over $1-\delta$ over the samples X^m, identifying h^*'s cluster $C(h^*(X^m))$ suffices for finding $\hat{h}$ with error $\leq \epsilon$.

Approximate Identification: Using this reduction, we may analyze the query complexity of approximate identification in the resulting learning game. In this game, the learner sets the data pool to be X^m (can be much smaller than $\mathcal{X}$) and aims to only learn *the cluster* h^* belongs to, $C(h^*(X^m))$.

We demonstrate how our E-VS representation can be adapted to apply Algorithm 2 in this approximate identification game. We first note that the original E-VS, defined over $\mathcal{H}$ and X^m will no longer suffice as state representation. Consider some $h \in \mathcal{H}$ such that $|C(h(X^m))| \geq 2$

with $\{h', h\} \subseteq C(h(X^m))$. Then, $h(X^m) = h'(X^m) \Rightarrow h'(X^m \setminus \emptyset) = h(X^m \setminus \emptyset)$, which results in the premature elimination of the entire $C(h(X^m))$ cluster at the very start.

To address this, we define a refinement of E-VS, X^m-E-VS. This fix follows from observing that in this game, we should only consider non-identifiability with respect to hypotheses from *other* clusters.

$$E^{X^m}(V, S_X) = \Big\{h \in V : \forall h' \in V \setminus \big\{\bar{h} : \bar{h}(X^m) = h(X^m), \bar{h} \in V\big\} : h'(X^m \setminus S_X) \neq h(X^m \setminus S_X)\Big\}$$

With this, we note that the X^m-E-VS bisection algorithm attains analogous near-optimal guarantees.

Corollary 1. *Consider Algorithm 2 instantiated with data pool X^m and state representation X^m-E-VS. When interacting with a labeling oracle T, it incurs total query cost at most* $\mathrm{GIC}^{X^m}(\mathcal{H}, \emptyset) \ln |\mathcal{H}| + 1$ *(see Definition 13). Furthermore, if the X^m-E-VS bisection algorithm interacts with an identifiable oracle T consistent with some $h^* \in \mathcal{H}$, then it identifies h^*.*

The only remaining consideration is how to efficiently search for the point that maximally bisects clusters in X^m-E-VS. Here, we show that we may adapt the membership check implemented in Algorithm 3 (with the data pool set to X^m) to check hypothesis membership in the coarser X^m-E-VS. That is, we still have an oracle-efficient way of accessing the X^m-E-VS, without needing to explicitly compute and iterate through the clusters.

Proposition 5. $h \notin E^{X^m}(V, S_X)$ *iff* $\hat{h}(X^m) \neq h(X^m)$*, where $\hat{h}$ is the minimizer of the C-ERM output on Algorithm 3, Line 3 with $\mathcal{X} = X^m$.*

2.4.2 Noised labeling

In some cases, a labeler can make honest mistakes simply due to human error. We can model this by assuming noised queries [57]: querying example x returns $h^*(x)$ w.p. $1-\delta(x)$, and $-h^*(x)$ w.p. $\delta(x)$. In this setup, we may use the common approach of repeatedly query a datum to estimate its label w.h.p. (e.g. as in [303]). This approach thus reduces the noised-label setting to cost-sensitive exact learning, where each x incurs differing cost $c(x)$ dependent on $\delta(x)$. In Appendix 2.9, we prove the generalized version of the results in Section 2.3 that factors in example-based cost, showing that Algorithm 2 can be applied in this setting with near-optimal guarantees.

2.4.3 Arbitrary labeling

Thus far, we have assumed a labeler who can (approximately) identify h^*. Here, we touch on when the labeler either does not know h^* (or h^*'s cluster), or myopically labels in a way that cannot guarantee the learning outcome. Since the labeler behaves arbitrarily, the learner now cannot be assured of any learning outcome guarantees. In this case, we note that the learner can use the E-VS to preemptively detect when the learning outcome cannot be realized, and halt the interaction. While the h^* is unknown, it is possible to detect when *no hypothesis/cluster* is learnable. This is when the E-VS is empty, certifying that the labeler cannot realize the learning outcome. Here, our Theorem 1 provides guarantees on the maximum number of times that a non-identifiable oracle will be queried.

Corollary 2 (of Theorem 1)**.** *Algorithm 2 guarantees bounded query complexity* $\mathrm{GIC}(\mathcal{H}, \emptyset) \ln |\mathcal{H}| + 1$ even when the *labeling oracle is non-identifiable.*

Finally, we note that our algorithm is sound in that if the labeler can identify h^*, then our algorithm learns h^*. Thus, in summary, Algorithm 2 is both sample-efficient with respect to an identifiable labeler, and robust to a non-identifiable one. Please see Appendix 2.11 for more details on this section.

2.5 Multi-Task learning from a Strategic Labeler

Multi-task setting: In most jobs, workers in fact perform multiple roles. This motivates the study of multi-task exact learning from a strategic labeler, which we now outline:

- The learner is now interested in learning multiple $h_i^* \in \mathcal{H}_i$, for tasks $i \in [n]$. Define learner's hypothesis class $\mathcal{H} = \times_{i=1}^n \mathcal{H}_i$ which contains $h^* = (h_1^*, \ldots, h_n^*)$. The learner can query from instance domain $\mathcal{X} \subseteq \times_{i=1}^n \mathcal{X}_i$, where $\mathcal{X}_i$ is the instance domain for task i.
- Labeler now provides multi-task labels $y \in \mathcal{Y}^n = \{+1, -1, \perp\}^n$, and for the label cost:

  i) One natural extension of the single task payoff is: $c_{one}(y) = \mathbb{1}(\exists i, y_i \neq \perp)$.

  ii) Another variant of the multi-task labeling payoff is: $c_{all}(y) = \mathbb{1}(\forall i, y_i \neq \perp)$.

We are interested in asking: can the labeler use the multi-task structure to *further* amplify the query complexity? To answer this question, we relate the multi-task query complexity to that of single-task.

Single-task setting:

- **Definition of** S_X^i**:** given queried data S_X, define the queried data for task i, S_X^i, as: $S_X^i = \mathcal{X}_i \setminus (\mathcal{X} \setminus S_X)_i$, where we use the notation that set $Z_i = \{x_i : x \in Z\}$ for $Z \subseteq \mathcal{X}$.

  In words, S_X^i are examples in $\mathcal{X}_i$ whose label can no longer be obtained. Note that in the multi-task setting, there may exist multiple points that can label some $x_i \in \mathcal{X}_i$. So abstention on one of those points does *not* necessarily mean that x_i cannot be labeled.

  Example: $\mathcal{X} = \{x_{11}, x_{12}\} \times \{x_{21}, x_{22}\}$. $S_X = \big\{[x_{11}, x_{21}], [x_{12}, x_{22}]\big\}$, then $S_X^i = \{\}$ for $i = 1, 2$. This is because it is still possible for the labeler to give labels on all points, i.e. x_{11}, x_{22} through $[x_{11}, x_{22}]$ and x_{12}, x_{21} through $[x_{12}, x_{21}]$.
- **Definition of** V_i**:** given the current multi-task version space V, we can naturally define the single-task version space for task i as: $(V)_i = V_i = \{h_i : h \in V\}$

2.5.1 Upper Bound

To understand if multi-task structure can inflate query complexity, we upper bound the multi-task complexity in terms of the sum of the single-task complexities. Proving an upper bound would imply that the labeler cannot increase the query complexity through the multi-task structure. We find that upper bounds only arise under certain regularity assumptions. Thus, we first provide complementary negative results without these assumptions, showing settings where the labeler *can* amplify the multi-task query complexity. All proofs in this section may be found in Appendix 2.12, where we also prove results in the non-abstention setting that may be of independent interest.

Proposition 6. *Under both label costs, there exists a non-Cartesian product version space $V \subseteq \mathcal{H}$ and query response $S \subseteq (\mathcal{X} \times \mathcal{Y})^*$ such that* $\mathrm{Cost}(V_i, S_X^i) \geq 0$ *for all i, and:* $\mathrm{Cost}(V, S_X) \geq \sum_{i=1}^n \mathrm{Cost}(V_i, S_X^i) + n - 1$.

Furthermore, we show that if the version space is allowed to be a Cartesian product, and the (more generous) c_{one} is used as label cost, the labeler can still increase the query complexity.

Proposition 7. *Assuming the version space is a Cartesian product, under label cost $c_{one}(y) = \mathbb{1}(\exists i, y_i \neq \perp)$, there exists V and S such that* $\mathrm{Cost}(V_i, S_X^i) = 1$, *but* $\mathrm{Cost}(V, S_X) = |\mathcal{X}|$. *This implies that:* $\mathrm{Cost}(V, S_X) > \sum_{i=1}^n \mathrm{Cost}(V_i, S_X^i)$.

Thus, for the labeler to be unable to increase multi-task query complexity, two necessary conditions are a) the VS is a cartesian product b) the payoff cost is c_{all} (and not c_{one}). Below, we prove the two conditions are sufficient, providing a full characterization when the upper bound can be achieved.

Theorem 3. *For all $V = \times_{i \in [n]} V_i$ and $S_X \subseteq \mathcal{X}$, under labeling cost $c_{all}(y) = \mathbb{1}(\forall i, y_i \neq \perp)$,* $\mathrm{Cost}(V, S_X) \leq \sum_{i=1}^n \mathrm{Cost}(V_i, S_X^i)$.

For the remainder of the section, we will prove results under the (more generous) label cost, c_{one}.

2.5.2 Lower Bound

Through lower bounds, we illustrate that the multi-task version space structure can in fact speed up learning as well. The intuition is that the structure in V may make it so that the multi-task E-VS shrinks faster due to unidentifiability. The following negative example evidences this.

Proposition 8. *There exists a non-Cartesian product version space V and query response S such that* $\mathrm{Cost}(V_i, S_X^i) \geq 0$ *for all i, but:* $\mathrm{Cost}(V, S_X) < \max_{i \in [n]} \mathrm{Cost}(V_i, S_X^i)$.

Proposition 9. *There exists a Cartesian product version space V and query response S with* $\mathrm{Cost}(V, S_X) < 0$ *such that:* $\mathrm{Cost}(V, S_X) < \max_{i \in [n]} \mathrm{Cost}(V_i, S_X^i)$.

Thus, we have that identifiability ($\mathrm{Cost}(V, S_X) \geq 0$), and V being a Cartesian product are needed to prove a lower bound.

Theorem 4. *For all $V = \times_{i \in [n]} V_i$ and $S_X \subseteq \mathcal{X}$, if* $\mathrm{Cost}(V, S_X) \geq 0$, *then:* $\mathrm{Cost}(V, S_X) \geq \max_{i \in [n]} \mathrm{Cost}(V_i, S_X^i)$.

2.6 Related Works

The theory of Active Learning [133] (AL) has a rich history and began with the study of realizable learning [13, 81, 83, 109, 138]. To the best of our knowledge, we are the first to consider a labeler whose objective is *at odds with* the learner. In face of such a strategic labeler, we develop an active learning algorithm with near-optimal query complexity guarantees.

Abstaining Labeler: The closest two papers to our work are Huang et al. [146], Yan et al. [303], which also study learning from an abstaining labeler. In Yan et al. [303], the labeler can abstain or noise, where the rate of an incorrect label/abstention is fixed apriori. Our work differs from that of Yan et al. [302, 303] in that the labeler can adaptively label (abstain) based on the full interaction history so far, thus allowing for more complex, sequential labeling strategies. In Huang

Notation	
S	$S = \big\{(x_1, y_1), (x_2, y_2), ...\big\}$, query responses in the interaction history
S_X	$S_X = \big\{x : (x, y) \in S\big\}$, indexes the queried examples in S
$S^{\perp}$	$S^{\perp} = \big\{x : (x, y) \in S, y = \perp\big\}$, queried examples that were given abstention
$V_x^y, V[(x, y)]$	$V_x^y, V[(x, y)] = \big\{h \in V : h(x) = y\big\}$, updated VS (used interchangeably)
$E(V, S_X)$	$E(V, S_X) = \big\{h \in V : \forall h' \in V \setminus \{h\} : h'(\mathcal{X} \setminus S_X) \neq h(\mathcal{X} \setminus S_X)\big\}$, effective VS
$S_{\mathcal{A},T}$	Interaction history between $\mathcal{A}$ and T
A_i	$A_i = \{x_i : i \in A\}$
S_X^i	$S_X^i = \mathcal{X}_i \setminus (\mathcal{X} \setminus S_X)_i$
$(V)_i$	$(V)_i = V_i = \{h_i : h \in V\}$
$c_{one}(y)$	$c_{one}(y) = \mathbb{1}(\exists i, y_i \neq \perp)$
$c_{all}(y)$	$c_{all}(y) = \mathbb{1}(\forall i, y_i \neq \perp)$

Table 2.2: Table of commonly used notation.

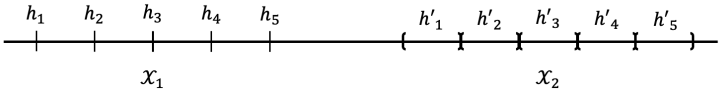


Figure 2.1: The setup behind Proposition 1 is that of learning an one-to-one threshold-interval hypothesis class $\mathcal{H} = \big\{(h_i, h_i')\big\}_{i \in [n]}$. The learner seeks to identify $(h_{i^*}, h_{i^*}^{'})$. The labeler can abstain on $\mathcal{X}_1$, and prevent the learner from learning through this sample-efficient part of the instance space. This forces the learner to learn the interval $h_i^{'*}$ (instead of threshold h_i^*) through $\mathcal{X}_2$, and incur much larger sample complexity.

et al. [146], the labeler abstains when uninformed, and after a number of abstentions in a region, learns to label the region (an "epiphany"). Our setting differs in that the labeler does know the labels for all regions, but instead strategically abstains to increase query complexity. Please see Appendix 2.14 for further discussion on related works and on alternative formulations of the learning game, including when the learner is allowed to query an example multiple times.

2.7 Proofs for Section 2.1

2.7.1 Technical Results

Proposition 10. *There exists a hypothesis class $\mathcal{H}$, instance domain $\mathcal{X}$ such that the exact learning sample complexity is $O(\log |\mathcal{X}|)$ if the labeler is unable to abstain, and $\Omega(|\mathcal{X}|)$ for any learning algorithm if the labeler is allowed to abstain.*

Proof. Let the $h_i : [0, 1] \to \{+1, -1\}$ for $i \in [n]$ denote intervals of length $1/n$ centered at $(2i - 1)/2n$ for $i \in [n]$, and $h_i' : (1, 2] \to \{+1, -1\}$ for $i \in [n]$ denote thresholds at $1 + i/n$ for

$i \in [n]$. Define hybrid-hypothesis class $\mathcal{H}$ of threshold-intervals, $\mathcal{H} = \{f_1, ..., f_n\}$, where:

$$f_i(x) = \begin{cases} h_i(x) & x \in [0,1] \\ h'_i(x) & x \in (1,2] \end{cases}$$

Let $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$, where $\mathcal{X}_1 = \left\{\frac{1}{2n}, ..., \frac{2n-1}{2n}\right\}$ and $\mathcal{X}_2 = \left\{1 + \frac{3}{2n}, ..., 1 + \frac{2n-1}{2n}\right\}$.

1) When the labeler is not allowed to abstain, the learner may binary search on $\mathcal{X}_2$ to identify h'_{i^*}, which identifies f_{i^*}. The required sample complexity is $O(\log n)$.

2) When the labeler is allowed to abstain, consider the following labeling strategy T:

i) $T(x) = \perp$ for all $x \in \mathcal{X}_2$

ii) $T(x) = h_{i^*}(x)$ for all $x \in \mathcal{X}_1$.

Note T is a labeling strategy that allows for identification. $\mathcal{H}[T(\mathcal{X})] = \mathcal{H}[T(\mathcal{X}_1)] = \{f_{i^*}\}$.

Interacting with T is equivalent to learning one of n disjoint intervals, which requires $\Omega(n)$ samples under any learning algorithm [81]. And so, T induces $\Omega(n)$ samples, which in turn lower bounds the sample complexity induced by the minimax labeling strategy. □

Remark 1. *We note that one may generalize the above result to any cross-space learning setting [274] with significant differences in query complexity among the instance spaces.*

The labeler's optimal strategy here is simple: label only through the instance space that leads to the highest query complexity, and abstain on all other (more informative) instance spaces.

Remark 2. *We also add that the labeling strategy need not be identifiable for this result to hold. One can simply define T to still abstain on all of $\mathcal{X}_2$ and output -1 on all of $\mathcal{X}_1$, which still induces $\Omega(|\mathcal{X}|)$ query complexity.*

2.8 Proofs for Section 2.2

2.8.1 The Minimax Learning Game

We present Protocol 4, which can be viewed as a relaxation of the original Protocol 1 by allowing h^* to be chosen aposteriori. This gives the labeler more freedom in answering the learner's queries, and therefore any query complexity upper bound here translates to query complexity upper bounds in Protocol 1. Recall that the optimal value function of this game is given in (2.1).

2.8.2 Preliminaries

We now come back to Protocol 1. The game strategy for the labeler and learner now corresponds to a labeling oracle, and a querying algorithm, which we formally define below.

Labeling Oracle Notation: Given $h \in \mathcal{H}$, recall that we define the set of labeling oracles consistent with h as,

$$\mathcal{T}_h = \{T : \mathcal{X} \to \{+1, -1, \perp\} | \forall x \in \mathcal{X} \text{ s.t } T_h(x) \neq \perp, T(x) = h(x)\}$$

Protocol 4 Minimax strategic slow learning game

```
Require: Instance domain $\mathcal{X}$, hypothesis class $\mathcal{H}$
  $S \leftarrow \emptyset$, $V \leftarrow \mathcal{H}$
  ▷ Throughout, the labeler needs to maintain that there is at least one classifier consistent with all labels so far and is identifiable
  while $|E(V, S_X)| \geq 2$ do
      Learner queries example $x \in \mathcal{X} \setminus S_X$
      Labeler provides label feedback $y \in \{-1, +1, \perp\}$
      Learner incurs cost $c(y)$, and updates its version space $V \leftarrow V_x^y$
      $S \leftarrow S \cup \{(x, y)\}$
  Nature sets $h^*$ to be the only model in $E(V, S_X)$ if $|E(V, S_X)| = 1$    ▷ Nature sides with the labeler, sets $h^*$ to be the remaining model at the end
```

Given subset $S_X \subseteq \mathcal{X}$, let us define $T(S_X)$ to be the set of labeled examples induced by oracle T on the examples S_X.

Suppose $V \subseteq \mathcal{H}$, let us define:

$$V[T(S_X)] = \{h \in V | h(x) = T(x), \forall x \in S_X \wedge T(x) \neq \perp\}$$

A labeling strategy $T \in \mathcal{T}_h$ is an identifiable oracle if $\mathcal{H}[T(\mathcal{X})] = \{h\}$.

Querying Algorithm Notation: Formally, a deterministic learning algorithm $\mathcal{A}$ consists of the following:

- Query function $f_{query} : (\mathcal{X} \times \mathcal{Y})^* \to \mathcal{X}$
- Termination function $f_{term} : (\mathcal{X} \times \mathcal{Y})^* \to \{\text{TRUE}, \text{FALSE}\}$
- Output function $f_{out} : (\mathcal{X} \times \mathcal{Y})^* \to \mathcal{H}$

$\mathcal{A}$ interacts with the labeler by:

Algorithm 5 The interaction process between $\mathcal{A}$ and labeler

```
  $S \leftarrow \emptyset$
  while $f_{term}(S) = \text{FALSE}$ do
      Query $x \leftarrow f_{query}(S)$
      Receive label $y$
      $S \leftarrow S \cup \{(x, y)\}$
  return $f_{out}(S)$
```

Properties of f_{term}:

- If $\mathcal{A}$ is an exact learning algorithm, $f_{term}(S) = \text{TRUE}$ if $|E(V, S_X)| \leq 1$.
- If $\mathcal{A}$ has a fixed budget N, f_{term} outputs TRUE when S is such that: $|\{(x, y) \in S : y \neq \perp\}| = N$

The formal interaction process between the learner using $\mathcal{A}$ and the labeler is summarized in Algorithm 5.

Learning Game Payoff: Denote $\mathrm{Cost}_{\mathcal{A},T}(V, S_X)$ as the learning game payoff under an exact learning querying strategy $\mathcal{A}$ and labeling strategy T. Formally, let point $x_{\mathcal{A},S}$ be queried by $\mathcal{A}$ after seeing interaction history S (corresponding to some sequentially labeled dataset) induced by labeling oracle T. With this, the value function of the learning game with strategies $\mathcal{A}$ and T may be recursively defined as follows:

$$\mathrm{Cost}_{\mathcal{A},T}(V, S_X) = \begin{cases} -\infty & E(V, S_X) = \emptyset \\ 0, & |E(V, S_X)| = 1 \\ \mathbb{1}(T(x_{\mathcal{A},S}) \neq \perp) + \mathrm{Cost}(V[(x_{\mathcal{A},S}, T(x_{\mathcal{A},S}))], S_X \cup \{x_{\mathcal{A},S}\}) & |E(V, S_X)| \geq 2, \end{cases}$$

2.8.3 Technical Results

Lemma 3. *Let the deterministic query algorithm $\mathcal{A}$ interact with labeling oracle $T \in \mathcal{T}_{h_0}$ for M queries, generating the following interaction history: $S_M = (x_1, T(x_1)), (x_2, T(x_2)), ..., (x_M, T(x_M))$. Suppose there exists a classifier h_1 and $T' \in \mathcal{T}_{h_1}$ such that for all $x \in \{x_1, ..., x_M\}$, $T(x_i) = T'(x_i)$. Then, $\mathcal{A}$ generates the same interaction history, when interacting with T' for M queries.*

Proof. As defined previously, algorithm $\mathcal{A}$ comprises of query function f_{query}, termination function f_{term} and output function f_{out}. We show by induction that for steps $i = 0, 1, ..., M$, the interaction histories of $\mathcal{A}$ with T and T' agree on their first i elements for $i \leq M$.

Base Case: For step $i = 0$, both interaction histories are empty and thus agree.

Induction Step: Suppose the statement holds up until step i for some $i < M$. That is, when $\mathcal{A}$ interacts with T and T' generates the same set of queried examples:

$$S_i = \{(x_1, y_1), ..., (x_i, y_i)\}$$

Consider step $i+1$. Firstly, $\mathcal{A}$ continues to make a query and does not terminate, since $f_{term}(S_i) = \mathrm{FALSE}$ for $i < M$.

Now, for the $(i+1)$-th query, $\mathcal{A}$ applies function f_{query} and queries $x_{i+1} = f_{query}(S_i)$. Since $T'(x_j) = T(x_j)$ for all j and in particular for $j = i + 1$, we have that $(x_{i+1}, T'(x_{i+1})) = (x_{i+1}, T(x_{i+1}))$. And so, with this and the induction hypothesis, we have that $\mathcal{A}$ when interacting with T' and T generates the same set of queried examples:

$$S_{i+1} = \{(x_1, y_1), ..., (x_{i+1}, y_{i+1})\}$$

up to step $i + 1$.

Using this, we can conclude that the interaction histories after M steps of $\mathcal{A}$ with T' and T are identical. □

Remark 3. *Suppose, after the Mth step, we have that $\mathrm{TRUE} = f_{term}(S_{\mathcal{A},T}) = f_{term}(S_M)$. And so, we have that $S_M = S_{\mathcal{A},T'}$, and the interaction of $\mathcal{A}$ with T' also terminates at the Mth step.*

Thus, for model output, we have $S_{\mathcal{A},T} = S_M = S_{\mathcal{A},T'} \Rightarrow f_{out}(S_{\mathcal{A},T}) = f_{out}(S_{\mathcal{A},T'})$.

Proposition 11. *Let N denote the labeling budget. Let $S_N^{\mathcal{A},T}$ be the interaction history of a deterministic algorithm $\mathcal{A}$ with oracle T up until the Nth label is given, or at termination (without using all of the budget). Let $(S_X)_N^{\mathcal{A},T}$ be the unlabeled examples queried during the interaction. For any deterministic algorithm $\mathcal{A}$, if $N < \mathrm{Cost}(\mathcal{H}, \emptyset)$, there exists some $h \in \mathcal{H}$ and identifiable oracle $T \in T_h$ such that $|E(\mathcal{H}[S_N^{\mathcal{A},T}], (S_X)_N^{\mathcal{A},T})| \geq 2$.*

Proof. Fix a deterministic algorithm $\mathcal{A}$. We will show the following. If $\mathcal{A}$ has already obtained an ordered sequence of queried examples S, and has a remaining label budget $N \leq \mathrm{Cost}(\mathcal{H}[S], S_X) - 1$, then there exists $h \in \mathcal{H}[S]$ and T_h such that, $\mathcal{A}$, when interacting with T_h:

1. obtains a sequence of queried examples S in the first $|S|$ rounds
2. when the interaction terminates, the E-VS has cardinality at least two: $|E(\mathcal{H}[S_N^{\mathcal{A},T_h}], (S_X)_N^{\mathcal{A},T_h})| \geq 2$.

The theorem follow from the second point of this claim by taking $S = \emptyset$.

We now turn to proving the above claim by induction on $\mathcal{A}$'s remaining label budget N.

Base Case: If $N = 0$, then $\mathrm{Cost}(\mathcal{H}[S], S_X) \geq 1$. By Lemma 8, we know that $|E(\mathcal{H}[S], S_X)| \geq 2$.

Construction of T_h:

Let $h \in E(\mathcal{H}[S], S_X)$.

Define T_h to be such that for $(x_i, y_i) \in S$, $T_h(x_i) = y_i = h(x_i)$ (the latter equality holds by definition of h) if $y_i \neq \perp$ and $T_h(x_i) = \perp$ if $y_i = \perp$, and define $T_h(x) = h(x)$ for all $x \in \mathcal{X} \setminus S_X$.

Since $h \in E(\mathcal{H}[S], S_X)$, we know that $h(\mathcal{X} \setminus S_\perp) \neq h'(\mathcal{X} \setminus S_\perp), \forall h' \neq h \in V$. And so, $\mathcal{H}[T(\mathcal{X})] = \mathcal{H}[T(\mathcal{X} \setminus S_\perp)] = \{h\}$, which implies that T is an identifiable oracle for h.

By construction and using Lemma 3, T_h's interaction with $\mathcal{A}$ results in S, satisfying the first item. Moreover, since $N = 0$, $S_0^{\mathcal{A},T_h} = S$. And so, $|E(\mathcal{H}[S_0^{\mathcal{A},T_h}], (S_X)_0^{\mathcal{A},T_h})| = |E(\mathcal{H}[S], S_X)| \geq 2$.

Induction Step: Suppose the claim holds for all $N \leq n$ for some $0 \leq n < \mathrm{Cost}(\mathcal{H}, \emptyset) - 1$.

Now, suppose during the interaction, algorithm $\mathcal{A}$ has remaining budget $N = n + 1$, and the obtained queried examples history S is such that $\mathrm{Cost}(\mathcal{H}[S], S_X) \geq N + 1 = n + 2$.

Our goal is to show the existence of h and T_h that satisfy the two listed properties under these two assumptions.

Define x'_j for index $j \geq 1$ to be the next example $\mathcal{A}$ queries such that a binary label y'_j is given (i.e $y'_j \neq \perp$), as we recursively unroll the Cost expression, via the construction procedure below.

Algorithm 6 The construction procedure for (x'_j, y'_j)

```
$L \leftarrow S$, $L_X \leftarrow S_X$, $j \leftarrow 1$
repeat
    Query $x'_k \leftarrow f(L)$ using $\mathcal{A}$
    Labeler return $y'_k = \mathrm{argmax}_{y \in \{-1,+1,\perp\}} \left( \mathbb{1}(y \neq \perp) + \mathrm{Cost}(\mathcal{H}[L \cup \{(x'_k, y)\}], L_X \cup \{x'_k\} \right)$
    $L \leftarrow L \cup \{(x'_k, y'_k)\}$
    $L_X \leftarrow L_X \cup \{x'_k\}$
until $y'_j \neq \perp$ or $f_{term}(L) = \mathrm{TRUE}$
```

There are two cases:

- If the final j satisfies $y'_j \neq \perp$, then after querying $\left\{(x'_i, y'_i)\right\}_{1:j}$, the learner has a remaining budget of $N - 1 = n$.
  Next, we see that with each abstention, the Cost value is non-decreasing, as justified in the first three steps:
  We have that:

$$\begin{aligned}
\mathrm{Cost}(\mathcal{H}[S], S_X) &\leq \max_{y_1 \in \{+1,-1,\perp\}} \mathbb{1}(y_1 \neq \perp) + \mathrm{Cost}(\mathcal{H}[S \cup \left\{(x'_1, y_1)\right\}], S_X \cup \left\{x'_1\right\}) \\
&= \mathbb{1}(y'_1 \neq \perp) + \mathrm{Cost}(\mathcal{H}[S \cup \left\{(x'_1, y'_1)\right\}], S_X \cup \left\{x'_1\right\}) \\
&= \mathrm{Cost}(\mathcal{H}[S \cup \left\{(x'_1, y'_1)\right\}], S_X \cup \left\{x'_1\right\}) \\
&\leq ... \quad \text{(unroll from } j-1 \text{ to 1, using } \mathbb{1}(y'_i \neq \perp) = 0 \text{ for } i < j \text{ and } \diamond) \\
&\leq \mathbb{1}(y'_j \neq \perp) + \mathrm{Cost}(\mathcal{H}[S \cup \left\{(x'_i, y'_i)\right\}_{1:j}], S_X \cup \left\{x'_i\right\}_{1:j}) \\
&= 1 + \mathrm{Cost}(\mathcal{H}[S \cup \left\{(x'_i, y'_i)\right\}_{1:j}], S_X \cup \left\{x'_i\right\}_{1:j}) \qquad (2.2)
\end{aligned}$$

  $(\diamond)$: We may use the non-decreasingness property to unroll, because from non-decreasingness, for all $l \leq j$, $\mathrm{Cost}(\mathcal{H}[S \cup \left\{(x'_i, y'_i)\right\}_{1:l}], S_X \cup \{x'_i\}_{1:l}) = \mathrm{Cost}(\mathcal{H}[S], S_X) \geq n + 2 \geq 2$. Therefore, $\left|E(\mathcal{H}[S \cup \left\{(x'_i, y'_i)\right\}_{1:l}], S_X \cup \{x'_i\}_{1:l})\right| \geq 2$, and we have that:

$$\begin{aligned}
&\mathrm{Cost}(\mathcal{H}[S \cup \left\{(x'_i, y'_i)\right\}_{1:l}], S_X \cup \left\{x'_i\right\}_{1:l}) = \\
&\quad \min_x \max_y \mathbb{1}(y \neq \perp) + \mathrm{Cost}(\mathcal{H}[S \cup \left\{(x'_i, y'_i)\right\}_{1:l} \cup \left\{(x, y)\right\}], S_X \cup \left\{x'_i\right\}_{1:l} \cup \{x\})
\end{aligned}$$

  Continuing (2.2), we get that:

$$n \leq \mathrm{Cost}(\mathcal{H}[S], S_X) - 2 \leq \mathrm{Cost}(\mathcal{H}[S \cup \left\{(x'_i, y'_i)\right\}_{1:j}], S_X \cup \left\{x'_i\right\}_{1:j}) - 1$$

  By induction hypothesis, there exists $h \in \mathcal{H}[S \cup \left\{(x'_i, y'_i)\right\}_{1:j}]$ and T_h, such that when $\mathcal{A}$ interacts with T_h (after obtaining query history $S \cup \left\{(x'_i, y'_i)\right\}_{1:j}$) and with label budget n, the final version space is of cardinality at least two:

$$|E(\mathcal{H}[S_N^{\mathcal{A},T_h}], (S_X)_N^{\mathcal{A},T_h})| \geq 2$$

  In addition, when interacting with T_h, $\mathcal{A}$ obtains history $S \cup \left\{(x'_i, y'_i)\right\}_{i=1}^{j}$ in its first $|S| + j$ rounds of interaction, which implies that it obtains example sequence S in its first $|S|$ rounds of interaction with T_h. This proves the first property also holds and completes the induction.
- Now, we consider the case the final j satisfies $y'_j = \perp$. This means that the other exit condition must hold: $f_{term}(L) = \mathrm{TRUE}$. And so, $\mathcal{A}$ terminates with all abstentions: $y'_i = \perp$ for $i \in [j]$.
  As above, we iteratively use the non-decreasingness of Cost with abstention $y'_i = \perp$ to get that:

$$n + 2 \leq \mathrm{Cost}(\mathcal{H}[S], S_X) \leq ... \leq \mathrm{Cost}(\mathcal{H}[L], L_X)$$

for the final state $\mathcal{H}[L], L_X$.
From this, we have that $|E(\mathcal{H}[L], L_X)| \geq 2$.
Pick some $h \in E(\mathcal{H}[L], L_X)$. As in the prior T_h construction, define T_h so that: $T_h(x) = y$ for all $(x, y) \in L$, and $T_h(x) = h(x)$ for all $x \in \mathcal{X} \setminus L_X$.
By construction and Lemma 3, T_h's interaction with $\mathcal{A}$ induces L.
Since $f_{term}(L) = \text{TRUE}$, $S_N^{\mathcal{A},T} = L$. And so, $|E(\mathcal{H}[S_N^{\mathcal{A},T_h}], (S_X)_N^{\mathcal{A},T_h})| = |E(\mathcal{H}[L], L_X)| \geq 2$, satisfying the second condition.
Finally, since $\mathcal{A}$'s interaction with T_h generates L, the first $|S|$ steps also matches S. This satisfies the first property. □

Proposition 12. *For any deterministic, exact learning algorithm $\mathcal{A}$,*

$$\max_{h \in \mathcal{H}, T \in \mathcal{T}_h} \text{Cost}_{\mathcal{A},T}(\mathcal{H}, \emptyset) \geq \text{Cost}(\mathcal{H}, \emptyset)$$

Proof. From Prop. 11, we know that for $N = \text{Cost}(\mathcal{H}, \emptyset) - 1$, there exists some $h \in \mathcal{H}$ and $T \in \mathcal{T}_h$ such that $|E(\mathcal{H}[S_N^{\mathcal{A},T}], (S_X)_N^{\mathcal{A},T})| \geq 2$.

We construct a labeling strategy T' that yields at least $N + 1$ binary labeled examples as follows:

1. Let $T'(x) = T(x)$ for $x \in S_N^{\mathcal{A},T}$.
2. Let $T'(x) = h(x)$ for $x \in \mathcal{X} \setminus S_N^{\mathcal{A},T}$.

Note that T' is an identifiable oracle for h by construction.
And so, we have that:

$$\begin{aligned}
\max_{h \in \mathcal{H}, T \in \mathcal{T}_h} \text{Cost}_{\mathcal{A},T}(\mathcal{H}, \emptyset) &\geq \text{Cost}_{\mathcal{A},T'}(\mathcal{H}, \emptyset) \\
&= N + \text{Cost}_{\mathcal{A},T'}(\mathcal{H}[S_N^{\mathcal{A},T'}], (S_X)_N^{\mathcal{A},T'}) && (\diamond) \\
&= \text{Cost}(\mathcal{H}, \emptyset) - 1 + \text{Cost}_{\mathcal{A},T'}(\mathcal{H}[S_N^{\mathcal{A},T'}], (S_X)_N^{\mathcal{A},T'}) \\
&\geq \text{Cost}(\mathcal{H}, \emptyset) - 1 + 1 && (\diamond\diamond)
\end{aligned}$$

Two steps in the above derivation are justified as follows:
$(\diamond)$: Since $T'(x) = T(x)$ for $x \in S_N^{\mathcal{A},T}$, by Lemma 3, we must have that $S_N^{\mathcal{A},T'} = S_N^{\mathcal{A},T}$, and $(S_X)_N^{\mathcal{A},T'} = (S_X)_N^{\mathcal{A},T}$.

In particular, note that this implies $|E(\mathcal{H}[S_N^{\mathcal{A},T'}], (S_X)_N^{\mathcal{A},T'})| = |E(\mathcal{H}[S_N^{\mathcal{A},T}], (S_X)_N^{\mathcal{A},T})| \geq 2$.

$(\diamond\diamond)$: Since $\mathcal{A}$ is an exact learning algorithm, it does not terminate at the $|S_N^{\mathcal{A},T'}|$th step, because $|E(S_N^{\mathcal{A},T'}, (S_X)_N^{\mathcal{A},T}))| \geq 2$.

And so, $\mathcal{A}$ will make at least one more query on some $x \in \mathcal{X} \setminus S_N^{\mathcal{A},T'}$. Since $T'(x) \neq \perp$ for any $x \in \mathcal{X} \setminus S_N^{\mathcal{A},T'}$, and T' is identifiable (yielding terminal cost 0), we have that $CC_{\mathcal{A},T'}(\mathcal{H}[S_N^{\mathcal{A},T'}], (S_X)_N^{\mathcal{A},T'}) \geq 1$. □

2.9 Proofs for Section 2.3

2.9.1 Example-dependent Cost Setting: Definitions

In this section, we consider the following generalization of our learning setting that allows each binary label to have varying cost dependent on the feature x:

- A cost function $c : \mathcal{X} \to (0, +\infty)$ is known to both the learner and the labeler ahead of time.
- The learner is interested in learning a hypothesis h^* in hypothesis class $\mathcal{H} \subset (\mathcal{X} \to \{+1, -1\})$ over a finite pool of unlabeled data $\mathcal{X}$, collected by the learner. A cost function
- The labeler knows h^*, and responds using labeling strategy T with response $T(x) \in \left\{h^*(x), \perp\right\}$.
- The learner repeatedly interacts with the labeler adaptively, and makes label queries on unqueried example x, and incurs cost $c(x)$ if $T(x) \neq \perp$, and cost 0 otherwise.

Note that the setting studied in the main text is a special case with cost function $c \equiv 1$. We aim to analyze the following generalization of Algorithm 2:

Algorithm 7 E-VS Bisection Algorithm

```
Require: Data pool $\mathcal{X}$, hypothesis class $\mathcal{H}$
$V \leftarrow \mathcal{H}$, $S \leftarrow \emptyset$                ▷ VS, queried dataset
while $\left|E(V, S_X)\right| \geq 2$ and $S_X \neq \mathcal{X}$ do
    Query:                                                        ▷ Maximal E-VS bisection point
        $x = \operatorname*{argmin}_{x \in \mathcal{X} \setminus S_X} \max_{y \in \{-1,+1\}} \frac{\left|E(V, S_X)[(x, y)]\right|}{c(x)}$
    Labeler $T$ provides label response: $y \in \{-1, +1, \perp\}$
    $S \leftarrow S \cup \left\{(x, y)\right\}$
    if $y \neq \perp$ then
        $V \leftarrow V[(x, y)]$
return $h$, the unique element in $E(V, S_X)$
```

For the analysis below, we slightly abuse notation and let $c(x, y)$ denote to $c(x)\mathbb{1}(y \neq \perp)$, the cost of querying example x and receiving label feedback y.

Definition 9 (Generalization of Definition 7)**.** *Given $\mathcal{H}, \mathcal{X}$ and cost c, define the global identification cost of version space $V \subset \mathcal{H}$ and example set S as*

$$\begin{aligned}\mathrm{GIC}(V, S_X) = \inf\{t \in \mathbb{R} : \forall T : \mathcal{X} \setminus S_X \to \{-1, +1, \perp\}\,, \\ \exists \Sigma \subseteq \mathcal{X} \setminus S_X \textit{ s.t. } \sum_{x \in \Sigma} c(x, T(x)) \leq t \wedge |E(V[T(\Sigma)], S_X \cup \Sigma)| \leq 1\}.\end{aligned}$$

Definition 10. *Define* $\Gamma_{V,S_X} : \mathbb{N} \to \{\text{TRUE}, \text{FALSE}\}$ *as:*

$$\begin{aligned}\Gamma_{V,S_X}(t) = \{ & \forall T : \mathcal{X} \setminus S_X \to \{-1, +1, \perp\}, \exists \Sigma \subseteq \mathcal{X} \setminus S_X \textit{ s.t.} \\ & \sum_{x \in \Sigma} c(x, T(x)) \le t \wedge |E(V[T(\Sigma)], S_X \cup \Sigma)| \le 1\}\end{aligned}$$

Note that Γ_{V,S_X} is monotonically increasing: for $t_1, t_2 \in \mathbb{N}$, if $t_1 < t_2$, then $\Gamma_{V,S_X}(t_1) \to \Gamma_{V,S_X}(t_2)$. Also, with this notation, $\text{GIC}(V, S_X) = \inf \left\{t : \Gamma_{V,S_X}(t) = \text{TRUE}\right\}$.

We have the following definition of all possible cumulative cost values that can appear in the learning process.

Definition 11. *Define* $C = \left\{\sum_{x \in S} c(x) : S \subset \mathcal{X}\right\}$.

Note that C is a finite set since $\mathcal{X}$ is finite.

The following lemma implies that the set $\left\{t : \Gamma_{V,S_X}(t) = \text{TRUE}\right\}$ is a left-closed interval.

Lemma 4. *If* $\{t_n\} \downarrow t$ *and* $\Gamma_{V,S_X}(t_n) = \text{TRUE}$ *for all* n*, then* $\Gamma_{V,S_X}(t) = \text{TRUE}$.

Proof. Since $\{t_n\} \downarrow t$ and C is a finite set, there exists n large enough such that for any z,

$$z \in C \wedge z \le t_n \implies z \le t.$$

Importantly, since for any $T : \mathcal{X} \setminus S_X \to \{-1, +1, \perp\}, \Sigma \subset \mathcal{X} \setminus S_X$, $\sum_{x \in \Sigma} c(x, T(x)) \in C$, we have:

$$\sum_{x \in \Sigma} c(x, T(x)) \le t_n \implies \sum_{x \in \Sigma} c(x, T(x)) \le t$$

and therefore, for any $T : \mathcal{X} \setminus S_X \to \{-1, +1, \perp\}$, there exists $\Sigma \subset \mathcal{X} \setminus S_X$ such that $\sum_{x \in \Sigma} c(x, T(x)) \le t_n$ (and thus $\sum_{x \in \Sigma} c(x, T(x)) \le t$) and $|E(V[T(\Sigma)], S_X \cup \Sigma)| \le 1$, proving that $\Gamma_{V,S_X}(t) = \text{TRUE}$. □

Remark 4. *The above lemma implies that in the definition of* GIC*, the infimum is achieved in the set* $\left\{t : \Gamma_{V,S_X}(t) = \text{TRUE}\right\}$*. In other words,*

$$\text{GIC}(V, S_X) = \min \left\{t : \Gamma_{V,S_X}(t) = \text{TRUE}\right\}.$$

And therefore,

$$\begin{aligned}& \text{GIC}(V, S_X) \le N \\ \iff & \Gamma_{V,S_X}(N) = \text{TRUE} \\ \iff & \forall T : \mathcal{X} \setminus S_X \to \{-1, +1, \perp\}, \exists \Sigma \subseteq \mathcal{X} \setminus S_X, \sum_{x \in \Sigma} c(x, T(x)) \le N \wedge |E(V[T(\Sigma)], S_X \cup \Sigma)| \le 1\end{aligned}$$

and

$$\begin{aligned}& \text{GIC}(V, S_X) > N_- \\ \iff & \Gamma_{V,S_X}(N_-) = \text{FALSE} \\ \iff & \exists T : \mathcal{X} \setminus S_X \to \{-1, +1, \perp\}, \forall \Sigma \subseteq \mathcal{X} \setminus S_X, \sum_{x \in \Sigma} c(x, T(x)) \le N_- \to |E(V[T(\Sigma)], S_X \cup \Sigma)| \ge 2\end{aligned}$$

2.9.1.1 Lemmas

We prove several lemmas on the properties of E-VS and Cost.

Lemma 5. *For any $V \subset \mathcal{H}$ and $S_X \subset \mathcal{X}$,*

$$E(V, S_X \cup \{x^*\}) \subseteq E(V, S_X)$$

Proof. It suffices to prove that $h \in E(V, S_X \cup \{x^*\}) \Rightarrow h \in E(V, S_X)$.

To see this, let $h \in E(V, S_X \cup \{x^*\})$. Then, $\forall h' \in V \setminus \{h\}, h((\mathcal{X} \setminus S_X) \setminus \{x^*\})) \neq h'((\mathcal{X} \setminus S_X) \setminus \{x^*\})) \Rightarrow \forall h' \in V \setminus \{h\}, h(\mathcal{X} \setminus S_X) \neq h'(\mathcal{X} \setminus S_X)$. This implies that $h \in E(V, S_X)$. □

Lemma 6. *We have the following:*

1. *For any $x \in \mathcal{X} \setminus S_X$ and $y \in \{-1, 1\}$,*

$$E(V[(x, y)], S_X \cup \{x\}) = E(V, S_X)[(x, y)].$$

2. *For any set of binary-labeled examples $W \subset (\mathcal{X} \times \{-1, 1\})$,*

$$E(V[W], S_X \cup W) = E(V, S_X)[W].$$

Proof. 1. We have the following equivalence:

$$\begin{aligned}
& h \in E(V[(x, y)], S_X \cup \{x\}) \\
\iff & h \in V[(x, y)] \wedge \forall h' \in V[(x, y)] \,.\, h' \neq h \rightarrow h'(\mathcal{X} \setminus (S_X \cup \{x\})) \neq h(\mathcal{X} \setminus (S_X \cup \{x\})) \\
\iff & h \in V \wedge h(x) = y \wedge \forall h' \in V[(x, y)] \,.\, h' \neq h \rightarrow h'(\mathcal{X} \setminus S_X) \neq h(\mathcal{X} \setminus S_X) \\
\iff & h \in V \wedge h(x) = y \wedge \forall h' \in V \,.\, h' \neq h \rightarrow h'(\mathcal{X} \setminus S_X) \neq h(\mathcal{X} \setminus S_X) \\
\iff & h(x) = y \wedge h \in E(V, S_X) \\
\iff & h \in E(V, S_X)[(x, y)]
\end{aligned}$$

where the first equality uses the definition of effective version space; the second equality uses the fact that for $h, h' \in V[(x, y)]$, $h'(\mathcal{X} \setminus (S_X \cup \{x\})) \neq h(\mathcal{X} \setminus (S_X \cup \{x\}))$ is equivalent to $h'(\mathcal{X} \setminus S_X) \neq h(\mathcal{X} \setminus S_X)$; the third equality follows from that for h such that $h(x) = y$, for all $h' \in V$ such that $h'(x) \neq y$, $h'(x) \neq h(x)$ and therefore $h'(\mathcal{X} \setminus S_X) \neq h(\mathcal{X} \setminus S_X)$ holds trivially; the fourth equality uses the definition of effective version space; the last equality uses the definition of version space with respect to labeled examples.

2. The claim follows by induction on $|W|$:

Base case. If $|W| = 1$, the claim follows from the previous item.

Inductive case. Assume that $E(V[W'], S_X \cup W') = E(V, S_X)[W']$ holds for any W' such that $|W'| < n$; Now consider any W of size n; W can be represented as $\big\{(x, y)\big\} \cup W'$ for some $(x, y) \in \mathcal{X} \times \{-1, 1\}$ and $|W'| = n - 1$. We have:

$$\begin{aligned}
E(V[W], S_X \cup W) =& E(V[W'][(x, y)], S_X \cup W' \cup \{x\}) && \text{(Definition of version space)} \\
=& E(V[W'], S_X \cup W')[(x, y)] && \text{(item 1)} \\
=& E(V, S_X)[W'][(x, y)] && \text{(Inductive hypothesis)} \\
=& E(V, S_X)[W] && \text{(Definition of version space)}
\end{aligned}$$

This completes the induction.

□

Lemma 7. $E(V, S_X) \neq \emptyset$ *iff* $\mathrm{Cost}(V, S_X) \geq 0$.

Proof. ($\Leftarrow$) From the first terminal condition in the definition of Cost, we know that $E(V, S_X) = \emptyset \implies \mathrm{Cost}(V, S_X) = -\infty < 0$. So $\mathrm{Cost}(V, S_X) \geq 0 \implies E(V, S_X) \neq \emptyset$.

($\Rightarrow$) By backward induction on $|S_X|$.

Base case. If $S_X = \mathcal{X}$, $|E(V, S_X)| = 0$ or 1. If $|E(V, S_X)| = 1$, we have by the base case of the definition of Cost, $\mathrm{Cost}(V, S_X) = 0$. Therefore, $E(V, S_X) \neq \emptyset \implies \mathrm{Cost}(V, S_X) \geq 0$.

Inductive case. Suppose $E(V, S_X) \neq \emptyset \implies \mathrm{Cost}(V, S_X) \geq 0$ holds for any dataset S_X of size $\geq j + 1$. Consider S_X of size j and V such that $E(V, S_X) \neq \emptyset$:

- If $|E(V, S_X)| = 1$, then $\mathrm{Cost}(V, S_X) = 0 \geq 0$.
- Otherwise, $|E(V, S_X)| \geq 2$; take $h_1 \in E(V, S_X)$; we have

  $$\mathrm{Cost}(V, S_X) \geq \min_x \big(\mathrm{Cost}(V[(x, h_1(x))], S_X \cup \{x\}) + 1)\big)$$

  By Lemma 6, $h_1 \in E(V[(x, h_1(x))], S_X \cup \{x\})$, by inductive hypothesis, $\mathrm{Cost}(V[(x, h_1(x))], S_X \cup \{x\}) \geq 0$, and therefore $\mathrm{Cost}(V, S_X) \geq 1 \geq 0$.

In summary, $\mathrm{Cost}(V, S_X) \geq 0$.
This completes the induction. □

Taking the contrapositive of the above lemma we obtain the following corollary.

Corollary 3. $\mathrm{Cost}(V, S_X) = -\infty$ *iff* $|E(V, S_X)| = 0$.

Lemma 8. $|E(V, S_X)| \geq 2$ *iff* $\mathrm{Cost}(V, S_X) \geq 1$.

Proof. ($\Leftarrow$) From the first two terminal conditions in the definition of Cost, we know that if $|E(V, S_X)| \leq 1 \Rightarrow \mathrm{Cost}(V, S_X) \leq 0$ and so, $\mathrm{Cost}(V, S_X) \geq 1 \Rightarrow |E(V, S_X)| \geq 2$.

($\Rightarrow$) Let $h_1 \in E(V, S_X)$, consider labeling strategy $T(x) = h_1(x)$ for all $x \in \mathcal{X} \setminus S$ (i.e. never abstains).

Following the definition of $\mathrm{Cost}(V, S_X)$, we have

$$\mathrm{Cost}(V, S_X) \geq \min_x \big(\mathrm{Cost}(V[(x, h_1(x))], S_X \cup \{x\}) + 1)\big)$$

Also, note that by Lemma 6,

$$E(V[(x, h_1(x))], S_X \cup \{x\}) = E(V, S_X)[(x, h_1(x))] \ni h_1$$

Therefore, by Lemma 7, for every x, $\mathrm{Cost}(V[(x, h_1(x))], S_X \cup \{x\}) \geq 0$, and thus $\mathrm{Cost}(V, S_X) \geq 1$. □

Because $\operatorname{Cost}(V, S_X)$ can have three possibilities: $\operatorname{Cost}(V, S_X) = \begin{cases} -\infty \\ = 0 \\ \geq 1 \end{cases}$, and $E(V, S_X)$ having three possibilities: $|E(V, S_X)| \begin{cases} = 0 \\ = 1 \\ \geq 2 \end{cases}$, the above two lemmas yield the following simple corollary.

Corollary 4. $\operatorname{Cost}(V, S_X) = 0 \Leftrightarrow |E(V, S_X)| = 1$.

Proposition 13. *For any* V*,* $|E(V, \mathcal{X})| \leq 1$.

Proof. We consider three cases:

1. If $V = \emptyset$, then $E(V, \mathcal{X}) = \emptyset$
2. If $|V| = 1$, then $E(V, \mathcal{X}) = V$
3. If $|V| \geq 2$, then $E(V, \mathcal{X}) = \emptyset$.
   This is because for any $h \in V$, consider some $h' \in V \setminus \{h\}$. h' trivially agrees with h on $\mathcal{X} \setminus \mathcal{X} = \emptyset$. And so, $h(\emptyset) = h'(\emptyset) \Rightarrow h \notin E(V, \mathcal{X})$.

In summary, in all three cases, $|E(V, \mathcal{X})| \leq 1$. □

Lemma 9. *Algorithm 2 maintains the invariant that* $\operatorname{GIC}(V, S_X) \leq \operatorname{GIC}(\mathcal{H}, \emptyset)$.

Proof. It suffices to show that $\operatorname{GIC}(V, S_X)$ is nonincreasing throughout. In other words, after obtaining queried sample $(x, T(x))$ during an iteration of the algorithm,

$$\operatorname{GIC}(V[T(x)], S_X \cup \{x\}) \leq \operatorname{GIC}(V, S_X) \tag{2.3}$$

Denote by $t = \operatorname{GIC}(V, S_X)$. It therefore suffices to show that, for any oracle $T' : \mathcal{X} \setminus (S_X \cup \{x\}) \to \{-1, +1, \perp\}$, there exists $\Sigma' \subset \mathcal{X} \setminus (S_X \cup \{x\})$ such that:

$$\sum_{x \in \Sigma'} c(x, T'(x)) \leq t \wedge \left| E(V[T(x)][T'(\Sigma')], S_X \cup \{x\} \cup \Sigma') \right| \leq 1. \tag{2.4}$$

Below we construct such a Σ' for each T'.

First, define oracle $\tilde{T} : \mathcal{X} \setminus S_X \to \{-1, +1, \perp\}$ as:

$$\tilde{T}(z) = \begin{cases} T(x) & z = x \\ T'(z) & z \neq x \end{cases}$$

By the definition of $\operatorname{GIC}(V, S_X)$, for this $\tilde{T}$, there exists $\tilde{\Sigma}$ such that:

$$\sum_{x \in \tilde{\Sigma}} c(x, \tilde{T}(x)) \leq t \wedge \left| E(V[\tilde{T}(\tilde{\Sigma})], S_X \cup \tilde{\Sigma}) \right| \leq 1.$$

We now construct Σ' by considering two cases of $\tilde{\Sigma}$ respectively:

1. If $x \in \tilde{\Sigma}$, we construct $\Sigma' := \tilde{\Sigma} \backslash \{x\}$. Note that $\sum_{x \in \Sigma'} c(x, T'(x)) \leq \sum_{x \in \tilde{\Sigma}} c(x, \tilde{T}(x)) \leq t$, and by the definition of $\tilde{T}$,

$$\begin{aligned}
&E(V[T(x)][T'(\Sigma')], S_X \cup \{x\} \cup \Sigma') \\
=&E(V[\tilde{T}(x)][\tilde{T}(\tilde{\Sigma} \setminus \{x\})], S_X \cup \{x\} \cup (\tilde{\Sigma} \setminus \{x\})) \\
=&E(V[\tilde{T}(\tilde{\Sigma})], S_X \cup \tilde{\Sigma})
\end{aligned}$$

and therefore has size ≤ 1.

2. If $x \notin \tilde{\Sigma}$, we construct $\Sigma' = \tilde{\Sigma}$. Note that $\sum_{x \in \Sigma'} c(x, T'(x)) = \sum_{x \in \tilde{\Sigma}} c(x, \tilde{T}(x)) \leq t$, and:

$$\begin{aligned}
&E(V[T(x)][T'(\Sigma')], S_X \cup \{x\} \cup \Sigma') \\
=&E(V[\tilde{T}(\tilde{\Sigma})][T(x)], S_X \cup \tilde{\Sigma} \cup \{x\}) && \text{(since } T'(\Sigma') = \tilde{T}(\tilde{\Sigma})\text{)} \\
\subseteq&E(V[\tilde{T}(\tilde{\Sigma})], S_X \cup \tilde{\Sigma}) && (\diamond)
\end{aligned}$$

and therefore has size ≤ 1. Here, for the last inequality $(\diamond)$, we use Lemma 6 (for when $T(x) \in \{+1, -1\}$) and Lemma 5 (for when $T(x) = \perp$) which implies that for any set $\mathcal{F} \subset \mathcal{H}$ and unlabeled examples U, $E(\mathcal{F}[T(x)], U \cup \{x\}) \subseteq E(\mathcal{F}, U)$.

In summary, there always exists Σ' that satisfies (2.4), and therefore (2.3) holds for every iteration of Algorithm 2. This concludes the proof of the lemma. □

2.9.2 Main Results

In this section, we prove the generalized version of results in Section 2.3, in which examples may incur differing costs.

Lemma 10. *For any V, S_X such that $\mathrm{GIC}(V, S_X)$ is finite, $\exists x \in \mathcal{X} \setminus S_X$ such that:*

$$\max_{y \in \{-1,+1\}} \big(|E(V[(x,y)], S_X \cup \{x\}))| - 1\big) \leq (|E(V, S_X)| - 1) \left(1 - \frac{c(x)}{\mathrm{GIC}(V, S_X)}\right).$$

Proof. Recall from Lemma 6 that we have: $E(V[(x,y)], S_X \cup \{x\})) = E(V, S_X)[(x,y)]$, it suffices to prove that there exists $x \in \mathcal{X} \setminus S_X$ such that

$$\max_{y \in \{-1,+1\}} \big(|E(V, S_X)[(x,y)])| - 1\big) \leq (|E(V, S_X)| - 1) \left(1 - \frac{c(x)}{\mathrm{GIC}(V, S_X)}\right).$$

Also, note that $|E(V, S_X)| = |E(V, S_X)[(x, -1)]| + |E(V, S_X)[(x, +1)]|$, as $E(V, S_X)[(x, -1)]$ and $E(V, S_X)[(x, +1)]$ form a disjoint partition of $E(V, S_X)$.

And so, equivalently, it suffices to show that there exists $x \in \mathcal{X} \setminus S_X$ such that:

$$\min\big(|E(V, S_X)[(x, -1)]|, |E(V, S_X)|[(x, +1)]\big) \geq c(x) \frac{|E(V, S_X)| - 1}{\mathrm{GIC}(V, S_X)}$$

So, assume towards contradiction that the statement above does not hold. Then, we have that $\forall x \in \mathcal{X} \setminus S_X$:

$$\min\big(|E(V, S_X)[(x, -1)]|, |E(V, S_X)|[(x, +1)]\big) < c(x) \frac{|E(V, S_X)| - 1}{\mathrm{GIC}(V, S_X)} \tag{2.5}$$

Define oracle $T_0 : \mathcal{X} \setminus S_X \to \{-1, +1, \perp\}$ such that,

$$T_0(x) = \operatorname*{argmax}_{y \in \{-1,1\}} |E(V, S_X)[(x, y)]|$$

With this, for every subset $\Sigma \subseteq \mathcal{X} \setminus S_X$ such that $\sum_{x \in \Sigma} c(x, T_0(x)) \leq \mathrm{GIC}(V, S_X)$, we have:

$$\begin{aligned}
|E(V[T_0(\Sigma)], S_X \cup \Sigma)| &= |E(V, S_X)[T_0(\Sigma)]| && \text{(Lemma 6, item 2)} \\
&= |E(V, S_X)| - |\{h \in E(V, S_X) : \exists x \in \Sigma, h(x) \neq T_0(x)\}| \\
& && \text{(Set algebra)} \\
&\geq |E(V, S_X)| - \sum_{x \in \Sigma} |E(V, S_X)[(x, \neg T_0(x))]| && \text{(Union bound)} \\
&= |E(V, S_X)| - \sum_{x \in \Sigma} \min_{y \in \{+1,-1\}} |E(V, S_X)[(x, y)]| \\
& && \text{(by definition of } T_0(x)\text{)} \\
&> |E(V, S_X)| - \sum_{x \in \Sigma} c(x, T_0(x)) \frac{|E(V, S_X)| - 1}{\mathrm{GIC}(V, S_X)} \\
& \qquad \text{(by (2.5) and } c(x) = c(x, T_0(x)) \text{ since } T_0(x) \in \{-1, +1\}\text{)} \\
&\geq |E(V, S_X)| - (|E(V, S_X)| - 1) = 1,
\end{aligned}$$

In summary, the constructed oracle T_0 is such that for any $\Sigma \subseteq \mathcal{X} \setminus S_X$ such that $\sum_{x \in \Sigma} c(x, T_0(x)) \leq \mathrm{GIC}(V, S_X)$, $|E(V[T_0(\Sigma)], S_X \cup \Sigma)| > 1$. Therefore, $\Gamma_{V, S_X}(GIC(V, S_X)) =$ FALSE, which contradicts the definition of $\mathrm{GIC}(V, S_X)$. □

For the lemma below, we will consider the following Algorithm that simulates the interaction between a query strategy and oracle.

Using this, we will aim to show that Cost upper bounds GIC.

In the following Algorithm, let us define T to be a labeling oracle that satisfies the properties in (2.6).

And we will define U to be the output of executing the following algorithm, Algorithm 8, which simulates the interaction between a specific label query strategy and the oracle T before a stopping criterion is reached.

Algorithm 8 Simulation process on letting T interacting with a targeted label query strategy

```
$U \leftarrow \emptyset$
**while** $U \neq \mathcal{X} \setminus S_X$ and $\sum_{x \in U} c(x, T(x)) \leq k$ **do**
    Choose example:
        $x = \mathrm{argmin}_{x \in \mathcal{X} \setminus (S_X \cup U)} \max_{y \in \{-1,+1,\perp\}} c(x, y) + \mathrm{Cost}\Big(V[T(U) \cup \big\{(x, y)\big\}], S_X \cup U \cup \{x\}\Big)$
    $U \leftarrow U \cup \{x\}$
**return** $U$
```

Lemma 11. *For any $V \subset \mathcal{H}$ and $S_X \subset \mathcal{X}$,*

$$\mathrm{GIC}(V, S_X) \leq \mathrm{Cost}(V, S_X)$$

Proof. Let $\epsilon > 0$ and $k = \mathrm{GIC}(V, S_X) - \epsilon$.

By the definition of GIC, $\Gamma_{V,S_X}(k) = \mathrm{FALSE}$. That is:

$$\exists T : \mathcal{X} \setminus S_X \to \{-1, +1, \perp\}, \forall \Sigma \subseteq \mathcal{X} \setminus S_X, \sum_{x \in \Sigma} c(x, T(x)) \leq k \Rightarrow \big|E(V[T(\Sigma)], S_X \cup \Sigma)\big| \geq 2 \tag{2.6}$$

With Algorithm 8, we first claim that $\sum_{x \in U} c(x, T(x)) > k$. Suppose not, we have $\sum_{x \in U} c(x, T(x)) \leq k$.

By the stopping criterion of Algorithm 8, we must have that $U = \mathcal{X} \setminus S_X$.

In this case, by (2.6), $|E(V[T(U)], S_X \cup U)| = |E(V[T(U)], \mathcal{X})| \geq 2$.

However, this contradicts Proposition 13 that for any V, $|E(V[T(U)], \mathcal{X})| \leq 1$. Therefore, $\sum_{x \in U} c(x, T(x)) > k$.

Denote by $x_1, \ldots, x_m$ the sequence of m examples queried by Algorithm 8. Under this notation, we have that $U = \{x_1, \ldots, x_m\}$.

Also, for $i \in \{0, 1, \ldots, m\}$, denote by $U_i := \{x_1, \ldots, x_i\}$ the set of first i examples queried, with the convention that $U_0 := \emptyset$.

We make two observations:

- For any $i \in \{0, 1, \ldots, m-1\}$, by the loop condition, $\sum_{x \in U_i} c(x, T(x)) \leq k$, therefore by (2.6), $\big|E(V[T(U_i)], S_X \cup U_i)\big| \geq 2$, and therefore, by the definition of Cost,

$$\mathrm{Cost}(V[T(U_i)], S_X \cup U_i) = \min_{x \in \mathcal{X} \setminus (S_X \cup U_i)} \max_{y \in \{-1,+1,\perp\}} \big(c(x, y) + \mathrm{Cost}(V[T(U_i)][(x, y)], S_X \cup U_i \cup \{x\})\big) \tag{2.7}$$

- $T(x_m) \neq \perp$. This is because $\sum_{i=1}^{m-1} c(x_i, T(x_i)) \leq k < \sum_{i=1}^{m} c(x_i, T(x_i))$, implying that $c(x_m, T(x_m)) > 0$. Furthermore, by our notation that $c(x, y) = c(x)\mathbb{1}(y \neq \perp)$,

$$\sum_{i=1}^{m-1} c(x_i, T(x_i)) + c(x_m, -1) = \sum_{i=1}^{m-1} c(x_i, T(x_i)) + c(x_m, +1) > k.$$

by (2.6), we also have $\big|E(V[T(U)], S_X \cup U)\big| \geq 2$ and by Lemma 8, $\mathrm{Cost}(V[T(U)], S_X \cup U) \geq 1$.

Based on these observations, we have:

$$
\begin{aligned}
\mathrm{Cost}(V, S_X) &= \min_{x \in \mathcal{X} \setminus S_X} \max_{y \in \{-1,+1,\perp\}} \big(c(x,y) + \mathrm{Cost}(V[(x,y)], S_X \cup \{x\})\big) \quad \text{(Eq. 2.7 with } i = 0\text{)} \\
&= \max_{y \in \{-1,+1,\perp\}} \big(c(x_1,y) + \mathrm{Cost}(V[(x_1,y)], S_X \cup \{x_1\})\big) \quad \text{(Eq. 3)} \\
&\geq c(x_1, T(x_1)) + \mathrm{Cost}(V[T(U_1)]), S_X \cup U_1) \\
&= c(x_1, T(x_1)) + \min_{x \in \mathcal{X} \setminus (S_X \cup U_1)} \max_{y \in \{-1,+1,\perp\}} \big(c(x,y) + \mathrm{Cost}(V[T(U_1)][(x,y)], S_X \cup U_1 \cup \{x\})\big) \\
&\qquad \text{(Eq. 2.7 with } i = 1\text{)} \\
&\geq \dots \\
&\geq \sum_{i=1}^{m-1} c(x_i, T(x_i)) + \mathrm{Cost}(V[T(U_{m-1})], S_X \cup U_{m-1}) \\
&\qquad \text{(Repeated application of Eqs. 2.7 and 3)} \\
&= \sum_{i=1}^{m-1} c(x_i, T(x_i)) + \\
&\quad \min_{x \in \mathcal{X} \setminus (S_X \cup U_1)} \max_{y \in \{-1,+1,\perp\}} \big(c(x,y) + \mathrm{Cost}(V[T(U_{m-1})][(x,y)], S_X \cup U_{m-1} \cup \{x\})\big) \\
&\geq \sum_{i=1}^{m-1} c(x_i, T(x_i)) + \max_{y \in \{-1,+1\}} \big(c(x_m,y) + \mathrm{Cost}(V[T(U_{m-1})][(x_m,y)], S_X \cup U_{m-1} \cup \{x_m\})\big) \\
&\qquad \text{(Eq. 3 and restricting the choice of } y\text{)} \\
&\geq \sum_{i=1}^{m} c(x_i, T(x_i)) > k
\end{aligned}
$$

Here, in the second to last inequality, we use the following observations: first, for any $c(x_m, -1) = c(x_m, +1) = c(x_m, T(x_m))$; second, $|E(V[T(U_{m-1}), S_X \cup U_{m-1}])| \geq 2$, which implies that there is at least one $y \in \{-1, +1\}$ such that $|E(V[T(U_{m-1})[(x,y)], S_X \cup U_{m-1} \cup \{x\}])| \geq 1$ (recall Lemma 6), and therefore

$$\mathrm{Cost}(V[T(U_{m-1})[(x,y)], S_X \cup U_{m-1} \cup \{x\}]) \geq 0.$$

In summary, for any $\epsilon > 0$, we have shown that $\mathrm{Cost}(V, S_X) \geq \mathrm{GIC}(V, S_X) - \epsilon$. The lemma statement follows by letting $\epsilon \downarrow 0$. □

Theorem 5. *If Algorithm 2 interacts with a labeling oracle T, then it incurs total query cost at most $\mathrm{GIC}(\mathcal{H}, \emptyset) \ln |\mathcal{H}| + 1$. Furthermore, if Algorithm 2 interacts with an identifiable oracle T consistent with some $h^* \in \mathcal{H}$, then it identifies h^*.*

Proof. First, we show that Algorithm 2 terminates and correctly identifies h^* when interacting with an identifiable oracle of h^*. Its termination can be seen by the fact that the size of S_X is increasing by 1 for each iteration and $S_X \neq \mathcal{X}$ is part of the stopping criterion.

We now show that when it returns, $E(V, S_X) = \{h^*\}$. This can be seen by:

- As T is an identifiable oracle that is consistent with h^*, the algorithm maintains the invariant that $h^* \in E(V, S_X)$.

  This is because if at some point $h^* \notin E(V, S_X)$, then exists some $h' \neq h^*$ in $V = \mathcal{H}[T(S_X)]$ such that $h'(\mathcal{X} \setminus S_X) = h^*(\mathcal{X} \setminus S_X)$. Then, we combine with that $h' \in \mathcal{H}[T(S_X)]$ to get that $h' \in \mathcal{H}[T(S_X) \cup h^*(\mathcal{X} \setminus S_X)] \subseteq \mathcal{H}[T(S_X) \cup T(\mathcal{X} \setminus S_X)] = \mathcal{H}[T(\mathcal{X})]$, which is in contradiction with that T is an identifiable oracle.
- We claim that when it returns, $|E(V, S_X)| = 1$. Since the E-VS always contains h^*, we must have $|E(V, S_X)| \geq 1$.

  And so, if it returns, we have the condition of the while loop being false, i.e., we either have $|E(V, S_X)| < 2 \implies |E(V, S_X)| = 1$, or $S_X = \mathcal{X} \implies |E(V, S_X)| = 1$ thanks to Proposition 13.

Next we bound the query cost complexity of Algorithm 2, when interacting with any labeling oracle.

Denote V_i and S_i as the value of V and S at the i-th iteration, and denote (x_i, y_i) by the example (x, y) obtained at the i-th iteration. We denote $(S_i)_X$ as the unlabeled part of S_i.

Therefore, $V_{i+1} = V[(x_i, y_i)]$ and $S_{i+1} = S_i \cup \{(x_i, y_i)\}$.

We claim that

$$(\left|E(V_{i+1}, (S_{i+1})_X)\right| - 1) \leq (\left|E(V_i, (S_i)_X)\right| - 1) \cdot \exp\left(-\frac{c(x_i, y_i)}{\mathrm{GIC}(\mathcal{H}, \emptyset)}\right). \tag{2.8}$$

To see this, we consider two cases:

1. If $y_i \in \{-1, +1\}$, then applying Lemma 10 with $V = V_i$, $S_X = (S_i)_X$, $x = x_i$, we have

$$\begin{aligned}
(|E(V_{i+1}, (S_{i+1})_X)| - 1) \leq & \max_{y \in \{-1,+1\}} \left(\left|E(V_i[(x_i, y)], (S_{i+1})_X)\right| - 1\right) \\
\leq & (|E(V_i, (S_i)_X)| - 1)\left(1 - \frac{c(x_i)}{\mathrm{GIC}(V_i, (S_i)_X)}\right) \\
& \text{(Lemma 10 since } y_i \in \{-1, +1\}\text{)} \\
\leq & (|E(V_i, (S_i)_X)| - 1)\left(1 - \frac{c(x_i)}{\mathrm{GIC}(\mathcal{H}, \emptyset)}\right) \\
& \text{(by Lemma 9, } \mathrm{GIC}(V_i, (S_i)_X) \leq \mathrm{GIC}(\mathcal{H}, \emptyset)\text{)} \\
\leq & (\left|E(V_i, (S_i)_X)\right| - 1) \cdot \exp\left(-\frac{c(x_i)}{\mathrm{GIC}(\mathcal{H}, \emptyset)}\right). \\
& \text{(since } 1 - x \leq e^{-x}\text{)}
\end{aligned}$$

2. If $y_i = \perp$, $c(x_i, y_i) = 0$. Therefore, to show (2.8), it suffices to show that $E(V_{i+1}, (S_{i+1})_X) \subseteq E(V_i, (S_i)_X)$. This follows from Lemma 5.

To summarize, (2.8) holds for each iteration i.

Consider the last iteration i_0 before the termination condition is reached; note that by the termination criterion, the penultimate E-VS is such that $|E(V_{i_0}, (S_{i_0})_X)| \geq 2$. We now upper bound the total cost up to iteration $i_0 - 1$. By repeatedly using (2.8) for $i = 1, \dots, i_0 - 1$, we have:

$$1 \leq \left|E(V_{i_0}, (S_{i_0})_X)\right| - 1 \leq \left|E(\mathcal{H}, \emptyset)\right| \cdot \exp\left(-\frac{\sum_{i=1}^{i_0-1} c(x_i, y_i)}{\mathrm{GIC}(\mathcal{H}, \emptyset)}\right)$$

Therefore, $\sum_{i=1}^{i_0-1} c(x_i, y_i) \leq \mathrm{GIC}(\mathcal{H}, \emptyset) \ln |\mathcal{H}|$ (since $E(\mathcal{H}, \emptyset) = \mathcal{H}$) and:

$$\sum_{i=1}^{i_0} c(x_i, y_i) = c(x_{i_0}, y_{i_0}) + \sum_{i=1}^{i_0-1} c(x_i, y_i) \leq \mathrm{GIC}(\mathcal{H}, \emptyset) \ln |\mathcal{H}| + 1.$$

□

2.10 Proofs for Subsections 2.3.1 and 2.3.2

2.10.1 Comparing VS versus E-VS

Consider the case when $\mathcal{H}$ is linear: $\mathcal{H} = \left\{h_w(x) = \mathrm{sign}(w^T x) | w = [w', 1], w' \in [0, 1]^d\right\}$. We observe that, for any set of points $\mathcal{X}$, $\mathcal{X}$ divide polytope $W = \left\{w = [w', 1] : w' \in [0, 1]^d\right\}$ into clusters, where every point in the cluster has the same labeling of $\mathcal{X}$. Thus, without loss of generality, we can treat each cluster formed by $\mathcal{X}$ as an element of $\mathcal{H}$, and $\mathcal{H}$ comprises of all the clusters that lie in polytope W. In this setting, the (conventional) version space is a single convex polytope, which we may access by sampling using any polytope sampler. The structural lemma below illustrates that, by contrast, the E-VS can be a more complicated object to access.

Proposition 14. *There exists an instance space $\mathcal{X} \subset \mathbb{R}^d$ and query responses S such that the resultant E-VS is a union of $e^{\Omega(d)}$ disjoint polytopes.*

Proof. **Defining the Instance Space:** We construct a $\mathcal{X}$ that allows us to easily reason about the E-VS. Consider any $3n$ positive reals a^j_k for $j \in [n], k \in [3]$ such that $0 < a^1_1 < a^1_2 < a^1_3 < ... < a^n_3 < 1$. Define $x^i_{jk} = [-e_i, a^j_k]$ for $i \in [d]$. As a concrete example, $x^1_{23} = [-1, 0, ..., a^3_2]$.

Define the instance space to be $\mathcal{X} = \left\{x^i_{jk} | i \in [d], j \in [n], k \in [3]\right\}$. With $\mathcal{X}$ defined, we see the clusters of W formed by $\mathcal{X}$ (referred to as *cells* subsequently) consists of: $\times_{i=1}^d I$, where $I = \left\{[0, a^1_1], [a^1_1, a^1_2], [a^1_2, a^1_3], ..., [a^n_3, 1]\right\}$.

Now, define the interaction history $S = \left\{(x^i_{jk}, \perp) | i \in [d], j \in [n], k = 2\right\}$. Note that then $S_X = \left\{x^i_{jk} | i \in [d], j \in [n], k = 2\right\}$.

Characterizing the E-VS: We first claim that for any cell with one of its faces a subset of a hyperplane in S_X cannot be in the E-VS. Specifically, if there $\exists i \in [d], j \in [n]$ such that $w_i \in [a^j_1, a^j_3]$, then the cell w belongs to is not in the E-VS.

To see this, WLOG $w_i \in [a^j_1, a^j_2]$; the case of $w_i \in [a^j_2, a^j_2]$ can be analyzed analogously.

Now, construct $\tilde{w} = [w_1, ..., w_{i-1}, \tilde{w}_i, w_{i+1}, ...1]$, for some $\tilde{w}_i \in [a^j_2, a^j_3]$. Note that by construction, w' does not lie in the same cell as w. Then, we see that $\mathrm{sign}(w'^T x) = \mathrm{sign}(w^T x)$, $\forall x \in \mathcal{X} \setminus \left\{x^i_{j2}\right\}$. And so, since $\mathcal{X} \setminus S_X \subseteq \mathcal{X} \setminus \left\{x^i_{j2}\right\}$, we have that $w(\mathcal{X} \setminus S_X) = w'(\mathcal{X} \setminus S_X) \Rightarrow w \notin E(V, S_X)$.

This means that only the set of disjoint cells $\times_{i=1}^{d} I'$, where $I' = \left\{[0, a_1^1], [a_3^1, a_1^2], \ldots, [a_3^n, 1]\right\}$, can be in the E-VS. Next, we will argue that the E-VS is all of $\times_{i=1}^{d} I'$.

Consider a classifier corresponding to some cell $c \in \times_{i=1}^{d} I'$. Consider any other cell classifier corresponding to cell $c' \in \times_{i=1}^{d} I$. Since $c \neq c'$, there must be at least one dimension, WLOG i, such that c and c' belong to different sub-intervals, when projected onto coordinate i.

We know that along dimension i, c's sub-interval is either of the form $[0, a_1^1]$, $[a_3^j, a_1^{j+1}]$ for some j, or $[a_3^n, 1]$.

We see that in the first case, $x_{11}^i \in \mathcal{X} \setminus S_X$ must separate c and c', since $c(x) = +1 \neq -1 = c'(x)$. Analogously, in the second case, either x_{j3}^i or $x_{(j+1)1}^i$ must separate c and c' (with both such points are in $\mathcal{X} \setminus S_X$). Finally, in the last case, $x_{n3}^i \in \mathcal{X} \setminus S_X$ must separate c and c'.

This shows that all of $\times_{i=1}^{d} I'$ is in the E-VS. And so, since I' comprises of $n + 1$ disjoint intervals, there are in total $(n + 1)^d$ number of disjoint cells, corresponding to distinct classifiers.

□

2.10.2 E-VS Membership Check

The key idea behind the membership check $h \in E(V, S_X)$ (lines 2 to 4 in Algorithm 3) is that we want to find a hypothesis $\hat{h}$ in V, different from h, that agrees on the rest of the unqueried samples. If we succeed in finding this $\hat{h}$, then this means that even if all of the remaining unqueried samples $\mathcal{X} \setminus S_X$ is labeled, h and $\hat{h}$ cannot be distinguished from each other. This implies that h is non-identifiable and does not belong to the E-VS.

Proposition 15. *Given some $h \in V$ and access to a* C-ERM *oracle, lines 2 to 4 in Algorithm 3 verifies whether $h \in E(V, S_X)$, with one call to the oracle.*

Proof. Firstly, note that by definition, $\forall h, h' \in \mathcal{H}$, $h \neq h' \Rightarrow h(\mathcal{X}) \neq h'(\mathcal{X})$.

Recall that in Algorithm 3, $S^\perp$ denotes the set of examples in S_X on which the labeler abstains. Now, we rewrite the definition of $h \in V$ not being in the E-VS:

$$\begin{aligned}
& h \notin E(V, S_X) \\
\Leftrightarrow & \exists h' \in V \setminus \{h\}, h'(\mathcal{X} \setminus S_X) = h(\mathcal{X} \setminus S_X) \\
\Leftrightarrow & \exists h', h'(S_X \setminus S^\perp) = h(S_X \setminus S^\perp) \wedge h'(\mathcal{X}) \neq h(\mathcal{X}) \wedge h'(\mathcal{X} \setminus S_X) = h(\mathcal{X} \setminus S_X) \\
\Leftrightarrow & \exists h', h'(S_X \setminus S^\perp) = h(S_X \setminus S^\perp) \wedge h'(S^\perp) \neq h(S^\perp) \wedge h'(\mathcal{X} \setminus S_X) = h(\mathcal{X} \setminus S_X) \\
\Leftrightarrow & \exists h', \exists x^\perp \in S^\perp, h'(S_X \setminus S^\perp) = h(S_X \setminus S^\perp) \wedge h'(x^\perp) \neq h(x^\perp) \wedge h'(\mathcal{X} \setminus S_X) = h(\mathcal{X} \setminus S_X)
\end{aligned}$$

And so, we may check for the existence of such a h' with one C-ERM call on $\mathcal{H}$, given some $h \in V$, using the following program:

$$\begin{aligned}
\min_{h' \in \mathcal{H}} & \sum_{x' \in S^\perp} \mathbb{1}\left\{h'(x') = h(x')\right\} \\
\text{s.t } & h'(x) = h(x), \forall x \in \mathcal{X} \setminus S^\perp
\end{aligned}$$

This may be emulated by defining data $Z_1 = \left\{(x, -h(x))\right\}_{x \in S^\perp}$, $Z_2 = \left\{(x, h(x))\right\}_{x \in \mathcal{X} \setminus S^\perp}$, and calling C-ERM on Z_1, Z_2 to compute $\hat{h} \in \operatorname{argmin}\left\{\operatorname{err}(h', Z_1) : h' \in \mathcal{H}, \operatorname{err}(h', Z_2) = 0\right\}$.

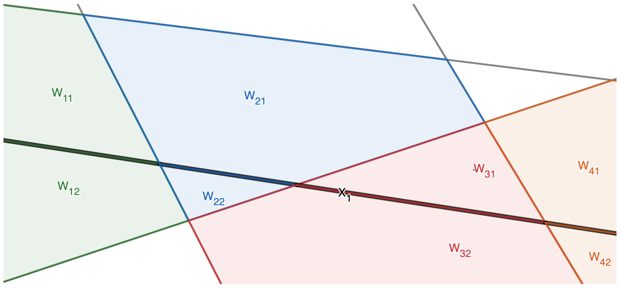


Figure 2.2: Geometric view of the linear hypothesis class in dual space (as in Tong and Koller [278]), with examples as hyperplanes and hypotheses as cells, illustrates: (i) Abstention on example x_1 (hyperplane in black) renders hypotheses w_{i1} and w_{i2} (cells of the same color) indistinguishable from each other. In this way, abstentions can carve up the VS (single polytope) into multiple polytopes, as in Proposition 3. (ii) In the approximate identifiability game (Subsection 2.4.1), if x_1 is not in pool X^m, then it induces clusters of merged $\{w_{i1}, w_{i2}\}$ for $i \in [4]$. The goal then is to only identify up to clusters (e.g. the blue cluster of $\{w_{21}, w_{22}\}$), instead of the exact hypothesis (e.g. cell w_{21}).

It can be now seen that: if C-ERM outputs $\hat{h} \neq h$, then $h \notin E(V, S_X)$; otherwise, $\hat{h} = h$ and therefore $h \in E(V, S_X)$. □

2.10.3 Contrasting E-VS bisection Algorithm with VS bisection

2.10.3.1 Proof of Theorem 2

In this section we prove Theorem 2, showing an exponential gap between our new E-VS bisection algorithm and the conventional VS bisection algorithm.

Setup: Our example will revolve around a hybrid hypothesis class of thresholds and intervals. Let $n \geq 8$. Our instance space $\mathcal{X} = \mathcal{X} \cup \mathcal{X}_2$, where $\mathcal{X}_1 = \left\{\frac{1}{2n}, \ldots, \frac{2n-3}{2n}\right\}$ and $\mathcal{X}_2 = \left\{1 + \frac{3}{2n}, \ldots, 1 + \frac{2n-1}{2n}\right\}$. Note that $|\mathcal{X}| = 2(n-1)$.

Let $f_i : (-\infty, 1] \to \{+1, -1\}$ denote intervals of length $1/n$, $f_i(x) = \mathbb{1}(x \in [(i-1)/n, i/n])$ for $i \in [n-1]$.

Let $f'_i : (1, +\infty) \to \{+1, -1\}$ denote thresholds, $f'_i(x) = \mathbb{1}(x \geq 1 + i/n)$ for $i \in [n]$.

Define $\mathcal{H} = \bigcup_{i=1}^{n-1} \left\{h_{f_i, f'_i}, h_{f_i, f'_{i+1}}\right\}$, where $h_{f,f'}(x) = \begin{cases} f(x), & x \leq 1 \\ f'(x), & x > 1 \end{cases}$.

2.10.3.2 Algorithm Analysis

Under the paired interval-threshold setup, we compare the algorithms based on the number of samples queried before termination.

In the case of the VS-bisection algorithm, it queries the point that maximally bisects the VS each time. Accordingly, the algorithm terminates when there is no point that can split the VS. This arises either because the set of unqueried points is non-empty but the VS agrees on all their labels, or the set of unqueried points is empty.

While for the E-VS bisection algorithm, it terminates either when the E-VS is of cardinality zero or of one.

Lemma 12 (E-VS bisection algorithm query complexity)**.** *In the paired interval-threshold hypothesis learning setting, the E-VS algorithm incurs $O(\log n)$ sample complexity against any labeling oracle.*

Proof. Define $\rho(E(V, S_X), x) = \min_{y\in\{+1,-1\}} |E(V, S_X)[x, y]|$.

1. Let $U_2 \subseteq \mathcal{X}_2$ denote the unlabeled part of $\mathcal{X}_2$ such that $U_2 = \left\{x : \rho(E(V, S_X), x) > 0, x \in \mathcal{X}_2\right\}$ (i.e. $x \in \mathcal{X}_2$ is in the disagreement region formed by the current E-VS).

   Definition 12. *A point $x \in U_2$ is balanced if there exists a three-point segments with $x^2_i + 2/n = x^2_{i+1} + 1/n = x^2_{i+2}$, $x^2_j + 2/n = x^2_{j+1} + 1/n = x^2_{j+2}$ such that $x^2_{i+2} < x < x^2_j$, where points $x^2_i, x^2_{i+1}, x^2_{i+2} \in U_2$, and $x^2_j, x^2_{j+1}, x^2_{j+2} \in U_2$.*

   We have that, if:
   a) x is a balanced point
   b) all queried points thus far have been in $\mathcal{X}_2$, then:

   $$\rho(E(V, S_X), x) \geq 2 = \max_{x'\in\mathcal{X}_1} \rho(E(V, S_X), x')$$

   This follows because if no points have been queried in $\mathcal{X}_1$, $x^2_i, x^2_{i+1}, x^2_{i+2} \in U_2$ implies that $h_{f_{i+1},f'_{i+1}}$ and $h_{f_{i+1},f'_{i+2}} \in E(V, S_X)$. Similarly, $x^2_j, x^2_{j+1}, x^2_{j+2} \in U_2$ implies that $h_{f_{j+1},f'_{j+1}}$ and $h_{f_{j+1},f'_{j+2}} \in E(V, S_X)$.
   Since $x^2_{i+2} < x < x^2_j$, the two pairs of models disagree on x (in the second coordinate).
   And so, if there is some point $x \in U_2$ that is balanced, and all points queried thus far have been in $\mathcal{X}_2$, then the E-VS algorithm will query a point in U_2 (we assume that in a tie-breaker, the E-VS algorithm will select the point in $\mathcal{X}_2$).
2. From Lemma 13, we have that the E-VS algorithm will query some point in $U_2 \subseteq \mathcal{X}_2$ so long as $|U_2| \geq 7$.
   The number of binary labeled samples needed to reach $|U_2| < 7$ is at most $\log n$. This because abstention decreases $|U_2|$ by 1, while a binary label removes $\lfloor |U_2|/2 \rfloor$ points from U_2.
   And so, since $|U_2| = n$, there can be at most $\log n$ binary labeled examples before $|U_2| < 7$.
3. It remains to count the number of binary label samples needed when $|U_2| < 7$ before the interaction finishes.
   We note that if $|U_2| < 7$, then the size of the $|E(V, S_X)| \leq 2 \cdot 6 + 2$ (since it always holds that $|E(V, S_X)| \leq 2|U_2| + 2$).
   As each binary label point removes at least one hypothesis from the E-VS, at most 11 more binary label points are needed.
   In summary, we have that the E-VS algorithm incurs $O(\log n)$ samples.

Below are the deferred lemmas:

Lemma 13. *If $|U_2| \geq 7$, then the E-VS algorithm will query some point $x \in U_2 \subseteq \mathcal{X}_2$.*

Proof. We will show the following properties about U_2^t, which is U_2 at the tth step.

If $|U_2^t| \geq 7$, then:

i) U_2^t is of the form $\{a_1 : b_1\} \cup \{b_2 : a_2\}$, where $b_1 \leq b_2$ ($\{a_1 : b_1\}$ is used to abbreviate $\big\{a_1, a_1 + 1/n, ..., b_1 - 1/n, b_1\big\}$).

ii) Some $x \in \{b_1, b_2\}$ satisfies the following: $||\big\{x' \in U_2^t : x' < x\big\}| - |\big\{x' \in U_2^t : x' > x\big\}|| \leq 1$.

iii) No points $x_1, ..., x_{t-1}$ will have been queried from $\mathcal{X}_1$.

iv) E-VS will query some point $x \in U_2^t$ at step t.

We will see that, at step t, proving property i), ii), iii) proves iv), which is the desired result.

We prove by induction on j, the number of queries, that i), ii), iii) and thus iv) holds.

Base Case: When $j = 0$, no points have been queried from $\mathcal{X}_1$. And so, properties i)-iii) are true with $U_2 = \big\{1 + 3/2n : 1 + (2n-1)/2n\big\}$. Since $n \geq 8$, $|U_2| = |\mathcal{X}_2| = 7$, and so Lemma 14 applies, meaning iv) is satisfied.

Induction Step: Suppose that if $|U_2^j| \geq 7$, properties i)-iv) holds for time step $j = 0, ..., k-1$.

Now consider time step $j = k$. Suppose $|U_2^k| \geq 7$.

This means that, at time step $k-1$, $|U_2^{k-1}| \geq |U_2^k| \geq 7$ (since the disagreement region only decreases in size).

From induction hypothesis, we know U_2^{k-1} satisfies i)-iv). Let $U_2^{k-1} = \{a_1' : b_1'\} \cup \{b_2' : a_2'\}$. Since iv) holds at time $j = k-1$ ($x_{k-1} \in \mathcal{X}_2$), combined with that iii) applies at time $k-1$ ($x_1, ..., x_{k-2} \in \mathcal{X}_2$) implies property iii) holds at time $j = k$ ($x_1, ..., x_{k-1} \in \mathcal{X}_2$)).

Since iv) is satisfied at time step $k-1$, we may WLOG $x_{k-1} = b_1'$. There are two cases to consider:

- If a label is given for x_{k-1}, then we know that U_2^k is either $\big\{a_1' : b_1' - 1/n\big\}$ or $\{b_2 : a_2\}$, in either case, both i) and ii) are satisfied at step $j = k$.
- If an abstention is given for x_{k-1}, then we know that $U_2^k = \big\{a_1' : b_1' - 1/n\big\} \cup \{b_2' : a_2'\}$, which proves i).
  Since $x_{k-1} = b_1'$, we have that $||\{a_1' : b_1'\}| - |\{b_2' : a_2'\}|| \leq 1$.
  If $|\{b_2' : a_2'\}| \geq |\{a_1' : b_1'\}|$, picking b_2' satisfies the property, else picking $b_1' - 1/n$ satisfies the property. And so, property ii) for U_2^k holds.

Finally, since iii), i) and ii) holds for U_2^k, using Lemma 14, we have that $x_k \in \mathcal{X}_2$, which means that iv) holds at $j = k$.

□

Lemma 14. *If $|U_2^t| \geq 7$, and i)-iii) holds at step t: the E-VS algorithm will query one of $b_1, b_2 \in U_2^t$.*

Proof. Due to ii), we know at least one of b_1, b_2 satisfies $||\big\{x' \in U_2^t : x' < x\big\}| - |\big\{x' \in U_2^t : x' > x\big\}|| \leq 1$.

WLOG let this be b_1 (assume that b_1 wins the E-VS algorithm tie-breaker if both b_1, b_2 satisfy this condition). We claim the E-VS algorithm will query b_1.

- For points in $\mathcal{X}_2 \setminus U_2^t$, they are not in the disagreement region and $\rho(E(V, S_X), x) = 0$, which means they will not be queried.

- For points in U_2^t, we have the following observation.
  Due to i) and iii):

$$\begin{aligned}\rho(E(V, S_X), x) &= \min(2 \cdot |\left\{x' \in U_2^t : x' < x\right\}| + 1, 2 \cdot |\left\{x' \in U_2^t : x' > x\right\}| + 1) \\ &= 2 \cdot \min(|\left\{x' \in U_2^t : x' < x\right\}|, |\left\{x' \in U_2^t : x' > x\right\}|) + 1\end{aligned}$$

  From this, we can see that from ii),

$$\begin{aligned}b_1 &= \operatorname*{argmax}_{x \in U_2^t} \min(|\left\{x' \in U_2^t : x' < x\right\}|, |\left\{x' \in U_2^t : x' > x\right\}|) \\ &= \operatorname*{argmax}_{x \in U_2^t} \rho(E(V, S_X), x)\end{aligned}$$

- For points $x \in \mathcal{X}_1$.
  We know that $|U_2^t| \geq 7 \Rightarrow \min(|\left\{x' \in U_2^t : x' < b_1\right\}|, |\left\{x' \in U_2^t : x' > b_1\right\}|) \geq 3$.
  Due to i), we know that $\left\{x' \in U_2^t : x' < b_1\right\}$ and $\left\{x' \in U_2^t : x' > b_1\right\}$ are contiguous. And so, one can find three-point segments to the left and right of b_1, which means that b_1 is balanced.
  And so, $\rho(E(V, S_X), b_1) \geq 2 = \max_{x \in \mathcal{X}_1} \rho(E(V, S_X), x)$.

In conclusion, b_1 is the point that maximally bisects the E-VS out of all unqueried points, and will thus be queried by the E-VS bisection algorithm. □

□

Remark 5. *In closing, we note that the construction is nontrivial in that the same result does not hold if the hypothesis class is simply* $\mathcal{H} = \left\{h_{f_1, f_1'}, \dots, h_{f_{n-1}, f_{n-1}'}\right\}$.

In this case, the E-VS-bisection algorithm will also have a linear label complexity, as abstention from U_2 *does not result in a reduction in the size of E-VS. For a formal justification of this, please refer to the proof of Proposition 1*

Theorem 6. *There exists a* $\mathcal{H}$ *and* $\mathcal{X}$ *such that the number of labeled examples queried by the E-VS bisection algorithm is* $O(\log |\mathcal{X}|)$*, while the VS bisection algorithm queries* $\Omega(|\mathcal{X}|)$.

Proof. From Lemma 12, we have shown the first part of the theorem. It remains to analyze the VS bisection query complexity.

VS bisection algorithm complexity: By contrast, we show that there exists a labeling oracle that induces $\Omega(n)$ sample complexity from the VS algorithm.

This labeling oracle T is as follows:
i) $T(x) = \perp$ for all $x \in \mathcal{X}_2$
ii) $T(x) = -1$ for all $x \in \mathcal{X}_1$

Under T, we have that labeling each point $x \in \mathcal{X}_1$ removes two hypotheses from the version space at any step in time. Namely, labeling $x_i^1 = [\frac{2i-1}{2n}, 0]$ removes $h_{f_i, f_i'}, h_{f_i, f_{i+1}'}$.

And so, $|\mathcal{X}_1| - 1$ samples $x \in \mathcal{X}_1$ will be queried. Because if there exists two unqueried points $x_i^1, x_j^1 \in \mathcal{X}_1$, then $h_{f_i, f_i'}$ and $h_{f_j, f_j'}$ are both in the VS. This means that the disagreement region is non-empty, and in particular contains both x_i^1, x_j^1.

Since each $x \in \mathcal{X}_1$ is given a binary label by T, the VS bisection algorithm incurs cost $n-1$. We note that in the end the VS will be of size 2, but the remaining sample in $\mathcal{X}_1$ cannot distinguish between the two. $\square$

We may also obtain a corresponding result for an identified setting, by tweaking the above setting slightly. In this setting, we still find that the VS-bisection algorithm still incurs an exponentially larger sample complexity relative to E-VS bisections.

Proposition 16. *There exists a $\mathcal{H}$, $\mathcal{X}$, and a labeling oracle that leads to identification, and the number of labeled examples queried by the E-VS bisection algorithm is $O(\log|\mathcal{X}|)$, while the VS bisection algorithm incurs $\Omega(|\mathcal{X}|)$ samples.*

Proof. **Setup:** Let $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \{\tilde{x}\}$, where $\mathcal{X}_1 = \left\{x_1^1, ..., x_{n-1}^1\right\} = \left\{\frac{1}{2n}, ..., \frac{2n-3}{2n}\right\}$, $\mathcal{X}_2 = \left\{x_1^2, ..., x_{n-1}^2\right\} = \left\{1 + \frac{3}{2n}, ..., 1 + \frac{2n-1}{2n}\right\}$, and $\tilde{x} = -\frac{1}{2n}$. So $|\mathcal{X}| = 2(n-1)+1$.

Let the $f_i : [-1,1] \to \{+1,-1\}$ denote intervals of length $1/n$, $f_i(x) = \mathbb{1}(x \in [(i-1)/n, i/n])$ for $i \in \{0, 1, \ldots, n-1\}$.

Let $f_i' : (1,2] \to \{+1,-1\}$ denote thresholds, $f_i'(x) = \mathbb{1}(x \geq 1 + i/n)$ for $i \in [n]$.

Define $\mathcal{H} = \bigcup_{i=1}^{n-1} \left\{h_{f_i,f_i'}, h_{f_i,f_{i+1}'}\right\} \cup \left\{h_{f_0,f_1'}\right\}$, where $h_{f,f'}(x) = \begin{cases} f(x), & x \leq 1 \\ f'(x), & x > 1 \end{cases}$.

Ensuring identifiability: Note that obtaining labeled example $(\tilde{x}, +1)$ identifies $\tilde{h} := h_{f_0,f_1'}$.

E-VS bisection algorithm complexity:

Note that for any V, S_X, $\rho(E(V,S_X),\tilde{x}) \leq 1$.

And so, in the case analysis of Lemma 14, we again find that as long as $|U_2| \geq 7$, the E-VS algorithm will query some point $x \in U_2$.

Thus, the E-VS algorithm will query at most $\log n$ labeled samples before reaching $|U_2| \leq 6$, at which point the E-VS contains at most $2 \cdot 6 + 2$ hypotheses and will thus require at most 13 more labeled examples before identification.

VS bisection algorithm complexity: We show that there exists an identifiable labeling oracle that induces $\Omega(n)$ samples with the VS algorithm.

This labeling oracle T goes as follows:

i) $T(x) = \perp$ for all $x \in \mathcal{X}_2$

ii) $T(x) = -1$ for all $x \in \mathcal{X}_1$

iii) $T(\tilde{x}) = 1$

It is clear that $\mathcal{H}[T(\mathcal{X})] = \left\{\tilde{h}\right\}$ and T is an identifiable oracle.

The main observation is that while $|S_X \cap \mathcal{X}_1| < |\mathcal{X}_1| - 1$, if a point in $\mathcal{X} \setminus \mathcal{X}_2$ is queried, then it will be a point in $\mathcal{X}_1$, and not $\tilde{x}$.

This is because $\tilde{x}$ for any V, S_X, is such that $\rho(E(V,S_X),\tilde{x}) = 1$. While for any $x \in \mathcal{X}_1 \setminus S_X$, $\rho(E(V,S_X),x) = 2$.

In more detail, if $x_i^1 \notin S_X$, then $h_{f_i,f_i'}, h_{f_i,f_{i+1}'} \in V[S]$, whose label for x_i^1 is $[1,-1]$. And when $|S_X \cap \mathcal{X}_1| < |\mathcal{X}_1| - 1$, there exists at least two other models in $V[S]$ that label x_i^1 with $[-1,-1]$.

Hence, since T never abstains on $x \in \mathcal{X}_1$, $|\mathcal{X}_1| - 1$ labels will be given, at which point the disagreement region is still non-empty. Then, the algorithm either queries the $\tilde{x}$ or the remaining element in $\mathcal{X}_1$ depending on the tie-breaker, both of which identifies $\tilde{h}$. $\square$

2.10.4 Comparison with EPI-CAL

EPI-CAL [146] is a "mellow" active learning algorithm that can handle labeler abstention in a streaming setting, wherein the learner *cannot* control the query order (unlike Algorithm 2), and performs PAC learning [282]. Despite the differences between this and our pool-based setup, we can nevertheless analyze what happens when the labeler can strategically abstain. Our finding is that a strategic labeler can again hold up learning and induce an arbitrarily large query complexity, when the data pool size is not finite and the query order cannot be decided by the learner. This may be evidenced in the simple setting of learning thresholds, where we note that the stream samples are drawn i.i.d, from a continuous distribution satisfying a standard regularity condition.

In particular, we find that in the infinite-support case, even if the data stream is made up of i.i.d samples, EPI-CAL can incur large sample complexity. This is because the learner experiences an arbitrarily large "hold-up", which may be evidenced even in the simple threshold example in the lemma below.

Proposition 17. *Fix some constant $\epsilon > 0$. Consider a PAC-learning task, where the learner seeks to learn a 1D threshold with at most $\epsilon-$risk with respect to continuous distribution $\mathcal{D}$. For any m i.i.d samples with m sufficiently large and $\mathcal{D}$ probability density bounded away from 0, there is a labeling strategy under which EPI-CAL queries $\Omega(\sqrt{m})$ labeled samples, with probability at least $1/2$.*

Proof. Let $h^* = h_0$ for the 1D threshold hypothesis class $\mathcal{H} = \{h_\theta = \mathbb{1}(x \geq \theta) : \theta \in [0,1]\}$.

Let $\mathcal{D}$ be some continuous distribution with $\text{supp}(\mathcal{D}) = [0,1]$. Let $X_1, .., X_m$ denote the m i.i.d samples from $\mathcal{D}$. By assumption, suppose the pdf of $\mathcal{D}$ is lower bounded by $\kappa > 0$, i.e. $\Pr(x) \geq \kappa, \forall x \in \text{supp}(\mathcal{D})$.

Then, $\Pr_{x\sim\mathcal{D}}(x \in (\epsilon, 1]) = \beta \geq (1-\epsilon)\kappa = \Omega(1)$.

Under $m \geq 6$, consider some β_0 with $\beta_0 \leq \frac{\ln\frac{4}{3}}{2m}$. Since the CDF is continuous, there exists r such that $\Pr_{x\sim\mathcal{D}}(x \leq r) < \beta_0$, which is such that:

$$\Pr(\forall i \in [m], x_i \notin [0,r]) \geq (1-\beta_0)^m \geq \exp(-2m\beta_0) \geq \frac{3}{4}$$

using that $1 - x \geq \exp(-2x)$ when $x \in [0, 1/2]$.

Define $\hat{r} = \min(r, \epsilon)$, which also satisfies the condition above since $[0,\hat{r}] \subseteq [0,r]$.

Now, we proceed to defining the labeling strategy:

1. Let $M = \sqrt{m}$. Using the continuity of $\Pr_{x\sim\mathcal{D}}(x < r)$ in r, we can find $1 = r_1 > ... > r_M > r_{M+1}$ with $r_{M+1} = \epsilon$, such that:

$$\Pr_{x\sim\mathcal{D}}(x \in [r_{i+1}, r_i]) = \frac{\beta}{M}$$

   Let $S_i = (r_{i+1}, r_i]$ for $i \in [M]$.
2. We make the observation that if EPI-CAL has only seen points from $S_{i_1}, ..., S_{i_j}$, then any point $x_k \in S_k$ with $k > \max(i_1, ..., i_j)$ will be accepted (bigger index means close to θ^*). This is because with labeled points only from $S_{i_1}, ..., S_{i_j}$, the resultant VS is a superset of $[0, r_{\max(i_1,...,i_j)+1}]$.

   And so, x_k is in the disagreement region, since $x_k \leq r_{\max(i_1,...,i_j)+1}$.

3. Now, we describe the sequential labeling strategy.
   a) Abstain on the region: $[\hat{r}, \epsilon]$.
   b) Label if $X_i \in [0, \hat{r})$. Note that labeling $[0, \hat{r})$ ensures that $\epsilon-$PAC learning is possible.
   For $X_i \in (\epsilon, 1]$, sequentially label as follows:
   i) Divide the m samples into M stages of M samples for $M = \sqrt{m}$.
   ii) At the ith stage, abstain if on the jth sample of this stage, $X_{ij} \notin S_i$.
   iii) The first time sample X_{ik} for $k \in [M]$ is such that $X_{ik} \in S_i$, label it and abstain for the rest of this stage.
   Using our previous point, we know that any point $X_{ik} \in S_i$ labeled will be accepted by EPI-CAL, since i is increasing.
   Intuitively, this labeling strategy slows down learning by only labeling points that shrink the VS by a little.
4. To analyze the total number of labeled points, let random variable Z_i denote whether a point is labeled at stage i. It is Bernoulli with probability:

   $$p = \Pr(\exists j \in [M], X_{ij} \in [r_{i+1}, r_i]) = 1 - (1 - \beta/M)^M \geq 1 - \exp(-\beta) = \Omega(1)$$

   Using one-sided Chernoff's for Binomial random variables for M sufficiently large (i.e. for $M \geq \frac{8 \ln 4}{p}$) with p constant, we have:

   $$\Pr(\sum_{i=1}^{M} Z_i \leq Mp/2) \leq \exp(-Mp/8) \leq 1/4$$

5. And so, using union bound, we have that:

   $$\begin{aligned}&\Pr(x_i \notin [0, \hat{r}], \forall i \in [m] \wedge \sum_{i=1}^{M} Z_i \geq Mp/2)\\ &\geq 1 - \Pr(\exists i \in [m], x_i \in [0, \hat{r}]) - \Pr(\sum_{i=1}^{M} Z_i < Mp/2)\\ &\geq 1 - 1/4 - 1/4\\ &= 1/2\end{aligned}$$

   And so, the probability that all m samples are seen (i.e. the interaction does not terminate before all m), and that at least $Mp/2 = \Omega(\sqrt{m})$ samples are labeled and accepted by EPI-CAL occurs with probability at least $1/2$.

□

Remark 6. *We remark that:*

- *Consider when there is no labeler abstention. Let $Z'_i = \mathbb{1}(x_i \leq \min_{j \in [i-1]} x_j)$. Then we see that the expected sample complexity is:*

  $$\mathbb{E}[\sum_{i=1}^{m} Z'_i] = \sum_{i=1}^{m} 1/i = O(\log m)$$

Thus, we see that this is yet another setting, where labeler abstention can significantly increase the sample complexity.

- *From the Erdős–Szekeres theorem, the $\Theta(\sqrt{m})$ result is tight in expectation.*

2.11 Additional Material on Section 2.4

In this section, we examine a few ways in which the labeler (e.g. a human worker) may be imperfect in both labeling and strategy, and extend our guarantees to such settings. We elaborate on the content covered in Section 2.4.

Note that in this paper, we make inroads into understanding the minimax strategies of the learning game. Analyzing minimax strategies is the canonical way of characterizing games, studying how players (e.g. a data provider company) may play rationally in the learning game. However, it has been recognized that players with bounded rationality (e.g. a human worker) may play behavioral strategies that are not minimax-optimal [53]. And so, we consider allow for the labeler labeling in a way that is sub-optimal.

2.11.1 Relaxed Learning Goal

In the previous section, it is assumed that the learner is interested in exact learning some h^*. One may consider the relaxed goal of PAC learning some $\hat{h}$ such that $\Pr_{x\sim\mathcal{D}}(\hat{h}(x) \neq h^*(x)) \leq \epsilon$ w.p. greater than $1-\delta$, for some distribution $\mathcal{D}$ supported on $\mathcal{X}$.

Reduction: Then, following the standard realizable, PAC learning (with VC class) reduction [283], one may reduce the PAC setting to the exact learning by sampling $m = O(\frac{VC(\mathcal{H})}{\epsilon}(\ln\frac{1}{\epsilon} + \ln\frac{1}{\delta}))$ i.i.d samples from $\mathcal{D}$.

More precisely, let this random subset be $X^m \subseteq \mathcal{X}$. X^m partitions $\mathcal{H}$ into clusters of equivalent hypotheses. If we let the projection of $\mathcal{H}$ on X^m be $\mathcal{H}_{|X^m} = \big\{h(X^m) : h \in \mathcal{H}\big\}$, then a cluster $C(y)$ of equivalent hypotheses is defined $C(y) = \big\{h(X^m) = y : y \in \mathcal{H}_{|X^m}, h \in \mathcal{H}\big\}$.

The reduction guarantees that, with probability better than $1-\delta$ over the samples X^m, identification of h^*'s cluster $C(h^*(X^m))$ is sufficient for $\epsilon-$PAC learning. X^m is such that w.h.p $\mathrm{diam}(C(h^*(X^m)) \leq \epsilon$, where diameter of a set H is defined as $\mathrm{diam}(H) = \max_{h,h'\in H} \Pr_{x\sim D}(h(x) \neq h'(x))$. With this, picking any one model $\hat{h} \in C(h^*(X^m))$ satisfies $\Pr_{x\sim\mathcal{D}}(\hat{h}(x) \neq h^*(x)) \leq \epsilon$, and PAC learning thus reduces to identifying cluster $C(h^*(X^m))$.

2.11.1.1 Approximate Identification Game

Using this reduction, we may analyze the query complexity of PAC learning as an exact learning game, where the learner chooses the data pool to be X^m (in place of $\mathcal{X}$). The goal is now only approximate identifiability, and identifying the cluster h^* belongs to, $C(h^*(X^m))$.

We demonstrate how our E-VS definition can be extended to develop a near-optimal algorithm under this approximate identifiable game. Our first observation is that the original E-VS, defined over $\mathcal{H}$ and X^m will no longer suffice:

$$E(V, S_X) = \big\{h \in V : \forall h' \in V \setminus \{h\} : h'(X^m \setminus S_X) \neq h(X^m \setminus S_X)\big\}$$

The issue is premature elimination. Consider some $h \in \mathcal{H}$ such that $|C(h(X^m))| \geq 2$ with $h' \in C(h(X^m)), h' \neq h$. Then, $h(X^m) = h'(X^m) \Rightarrow \exists h' \in \mathcal{H}, h'(X^m \setminus \emptyset) = h(X^m \setminus \emptyset)$, which results in the elimination of the entire $C(h(X^m))$ cluster at the very start. $E(\mathcal{H}, \emptyset)$ will not contain any clusters with cardinality more than one.

To address this degeneracy, we define a modification of the E-VS, X^m-E-VS, with relaxed elimination condition. This is a coarser E-VS, and so, we observe that we should only consider non-identifiability with respect to hypotheses from other clusters:

$$E^{X^m}(V, S_X) = \Big\{h \in V : \forall h' \in V \setminus \big\{\bar{h} : \bar{h}(X^m) = h(X^m), \bar{h} \in V\big\} : h'(X^m \setminus S_X) \neq h(X^m \setminus S_X)\Big\}$$

The added constraint of $V \setminus \big\{\bar{h} : \bar{h}(X^m) = h(X^m), \bar{h} \in V\big\}$ means that two h, h' within the same cluster do not render each other un-identifiable. And so, we only consider h''s from another cluster (that differs on X^m) that can render h (h's cluster) un-identifiable.

Remark 7. *Through this we see that either an entire cluster is in the X^m-E-VS or it is not.*

We also define the global identification cost in the approximate identification game accordingly:

Definition 13. *Given $\mathcal{H}$ and a set of unlabeled examples X^m, define the global identification cost of version space $V \subset \mathcal{H}$ and instance set S_X:*

$$\begin{aligned}\mathrm{GIC}^{X^m}(V, S_X) = \min\{t \in \mathbb{N} : \forall T : X^m \setminus S_X \to \{+1, -1, \perp\}\,,\\ \exists \Sigma \subseteq X^m \setminus S_X \textit{ s.t. } \sum_{x \in \Sigma} \mathbb{1}(T(x) \neq \perp) \leq t \wedge |E^{X^m}(V[T(\Sigma)], S_X \cup \Sigma)| \leq 1\}.\end{aligned}$$

Under the new definitions of X^m-E-VS and X^m-GIC, we may prove that the X^m-E-VS bisection algorithm similarly attains near-optimal guarantees. One may follow the same proof structure as in Lemma 10 and Theorem 5 to show both results also hold under X^m-E-VS. Thus, it suffices to prove the following two lemmas, which are analogues of Lemmas 5 and 6.

Lemma 15. *For any $V \subset \mathcal{H}$ and $S_X \subset \mathcal{X}$,*

$$E^{X^m}(V, S_X \cup \{x\}) \subseteq E^{X^m}(V, S_X)$$

Proof. It suffices to prove that $h \in E^{X^m}(V, S_X \cup \{x\}) \Rightarrow h \in E^{X^m}(V, S_X)$.

To see this, let $h \in E^{X^m}(V, S_X \cup \{x\})$. Then if h is such that:

$$\begin{aligned}&\forall h' \in V, h'(X^m) \neq h(X^m), h((\mathcal{X} \setminus S_X) \setminus \{x\})) \neq h'((\mathcal{X} \setminus S_X) \setminus \{x\}))\\ &\Rightarrow \forall h' \in V, h'(X^m) \neq h(X^m), h(\mathcal{X} \setminus S_X) \neq h'(\mathcal{X} \setminus S_X)\\ &\Rightarrow h \in E(V, S_X)\end{aligned}$$

□

Lemma 16. *For any $x \in \mathcal{X} \setminus S_X$ and $y \in \{-1, 1\}$,*

$$E^{X^m}(V[(x, y)], S_X \cup \{x\}) = E^{X^m}(V, S_X)[(x, y)]$$

Proof. The proof is identical to the one for the fine-grain E-VS:

$$
\begin{aligned}
& h \in E^{X^m}(V[(x,y)], S_X \cup \{x\}) \\
\iff & h \in V[(x,y)] \wedge \forall h' \in V[(x,y)] \,.\, h'(X^m) \neq h(X^m) \to h'(X^m \setminus (S_X \cup \{x\})) \neq h(X^m \setminus (S_X \cup \{x\})) \\
\iff & h \in V \wedge h(x) = y \wedge \forall h' \in V[(x,y)] \,.\, h'(X^m) \neq h(X^m) \to h'(X^m \setminus (S_X \cup \{x\})) \neq h(X^m \setminus (S_X \cup \{x\} \\
\iff & h \in V \wedge h(x) = y \wedge \forall h' \in V \,.\, h'(X^m) \neq h(X^m) \to h'(\mathcal{X} \setminus S_X) \neq h(\mathcal{X} \setminus S_X) \\
\iff & h(x) = y \wedge h \in E^{X^m}(V, S_X) \\
\iff & h \in E^{X^m}(V, S_X)[(x,y)]
\end{aligned}
$$

□

Guarantee from learning from labeler with h' that approximates h^*: Suppose the labeler labels with h' and $\Pr(h'(x) \neq h^*(x)) \leq \epsilon/2$. One may consider the approximate identifiability learning game with precision $\epsilon/2$. Approximately-identifying some $\hat{h} \in C(h'(X^m))$ will be such that $\Pr(\hat{h}(x) \neq h'(x)) \leq \epsilon/2$. From this, we can conclude that:

$$
\begin{aligned}
\Pr(\hat{h}(x) \neq h^*(x)) &= \Pr(\hat{h}(x) = h'(x) \wedge h'(x) \neq h^*(x)) + \Pr(\hat{h}(x) \neq h'(x) \wedge h'(x) = h^*(x)) \\
&\leq \Pr(h'(x) \neq h^*(x)) + \Pr(\hat{h}(x) \neq h'(x)) \\
&\leq \epsilon
\end{aligned}
$$

2.11.1.2 Accessing the X^m−E-VS

After modifying the E-VS definition, the remaining issue is that we wish to find the maximal bisection point for coarse, X^m-E-VS. Here, we show that for the coarsened E-VS, the membership check implemented in Algorithm 3 (with the pool being X^m) is still sound. That is, we still have an oracle-efficient way of accessing the coarser X^m-E-VS, and can can implicitly track clusters through calls to the C-ERM oracle.

Proposition 18. *$h \notin E_{X^m}(V, S_X)$ iff $\hat{h}(X^m) \neq h(X^m)$, where $\hat{h}$ is the minimizer of the C-ERM output below:*

$$
\begin{aligned}
\hat{h} = \operatorname*{argmin}_{h' \in \mathcal{H}} & \sum_{x' \in S^\perp} \mathbb{1}\left\{h'(x') = h(x')\right\} \\
& \textit{s.t } h'(x) = h(x), \forall x \in X^m \setminus S^\perp
\end{aligned}
$$

Proof.

$$
\begin{aligned}
\neg(h \in E_{X^m}(V, S_X)) &\Leftrightarrow \neg(\forall h' \in V \setminus \left\{\bar{h} : \bar{h}(X^m) = h(X^m), \bar{h} \in V\right\} \mathbin{.} h'(X^m \setminus S_X) \neq h(X^m \setminus S_X)) \\
&\Leftrightarrow \exists h' \in V \setminus \left\{\bar{h} : \bar{h}(X^m) = h(X^m), \bar{h} \in V\right\} \mathbin{.} h'(X^m \setminus S_X) = h(X^m \setminus S_X) \\
&\Leftrightarrow \exists h' \in V \mathbin{.} h'(X^m) \neq h(X^m) \mathbin{.} h'(X^m \setminus S_X) = h(X^m \setminus S_X) \\
&\Leftrightarrow \exists h' \mathbin{.} h'(S^X \setminus S^\perp) = h(S^X \setminus S^\perp) \mathbin{.} h'(X^m) \neq h(X^m) \mathbin{.} h'(X^m \setminus S_X) = h(X^m \setminus S_X) \\
&\Leftrightarrow \exists h' \mathbin{.} h'(S^X \setminus S^\perp) = h(S^X \setminus S^\perp) \mathbin{.} h'(S^\perp) \neq h(S^\perp) \mathbin{.} h'(X^m \setminus S_X) = h(X^m \setminus S_X) \\
&\Leftrightarrow \exists h' \mathbin{.} h'(S^\perp) \neq h(S^\perp) \mathbin{.} h'(X^m \setminus S^\perp) = h(X^m \setminus S^\perp) \\
&\Leftrightarrow \exists h' \mathbin{.} \sum_{x' \in S^\perp} \mathbb{1}\left\{h'(x') = h(x')\right\} < |S^\perp| \mathbin{.} h'(X^m \setminus S^\perp) = h(X^m \setminus S^\perp) \\
&\Leftrightarrow \hat{h}(X^m) \neq h(X^m) \mathbin{.} \hat{h}(X^m \setminus S^\perp) = h(X^m \setminus S^\perp)
\end{aligned}
$$

□

2.11.2 Noised labeling

It may be reasonable that in some cases, a labeler can make mistakes (even when they have tried their best) due to differing opinion and/or human error. For example, for medical diagnoses, doctors may hold differing opinions for the same case. This can be naturally modeled by the noised learning setting, as in [57]: querying example x returns $h^*(x)$ with known probability $1 - \delta(x)$, and $-h^*(x)$ with noise rate $\delta(x)$.

In this setup, we may use the common approach of repeatedly query a datum to estimate its label w.h.p. (e.g. as [303]). This approach reduces noised-label exact learning to cost-sensitive exact learning, where for each x there is some known query cost $c(x)$ — associated with determining $h^*(x)$ with high probability. With this, we may apply the results from Subsection 2.9.2 to see that E-VS bisection algorithm will have near-optimal guarantees in this setting with example-dependent costs.

2.11.3 Myopic labeling

Some labelers may want to enlarge the query complexity, but myopically may not have a near-optimal identifiable strategy. Instead, the labeler may have only a heuristic, which is only h^*-labeling, and can be non-identifiable. Non-identifiability is something neither parties want: the learner wants to learn h^*, and the labeler wants to be paid, which can only happen if h^* is learned.

In this light, we believe that the E-VS game representation is not only useful for the learner, but also for a labeler to reason about the game's state. For the labeler, there is an oracle-efficient way through which identifiability can be checked without enumerating the entire E-VS: simply apply the membership check on h^* as in Line 3 of Algorithm 3.

So even if the labeler is using some sub-optimal heuristic that may lead to non-identifiability of h^*, the labeler can prevent the next label from leading to non-identifiability by performing a membership check with a single C-ERM call. We add that only verifying that h^* is in E-VS, need not require enumerating all of the E-VS, and is thus tractable provided access to a C-ERM oracle.

2.12 Proofs for Section 2.5

2.12.1 Lemmas used

Lemma 17. *For all* $V = \times_{i=1}^n V_i$ *and* S_X*,*

$$E(V, S_X) = \times_{i=1}^n E(V_i, S_X^i)$$

Proof. We show both that:

1. For $V = \times_{i=1}^n V_i$, $\times_{i=1}^n E(V_i, S_X^i) \subseteq E(V, S_X)$:
   It suffices to show that if $h_i \in E(V_i, S_X^i)$ for $i \in [n]$, then $h = (h_1, .., h_n) \in E(V, S_X)$ for $V = \times_{i=1}^n V_i$.
   Firstly, since $h_i \in V_i$ and $V = \times_{i \in [n]} V_i$, we have that $h \in V$.
   Now suppose there is some $h' \in V$ such that $h'(\mathcal{X} \setminus S_X) = h(\mathcal{X} \setminus S_X)$; we would like to show that $h' = h$ – proving this would conclude that $h \in E(V, S_X)$.
   Indeed, consider any i; we have $h'_i((\mathcal{X} \setminus S_X)_i) = h_i((\mathcal{X} \setminus S_X)_i)$; equivalently, $h'_i(\mathcal{X}_i \setminus S_X^i) = h_i(\mathcal{X}_i \setminus S_X^i)$.
   As $h_i \in E(V_i, S_X^i)$ and $h'_i \in V_i$, we have that $h'_i = h_i$. Therefore h' and h are equal in all components, and $h' = h$.
2. For $V = \times_{i=1}^n V_i$, $E(V, S_X) \subseteq \times_{i=1}^n E(V_i, S_X^i)$:
   Consider any $h \in E(V, S_X)$; we would like to show that for any i, $h_i \in E(V_i, S_X^i)$.
   Suppose not, then there exists i, $h' \in V_i$ and $h' \neq h_i$ such that $h'(\mathcal{X}_i \setminus S_X^i) = h_i(\mathcal{X}_i \setminus S_X^i)$.
   This implies that $h'((\mathcal{X} \setminus S_X)_i) = h_i((\mathcal{X} \setminus S_X)_i)$, therefore, consider

   $$\tilde{h} = (h_1, \ldots, h_{i-1}, h', h_{i+1}, \ldots, h_n)$$

   We have that $\tilde{h} \in V$, $\tilde{h} \neq h$, and $\tilde{h}$ agrees with h on $\mathcal{X} \setminus S_X$, which contradicts the assumption that $h \in E(V, S_X)$.

□

Lemma 18. *For any data point* (x_1, y_1) *for* $x_1 \notin S_X$ *and* $y_1 \in \{+1, -1, \perp\}$*:*

$$\text{Cost}(V[(x_1, y_1)], S_X \cup \{x_1\}) \leq \text{Cost}(V, S_X)$$

Proof. We prove this by induction on $|S_X|$.

Base Case:

The base case is when $|S_X| = |\mathcal{X}| - 1$. Here $S_X \cup \{x_1\} = \mathcal{X}$. We have two subcases:

- $E(V[(x_1, y_1)], S_X \cup \{x_1\}) = \emptyset$.
  In this case, the inequality is satisfied.
- $|E(V[(x_1, y_1)], S_X \cup \{x_1\})| = 1$.
  We will show in general that $E(V[(x_1, y_1)], S_X \cup \{x_1\}) \subseteq E(V, S_X)$:
  i) If $y \neq \perp$, we know from Lemma 6 that $E(V[(x_1, y_1)], S_X \cup \{x_1\}) = E(V, S_X)[(x_1, y_1)] \subseteq E(V, S_X)$.
  ii) If $y = \perp$, then $E(V[(x_1, y_1)], S_X \cup \{x_1\}) = E(V, S_X \cup \{x_1\}) \subseteq E(V, S_X)$.
  And so, $|E(V, S_X)| \geq 1 \Rightarrow \text{Cost}(V, S_X) \geq 0 = \text{Cost}(V[(x_1, y_1)], S_X \cup \{x_1\})$.

Induction Step:

For the inductive case, suppose the induction hypothesis holds for $|S_X| = |\mathcal{X}| - 1, .., j + 1$. Consider some S_X with $|S_X| = j$.

We have three subcases:

- $E(V[(x_1, y_1)], S_X \cup \{x_1\}) = \emptyset$
  In this case, the inequality is satisfied.
- $|E(V[(x_1, y_1)], S_X \cup \{x_1\})| = 1$
  As shown before, $E(V[(x_1, y_1)], S_X \cup \{x_1\}) \subseteq E(V, S_X)$.
  And so, we have that $|E(V, S_X)| \geq 1 \Rightarrow \text{Cost}(V, S_X) \geq 0 = \text{Cost}(V[(x_1, y_1)], S_X \cup \{x_1\})$.
- $|E(V[(x_1, y_1)], S_X \cup \{x_1\})| \geq 2$.
  Using similar logic as before, $|E(V[(x_1, y_1)], S_X \cup \{x_1\})| \geq 2 \Rightarrow |E(V, S_X)| \geq 2$.
  Define
  $$x' \in \operatorname*{argmin}_{x \in \mathcal{X} \setminus S_X} \max_y \mathbb{1}(y \neq \perp) + \text{Cost}(V[(x', y)], S_X \cup \left\{x'\right\})$$
  With this definition,
  $$\text{Cost}(V, S_X) = \max_y \mathbb{1}(y \neq \perp) + \text{Cost}(V[(x', y)], S_X \cup \left\{x'\right\})$$
  If $x' = x_1$, then the result follows since $\text{Cost}(V, S_X) \geq \mathbb{1}(y_1 \neq \perp) + \text{Cost}(V[(x_1, y_1)], S_X \cup \{x_1\})$.
  If $x' \neq x_1$, then $x' \in \mathcal{X} \setminus S \cup \{x_1\}$, and we can write:
  $$\begin{aligned}
  \text{Cost}(V[(x_1, y_1)], S_X \cup \{x_1\}) &\leq \max_y \mathbb{1}(y \neq \perp) + \text{Cost}(V[(x_1, y_1), (x', y)], S_X \cup \left\{x_1, x'\right\}) \\
  &\qquad \text{(as } |E(V[(x_1, y_1)], S_X \cup \{x_1\})| \geq 2 \text{ so we can unroll, and } x' \in \mathcal{X} \setminus S \cup \{x_1\}) \\
  &\leq \max_y \mathbb{1}(y \neq \perp) + \text{Cost}(V[(x', y)], S_X \cup \left\{x'\right\}) \\
  &\qquad \text{(using induction hypothesis since } |S_X \cup \{x'\}| = j + 1) \\
  &= \text{Cost}(V, S_X)
  \end{aligned}$$

□

Lemma 19. *For $y \neq \perp$, $x \in \mathcal{X} \setminus S_X$:*

$$\text{Cost}(V[(x, y)], S_X) = \text{Cost}(V[(x, y)], S_X \cup \{x\})$$

Proof. Firstly, we have that:

$$\begin{aligned}
E(V[(x, y)], S_X) &= \left\{h \in V[(x, y)] : \forall h' \in V[(x, y)] \setminus \{h\}, h'(\mathcal{X} \setminus S_X) \neq h(\mathcal{X} \setminus S_X)\right\} \\
&= \left\{h \in V[(x, y)] : \forall h' \in V[(x, y)] \setminus \{h\}, h'(\mathcal{X} \setminus (S_X \cup \{x\}) \neq h(\mathcal{X} \setminus S_X \cup \{x\})\right\} \\
&= E(V[(x, y)], S_X \cup \{x\})
\end{aligned}$$

Hence the statement holds when $S_X = \mathcal{X} \setminus \{x\}$, or more generally, when $\text{Cost}(V[(x, y)], S_X \cup \{x\})$ or $\text{Cost}(V[(x, y)], S_X)$ is at its base case (one implies the other due to having the same E-VS).

Now, we will induct on the size of $|S_X|$, since the base case of $S_X = \mathcal{X} \setminus \{x\}$ is satisfied.

Base case: $|S_X| = |\mathcal{X}| - 1$.

If $E(V, S_X) = E(V, S_X \cup \{x\}) = \emptyset$, then $LHS = RHS = -\infty$;

If $|E(V, S_X)| = |E(V, S_X \cup \{x\})| = 1$, then $LHS = RHS = 0$.

Induction Step: Suppose the statement holds for when $|S_X| = |\mathcal{X}|, ..., j+1$. Let $|S_X| = j$.

We first handle the base cases:

If $E(V, S_X) = E(V, S_X \cup \{x\}) = \emptyset$, then $LHS = RHS = -\infty$;

If $|E(V, S_X)| = |E(V, S_X \cup \{x\})| = 1$, then $LHS = RHS = 0$.

Finally, it remains to consider when $|E(V, S_X)| = |E(V, S_X \cup \{x\})| \geq 2$. In this case,

$$\mathrm{Cost}(V, S_X) = \min_{x' \in \mathcal{X} \setminus S_X} \max_{y' \in \{+1,-1,\perp\}} \mathbb{1}(y' \neq \perp) + \mathrm{Cost}(V[(x', y')], S_X \cup \left\{x'\right\}).$$

Define $x^* \in \mathrm{argmin}_{x' \in \mathcal{X} \setminus S_X} \max_{y' \in \{+1,-1,\perp\}} \mathbb{1}(y' \neq \perp) + \mathrm{Cost}(V[(x', y')], S_X \cup \{x'\})$.

We will show that $x^* \neq x$.

In fact, for any $x' \in \mathcal{X} \setminus S$, $x' \neq x^*$ (which exists because $\{x\} \subset \mathcal{X} \setminus S_X$) we have:

$$\begin{aligned}
&\max_{y' \in \{+1,-1,\perp\}} \mathbb{1}(y' \neq \perp) + \mathrm{Cost}(V_x^y[(x, y')]], S_X \cup \{x\}) \\
&= \max(1 + \mathrm{Cost}(V_x^y, S_X \cup \{x\}), 1 + \mathrm{Cost}(\emptyset, S_X \cup \{x\}), \mathrm{Cost}(V_x^y, S_X \cup \{x\})) \\
&= 1 + \mathrm{Cost}(V_x^y, S_X \cup \{x\}) && \text{(maximized at when } y' = y) \\
&\geq \max_{y' \in \{+1,-1,\perp\}} \mathbb{1}(y' \neq \perp) + \mathrm{Cost}(V_x^y[(x', y')], S_X \cup \left\{x, x'\right\}) \\
& && \text{(using } 1 \geq \mathbb{1}(y \neq \perp) \text{ and Lemma 18)} \\
&= \max_{y' \in \{+1,-1,\perp\}} \mathbb{1}(y' \neq \perp) + \mathrm{Cost}(V_x^y[(x', y')], S_X \cup \left\{x'\right\}) \\
& && \text{(using induction hypothesis since } |S_X \cup \{x'\}| = j+1)
\end{aligned}$$

And so,

$$\begin{aligned}
\mathrm{Cost}(V[(x, y)], S_X) &= \min_{x' \in \mathcal{X} \setminus S_X} \max_{y' \in \{+1,-1,\perp\}} \mathbb{1}(y' \neq \perp) + \mathrm{Cost}(V_x^y[(x', y')]], S_X \cup \left\{x'\right\}) \\
&= \min_{x' \in \mathcal{X} \setminus (S_X \cup \{x\})} \max_{y' \in \{+1,-1,\perp\}} \mathbb{1}(y' \neq \perp) + \mathrm{Cost}(V_{x'}^{y'}[(x, y)]], S_X \cup \left\{x'\right\}) \\
& \qquad \text{(since we have just shown that } x^* \neq x) \\
&= \min_{x' \in \mathcal{X} \setminus (S_X \cup \{x\})} \max_{y' \in \{+1,-1,\perp\}} \mathbb{1}(y' \neq \perp) + \mathrm{Cost}(V_{x'}^{y'}[(x, y)]], (S_X \cup \left\{x'\right\}) \cup \{x\}) \\
& \qquad \text{(using induction hypothesis since } |S_X \cup \{x'\}| = j+1) \\
&= \min_{x' \in \mathcal{X} \setminus (S_X \cup \{x\})} \max_{y' \in \{+1,-1,\perp\}} \mathbb{1}(y' \neq \perp) + \mathrm{Cost}(V_x^y[(x', y')]], (S_X \cup \{x\}) \cup \left\{x'\right\}) \\
& \qquad \text{(rearranging)} \\
&= \mathrm{Cost}(V[(x, y)], S_X \cup \{x\})
\end{aligned}$$

□

2.12.2 Upper Bound

2.12.2.1 Negative Results

Upper Bound when there is Identifiability:

We first observe that without assumptions on the structure of V, there exists a setting, in which the upper bound does not hold.

Proposition 19. *There exists a non-Cartesian product version space $V \subseteq \mathcal{H}$ and query response $S \subseteq (\mathcal{X} \times \mathcal{Y})^*$ such that* $\mathrm{Cost}(V_i, S_X^i) \geq 0$ *for all i, but:*

$$\mathrm{Cost}(V, S_X) \geq \sum_{i=1}^{n} \mathrm{Cost}(V_i, S_X^i) + n - 1$$

Proof. We will construct a V and S such that $\mathrm{Cost}(V, S_X) \geq n - 1$, but $\mathrm{Cost}(V_i, S_X^i) = 0$.

Hypothesis Class: Define thresholds functions $f_1 = \mathbb{1}(x \geq 1/4), f_2 = \mathbb{1}(x \geq 1/2), f_3 = \mathbb{1}(x \geq 3/4)$ for $x \in [0, 1]$.

Define $\mathcal{H}'$ as:

$$\mathcal{H}' = \big\{(f_1, f_2, ..., f_2), (f_2, f_1, ..., f_2), ..., (f_2, f_2, , ..., f_1)\big\}$$

where the jth model has its jth task model as f_1 instead of f_2.

Define the non-Cartesian product hypothesis class as:

$$\mathcal{H} = \mathcal{H}' \cup \big\{(f_2, f_2, ..., f_2), (f_3, f_3, ..., f_3)\big\}$$

We have that $\mathcal{H}_i = \{f_1, f_2, f_3\}$.

Data: Let $\mathcal{X}_1 = \{x_{i1}\}_{i=1}^n$ and $\mathcal{X}_2 = \{x_{i2}\}_{i=1}^n$, where $x_{i1} = 1/3e_i$ and $x_{i2} = 2/3e_i$. Let $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$.

Query Responses: Suppose $S = \big\{(x_{i2}, [\perp, ..., \perp]) : i \in [n]\big\}$.

This means that $S_X = \big\{x_{i2} : i \in [n]\big\}$, and that $S_X^i = \big\{2/3\big\}$, since the only $x \in \mathcal{X}$ such that $x_i = 2/3$ is x_{i2} and $x_{i2} \in S_X$.

Define $V = \mathcal{H}[S] = \mathcal{H}$. And so, $V_i = \{f_1, f_2, f_3\}$.

We have that $E(V_i, S_X^i) = \{f_1\}$, and so, $\mathrm{Cost}(V_i, S_X^i) = 0$.

Now, it remains to show that $E(V, S_X) = \mathcal{H}'$.

Firstly, since $V = \mathcal{H}[S] = \mathcal{H}$, we examine each model in $\mathcal{H}$.

The model $(f_2, f_2, ..., f_2)$ and $(f_3, f_3, ..., f_3)$'s predictions on x_{i1} (for any i) are both $(-1, -1, \ldots, -1)$. Thus, they have the same predictions on $\{x_{i1}\}_{i \in [n]} = \mathcal{X} \backslash S_X$, and so, $(f_2, f_2, ..., f_2), (f_3, f_3, ..., f_3) \notin E(V, S_X)$.

With this, we see that $E(V, S_X) = \mathcal{H}'$, because for the ith element of $\mathcal{H}'$, it disagrees with every other element on x_{i1}.

Finally, we will show that $\mathrm{Cost}(V, S_X) \geq n - 1$.

Consider a labeling strategy that returns label $(-1, ..., -1)$ for any x_{i1} queried.

This strategy identifies some $h \in \mathcal{H}$, since each point in $\mathcal{X}_1$ that is queried removes one model from E-VS. And so, after $n - 1$ queries on points in $\mathcal{X}_1$, the E-VS has one hypothesis and the learning interaction finishes since the identification condition is met.

We note that any querying algorithm will require $n-1$ labeled queries. Each binary labeled example removes only one model from the E-VS, thus $n-1$ labels are required for identification under any querying algorithm. And so, we have that $\mathrm{Cost}(V, S_X) \geq n-1$.

□

Upper Bound when there is no Identifiability:

Proposition 20. *For non-Cartesian product hypothesis class V, there exists V, S such that $\mathrm{Cost}(V_i, S_X^i) = -\infty$ for some i, but $\mathrm{Cost}(V, S_X) \geq 1$.*

Proof. Consider $\mathcal{H} = \big\{(h_1, h_2), (h_3, h_4)\big\}$.

$\mathcal{X} = \big\{[x_1, 0], [0, x_2]\big\}$, where for $x_1, x_2 \neq 0$, $h_1(x_1) \neq h_3(x_1)$ and $h_2(x_2) \neq h_4(x_2)$. $h_1(0) = h_3(0)$ and $h_2(0) = h_4(0)$.

Consider query response $S = \big\{([x_1, 0], [\perp, \perp])\big\}$. $S_X = \big\{[x_1, 0]\big\}$, $S_X^1 = \{x_1\}$, $S_X^2 = \{0\}$.

$V = \mathcal{H}[S] = \mathcal{H}$. $V_1 = \{h_1, h_3\}$ and $V_2 = \{h_2, h_4\}$.

$E(V_1, \{x_1\}) = E(\{h_1, h_3\}, \{x_1\}) = \emptyset$. However, $E(V, \big\{[x_1, 0]\big\}) = \mathcal{H}$, since (h_1, h_2) and (h_3, h_4) differ on $[0, x_2]$.

And so, $1 = \mathrm{Cost}(V, S_X) > \sum_{i=1}^{2} \mathrm{Cost}(V_i, S_X^i) = -\infty$, since $\mathrm{Cost}(V_1, S_X^1) = -\infty$. □

Remark 8. *In conclusion, to show the upper bound, need to impose Cartesian product condition.*

Negative Example motivating the need to assume a particular label cost definition:

When the label cost is c_{one}, there are settings where $\mathrm{Cost}(V, S_X)$ can be much larger i.e. $\mathrm{Cost}(V, S_X) \gg \sum_{i=1}^{n} \mathrm{Cost}(V_i, S_X^i)$.

Proposition 21. *Assuming the version space is a Cartesian product, under label cost $c_{one}(y) = \mathbb{1}(\exists i, y_i \neq \perp)$, there exists V and S such that $\mathrm{Cost}(V_i, S_X^i) = 1$, but $\mathrm{Cost}(V, S_X) = |\mathcal{X}|$. This implies that: $\mathrm{Cost}(V, S_X) > \sum_{i=1}^{n} \mathrm{Cost}(V_i, S_X^i)$.*

Proof. Consider $V = \{h_1, h_2\} \times \{h_3, h_4\}$, where $h_1, h_2 \in V_1$ are thresholds functions $h_1 = \mathbb{1}(x \geq 0), h_2 = \mathbb{1}(x \geq 1)$ and $h_3, h_4 \in V_2$ are also thresholds $h_3 = \mathbb{1}(x \geq 0), h_4 = \mathbb{1}(x \geq 1)$.

$\mathcal{X} = \Big\{[\frac{1}{m+1}, \frac{1}{m+1}], ..., [\frac{m}{m+1}, \frac{m}{m+1}]\Big\}$, which means that $\mathcal{X}_1 = \mathcal{X}_2 = \Big\{\frac{1}{m+1}, ..., \frac{m}{m+1}\Big\}$.

We will show that:

$$\mathrm{Cost}(V, \emptyset) \gg \mathrm{Cost}(V_1, \emptyset) + \mathrm{Cost}(V_2, \emptyset)$$

We first have that $\mathrm{Cost}(V_1, \emptyset), \mathrm{Cost}(V_2, \emptyset) = 1$, since only one labeled sample is needed to distinguish between h_1, h_2 and between h_3, h_4.

However, we have $\mathrm{Cost}(V, \emptyset) \geq m = |\mathcal{X}|$ with the following labeling strategy T:

1) As long as $|S_X| < m-1$, for queried point $[\frac{i}{m+1}, \frac{i}{m+1}]$, return $(\perp, h_3(\frac{i}{m+1}))$.

2) Only when $|S_X| = m-1$, for queried point $[\frac{j}{m+1}, \frac{j}{m+1}]$, return $(h_1(\frac{j}{m+1}), h_3(\frac{j}{m+1}))$.

We can first that this is an identifiable labeling strategy that identifies (h_1, h_3).

And, for any querying algorithm, h^* is only identified when $S_X = \mathcal{X}$.

Thus, $|\mathcal{X}|$ labeled samples need to be queried, making $\mathrm{Cost}(V, \emptyset) = |\mathcal{X}|$.

□

Remark 9. *To prove the above bound, we need to assume the label cost to be: $\mathbb{1}(y \neq \perp) = \mathbb{1}(\forall i, y_i \neq \perp) = c_{all}(y)$.*

2.12.2.2 Positive Results

Change in Definition of the Game:

- To prove the upper bound, we have a changed definition in labeling payoff, which is now:

$$\mathbb{1}(y \neq \perp) := \mathbb{1}(\forall i, y_i \neq \perp)$$

- The earlier negative example motivates requiring the assumption that V is a Cartesian product.

Theorem 7. *For all $V = \times_{i\in[n]} V_i$ and $S_X \subseteq \mathcal{X}$, under labeling cost $c_{all}(y) = \mathbb{1}(\forall i, y_i \neq \perp)$:*

$$\mathrm{Cost}(V, S_X) \leq \sum_{i=1}^{n} \mathrm{Cost}(V_i, S_X^i)$$

Proof. We prove this by induction on the size of S_X.

Base Case: When $S_X = \mathcal{X} \Rightarrow S_X^i = \mathcal{X}_i$. So for all i, $|E(V_i, S_X^i)| \leq 1$.

It suffices to check that $\mathrm{Cost}(V, S_X) = 0 \Rightarrow \forall i, \mathrm{Cost}(V_i, S_X^i) = 0$.

Indeed, if $\mathrm{Cost}(V, S_X) = 0$, then $|E(V, \mathcal{X})| = 1$. Denote by h the only element of $E(V, \mathcal{X})$.

We must have $V = \{h\}$, which in turn implies that for all i $V_i = \{h_i\}$. Therefore, for all i, $|E(V, \mathcal{X})| = \{h_i\} = 1$, which implies $\forall i, \mathrm{Cost}(V_i, S_X^i) = 0$.

Induction Step:

Suppose the following holds for $S_X \subset X$ for $|S_X| = |\mathcal{X}|, ..., j+1$. Now let $|S_X| = j$ (note that $S_X \subset \mathcal{X}$).

We will analyze the three cases:

- $\exists i, \mathrm{Cost}(V_i, S_X^i) = -\infty$
- $\forall i, \mathrm{Cost}(V_i, S_X^i) \geq 0$ and $\forall i, \mathrm{Cost}(V_i, S_X^i) = 0$
- $\forall i, \mathrm{Cost}(V_i, S_X^i) \geq 0$ and $\exists i, \mathrm{Cost}(V_i, S_X^i) \geq 1$.

1. **If there is at least one i such that $\mathrm{Cost}(V_i, S_X^i) = -\infty$.**
   It suffices to verify that $\exists i, E(V_i, S_X^i) = \emptyset \Rightarrow E(V, S_X) = \emptyset$.
   This follows immediately from that $E(V, S_X) = \times_{i=1}^{n} E(V_i, S_X^i)$ (Lemma 17).
2. **For all i, $\mathrm{Cost}(V_i, S_X^i)$ is at its base case and $\mathrm{Cost}(V_i, S_X^i) = 0$.**
   That is, we have $\forall i, |E(V_i, S_X^i)| = 1$.
   From Lemma 17, we have that $E(V, S_X) = \times_{i=1}^{n} E(V_i, S_X^i)$, which means that $|E(V, S_X)| = 1$. And so, $\mathrm{Cost}(V, S_X) = 0 = \sum_{i=1}^{n} \mathrm{Cost}(V_i, S_X^i)$.
3. **Exists i such that $\mathrm{Cost}(V_1, S_X^1) \geq 1$, and $\mathrm{Cost}(V_i, S_X^i) \geq 0$ for all i.**
   Without loss of generality, $i = 1$.
   Note that if $|E(V, S_X)| \leq 1$, then $\mathrm{Cost}(V, S_X) \leq 0 \leq \sum_{=1}^{n} \mathrm{Cost}(V_i, S_X^i)$.
   And so, throughout the rest of the proof, we focus on the case that $|E(V, S_X)| \geq 2$. Also, recall that since $\mathrm{Cost}(V_1, S_X^1) \geq 1$ implies that $E(V_1, S_X^1) \geq 2$.
   Define
   $$x_1^* = \operatorname*{argmin}_{x \in \mathcal{X}_1 \backslash S_X^1} \max_{y \in \mathcal{Y}} \mathbb{1}(y \neq \perp) + \mathrm{Cost}(V_1[(x_1^*, y)], S_X^1 \cup \{x_1^*\})$$

We may express:

$$\mathrm{Cost}(V_1, S_X^1) = \max_{y\in\mathcal{Y}} \mathbb{1}(y \neq\perp) + \mathrm{Cost}(V_1[(x_1^*, y)], S_X^1 \cup \{x_1^*\})$$

And since $x_1^* \in \mathcal{X}_1 \setminus S_X^1$, the set $X_1^* = \{x' \in \mathcal{X} \setminus S_X : x_1' = x_1^*\}$ is non-empty. Denote $L_X = \{x : (x, y) \in L\}$. Consider the following procedure:

```
repeat
    $L = \emptyset$
    Query some $x \in X_1^*$
    Labeler returns $y$:

        $y = \operatorname{argmax}_y \mathbb{1}(y \neq\perp) + \mathrm{Cost}(V[L \cup \{(x, y)\}], S_X \cup L_X \cup \{x\})$

    $X_1^* \leftarrow X_1^* \setminus \{x\}$
    $L \leftarrow L \cup \{(x, y)\}$
until $y_1 \neq\perp$ or $X_1^* = \emptyset$
```

Denote by $\hat{y}_1$ the value of y_1 at the end of the procedure, let $|L| = m$ and, in order, interaction history L is such that $L = \{(x^1, y^1), ..., (x^m, y^m)\}$. Let $L^i = \{(x_i, y_i) : (x, y) \in L, y_i \neq\perp\}$ index the binary labeled data for the ith task.

$$\begin{aligned}
\mathrm{Cost}(V, S_X) &\leq \mathbb{1}(y^1 \neq\perp) + \mathrm{Cost}(V[(x^1, y^1)], S_X \cup \{x^1\}) \quad \text{(since } x^1 \in X_1^* \subseteq \mathcal{X} \setminus S_X\text{)}\\
&= \mathrm{Cost}(V[(x^1, y^1)], S_X \cup \{x^1\}) \quad \text{(since } y_1^1 =\perp\text{)}\\
&\leq ...\\
&\text{(unrolling according to } L\text{, which is possible as } \mathrm{Cost}(V, S_X) \geq 1 \Rightarrow \mathrm{Cost}(V[L], S_X \cup L_X) \geq 1\text{)}\\
&\leq \mathbb{1}(y^m \neq\perp) + \mathrm{Cost}(V[L], S_X \cup L_X)\\
&\leq \mathbb{1}(\hat{y}_1 \neq\perp) + \mathrm{Cost}(V[L], S_X \cup L_X)\\
&\qquad (\mathbb{1}(\forall i, y_i^m \neq\perp) \leq \mathbb{1}(\hat{y}_1 \neq\perp) \text{ since } y_1^m = \hat{y}_1)\\
&= \mathbb{1}(\hat{y}_1 \neq\perp) + \mathrm{Cost}(\times_{i\in[n]} V_i[L^i], S_X \cup L_X) \quad (V \text{ is a Cartesian product})\\
&\leq \mathbb{1}(\hat{y}_1 \neq\perp) + \sum_{i=1}^n \mathrm{Cost}(V_i[L^i], (S_X \cup L_X)^i)\\
&\qquad \text{(using induction hypothesis as } |L_X| \geq 1\text{)}\\
&= \mathbb{1}(\hat{y}_1 \neq\perp) + \mathrm{Cost}(V_1[(x_1^*, \hat{y}_1)], S_X^1 \cup \{x_1^*\}) + \sum_{i=2}^n \mathrm{Cost}(V_i[L^i], (S_X \cup L_X)^i)\\
&\qquad (\diamond)\\
&\leq \mathrm{Cost}(V_1, S_X^1) + \sum_{i=2}^n \mathrm{Cost}(V_i[L^i], (S_X \cup L_X)^i) \quad \text{(by definition of } x_1^*\text{)}\\
&\leq \mathrm{Cost}(V_1, S_X^1) + \sum_{i=2}^n \mathrm{Cost}(V_i, S_X^i) \quad (\diamond\diamond)
\end{aligned}$$

($\diamond$): For the fourth step, there are two cases:

- If upon exit, $X_1^* = \emptyset$:
  Then using the definition of S_X^1, since $\nexists x \in \mathcal{X} \setminus (S_X \cup L_X)$ with $x_1 = x_1^*$, we have that $(S_X \cup L_X)^1 = S_X^1 \cup \{x_1^*\}$.
  Therefore, $\mathrm{Cost}(V_1[L^1], (S_X \cup L_X)^1) = \mathrm{Cost}(V_1[(x_1^*, \hat{y}_1)], S_X^1 \cup \{x_1^*\})$.
- Otherwise, upon exit, $X_1^* \neq \emptyset$. Then, we must have that $\hat{y} \neq \perp$:
  So $\exists x \in \mathcal{X} \setminus (S_X \cup L_X)$ with $x_i = x_i^*$.
  Therefore, $(S_X \cup L_X)^1 = S_X^1$, hence $\mathrm{Cost}(V_1[L^1], (S_X \cup L_X)^1) = \mathrm{Cost}(V_1[(x_1^*, \hat{y}_1)], S_X^1)$.
  From Lemma 19, we have that $\mathrm{Cost}(V_1[(x_1^*, \hat{y}_1)], S_X^1) = \mathrm{Cost}(V_1[(x_1^*, \hat{y}_1)], S_X^1 \cup \{x_1^*\})$.

($\diamond\diamond$): For the last step, consider each task i for $i \in \{2, \dots, n\}$:
Define:

- $L_X^{i1} = \left\{x' : \exists (x,y) \in L, x_i = x', y_i \neq \perp \wedge x' \in (S_X \cup L_X)^i\right\}$
- $L_X^{i2} = \left\{x' : \forall (x,y) \in L, x_i = x', y_i = \perp \wedge x' \in (S_X \cup L_X)^i\right\}$
- $L_X^{i3} = \left\{x' : \exists (x,y) \in L, x_i = x', y_i \neq \perp \wedge x' \notin (S_X \cup L_X)^i\right\}$
- $L_X^{i4} = \left\{x' : \forall (x,y) \in L, x_i = x', y_i = \perp \wedge x' \notin (S_X \cup L_X)^i\right\}$

With these definitions, we have $(S_X \cup L_X)^i = S_X^i \cup L_X^{i1} \cup L_X^{i2}$. The binary labeled examples comprise of $L_X^i = L_X^{i1} \cup L_X^{i3}$.
We have that:

$$\begin{aligned}
\mathrm{Cost}(V_i[L^i], (S_X \cup L_X)^i) &= \mathrm{Cost}(V_i[L^i], S_X^i \cup L_X^{i1} \cup L_X^{i2}) \\
&= \mathrm{Cost}(V_i[L^i], S_X^i \cup L_X^{i1} \cup L_X^{i2} \cup L_X^{i3}) \\
&\qquad \text{(using Lemma 19 on } L_X^{i3}) \\
&= \mathrm{Cost}(V_i[L^i \cup \left\{(x, \perp) : x \in L_X^{i2}\right\}], S_X^i \cup L_X^{i1} \cup L_X^{i2} \cup L_X^{i3}) \\
&\leq \mathrm{Cost}(V_i, S_X^i) \\
&\qquad \text{(iteratively applying Lemma 18 on } L_X^{i1} \cup L_X^{i2} \cup L_X^{i3})
\end{aligned}$$

□

2.12.3 Lower Bound

Label Cost Function: From this point onwards, we assume that the label cost is (the more generous) c_{one}.

2.12.3.1 Negative Results

Lower Bound when there is Identifiability:

The following example leverages the fact that structure in the multi-task hypothesis class constrains the target hypotheses across all n tasks. And so, abstentions can lead to the multi-task setting requiring fewer samples than even the single-task setting with the highest sample complexity.

Proposition 22. *There exists a non-Cartesian product version space V and query response S such that* $\mathrm{Cost}(V_i, S_X^i) \geq 0$ *for all i, but:*

$$\mathrm{Cost}(V, S_X) < \max_{i \in [n]} \mathrm{Cost}(V_i, S_X^i)$$

Proof. **Hypothesis Class:** Define all zero-classifier, $h_0(x) = 0$ for all x. Let $h_i = \mathbb{1}(x \in [i, i+1))$ for $i \in [n]$ be the ith interval.

Let g_1, g_2, g_3 be three distinct threshold functions, $g_1 = \mathbb{1}(x \geq 1/4), g_2 = \mathbb{1}(x \geq 1/2), g_3 = \mathbb{1}(x \geq 3/4)$ for $x \in [0, 1]$.

Set $\mathcal{H}$ to be $\left\{(h_0, g_1), (h_0, g_2), \left\{(h_i, g_3)\right\}_{i=1}^{n}\right\}$.

Data: Define $\mathcal{X} = \left\{[x_{11}, 0], ..., [x_{1n}, 0], [0, x_{21}], [0, x_{22}]\right\}$ where $x_{1i} = i + 1/2$ for $i \in [n]$ and $x_{21} = 1/3$, $x_{22} = 2/3$. By construction, $g_1(x_{21}) \neq g_2(x_{21})$ and $g_2(x_{22}) \neq g_3(x_{22})$.

Define $S = \left\{([0, x_{21}], [\perp, \perp])\right\}$. $S_X = \left\{[0, x_{21}]\right\}$, $S_X^1 = \{\}$, $S_X^2 = \{x_{21}\}$.

We have $V = \mathcal{H}[S] = \mathcal{H}$. $V_1 = \mathcal{H}_1 = \{h_0, h_1, h_2, h_3, ..., h_n\}$ and $V_2 = \mathcal{H}_2 = \{g_1, g_2, g_3\}$.

$g_1((\mathcal{X} \setminus S_X)_2) = g_2((\mathcal{X} \setminus S_X)_2) \Rightarrow (h_0, g_1), (h_0, g_2) \notin E(V, S_X)$.

We have $E(V, S_X) = \left\{(h_i, g_3)\right\}_{i=1}^{n}$, because for any $i \neq j$, (h_i, g_3) and (h_j, g_3) differ on $[x_{1j}, 0]$.

From this, we get that $\mathrm{Cost}(V, S_X) = n - 1$. Querying any point $[x_{1i}, 0]$ at any time removes only one model from the E-VS. Since the E-VS is of size n, $n - 1$ binary labeled examples are needed to reduce the E-VS size to at most 1.

On the other hand, we have that for $\mathrm{Cost}(V_1, S_X^1)$ with $|V_1| = n+1$ and $S_X^1 = \emptyset$, $\mathrm{Cost}(V_1, S_X^1) = n > \mathrm{Cost}(V, S_X)$.

□

Lower Bound when there is no Identifiability even with Cartesian product assumption:

Proposition 23. *There exists a Cartesian product version space V and query response S with* $\mathrm{Cost}(V, S_X) < 0$ *such that:*

$$\mathrm{Cost}(V, S_X) < \max_{i \in [n]} \mathrm{Cost}(V_i, S_X^i)$$

Proof. Let $\mathcal{H} = \{h_{11}, h_{12}\} \times \{h_{21}, h_{22}\}$, where $h_{11} = \mathbb{1}(x \geq 0), h_{12} = \mathbb{1}(x \geq 1)$ are intervals, and $h_{21} = \mathbb{1}(x \geq 0), h_{22} = \mathbb{1}(x \geq 1)$ are intervals.

$\mathcal{X} = \left\{[x_1, 0], [0, x_2]\right\}$ where $x_1 = 1/2$, $x_2 = 1/2$.

Labeling is: $S = \left\{([x_1, 0], [\perp, 1])\right\}$. $S_X = \left\{[x_1, 0]\right\}$, $S_X^1 = \{x_1\}$, $S_X^2 = \{0\}$.

So $V = \mathcal{H}[S] = \mathcal{H}$. $V_1 = \{h_{11}, h_{12}\}$ and $V_2 = \{h_{21}, h_{22}\}$.

Under S, we observe that $E(V, S_X) = \emptyset$, since (h_{11}, h) and (h_{12}, h) for $h \in V_2 = \{h_{21}, h_{22}\}$, predict the same on $\left\{[0, x_2]\right\} = \mathcal{X} \setminus S_X$. Hence, $\mathrm{Cost}(V, S_X) = -\infty$.

However, $\mathrm{Cost}(V_2, S_X^2) = \mathrm{Cost}(\{h_{21}, h_{22}\}, \{0\}) = 1 > \mathrm{Cost}(V, S_X)$.

□

Remark 10. *To prove the lower bound, need to impose both identifiability* $\mathrm{Cost}(V, S_X) \geq 0$ **and* Cartesian product condition.*

2.12.3.2 Positive Results

Theorem 8. *For all* $V = \times_{i\in[n]} V_i$ *and* $S_X \subseteq \mathcal{X}$*, if* $\mathrm{Cost}(V, S_X) \geq 0$*, then:*

$$\mathrm{Cost}(V, S_X) \geq \max_{i\in[n]} \mathrm{Cost}(V_i, S_X^i)$$

Proof. We prove this by induction on the size of S_X.

Base Case: When $S_X = \mathcal{X} \Rightarrow S_X^i = \mathcal{X}_i$, so for all i, $\mathrm{Cost}(V_i, S_X^i) \leq 0 \leq \mathrm{Cost}(V, S_X)$.

Induction Step: Suppose the following holds for $|S_X| = |\mathcal{X}|, ..., j+1$.

Now let $|S_X| = j$. Note that this implies $S_X \subset X$.

First, consider the case when $\mathrm{Cost}(V, S_X) = 0$. We have that $|E(V, S_X)| = 1$. And so, using Lemma 17, for all i, $|E(V_i, S_X^i)| = 1$. Thus, $\mathrm{Cost}(V_i, S_X^i) = 0$ for all i.

Now, we consider the case when $\mathrm{Cost}(V, S_X) \geq 1$.

Let $k = \mathrm{argmax}_{i\in[n]} \mathrm{Cost}(V_i, S_X^i)$. It suffices to verify the statement when $\mathrm{Cost}(V_k, S_X^k) \geq 1$.

Since $\mathcal{X} \setminus S_X$ is non-empty due to $S_X \subset \mathcal{X}$, define:

$$x^{min} = \operatorname*{argmin}_{x\in\mathcal{X}\setminus S_X} \max_{y'\in\mathcal{Y}} \mathbb{1}(y' \neq \perp) + \mathrm{Cost}(V_x^{y'}, S_X \cup \{x\})$$

We have that $\mathcal{X}_k \setminus S_X^k = (\mathcal{X} \setminus S_X)_k = \left\{x' \in \mathcal{X}_k : \exists x \in \mathcal{X} \setminus S_X, x_k = x'\right\}$, and so $x_k^{min} \in \mathcal{X}_k \setminus S_X^k$ since $x^{min} \in \mathcal{X} \setminus S_X$.

Since $\mathrm{Cost}(V_k, S_X^k) \geq 1$, we know there exists $\tilde{y}_k$ such that:

$$\mathrm{Cost}(V_k, S_X^k) \leq \mathbb{1}(\tilde{y}_k \neq \perp) + \mathrm{Cost}(V_k[(x_k^{min}, \tilde{y}_k)], S_X^k \cup \left\{x_k^{min}\right\}).$$

Note in particular that $E(V_k[(x_k^{min}, \tilde{y}_k)], S_X^k \cup \left\{x_k^{min}\right\}) \neq \emptyset$, as otherwise $\mathrm{Cost}(V_k, S_X^k) \leq -\infty$ which would contradict our assumption that $\mathrm{Cost}(V_k, S_X^k) \geq 1$.

$$\begin{aligned}
\mathrm{Cost}(V, S_X) &= \max_{y'\in\mathcal{Y}} \mathbb{1}(y' \neq \perp) + \mathrm{Cost}(V_{x^{min}}^{y'}, S_X \cup \left\{x^{min}\right\}) \\
&\geq \mathbb{1}(y \neq \perp) + \mathrm{Cost}(\times_{i\in[n]}(V_i)_{x_i^{min}}^{y_i}, S_X \cup \left\{x^{min}\right\}) \\
&\qquad \text{(setting } y' = y \text{ as constructed in Lemma 20 and using that } V_{x^{min}}^{y} = \times_{i\in[n]}(V_i)_{x_i^{min}}^{y_i}\text{)} \\
&\geq \mathbb{1}(y \neq \perp) + \max_{i\in[n]} \mathrm{Cost}((V_i)_{x_i^{min}}^{y_i}, (S_X \cup \left\{x_i^{min}\right\})^i) \\
&\qquad \text{(using induction hypothesis since } x^{min} \notin S_X \text{, so } |S_X \cup \left\{x^{min}\right\}| = j+1\text{)} \\
&\geq \mathbb{1}(\tilde{y}_k \neq \perp) + \mathrm{Cost}((V_k)_{x_k^{min}}^{\tilde{y}_k}, (S_X \cup \left\{x^{min}\right\})^k) \\
&\qquad (\mathbb{1}(y \neq \perp) \geq \mathbb{1}(y_k \neq \perp) = \mathbb{1}(\tilde{y}_k \neq \perp) \text{ as } y_k = \tilde{y}_k \text{ by construction)} \\
&\geq \mathbb{1}(\tilde{y}_k \neq \perp) + \mathrm{Cost}((V_k)_{x_k^{min}}^{\tilde{y}_k}, S_X^k \cup \left\{x_k^{min}\right\}) \\
&\qquad \text{(note that } x_k^{min} \in (\mathcal{X} \setminus S_X)_k \text{, so } x_k^{min} \in \mathcal{X}_k \setminus S_X^k \text{ and } \diamond\text{)} \\
&\geq \mathrm{Cost}(V_k, S_X^k)
\end{aligned}$$

($\diamond$): Either we have $(S_X \cup \left\{x^{min}\right\})^k = S_X^k \cup \left\{x_k^{min}\right\}$ or $(S_X \cup \left\{x^{min}\right\})^k = S_X^k$. The former case yields equality and the statement holds.

For the latter case, we can use Lemma 18 (for $\tilde{y}_k = \perp$) or Lemma 19 (for $\tilde{y}_k \neq \perp$) to get that: $\mathrm{Cost}((V_k)^{\tilde{y}_k}_{x_k^{min}}, (S_X \cup \left\{x^{min}\right\})^k) = \mathrm{Cost}((V_k)^{\tilde{y}_k}_{x_k^{min}}, S^k_X) \geq \mathrm{Cost}((V_k)^{\tilde{y}_k}_{x_k^{min}}, S^k_X \cup \left\{x_k^{min}\right\})$.

□

Lemma 20. *Suppose* $C(V, S_X) \geq 0$ *and* $x^{min} = \mathrm{argmin}_{x \in \mathcal{X} \setminus S_X} \max_{y \in \mathcal{Y}} \mathbb{1}(y \neq \perp) + \mathrm{Cost}(V^y_x, S_X \cup \{x\})$. *If there* $\tilde{y}_k$ *such that* $\mathrm{Cost}(V_k, S^k_X) \leq \mathbb{1}(\tilde{y}_k \neq \perp) + \mathrm{Cost}(V_k[(x_k^{min}, \tilde{y}_k)], S^k_X \cup \left\{x_k^{min}\right\})$ *for* $\mathrm{Cost}(V_k, S^k_X) \geq 0$*, then there exists* y *such that its kth coordinate* $y_k = \tilde{y}_k$ *such that:*

$$\mathrm{Cost}(V[(x^{min}, y)], S_X \cup \left\{x^{min}\right\}) \geq 0$$

Proof. We explicitly construct some y such that $y_k = \tilde{y}_k$ and the above holds:

- Firstly, $\mathrm{Cost}(V, S_X) \geq 0$, which implies there exists $h \in E(V, S_X)$.
  $h \in V$ implies that $\forall i, h_i \in V_i$.
  Also, $\mathrm{Cost}(V_k[(x_k^{min}, \tilde{y}_k)], S^k_X \cup \left\{x_k^{min}\right\}) \geq \mathrm{Cost}(V_k, S^k_X) - 1 \geq 0$. This implies that there exists some $\tilde{h}_k \in E(V_k[(x_k^{min}, \tilde{y}_k)], S^k_X \cup \left\{x_k^{min}\right\})$.
- We claim that $y = (h_1(x_1^{min}), ..., \tilde{h}_k(x_k^{min}), ..., h_n(x_n^{min}))$ satisfies the condition.
  To show this, define $\tilde{h} = (h_1, ..., \tilde{h}_k, ..., h_n)$.
  Firstly, since $h_i \in V_i$ (for $i \neq k, i \in [n]$) and $\tilde{h}_k \in V_k$, we have that $\tilde{h} \in \times_{i \in [n]} V_i = V$. Also, $\tilde{h}(x^{min}) = y$. Therefore, $\tilde{h} \in V^y_{x^{min}}$.
- We will show that $\tilde{h} \in E(V^y_{x^{min}}, S_X \cup \left\{x^{min}\right\})$, which proves the result.
  From Lemma 5, We have that:

  $$\tilde{h}_k \in E(V_k[(x_k^{min}, \tilde{y}_k)], S^k_X \cup \left\{x_k^{min}\right\}) \subseteq E(V_k[(x_k^{min}, \tilde{y}_k)], (S_X \cup \left\{x^{min}\right\})^k)$$

  since $S^k \cup \left\{x_k^{min}\right\} \supseteq (S_X \cup \left\{x^{min}\right\})^k$.
  For all $i \neq k$, we have:

  $$h \in E(V, S_X) \Rightarrow h_i \in E(V_i, S^i_X) \Rightarrow h_i \in E(V_i[(x_i^{min}, y_i)], S^i_X \cup \left\{x_i^{min}\right\})$$

  since for all $h' \in V_i \setminus \{h_i\}$ with $h'(x_i^{min}) = y_i = h_i(x_i^{min})$, h' must be such that $h'(\mathcal{X} \setminus (S^i_X \cup \left\{x_i^{min}\right\})) \neq h_i(\mathcal{X} \setminus (S^i_X \cup \left\{x_i^{min}\right\}))$. Since this holds for all $h' \in V_i[(x_i^{min}, y_i)] \setminus \{h_i\}$, we have $h_i \in E(V_i[(x_i^{min}, y_i)], S^i_X \cup \left\{x_i^{min}\right\})$.
  From Lemma 5, We have that:

  $$h_i \in E(V_i[(x_i^{min}, y_i)], S^i_X \cup \left\{x_i^{min}\right\}) \subseteq E(V_i[(x_i^{min}, y_i)], (S_X \cup \left\{x^{min}\right\})^i)$$

  since $S^i_X \cup \left\{x_i^{min}\right\} \supseteq (S_X \cup \left\{x^{min}\right\})^i$.
  Hence,

  $$\tilde{h} \in \times_{i=1}^k E(V[(x_i^{min}, y_i)], (S_X \cup \left\{x^{min}\right\})^i)) \Rightarrow \tilde{h} \in E(V[(x^{min}, y)], S_X \cup \left\{x^{min}\right\}))$$

  since from Lemma 17, we have that:

  $$E(V[(x^{min}, y)], S_X \cup \left\{x^{min}\right\})) = \times_{i=1}^k E(V[(x_i^{min}, y_i)], (S_X \cup \left\{x^{min}\right\})^i))$$

□

Remark 11. *As* $\mathrm{Cost}(\times_{i \in [n]} (V_i)^{y_i}_{x_i^{min}}, S_X \cup \left\{x^{min}\right\}) \geq 0$*, the precondition for induction hypothesis holds.*

2.12.4 Multi-task Active Learning without Abstention

We also investigate the related multi-task, minimax active learning setting without abstention, which may be of independent interest. To our knowledge, this is also an open problem. Our goal is again to relate the multi-task complexity to the single-task complexity. Since abstention is the cause of several of the negative examples above, one can prove more general upper bounds when labels have to be given.

2.12.4.1 Game Setup

Without abstention, the state may now be tracked simply with VS (instead of E-VS). The analogous game value may be defined as follows:

$$\mathrm{Cost}(V, S_X) = \begin{cases} -\infty & |V| = 0 \\ 0, & |V| = 1 \\ \min_{x \in \mathcal{X} \setminus S_X} \max_{y \in \{-1,+1\}} \left(1 + \mathrm{Cost}(V_x^y, S_X \cup \{x\})\right), & |V| \geq 2 \end{cases}$$

2.12.4.2 Lemmas Used

Lemma 21. *For any S_X, $|V| \geq 1 \Leftrightarrow \mathrm{Cost}(V, S_X) \geq 0$.*

Proof. **Base Case:** We prove this by induction on $|S_X|$. If $S_X = \mathcal{X}$, then $|V| \geq 1 \Rightarrow |V| = 1 \Rightarrow \mathrm{Cost}(V, S_X) = 0$.

Induction Step: Suppose this is true for $|S_X| = |\mathcal{X}|, ..., j + 1$. Now $|S_X| = j$. Let $h \in V$. If $|V| = 1$, then the result holds.

Otherwise, $|V| \geq 2$. We will show that $|V| \geq 2 \Rightarrow \mathrm{Cost}(V, S_X) \geq 1$:

$$\begin{aligned} \mathrm{Cost}(V, S_X) &= \min_{x \in \mathcal{X} \setminus S_X} \max_{y \in \{+1,-1\}} 1 + \mathrm{Cost}(V_x^y, S_X \cup \{x\}) \\ &\geq 1 + \mathrm{Cost}(V[(x^*, h(x^*)], S_X \cup \{x^*\})) \\ &\text{(for } x^* = \mathrm{argmin}_{x \in \mathcal{X} \setminus S_X} \max_{y \in \{+1,-1\}} 1 + \mathrm{Cost}(V_x^y, S_X \cup \{x\})) \\ &\geq 1 \end{aligned}$$

The last step that $\mathrm{Cost}(V[(x^*, h(x^*)], S_X \cup \{x^*\})) \geq 0$ follows from induction hypothesis, whose precondition is satisfied because $h \in V \Rightarrow h \in V[(x^*, h(x^*)]$.

($\Leftarrow$) $|V| = 0 \Rightarrow \mathrm{Cost}(V, S_X) = -\infty < 0$, hence $\mathrm{Cost}(V, S_X) \geq 0 \Rightarrow |V| \geq 1$. □

Corollary 5. *We have that:*

1. $\mathrm{Cost}(V, S_X) = -\infty \Leftrightarrow |V| = 0$
2. $\mathrm{Cost}(V, S_X) = 0 \Leftrightarrow |V| = 1$

Proof. 1. ($\Rightarrow$): Follows from that $\mathrm{Cost}(V, S_X) < 0 \Rightarrow |V| < 1 \Rightarrow |V| = 0$.
($\Leftarrow$): Follows from the base case definition of Cost.

2. ($\Rightarrow$): From the above, we have that $|V| \geq 2 \Rightarrow \mathrm{Cost}(V, S_X) \geq 1$. And so, $\mathrm{Cost}(V, S_X) \leq 0 \Rightarrow |V| \leq 1$.
   The result follows since $\mathrm{Cost}(V, S_X) = 0 \neq -\infty \Rightarrow |V| \neq 0 \Rightarrow |V| = 1$.
   ($\Leftarrow$): Follows from the base case definition of Cost.

□

Lemma 22. *For $V' \subseteq V$ and any $S_X \subseteq \mathcal{X}$:*

$$\mathrm{Cost}(V, S_X) \geq \mathrm{Cost}(V', S_X)$$

Proof. We will prove this statement by induction on the size of S_X.

Base Case: $S_X = \mathcal{X}$. This means $\mathrm{Cost}(V, S_X), \mathrm{Cost}(V', S_X)$ are at the base-case. If $|V'| = 1 \Rightarrow |V| = 1$, and the statement holds. If $|V'| = 0$, the statement holds since RHS is equal to $-\infty$.

Induction Step: Suppose the statement holds for $|S_X| = |\mathcal{X}|, ..., j+1$ and any $V' \subseteq V$. Consider some S_X such that $|S_X| = j$.

(a) First, we examine what happens if $|V| \leq 1$.

(i) if $|V| = 0 \Rightarrow |V'| = 0$, then $\mathrm{Cost}(V, S_X) = -\infty = \mathrm{Cost}(V', S_X)$

(ii) if $|V| = 1 \Rightarrow |V'| \leq 1$, so $\mathrm{Cost}(V, S_X) = 0 \geq \mathrm{Cost}(V', S_X)$.

(b) If $|V| \geq 2$ and $|V'| \leq 1$, then since $|V| \geq 1$, we have $\mathrm{Cost}(V, S_X) \geq 0 \geq \mathrm{Cost}(V', S_X)$ using Lemma 21.

(c) The remaining case is when $|V| \geq 2$ and $|V'| \geq 2$.

We have that:

$$\begin{aligned}
\mathrm{Cost}(V, S_X) &= \min_{x \in \mathcal{X} \setminus S_X} \max_{y \in \{+1,-1\}} 1 + \mathrm{Cost}(V_x^y, S_X \cup \{x\}) \qquad \text{(since } |V| \geq 2\text{, we can unroll)} \\
&\geq \min_{x \in \mathcal{X} \setminus S_X} \max_{y \in \{+1,-1\}} 1 + \mathrm{Cost}((V')_x^y, S_X \cup \{x\}) \\
&\qquad \text{(for all } x, y,\ V' \subseteq V \Rightarrow V'[(x,y)] \subseteq V[(x,y)]\text{, so we may apply induction hypothesis)} \\
&= \mathrm{Cost}(V', S_X)
\end{aligned}$$

□

Lemma 23. *For any data point (x_1, y_1) for $x_1 \notin S_X$ and $y_1 \in \{+1, -1\}$:*

$$\mathrm{Cost}(V[(x_1, y_1)], S_X \cup \{x_1\}) \leq \mathrm{Cost}(V, S_X)$$

Proof. **Base Case:**

We first handle the case when $|V[(x_1, y_1)]| \leq 1$:

If $|V[(x_1, y_1)]| = 0$, then the result holds.

If $|V[(x_1, y_1)]| = 1 \Rightarrow |V| \geq 1$, and the result holds from Lemma 21.

This covers the base case when $S_X = \mathcal{X}$.

Induction Step: Suppose the statement holds for when $|S_X| = |\mathcal{X}|, ..., j+1$. Let $|S_X| = j$.

It suffices to examine the case that $|V[(x_1, y_1)]| \geq 2$, which implies that $|V| \geq 2$.

Define

$$x' \in \operatorname*{argmin}_{x \in \mathcal{X} \setminus S_X} \max_y 1 + \mathrm{Cost}(V[(x', y)], S \cup \{x'\});$$

with this definition,

$$\text{Cost}(V, S_X) = \max_y 1 + \text{Cost}(V[(x', y)], S_X \cup \left\{x'\right\})$$

If $x' = x_1$, then the result follows.

If $x' \neq x_1$, then $x' \in \mathcal{X} \setminus S \cup \{x_1\}$, and we can write:

$$\begin{aligned}
\text{Cost}(V[(x_1, y_1)], S_X \cup \{x_1\}) &\leq \max_y 1 + \text{Cost}(V[(x_1, y_1), (x', y)], S_X \cup \left\{x_1, x'\right\}) \\
&\qquad \text{(as } |V[(x_1, y_1)| \geq 2 \text{ so we can unroll with } x' \in \mathcal{X} \setminus S_X \cup \{x_1\}\text{)} \\
&\leq \max_y 1 + \text{Cost}(V[(x', y)], S_X \cup \left\{x'\right\}) \\
&\qquad \text{(using induction hypothesis)} \\
&= \text{Cost}(V, S_X)
\end{aligned}$$

□

Lemma 24. *For $x \in \mathcal{X} \setminus S_X$ and some $y \in \{+1, -1\}$:*

$$\text{Cost}(V[(x, y)], S_X) = \text{Cost}(V[(x, y)], S_X \cup \{x\})$$

Proof. We show this by induction on size of S_X.

Base Case: Firstly, the version space are the same, $V[(x, y)]$.

So LHS is equal to RHS when $|V[(x, y)]| \leq 1$ in the base case. This covers the case when $S_X = \mathcal{X}$.

Induction Step: Suppose the statement holds for when $|S_X| = |\mathcal{X}|, ..., j + 1$. Let $|S_X| = j$.

It suffices to consider when $|V[(x, y)]| \geq 2$. We may write:

$$\text{Cost}(V, S_X) = \min_{x' \in \mathcal{X} \setminus S_X} \max_{y' \in \{+1, -1\}} 1 + \text{Cost}(V[(x', y')]], S_X \cup \left\{x'\right\})$$

Define $x^* \in \text{argmin}_{x' \in \mathcal{X} \setminus S_X} \max_{y' \in \{+1,-1\}} 1 + \text{Cost}(V[(x', y')]], S_X \cup \{x'\})$.

We will show that $x^* \neq x$.

In fact, for any $x' \in \mathcal{X} \setminus S_X$, $x' \neq x^*$ (which exists because $\{x\} \subset \mathcal{X} \setminus S_X$) we have:

$$\begin{aligned}
&\max_{y' \in \{+1, -1\}} 1 + \text{Cost}(V_x^y[(x, y')]], S_X \cup \{x\}) \\
&= \max(1 + \text{Cost}(V_x^y, S_X \cup \{x\}), 1 + \text{Cost}(\emptyset, S_X \cup \{x\})) \\
&= 1 + \text{Cost}(V_x^y, S_X \cup \{x\}) && \text{(maximized at when } y' = y\text{)} \\
&\geq \max_{y' \in \{+1, -1\}} 1 + \text{Cost}(V_x^y[(x', y')]], S_X \cup \left\{x, x'\right\}) && \text{(using Lemma 23)} \\
&= \max_{y' \in \{+1, -1\}} 1 + \text{Cost}(V_x^y[(x', y')]], S_X \cup \left\{x'\right\}) \\
&\qquad \text{(using induction hypothesis since } |S_X \cup \{x'\}\,| = j + 1\text{)}
\end{aligned}$$

And so,

$$\begin{aligned}
\mathrm{Cost}(V[(x,y)],S_X) &= \min_{x'\in\mathcal{X}\setminus S_X}\ \max_{y'\in\{+1,-1\}} 1+\mathrm{Cost}(V_x^y[(x',y')]],S_X\cup\left\{x'\right\}) \\
&= \min_{x'\in\mathcal{X}\setminus(S_X\cup\{x\})}\ \max_{y'\in\{+1,-1\}} 1+\mathrm{Cost}(V_{x'}^{y'}[(x,y)]],S_X\cup\left\{x'\right\}) \\
&\hspace{20em}\text{(since } x^*\neq x\text{)} \\
&= \min_{x'\in\mathcal{X}\setminus(S_X\cup\{x\})}\ \max_{y'\in\{+1,-1\}} 1+\mathrm{Cost}(V_{x'}^{y'}[(x,y)]],(S_X\cup\left\{x'\right\})\cup\{x\}) \\
&\hspace{8em}\text{(using induction hypothesis since } |S_X\cup\{x'\}\,| = j+1\text{)} \\
&= \min_{x'\in\mathcal{X}\setminus(S_X\cup\{x\})}\ \max_{y'\in\{+1,-1\}} 1+\mathrm{Cost}(V_x^y[(x',y')]],(S_X\cup\{x\})\cup\left\{x'\right\}) \\
&\hspace{20em}\text{(rearranging)} \\
&= \mathrm{Cost}(V[(x,y)],S_X\cup\{x\})
\end{aligned}$$

□

2.12.4.3 Upper Bound

Theorem 9. *For all* $V\subseteq\mathcal{H}$ *and* $S_X\subseteq\mathcal{X}$*:*

$$\mathrm{Cost}(V,S_X)\leq\sum_{i=1}^{n}\mathrm{Cost}(V_i,S_X^i)$$

Proof. We will proceed by induction on the size of S_X:

Base Case: When $S_X=\mathcal{X}$. In this case, $S_X^i=\mathcal{X}_i$. So all Cost's are at the base-case.

It suffices to check that if $\mathrm{Cost}(V,S_X)=0\Rightarrow\forall i,\mathrm{Cost}(V_i,S_X^i)=0$.

This follows because $\mathrm{Cost}(V,S_X)=0\Leftrightarrow|V|=1$. By definition of V_i, $|V_i|=1$. And so, $\mathrm{Cost}(V,S_X)=0=\sum_{i=1}^n\mathrm{Cost}(V_i,S_X^i)$.

Induction Step:

Suppose the following holds for $S_X\subset X$ for $|S_X|=|\mathcal{X}|,...,j+1$. Now let $|S_X|=j$ (with $S_X\subset\mathcal{X}$).

We consider three cases:

- $\exists i,V_i=\emptyset$
- $\forall i,|V_i|\geq1$ and $\forall i,|V_i|=1$
- $\forall i,|V_i|\geq1$ and $\exists i,|V_i|\geq2$

1. **If there is** i **such that** $\mathrm{Cost}(V_i,S_X^i)=-\infty$**.**
   Then $V_i=\emptyset\Rightarrow V=\emptyset$, and therefore, $\mathrm{Cost}(V,S_X)=-\infty$.
2. **For all** i**,** $\mathrm{Cost}(V_i,S_X^i)=0$**.**
   This means that for all i, $|V_i|=1$. And we wish to show that $|V|\leq1$, which would imply that $\mathrm{Cost}(V,S_X)\leq0=\sum_{i=1}^n\mathrm{Cost}(V_i,S_X^i)$.
   Suppose not, there exists $h,h'\in V$. Then, $h\neq h'\Rightarrow\exists i$ such that $h_i\neq h_i'\Rightarrow h_i,h_i'\in V_i\Rightarrow|V_i|\geq2$, which is a contradiction.

3. **Exists i such that $\text{Cost}(V_i, S_X^i) \geq 1$, and $\text{Cost}(V_j, S_X^j) \geq 0$ for all j.**
   Assume WLOG $i = 1$. Note that if $|V| \leq 1$, then $\text{Cost}(V, S_X) \leq 0 \leq \sum_{i=1}^{n} \text{Cost}(V_i, S_X^i)$. And so, we will consider the case when $|V| \geq 2$ and $|V_1| \geq 2$.
   Define
   $$x_1^* \in \operatorname*{argmin}_{x \in \mathcal{X}_1 \setminus S_X^1} \max_{y \in \{+1,-1\}} 1 + \text{Cost}(V_1[(x_1^*, y)], S_X^1 \cup \{x_1^*\})$$
   we may express:
   $$\text{Cost}(V_1, S_X^1) = \max_{y \in \{+1,-1\}} 1 + \text{Cost}(V_1[(x_1^*, y)], S_X^1 \cup \{x_1^*\}) \tag{2.9}$$
   Moreover, we have that $\exists x^* \in \mathcal{X} \setminus S_X$ with the first coordinate equal to x_1^*. And so,
   $$\text{Cost}(V, S_X) \leq \max_{y \in \{+1,-1\}^n} 1 + \text{Cost}(V[(x^*, y)], S_X \cup \{x^*\}) = 1 + \text{Cost}(V[(x^*, y')], S_X \cup \{x^*\})$$
   With this,
   $$\begin{aligned}
   \text{Cost}(V, S_X) &\leq 1 + \text{Cost}(V[(x^*, y')], S_X \cup \{x^*\}) \\
   &\leq 1 + \sum_{i=1}^{n} \text{Cost}((V[(x^*, y')])_i, (S_X \cup \{x^*\})^i) \quad \text{(using induction hypothesis)} \\
   &= 1 + \text{Cost}((V[(x^*, y')])_1, (S_X \cup \{x^*\})^1) + \sum_{i=2}^{n} \text{Cost}((V[(x^*, y')])_i, (S_X \cup \{x^*\})^i) \\
   &\leq 1 + \text{Cost}(V_1[(x_1^*, y_1')], S_X^1 \cup \{x_1^*\}) + \sum_{i=2}^{n} \text{Cost}((V[(x^*, y')])_i, (S_X \cup \{x^*\})^i) \\
   &\qquad \text{(using Lemma 22 and } \diamond \text{ for task 1)} \\
   &\leq \text{Cost}(V_1, S_X^1) + \sum_{i=2}^{n} \text{Cost}((V[(x^*, y')])_i, (S_X \cup \{x^*\})^i) \\
   &\qquad \text{(using Equation 2.9)} \\
   &\leq \text{Cost}(V_1, S_X^1) + \sum_{i=2}^{n} \text{Cost}(V_i[(x_i^*, y_i')], S_X^i \cup \{x_i^*\}) \\
   &\qquad \text{(using Lemma 22 and } \diamond \text{ for tasks 2 to } n) \\
   &\leq \text{Cost}(V_1, S_X^1) + \sum_{i=2}^{n} \text{Cost}(V_i, S_X^i) \qquad \text{(using Lemma 23 for tasks 2 to } n)
   \end{aligned}$$

For any task i:

Lemma 25. *For any x, y and V,*
$$(V[(x, y)])_i \subseteq V_i[(x_i, y_i)]$$

Proof. We have that $h'_i \in (V[(x,y)])_i \Rightarrow \exists h \in V[(x,y)], h_i = h'_i$.
$h_i \in V_i[(x_i, y_i)]$, since $h \in V[(x,y)] \Rightarrow h_i \in V_i \wedge h_i(x_i) = y_i$ (from $h(x) = y$).
And so, we get that $h'_i = h_i \in V_i[(x_i, y_i)]$.

□

Using this lemma, we may apply Lemma 22 to get that:

$$\mathrm{Cost}((V[(x^*, y')])_i, (S_X \cup \{x^*\})^i) \leq \mathrm{Cost}(V_i[(x^*_i, y'_i)], (S_X \cup \{x^*\})^i)$$

We will show below that:

$$\mathrm{Cost}(V_i[(x^*_i, y'_i)], (S_X \cup \{x^*\})^i) = \mathrm{Cost}(V_i[(x^*_i, y'_i)], S^i_X \cup \{x^*_i\})$$

($\diamond$): There are two cases to consider:

- Case 1: $(S_X \cup \{x^*\})^i = S^i_X \cup \{x^*_i\}$; in this case, $\mathrm{Cost}(V_i[(x^*_i, y'_i)], (S_X \cup \{x^*\})^i) = \mathrm{Cost}(V_i[(x^*_i, y'_i)], S^i_X \cup \{x^*_i\})$ holds;
- Case 2: $(S_X \cup \{x^*\})^i = S^i_X$, in this case, $\mathrm{Cost}(V_i[(x^*_i, y'_i)], (S_X \cup \{x^*\})^i) = \mathrm{Cost}(V_i[(x^*_i, y'_i)], S^i_X) = \mathrm{Cost}(V_i[(x^*_i, y'_i)], S^i_X \cup \{x^*_i\})$, where the last equality uses Lemma 24.

□

2.12.4.4 Lower Bound

Example of non-Cartesian Product V can reverse inequality:
Proposition 24. *There exists a non-Cartesian product version space V and S_X such that:*

$$\mathrm{Cost}(V, S_X) < \max_{i \in [n]} \mathrm{Cost}(V_i, S^i_X)$$

Proof. Consider $\mathcal{H} = \{(h_1, g_1), (h_2, g_1), (h_3, g_2)\}$. h_i and g_j's are thresholds.

Let $\mathcal{X} = \{[x_{11}, x_2], [x_{12}, x_2]\}$, where x_{11} separates h_1, h_2, x_{12} separates h_2, h_3 and x_2 separates g_1, g_2.

Let $S = \emptyset$, so $S_X = S^1_X = S^2_X = \emptyset$.

$V = \mathcal{H} = \{(h_1, g_1), (h_2, g_1), (h_3, g_2)\}$, $V_1 = \{h_1, h_2, h_3\}$, $V_2 = \{g_1, g_2\}$.

Then, we have that $\mathrm{Cost}(V_1, \emptyset) = 2$ for $V_1 = \{h_1, h_2, h_3\}$. However, $\mathrm{Cost}(V, \emptyset) = 1$, since one needs to query $[x_{11}, x_2]$ only. □

Remark 12. *The observation is that x_{11} helps to distinguish between h_1 and $h_2 \in V_1$, while x_2 helps with distinguishing between g_1 and $g_2 \in V_2$, which in turn helps to distinguish between $\{h_1, h_2\}$ and $\{h_3\} \subset V_1$.*
Theorem 10. *For all $V = \times_{i \in [n]} V_i$ and $S_X \subseteq \mathcal{X}$ such that $\mathrm{Cost}(V, S_X) \geq 0$:*

$$\mathrm{Cost}(V, S_X) \geq \max_{i \in [n]} \mathrm{Cost}(V_i, S^i_X)$$

Proof. We prove this by induction on the size of S_X.

Base Case: $S_X = \mathcal{X} \Rightarrow S_X^i = \mathcal{X}_i$.

If $\mathrm{Cost}(V, \mathcal{X}) = 0$, then $|V| = 1 \Rightarrow |V_i| = 1, \forall i \Rightarrow \mathrm{Cost}(V_i, S_X^i) = 0$ for all i.

Induction Step: Suppose the following holds for $|S_X| = |\mathcal{X}|, ..., j+1$. Now let $|S_X| = j$, note that $S_X \subset X$.

We first handle the base cases.

If $\mathrm{Cost}(V, S_X) = 0$, then $V = \{h\} \Rightarrow \forall i, V_i = \{h_i\}$ (due to the Cartesian product structure of V) $\Rightarrow \mathrm{Cost}(V_i, S_X^i) = 0$.

Now, if $\mathrm{Cost}(V, S_X) \geq 1$ and if $k = \mathrm{argmax}_{i \in [n]} \mathrm{Cost}(V_i, S_X^i)$, then it suffices to verify the statement when $\mathrm{Cost}(V_k, S_X^k) \geq 1$.

Define:

$$x^{min} = \operatorname*{argmin}_{x \in \mathcal{X} \setminus S_X} \max_{y' \in \mathcal{Y}} \mathbb{1}(y' \neq \perp) + \mathrm{Cost}(V_x^{y'}, S_X \cup \{x\})$$

From definition, $\mathcal{X}_k \setminus S_X^k = (\mathcal{X} \setminus S_X)_k = \left\{x' \in \mathcal{X}_k : \exists x \in \mathcal{X} \setminus S_X, x_k = x'\right\}$. And so $x_k^{min} \in \mathcal{X}_k \setminus S_X^k$ since $x^{min} \in \mathcal{X} \setminus S_X$. Since $\mathrm{Cost}(V_k, S_X^k) \geq 1$, we know there exists $\tilde{y}_k$ such that:

$$\mathrm{Cost}(V_k, S_X^k) \leq 1 + \mathrm{Cost}(V_k[(x_k^{min}, \tilde{y}_k)], S_X^k \cup \left\{x_k^{min}\right\})$$

Note in particular that $V_k[(x_k^{min}, \tilde{y}_k)] \neq \emptyset$ as otherwise $\mathrm{Cost}(V_k, S_X^k) \leq -\infty$ (which contradicts our assumption):

$$\begin{aligned}
\mathrm{Cost}(V, S_X) &= \min_{x \in \mathcal{X} \setminus S_X} \max_{y' \in \mathcal{Y}} 1 + \mathrm{Cost}(V_x^{y'}, S_X \cup \{x\}) \quad (\mathcal{X} \setminus S_X \text{ is non-empty, since } S_X \subset \mathcal{X}) \\
&= \max_{y' \in \mathcal{Y}} 1 + \mathrm{Cost}(V_{x^{min}}^{y'}, S_X \cup \left\{x^{min}\right\}) \\
&\geq 1 + \mathrm{Cost}(\times_{i \in [n]} (V_i)_{x_i^{min}}^{y_i}, S_X \cup \left\{x^{min}\right\}) \\
&\qquad \text{(setting } y' = y \text{ as constructed below (†) and using that } V_{x^{min}}^{y} = \times_{i \in [n]} (V_i)_{x_i^{min}}^{y_i}) \\
&\geq 1 + \max_{i \in [n]} \mathrm{Cost}((V_i)_{x_i^{min}}^{y_i}, (S_X \cup \left\{x_i^{min}\right\})^i) \\
&\qquad \text{(using induction hypothesis since } x^{min} \notin S_X \text{, so } |S_X \cup \left\{x^{min}\right\}| = j+1) \\
&\geq 1 + \mathrm{Cost}((V_k)_{x_k^{min}}^{\tilde{y}_k}, (S_X \cup \left\{x^{min}\right\})^k) \qquad \text{(by construction, } y_k = \tilde{y}_k) \\
&= 1 + \mathrm{Cost}((V_k)_{x_k^{min}}^{\tilde{y}_k}, S_X^k \cup \left\{x_k^{min}\right\}) \\
&\qquad \text{(note that } x_k^{min} \in (\mathcal{X} \setminus S_X)_k \text{, so } x_k^{min} \in \mathcal{X}_k \setminus S_X^k \text{ and } \diamond) \\
&\geq \mathrm{Cost}(V_k, S_X^k)
\end{aligned}$$

(†) : **Claim:** There exists some y such that $y_k = \tilde{y}_k$ and $V_{x^{min}}^{y} \neq \emptyset$ (that is, $(V_i)_{x_i^{min}}^{y_i} \neq \emptyset$ for each i).

Firstly, $\mathrm{Cost}(V, S_X) \geq 0 \Rightarrow |V| \geq 1$. This means that there exists $h \in V$, and that $\forall i, \exists h_i \in V_i$.

Since $V_k[(x_k^{min}, \tilde{y}_k)] \neq \emptyset$, there exists some $\tilde{h}_k \in V_k[(x_k^{min}, \tilde{y}_k)] \neq \emptyset$.

We claim that $y = (h_1(x_1^{min}), ..., \tilde{h}_k(x_k^{min}), ..., h_n(x_n^{min}))$ satisfies the property.

Let $h = (h_1, ..., \tilde{h}_k, ..., h_n)$. Then we have $h \in V^y_{x^{min}}$, since:

i) $h_i \in V_i, \tilde{h}_k \in V_k$ implies $h \in \times_{i \in [n]} V_i = V$

ii) $h(x^{min}) = y$.

And so, $|V^y_{x^{min}}| \geq 1 \Rightarrow \text{Cost}(V^y_{x^{min}}, S_X \cup \left\{x^{min}\right\}) \geq 0$, which means we meet the precondition needed to use the induction hypothesis.

($\diamond$): For task k, We know that $(S_X \cup \left\{x^{min}\right\})^k$ is either S_X^k or $S_X^k \cup \left\{x_k^{min}\right\}$. In the latter case, equality holds.

In the former case, we may use Lemma 24 to get that equality also holds:

$\text{Cost}((V_k)^{\tilde{y}_k}_{x_k^{min}}, (S_X \cup \left\{x^{min}\right\})_k) = \text{Cost}((V_k)^{\tilde{y}_k}_{x_k^{min}}, S_X^k) = \text{Cost}((V_k)^{\tilde{y}_k}_{x_k^{min}}, S_X^k \cup \left\{x_k^{min}\right\})$. □

2.13 Miscellaneous

2.13.1 Data-based Game Representation

We begin with defining a natural state representation of the minimax learning game in Protocol 4, using the examples queried by the learner so far, motivated by the definition of identifiability for determining the termination condition.

Definition 14. *Given the set of labeled examples and their labels S, and the queried examples S_X, classifier $h \in \mathcal{H}$ is said to be identifiable with respect to (S, S_X), if (1) h is consistent with S; (2) for all $h' \in \mathcal{H}$ consistent with S,*

$$h'(\mathcal{X} \setminus S_X) = h(\mathcal{X} \setminus S_X) \implies h' = h$$

The above definition naturally motivates the following definition of effective version space:

Definition 15. *Given the set of labeled examples and their labels S, and the queried examples S_X, define its induced effective version space as*

$$F(S, S_X) = \left\{h \in \mathcal{H} : h \textit{ is identifiable with respect to } (S, S_X)\right\}$$

With this, it is natural to recursively define the game and its optimal value function using this state representation:

$$f(S, S_X) = \begin{cases} -\infty, & F(S, S_X) = \emptyset \\ 0, & |F(S, S_X)| = 1 \\ \min_{x \in \mathcal{X} \setminus S_X} \max \begin{pmatrix} f(S \cup \left\{(x, \perp)\right\}, S_X \cup \{x\}) \\ 1 + f(S \cup \left\{(x, +1)\right\}, S_X \cup \{x\}) \\ 1 + f(S \cup \left\{(x, -1)\right\}, S_X \cup \{x\}) \end{pmatrix}, & |F(S, S_X)| \geq 2, \end{cases}$$

Here, we use the base-case game payoffs to encode the labeler's promise of identifiability. Non-identifiability ($F(S, S_X) = \emptyset$) leads to a terminal payoff of $-\infty$. Identifiability constrains the labeler to not provide arbitrary labels and "string along" the learner for as long as possible. As we will later see, this constraint is not crucial, as the algorithm we develop is also robust to a labeler that does not guarantee identifiability.

2.13.1.1 Version Space-based Game Representation

We now turn to the version space game representation, which we use throughout, and prove it is correct.

Definition 16. *Given a labeled dataset S and a set of classifiers V, define version space $V[S] = \big\{h \in V : \forall (x,y) \in S \wedge y \neq \perp, h(x) = y\big\}$ as the subset of classifiers in V consistent with S.*

Definition 17. *Given the set of labeled examples and their labels S, and the queried examples S_X, classifier $h \in \mathcal{H}$ is said to be identifiable with respect to (S, S_X) if:*

- *h is consistent with S, $h \in \mathcal{H}[S]$.*
- *for all other consistent $h' \in \mathcal{H}[S]$: $h'(\mathcal{X} \setminus S_X) = h(\mathcal{X} \setminus S_X) \implies h' = h$, where for brevity we denote $h_1(S_X) = h_2(S_X) \iff \forall x \in S_X \,\text{.}\, h_1(x) = h_2(x)$.*

Definition 18. *Given a set of classifiers V and a set of queried examples S_X, define*

$$E(V, S_X) = \big\{h \in V : \forall h' \in V \setminus \{h\} : h'(\mathcal{X} \setminus S_X) \neq h(\mathcal{X} \setminus S_X)\big\}$$

as the effective version space (E-VS) with respect to V and S_X.

The following proposition relates the effective version space to the classical notion of version space:

Proposition 25.

$$F(S, S_X) = E(\mathcal{H}[S], S_X)$$

Proof.

$$\begin{aligned} h \in F(S, S_X) &\Leftrightarrow h \in \mathcal{H}[S] \wedge \forall h' \in \mathcal{H}[S], h'(\mathcal{X} \setminus S_X) = h(\mathcal{X} \setminus S_X) \implies h' = h \\ &\Leftrightarrow h \in \mathcal{H}[S] \wedge \forall h' \in \mathcal{H}[S], h' \neq h \implies h'(\mathcal{X} \setminus S_X) \neq h(\mathcal{X} \setminus S_X) \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(taking the contrapositive)} \\ &\Leftrightarrow h \in E(\mathcal{H}[S], S_X) \end{aligned}$$

□

Thus, another potential state space representation is using the version space and the unlabeled examples that has been queried. The following structural lemma justifies that this is also a valid representation.

Lemma 26. $f(S, S_X) = \mathrm{Cost}(\mathcal{H}[S], S_X)$

Proof. We prove this by backward induction on S_X.

Base case: $S_X = \mathcal{X}$**.** In this case, $F(S, \mathcal{X}) = E(\mathcal{H}[S], \mathcal{X})$ has size 0 or 1; in both cases, $f(S, S_X) = \mathrm{Cost}(\mathcal{H}[S], S_X)$ by their respective definitions in the bases cases.

Inductive case. Suppose $f(S, S_X) = \mathrm{Cost}(\mathcal{H}[S], S_X)$ holds for any S and any S_X such that $|S_X| \geq j+1$. Now consider any S and any S_X of size j.

If $F(S, S_X) = E(\mathcal{H}[S], S_X)$ has size 0 or 1, $f(S, S_X) = \mathrm{Cost}(\mathcal{H}[S], S_X)$ holds true.

Otherwise, $|F(S, S_X)| = |E(\mathcal{H}[S], S_X)| \geq 2$. By inductive hypothesis, for any $x \in \mathcal{X} \setminus S_X$:

$$\begin{aligned}
f(S \cup \{(x, \perp)\}, S_X \cup \{x\}) &= \mathrm{Cost}(\mathcal{H}[S \cup \{(x, \perp)\}], S_X \cup \{x\}) \\
f(S \cup \{(x, +1)\}, S_X \cup \{x\}) &= \mathrm{Cost}(\mathcal{H}[S \cup \{(x, +1)\}], S_X \cup \{x\}) \\
f(S \cup \{(x, -1)\}, S_X \cup \{x\}) &= \mathrm{Cost}(\mathcal{H}[S \cup \{(x, -1)\}], S_X \cup \{x\})
\end{aligned}$$

Therefore, for any x:

$$\max \begin{pmatrix} f(S \cup \{(x, \perp)\}, S_X \cup \{x\}) \\ 1 + f(S \cup \{(x, +1)\}, S_X \cup \{x\}) \\ 1 + f(S \cup \{(x, -1)\}, S_X \cup \{x\}) \end{pmatrix} = \max \begin{pmatrix} \mathrm{Cost}(\mathcal{H}[S \cup \{(x, \perp)\}], S_X \cup \{x\}) \\ 1 + \mathrm{Cost}(\mathcal{H}[S \cup \{(x, +1)\}], S_X \cup \{x\}) \\ 1 + \mathrm{Cost}(\mathcal{H}[S \cup \{(x, -1)\}], S_X \cup \{x\}) \end{pmatrix}$$

Taking minimum over $x \in \mathcal{X} \setminus S_X$, we also have $f(S, S_X) = \mathrm{Cost}(\mathcal{H}[S], S_X)$.

This completes the induction. □

2.14 Discussions on Additional Related Works and Formulation

2.14.1 Additional Related Works

More related AL works: Our technical results are inspired by the minimax results on exact learning in Hanneke [131]. The noisy setup we consider is similar to that of e.g. Castro and Nowak [57]. Our algorithm belongs the class of "aggressive" learning algorithms [81, 124], which has been of interest for their sample-efficiency. As in [245], we also study label-dependent cost.

Abstaining Classifiers: Prior works have studied the task of learning a predictor with the ability to abstain [233, 326]. Our settings differ in that we aim to learn the true classifier that does not abstain. Rather, it is the labeler that can abstain during the learning process to slow-down learning.

Cross space learning: One of our constructions is related to the cross space learning [274] setup, where each sample is represented in multiple instance spaces. The key observation is that a strategic labeler can force learning on the instance space with the highest sample complexity, by abstaining on all other instance spaces.

Strategic Machine Learning: Strategic ML is a line of work concerned with agent manipulation of inputs into the ML model [136]. Much of this topic has focused on inference-time feature manipulation to influence the model output. And among this large body of work, there is a subset that deal with strategic manipulation of labels. In these settings, there are multiple agents, each of whom can (mis)reports their data point label to manipulate the final model trained on all of their collective data [65, 89, 227]. This line of work largely focuses on the linear-regression setting, under various notions of strategyproofness.

Our work differs from this body of work in considering, at training time (instead of at inference time), how a single labeler can maximize the query complexity of a learner under general hypothesis classes, which includes the linear hypothesis class.

Economics of Knowledge Transfer: We note that the idea of strategically slowing down the transfer of knowledge is not a novel conception. It is a real strategy that people have been documented to use in apprenticeships for example [110, 111], spanning across several industries such as law, entertainment and culinary arts. There are two reasons that motivate the slowed transfer of expertise.

Firstly, as described in [110, 111], before the apprentice has learned everything and can graduate, he will be working for the teacher (or master as is often used in apprenticeship parlance) and performing labor for cheap. Thus, this incentivizes the master to slowly down training, so that the apprentice takes longer to graduate and the master can enjoy this cheap labor for longer.

Secondly, the master can better protect the value of his expertise by slowing down the transfer of his expertise. Overly fast transfer of the master's know-how would graduate too many apprentices too quickly, all of whom also have the same expertise and could thus reduce the value of the master's expertise.

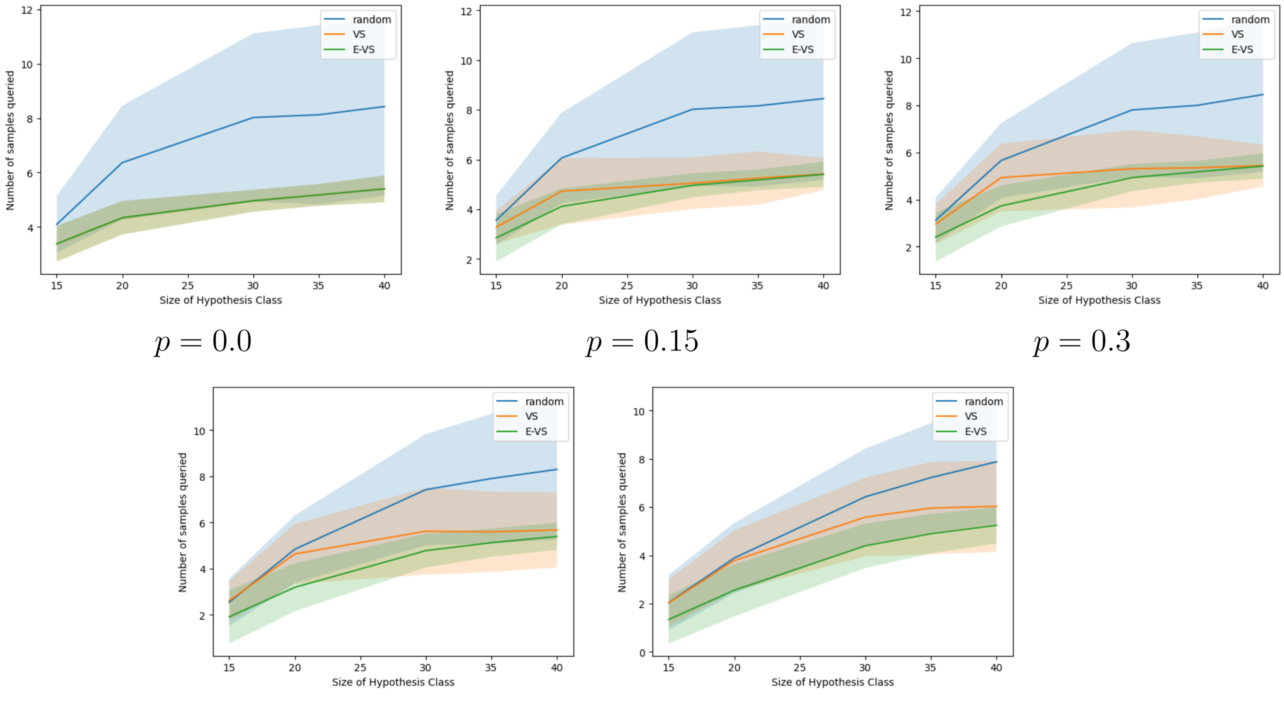
In our setting, we consider the relationship between a human teacher (labeler) and a student (machine). There is a similar incentive at play in that, while the learner has yet to learn h^*, the labeler is paid by the learner for the training labels provided. But once h^* is identified, the student has no need for the teacher. And so, this incentivizes the labeler to slow down learning, in order to give and be paid for as many labels as possible. One difference we note is that in this setting, the transfer of expertise has more serious consequences in rendering the labeler's expertise obsolete, which is not the case in the apprenticeship setting.

2.15 Experiments

To supplement our theoretical minimax analysis in the main section, we examine the performance of three learning algorithms, E-VS bisection, VS-bisection and randomly query (a point), in "average-case" settings by randomly generating learning instances.

Experiment Setup: We consider five sizes for the hypothesis class ranging from 15 to 40. Given a particular hypothesis class size $|\mathcal{H}|$, we generate 50 random learning instances by randomly generating the binary labels of hypotheses on examples $x \in \mathcal{X}$, where the number of data points $|\mathcal{X}|$ is varied from 5 to 30. Given a learning instance, we consider setting (the underlying hypothesis) h^* to be every $h \in \mathcal{H}$, and thus average the query complexity across random instances as well as across $\mathcal{H}$. This is done to explore the average-case query complexity, where we do not focus on the query complexity of one particular $h^* = h \in \mathcal{H}$ (as was done in some of the worst-case analyses).

We investigate two possible labeling strategies, with varying amounts of abstention $p = 0.0, 0.15, 0.3, 0.45, 0.6$. The first strategy is that given the underlying hypothesis $h^* \in \mathcal{H}$, it abstains on labeling a point x with probability p, and outputs $h^*(x)$ otherwise (w.p. $1 - p$). This labeling strategy may be viewed as one that abstains arbitrarily, and may compromise identifiability. This models the labeling strategy of a myopic labeler. The second strategy is a more careful, adaptive labeling strategy that always ensures identifiability. Given the underlying



$p = 0.0$ $p = 0.15$ $p = 0.3$

$p = 0.45$ $p = 0.6$

Figure 2.3: The average number of examples queried by each algorithm across 50 randomly generated instances, along with its standard deviation (shaded region). For this set of plots, the labeling oracle is random (and may not ensure identifiability), with varying probability of abstention p. In the plots, the lower the average, the better the algorithm (needing fewer samples).

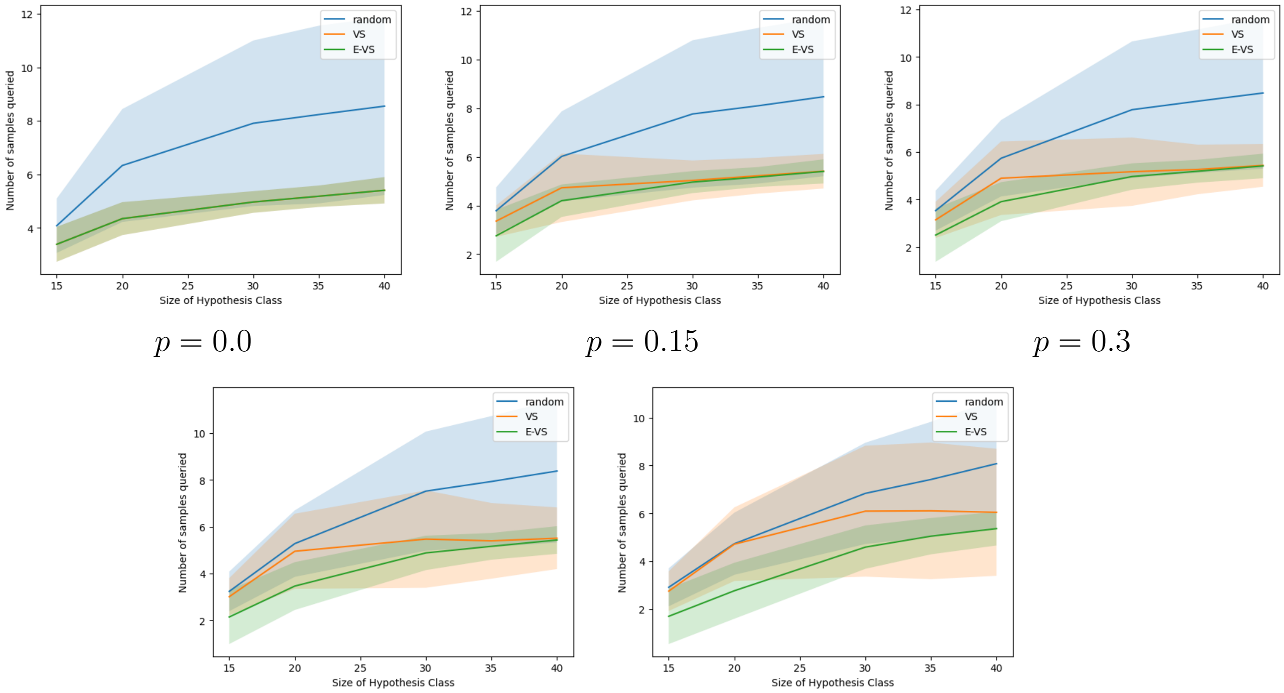
h^*, when x is queried, it computes the resultant E-VS if x was abstained upon. If abstention leads to non-identifiability, it labels x and returns $h^*(x)$. Otherwise, it abstains with probability p and provides the label otherwise. This may be viewed as a more shrewd labeling strategy that always ensures identifiability, while using some abstention.

Results: We plot results in Figure 2.3 and Figure 2.4, with Figure 2.3 corresponding to the first (random labeling) strategy and Figure 2.4 corresponding to the identifiable labeling strategy.

We have a few observations. First, as a sanity check, we observe that in the absence of abstention ($p = 0.0$), the E-VS and VS algorithm behave exactly the same and thus their performance should match, which they do as in the first plot of both Figure 2.3 and Figure 2.4.

Next, we observe the general trend that the E-VS algorithm attains the lowest query complexity, followed by the VS algorithm and then the random querying algorithm. Moreover, the gap becomes more pronounced with the amount of abstention. This makes sense because the E-VS representation is designed to handle abstention, while the VS is not. This trend thus illustrates the effectiveness of using the E-VS representation in face of an abstaining labeler.

Finally, we see that the gap is most significant in face of a non-identifying labeler (as in plots of Figure 2.3). This is because the E-VS algorithm can do early detection of non-identifiability and aptly halt the interaction, while the VS bisection and random querying algorithm cannot detect non-identifiability due to the use of the VS representation. We proved that the query complexity



$p = 0.0$ $p = 0.15$ $p = 0.3$

$p = 0.45$ $p = 0.6$

Figure 2.4: The average number of examples queried by each algorithm across 50 randomly generated instances, along with its standard deviation (shaded region). For this set of plots, the labeling oracle is identifiable, with varying probability of abstention p. In the plots, the lower the average, the better the algorithm (needing fewer samples).

can be significantly larger in a worst-case setup in Theorem 2. And here, we see that in addition to the worst-case setting (as in Theorem 2), the E-VS also fares better in the average-case. Thus, this again affirms the robustness of the E-VS algorithm in face of a non-identifying labeler.

Chapter 3

Strategic Prediction

3.1 Introduction

With the increasing use of machine learning models in automating decision making, there is growing concern over the opacity of these models. Such concerns have given rise to laws, such as the European GDPR, which aim to provide a "Right to Explanation"[98, 251, 286]. However, one stumbling block to this solution is the tension between transparency and gaming. Greater transparency into the predictive model gives rise to gaming – individuals strategically misreporting their features to induce desired classification outcomes from the ML model.

As a result, government agencies are still reluctant to reveal details about their deployed ML algorithms that make predictions on strategic individuals, which we term strategic prediction. Subsequently, Freedom of Information requests have been filed by civil interest groups in the Netherlands [293]. And organized movements such as the OpenSCHUFA project [219] have formed, through which citizens take matters into their own hands and crowd-source data in order to reverse-engineer the algorithms.

In this work, we investigate this tension in strategic prediction through a natural, formal model. To the best of our knowledge, this is the *first formal model* that captures the tradeoff between transparency and gaming in strategic machine learning.

The setting we will study is one where an organization uses model $h^* : \mathcal{X} \to \{-1, +1\}$ to perform classification over feature space $\mathcal{X}$. At the same time, the organization provides transparency through model explanations. We focus on example-based explanations $\mathcal{E}$, which have been found to be one of the most intuitive types of explanations in a recent human study [155], and in particular on prototype-based explanations (e.g k-medoid or MMD-critic [162]).

In more detail, the explanation mechanism $\mathcal{E} : \mathcal{X} \to 2^{\mathcal{X}}$ will select a representative subset of $\mathcal{X}$ to label and explanations $\{(x, h^*(x)) \mid x \in \mathcal{E}(\mathcal{X})\}$ will be released. For example, for loan applications, such explanation could be in the form of past, anonymized (un)successful profiles.

Intuitively, the concern with releasing explanations is that applicants may use the knowledge of the hypothesis class $\mathcal{H} \ni h^*$ along with the explanations to construct the version space (VS), $\mathcal{H}_C = \{h \in \mathcal{H} \mid h(x) = h^*(x), \forall x \in \mathcal{E}(\mathcal{X})\}$, to infer h^*. If the explanation is "good" and allows for "simulatability" of h^* [209], then the few models in $\mathcal{H}_C$ would be constrained by the explanations to have very similar predictions on $\mathcal{X}$ as h^*. And so, even though the VS does not *directly* identify h^*, the VS allows one to estimate h^*'s prediction with high certainty. This we will formalize soon.

To address this issue, we propose *margin-distancing* as a simple and general method that can make the *tradeoff* between transparency and gaming. We show that with margin-distancing it need not be one or the other: it is possible to offer individuals *some idea* of how the model works while still preventing gaming in strategic prediction.

Concretely, given classification models h^* and input example x, we use $f^* : \mathcal{X} \to \mathbb{R}$ to denote a function that outputs an underlying margin score, $h^*(x) = \mathrm{sign}(f^*(x))$, where $\mathrm{sign}(a) = +1$ for $a \geq 0$ and $\mathrm{sign}(a) = -1$ otherwise. Margin-distancing selects a subset of $\mathcal{X}$ whose margin score $\left|f^*(x)\right|$ is greater than some threshold α. This is done to induce a sufficiently large $\mathcal{H}_C$ and, as a result, sufficiently low certainty on how h^* predicts to dissuade gaming.

This approach is compatible with any example-based explanations. We note that our approach is also applicable with local surrogate based methods with bounded fidelity region. Indeed, these methods may be viewed as example-based explanation methods that impart labels for all points within the fidelity regions.

Our Contributions:

(1) We formalize the tradeoff between transparency and gaming, and propose *margin-distancing* as a way of making this tradeoff.

(2) We prove that margin-distancing does *monotonically* decrease decision boundary certainty under a uniform prior over homogeneous linear models and spherical feature space. We also give a set of complementary negative results showing that monotonicity does not hold in general.

(3) We evaluate boundary points' certainty using sampling for general model classes. Our empirical studies suggest margin-distancing does reduce boundary certainty in a relatively monotonic fashion, and in some cases, completely monotonically, which would enable binary search as a computationally efficient means of finding the optimal amount of explanations to release.

3.2 Related Works

Transparency vs Gaming: To the best of our knowledge, there has been only one technical paper [281] that examines the tension between explanation and gaming. In this work, an organization focuses on releasing an optimal set of counterfactual explanations S to induce agents to change their reports in a way that maximizes the organization's utility; this work does not focus on examining the tradeoff explored in our paper. Moreover, the key assumption that differs from our setting is that all feature alteration is viewed as being causal. Lastly, in our work, we do not assume that agents can only change to points in S (if possible), but rather to any point $\hat{x}$ in the neighborhood of x.

Strategic ML: Similar to most of strategic classification literature [65, 94, 136, 163], we assume strategic behavior is gaming. However, different from most, past formulations, agents in our setting do not have *full knowledge* of h^* and have to best respond with only partial knowledge (explanations) of h^*.

In the interest of space, we have included further related works on topics including Improvement vs Gaming, Explanation Manipulation in Appendix 3.10.

3.3 Problem Formulation

Gaming: We assume all individuals desire to be classified the positive label (e.g "loan granted") by h^*. An individual with profile x may use the explanations of h^* to compute and misreport $\hat{x} \neq x$ so as to improve the chance of being classified as the positive label. As is standard in strategic classification, this act of misreporting is referred to as *gaming* [136].

In face of gaming, the organization wishes to have its predictions be unaffected by the release of explanations $\mathcal{E}(\mathcal{X})$: $h^*(\hat{x}) = h^*(x), \forall x \in \mathcal{X}$.

For our analysis, we first assume that applicants cannot report arbitrary profiles – otherwise everyone will simply report some $x \in \mathcal{E}(\mathcal{X})$ with a positive label. This assumption may also be motivated as follows: in strategic ML literature, individuals are typically assumed to have a cost function. This naturally induces a region beyond which it is too costly to change to. For modeling purposes, we assume that if an applicant has feature x, then $\hat{x} \in \mathcal{R}_r(x) := \{x' \mid \|x - x'\| < r, x' \in \mathcal{X}\}$, with $r > 0$ being the maximum extent of manipulation. Additionally, we assume that applicants are aware of the model class $\mathcal{H} \ni h^*$ used by the organization.

Next, since the explanations only allow one to conclude that $h^* \in \mathcal{H}_C$, we need to specify how individuals reason about whether to misreport x' or report x truthfully with only *partial knowledge* about h^*. To model this calculus, as is common in Economics, we assume that the individual is Bayesian and calculates the *increased* chance of obtaining positive label under x' instead of x through a prior distribution $\mathcal{U}$ that gets updated to posterior $\mathcal{U}(\mathcal{H}_C)$ (the restriction of $\mathcal{U}$ on the set $\mathcal{H}_C$) with knowledge of $\mathcal{E}(\mathcal{X})$:

$$\begin{aligned}\pi(x, x') = \mathrm{Pr}_{h \sim \mathcal{U}(\mathcal{H}_C)}(h(x') = 1) \\ - \mathrm{Pr}_{h \sim \mathcal{U}(\mathcal{H}_C)}(h(x) = 1).\end{aligned}$$

A natural choice for $\mathcal{U}$ is the uniform distribution, though it need not be so. We assume that the organization also knows $\mathcal{U}$.

Naturally, individuals will choose to misreport if there is a sufficiently high certainty of success, since they obtain positive utility for getting the positive label (i.e if h^* is s.t $h^*(\hat{x}) = 1$). However, in misreporting, they incur negative utility for the cost of manipulation: $x \to x'$. These two may be weighted linearly in rational agents or nonlinearly in behavioral agents due to risk-aversion [160]. Following the formal model of the rationality of crime as introduced by Becker [39], we abstract this away by assuming that there is some threshold κ such that if $\pi(x, x') \leq \kappa$, the individual is too risk-averse to misreport $\hat{x} = x'$: the cost of manipulation offsets the increased likelihood of obtaining positive utility through positive classification.

This brings us to our main insight: we only need $\mathcal{H}_C$ to be sufficiently ambiguous near the decision boundary because *only* individuals with points near the boundary can misreport in a way that flips h^*'s prediction.

Formally, define the set of *boundary points* to be all x's where such a label flip is possible: $\mathcal{N}_r(\mathcal{X}) := \{x \in \mathcal{X} \mid \exists x' \in \mathcal{R}_r(x) \wedge h^*(x') \neq h^*(x)\}$. Similarly, we define *boundary pairs* to be pairs (x, x') that are within a distance of r, but predicted differently by h^*; formally, $\mathcal{M}_r(\mathcal{X}) := \left\{(x, x') \in \mathcal{X}^2 \mid x' \in \mathcal{R}_r(x) \wedge h^*(x') \neq h^*(x)\right\}$. Observe that $\mathcal{M}_r(\mathcal{X}) \subset \mathcal{N}_r(\mathcal{X})^2$.

Margin-distancing: To make it difficult to infer the decision boundary through $\mathcal{H}_C$, it is natural to remove explanations that are close to the decision boundary. This gives rise to our

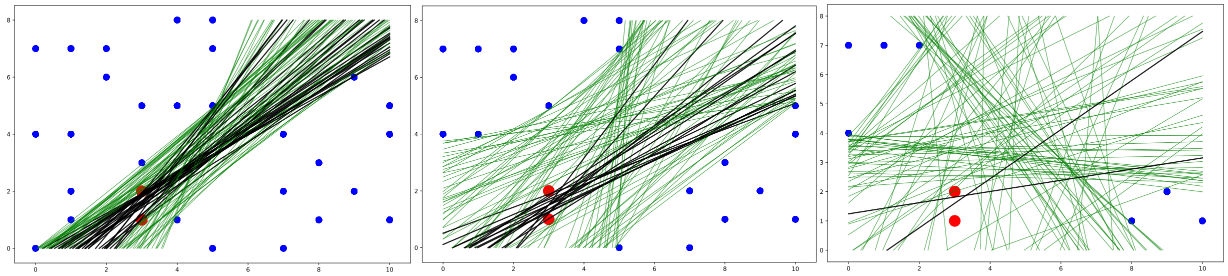


Figure 3.1: Visualization of $\mathcal{H}_C$ in a toy example where the amount of explanations (blue points) is varied $(80, 50, 20$ percent of all explanations is kept). In red is one randomly chosen boundary pair. 100 lines (green and black) are randomly sampled from $\mathcal{H}_C$; in black are lines that predict the pair like h^* (opposite labels), and green the same.

approach of *margin-distancing*. We will designate some indicator function Λ_α for choosing explanations, which evaluates to 1 iff the examples' classification margin score is greater than cutoff α; formally, $\mathcal{E}_{h^*}(\mathcal{X}, \alpha) = \big\{x \in \mathcal{X} : \Lambda_\alpha(x) = 1\big\}$. Note that $\mathcal{H}_C$ is a function of α, since $\mathcal{H}_C$ is a function of the explanations, which are in turn a function of α. Intuitively, a big α that only retains explanations with large margins would decrease *boundary certainty*, which we define as $\max_{(x,x') \in \mathcal{M}_r(\mathcal{X})} \pi(x, x')$.

Policy Goals: Herein lies the tradeoff for the organization:

1) Provide explanation $\mathcal{E}_{h^*}(\mathcal{X}, \alpha)$ such that the boundary certainty is made sufficiently low: $\max_{(x,x') \in \mathcal{M}_r(\mathcal{X})} \pi(x, x') \leq \kappa$. This makes all individuals $x \in \mathcal{X}$ too risk-averse to misreport $\hat{x} \in \mathcal{R}_r(x)$ with $h^*(\hat{x}) \neq h^*(x)$, thus preventing gaming.

2) The explanation provided $\mathcal{E}_{h^*}(\mathcal{X}, \alpha)$ is as transparent as possible. That is, α is as small as possible to retain as many explanations from the full set of explanations as possible. Naturally, in our setting, we define transparency to be the amount of explanations that remain after margin-distancing.

The technical problem we study is:

> How can we search for the smallest threshold α possible such that $\max_{(x,x') \in \mathcal{M}_r(\mathcal{X})} \pi(x, x') \leq \kappa$, which is needed to prevent gaming?

Before we proceed, we obtain some intuition first through a qualitative visualization of $\mathcal{H}_C$ in a toy example, Figure 3.1. This figure helps to confirm that allowing explanations with small margins "boxes in" the version space too much, and makes models in $\mathcal{H}_C$ too similar to h^*. And so, removing explanations with small margin help enlarge $\mathcal{H}_C$ and decrease boundary-certainty.

Simple Example: Next, for a quantitative toy example, consider when $\mathcal{X} = [0, 1]$ and $\mathcal{H} = \{h_w(x) := \text{sign}(x - w) \mid w \in [0, 1]\}$ is the class of 1D thresholds. Let $\mathcal{U}$ be the uniform distribution over $\mathcal{H}$. We know then that $w^* \in [x^-, x^+]$, where x^- is the largest negative point in $\mathcal{E}(\mathcal{X})$ and x^+ the smallest positive point. Therefore, for $x \in (x^-, x^+)$ and some $x' \in \mathcal{R}_r(x) > x$, we have that $\pi(x, x') = \frac{\min\{x', x^+\} - x}{x^+ - x^-}$. In this case, it is evident that margin-distancing (i.e increasing x^+ and decreasing x^-) decreases boundary certainty $\pi(x, x')$.

In the section that follow, we study a more general hypothesis class and verify that the intuitive trend of removing information around the decision boundary does make it more difficult to infer the decision boundary, thus reducing boundary certainty.

r	max extent of manipulation
α	min distance from the margin
$\Pi(\alpha)$	boundary certainty, $\Pi(\alpha) = \max_{(x,x') \in \mathcal{M}_r(\mathcal{X})} \pi_\alpha(x, x')$
ϕ	max angle between $w \in \mathcal{H}_C$ and w^*; related to α by $\alpha = \sin \phi$
ψ	max angle: related to r by $\cos \psi = 1 - r^2/2$

Table 3.1: A table of notations that appears in Section 3.4.

3.4 Homogeneous Linear Models

We focus our theoretical study on the property of *monotonicity*, which if true, allows for binary search as an efficient way to compute the optimal α. In this section, we identify homogeneous linear models in $\mathbb{R}^d$, i.e. $\mathcal{H} = \{h_w \mid \|w\|_2 = 1\}$ (where $h_w := x \mapsto \text{sign}(\langle w, x \rangle)$), as one setting where margin-distancing monotonically leads to decreased boundary certainty.

For the results that follow, we also assume that individuals have uniform prior $\mathcal{U}$ over $\mathcal{H}$. We will also focus on when the feature space $\mathcal{X}$ is the origin-centered unit sphere, i.e., $\mathcal{X} = \{x \in \mathbb{R}^d \mid \|x\|_2 = 1\}$, which means that $r \leq 2$. Intuitively, this corresponds to a normalized dataset with profiles of "all kinds", which is not unreasonable for profiles of a general population. We handle more general settings in the following section.

For linear models, it is natural to take Λ to be a function of the margin of a point with respect to w^* (the parameter of h^*): $\Lambda_\alpha(x) = \mathbb{1}\{|\langle w^*, x \rangle| > \alpha\}$, for $\alpha \in [0, 1)$. Therefore, for every α, its associated set of explanations is $\mathcal{E}_{h^*}(\mathcal{X}, \alpha) = \Big\{x \in \mathcal{X} : |\langle w^*, x \rangle| > \alpha\Big\}$.

Under this "nice" setting, we first show that we can give a simple characterization of the version space in terms of α:

Lemma 27. *Fix $\alpha \in [0, 1)$. Recall that $\mathcal{H}_C = \{h \in \mathcal{H} \mid h(x') = h^*(x'),\ \forall x' \in \mathcal{E}_{h^*}(\mathcal{X}, \alpha)\}$ is the version space induced by explanation $\mathcal{E}_{h^*}(\mathcal{X}, \alpha)$. $\mathcal{H}_C$ can be equivalently written as:*

$$\mathcal{H}_C = \Big\{h_w \mid \|w\|_2 = 1, w \cdot w^* \geq \sqrt{1 - \alpha^2}\Big\}.$$

For ease of the exposition of the next theorem, we reason in the spherical counterpart to α and r:

- Define ϕ to be the maximum angle between any $w \in \mathcal{H}_C$ and w^*. From Lemma 27, under explanation $\mathcal{E}_{h^*}(\mathcal{X}, \alpha)$, $\phi = \arccos(\sqrt{1 - \alpha^2}) = \arcsin \alpha$. Intuitively, ϕ measures how large $\mathcal{H}_C$ is and shrinks with a bigger set of explanations.
- Define $\psi = 2 \arcsin(\frac{r}{2}) = \arccos(1 - \frac{r^2}{2})$. The boundary region $\mathcal{N}_r(\mathcal{X})$ may then be described as the set of points $\{x \in \mathcal{X} \mid \langle w^*, x \rangle \in [-\sin \psi, \sin \psi)\}$. Intuitively, ψ measures how "thick" the boundary region is. Geometrically, this means that $\theta(x, w^*) \in [\pi/2 - \psi, \pi/2 + \psi]$ for x in the boundary region, where $\theta(x, w^*)$ denotes the angle between x and w^* the decision boundary: $\theta(u, v) = \arccos(\frac{\langle u, v \rangle}{\|u\|_2 \|v\|_2}) \in [0, \pi]$.

Please refer to Figure 3.2 for an illustration of notation ϕ and ψ, which we note are both acute by definition, and refer to Table 3.1 for a summary of definitions.

Firstly, it is clear that increasing boundary thickness ψ leads to a larger $\mathcal{M}_r(\mathcal{X})$, therefore a higher $\max_{(x,x') \in \mathcal{M}_r(\mathcal{X})} \pi_\alpha(x, x')$. We derive an analytical form of $\max_{(x,x') \in \mathcal{M}_r(\mathcal{X})} \pi_\alpha(x, x')$

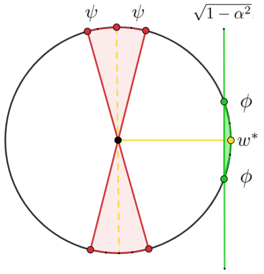


Figure 3.2: Visualization of the notation: $\mathcal{H}_C$ in green, boundary region in red and true model w^* in yellow.

below that formalizes this.

Theorem 11. *We have:*

$$\max_{(x,x')\in\mathcal{M}_r(\mathcal{X})} \pi_\alpha(x,x') = \begin{cases} \frac{\int_0^{\psi/2} F(\theta)d\theta}{\int_0^{\phi} F(\theta)d\theta} & \psi \leq 2\phi \\ 1 & \psi > 2\phi, \end{cases}$$

where $F(\theta) = (1 - \frac{\cos^2\phi}{\cos^2\theta})^{d/2-1}$*; therefore, it is strictly increasing for* ψ *in* $[0, 2\phi]$.

Our next two theorems consider the margin-distancing effect in terms of α. For simplicity and to relate α's effect on $\mathcal{H}_C$ through explanations $\mathcal{E}_{h^*}(\mathcal{X},\alpha)$, we subsequently abbreviate boundary certainty $\max_{(x,x')\in\mathcal{M}_r(\mathcal{X})} \pi_\alpha(x,x')$ as $\Pi(\alpha)$.

To recap, a higher threshold α, corresponding to more margin-distancing, leads to a smaller set of explanations $\mathcal{E}_{h^*}(\mathcal{X},\alpha)$ (lowered transparency since more explanations are removed) and thus a bigger $\mathcal{H}_C$. This leads to lower boundary certainty $\Pi(\alpha)$, preventing gaming.

In the next result, we show that $\Pi(\alpha)$ is provably monotonically decreasing in α. Thus, this enables the use of binary search to efficiently find the optimal α. Indeed, it is not clear that decreasing the amount of explanations and enlarging the version space will always decrease $\Pi(\alpha)$. The reason is that enlarging $\mathcal{H}_C$ increases both models that agree with h^* on x, x' (black lines in Figure 3.1) and models that do not (green lines). If *proportionally* more of them do predict like h^*, then the new $\pi_\alpha(x,x')$ will actually increase. We prove Theorem 12 that shows this is not so in this "nice" setting; the proof may be found in Appendix 3.7.1.

Theorem 12. $\Pi(\alpha)$ *is decreasing in* α*, for* $\alpha \in [0,1)$*, and is strictly decreasing in* $[\sin(\psi/2), 1)$.

Finally, in some cases, we may skip the search if we can analytically derive conditions on ϕ, ψ in which $\Pi(\alpha)$ is upper bounded. Next, we show that there exists some constant c such that $\lim_{\alpha\to 1} \Pi(\alpha) \leq c\psi$. Thus, when ψ is small and α increases to 1, $\Pi(\alpha)$ decreases to a small value.

Theorem 13.
1. *If* $\alpha \geq 1 - \frac{1}{8d}$*, then* $\Pi(\alpha) \leq 9\psi$.
2. *For any* $C_1 \in (0,1)$*, there exists* $C_2 > 0$ *such that the following holds: if* $\alpha \leq 1 - \frac{1}{\sqrt{d}}$ *and* $\psi \geq \frac{C_2}{d^{1/4}}$*, then* $\Pi(\alpha) \geq 1 - C_1$.

A more refined version of this theorem and proofs of other theorems may be found in Appendix 3.7.

3.5 General Models

For arbitrary feature spaces, it is unclear if it is possible to explicitly characterize $\mathcal{H}_C$ even for non-homogeneous linear models. Still, let us suppose we have devised some function Λ parameterized by threshold parameter α. Algorithmically, how do we search for the smallest α such that $\Pi(\alpha) < \kappa$ for a given κ?

First, we will need an approach to approximate $\Pi(\alpha)$ under a given threshold α. Indeed, there is generally no closed-form expression for $\Pi(\alpha)$, so we will assume access to an algorithm that can sample from the posterior distribution $\mathcal{U}(\mathcal{H}_C)$. Our approach is simply to draw samples $h_1, ..., h_n$ using the algorithm and evaluate: $\hat{\rho}(x') - \hat{\rho}(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{h_i(x') = 1\} - \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{h_i(x) = 1\}$.

To understand the sample complexity needed, we see that, $\hat{\rho}(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{h_i(x) = 1\} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{H^*_x(h_i) = 1\}$, where for a fixed x, $H^*_x : h \mapsto h(x)$ is its associated *dual* function.

Definition 19 (Dual Class)**.** *For any domain $\mathcal{X}$ and set of functions $\mathcal{H}$ whose image is $\{-1, +1\}$, the dual class of $\mathcal{H}$ is defined as $\mathcal{H}^* := \{H^*_x \mid x \in \mathcal{X}\}$.*

As introduced in [17], $\mathrm{VC}(\mathcal{H}^*)$ is finite as long as $\mathrm{VC}(\mathcal{H})$ is finite. And so, with $O\left(\frac{\mathrm{VC}(H^*)+\log 1/\delta}{\epsilon^2}\right)$ random draws, we may obtain an $2\epsilon-$accurate estimation of $\hat{\rho}(x) - \hat{\rho}(x')$ for *all* boundary pairs x, x', due to uniform convergence. This gives us a 4ϵ-accurate estimation of $\Pi(\alpha)$. In the case of linear models, due to point-line duality, we know that $\mathrm{VC}(\mathcal{H}^*) = \mathrm{VC}(\mathcal{H}) = O(d)$, which informs us how many samples are needed to calculate a high fidelity approximation of $\pi_\alpha(x, x')$.

Search: Once we know how to approximate $\max \pi_\alpha(x, x')$ for a given α, if monotonicity does hold, then search for the optimal threshold may be efficiently done through binary search. Recall from Theorem 12 that, if a) the feature space is spherical, and b) the prior distribution over the hypothesis class is uniform, and c) the hypothesis class is homogeneous halfspaces, then $\Pi(\alpha)$ decreases monotonically to $O(\psi)$. To complement this result, we next show that removing one of a, b or c (and keeping the rest) breaks this pattern.

Our next two proposition show that, removing the spherical feature space condition, or removing the assumption of $\mathcal{U}$ being uniform, can cause boundary certainty to *increase* with increasing margin distancing parameter α in worst-case settings.

Proposition 26. *Suppose $d = 2$. We have uniform prior over homogeneous linear models $\mathcal{H} = \{w \in \mathbb{R}^d \mid \|w\| = 1\}$, there exists a feature space $\mathcal{X}$ and thresholds $0 < \alpha_2 < \alpha_1$ such that $\Pi(\alpha_2) < \Pi(\alpha_1)$.*

Proposition 27. *Suppose $\mathcal{X}$ is the d-dimensional unit sphere with $d \geq 3$. There exists a non-uniform distribution $\mathcal{U}$ over homogeneous linear models $\mathcal{H}$, such that there exists thresholds $0 < \alpha_2 < \alpha_1$ with $\Pi(\alpha_2) < \Pi(\alpha_1)$.*

Finally, we show that by removing the assumption that the hypothesis class is the set of homogeneous linear models, $\Pi(\alpha)$ can stay at a high value for all $\alpha \in (0, 1]$ and all $\psi \in (0, \pi]$. This is in sharp contrast with the homogeneous linear model class setting, in which $\lim_{\alpha \to 1} \Pi(\alpha) \leq O(\psi)$ and could thus be made arbitrarily small with $\psi \to 0$.

Proposition 28. *There exists a class of non-homogeneous linear models, with spherical $\mathcal{X}$ such that $\Pi(\alpha)$ decreases monotonically (and strictly so at some point) with increasing α, and yet $\Pi(\alpha) \geq 1/3$ for all $\alpha \in [0, 1)$ and $\psi \in (0, \pi]$.*

Thus, we have that in general monotonicity does not hold. However, our negative results are worst-case in nature. Next, we turn to experiments to examine the relationship between margin-distancing and boundary-certainty on real-world, non-worst case datasets.

3.6 Experiments

In this section, we empirically chart the relationship between margin distancing (the amount of explanation omission) and boundary certainty. We experiment with linear and multi-layer Perceptron (MLP) models.

Explanation Methods: As mentioned in the formulation, we focus on example-based explanation methods that can return a subset of prototypical instances that serve as explanations. This leads us to use k-medoid and MMD-critic [162], and rules out other example-based explanation methods such as [166] that return a single (and not subset), most "influential" data point out of the training set. Note also, that counterfactual and contrastive-based explanations are ruled out by the need to margin-distance. Indeed, by construction, counterfactual/contrastive-based explanations are boundary points, whose release greatly increase the users' boundary certainty – in fact, $\max_{(x,x') \in \mathcal{M}_r(\mathcal{X})} \pi(x, x') = 1$. Thus, if manipulation (gaming) is to be prevented, the use and release of this type of explanations is a non-starter.

Our experimental procedure goes as follows:

1) The explanation method (e.g k-medoid) is used to compute the full set of explanations.

2) Then, we vary the degree of margin-distancing and remove explanations that are too close to the decision boundary. To measure the closeness of an explanation point with respect to the decision boundary, we look at its percentile in the distribution of all explanations' margin scores. This allows us to identify which points are in the top l percent of all explanations closest to the margin. We do this separately for positive and negative explanations as they have different distributions of margin scores.

3) To compute boundary certainty, we remove this top l percent closest explanations, compute models $\mathcal{H}_C$ consistent with the remaining explanations $\mathcal{E}_{h^*}(\mathcal{X}, \alpha)$ and compute $\pi(x, x')$ using $\mathcal{H}_C$.

4) To generate our plots, we vary l for l ranging from 0 to 75 (on the x-axis) and plot this against three metrics that capture boundary certainty (on the y-axis). The three metrics that summarize $\pi(x, x')$ for all boundary pairs (x, x') are: $\max_{(x,x') \in \mathcal{M}_r(\mathcal{X})} \pi(x, x')$ (worst boundary pair), average of top 5 percent of $\pi(x, x')$'s (somewhat worse case) and average of all $\pi(x, x')$.

3.6.1 Linear Models

Procedure: We train a linear model on the `Credit Card Default` dataset [314] using Logistic Regression to obtain w^*. We focus on mutable features only that preclude features age and marital status. We take Λ to be margin distance $\langle w^*, x \rangle$. For these experiments, at a given r, we focus on and use w^* to find the set of all pairs of boundary points (x, x') that lead to a positive flip: $\left\{(x, x') : w^\star \cdot x < 0, w^\star \cdot x' \geq 0\right\}$. This is relatively cheap since by Cauchy-Schwarz, we only need to try all pairs of points whose margin score is $\leq r$, a much smaller set.

For a given set of explanations, we construct and sample from $\mathcal{H}_C$, which is a polytope. Sampling from polytopes is a well-studied problem and we use the state-of-the-art John's Walk

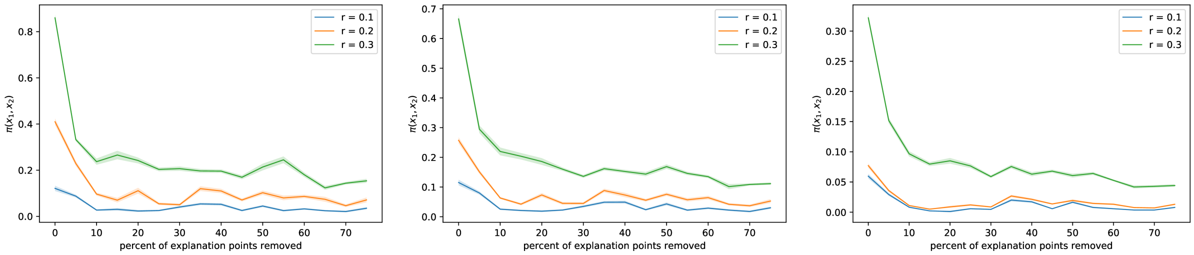


Figure 3.3: Plots of the $\max_{(x,x')\in\mathcal{M}_r(\mathcal{X})} \pi(x,x')$ (left), average of top 5 percent of all $\pi(x,x')$ (middle) and average of all $\pi(x,x')$ (right) under k-medoid explanations for linear models.

[67] with mixing time $O(d^2)$. We assume uniform $\mathcal{U}$ over $\mathcal{H}$. Thus, with these samples, we compute the empirical $\max_{(x,x')\in\mathcal{M}_r(\mathcal{X})} \hat{\pi}(x,x')$ with w's sampled uniformly from $\mathcal{H}_C$. We repeat this sampling 16 times for each set of explanations corresponding to a margin-distance percentile.

Monotonicity: We present our results in Figure 3.3. Qualitatively, we observe a generally smooth decreasing trend with increased distance of explanations from the margin and we observe some non-monotonicity under all three metrics, most prominently under the $\max$ metric. For all three metrics, we see that the trend levels out quickly. This suggests that trying smaller values of α (small amounts of explanation omission) can quickly decrease various measures of boundary certainty and this strategy is effective in this setting.

Quantitatively, we check if the trend is generally monotonic in an experiment that goes as follows. We pick 10 target boundary certainty values evenly spaced out from the attainable boundary certainties as found on the y-axis. Then, for each target value, we find the minimum percent of explanation points that need to be removed to bring the boundary certainty below the target; this optimal percentage is found simply by sweeping through all (percentage, certainty) pairs we have from left to right. Finally, we obtain the percentage that need to be removed as found by binary search and compute the difference between the percentage found by binary search against the optimal.

Under k-medoid explanations for linear model, we summarize the results by looking at the average of the difference and the max difference, which we report as follows. For plots of the $\max_{(x,x')\in\mathcal{M}_r(\mathcal{X})} \pi(x,x')$: $r = 0.1$, 7, 35; $r = 0.2$, 11, 55; $r = 0.3$, 0, 0. For plots of average of top 5 percent of all $\pi(x,x')$: $r = 0.1$, 7, 35; $r = 0.2$, 10, 50; $r = 0.3$, 0, 0. For plots of average of all $\pi(x,x')$: $r = 0.1$, 6, 30; $r = 0.2$, 11, 55; $r = 0.3$, 0, 0. We record the full set of differences in tables in Appendix 3.8.4.

As a synopsis, we observe that the difference is generally small for higher r's and larger for lower r's. The relatively jagged line means that binary search is likely to be quite far off. Here we wish to note that this problem may be alleviated by electing to try the smaller amounts of explanation omission instead of binary search, in the case that we find that the boundary certainties are close at the extremes. Indeed, the closeness would suggest that not much decrease in boundary certainty could be obtained by significantly increasing the percentage of explanation omission.

We also observe the result from varying the allowed extent of manipulation r. As expected, the larger the manipulation extent r, the higher the $\pi(x,x')$ that may be attainable.

3.6.2 Neural Network Models

Procedure: We train MLPs with one or two hidden layers on the `givemecredit`[1] dataset. We present the one layer MLP experiment results in the main body and the two layer in the appendix. We experiment with k-medoid and MMD-critic [162], whose results we present in the appendix. To measure of distance from margin, we take $\Lambda_\alpha(x)$ to be the model's confidence of a point: $\Lambda_\alpha(x) = \mathbb{1}\{|f^*(x)| \geq \alpha\}$, where $f^* : \mathcal{X} \to [-\frac{1}{2}, \frac{1}{2}]$ represents the MLP's predictive probability of class 1, offset by $-\frac{1}{2}$.

To the best of our knowledge, there is no known algorithm that provably sample uniformly from neural network version spaces. Indeed, this is an important problem described by recent works on the "Rashomon effect" [80, 196, 252]. We use the procedure in [80] used to probe the version space: randomly initialize the network with different seeds to obtain different models consistent with the explanations. For computational tractability, we sample 100 MLPs this way with 4 repetitions per margin-distance percentile.

Observations: Our first observation is that varying just the initialization is not an effective sampling procedure under the `givemecredit` dataset. We find small variation in the MLPs produced. To showcase this, we randomly sample 100 pairs of MLPs from the $\mathcal{H}_C$ we collected and calculate their label agreement on the boundary points, $\Pr_{h,h' \sim \mathcal{U}(\mathcal{H}_C), x \sim \text{Unif}(\mathcal{M}_r(\mathcal{X}))}(h(x) = h'(x))$. The high average consistency of $\mathcal{H}_C$ is charted in green in Figure 3.5.

We also compute the three metrics in this setting (Figure 3.6), which interestingly are very high despite the overall low agreement with respect to h^* – defined as $\Pr_{h \sim \mathcal{U}(\mathcal{H}_C), x \sim \text{Unif}(\mathcal{M}_r(\mathcal{X}))}(h(x) = h^*(x))$ (please see right figure in Figure 3.5). This seems to be due to a small fraction of points which most MLPs in $\mathcal{H}_C$ consistently agree with h^* on. The large values of $\max_{(x,x') \in \mathcal{M}_r(\mathcal{X})} \pi(x, x')$ in this case suggests the difficulty of preventing worst-case manipulation when the full set of hyperparameters used to train the network is known.

Indeed, as is noted in [153], it seems generally implausible for attackers to know the *exact* hyperparameters used to train the networks, which has been the assumption in the past model extraction works. And so, from hereon, we experiment with the natural, sampling procedure in the absence of such knowledge, which is just to randomly initialize the network and also the set of hyperparameters (ℓ_2 regularization constant, learning rate, momentum, batch size). These are randomly sampled from uniform distributions that contain the hyperparameters' true values. Verily, this leads to greater variation (please see the yellow barplots in Figure 3.5).

Since neural networks may require higher sample complexity, we also examine data augmentation techniques that one might consider to enhance the explanation set. In addition to 1) just the explanations, we consider 2) explanations plus random draws from Gaussian balls of radius 0.1 around the explanations 3) the full $\{x \mid x \in \mathcal{X}, \Lambda_\alpha(x) = 1\}$, which would correspond to "perfect" extrapolation of the feature space based off of $\mathcal{E}(\mathcal{X})$. The plots are given in Figure 3.4.

Comparing the effectiveness of the data augmentation, We observe small change in the π with mildly augmented data as in 1). However, the full knowledge of the $\{x \mid x \in \mathcal{X}, \Lambda_\alpha(x) = 1\}$ results in higher measures of boundary certainty. Indeed, this is to be expected since more labeled data naturally induces higher boundary certainty.

Monotonicity: In terms of the general trend for monotonicity, we again observe that margin-

[1] http://www.kaggle.com/c/GiveMeSomeCredit/

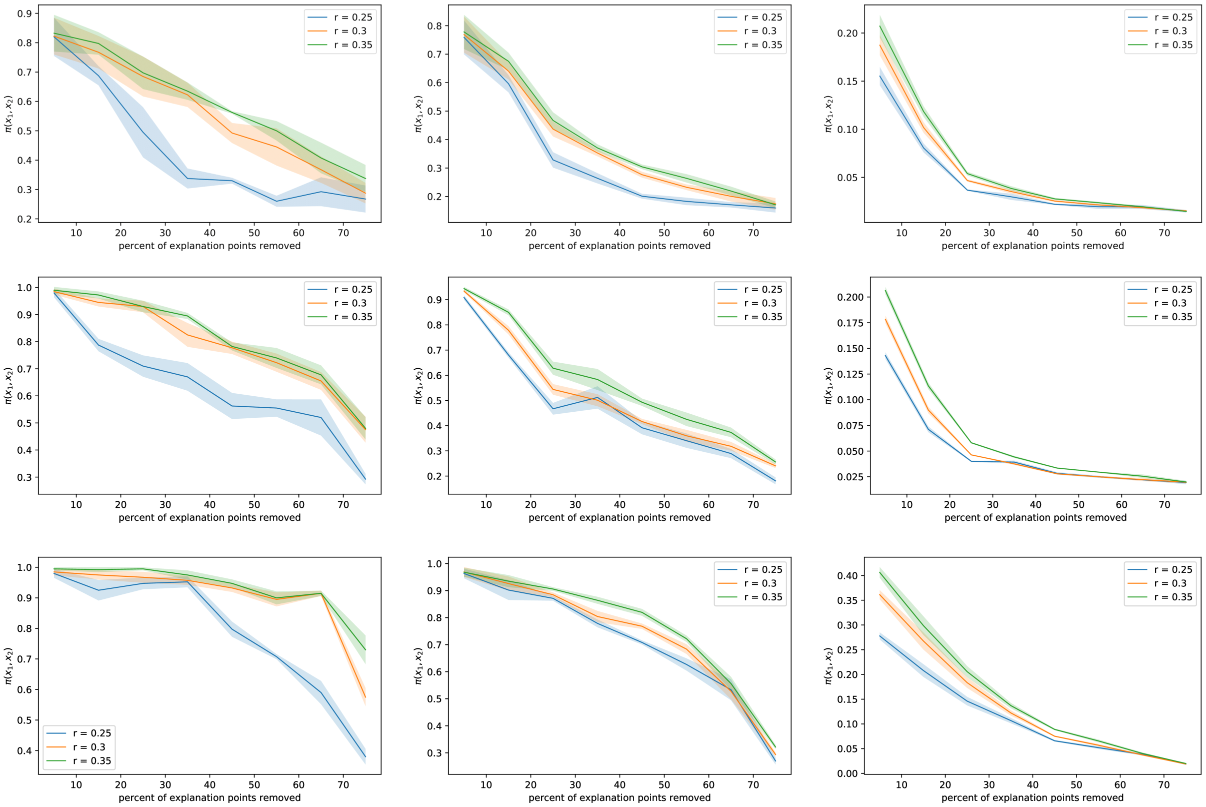


Figure 3.4: MLP results: k-medoid explanations (top), k-medoid explanations + random draws from small balls around the explanations (middle), full $\{x \mid x \in \mathcal{X}, \Lambda_\alpha(x) = 1\}$ (bottom). The three metrics are in column: max $\pi(x, x')$ (left), top 5 percent of all $\pi(x, x')$'s (middle), average $\pi(x, x')$ (right).

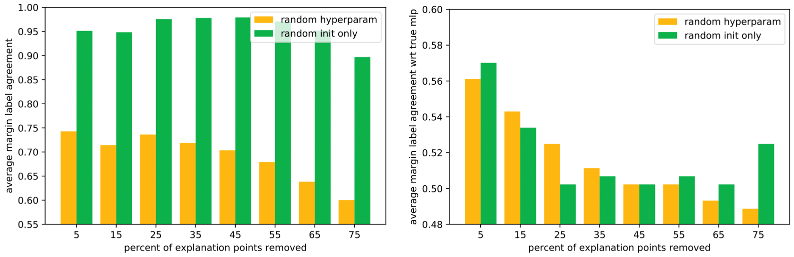


Figure 3.5: Boundary point label agreement within $\mathcal{H}_C$ (left), and boundary point label agreement of $\mathcal{H}_C$ with respect to h^* (right). This is estimated by sampling h from version space using random initializations of parameters (green) and hyperparameters (yellow), respectively.

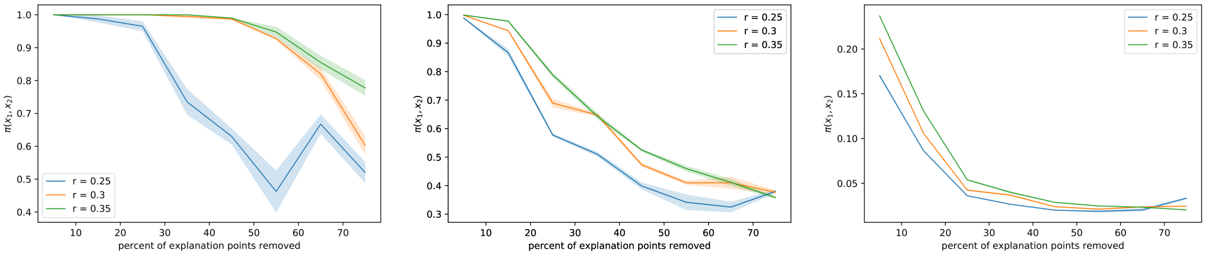


Figure 3.6: Plots of the $\max_{(x,x')\in\mathcal{M}_r(\mathcal{X})} \pi(x, x')$ (left), average of top 5 percent of all $\pi(x, x')$'s (middle) and average of all $\pi(x, x')$ (right) for the MLP case with random initialization only under k-medoid explanations.

distancing does help to reduce all three metrics. Qualitatively, $\max_{(x,x')\in\mathcal{M}_r(\mathcal{X})} \pi(x, x')$ trend is non-monotonic and jagged at places, but smooths out with even a bit of averaging (the latter two metrics). In fact, we see that the average of top 5 percent of $\pi(x, x')$'s and average of all $\pi(x, x')$ metrics are monotonic. This is instructive in that it suggests that binary search could be used to efficiently search for the appropriate threshold.

Quantitatively, we verify if this trend is generally monotonic as before. We pick 10 target boundary certainty values evenly spaced out from the attainable boundary certainties as found on the y-axis. For each target value, we find the minimum percent of explanation points that need to be removed to bring the boundary certainty below the target and compare against the percentage found by binary search.

Under k-medoid explanations for MLP models, we again summarize the results by looking at the average of the difference and the max difference. Here, due to the much smoother curves (relative to those of the linear models) and the large discrepancy in boundary certainties at the two extremes, we find that under all three r's, binary search is able to match the optimal percentage needed to bring the boundary certainty below the target value.

3.6.3 Fair accessibility to explanations

A notable concern that may arise with margin distancing is that though omission of prototypical explanations is necessary, it may disproportionately affect individuals in regions close to the boundary. We plot the composition of the boundary region in the appendix under linear models logistic and SVM models. We observe that margin-distancing does disparately affect the release of explanations to different groups. Verily, this is another important factor that needs to be taken into account in the explanation release process.

3.7 Proofs

3.7.1 Deferred Proofs from Section 3.4

Recall that in Section 3.4, $\mathcal{X}$ is the origin-centered unit sphere in $\mathbb{R}^d$, and $\mathcal{H}$ is the set of homogeneous linear classifiers in $\mathbb{R}^d$, and $\mathcal{U}$ denotes the uniform distribution over $\mathcal{H}$.

In the proofs that follow, we will mainly work in terms of polar angles ϕ and ψ. Recall $\phi = \arcsin \alpha$ is defined to be the maximum angle between any $w \in \mathcal{H}_C$ and w^*, and $\psi = 2\arcsin(\frac{r}{2})$ measures the thickness of the boundary region $\mathcal{N}_r(\mathcal{X})$.

Now, we prove a characterization of the boundary region in terms of ψ.

Fact 1. $\mathcal{N}_r(\mathcal{X}) = \big\{x \in \mathcal{X} \mid \langle w^*, x\rangle \in [-\sin\psi, \sin\psi)\big\}$.

Proof. Recall our definition that $\mathcal{N}_r(\mathcal{X}) := \{x \in \mathcal{X} \mid \exists x' \in \mathcal{R}_r(x) \wedge h^*(x') \neq h^*(x)\}$, where $h^*(x) = \operatorname{sign}(\langle w^*, x\rangle)$. Thus, it suffices to show that

$$\langle w^*, x\rangle \in [-\sin\psi, \sin\psi) \iff \exists x' \in \mathcal{R}_r(x) \centerdot \operatorname{sign}(\langle w^*, x'\rangle) \neq \operatorname{sign}(\langle w^*, x\rangle).$$

We show the implications in both directions.

($\Rightarrow$): Suppose we are given x such that $\langle w^*, x\rangle \in [-\sin\psi, \sin\psi)$. Then x can be represented as $x = \beta w^* + \sqrt{1-\beta^2}x_\perp$, for some $\beta \in [-\sin\psi, \sin\psi)$, and $x_\perp$ is a unit vector perpendicular to w^*. Observe that $x - x_\perp = \beta w^* + (\sqrt{1-\beta^2} - 1)x_\perp$, and therefore,

$$\|x - x_\perp\|_2 = \sqrt{\beta^2 + (\sqrt{1-\beta^2} - 1)^2} = \sqrt{2(1 - \sqrt{1-\beta^2})}.$$

We now consider two cases of β:

1. If $\beta \in [-\sin\psi, 0)$, we consider $x' = x_\perp$. First observe that $x' \in \mathcal{R}_r(x)$. Indeed,

   $$\|x - x'\|_2 = \sqrt{2(1 - \sqrt{1-\beta^2})} \leq \sqrt{2(1 - \cos\psi)} = r.$$

   Meanwhile, $\operatorname{sign}(\langle w^*, x'\rangle) = \operatorname{sign}(0) = 1 \neq -1 = \operatorname{sign}(\beta) = \operatorname{sign}(\langle w^*, x\rangle)$, which establishes the claim.
2. If $\beta \in [0, \sin\psi)$, we first observe that $\|x - x_\perp\| = \sqrt{2(1 - \sqrt{1-\beta^2})} < \sqrt{2(1-\cos\psi)} = r$. Therefore, there exists a small enough $\gamma > 0$, such that $x' = -\gamma w^* + \sqrt{1-\gamma^2}x_\perp$ is close enough to $x_\perp$, and hence lie in $\mathcal{R}_r(x)$. Now, $\operatorname{sign}(\langle w^*, x'\rangle) = \operatorname{sign}(-1) = -1 \neq 1 = \operatorname{sign}(\beta) = \operatorname{sign}(\langle w^*, x\rangle)$, which establishes the claim.

($\Leftarrow$): Assume toward contradiction that $\langle w^*, x\rangle \in [-1, -\sin\psi) \cup [\sin\psi, +1]$. Without loss of generality (due to spherical symmetry) suppose that $w^* = (1, 0, \ldots, 0)$ and $x = (\sin\theta, \cos\theta, 0, \ldots, 0)$ with $\theta \in [-\frac{\pi}{2}, -\psi) \cup [\psi, \frac{\pi}{2}]$.

Consider any $z \in \mathcal{X} \cap \mathcal{R}_r(x)$. We have:

$$\sum_{i=1}^{d} z_i^2 = 1,$$

$$(z_1 - \sin\theta)^2 + (z_2 - \cos\theta)^2 + \sum_{i=3}^{d} z_i^2 \leq r^2,$$

holding simultaneously. Combining the above two equations, we get

$$\sin\theta z_1 + \cos\theta z_2 \geq 1 - \frac{r^2}{2} = \cos\psi.$$

We now consider two cases of θ:

1. $\theta \in [\psi, \frac{\pi}{2}]$. In this case, $\cos\theta \leq \cos\psi$. And so, $\sin\theta z_1 \geq \cos\psi - \cos\theta z_2 \geq 0$. Therefore, for all $z \in \mathcal{R}_r(x)$, $\sin\theta \cdot z_1 \geq 0$ and hence $z_1 \geq 0$. In this case, $\mathrm{sign}(\langle w^*, x\rangle) = \mathrm{sign}(\sin\theta) = 1 = \mathrm{sign}(z_1) = \mathrm{sign}(\langle w^*, z\rangle)$.
2. $\theta \in [-\frac{\pi}{2}, -\psi)$. In this case, $\cos\theta < \cos\psi$. And so, $\sin\theta z_1 \geq \cos\psi - \cos\theta z_2 > 0$. Therefore, for all $z \in \mathcal{R}_r(x)$, $\sin\theta \cdot z_1 > 0$ and hence $z_1 < 0$. In conclusion, $\mathrm{sign}(\langle w^*, x\rangle) = \mathrm{sign}(\sin\theta) = -1 = \mathrm{sign}(z_1) = \mathrm{sign}(\langle w^*, z\rangle)$.

In either case, $\mathrm{sign}(\langle w^*, x\rangle) = \mathrm{sign}(\langle w^*, z\rangle)$ holds for all $z \in \mathcal{R}_r(x)$, which contradicts the assumption that $\exists x' \in \mathcal{R}_r(x) \centerdot \mathrm{sign}(\langle w^*, x'\rangle) \neq \mathrm{sign}(\langle w^*, x\rangle)$. This concludes the proof. □

Recall that we define $\Lambda_\alpha(x) = \mathbb{1}(\left|\langle w^*, x\rangle\right| > \alpha)$ and assume a uniform prior over homogeneous linear model class $\mathcal{H}$ and that $\mathcal{X}$ is the origin-centered unit sphere in $\mathbb{R}^d$. With this, we show that the trend of monotonicity exists in this "nice" setting and we can also develop direct upper bounds on Π.

To do this, we first begin by characterizing the version space,

Lemma 28 (Restatement of Lemma 27)**.** *Fix $\alpha \in [0, 1)$. Recall that $\mathcal{H}_C = \{h \in \mathcal{H} \mid h(x') = h^*(x'),\ \forall x' \in \mathcal{E}_{h^*}(\mathcal{X}, \alpha)\}$ is the version space induced by explanation $\mathcal{E}_{h^*}(\mathcal{X}, \alpha)$. $\mathcal{H}_C$ can be equivalently written as:*

$$\mathcal{H}_C = \left\{h_w \mid \|w\|_2 = 1, w \cdot w^* \geq \sqrt{1-\alpha^2}\right\}.$$

Proof. First observe that $w^* \in \mathcal{E}_{h^*}(\mathcal{X}, \alpha)$. We will show

$$(\forall x \in \mathcal{E}_{h^*}(\mathcal{X}, \alpha) \centerdot \mathrm{sign}(\langle w, x\rangle) = \mathrm{sign}(\langle w^*, x\rangle)) \iff \langle w, w^*\rangle \geq \sqrt{1-\alpha^2}.$$

We show the implications in both directions:

($\Rightarrow$) First, since $w^* \in \mathcal{E}_{h^*}(\mathcal{X}, \alpha)$, we must have $\langle w, w^*\rangle \geq 0$.

Assume towards contradiction that $\langle w, w^*\rangle < \sqrt{1-\alpha^2}$, then w can be represented as $w = \sqrt{1-\beta^2} w^\star + \beta w_\perp$, where $\beta > \alpha$ and $w_\perp$ is a unit vector perpendicular to $w^\star$. We now show that there is an $x_0 \in \mathcal{E}_{h^*}(\mathcal{X}, \alpha)$ such that $\mathrm{sign}(\langle w^*, x_0\rangle) \neq \mathrm{sign}(\langle w, x_0\rangle)$, which will reach contradiction.

Choose $\gamma \in (\alpha, \beta)$, and define $x_0 = \gamma w^\star - \sqrt{1-\gamma^2} w_\perp$. It can be readily checked that $\langle w^*, x_0\rangle = \gamma > \alpha$, so $x_0 \in \mathcal{E}_{h^*}(\mathcal{X}, \alpha)$. Meanwhile, because $\gamma < \beta$,

$$\langle w, x_0\rangle = \sqrt{1-\beta^2}\gamma - \beta\sqrt{1-\gamma^2} = \sqrt{1-\beta^2}\gamma\left(1 - \frac{\beta}{\gamma} \cdot \frac{\sqrt{1-\gamma^2}}{\sqrt{1-\beta^2}}\right) < 0,$$

implying $\mathrm{sign}(\langle w, x_0\rangle) = -1 \neq 1 = \mathrm{sign}(\langle w^*, x_0\rangle)$.

(⇐) If $\langle w, w^* \rangle \geq \sqrt{1-\alpha^2}$, then w can be represented as $w = \sqrt{1-\beta^2} w^\star + \beta w_\perp$, where $\beta \leq \alpha$ and $w_\perp$ is a unit vector perpendicular to $w^\star$.

Now consider any $x \in \mathcal{E}_{h^*}(\mathcal{X}, \alpha)$; we would like to show that $\text{sign}(\langle w^*, x_0 \rangle) = \text{sign}(\langle w, x_0 \rangle)$. First, since $x \in \mathcal{E}_{h^*}(\mathcal{X}, \alpha)$, x can be represented as $x = \xi w^\star + \sqrt{1-\xi^2} x_\perp$, where $\xi \in [-1, -\alpha) \cup (\alpha, +1]$ and $x_\perp$ is a unit vector perpendicular to $w^\star$.

Without loss of generality, assume that $\xi \in (\alpha, +1]$; the case of $\xi \in [-1, \alpha)$ is symmetric. In this case, we have $\text{sign}(\langle w^\star, x \rangle) = 1$. Meanwhile,

$$\begin{aligned}
\langle w, x \rangle =& \langle \sqrt{1-\beta^2} w^\star + \beta w_\perp, \xi w^\star + \sqrt{1-\xi^2} x_\perp \rangle \\
=& \sqrt{1-\beta^2} \xi + \beta \sqrt{1-\xi^2} \langle w_\perp, x_\perp \rangle \\
\geq& \sqrt{1-\beta^2} \xi - \beta \sqrt{1-\xi^2} \\
=& \sqrt{1-\beta^2} \xi (1 - \frac{\beta}{\xi} \cdot \frac{\sqrt{1-\xi^2}}{\sqrt{1-\beta^2}}) > 0,
\end{aligned}$$

where the first inequality is by Cauchy-Schwarz; the second inequality uses the observation that $\beta \leq \alpha < \xi$. The above implies that $\text{sign}(\langle w, x \rangle) = 1 = \text{sign}(\langle w^\star, x \rangle)$. □

It is clear that increasing margin thickness ψ leads to a strictly bigger margin region, and a higher $\max_{(x,x') \in \mathcal{M}_r(\mathcal{X})} \pi_\alpha(x, x')$. We derive an analytical form of this.

Theorem 14 (Restatement of Theorem 11). $\max_{(x,x') \in \mathcal{M}_r(\mathcal{X})} \pi_\alpha(x, x')$ *can be written as:*

$$\max_{(x,x') \in \mathcal{M}_r(\mathcal{X})} \pi_\alpha(x, x') = \begin{cases} \frac{\int_0^{\psi/2} F(\theta) d\theta}{\int_0^{\phi} F(\theta) d\theta} & \psi \leq 2\phi \\ 1 & \psi > 2\phi, \end{cases}$$

where $F(\theta) = (1 - \frac{\cos^2 \phi}{\cos^2 \theta})^{(d-2)/2}$*; therefore, it is strictly increasing for* ψ *in* $[0, 2\phi]$.

Proof. Denote by $F_+(\theta) = (1 - \frac{\cos^2 \phi}{\cos^2 \theta})_+^{(d-2)/2}$, where $(z)_+ := \max(z, 0)$. Note that $F_+(\theta) = 0$ if $\theta \notin [-\phi, \phi]$.

To show the theorem statement, note that $\int_{-\pi}^{\pi} F_+(\theta) d\theta = 2 \int_0^{\phi} F(\theta) d\theta$; it therefore suffices to show that,

$$\max_{(x,x') \in \mathcal{M}_r(\mathcal{X})} \pi_\alpha(x, x') = \frac{\int_{-\psi/2}^{\psi/2} F_+(\theta) d\theta}{\int_{-\pi}^{\pi} F_+(\theta) d\theta}$$

We show the left hand side is both at most and at least the right hand side, respectively. Without loss of generality, let $w^* = (1, 0, \ldots, 0)$.

1. LHS ≥ RHS: We choose $x' = (\sin \frac{\psi}{2}, \cos \frac{\psi}{2}, 0, \ldots, 0)$, $x = (-\sin \frac{\psi}{2}, \cos \frac{\psi}{2}, 0, \ldots, 0)$. It can be seen that $\|x - x'\|_2 = 2 \sin \frac{\psi}{2} = r$, and $\langle w^*, x' \rangle > 0$, $\langle w^*, x \rangle < 0$, and therefore (x, x') is indeed a boundary pair (i.e. in $\mathcal{M}_r(\mathcal{X})$).
   In addition, for $w = (w_1, w_2)$, denote by $\phi(w) \in (-\pi, \pi]$ its polar angle with respect to $(1, 0)$ (so that $\phi((1, 0)) = 0$).

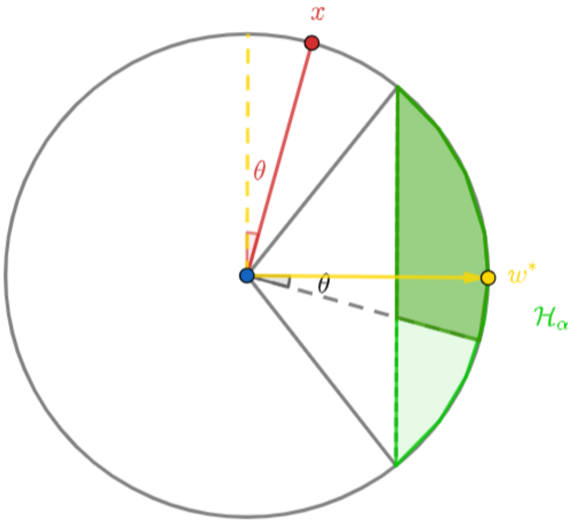


Figure 3.7: An illustration of $\Pr_{h_w \sim \mathcal{U}(\mathcal{H}_C)}(\langle w, x' \rangle \geq 0)$ in the proof of Theorem 11. Suppose x (red dot) has angle $\frac{\pi}{2} - \theta$ with w^*, and we project $\mathcal{U}(\mathcal{H}_C)$ to the 2-dimensional plane spanned by w^* and x; $\mathcal{U}(\mathcal{H}_C)$ (after projection) is supported on the green circle segment (the union of the dark and light green regions), whereas the subset $\left\{h_w \in \mathcal{H}_C : \langle w, x' \rangle \geq 0\right\}$ corresponds to the dark green region.

In this case, by Claim 1 given below (see also Figure 3.7 for an illustration), we have:

$$\begin{aligned}\Pr_{h_w \sim \mathcal{U}(\mathcal{H}_C)}(\langle w, x' \rangle \geq 0) =& \Pr_{h_w \sim \mathcal{U}(\mathcal{H}_C)}(\phi(w) \in [-\psi/2, \pi/2]) \\ =& \frac{\int_{-\psi/2}^{\pi/2} F_+(\theta) d\theta}{\int_{-\pi}^{\pi} F_+(\theta) d\theta},\end{aligned}$$

and

$$\begin{aligned}\Pr_{h_w \sim \mathcal{U}(\mathcal{H}_C)}(\langle w, x \rangle \geq 0) =& \Pr_{h_w \sim \mathcal{U}(\mathcal{H}_C)}(\phi(w) \in [\psi/2, \pi/2]) \\ =& \frac{\int_{\psi/2}^{\pi/2} F_+(\theta) d\theta}{\int_{-\pi}^{\pi} F_+(\theta) d\theta},\end{aligned}$$

and therefore,

$$\max_{(x,x') \in \mathcal{M}_r(\mathcal{X})} \pi_\alpha(x, x') \geq \Pr_{h_w \sim \mathcal{U}(\mathcal{H}_C)}(\langle w, x' \rangle \geq 0) - \Pr_{h_w \sim \mathcal{U}(\mathcal{H}_C)}(\langle w, x \rangle \geq 0) = \frac{\int_{-\psi/2}^{\psi/2} F_+(\theta) d\theta}{\int_{-\pi}^{\pi} F_+(\theta) d\theta}$$

2. LHS $\leq$ RHS: First, for every $z \in \mathbb{R}^d$, denote by $\theta(w^*, z) = \arccos(\frac{\langle w^*, z \rangle}{\|w^*\| \|z\|}) \in [0, \pi]$ the angle between z and w^*.

$$\Pr_{h_w \sim \mathcal{U}(\mathcal{H}_C)}(\langle w, z \rangle \geq 0) = \frac{\int_{\theta(w,z) - \frac{\pi}{2}}^{\frac{\pi}{2}} F_+(\theta) d\theta}{\int_{-\pi}^{\pi} F_+(\theta) d\theta}$$

To see this, without loss of generality, let $z = (z_1, z_2, 0, \dots, 0)$. Then, by Claim 1 (given below), we have

$$\Pr_{h_w \sim \mathcal{U}(\mathcal{H}_C)}(\langle w, z \rangle \geq 0) = \Pr_{h_w \sim \mathcal{U}(\mathcal{H}_C)}\left(\phi((z_1, z_2)) \in [\theta(w, z) - \frac{\pi}{2}, \frac{\pi}{2}]\right) = \frac{\int_{\theta(w,z)-\frac{\pi}{2}}^{\frac{\pi}{2}} F_+(\theta)d\theta}{\int_{-\pi}^{\pi} F_+(\theta)d\theta}.$$

Therefore, for every $(x, x') \in \mathcal{M}_r(\mathcal{X})$,

$$\begin{aligned}
&\Pr_{h_w \sim \mathcal{U}(\mathcal{H}_C)}(\langle w, x' \rangle \geq 0) - \Pr_{h_w \sim \mathcal{U}(\mathcal{H}_C)}(\langle w, x \rangle \geq 0) \\
=&\frac{\int_{\theta(w,x')-\frac{\pi}{2}}^{\theta(w,x)-\frac{\pi}{2}} F_+(\theta)d\theta}{\int_{-\pi}^{\pi} F_+(\theta)d\theta} \\
\leq&\frac{\max\left\{\int_a^b F_+(\theta)d\theta : b - a \leq \psi\right\}}{\int_{-\pi}^{\pi} F_+(\theta)d\theta},
\end{aligned}$$

where the inequality follows by observing $\theta(w, x) - \theta(w, x') \leq \theta(x, x') \leq \psi$, which follows from $2 \sin \frac{\theta(x,x')}{2} = \|x - x'\| \leq r = 2 \sin \frac{\psi}{2}$ and that $\psi/2$ is acute by definition, which means that $\theta(x, x')/2 \leq \psi/2$ and $\psi/2$ are both acute. It suffices to show that for every a, b such that $b - a \leq \psi$,

$$\int_a^b F_+(\theta)d\theta \leq \int_{-\psi/2}^{\psi/2} F_+(\theta)d\theta. \tag{3.1}$$

As $F_+(\theta) \geq 0$ for any $\theta \in \mathbb{R}$, the max must be achieved at $b - a = \psi$ and so it suffices to show $\forall c$,

$$\int_{c-\psi/2}^{c+\psi/2} F_+(\theta)d\theta \leq \int_{-\psi/2}^{\psi/2} F_+(\theta)d\theta.$$

Let $F(c) = \int_{c-\psi/2}^{c+\psi/2} F_+(\theta)d\theta$; it can be seen that $F'(a) = F_+(c + \psi/2) - F_+(c - \psi/2)$. Therefore,

$$F'(c) \begin{cases} \geq 0 & c \leq -\psi/2 \\ \geq 0 & -\psi/2 \leq c \leq 0 \\ \leq 0 & 0 \leq c \leq \psi/2 \\ \leq 0 & c \geq \psi/2, \end{cases}$$

and hence $\max_{c \in \mathbb{R}} F(c) = F(0) = \int_{-\psi/2}^{+\psi/2} F_+(\theta)d\theta$, which concludes the proof of Equation (3.1), and concludes that LHS $\leq$ RHS. □

Fact 2. *The probability density function of the uniform distribution over unit sphere projected onto the first two dimensions is*

$$p(w_1, w_2) = \frac{d-2}{2\pi}(1 - w_1^2 - w_2^2)^{\frac{d-4}{2}}.$$

Claim 1. *In the notation of the proof of Theorem 11 above, for every $a < b$ such that $[a,b] \subset (-\pi,\pi]$,*

$$\Pr_{h_w \sim \mathcal{U}(\mathcal{H}_C)}\left(\phi((w_1,w_2)) \in [a,b]\right) = \frac{\int_a^b F_+(\theta)d\theta}{\int_{-\pi}^{\pi} F_+(\theta)d\theta}$$

Proof. Recall Lemma 27 that characterizes $\mathcal{H}_C$ (see also Figure 3.2), we have:

$$\Pr_{h_w \sim \mathcal{U}(\mathcal{H}_C)}\left(\phi((w_1,w_2)) \in [a,b]\right) = \frac{\Pr_{h_w \sim \mathcal{U}}\left(w_1 \geq \sqrt{1-\alpha^2}, \phi((w_1,w_2)) \in [a,b]\right)}{\Pr_{h_w \sim \mathcal{U}}\left(w_1 \geq \sqrt{1-\alpha^2}\right)}$$

From Fact 2 above, we can express the numerator and the denominator in integral form. For the denominator, by changing of variables to the polar coordinates,

$$\begin{aligned}
&\Pr_{h_w \sim \mathcal{U}}\left(w_1 \geq \sqrt{1-\alpha^2}\right) \\
=&\int_{-\phi}^{\phi}\left(\int_{\frac{\cos\phi}{\cos\theta}}^{1} \frac{d-2}{2\pi}(1-r^2)^{\frac{d-4}{2}} r dr\right) d\theta \\
=&\frac{1}{2\pi}\int_{-\phi}^{\phi}\left(1 - \frac{\cos^2\phi}{\cos^2\theta}\right)^{\frac{d-2}{2}} d\theta \\
=&\frac{1}{2\pi}\int_{-\pi}^{\pi} F_+(\theta)d\theta.
\end{aligned}$$

For the numerator,

$$\begin{aligned}
&\Pr_{h_w \sim \mathcal{U}}\left(w_1 \geq \sqrt{1-\alpha^2}, \phi((w_1,w_2)) \in [a,b]\right) \\
=&\int_{\max(-\phi,a)}^{\min(\phi,b)}\left(\int_{\frac{\cos\phi}{\cos\theta}}^{1} \frac{d-2}{2\pi}(1-r^2)^{\frac{d-4}{2}} r dr\right) d\theta \\
=&\frac{1}{2\pi}\int_{\max(-\phi,a)}^{\min(\phi,b)}\left(1 - \frac{\cos^2\phi}{\cos^2\theta}\right)^{\frac{d-2}{2}} d\theta \\
=&\frac{1}{2\pi}\int_{a}^{b} F_+(\theta)d\theta.
\end{aligned}$$

The lemma follows by combining two equalities above. □

Theorem 15 (Restatement of Theorem 12). *$\Pi(\alpha)$ is decreasing in α, for $\alpha \in [0,1)$, and is strictly decreasing in $[\sin(\psi/2), 1)$.*

Proof. Consider $\Pi(\alpha)$ for $\alpha = \sin\phi \in [\sin(\psi/2), 1]$, which, from the proof of Theorem 11, has the following form:

$$
\begin{aligned}
\Pi(\alpha) =& \Pr_{h_w \sim \mathcal{U}(\mathcal{H}_C)} \left(\phi(w) \in [-\psi/2, \psi/2]\right) \\
=& \frac{\Pr_{h_w \sim \mathcal{U}} \left(w_1 \geq \sqrt{1-\alpha^2}, \phi((w_1, w_2)) \in [-\psi/2, \psi/2]\right)}{\Pr_{h_w \sim \mathcal{U}} \left(w_1 \geq \sqrt{1-\alpha^2}\right)} \\
=& \frac{\int_{\sqrt{1-\alpha^2}}^{1} \left(\int_0^{w_1 \tan\psi} p(w_1, w_2) dw_2\right) dw_1}{\int_{\sqrt{1-\alpha^2}}^{1} \left(\int_0^{\sqrt{1-w_1^2}} p(w_1, w_2) dw_2\right) dw_1},
\end{aligned}
$$

where $p(w_1, w_2) = \frac{d-2}{2\pi}(1 - w_1^2 - w_2^2)^{(d-4)/2}$ is the pdf of (w_1, w_2) when $h_w \sim \mathcal{U}$ (Fact 2).

Consider $f(w_1) = \int_0^{w_1 \tan\psi} p(w_1, w_2) dw_2$, and $g(w_1) = \int_0^{\sqrt{1-w_1^2}} p(w_1, w_2) dw_2$, and $F(t) = \frac{\int_t^1 f(w_1) dw_1}{\int_t^1 g(w_1) dw_1}$. ;with this, $\Pi(\alpha) = F(\sqrt{1-\alpha^2})$. It suffices to show that $F(t)$ is monotonically increasing, i.e. $F'(t) \geq 0$ for all t.

To show this, first observe that $\frac{f(w_1)}{g(w_1)}$ is monotonically increasing: indeed,

$$
\frac{f(w_1)}{g(w_1)} = \frac{\int_0^{\frac{w_1 \tan\psi}{\sqrt{1-w_1^2}}} (1-v^2)^{\frac{d-4}{2}} dv}{\int_0^1 (1-v^2)^{\frac{d-4}{2}} dv},
$$

which is increasing in w_1. As a consequence,

$$
\int_t^1 f(w_1) dw_1 = \int_t^1 g(w_1) \cdot (\frac{f(w_1)}{g(w_1)}) dw_1 \geq \frac{f(t)}{g(t)} \cdot \int_t^1 g(w_1) dw_1 \tag{3.2}
$$

Therefore,

$$
F'(t) = \frac{-f(t) \int_t^1 g(w_1) dw_1 + g(t) \int_t^1 f(w_1) dw_1}{(\int_t^1 g(w_1) dw_1)^2} \geq 0,
$$

where the last inequality is from Equation (3.2). □

Below, we derive bounds on $\Pi(\alpha)$ given specific assumptions on ϕ and ψ.

Theorem 16 (Refined version of Theorem 13)**.** *We have the following:*

1. *If* $\cos\phi \leq \frac{1}{2d^{1/4}}$*, then* $\Pi(\alpha) \leq 6 \cdot \left(\psi(1 + d^{\frac{1}{2}} \cos\phi)\right)$.
2. *For any* $c_1, c_2 > 0$*, there exists* $c_3 > 0$ *such that the following holds: given any* $\phi \in \left[c_1, \frac{\pi}{2}\right)$*, and*

$$
\psi \geq c_3 \max\left(\cos\phi, \frac{1}{d^{\frac{1}{2}} \cos\phi} \sqrt{\ln\frac{4}{c_2} + \ln\left(1 + \frac{1}{d^{\frac{1}{2}} \cos\phi}\right)}\right), \tag{3.3}
$$

then $\Pi(\alpha) \geq 1 - c_2$.

Before presenting the proof of Theorem 16, we first show how it concludes the proof of Theorem 13.

Proof of Theorem 13. We show the two items respectively.

1. Recall that $\alpha = \sin\phi$. If $\alpha \geq 1 - \frac{1}{8d}$, then $\cos^2\phi = 1 - \alpha^2 \leq \frac{1}{4d}$, implying that $\cos\phi \leq \frac{1}{2\sqrt{d}}$. As $\frac{1}{2\sqrt{d}} \leq \frac{1}{2d^{1/4}}$, the conditions of item 1 of Theorem 16 is satisfied. As a result,
$$\Pi(\alpha) \leq 6 \cdot \left(\psi(1 + d^{\frac{1}{2}}\cos\phi)\right) \leq 9\psi.$$

2. Let $C_1 \in (0, 1)$. Choose $\phi' := \arccos(\frac{1}{d^{1/4}})$. Note that $\phi' \geq \phi$, since $1 - \cos^2\phi = \alpha^2 = (1 - \frac{1}{\sqrt{d}})^2 \leq 1 - \frac{1}{\sqrt{d}} = 1 - \cos^2\phi'$. Denote by $\alpha := \sin\phi$ and $\alpha' := \sin\phi'$; we have $\alpha' \geq \alpha$. In addition, as $\phi' = \arccos(\frac{1}{d^{1/4}})$, there exists some numerical constant $c_1 > 0$ such that $\phi' \geq c_1$. Now, by item 2 of Theorem 16, there exists some $c_3 > 0$, such that when $\psi \geq \frac{c_3\sqrt{\ln\frac{8}{C_1}}}{d^{1/4}} \geq c_3 \max\left(\frac{1}{d^{1/4}}, \frac{\sqrt{\ln\frac{4}{C_1} + \ln\left(1 + \frac{1}{d^{1/4}}\right)}}{d^{1/4}}\right)$, $\Pi(\alpha') \geq 1 - C_1$. Now, as $\Pi(\cdot)$ is monotonically decreasing in α, $\Pi(\alpha) \geq \Pi(\alpha') \geq 1 - C_1$. Therefore, the theorem statement holds with $C_2 = c_3\sqrt{\ln\frac{8}{C_1}}$. □

We now present the proof of Theorem 16.

Proof. Recall that
$$\Pi(\alpha) = \begin{cases} \frac{\int_0^{\psi/2} F(\theta)d\theta}{\int_0^{\phi} F(\theta)d\theta}, & \arcsin\alpha = \phi \geq \psi/2 \\ 1, & \arcsin\alpha = \phi < \psi/2. \end{cases}$$

1. First we note that $\cos\phi \leq \frac{1}{2d^{1/4}}$ implies that $\phi \geq \frac{\pi}{3}$.
If $\phi \leq \psi/2$, then $\psi \geq \frac{2}{3}\pi$. Therefore, $\Pi(\alpha) = 1 \leq 6\psi \leq 6 \cdot \left(\psi(1 + d^{\frac{1}{2}}\cos\phi)\right)$ holds.
For the rest of the proof, we focus on the case of $\phi > \psi/2$. In this case, $\Pi(\alpha)$ equals the integral ratio $\frac{\int_0^{\psi/2} F(\theta)d\theta}{\int_0^{\phi} F(\theta)d\theta}$. With foresight, define $\theta' = \min\left(\frac{\phi}{2}, \arctan(\frac{1}{d^{\frac{1}{2}}\cos\phi}), \arccos(d^{\frac{1}{4}}\cos\phi)\right)$.
As we will see below, this is a "critical threshold" of the integral $\int_0^{\phi} F(\theta)d\theta$, in the sense that the contribution of $[\theta', \psi]$ to the integral is negligible.
By our assumption that $\cos\phi \leq \frac{1}{2d^{\frac{1}{4}}}$, $\arccos(d^{\frac{1}{4}}\cos\phi) \geq \frac{\pi}{3}$. In addition, $\arctan(\frac{1}{d^{\frac{1}{2}}\cos\phi}) \geq \min\left(\frac{\pi}{4}, \frac{1}{2d^{\frac{1}{2}}\cos\phi}\right)$ by Lemma 32 given after the proof. Moreover, recall that $\phi \geq \frac{\pi}{3}$.
Combining the above bounds, $\theta' \geq \min\left(\frac{\pi}{6}, \frac{1}{2d^{\frac{1}{2}}\cos\phi}\right)$.
We now upper bound $\Pi(\alpha)$. First we upper bound the numerator:
$$\int_0^{\psi/2} F(\theta)d\theta \leq \psi/2 \cdot F(0) = \frac{\psi}{2}(1 - \cos^2\phi)^{\frac{d-2}{2}} \leq \frac{\psi}{2}\exp\left(-\frac{d-2}{2}\cos^2\phi\right).$$

We next lower bound the denominator. As $\theta' \le \frac{\phi}{2} \le \frac{\pi}{4}$ (since by definition, $\phi/2 \le \pi/2$), this implies that $\cos^2\theta' \ge \frac{1}{2}$ and hence $\phi \ge \pi/3 \Rightarrow \frac{\cos^2\phi}{\cos^2\theta'} \in [0, \frac{1}{2}]$. Therefore,

$$\int_0^{\phi} F(\theta)d\theta \ge \int_0^{\theta'} F(\theta)d\theta \ge \theta' F(\theta') = \theta' \left(1 - \frac{\cos^2\phi}{\cos^2\theta'}\right)^{\frac{d-2}{2}} \ge \theta' \exp\left(-\frac{d-2}{2}\left(\frac{\cos^2\phi}{\cos^2\theta'} + \frac{\cos^4\phi}{\cos^4\theta'}\right)\right)$$

where the last inequality uses the elementary fact that $1 - x \ge \exp(-x - x^2)$ for $x \in [0, \frac{1}{2}]$. Combining the upper and lower bounds, we get that the integral ratio is bounded by:

$$\frac{\int_0^{\psi/2} F(\theta)d\theta}{\int_0^{\phi} F(\theta)d\theta} \le \frac{\psi}{2\theta'} \exp\left(\frac{d-2}{2}\left(\cos^2\phi\tan^2\theta' + \frac{\cos^4\phi}{\cos^4\theta'}\right)\right)$$

From our choice of θ', it can be easily seen that: (1) $\cos^2\phi\tan^2\theta' \le \cos^2\phi \cdot \frac{1}{d\cos^2\phi} \le \frac{1}{d}$, and (2) $\frac{\cos^4\phi}{\cos^4\theta'} \le \frac{\cos^4\phi}{(d^{\frac{1}{4}}\cos\phi)^4} \le \frac{1}{d}$. This implies that the exponential term is at most $\exp\left(\frac{d-2}{2} \cdot \frac{2}{d}\right) \le e$.

In conclusion, we have that:

$$\frac{\int_0^{\psi/2} F(\theta)d\theta}{\int_0^{\phi} F(\theta)d\theta} \le \frac{e}{2} \cdot \frac{\psi}{\theta'} \le 6 \cdot \left(\psi(1 + d^{\frac{1}{2}}\cos\phi)\right),$$

where in the last inequality we recall that $\theta' \ge \min\left(\frac{\pi}{6}, \frac{1}{2d^{\frac{1}{2}}\cos\phi}\right)$, and use that for $A, B > 0$,$\max(A, B) \le A + B$.

2. Fix $c_1, c_2 > 0$, and let $\phi \ge c_1$.
   If $\phi \le \psi/2$, then $\Pi(\alpha) = 1 \ge 1 - c_2$ holds.
   For the rest of the proof, we focus on the case of $\phi > \psi/2$. As $\phi \ge c_1 > 0$, $\cos\phi \le \cos c_1 < 1$.
   Therefore there exists some small constant $c_5 > 0$ such that $\cos\phi \le 1 - 2c_5$; meanwhile there exists some small enough constant $c_4 < \frac{1}{4}$ such that $\cos^2(c_4\psi) \ge 1 - c_5$ since $c_4\psi \le \pi/4$; as a consequence, $\cos^2\phi/\cos^2(c_4\psi) \le \frac{1-2c_5}{1-c_5} \le 1 - c_5$. In summary, there exist some small enough constants $c_4, c_5 > 0$ (independent of ϕ), such that $c_4 < \frac{1}{4}$ and $\frac{\cos^2\phi}{\cos^2(c_4\psi)} \le 1 - c_5$.
   By Lemma 31 (deferred after the proof), there exists some constant $c_6 > 0$ (independent of ϕ) such that

$$1 - \frac{\cos^2\phi}{\cos^2(c_4\psi)} \ge \exp\left(-\left(\frac{\cos\phi}{\cos(c_4\psi)}\right)^2 - c_6\left(\frac{\cos\phi}{\cos(c_4\psi)}\right)^4\right). \tag{3.4}$$

Therefore,

$$
\begin{aligned}
\frac{\int_0^{\psi/2} F(\theta) d\theta}{\int_{\psi/2}^{\phi} F(\theta) d\theta} \geq & \frac{\int_0^{c_4\psi} F(\theta) d\theta}{\int_{\psi/2}^{\phi} F(\theta) d\theta} \\
\geq & \frac{c_4\psi \cdot F(c_4\psi)}{\phi \cdot F(\psi/2)} \\
\geq & \frac{2c_4\psi}{\pi} \cdot \frac{\left(1 - \frac{\cos^2\phi}{\cos^2(c_2\psi)}\right)^{(d-2)/2}}{\left(1 - \frac{\cos^2\phi}{\cos^2(\psi/2)}\right)^{(d-2)/2}} \\
\geq & \frac{2c_4\psi}{\pi} \cdot \frac{\exp\left(-\frac{d-2}{2}\left(\frac{\cos^2\phi}{\cos^2(c_4\psi)} + c_6\left(\frac{\cos^2\phi}{\cos^2(c_4\psi)}\right)^2\right)\right)}{\exp\left(-\frac{d-2}{2}\frac{\cos^2\phi}{\cos^2(\psi/2)}\right)} \\
= & \frac{2c_4\psi}{\pi} \cdot \exp\left(\frac{d-2}{2}\cos^2\phi\left(\frac{1}{\cos^2(\psi/2)} - \frac{1}{\cos^2(c_4\psi)} - c_6\frac{\cos^2\phi}{\cos^4(c_4\psi)}\right)\right),
\end{aligned}
$$

where the first inequality is because $c_4 \leq \frac{1}{4}$; the second inequality is because $F(\theta)$ is monotonically decreasing for $\theta \geq 0$; the third inequality follows from the definition of $F(\theta)$, and $\phi \leq \frac{\pi}{2}$; the fourth inequality is from Equation (3.4) as well as using $1 - x \leq \exp(-x)$ to upper bound the denominator; the equality is by algebra.

Observe:

$$
\begin{aligned}
\frac{1}{\cos^2(\psi/2)} - \frac{1}{\cos^2(c_4\psi)} = & \frac{\cos^2(c_4\psi) - \cos^2(\psi/2)}{\cos^2(c_4\psi) \cdot \cos^2(\psi/2)} \\
= & \frac{\sin^2(\psi/2) - \sin^2(c_4\psi)}{\cos^2(c_4\psi) \cdot \cos^2(\psi/2)} \\
= & \frac{(\sin(\psi/2) + \sin(c_4\psi))(\sin(\psi/2) - \sin(c_4\psi))}{\cos^2(c_4\psi) \cdot \cos^2\psi} \\
\geq & \frac{\frac{\psi}{2\pi} \cdot \cos(\psi/2)\frac{\psi}{4}}{\cos^2(c_4\psi) \cdot \cos^2\psi} \\
\geq & \frac{\psi^2}{8\pi}.
\end{aligned}
$$

where the first inequality uses, $\sin(\psi/2) \geq \frac{\psi}{2\pi}$, and the Lagrange mean value theorem and the choice of c_4, such that $c_4 \leq \frac{1}{4}$ so that $\sin(\psi/2) - \sin(c_4\psi) = (\psi/2 - c_4\psi)\cos\xi$ for some $\xi \in [c_4\psi, \psi/2]$, which in turn is $\geq \frac{\psi}{4}\cos(\psi/2)$; the second inequality uses that $\cos(c_4\psi) \geq \cos(\psi/2)$, and $\cos\gamma \leq 1$ for any γ.

With foresight, we will choose $c_3 \geq 16\sqrt{c_6}$, and defer the exact setting of c_3 to the next paragraph. By the assumption of lower bound on ψ (Equation (3.3)), We have $\psi \geq 16\sqrt{c_6}\cos\phi$, and therefore $\frac{\psi^2}{8\pi} \geq 8c_6\cos^2\phi$. In addition, recall that $c_4 \leq \frac{1}{4}$, $c_6\frac{\cos^2\phi}{\cos^4(c_4\psi)} \leq$

$c_6 \cdot \frac{\cos^2\phi}{\cos^4(\frac{\pi}{8})} \leq 4c_6 \cos^2\phi$. Hence,

$$\frac{1}{\cos^2\psi} - \frac{1}{\cos^2(c_2\psi)} - c_4 \frac{\cos^2\phi}{\cos^4(c_2\psi)} \geq \frac{\psi^2}{8\pi} \cdot (1 - \frac{1}{2}) \geq \frac{\psi^2}{16\pi}.$$

We would also like to set $c_3 > 0$ such that

$$\exp\left(\frac{d-2}{2}\cos^2\phi \cdot \frac{\psi^2}{16\pi}\right) \geq \frac{\pi}{c_2 c_4 \psi}, \tag{3.5}$$

because this would imply that

$$\frac{\int_0^{\psi/2} F(\theta)d\theta}{\int_{\psi/2}^{\phi} F(\theta)d\theta} \geq \frac{2c_4\psi}{\pi} \cdot \exp\left(\frac{d-2}{2}\cos^2\phi \cdot \frac{\psi^2}{16\pi}\right) \geq \frac{2}{c_2},$$

which in turn implies

$$\frac{\int_0^{\psi/2} F(\theta)d\theta}{\int_0^{\phi} F(\theta)d\theta} = \frac{1}{1 + \frac{\int_{\psi/2}^{\phi} F(\theta)d\theta}{\int_0^{\psi/2} F(\theta)d\theta}} = \frac{1}{1 + c_2/2} \geq 1 - c_2.$$

We analyze a sufficient condition for Equation (3.5) to hold:

$$\begin{aligned}
&\exp\left(\frac{d-2}{2}\cos^2\phi \cdot \frac{\psi^2}{16\pi}\right) \geq \frac{\pi}{c_2 c_4 \psi} \\
\Leftarrow\ & \frac{d-2}{2}\cos^2\phi \cdot \frac{\psi^2}{16\pi} \geq \ln\left(\frac{\pi}{c_2 c_4} \cdot \frac{1}{\psi}\right) \\
\Leftarrow\ & \psi^2 \geq \frac{96\pi}{d\cos^2\phi} \ln\left(\frac{2\pi}{c_2 c_4} \frac{1}{\psi^2}\right) \\
\Leftarrow\ & \psi^2 \geq \frac{192\pi}{d\cos^2\phi}\left(\ln\frac{8\pi}{c_2 c_4} + \ln\left(1 + \frac{96\pi}{d\cos^2\phi}\right)\right) \\
\Leftarrow\ & \psi \geq \sqrt{\frac{192\pi}{d\cos^2\phi}\left(\ln\frac{8\pi}{c_2 c_4} + \ln\left(1 + \frac{96\pi}{d\cos^2\phi}\right)\right)}
\end{aligned}$$

Therefore, choosing $c_3 = \max\left(16\sqrt{c_6}, 2, 1 + \frac{\ln(96\pi) + \ln(\frac{2\pi}{c_4})}{\ln\frac{4}{c_2}}\right)$ (which is independent of ϕ), and by algebra, it satisfies $c_3 \frac{1}{d^{\frac{1}{2}}\cos\phi}\sqrt{\ln\frac{4}{c_2} + \ln\left(1 + \frac{1}{d^{\frac{1}{2}}\cos\phi}\right)} \geq \sqrt{\frac{192\pi}{d\cos^2\phi}\left(\ln\frac{8\pi}{c_2 c_4} + \ln\left(1 + \frac{96\pi}{d\cos^2\phi}\right)\right)}$, we have that Equation (3.5) is satisfied, and therefore $\Pi(\alpha) = \frac{\int_0^{\psi/2} F(\theta)d\theta}{\int_0^{\phi} F(\theta)d\theta} \geq 1 - c_2$. □

Lemma 29. *For $a, b > 0$, $\zeta \in (0,1)$, if $a \geq 2b\left(\ln\frac{4}{\zeta} + \ln(1 + \frac{1}{b})\right)$, then $a \geq b\ln\frac{1}{\zeta a}$.*

Proof. If $a \geq 2b\left(\ln\frac{4}{\zeta} + \ln(1+\frac{1}{b})\right) = 2b\left(\ln\frac{1}{\zeta} + \ln(4+\frac{4}{b})\right)$, then $a \geq 2b\ln\frac{1}{\zeta}$ and $a \geq 2b\ln(\max(e, \frac{1}{2b}))$ hold simultaneously.

The latter condition implies that $\frac{1}{a} \leq \frac{\frac{1}{2b}}{\ln(\max(e,\frac{1}{2b}))}$. By Lemma 30, this gives $\frac{1}{a}\ln\frac{1}{a} \leq \frac{1}{2b}$, in other words, $a \geq 2b\ln\frac{1}{a}$.

Now combine this with $a \geq 2b\ln\frac{1}{\zeta}$ by taking average on both sides, we get $a \geq \frac{1}{2}(2b\ln\frac{1}{\zeta} + 2b\ln\frac{1}{a}) = b\ln\frac{1}{a\zeta}$. The lemma follows. □

Lemma 30. *For $y > 0$, and $x \leq \frac{y}{\ln(\max(e,y))}$, then $x\ln x \leq y$.*

Proof. Define $x_0 := \frac{y}{\ln(\max(e,y))}$. We first verify that $x_0 \ln x_0 \leq y$.

1. If $y \leq e$, then $x_0 = y$; in this case, $x_0 \ln x_0 = y\ln y \leq y$ holds.
2. Otherwise, $y > e$. In this case, $x_0 = \frac{y}{\ln y} \leq y$. Therefore, $x_0 \ln x_0 \leq x_0 \ln y = y$.

Now, given $x \leq x_0$, we consider two cases of x:

1. If $x \leq \frac{1}{e}$, then $x\ln x < 0 < y$ holds.
2. Otherwise, $x > \frac{1}{e}$, and since $f(x) = x\ln x$ is monotonically increasing in $(\frac{1}{e}, +\infty)$, we have that $x\ln x \leq x_0\ln x_0 \leq y$.

In summary, if $x \leq x_0$, we must have $x\ln x \leq y$. □

Lemma 31. *For any $c_5 > 0$, there exists $c_6 > 0$ such that*

$$1 - x \geq \exp(-x - c_6 x^2), \quad \forall x \in [0, 1-c_5].$$

Proof. It suffices to choose $c_6 > 0$ such that

$$-\ln(1-x) \leq x + c_6 x^2, \quad \forall x \in [0, 1-c_5].$$

By Taylor's expansion,

$$\begin{aligned} -\ln(1-x) =& x + \sum_{i=2}^{\infty}\frac{x^i}{i} \\ \leq& x + \frac{x^2}{2}\left(\sum_{i=0}^{\infty} x^i\right) \\ \leq& x + \frac{x^2}{2(1-x)}, \end{aligned}$$

therefore, it suffices to choose $c_6 = \frac{1}{2c_5}$ such that the above is at most $x + c_6x^2$ for all $x \in [0, 1-c_5]$. □

Lemma 32. *For $x \geq 0$, $\arctan(x) \geq \min(\frac{\pi}{4}, \frac{x}{2})$.*

Proof. We consider two cases:

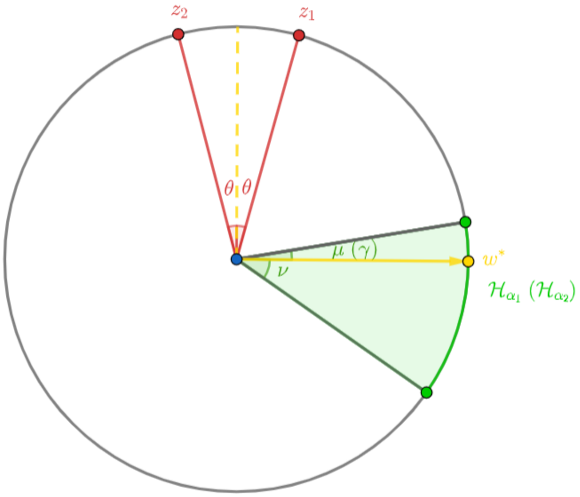


Figure 3.8: An illustration of $\mathcal{H}_{\alpha_1}$ and $\mathcal{H}_{\alpha_2}$ in the proof of Proposition 1.

1. If $x \geq 1$, $\arctan(x) \geq \frac{\pi}{4} \geq \min(\frac{\pi}{4}, \frac{x}{2})$.
2. If $x < 1$, by mean value theorem, there exists some $\xi \in [0, x]$, such that $\arctan(x) = 0 + x \cdot (\arctan(z))'\big|_{z=\xi} = \frac{x}{1+\xi^2} \geq \frac{x}{2} \geq \min(\frac{\pi}{4}, \frac{x}{2})$.

The lemma follows by combining the two cases. □

3.7.2 Deferred Proofs from Section 3.5

In this section, we provide complementary negative results to the positive results obtained under the assumptions that: 1) $\mathcal{X}$ is a sphere; and 2) $\mathcal{U}$ is the uniform distribution over $\mathcal{H}$, the class of homogeneous linear models. We show that removing one of the two conditions, i.e either allowing for non-spherical features (Proposition 1) or allowing $\mathcal{U}$ to be non-uniform over $\mathcal{H}$ (Proposition 2), leads to non-monotonicity.

Proposition 1. *Suppose $d = 2$. We have uniform prior over homogeneous linear models $\mathcal{H} = \{h_w \mid w \in \mathbb{R}^d, \|w\| = 1\}$, there exists a feature space $\mathcal{X}$ and thresholds $0 < \alpha_2 < \alpha_1$ such that $\Pi(\alpha_2) < \Pi(\alpha_1)$.*

Proof. Define $\mathcal{X} = \{x^1, x^2, x^3, z^1, z^2\}$, with the choices of x^1, x^2, x^3, z^1, z^2 specified shortly.

Let $w^\star = (1, 0)$, and therefore $h^\star((x_1, x_2)) = \text{sign}(x_1)$. Let $\theta \in (0, \frac{\pi}{4})$ be an angle. Define $z^1 = (\frac{r}{2}\sin\theta, \frac{r}{2}\cos\theta)$, $z^2 = (-\frac{r}{2}\sin\theta, \frac{r}{2}\cos\theta)$; it can be readily seen that $\|z^1 - z^2\| \leq r$ and $\text{sign}(h^\star(z^1)) = +1 \neq -1 = \text{sign}(h^\star(z^2))$; therefore $(z^1, z^2) \in \mathcal{M}_r(\mathcal{X})$. As we will see shortly, this is the only pair in $\mathcal{M}_r(\mathcal{X})$ up to reordering.

Let α_1', α_2' be such that $0 < r < \alpha_2' < \alpha_1'$, and angles γ, μ, ν be such that $\gamma < \mu < \theta < \nu$, and $\theta + \nu < \frac{\pi}{2}$. Define $x^1 = (\alpha_1', -\alpha_1' \cot\mu)$, $x^2 = (\alpha_1', \alpha_1' \cot\nu)$, and $x^3 = (\alpha_2', -\alpha_2' \cot\gamma)$. It can be seen that $h^*(x^1) = h^*(x^2) = h^*(x^3) = +1$; in addition, note that all of $\|x^1 - z^2\|$, $\|x^2 - z^2\|$, $\|x^3 - z^2\|$ are $> r$, ensuring that $\mathcal{M}_r(\mathcal{X}) = \{(z^1, z^2), (z^2, z^1)\}$.

Let $\alpha_2 = \alpha_2'/2$ and $\alpha_1 = (\alpha_1' + \alpha_2')/2$. Observe that $\{x \in \mathcal{X} : \Lambda_{\alpha_1}(x) = 1\} = \{x^1, x^2\}$, and $\{x \in \mathcal{X} : \Lambda_{\alpha_2}(x) = 1\} = \{x^1, x^2, x^3\}$.

Numerical Example. For concreteness, we can take $\alpha_1' = 10$, $\alpha_2' = 5$, $\alpha_1 = 7.5$, $\alpha_2 = 2.5$, $r = 1$, $\gamma = \frac{\pi}{16}$, $\mu = \frac{\pi}{12}$, $\theta = \frac{\pi}{8}$, and $\nu = \frac{\pi}{4}$, which satisfy all requirements above.

Given $w = (w_1, w_2) \in \mathbb{R}^2$, denote by $\phi(w) \in (-\pi, \pi]$ its polar angle with respect to $(1, 0)$ (so that $\phi((1, 0)) = 0$).

We now calculate $\Pi(\alpha_1)$. First, observe that

$$\mathcal{H}_{\alpha_1} = \big\{h \in \mathcal{H} : h(x^1) = 1, h(x^2) = 1\big\} = \big\{h_w : \|w\|_2 = 1, \phi(w) \in [-\nu, \mu]\big\}$$

Therefore,

$$\begin{aligned}
\Pi(\alpha_1) =& \max_{(x,x') \in \mathcal{M}_r(\mathcal{X})} \Big(\mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_1})}(\langle w, x\rangle \geq 0) - \mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_1})}(\langle w, x'\rangle \geq 0)\Big) \\
=& \Big|\mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_1})}(\langle w, z^1\rangle \geq 0) - \mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_1})}(\langle w, z^2\rangle \geq 0)\Big| \\
=& \left|\frac{\mu + \theta}{\mu + \nu} - 0\right| = \frac{\mu + \theta}{\mu + \nu}.
\end{aligned}$$

We now calculate $\Pi(\alpha_2)$. First observe that

$$\mathcal{H}_{\alpha_2} = \big\{h \in \mathcal{H} : h(x^1) = 1, h(x^2) = 1, h(x^3) = 1\big\} = \big\{h_w : \|w\|_2 = 1, \phi(w) \in [-\nu, \gamma]\big\}$$

Therefore,

$$\begin{aligned}
\Pi(\alpha_2) =& \max_{(x,x') \in \mathcal{M}_r(\mathcal{X})} \Big(\mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_2})}(\langle w, x\rangle \geq 0) - \mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_2})}(\langle w, x'\rangle \geq 0)\Big) \\
=& \Big|\mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_2})}(\langle w, z^1\rangle \geq 0) - \mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_2})}(\langle w, z^2\rangle \geq 0)\Big| \\
=& \left|\frac{\gamma + \theta}{\gamma + \nu} - 0\right| = \frac{\gamma + \theta}{\gamma + \nu}.
\end{aligned}$$

In conclusion,

$$\Pi(\alpha_1) = \frac{\mu + \theta}{\mu + \nu} \geq \frac{\gamma + \theta}{\gamma + \nu} = \Pi(\alpha_2).$$

Proposition 2. *Suppose $\mathcal{X}$ is the d-dimensional unit sphere with $d \geq 3$. There exists a non-uniform distribution $\mathcal{U}$ over homogeneous linear models $\mathcal{H}$, such that there exists thresholds $0 < \alpha_2 < \alpha_1$ with $\Pi(\alpha_2) < \Pi(\alpha_1)$.*

Proof. WLOG, we assume that $w^* = (1, 0, \ldots, 0)$. Define $x = (-\sin(\psi/2), \cos(\psi/2), 0, \ldots, 0)$ and $x' = (\sin(\psi/2), \cos(\psi/2), 0, \ldots, 0)$ which will be used later. It can be seen that x, x' and w^* are on the same 2-dimensional plane.

Let α_2, α_1 be such that $0 < \alpha_2 < \alpha_1 < 1$ and with $\phi_1 = \arcsin \alpha_1$ and $\phi_2 = \arcsin \alpha_2$, $\phi_1 > \phi_2 > \psi/2$. We know from Lemma 27 that

$$\mathcal{H}_{\alpha_2} = \left\{h_w : \|w\|_2 = 1, \langle w, w^*\rangle \geq \sqrt{1 - \alpha_2^2}\right\} \subset \mathcal{H}_{\alpha_1} = \left\{h_w : \|w\|_2 = 1, \langle w, w^*\rangle \geq \sqrt{1 - \alpha_1^2}\right\},$$

and that $\mathcal{H}_{\alpha_1} \backslash \mathcal{H}_{\alpha_2} = \Big\{h_w : \|w\|_2 = 1, \langle w, w^*\rangle \in [\sqrt{1 - \alpha_1^2}, \sqrt{1 - \alpha_2^2})\Big\}$.

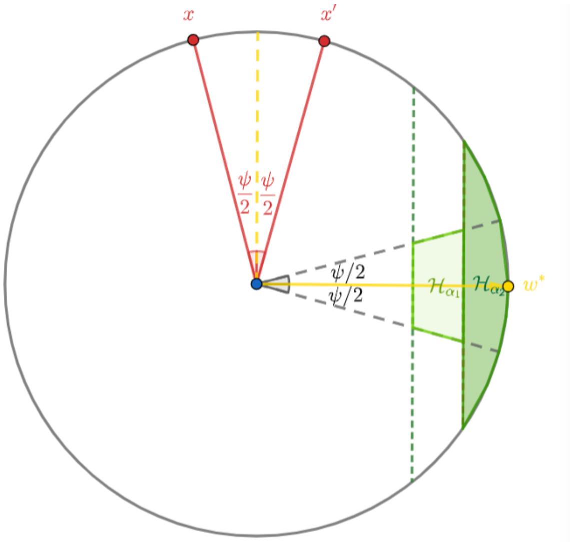


Figure 3.9: In the proof of Proposition 2, a projection of $\mathcal{U}$ onto the 2-dimensional plane spanned by w^*, x and x'; it is uniform when restricted to $\mathcal{H}_{\alpha_2}$ (the dark green region), and is concentrated in $\{h_w \in \mathcal{H}_{\alpha_1} \backslash \mathcal{H}_{\alpha_2} : -1 = \text{sign}(w \cdot x) \neq \text{sign}(w \cdot x') = +1\}$ (the light green region) when restricted to $\mathcal{H}_{\alpha_1} \setminus \mathcal{H}_{\alpha_2}$.

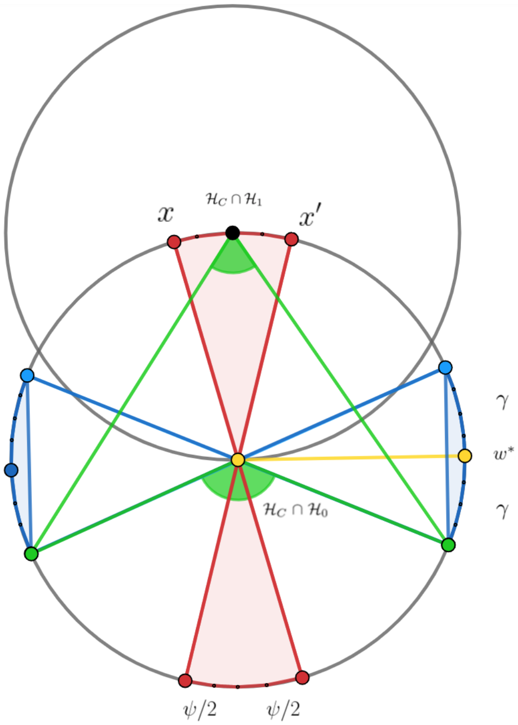
We define the density of the non-uniform prior $\mathcal{U}$ as follows. Let $\mathcal{U}$ be uniform when restricted to $\mathcal{H}_{\alpha_2}$. And let $\mathcal{U}$ have positive density that is uniform over $\{h_w : w \in \mathcal{H}_{\alpha_1} \backslash \mathcal{H}_{\alpha_2}, -1 = \text{sign}(w \cdot x) \neq \text{sign}(w \cdot x') = +1\}$; note that this is an non-empty set as it comprises of all w's whose projection onto w^* has value in $[\sqrt{1-\alpha_1^2}, \sqrt{1-\alpha_2^2}]$ and has polar angle wrt w^* in $[-\psi/2, \psi/2]$. Finally, let $\mathcal{U}$ have zero density over all other parts of $w \in \mathcal{H}_{\alpha_1} \backslash \mathcal{H}_{\alpha_2}$. The density of $\mathcal{U}$ outside $\mathcal{H}_{\alpha_1}$ can be chosen arbitrarily. See Figure 3.9 for an illustration.

By the definition of x, x', and the fact that $\mathcal{U}$ is uniform when restricted to $\mathcal{H}_{\alpha_2}$, from the proof of Theorem 11, $(x, x') \in \arg\max_{(x,x') \in \mathcal{M}_r(\mathcal{X})} \pi_{\alpha_2}(x, x')$; in other words, $\pi_{\alpha_2}(x, x') = \Pi(\alpha_2)$.

With this, we know that since $\phi_0 > \psi$, $\Pi(\alpha_2) = \pi_{\alpha_2}(x, x') < 1$. Then,

$$\begin{aligned}
\Pi(\alpha_1) &\geq \pi_{\alpha_1}(x, x') \\
&= \mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_1})}(\langle w, x' \rangle \geq 0) - \mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_1})}(\langle w, x \rangle \geq 0) \\
&= \Big(\mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_2})}(\langle w, x' \rangle \geq 0) - \mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_2})}(\langle w, x \rangle \geq 0)\Big) \mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_1})}(w \in \mathcal{H}_{\alpha_2}) + \\
&\Big(\mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_1} \backslash \mathcal{H}_{\alpha_2})}(\langle w, x' \rangle \geq 0) - \mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_1} \backslash \mathcal{H}_{\alpha_2})}(\langle w, x \rangle \geq 0)\Big) \mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_1})}(w \in \mathcal{H}_{\alpha_1} \backslash \mathcal{H}_{\alpha_2}) \\
&= \pi_{\alpha_2}(x, x') \cdot \mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_1})}(w \in \mathcal{H}_{\alpha_2}) + \mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_1})}(w \in \mathcal{H}_{\alpha_1} \backslash \mathcal{H}_{\alpha_2}) \\
&> \pi_{\alpha_2}(x, x') = \Pi(\alpha_2),
\end{aligned}$$

where the first inequality is from the definition of $\Pi(\alpha_1)$; the first equality is by the definition of $\pi(\alpha_1)$; the second equality is by the total law of probability; the third equality is by the construction that $\mathcal{U}$ has zero density in $\{h_w : w \in \mathcal{H}_{\alpha_1} \backslash \mathcal{H}_{\alpha_2}, \text{sign}(w \cdot x) = +1 \vee \text{sign}(w \cdot x') = -1\}$, so that $\mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_1} \backslash \mathcal{H}_{\alpha_2})}(\langle w, x' \rangle \geq 0) = 1$ and $\mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_1} \backslash \mathcal{H}_{\alpha_2})}(\langle w, x \rangle \geq 0) = 0$, along with the definition of $\pi_{\alpha_2}(x, x')$; the last inequality is strict because $\mathbb{P}_{h_w \sim \mathcal{U}(\mathcal{H}_{\alpha_1})}(w \in \mathcal{H}_{\alpha_1} \backslash \mathcal{H}_{\alpha_2}) > 0$ and



$\mathcal{H}_C \cap \mathcal{H}_1$
x
x'
γ
w^*
γ
$\mathcal{H}_C \cap \mathcal{H}_0$
$\psi/2$
$\psi/2$

Figure 3.10: The construction in Proposition 3. In blue are the explanations, in green are the decision boundaries of models in the version space, in red is the margin region and in yellow is w^*.

that $\Pi(\alpha_2) = \pi_{\alpha_2}(x, x') < 1$. □

Lastly, fixing assumptions 1 and 2, one may also wonder if it is possible to achieve any threshold κ in the more general, non-homogeneous linear models. We saw that this is not so asymptotically in the homogeneous case (Theorem 13). Here, we demonstrate that this does not hold in general.

Proposition 3. *There exists a class of 2-dimensional non-homogeneous linear models, with spherical $\mathcal{X}$ such that $\Pi(\alpha)$ decreases monotonically (and strictly so at some point) with increasing α, and yet $\Pi(\alpha) \geq 1/3$ for all $\alpha \in [0, 1)$ and $\psi \in (0, \pi]$.*

Proof. Let the hypothesis class of interest be $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_0$, where

$$\mathcal{H}_0 = \left\{ x \mapsto \operatorname{sign}(w_1 x_1 + w_2 x_2) : \|w\|_2 = 1 \right\}$$

is its homogeneous part, and

$$\mathcal{H}_1 = \left\{ x \mapsto \operatorname{sign}(w_1 x_1 + w_2 (x_2 - 1)) : \|w\|_2 = 1 \right\}$$

is its non-homogeneous part.

We will take same setting as before $\mathcal{X}$ is a unit circle centered at $(0, 0)$ and $\mathcal{E}_{h^*}(\mathcal{X}, \alpha) = \{x \in \mathcal{X} \mid \Lambda_\alpha(x) = 1\}$. We assume an uniform prior $\mathcal{U}$ over $\mathcal{H}$, i.e. drawing $i \sim \operatorname{Bern}(\frac{1}{2})$, and chooses a classifier uniformly at random from $\mathcal{H}_i$ induces $\mathcal{U}$.

Let $h^*(x) = x \mapsto \text{sign}(x_1)$, which is a member of $\mathcal{H}$. We consider a boundary pair $(x, x') \in \mathcal{M}_r(\mathcal{X})$ where $\|x' - x\|_2 \le r$, $h^*(x') = +1 \neq -1 = h^*(x)$.

Given $w = (w_1, w_2) \in \mathbb{R}^2$, denote by $\phi(w) \in (-\pi, \pi]$ its polar angle with respect to $(1, 0)$ (so that $\phi((1, 0)) = 0$).

Given a value of $\alpha \in [0, 1)$, the induced explanation set

$$\mathcal{E}_{h^*}(\mathcal{X}, \alpha) = \left\{x \in \mathcal{X} : \phi(x) \in [-\pi, -\pi + \gamma) \cup (-\gamma, \gamma) \cup (\pi - \gamma, \pi]\right\},$$

with $\gamma = \arccos \alpha \in (0, \frac{\pi}{2}]$.

We will examine the structure of version space $\mathcal{H}_C$ and count how much of it predicts (x, x') differently. Please refer to Figure 3.10 for an illustration. We will look at $\mathcal{H}_C \cap \mathcal{H}_1$ and $\mathcal{H}_C \cap \mathcal{H}_0$ respectively.

Part 1: $\mathcal{H}_C \cap \mathcal{H}_1$. For any $h \in \mathcal{H}_C \cap \mathcal{H}_1$, it always holds that $h(x) = +1$ and $h(x') = -1$ as long as $\gamma > 0$. This is because if the explanation is nonempty, then it includes points $(-1, 0)$ and $(1, 0)$, which enforces that any $h \in \mathcal{H}_C \cap \mathcal{H}_1$ must be a subset of $h \in \mathcal{H}_1$ with polar angle in interval $[-\pi/4, \pi/4]$ and all such h's predict (x, x') differently. More specifically,

$$\mathcal{H}_C \cap \mathcal{H}_1 = \left\{x \mapsto \text{sign}(w_1 x_1 + w_2(x_2 - 1)) : \|w\|_2 = 1, \phi(w) \in \left[-(\frac{\pi}{4} - \frac{\gamma}{2}), \frac{\pi}{4} - \frac{\gamma}{2}\right]\right\},$$

whose total arc length of $\frac{\pi}{2} - \gamma$. To summarize,

$$\mathbb{P}_{h \sim \mathcal{U}}\left(h \in \mathcal{H}_C \cap \mathcal{H}_1\right) = \frac{1}{2} \cdot \frac{\frac{\pi}{2} - \gamma}{2\pi} = \frac{\frac{\pi}{2} - \gamma}{4\pi},$$

and

$$\mathbb{P}_{h \sim \mathcal{U}(\mathcal{H}_C \cap \mathcal{H}_1)}(h(x') = +1) - \mathbb{P}_{h \sim \mathcal{U}(\mathcal{H}_C \cap \mathcal{H}_1)}(h(x) = +1) = 1.$$

Part 2: $\mathcal{H}_C \cap \mathcal{H}_0$. As we showed in Lemma 1,

$$\mathcal{H}_0 = \left\{x \mapsto \text{sign}(w_1 x_1 + w_2 x_2) : \|w\|_2 = 1, \phi(w) \in \left[-(\frac{\pi}{2} - \gamma), \frac{\pi}{2} - \gamma\right]\right\},$$

whose total arc length is $\pi - 2\gamma$.

In addition, by Theorem 11 with $d = 2$ with $\phi = \frac{\pi}{2} - \gamma$, we have

$$\max_{(x,x') \in \mathcal{M}_r(\mathcal{X})} \left(\mathbb{P}_{h \sim \mathcal{U}(\mathcal{H}_C \cap \mathcal{H}_0)}(h(x') = +1) - \mathbb{P}_{h \sim \mathcal{U}(\mathcal{H}_C \cap \mathcal{H}_0)}(h(x) = +1)\right) = \begin{cases} \frac{\psi}{2(\frac{\pi}{2} - \gamma)} & \psi \le 2(\frac{\pi}{2} - \gamma), \\ 1 & \psi > 2(\frac{\pi}{2} - \gamma). \end{cases}$$

To summarize,

$$\mathbb{P}_{h \sim \mathcal{U}}\left(h \in \mathcal{H}_C \cap \mathcal{H}_0\right) = \frac{2(\frac{\pi}{2} - \gamma)}{4\pi}$$

which is twice $\mathbb{P}_{h \sim \mathcal{U}}\left(h \in \mathcal{H}_C \cap \mathcal{H}_1\right)$ and,

$$\max_{(x,x')\in\mathcal{M}_r(\mathcal{X})} \left(\mathbb{P}_{h\sim\mathcal{U}(\mathcal{H}_C\cap\mathcal{H}_0)}(h(x')=+1) - \mathbb{P}_{h\sim\mathcal{U}(\mathcal{H}_C\cap\mathcal{H}_0)}(h(x)=+1)\right) = \min\left(1, \frac{\psi}{2(\frac{\pi}{2}-\gamma)}\right)$$

Combining the two parts, observe that $\mathbb{P}_{h\sim\mathcal{U}(\mathcal{H}_C)}(h\in\mathcal{H}_C\cap\mathcal{H}_0)=\frac{2}{3}$, and by the law of total probability,

$$\begin{aligned}
\Pi(\alpha) &= \max_{(x,x')\in\mathcal{M}_r(\mathcal{X})} \pi_\alpha(x,x') \\
&= \max_{(x,x')\in\mathcal{M}_r(\mathcal{X})} \left(\mathbb{P}_{h\sim\mathcal{U}(\mathcal{H}_C)}(h(x)=+1) - \mathbb{P}_{h\sim\mathcal{U}(\mathcal{H}_C)}(h(x')=+1)\right) \\
&= \max_{(x,x')\in\mathcal{M}_r(\mathcal{X})} \Big(\mathbb{P}_{h\sim\mathcal{U}(\mathcal{H}_C)}(h\in\mathcal{H}_C\cap\mathcal{H}_0)\cdot\left(\mathbb{P}_{h\sim\mathcal{U}(\mathcal{H}_C\cap\mathcal{H}_0)}(h(x')=+1) - \mathbb{P}_{h\sim\mathcal{U}(\mathcal{H}_C\cap\mathcal{H}_0)}(h(x)=+1)\right) \\
&\quad +\mathbb{P}_{h\sim\mathcal{U}(\mathcal{H}_C)}(h\in\mathcal{H}_C\cap\mathcal{H}_1)\cdot\left(\mathbb{P}_{h\sim\mathcal{U}(\mathcal{H}_C\cap\mathcal{H}_1)}(h(x')=+1) - \mathbb{P}_{h\sim\mathcal{U}(\mathcal{H}_C\cap\mathcal{H}_1)}(h(x)=+1)\right)\Big) \\
&= \frac{2}{3}\cdot\min\left(1, \frac{\psi}{2(\frac{\pi}{2}-\gamma)}\right) + \frac{1}{3} \\
&\geq \frac{1}{3}
\end{aligned}$$

through which we see that $\Pi(\alpha)$ is increasing in γ and strictly so for when $\pi/2-\gamma>\psi/2$. In other words, $\Pi(\alpha)$ is identically 1 for $\alpha\in[0,\sin(\psi/2)]$, and is strictly decreasing in α for $\alpha\in[\sin(\psi/2),1)$. □

3.8 Additional Experiments

3.8.1 Fair accessibility to explanations

A notable concern that may arise with margin distancing is that omission of prototypical explanations is necessary for regions close to the margin. Thus, this could disproportionately affect individuals in those regions, since they will not have their representative explanation be in the explanation set. We plot the composition of margin set in Figure 3.11 with a threshold of 0.03 for both logistic and SVM models and note that there is some disproportionate effect. Verily, this is another important factor that needs to be taken into account in the explanation generation process.

3.8.2 MMD Explanations

We include results on the trend of the three metrics under MMD-Critic explanations to further empirically trace how the boundary certainty varies with explanation omission. Similar to the MLP results under k-medoid, we see that in Figure 3.12 the trend is almost monotonic everywhere. One difference however, is that the boundary certainty does not drop off as fast as in the k-medoid setting. This suggests that the search strategy of trying small omission percentages may work with some explanation methods such as the k-medoid, but will not with others like MMD-Critic.

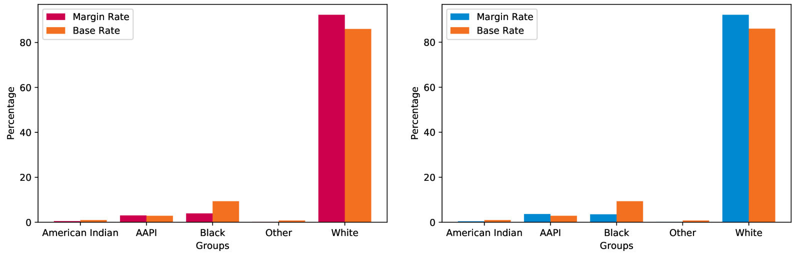


Figure 3.11: Racial composition of margin points under LR (left) and SVM (right).

Target Certainty	Binary Search	Optimal	Difference
0.036	45	10	35
0.046	45	10	35
0.055	10	10	0
0.065	10	10	0
0.075	10	10	0
0.084	10	10	0
0.094	5	5	0
0.103	5	5	0
0.113	5	5	0
0.122	5	5	0

Table 3.2: Difference table with the max metric and at $r = 0.1$

3.8.3 Effects of Larger Models

We include results on the trend of the three metrics for a two hidden-layer MLP to showcase the effects of larger models. In Figure 3.13, we see similar trends under both explanations, but with higher values across the board in comparison with the one-layer case. Again, as in the one-layer MLP case, under MMD-critic explanations, the drop in the metrics are slower than the drop under k-medoid explanations.

3.8.4 Monotonicity Tables

We present tables charting the differences between the percentage of explanations omitted calculated through binary search and the optimal percentage of explanation calculated through a left-to-right linear search, for ten, equally spaced out values of target boundary certainty corresponding to Figure 3.3 in Tables 3.2 through 3.10.

3.9 Additional Modeling Discussion

One objection with our modeling assumption could be that if it is the case that most of the $\mathcal{X}$ is in $\mathcal{M}_r(\mathcal{X})$, then margin-distancing could remove most of the representative-based explanations

Target Certainty	Binary Search	Optimal	Difference
0.071	70	15	55
0.11	45	10	35
0.15	10	10	0
0.18	10	10	0
0.22	10	10	0
0.26	5	5	0
0.30	5	5	0
0.33	5	5	0
0.37	5	5	0
0.41	5	5	0

Table 3.3: Difference table with the max metric and at $r = 0.2$

Target Certainty	Binary Search	Optimal	Difference
0.16	65	65	0
0.23	25	25	0
0.31	10	10	0
0.39	5	5	0
0.47	5	5	0
0.55	5	5	0
0.63	5	5	0
0.7	5	5	0
0.78	5	5	0
0.86	5	5	0

Table 3.4: Difference table with the max metric and at $r = 0.3$

Target Certainty	Binary Search	Optimal	Difference
0.03	45	10	35
0.04	45	10	35
0.05	10	10	0
0.06	10	10	0
0.07	10	10	0
0.08	10	10	0
0.09	5	5	0
0.1	5	5	0
0.11	5	5	0
0.12	5	5	0

Table 3.5: Difference table with the top 5 percentile average and at $r = 0.1$

Target Certainty	Binary Search	Optimal	Difference
0.05	65	15	50
0.07	40	10	30
0.1	10	10	0
0.12	10	10	0
0.14	10	10	0
0.17	5	5	0
0.19	5	5	0
0.21	5	5	0
0.24	5	5	0
0.26	5	5	0

Table 3.6: Difference table with the top 5 percentile average and at $r = 0.2$

Target Certainty	Binary Search	Optimal	Difference
0.11	65	65	0
0.17	25	25	0
0.23	10	10	0
0.3	10	10	0
0.36	5	5	0
0.42	5	5	0
0.48	5	5	0
0.54	5	5	0
0.6	5	5	0
0.66	5	5	0

Table 3.7: Difference table with the top 5 percentile average and at $r = 0.3$

Target Certainty	Binary Search	Optimal	Difference
0.008	45	15	30
0.014	45	10	35
0.019	40	10	30
0.025	10	10	0
0.031	5	5	0
0.037	5	5	0
0.042	5	5	0
0.048	5	5	0
0.054	5	5	0
0.06	5	5	0

Table 3.8: Difference table with the average and at $r = 0.1$

Target Certainty	Binary Search	Optimal	Difference
0.013	65	10	55
0.02	45	10	35
0.027	10	10	0
0.034	10	10	0
0.041	5	5	0
0.049	5	5	0
0.056	5	5	0
0.063	5	5	0
0.07	5	5	0
0.077	5	5	0

Table 3.9: Difference table with the average and at $r = 0.2$

Target Certainty	Binary Search	Optimal	Difference
0.044	65	65	0
0.075	40	30	10
0.106	10	10	0
0.137	10	10	0
0.168	5	5	0
0.199	5	5	0
0.229	5	5	0
0.26	5	5	0
0.291	5	5	0
0.322	5	5	0

Table 3.10: Difference table with the average and at $r = 0.3$

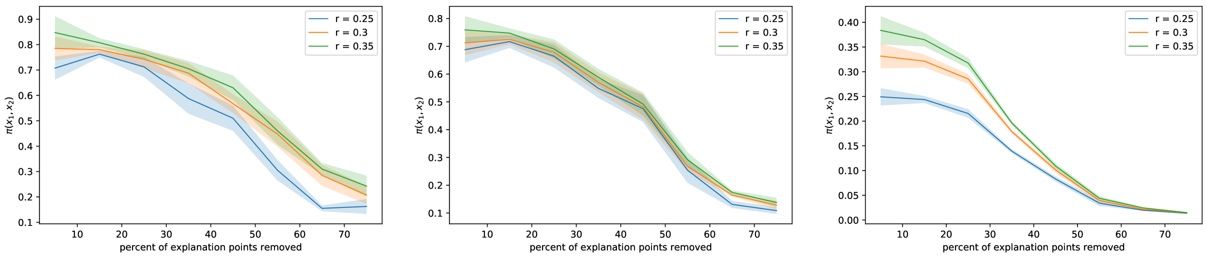


Figure 3.12: MLPs results with MMD-Critic explanations: max (left), top 5 percentile average (middle), average $\pi(x, x')$ (right). We observe similar trends as in the k-medoid case with one difference being that the drop off rate is slower in the MMD-Critic case.

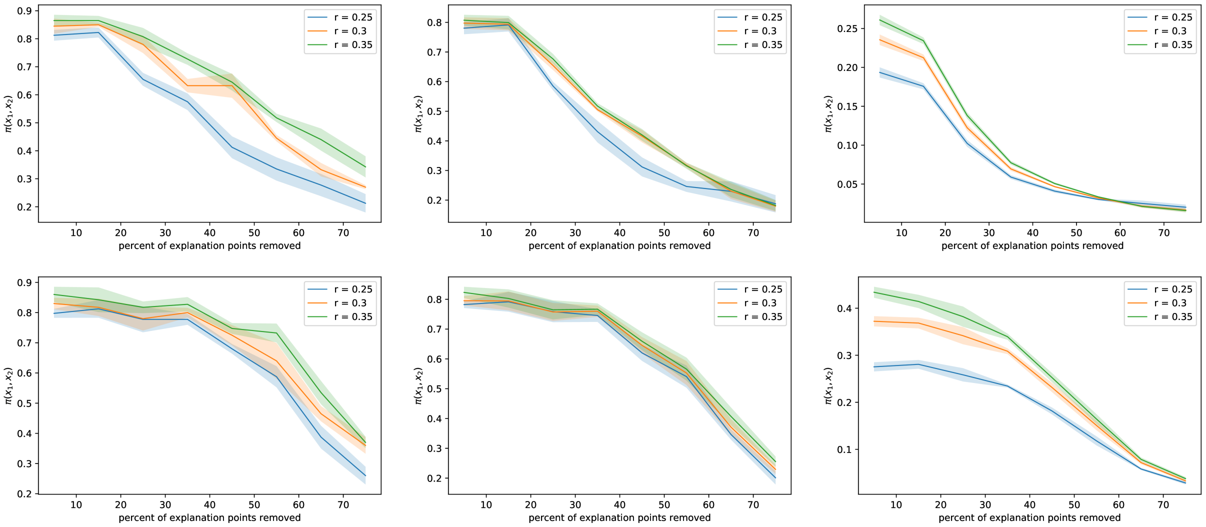


Figure 3.13: Two layer MLP results: under k-medoid explanations (top), under MMD explanations (bottom). The three metrics are in column: max (left), top 5 percentile average (middle), average $\pi(x, x')$ (right).

$\mathcal{E}(\mathcal{X})$. We assume this is not the case and that $\mathcal{M}_r(\mathcal{X})$ is only a small fraction of $\mathcal{X}$.

Indeed, this assumes that the feature collection and modeling is done well and that most points are not within r of another point with the opposite label.

3.10 Additional Related Works

Improvement vs Gaming: A crucial point about feature alteration is whether to think of it as causal (beneficial) or gaming [204]. In our setting, the organization first offers individuals transparency into how the model "works" and predicts based on the reported features. We assume individuals are not aware of the underlying causal model. Hence, we view misreporting in the first stage as gaming.

Explanation Manipulation: There has been work focusing on how organizations may manipulate an unfair model's explanation to make it look more fair than it actually is [8, 12, 263]. By contrast, we study how to provide explanations that are informative and cover as much of $\mathcal{X}$ as possible while protecting boundary points' label information.

Security of ML models: Our work is also related to model extraction literature [205, 280] that assumes one can query an API for model prediction/gradient-based explanation on any point. We view our work as a study on how to "limit" the API so as to prevent a new type of attack – individual-level gaming, which need not require the full model extraction in order to carry out the attack [153].

Model Multiplicity: The set of models consistent with labelled data is also referred to as version space [206]. Our paper thus pertains to a recent line of work highlighting the existence of the "Rashomon effect" [80, 252] or model multiplicity [196]. These papers do not focus on strategic manipulation, but study or raise the importance of developing sampling algorithms that can explore the version space.

Chapter 4

Causal Strategic Modeling

4.1 Introduction

In consequential settings, machine learning models do more than predict. They also drive decisions that impact people's lives. For example, credit scores may simultaneously serve as predictions of the likelihood of repayment and as the basis on which loans are approved. When decisions impact individuals whose features are manipulable, these individuals will be *incentivized* to intervene on their features in order to raise the model scores. Whether or not these increases in score (e.g., predicted likelihood of repayment) result in improvements in the outcome of interest (e.g., actual likelihood of repayment) depends on the *causal* relationships between the features and the outcome. Thus, this causal knowledge is crucial to designing scoring *mechanisms* that serve as both accurate predictors and beneficial incentives.

A blossoming line of research on strategic machine learning studies these incentive effects [7, 38, 52, 55, 66, 94, 113, 136, 163, 184, 201, 226, 271, 309, 319]. Hardt et al. [136] conceive of feature manipulations as gaming, putting aside the possibility that manipulations might change the outcome of interest. More recently, researchers have recognized that manipulations can causally influence the outcome interest, and seek to learn optimal scoring mechanisms for outcome improvement [163, 258]. However, most works thus far assume that the underlying causal structure is known. A notable exception is Miller et al. [204] who demonstrate that producing an optimal scoring mechanism is at least as hard as identifying the underlying causal graph. However, they do not explore how the ability to deploy mechanisms and observe the induced strategic responses can be leveraged to efficiently identify the underlying causal structure, and in turn, derive the optimal scoring mechanism.

This motivates our study of *Causal Strategic Prediction*, wherein a principal designs a predictor that also serves a *reward model* that incentivizes interventions on the part of the agent. These interventions are applied on variables (features) related by a Structural Causal Model (SCM), resulting in an overall change in the outcome of interest [224]. In this setup, the principal seeks to design a mechanism that is simultaneously a good predictor and a good reward model through multiple rounds of interaction with the agent.

In more detail, we may view feature manipulations as soft interventions on the underlying causal graph. Subject to some cost structure, individuals apply additive perturbations to variables, which influence both the value of the intervened-upon variable and all downstream variables in the graph (possibly, but not necessarily, including the outcome of interest). Capturing the causal effect of feature manipulation lies at the heart of strategic ML and our formulation thus allows us to quantify the causal effect incentivized by the predictor, distinguishing the good (improvement) from the bad (gaming).

4.1.1 The general problem of Reward Design

More generally, we can view Causal Strategic Prediction as one instance of the more general reward design problem. This problem surfaces in many principal-agent settings, wherein the principal wishes to design a reward (mechanism) that incentivizes the agent to perform the desired action.

Reward model design has long been known to be difficult, in part due to the challenge of specification. Hence, *reward hacking* can arise due to imperfectly designed rewards, which one may view as the incentivization of sub-optimal agent actions (interventions) by the reward model. The end result is thus sub-optimal improvement in the outcome of interest incentivized by the principal's inapt reward model.

A general approach in prior literature for reward design is the use of some predictive proxy of the reward model, with learned reward models being one such example that has seen sizable success recently [220]. This is indeed a natural idea. When the true reward model is too difficult or complex to specify or design, a proxy that is predictive of and highly correlated with the true reward model seems sensible as a substitute.

Yet, despite this motivation and some success, learned reward models are still known to contain defects, despite its high correlation with the true reward. Correlation does not imply causation and a common criticism of the learned reward models is that they are only *proxies* of the true reward [56, 222, 276]. Since proxies can be analyzed through the causal lens, this motivates the more general study of reward models through the lens of causality. We focus in particular on the setting where the principal/designer is unable to directly specify the reward of interest, as otherwise the agent can directly optimize the true reward specified by the principal, without concern for reward hacking.

Towards building this understanding, we consider a general, theoretical setup useful for quantitatively analyzing reward hacking. In this setup, a designer (principal) iteratively sets a (surrogate) reward model that an agent seeks to optimize in relation to the underlying world model. Using this, we can chart how well a reward model can incentivize increase in the true reward, contrasting it with how well it predicts the true reward. We answer:

- What is the relationship between predictiveness and reward hacking in learned reward models? Would the designer need to sacrifice predictiveness to reduce reward hacking?
- Can a reward model's high predictiveness belie the improvement in true reward it incentivizes?

We answer both questions and include analysis for two natural RMs: invariant causal predictor and proxy (descendants of R) reward models [231]. Through our result, we hope to challenge the common adage that the invariant causal predictor is always a sound choice for a predictor and a reward model, and that proxy rewards always make for poor reward models.

Finally, we obtain results on how to do experiment design with reward models, so as to incentivize interventions that reveal causal structure. We develop algorithms that leverage the resultant data to enable causal discovery. With these algorithms, the principal can then uncover the underlying causal mechanisms, and design more apt reward models accordingly.

4.2 Preliminaries

We consider a general setup where a designer designates a reward model (RM) optimized by the agent, with respect to the unknown, underlying world model. The interaction protocol is described as in Protocol 9.

4.2.1 World Model

First, to describe the world model, the agent acts in a world model $\mathcal{M}$ that corresponds to a causal graph consisting of endogenous nodes $(X_1, ..., X_n, R)$ and exogenous nodes $(U_1, ..., U_n, U_R)$. We assume this causal graph is acyclic. The associated DAG has a directed edge $X_i \to X_j$, if X_i is a direct cause of X_j. Let $pa(i)$ denote the indices of parent nodes of i. Each node X_i is related to its parents $X_{pa(i)}$ (used to denote $\{X_j : j \in pa(i)\}$ for brevity) by the structural equation with continuously differentiable function $g_i \in C^1$:

$$X_i = g_i(X_{pa(i)}, U_i), \forall i \in [n].$$

In the causal discovery section of this work, a general class of SCMs we consider is Additive Noise Models (ANMs) with zero-mean exogenous noise of the form: $X_i = g_i(X_{pa(i)}) + U_i$, $\mathbb{E}[U_i] = 0$ [231].

We assume that $\mathcal{M}$ satisfies causal sufficiency: there are no unobserved common causes of the endogenous nodes. And so, the exogenous noises $U_1, .., U_n, U_R$ are mutually independent. $\mathcal{D}_0$ denote the observational distribution of $\mathcal{M}$, pre-intervention.

4.2.2 Reward Model

The reward node R in $\mathcal{M}$ is the node the designer wishes to incentivize the agent to optimize. Crucially, we assume it is *only* possible for the designer to specify the reward model f in terms of observable nodes $\{X_i\}_{i=1}^n$ and not R (and nor the latent noise nodes U). The same applies for the agent, who can only observe variables $\{X_i\}_{i=1}^n$ (and not U) and intervenes on X in accordance of f and $\mathcal{M}$.

In this way, R can only be optimized through interventions on observable nodes, which captures the key difficulty with reward modeling. The designer is unable to have the agent optimize the true reward R directly, which in many cases is too difficult or abstract to specify as a reward model (e.g. alignment with humans values). The challenge then is for the designer to communicate an apt surrogate RM, as a function of observable nodes X. A natural choice for the reward model f is a model predictive of R. Here, we choose MSE as a standard measure of the predictiveness of f.

Algorithm 9 Agent Interaction Protocol

```
Designer chooses reward model $f$ from the reward model class $\mathcal{F}$.
Agent observes pre-intervention $X$ in $\mathcal{M}$ and learns to optimize $f$, eventually learning to play optimal policy $a^*(f)$ as in Equation 4.1.
Designer observes the post-intervention world model distribution $\mathcal{D}_i$ under the agent's optimal policy intervention $a^*(f)$.
```

Definition 20 (RM Predictiveness)**.** *Let the measure of the predictiveness of reward model f be the population squared loss (MSE) with respect to the observational distribution:*

$$\mathit{Risk}(f) = \mathbb{E}_{\mathcal{D}_0}[(f(X) - R)^2]$$

As such, we will analyze RMs both by its risk and its degree of reward hacking (to be defined subsequently). In this paper, we focus on both in the infinite-sample setting, deferring finite sample analysis to future works.

4.2.3 Agent Optimization of Reward Model f

For generality, we assume the agent is a Von Neumann-Morgenstern Expected Utility Maximizer [285]. During a given episode, the agent learns (through e.g. some RL algorithm) to optimize the expected reward model and learns an intervention policy mapping pre-intervention state X to intervention ($a : X \to \mathbb{R}^n$). We assume this agent's optimization algorithm is sound, in that the agent eventually learns the optimal policy $a^*(f) : X \to \mathbb{R}^n$, which is defined as follows:

$$\begin{aligned} a^*(f) = \arg\max_{a : X \to \mathbb{R}^n} \quad & \mathbb{E}_U[f(X'_1, ..., X'_n)] \\ \text{s.t.} \quad & X'_j = g_j(X'_{pa(j)}, U_j) + a_j(X) \;\; \forall j \in [n] \\ & \mathbb{E}[c(a_1(X), ..., a_n(X))] \leq b, \end{aligned} \tag{4.1}$$

Under policy $a^*(f)$, its additive soft interventions shift X_j to $X'_j(a^*(f), U) = g_j(X'_{pa(j)}, U_j) + a^*_j(f)$, defined recursively. In what follows, we will use $X'(a(X), U)$ to denote post-intervention joint distribution $X'(a(X), U) = (X'_1(a(X), U), ..., X'_n(a(X), U)$ and $R(X'(a(X), U), U) = g_R(X'_{\text{pa}(R)}(a(X), U), U_R)$ under intervention policy $a(X)$.

Finally, c is the agent's cost function and b the budget in the constrained optimization program, where we assume c is convex and strictly increasing in each coordinate ($\partial c/\partial a_i > 0$ s.t. no intervention is cost-free). We note that this generalizes the formulation of agent optimization considered in prior works on reward misspecification [327].

4.3 Reward Hacking

We are now ready to write down the causal definition of reward hacking. Reward hacking arises due to the sub-optimal choice of interventions on nodes that do not maximally increase the true reward R. We measure the extent of reward hacking of f as the fraction of the reward increase lost by optimizing the chosen reward model f instead of the optimal RM in the reward model class.

Definition 21 (Degree of Reward Hacking)**.** *For a sufficiently expressive reward model class $\mathcal{F}$ and reward model $f \in \mathcal{F}$, define the degree of reward hacking $\tau(f)$ as:*

$$\tau(f) = 1 - \frac{\mathbb{E}_U[R(X'(a^*(f), U), U)] - \mathbb{E}_U[R]}{\max_{f' \in \mathcal{F}} \mathbb{E}_U[R(X'(a^*(f'), U), U)] - \mathbb{E}_U[R]}$$

where $\mathcal{F}$ is such that $\max_{f' \in \mathcal{F}} \mathbb{E}_U[R(X'(a^*(f'), U), U)] - \mathbb{E}_U[R] > 0$ (leading to a non-zero denominator).

A high degree of reward hacking for a learned RM implies that f can be very correlated with R, but (deceptively) poor at inducing a policy that increases R in expectation.

4.3.1 Analytical Examples of Reward Model Optimization in Linear Graphs

For a concrete example of the agent's optimization problem, consider when the underlying SCM is linear. We can write $x = B_g x + u \Rightarrow x = (I - B_g)^{-1} u = Bu$. Under soft intervention a, $x' = B(u + a) = x + Ba$.

Quadratic Cost Example: Thus, under linear reward model $f(x') = w^T x'$, quadratic cost C (diagonal matrix) and $b = 1$, the agent's optimization program is as follows:

$$\begin{aligned} \max_a \quad & \mathbb{E}[w^T(X + Ba)] \\ \text{s.t.} \quad & \frac{1}{2} a^T C a \leq 1. \end{aligned}$$

Note that since the optimization objective is $w^T Ba$, the optimal intervention $a^*(w) = \frac{1}{\lambda} C^{-1} B^T w$ (with Lagrange Multiplier $\lambda = \sqrt{\frac{1}{2b} w^T B C^{-1} B^T w}$) is conveniently constant in X. This closed form solution shows that the optimal policy and thus reward increase is smooth in the reward model w.

Linear Cost Example: Under linear cost c, this corresponds to a linear objective with linear constraints.

$$\begin{aligned} \max_a \quad & \mathbb{E}[w^T(X + Ba)] \\ \text{s.t.} \quad & c^T a \leq b. \end{aligned}$$

The optimal policy is thus a corner solution, leading to sharp changes in the single-node intervention. Hence, the reward increase (and thus degree of reward hacking) is no longer smooth in w. This simple example emulates the sudden phase shift that are possible with reward hacking, as first documented in [222].

4.4 Related Works

The problem of reward hacking is one of the main challenges in AI safety [11]. Our work builds upon causality-based research that aims to address this challenge.

Theoretical Works: Theoretically, [103] is an early work that formalizes reward hacking through the lens of causal influence diagrams. They distinguish between the intended goal and the specified reward function, with the insight that reward hacking is where an inapt reward model results in a suboptimal intervention on the part of the agent. More broadly, causal incentives is a well-known formalism in AI safety for analyzing agents that are incentivized by the reward model to perform causal interventions in the world [102, 105, 290].

Empirical Works: there have been several papers that empirically demonstrate reward hacking and/or provide causality-based mitigation methods. [171, 276] studies causal confusion and how agents can exploit incorrect causal models, proposing remedies such as learning from diverse environments or causal discovery. [192] introduces a framework to mitigate undesirable causal influences in RL, by decomposing the value function into contributions from distinct causal pathways. This is so that spurious paths that contribute to reward hacking can be penalized. [264] proposes Causal Reward Adjustment (CRA) that trains sparse autoencoders to recover interpretable features from PRM activations. Using explicit causal modeling, a backdoor adjustment is used to correct for confounding semantic features that spuriously correlate with rewards. [190] trains robust reward models that disentangle prompt-driven preferences from prompt-independent artifacts, by using data augmentation techniques to eliminate spurious correlations. Finally, [289] trains causal reward models to be counterfactual invariant by remaining consistent when irrelevant variables are altered, with the latter used to explicitly target spurious correlations such as length bias and sycophancy in LLM alignment.

Our work builds on the theoretical line of work in causal incentives, with a specific focus on analyzing learned RMs. In particular, we chart the relationship between a RM's predictiveness vs reward hacking — in general SCMs. Furthermore, we develop discovery algorithms that leverage reward hacking to uncover causal structure so that a better RM can be designed. Altogether, our theoretical results aim to complement the body of causality-inspired empirical works that reduce reward hacking.

4.5 Characterization of Reward Hacking in Learned Reward Models

4.5.1 Causal Invariant Predictor

We begin by analyzing the setting when the designer has some predictive RM. A common recommendation for this RM across several works is the invariant predictor of the reward across different environments, which verily corresponds to $\mathbb{E}[R|X_{\text{pa}(R)}]$ [230]. Suppose the designer has this model, it is natural to ask: does the invariant causal predictor induce reward hacking? When R has no descendants, the following result shows that $\mathbb{E}[R|X_{\text{pa}(R)}]$ is in fact an optimal reward model, which achieves the best of both worlds.

Proposition 4 (Invariant Causal Predictor is risk-minimizing and reward-maximizing)**.** *In any causal graph where R has no descendants, $f = \mathbb{E}_U[R|X_{pa(R)}]$ is a reward model with minimal risk and zero reward hacking.*

Proof. $\mathbb{E}_U[R|X_{pa(R)}]$ **is risk-minimizing:**

This follows from that $\mathbb{E}_U[R|X]$ is the risk-minimizer. And due to no descendants of R in X, we have that $R \perp\!\!\!\perp X \setminus X_{pa(R)}|X_{pa}(R) \Rightarrow \mathbb{E}_U[R|X] = \mathbb{E}_U[R|X_{pa(R)}]$. Hence, $\mathbb{E}_U[R|X_{pa(R)}]$ is also risk-minimizing.

$\mathbb{E}_U[R|X_{pa(R)}]$ **is reward-maximizing:**

We will show that maximizing $\mathbb{E}_U[R|X_{pa(R)}] = \mathbb{E}_{U_R}[R|X_{pa(R)}]$ is the same as optimizing R in the optimization objective. The key aspect to handle is that the resulting intervention $a(X)$ could be a function of X due to the reward model.

Write $f(x) = \mathbb{E}_{U_R}[g_R(x, U_R)]$.

We will show that the optimization objective under f matches that of R under $X'_{\text{pa}(R)}(a(X), U)$ for every $a(X)$. f thus incentivizes direct optimization of R and yields maximal reward increase.

$$
\begin{aligned}
&\mathbb{E}_U[f(X'_{\text{pa}(R)}(a(X), U)] && \text{(objective under RM } f\text{)} \\
&= \mathbb{E}_{(U_R, U_{-R})}[\mathbb{E}_{U_R}[g_R(X'_{\text{pa}(R)}(a(X), U), U_R)]] && \text{(definition of } f\text{)} \\
&= \mathbb{E}_{(U_R, U_{-R})}[\mathbb{E}_{U_R}[g_R(X'_{\text{pa}(R)}(a(X), U), U_R)|X'_{\text{pa}(R)}(a(X), U)]] && (\star) \\
&= \mathbb{E}_{U_{-R}}[\mathbb{E}_{U_R}[g_R(X'_{\text{pa}(R)}(a(X), U), U_R)|U_{-R}]] \\
& && \text{(inner expectation already marginalizes over } U_R\text{)} \\
&= \mathbb{E}_{(U_R, U_{-R})}[g_R(X'_{\text{pa}(R)}(a(X), U), U_R)] && \text{(law of iterated expectations)} \\
&= \mathbb{E}_{(U_R, U_{-R})}[R(X'_{\text{pa}(R)}(a(X), U), U)]
\end{aligned}
$$

$(\star)$: $X'_{\text{pa}(R)}(a(X), U)$ is a function of U_{-R}, thus $U_R \perp\!\!\!\perp U_{-R} \Rightarrow U_R \perp\!\!\!\perp X'_{\text{pa}(R)}(a(X), U)$. Crucially, this is the step where we use that intervention $a(X)$ is a function of X, which are all upstream of R by assumption. And, $a(X)$ is a function of U_{-R} and thus independent from U_R.

□

Next, we provide the complementary negative result to the positive result above. We formalize the intuition above that descendants/proxy is useful in terms of aptly designing better reward models (dependent on U_R), which can incentivize more targeted interventions. In doing so, we show that perhaps surprisingly, if R does have descendants, the invariant causal predictor may not be the best reward model and can incentivize ineffective (but not ineffectual) interventions.

Proposition 5 (Suboptimality of Invariance Causal Predictor)**.** *For every $\epsilon > 0$, there exists a causal graph where R has descendants and $\mathcal{F}$ such that $f = \mathbb{E}_U[R|X_{pa(R)}]$ attains $1 - \epsilon$ degree of reward hacking.*

Proof. Consider the setting where the cost is quadratic and equal across the two features, $c(a) = a_1^2 + a_2^2$, and budget $b = 1$.

Now, suppose the causal graph is as follows: $X_1 \to R \to X_2$. And the SCM is such that $R = U_R X_1$, $X_2 = R$, where $X_1 \sim U[2, 3]$ and $U_R \in \{\pm 1\}$ where $P(U_R = 1) = p$. Let $p = (1 + \epsilon)/2$ ($p > 1/2$).

$\mathbb{E}[R|X_{\text{pa}(R)}] = \mathbb{E}[U_R]X_1 = (2p - 1)X_1$. This will induce an intervention of $a_1^* = 1$, and thus an expected reward increase of $p - (1 - p) = 2p - 1$.

However, the optimal reward model can in fact induce optimal reward increase of 1. If we let $\mathcal{F}$ be the class of Piecewise Linear Functions with finitely many pieces, one optimal reward model is:

$$f = \mathbb{1}\{X_2 < 0\}(-X_1) + \mathbb{1}\{X_2 \geq 0\}(X_1).$$

To see this, first note that the optimal policy's intervention will be such that $a_2^* = 0$. Since $|x_2| \geq 2$, any change to x_2 will not flip the sign and thus change the objective.

Now, since $X_2 < 0 \Leftrightarrow U_R < 0 \Leftrightarrow g_R = -X_1$, when $U_R = -1$ and we incentivize with RM $f = -X_1$, $a^* = -1$. And similarly, since f is monotonic in a_1, when $U_R = 1$ and we incentivize with RM $f = X_1$, $a^* = 1$.

The reward increase induced by this RM is thus always 1, which is optimal as the maximal change in R is 1: since $a_1^2 \leq 1$, the per episode change (for any U) from intervention on X_1 by a_1 results in reward increase of $U_R(X_1 + a_1) - R = R + U_R a_1 - R \leq 1$.

□

To further build on the characterization, in terms of risk, we show that $f = \mathbb{E}_U[R|X_{pa(R)}]$'s optimal pre-intervention risk can be illusory. There exists settings where it attains arbitrarily high risk under the post-intervention distribution.

Proposition 6. *There exists a causal graph and cost structure such that $\mathbb{E}_U[R|X_{pa(R)}]$ has arbitrarily small population MSE on the observational distribution, but arbitrarily large MSE under the post-intervention distribution.*

Finally, one may wonder: can we always expect to find a best of both worlds RM? To complement Proposition 4, we show that when R has descendants, there exists a (family of) causal graph, where no reward model is simultaneously risk minimizing and reward maximizing. This negative result completes the characterization, showing that in general, the reward-maximizing RM may not be among the risk-minimizing predictors of R.

Theorem 17. *There exists a causal graph and cost structure where no reward model achieves the optimal risk and zero degree of reward hacking.*

Proof. We consider again the linear–Gaussian SCM: where $X_1 \sim N(0, \sigma_1^2)$, $R = X_1 + U_R, U_R \sim N(0, \sigma_R^2)$ and $X_2 = R + U_2, U_2 \sim N(0, \sigma_2^2)$. Let the cost be quadratic $\frac{1}{2}a^\top Ca \leq b$, where $C = diag(c_1, c_2)$.

Bayes Predictor: Our first observation is that the Bayes predictor $f^\star(X) = \mathbb{E}[R \mid X]$ is the unique minimizer of population risk over measurable functions:

$$\mathcal{R}(f) - \mathcal{R}(f^\star) = \mathbb{E}\big[(f(X) - f^\star(X))^2\big] \geq 0,$$

with equality iff $f = f^\star$ a.s.

In this Gaussian model $f^\star$ is linear, it has closed form:

$$f^\star(X) = w_1X_1 + w_2X_2, w_1 = \frac{\sigma_2^2}{\sigma_R^2 + \sigma_2^2}, w_2 = \frac{\sigma_R^2}{\sigma_R^2 + \sigma_2^2}.$$

Reward Increase under Bayes Predictor: As derived before, the optimal intervention under $f^\star$ is: $a^*(f^\star) = \lambda C^{-1}w$ where $\lambda = \sqrt{\frac{2b}{w^\top C^{-1}w}}$. Therefore, the expected increase in R under $f^\star$ is: $a_1^*(f^\star) = \lambda \frac{w_1}{c_1}$.

Reward Increase under other RM: Now consider the reward maximizing RM $f(x) = x_1$. We have that: $a^*(f) = \sqrt{\frac{2b}{1/c_1}} C^{-1} e_1$, which means the expected reward increase under f is $a_1^*(f) = \sqrt{\frac{2b}{c_1}}$.

One can verify that so long as:

$$\frac{w_2^2}{c_2} > 0 \Rightarrow a_1^*(f) > a_1^*(f^\star).$$

which can happen when e.g. when $w_2 \neq 0$. And so, we conclude that no reward model simultaneously minimizes population risk and maximizes expected true reward under this SCM and cost. □

4.5.2 A Closer Look at Proxy Rewards

As we saw, the existence of proxies can complicate our optimal choice of a reward model. When proxies (descendants) exist, we know that the invariant causal predictor may no longer be the optimal RM choice. Thus, a natural next question is: what about proxy reward models? Do very predictive proxy reward models result in very high reward hacking? We find that the answer to this depends on depends on the underlying SCM and cost function, as a range of possibilities exist.

Proposition 7 (Proxy RMs can exhibit both extremes of degrees of reward hacking)**.** *For any ϵ, there exists an ANM and cost structure, and a proxy Reward Model such that:*

- *It has risk $o(\epsilon)$ and $1 - O(\epsilon)$ degree of reward hacking.*
- *It has risk $o(\epsilon)$ and $O(\epsilon)$ degree of reward hacking.*

Proof. Consider the setting where the cost is quadratic and equal across the two nodes, $c(a) = a_1^2 + a_2^2$, and budget $b = 1$.

Consider the causal graph $X_1 \to R \to X_2$. It has SCM: $R = \beta_1 X_1 + U_R$ and $X_2 = \beta_2 R + U_2$.

Define $v_1 = \text{var}(X_1), v_R = \text{var}(U_R)$ and $v_2 = \text{var}(U_2)$. Then, for any linear RM $f = w_1 X_1 + w_2 X_2$, its MSE in terms of the variances has closed-form: $(w_1 + \beta_2 w_2 - 1)^2 v_1 + (w_2 \beta_2 - 1)^2 v_R + w_2^2 v_2$.

The optimal policy has intervention $(a_1^*, a_2^*) = (\frac{w_1 + \beta_2 w_2}{\sqrt{(w_1+\beta_2 w_2)^2 + w_2^2}}, \frac{w_2}{\sqrt{(w_1+\beta_2 w_2)^2 + w_2^2}})$. And so, the reward increase induced by reward model f has closed form: $\frac{w_1 + \beta_2 w_2}{\sqrt{(w_1+\beta_2 w_2)^2 + w_2^2}}$. The optimal possible reward increase is 1 and is attainable with $w = (1, 0)$ (only the parent is intervened upon).

Let SCM be such that $v_1 = v_R = 1/\epsilon$ and $v_2 = \epsilon^4$. This means that the variance of U_R is large, which makes X_1 a noisy predictor of R. By contrast, the variance of proxy X_2 is small, which makes X_2 a good predictor of R:

- First consider a SCM where $\beta_2 = \epsilon$ is small.
  We know that the optimal risk attainable is upper bounded by ϵ^2, which is the risk of $w = (0, 1/\beta_2) = (0, 1/\epsilon)$. And so, it is possible to have very predictive models.

Now, for any predictive model with near-optimal risk $o(\epsilon)$, we need that $|w_2\beta_2 - 1| = o(\epsilon)$ and $|w_1 + \beta_2 w_2 - 1| = o(\epsilon)$. The former implies that $w_2 = 1/\epsilon + o(1)$ and in combination with the second condition implies that $w_1 = o(\epsilon)$.
This means that $w_1 + \beta_2 w_2 = 1 + o(\epsilon)$. And so, the reward increase of any RM can be at most $O(\epsilon)$, and thus reward hacking degree of any model at least $1 - O(\epsilon)$.

- Now, consider a SCM where $\beta_2 = 1/\epsilon$ is large.
  We claim that $f = 1/\beta_2 X_2$ will now have both low risk and low degree of reward hacking. Indeed, under $w = (0, 1/\beta_2) = (0, \epsilon)$, the risk is $\epsilon^2 \cdot v_2 = \epsilon^6$.
  Moreover, w attains a near-optimal reward increase (and hence degree of reward hacking): $\frac{w_1+\beta_2 w_2}{\sqrt{(w_1+\beta_2 w_2)^2+w_2^2}} = \frac{1}{\sqrt{1+\epsilon^2}} \geq 1 - \epsilon$.

The key difference here is that while we again have to resort to using proxy X_2 to predict R, since β_2 is large, under equal cost, almost all the budget will be used to intervene on X_1 instead of X_2. Because increasing X_1 increases X_2 much more than increasing X_2 itself, X_2 is both a good predictor of the reward and a good incentive for inducing interventions that maximally increase R.
□

And so, depending on the SCM parameter and cost, it is possible that proxies reward models can be very effective RMs. The reason is that while the interventions on the proxy node itself is not reward maximizing, the SCM parameters and cost could be that maximizing the proxy incentivizes mostly interventions on upstream nodes, which in turn greatly increases R. In such settings, of which we have detailed one, proxy reward models are not all bad. In addition to being good predictors of R, proxies *can* also make for good reward models too.

4.6 Leveraging Reward Hacking for Causal Discovery

As we saw through the simple, minimalistic examples in the previous section, proxy reward models can incur a varying degrees of reward hacking. And this is dependent on the unknown, underlying causal world model. In this section, we illustrate an interesting application of reward hacking. Reward hacking can be used to do causal discovery. The high level idea is that given a conjectured causal structure, we can design a RM that incentivizes an expected intervention. And when we observe that this intervention is not reflected in the agent's optimal policy, this reveals causal structure and helps us to revise our conjecture.

To this end, we develop Algorithm 10 that can be used to orient any ANM. It requires access to observational distribution $\mathcal{D}_0$ and the function class of g_i as in [141, 229]. Notably, using $\mathcal{D}_0$, we may compute the graph skeleton using e.g. the PC algorithm [265]. And in addition to regression, the function class can be used to recover the full SCM parameters once the causal graph is oriented by Algorithm 10.

The approach to discovery leverages the following result: if we guess $X_{\mathrm{pa}(i)}$ correctly, we should expect there to be only a single intervention on X_i induced by RM f_{X_i}. All the proofs in this section may be found in Appendix 4.8.

Proposition 8 (Targeted Reward Model)**.** *For any node X_i, choosing reward model $f_{X_i} = X_i - g_i(X_{pa(i)})$ induces an optimal intervention only on node X_i.*

Algorithm 10 Discovery Algorithm for ANM under Finite Cost

```
Input: Distribution $\mathcal{D}_0$, Graph Skeleton GS
$\mathcal{G}$ = GS                                    ▷ partially oriented graph $\mathcal{G}$
SG = $\{X_i\}_{i \in [n]} \cup \{R\}$                 ▷ subgraph of unoriented nodes $SG$
$S = \{\}$                                            ▷ set of oriented nodes $S$ (complement of nodes in $SG$)
while $|SG| > 1$ do
    R_leaf = True
    for $X_i \in$ SG $\setminus \{R\}$ do
        $X_{P_i} \leftarrow$ nodes adjacent to $X_i$ in GS
        if $P_i = \emptyset$, $\hat{g}_i = 0$; else $\hat{g}_i = \mathbb{E}_{\mathcal{D}_0}[X_i | X_{P_i}]$
        Deploy $f_{X_i} = X_i - \hat{g}_i(X_{P_i})$ to obtain distribution $\mathcal{D}_i$
        X_leaf = True
        for node $V \in$ SG $\setminus \{X_i\}$ do        ▷ test if $X_i$ is a leaf node in subgraph $SG$
            if $\mathbb{E}_{\mathcal{D}_i}[V] \neq \mathbb{E}_{\mathcal{D}_0}[V]$ then
                X_leaf = False
                break
        if X_leaf then
            for node $V \in$ SG adjacent to $X_i$ in GS do   ▷ each adj node is parent as $X_i$ is leaf
                Orient $V \to X_i$ in $\mathcal{G}$, Remove edge $V - X_i$ from GS
            SG $\leftarrow$ SG $\setminus \{X_i\}$, $S \leftarrow S \cup \{X_i\}$   ▷ update set of (un)oriented nodes $SG$ and $S$
            R_leaf = False
            break
    if R_leaf then   ▷ no $X$ leaf node found in $SG$, by elimination, $R$ must be the (only) leaf
        for node $X_j \in$ SG adjacent to $R$ in GS do   ▷ each adj node is parent as $R$ is leaf
            Orient $X_j \to R$ in in $\mathcal{G}$, Remove edge $X_j - R$ from GS
        SG $\leftarrow$ SG $\setminus \{R\}$, $S \leftarrow S \cup \{R\}$   ▷ update set of (un)oriented nodes $S$ and $SG$
return $\mathcal{G}$                                  ▷ returns fully oriented graph
```

To prove the correctness of the algorithm, we will also need a mild form of Mean Interventional Faithfulness [320]. At a high level, this is so that contrapositive of the proposition holds: if we get the parents of X_i incorrect, then at least one other node besides X_i will have its mean shift. Please see Appendix 4.8 for the full details.

Theorem 18. *Algorithm 10 orients the full causal graph after at most $n(n-1)/2$ episodes.*

Algorithm 10 is a bottom-up algorithm that iteratively discovers leaf nodes in the current subgraph. The leaf-node test is based on the observation that intervening on a leaf only changes the leaf node itself and no other node in the subgraph. Thus, when we observe an unexpected intervention elsewhere, we know our guess for the leaf node is off. When we have identified the leaf node, this means that every node it is adjacent to in the subgraph must be its parent. We can then orient edges accordingly, and recurse on the remaining subgraph.

Comparison with Existing Algorithms: An astute reader may question why Algorithm 10 is needed given there already exists causal discovery algorithms for ANMs [141, 229]. To this, we note that Algorithm 10 has the advantage of reducing orientation to one-dimensional mean shift detection, in place of the much higher dimensional conditional independence tests as needed in [229].

Thus, despite its current guarantees holding in infinite-sample regimes, we believe Algorithm 10 is a promising method to realize the benefit of interventional data in this (important) setting, but non-standard setting. In standard causal discovery settings, the experimenter can directly pick interventions exogenously. However, this setting is more challenging in that interventions are realized endogenously by the agent's optimization of the RM. Our algorithm thus offers one way to nevertheless do discovery, wherein the experimenter uses aptly chosen RMs to indirectly incentivize interventions useful for discovery.

When the Causal Graph is Unidentifiable: Finally, one caveat with Algorithm 10 is that it is predicated on every node being mutable (i.e. finite cost). This may not always be the case as some nodes may have infinite cost, making the true causal graph unidentifiable. Towards handling this challenging setting, we develop Algorithm 11 for the linear graph setting, showing that we *need not* have to discover the causal structure in order to find the reward maximizing reward model. Our key idea is that, through apt choices of reward models, we can provably incentivize all possible interventions that can be induced, and then simply select the one that is reward maximizing.

Theorem 19. *Algorithm 11 finds the reward maximizing policy using at most $2n$ episodes.*

Algorithm 11 Optimization Algorithm under Linear SCM and Linear Cost

```
Deploy $f(x) = x_1$, $f(x) = -x_1$                ▷ collect initial pair of distributions
Compute $\mathbb{E}_{\mathcal{D}_0}[X] = (\mathbb{E}_{\mathcal{D}_1}[X] + \mathbb{E}_{\mathcal{D}_{-1}}[X])/2$
Initialize $S = \{\mathcal{D}_1, \mathcal{D}_{-1}\}$, $W = \{\mathbb{E}_{\mathcal{D}_1}[X] - \mathbb{E}_{\mathcal{D}_{-1}}[X]\}$, $\Pi = \{\}$
for $i = 2, ..., n$ do
    Compute some $w_i$ in the nullspace of $W^T$ using SVD of $WW^T$   ▷ $w_i$ is such that all prior interventions cannot change the reward under $w_i$
    Deploy RM $f(x) = w_i^T x$, $f(x) = -w_i^T x$ and obtain $\mathcal{D}_i, \mathcal{D}_{-i}$
    if $\mathcal{D}_i \in S$ then                     ▷ encounter duplication
        $W \leftarrow W \cup \{w_i\}$
    else            ▷ observe pair of new distributions with new underlying intervention
        $S \leftarrow S \cup \{\mathcal{D}_i, \mathcal{D}_{-i}\}$
        $W \leftarrow W \cup \{\mathbb{E}_{\mathcal{D}_i}[X] - \mathbb{E}_{\mathcal{D}_0}[X]\}$
        $\Pi \leftarrow \{(w_i, \mathbb{E}_{\mathcal{D}_i}[R]), (-w_i, \mathbb{E}_{\mathcal{D}_{-i}}[R])\}$
        if $|S| = 2k$ : break          ▷ only $2k$ pairs of distinct distributions are possible
return reward maximizing policy in $\Pi$
```

4.7 Discussion

In this paper, we study the relationship between a reward model's risk (with respect to R) and its degree of reward hacking. We develop a general, theoretical setup useful for analyzing both. We believe it is a natural, theoretical testbed with simple and minimal examples of causal reward hacking, useful for examining other future approaches (besides predictiveness) to selecting RMs.

We use this testbed to characterize when the causal invariant predictor is optimal, and analyze when proxy RMs may in fact be better. We show that this is a possibility, and in general, a range of degree hacking is attainable. Through our results, we aim to highlight that proxies can be useful for reward modeling and that a designer need not always choose the invariant causal predictor. Indeed, this choice depends on the underlying causal graph. And so, we also develop algorithms that leverage incentivized interventions to discover the underlying graph, which can in turn allow us to design better reward models.

4.8 All Deferred Proofs

Proposition 9. *There exists a causal graph and cost structure such that $\mathbb{E}_U[R|X_{pa(R)}]$ has arbitrarily small population MSE on the observational distribution, but arbitrarily large MSE under the post-intervention distribution.*

Proof. Let $X_1 \sim \mathrm{Unif}([-1,1])$ (observationally) and let $U_R \sim \mathcal{N}(0,1)$ be independent of X_1. Fix constants $\varepsilon > 0$ and $M > 0$, and define the measurable function

$$s(x) = \begin{cases} \varepsilon, & |x| \leq 1, \\ M, & |x| > 1. \end{cases}$$

Define the structural equation

$$R = X_1 + s(X_1)\, U_R.$$

Then for every x we have $\mathbb{E}[R \mid X_1 = x] = x$, hence $f_{\mathrm{pa}(R)}(x) = x$.

The observational MSE is

$$\mathcal{R}_{\mathrm{obs}}(f_{\mathrm{pa}(R)}) = \mathbb{E}_{X_1}[s(X_1)^2] = \varepsilon^2,$$

since $X_1 \in [-1,1]$ a.s.

Let $c = a_1^2$ and $b = 1$. Then, the agent maximizes $\mathbb{E}[f_{\mathrm{pa}(R)}(X_1 + a_1)] = \mathbb{E}[X_1] + a_1 = a_1$, with constraint $a_1^2 \leq 1$. So the optimal intervention is $a_1^* = 1$. Hence, the post-intervention state $X_1' = X_1 + 1$ is $\mathrm{Unif}([0,2])$.

The post-intervention MSE of $\mathbb{E}[R \mid X_1 = x]$ is

$$\mathcal{R}_{\mathrm{post}}(f_{\mathrm{pa}(R)}) = \mathbb{E}_{X_1'}[s(X_1')^2] = \tfrac{1}{2}\varepsilon^2 + \tfrac{1}{2}M^2 = \frac{\varepsilon^2 + M^2}{2},$$

because $\Pr(X_1' \in [0,1]) = \Pr(X_1' \in (1,2]) = 1/2$. By choosing ε arbitrarily small and M arbitrarily large we can make $\mathcal{R}_{\mathrm{obs}}(f_{\mathrm{pa}(R)})$ arbitrarily small while $\mathcal{R}_{\mathrm{post}}(f_{\mathrm{pa}(R)})$ is arbitrarily large. □

Remark 13. *This result crucially requires heteroskedasticity, while in ANMs with homoskedastic additive noise the conditional-mean predictor's MSE is invariant to marginal changes of parents. So such divergence does not arise.*

Proposition 10 (Targeted Reward Model)**.** *For any node X_i, choosing reward model $f_{X_i} = X_i - g_i(X_{pa(i)})$ induces an optimal intervention only on node X_i.*

Proof. With this choice of RM, the optimization objective is:

$$\begin{aligned} \max_a \quad & \mathbb{E}[X'_i - g_i(X'_{\text{pa}(i)})] \\ \text{s.t.} \quad & X'_j = g_j(X'_{\text{pa}(j)}, U_j) + a_j \quad \forall j \in [n] \\ & c(a_1, ..., a_n; x) \leq b \end{aligned}$$

Since g_i is additive, we may plug in $g_i(X'_{\text{pa}(i)}, U_i) = g_i(X'_{\text{pa}(i)}) + U_i$ and the objective becomes $\mathbb{E}[X'_i - g_i(X'_{\text{pa}(i)})] = \mathbb{E}[g_i(X'_{\text{pa}(i)}, U_i) + a_i - g_i(X'_{\text{pa}(i)})] = \mathbb{E}[U_i + a_i]$. And so, the agent is optimizing:

$$\begin{aligned} \max_a \quad & \mathbb{E}[U_i] + a_i \\ \text{s.t.} \quad & x'_j = g_j(x'_{\text{pa}(j)}, u_j) + a_j \quad \forall j \in [n] \\ & \sum_{j=1}^{n} c_j(a_j; x) \leq b \end{aligned}$$

Since the objective is not a function of X, a^* is constant in X. Next, because each cost function c_j is strictly increasing in the magnitude of a_j, we must have that $a^*_j = 0$ for $j \neq i$ (otherwise one can increase a_i instead to increase the objective). And since the objective is strictly increasing in a (in particular a_i). The budget constraint is binding and thus we have that only $a^*_i \neq 0$.

□

Assumptions: To show the following Theorem, we will need the following assumptions:

1. (Mean Interventional Faithfulness): Let $V \in G$ be any node in G. Let $\mathcal{I}^i$ be the set of all non-`null`, intervention nodes under intervention induced by f_{X_i}, resulting in distribution $\mathcal{D}_i$:

   $$\exists I \in \mathcal{I}^i \text{ s.t } I \not\perp\!\!\!\perp V' \Leftrightarrow \mathbb{E}_{\mathcal{D}_i}[V] \neq \mathbb{E}_{\mathcal{D}_0}[V].$$

   This is a mild faithfulness assumption, and for the proof to go through, we will actually only require a particular instantiation of the faithfulness assumption above (as used also in e.g [320]).

   This is that if node X_i is intervened upon, and V is its child highest in the topological order, then the mean of V in the interventional distribution shifts. Put another way, this assumes that the event that the interventional values on X_i and V, and the SCM parameter relating the two nodes are not pathologically such that the interventions cancel out exactly. And so, the mean of V changes almost surely.

2. Secondly, our algorithm will require access to the observational distribution of the world model ($\mathcal{D}_0$) and the function class for g_i as in [141, 229]. Using $\mathcal{D}_o$, we note that we may compute the graph skeleton e.g. using the PC algorithm [265]. And the function class can be used to recover the SCM parameters once the causal graph is oriented as in Algorithm 10.

Theorem 20. *Algorithm 10 orients the full causal graph after at most $n(n-1)/2$ episodes.*

Proof. We will prove that the parents of each node are correctly identified by the algorithm, which implies that the full graph is correctly identified. To do this, we will show that Algorithm 10 always maintains the invariant property (1) that, each iteration, the node that is added to S from the subgraph SG is always a leaf node.

(1) has the implication that (2) no node is added before all of its descendants. Indeed, a node is only added when it is a leaf, and if a node does have at least one descendant in the subgraph, it is not a leaf and cannot be added.

Thus, with (1), the algorithm will be such that the following holds: (3) that every node in S has its parents correctly and completely identified. When a new node is added to S, we identify all nodes in SG adjacent to the new node as its parents. Since the node is a leaf, every such node in SG can only be its parents, and by (2) must be all of its parents. And so, this ensures that this new node's parents also satisfy (3).

We see that (1) is satisfied for S at initialization. To prove (1) always holds, it suffices to show that leaf nodes in any subgraph SG will be such that Condition 13 is always false and no non-leaf node in any subgraph SG will be such that Condition 13 is always false.

Let $\mathcal{X}$ be the set of $\{X_i\}_{i=1}^n$ nodes. For a particular subgraph SG, suppose SG and $S = \mathcal{X} \backslash SG$ satisfies (2).

Non-Leaf Nodes in $\mathcal{X}$ do not pass test: First, note that there has to exist at least one node that is intervened upon. This is because f' is monotonically increasing in a_i.

Next, since the policy is a function of X_{P_i} and X_i, the intervention will take place on node(s) that are ancestors of nodes of $\{i\} \cup P_i$. This is again because for any $j \notin \text{anc}(i) \cup \text{anc}(P_i)$, changing a_j will not change X'_{P_i} and X'_i (and thus the objective), but strictly increases costs. By (2), since $\{i\} \cup P_i \in SG$, $\text{anc}(i) \cup \text{anc}(P_i) \in SG$. That is, every node that will be intervened upon when $f_{X_i} = X_i - \hat{g}_i(X_{P_i})$ is deployed will be in the subgraph SG.

Out of all nodes which are intervened upon under $f = X_i - \hat{g}_i(X_{P_i})$, let k be the index of a node such that none of its ancestors is intervened upon (i.e an intervened node that is highest in topological order). If $k \neq i$, we have $\mathbb{E}_{\mathcal{D}_i}[X_k] \neq \mathbb{E}_{\mathcal{D}_0}[X_k]$ since X_k is dependent on I_k^i.

Else, we have that $k = i$. We know that since X_i is not a leaf, it must have at least one child. Let V in SG be the child of i with the highest topological order. We have that V is dependent on I_i^i due to chain $I_i^i \to X_i \to V$. And so, by our faithfulness assumption, its expectation under $\mathcal{D}_i$ will change due to the intervention on X_i. Note that it may be that under f, V may also be intervened upon; our faithfulness assumption is that the SCM parameters are not such that the two interventions cancel out exactly.

Either way, we conclude that Condition 13 will hold for at least one node in the subgraph.

Leaf Nodes in $\mathcal{X}$ pass test: Suppose first that subgraph SG has a leaf node X_i. Then, all its parents must be still in the subgraph SG by property (2). Since it is a leaf in SG, none of its children is in SG. And so, all the nodes P_i adjacent to X_i in the GS must be its parents and only its parents. Thus, in additive SCMs, the model $\hat{g}_i = \mathbb{E}_{\mathcal{D}_0}[X_i|X_{P_i}]$ identifies g_i, the true SCM parameter, up to a fixed constant which does not affect the best response. Thus, from Proposition 10, we have that in $\mathcal{D}_i$, only X_i is intervened upon.

With this, we can conclude that no other node in SG has its distribution change since X_i has no descendants; X_i's intervention only changes the distribution of X_i. Hence if X_i is a leaf, Condition 13 will always be false. (1) will be satisfied as we have just shown that a node of the subgraph will make Condition 13 always false iff it is a leaf.

Lone R leaf: Finally, the remaining case is when R is the only leaf of the current subgraph. We have just shown that if there is a leaf in SG and in $\mathcal{X}$, it will meet the criteria. We have also shown earlier that no non-leaf node in $\mathcal{X}$ can meet the criteria. So if it is the case that no nodes in $\mathcal{X}$ meets the criteria, then by the process of elimination, R must be the only leaf in the subgraph.

Termination: The algorithm terminates when there is only one node left in the subgraph. By (2), it must be a root node in the full graph. This means that we have also managed to identify its parents, which is the empty set.

Complexity: The algorithm adds one node to S per iteration and there are at most n iterations. During each iteration, we run at most $|SG|$ regressions and $|SG|$ episodes. And so, at most $n(n-1)/2$ regressions and episodes are needed to discover the graph.

□

Remark 14. *Intuitively, leaves of subgraphs are useful since intervention and change in distribution is isolated to the leaf nodes. By contrast, for root nodes, interventions will change nodes of the entire subgraph.*

Optimal Policy under Linear Graph and Linear Cost Under this setting, the agent is optimizing:

$$\begin{aligned} \max_{a} \quad & w^T B a \\ \text{s.t.} \quad & \sum_{i=1}^{n} c_i |a_i| \leq b \end{aligned}$$

Then the optimal intervention $a^*(w)$ is as follows: with $i^* = \text{argmax}_{j \in [n]} \frac{|(B^T w)_j|}{c_j}$:

$$a^*(w) = \text{sign}((B^T w)_{i^*}) \left[\frac{b}{c_{i^*}} e_{i^*} \right] \tag{4.2}$$

From this, we observe that at most $2n$ types of interventions may be induced: $\pm \frac{b}{c_i} e_i$. Moreover, each one intervention can be induced. For example, we note that $a^*(w) = \frac{b}{c_i} e_i$ for $w = (B^T)^{-1} e_i$.

Now to address the tie-breaker in the case when some nodes can be immutable, let $S_M \subseteq [n]$ be the subset of nodes which are mutable. That is, $c_i \neq \infty \Leftrightarrow i \in S_M$. We will assume that if w is such that $(B^T w)_j = 0$ for all $j \in S_M$, then the optimal $a^*(w)$ we observe will be some intervention $i \in S_M$. We will make no assumption on how this tie-breaking is done and which index $i \in S_M$ is chosen, just that the agent's tie-breaking on which node is intervened on by the same RM w is consistent across episodes.

Theorem 21. *Algorithm 11 finds the reward maximizing policy using at most $2n$ episodes.*

Proof. We will prove algorithm correctness in several parts:

Estimation of $\mathbb{E}_{\mathcal{D}_0}[X]$ Through the distribution induced by w, we may observe $\mathbb{E}_{\mathcal{D}_0}[X] + Ba^*(w)$. The first step of the problem is to estimate $\mathbb{E}_{\mathcal{D}_0}[X]$ such that we may observe $Ba^*(w)$ directly.

To do this, we deploy $w = e_1$, $w = -e_1$. It remains to argue that $a^*(w) = -a^*(-w)$. This follows because $i^* = \text{argmax}_{j\in[n]} \frac{|(B^T w)_j|}{c_j} \Leftrightarrow i^* = \text{argmax}_{j\in[n]} \frac{|(B^T(-w))_j|}{c_j}$. From this, we can conclude $a^*(w) = \text{sign}((B^T w)_{i^*}) \left[\frac{b}{c_{i^*}} e_{i^*}\right] = -\text{sign}((B^T(-w))_{i^*}) \left[\frac{b}{c_{i^*}} e_{i^*}\right] = -a^*(-w)$, using the closed form optimal solution in Equation 4.2.

Elicitation of all possible distributions WLOG $S_M = \{1, ..., k\}$, where $k \leq n$ is the number of mutable features. Let W^0 denote the nullspace of $[Be_1; ...; Be_k]$, where e_i corresponds to the standard basis vector wrt node i.

We will show that, after $n-1$ iterations of for-loop 4, we will not have observed a new distribution (corresponding to a new underlying intervention) $n-k$ times (reaching Condition 7). From this, we must have observed $n-1-(n-k) = k-1$ new underlying interventions, which must correspond to the rest of the $k-1$ interventions that are possible. Thus, when the algorithm terminates, we would have observed all $2k$ distributions that are possible, corresponding to the k possible interventions, with both signs possible for each intervention.

Consider iteration i and suppose we have observed distributions corresponding to interventions on nodes $\{i_1, ..., i_{k'}\}$ for $k' < k$. The algorithm uses SVD to find a vector $w_i \neq 0$ in the null-space of W, which means it is also in the nullspace of $[Ba^*(w'_1); ...; Ba^*(w'_{k'})]$ where w'_j denotes the model that induced intervention i_j. Note that since rank$(W) \leq i < n$, the nullspace of W is non-empty and we can always find such a w_i.

Next, notice that since w_i is in the nullspace of $[Ba^*(w'_1); ...; Ba^*(w'_{k'})]$, it must also be in the null-space of $[Be_{i_1}; ..., Be_{i_{k'}}]$. With this, w_i must be such that $w_i^T Be_{i_j} = 0 \Leftrightarrow (B^T w_i)_{i_j} = 0$ for all $j \in [k']$. Moreover, we know that since B^T is full-rank, $B^T w_i \neq 0$. And so, if $w_i^T Be_j \neq 0$ for some $j \in [k] \setminus \{i_1, ..., i_{k'}\}$, then we will observe a new distribution corresponding to some intervention in $[k] \setminus \{i_1, ..., i_{k'}\}$.

If it is the case that we do not observe a new distribution, we must have that $w_i^T Be_j = 0$ for all $j \in [k] \setminus \{i_1, ..., i_{k'}\}$ as well. Therefore, $w_i \in W^0$.

Suppose by contradiction, we reach Condition 7 more than $n-k$ times. This means that there exists at least $n-k+1$ vectors $w_{j_1}, .., w_{j_{n-k+1}}$ in W^0. By construction, each vector is orthogonal to the rest, which means $w_{j_1}, .., w_{j_{n-k+1}}$ are linearly independent. This implies that $\dim(W^0) \geq n-k+1$.

This however is a contradiction, because we have:

$$\dim(W^0) + \dim(\{Be_1, ..., Be_k\}) = \dim(W^0) + k = n,$$

since B is full rank and its columns are linearly independent.

With this, we can observe all possible interventions that can be induced, and thus determine the policy that maximizes R.

□

4.9 Additional Related Works

Broadly on the topic of reward hacking and its definitions/properties, [262] provides a first theoretical definition of reward hacking in general in RL. This happens when an agent achieves high proxy rewards, while performing poorly with respect to the true reward. Moreover, this work shows that reward hacking can emerge even when the proxy and true rewards are positively correlated, highlighting the severity of the problem. [175] adds a new definition of reward hacking, based on the correlation between proxy and true rewards under a reference policy distribution. Their key insight is that reward hacking occurs when this correlation breaks down during optimization, even if the proxy initially appears well-aligned. Finally, [222] provides interesting theoretical insights into reward hacking through their study of phase transitions in agent behavior. They demonstrate that as optimization power increases, agents can undergo sudden behavioral shifts from benign to highly exploitative strategies, often with little warning.

Chapter 5

Finite-Sample Causal Discovery

5.1 Introduction

Causal discovery is a fundamental goal of natural and social sciences, with widespread use across fields such as biology, physics and economics [224, 265]. As a result, there has been great interest in discovery methods with *provable guarantees*. In the task of causal discovery, one assumes access to the observational distribution, from which one can compute the undirected graph skeleton $\overline{G}$ with an unoriented edge between every cause and effect. Under specific functional assumptions on the graph, the underlying causal DAG can be identified from observational data alone. In more general settings, interventional data is needed for discovery. The goal of causal discovery is thus to minimize the amount of interventional data needed to identify the true causal graph. A typical discovery algorithm is outlined as in Algorithm 12, where the two key subroutines are the "query step" (adaptively determine which interventional data to collect next) and the "update step" (given the latest sample, orient edges in G using all the data collected so far).

The existing line of work on causal discovery with provable guarantees have largely focused on the query step; a non-exhaustive list of such papers include [70, 97, 114, 143, 148, 164, 188, 257]. Key to the analysis is the assumption of hard intervention (under infinite samples), an idealized model of node intervention. That is, when node v is intervened on, the orientation of all edges in $\overline{G}$ incident to v is revealed. Thus, the update step can be easily implemented, and the algorithm performance be neatly defined in terms of the number of intervened nodes needed to fully orient the graph.

Importantly, this idealized model of node intervention overlooks the statistical complexity of orienting an edge in real world settings. If we view each edge orientation as a hypothesis, then almost always *multiple* samples are needed to reject with high probability (w.h.p.) an incorrect hypothesis (edge orientation), due to stochasticity in the data samples. Thus, towards studying finite-sample discovery, we consider the setup considered by Greenewald et al. [125]. An experiment with intervention v now provides *one sample* from v's interventional distribution, which by itself may not be sufficient to orient the edge.

In this setting, it is no longer trivial to implement the update step. Thus, to *even begin* to study the finite-sample setting, we first need a framework that can implement the update step: given the interventional data obtained so far, decide which edges can be oriented. Put another way, a correct

Algorithm 12 Causal Discovery Algorithm Template

```
Input: Essential graph $G$, Query algorithm $\mathcal{A}$
while $|MEC(G)| > 1$ do                                  ▷ multiple graphs in the Markov Equivalence Class
    $\mathcal{A}(G) \to X^t$                                              ▷ query step
    Observe a sample from interventional distribution $(x_1^t, ..., x_n^t) \sim X_1, ..., X_n | do(X^t)$    ▷ collect new data
    Test orientation of each unoriented edge using data $\{(x_1^j, ..., x_n^j)\}_{j=1}^t$ collected so far, and update $G$ accordingly    ▷ update step
Return $G$
```

implementation of the update step is needed to *measure* algorithm performance. And only after we have this can we get to developing algorithms with provably good performance. Specifically, we note the following two properties are desirable for the framework to have:

1. **Anytime Valid Testing:** The most basic property required of any framework that implements the update step is correctness. That is, any edge that is oriented at any timestep should be correct w.h.p. In the finite-sample causal setting, this means that the testing framework has to have anytime validity.

   To see why, note that the number of samples needed for orientation varies depends on the unknown, underlying edge strength. For instance, many fewer samples are needed to orient $X_1 \xrightarrow{1000} X_2$ w.h.p. compared to that of $X_1 \xrightarrow{0.001} X_2$. And so, *anytime valid* testing is needed as hypotheses (corresponding to edge orientation e.g. of $X_1 \to X_2$) will be tested a number of times, where this number is unknown apriori.

2. **Encoding Propagation Implications:** Efficient discovery algorithms under hard intervention orient edges by considering the propagation implications of node interventions. Intervening on an "informative" node orients edges, whose orientations in turn propagate to many other edges via Meek rules [200].

   Thus in the finite-sample setting, a secondary, useful property for the framework to have is to be able to encode this structure, and relate hypotheses (edge orientations). We note that that this structure is useful for obtaining higher power tests. For a simple example, consider testing $X_1 \to X_2$ in $X_1 - X_2 - X_3$. Evidence against $X_2 \to X_3$ also serves as evidence against $X_1 \to X_2$, since by Meek rule $X_1 \to X_2 \Rightarrow X_2 \to X_3 \therefore \neg X_2 \to X_3 \Rightarrow \neg X_1 \to X_2$.

In this chapter, we develop a framework that has both properties 1 and 2. To the best of our knowledge, our framework is the first that has these requisite properties. It perform anytime valid testing using the collected interventional data, with controlled error rate. That is, at any point in time (for however long it takes for the graph to be fully oriented), every oriented edge is correct w.h.p. This allows our framework to be paired with any causal discovery strategy (that implements the "query step") to perform finite-sample causal discovery.

The key observation used to develop the framework is that causal discovery can be viewed as structured, anytime hypothesis testing. The orientation of each edge in $\overline{G}$ corresponds to two hypotheses, one for each possible orientation. There is structure among the hypotheses due to the Meek rules. Accordingly, our framework makes use of *e-processes* for testing. This is a type of test statistics that allows for both anytime valid testing and flexible combination of test statistics [235].

Our Contributions: First, in Section 5.3, we develop test statistics that is anytime valid (Property 1). In Section 5.4, we consider how one may combine test statistics to leverage graph structure (Property 2). In Section 5.5, we empirically verify the validity of our framework. Finally in Section 5.6, to make use of our testing framework, we develop a novel multi-constraint bandit algorithm for causal verification.

5.2 Problem Setup

We consider a linear graph with n nodes, where $X_i = \theta_i^T X_{\text{pa}(i)} + u_i$ with the set of exogenous noises U sub-Gaussian: $u_i \sim subG(\sigma^2)$. We note that our results generalize to additive graphs, provided knowledge of upper bounds on the variance of intervention distributions.

In line with the canonical setup in theoretical causal discovery literature, we assume causal sufficiency and access to the observational distribution $\mathcal{D}_0$ and graph skeleton $\overline{G}$ [70, 97, 114, 143, 148, 164, 188, 257]. In certain settings, existing algorithms such as the PC algorithm [265] can orient additional edges on top of the graph skeleton. We note that our results are applicable to any essential graph returned by such algorithms. For most of our results, we are concerned with the worst-case setting wherein we only know the graph up to the graph skeleton. Still, our framework can be used to efficiently test and orient the *remaining* unoriented edges given any essential graph. Finally, with consideration for real-world robustness, we also consider the setting where the graph skeleton contains spurious edges in Section 5.3.4. We demonstrate that we can construct robust test statistics that do not propagate the error in the graph skeleton.

In addition to causal sufficiency, we also assume faithfulness: when a node is intervened upon, the expectation of each of its children nodes does change. Just as in the setup of [125], we consider a mildly stronger form of faithfulness where, for every cause-effect pair, there is a *minimal* causal effect b. That is, if we let the causal effect of i on j be $\mu_j(i) := \mathbb{E}[X_j|do(X_i)]$, then $\mu_j(i) \neq 0 \Rightarrow |\mu_j(i)| > b$.

For experimentation, we assume the scientist can perform sequential, single-node interventions with interventional value ν. In our setting, we focus on soft intervention, and we note that our testing framework is also applicable in the hard intervention setting, when mean-shift detection is used for edge orientation. Let I_t denote the node intervened on at time t. As in [125], following an intervention on node I_t, we observe one sample from the joint distribution, $\mathbf{X_{I_t}} \sim \Pr(X_1, ..., X_n|do(X_{I_t}))$.

Our primary goal in this chapter is to design a framework where we can use the data we collect from interventions to construct a sequence of partially oriented graphs $(\widehat{G}_t)$ such that *every* edge is oriented correctly at all time steps $t \in \mathbb{N}$ with high probability.

Definition 22 (Anytime-valid partially oriented graph)**.** *Let $\widehat{G}_t$ denote the set of oriented edges after the first t interventions. A sequence of partially oriented graphs $(\widehat{G}_t)$ is an anytime-valid partially oriented graph if it satisfies:*

$$\Pr(\exists t \in \mathbb{N} : \textit{exists } \textbf{\textit{incorrect}} \textit{ edge in } \widehat{G}_t) \leq \alpha \tag{5.1}$$

for some predetermined error rate $\alpha \in [0, 1]$.

For further discussion on other relevant works and setups, please refer to Section 5.7.

5.3 Anytime-valid testing via e-processes

First, in Section 5.3.1, we show that if we are able to construct an e-process for each edge orientation, then we can correctly implement the update step. This is because testing using e-processes guarantees that every edge that is oriented *at any point in time* is correct w.h.p. That is, the update step is correct across time w.h.p. With this motivation in mind, in Section 5.3.2, we construct e-processes that can be used for edge orientation. We begin with some definitions.

Definition 23 (Canonical Filtration)**.** *A* filtration $(\mathcal{F}_t)_{t\in\mathbb{N}_0}$ *is a sequence of nested sigma-algebras, i.e.,* $\mathcal{F}_t \subseteq \mathcal{F}_t$ *for all* $t \in \mathbb{N}$. *We define the* canonical filtration *to have elements* $\mathcal{F}_t \coloneqq \sigma(\{\mathbf{X}_k\}_{k\in[t]} \cup \{U\})$ *for each* $t \in \mathbb{N}$ *and let* $\mathcal{F}_0 \coloneqq \sigma(\{U\})$. $(\mathcal{F}_t)$ *is essentially the sequence of variables observed after each intervention, and any internal randomness in the algorithm for selecting* I_i *for the first* t *interventions.*

Definition 24 (Intervention-specific Filtration)**.** *Define* $(\mathcal{F}^i_t)$ *as the filtration over data just from interventions on* i*:* $\mathcal{F}^i_t \coloneqq \sigma(\{\mathbf{X}_k\}_{k:k\in[t],I_t=i} \cup \{U\})$ *for each* $t \in \mathbb{N}$.

Definition 25. *Define a* supermartingale *w.r.t. to filtration* $(\mathcal{F}'_t)$ *be any process* $(M_t)_{t\in\mathbb{N}}$ *s.t.* $\mathbb{E}[M_t \mid \mathcal{F}'_{t-1}] \leq M_{t-1}$ *and* M_t *is measurable w.r.t.* $\mathcal{F}'_t$ *for each* $t \in \mathbb{N}$. *For simplicity, we will always let* nonnegative supermartingale (NSM) (M_t) *satisfy* $\mathbb{E}[M_1] \leq 1$.

Definition 26. *Define an* e-process $(E_t)_{t\in\mathbb{N}}$ *w.r.t. to* $(\mathcal{F}'_t)$ *as a nonnegative process where there exists an NSM w.r.t. to* $(\mathcal{F}'_t)$*,* (M_t)*, s.t.* $E_t \leq M_t$ *for all* $t \in \mathbb{N}$ *almost surely, and* $\mathbb{E}[M_1] \leq 1$. *Note that every NSM is an e-process. Equivalently,* (E_t) *is an e-process iff it satisfies* $\mathbb{E}[E_\tau] \leq 1$ *for any stopping time* τ.

The only (key) property we use about e-processes is that it satisfies the following anytime guarantee, per Ville's inequality. At a high level, e-processes may be thought of something that satisfies the following crucial property, which is what enables sequential testing with provable error control.

Fact 3 (Ville's inequality [284])**.** *For any* e-process $(E_t)_{t\in\mathbb{N}}$*:* $\Pr(\exists t \in \mathbb{N} : E_t \geq 1/\alpha) \leq \alpha$.

5.3.1 A general approach for constructing anytime-valid partially oriented graphs

As mentioned previously, we may view each edge orientation as a hypothesis test. For an oriented edge $i \to j$, we may define the associated null hypothesis to be:

$$H_0^{i\to j} : \text{edge } (i, j) \text{ has orientation } i \to j \text{ in } G^*.$$

To test a hypothesis $H_0^{i\to j}$ with anytime validity, our testing framework simply requires the construction of an process $(E_t^{i\to j})$ that satisfies the following condition:

$$H_0^{i\to j} \text{ holds } \Rightarrow E_t^{i\to j} \text{ is an } \textit{e-process}$$

Note that this framework is general and one may design test statistics specific to the problem at hand, so long as the test statistic is an e-process under the null. Once we have such an $(E_t^{i\to j})$, our test is $\varphi_t^{i\to j}(\alpha) := \mathbb{1}\{E_t^{i\to j} \geq 1/\alpha\}$ and we may test as follows:

$$\begin{aligned}&\text{Reject } H_0^{i\to j} \text{ (i.e. claim } j \to i \text{ is correct)}\\ &\quad\text{if } \varphi_t^{i\to j}(\alpha) = 1 \text{ at any } t \in \mathbb{N}.\end{aligned} \tag{5.2}$$

Proposition 11. $(\varphi_t^{i\to j})$ *is an* anytime-valid test*. That is, the procedure in* (5.2) *ensures that for all error rates* $\alpha \in [0,1]$*:*

$$\mathbb{P}(H_0^{i\to j} \textit{ is rejected } \mid H_0^{i\to j} \textit{ is true}) =$$
$$\mathbb{P}(\textit{exists } t \in \mathbb{N} : \varphi_t^{i\to j}(\alpha) = 1 \mid H_0^{i\to j} \textit{ is true}) \leq \alpha$$

Being able to construct anytime-valid test statistics is useful, because one can use it to produce anytime-valid partially-oriented graphs.

Using anytime-valid tests, we can construct an anytime-valid partially oriented graph by union bounding across the $|\overline{G}|$ tests.

Proposition 12. *Given an anytime-valid test* $(\varphi_t^{i\to j})$*, orient edge* $i \to j$ *in* $\widehat{G}_t$ *the first time* $\varphi_t^{j\to i}(\alpha/|\overline{G}|) = 1$*. Then,* $(\widehat{G}_t)$ *is an anytime-valid partially oriented graph.*

In summary, if we are able to construct anytime valid partially oriented graphs through anytime valid test statistics (such as e-processes), then we have in hand a testing framework that can correctly execute the update step w.h.p.

5.3.2 Construction of per-edge base e-processes

One way to construct e-processes is by combing a sequence of sequential e-values, defined as follows.

Definition 27. *A sequence of* sequential e-values (S_t) *w.r.t. to a filtration* $(\mathcal{F}'_t)$ *under null hypothesis* H_0 *is defined as satisfying:* $\mathbb{E}[S_t \mid \mathcal{F}'_{t-1}] \leq 1$ *for all* $t \in \mathbb{N}$ *under* H_0*.*

To develop a test statistic for testing hypothesis $i \to j$, we develop sequential e-values for testing $H_0^{i\to j}$.

It is natural to start by considering evidence from interventional data on node i and j. Both interventions provide evidence against $i \to j$ if the edge is actually $j \to i$. Below, we construct e-values under $do(i)$ and $do(j)$, which allows us to construct an e-process when we are given interventional data from i and j respectively.

Intervention on j: Suppose $I_t = j$, under $do(j)$, it natural to look at X_t^i. If $i \to j$, then X_t^i would still be mean 0, sub-Gaussian random variable, since the cause is not changed by changes in the effect. However, if $i \leftarrow j$, then X_t^i would have a shifted mean.

Thus, we define updates $S_t^{i\to j,+}(j), S_t^{i\to j,-}(j)$, which we show are sequential e-values:

$$S_t^{i\to j,+}(j) := \exp\left(\lambda_t X_t^i - \frac{\lambda_t^2 \sigma_i^2}{2}\right)$$
$$S_t^{i\to j,-}(j) := \exp\left(\lambda_t (-X_t^i) - \frac{\lambda_t^2 \sigma_i^2}{2}\right).$$

where (λ_t) is adapted to $(\mathcal{F}_t)$.

Proposition 13 (Effect on cause)**.** *For any sequence* (λ_t) *that is predictable w.r.t.* $(\mathcal{F}^j_t)$*,* $S^{i\to j,+}_t(j)$ *and* $S^{i\to j,-}_t(j)$ *are both sequential e-values under* $H^{i\to j}_0$ *w.r.t. filtration* $(\mathcal{F}^j_t)$*.*

Intervention on i**:** Suppose $I_t = i$, under $do(i)$, the assumption of minimal causal effect, b, allows us to include further evidence. We have that $H^{i\to j}_0 = H^{i\to j,+}_0 \cup H^{i\to j,-}_0$, where the two hypotheses are defined:

$$H^{i\to j,+}_0 : H^{i\to j}_0 \text{ is true and } \mu_i(j) \geq 0$$
$$H^{i\to j,-}_0 : H^{i\to j}_0 \text{ is true and } \mu_i(j) < 0$$

That is, if i causes j, then the casual effect of i on j is either positive or negative.

Since interventions result in a minimal shift of b in the mean, we can construct the following sequential e-values:

$$S^{i\to j,+}_t(i) := \exp\Big(\lambda_t(b - X^j_t) - \lambda_t^2\sigma_j^2/2\Big) \text{ if } \mu^j(i) > 0 \ ,$$
$$S^{i\to j,-}_t(i) := \exp\Big(\lambda_t(b + X^j_t) - \lambda_t^2\sigma_j^2/2\Big) \text{ if } \mu^j(i) < 0$$

Proposition 14 (Cause on effect)**.** *Under the minimal causal effect condition, we have the following:*

Under $H^{i\to j,+}_0$*,* $S^{i\to j,+}_t(i)$ *are sequential e-values w.r.t. filtration* $(\mathcal{F}^i_t)$*.*

Under $H^{i\to j,-}_0$*,* $S^{i\to j,-}_t(i)$ *are sequential e-values w.r.t. filtration* $(\mathcal{F}^i_t)$*.*

With these e-values, we may construct aggregate test statistics under interventional data i and j, which we prove are e-processes.

Proposition 15. *Under* $H^{i\to j}_0$*, the following processes are e-processes w.r.t. filtrations* $(\mathcal{F}^j_t)$*,* $(\mathcal{F}^i_t)$ *respectively:*

$$E^{i\to j}_t(j) := \frac{1}{2}\left(\prod_{k:I_k=j}^{t} S^{i\to j,-}_k(j) + \prod_{k:I_k=j}^{t} S^{i\to j,+}_k(j)\right)$$
$$E^{i\to j}_t(i) := \min\left(\prod_{k:I_k=i}^{t} S^{i\to j,-}_k(i), \prod_{k:I_k=i}^{t} S^{i\to j,+}_k(i)\right)$$

5.3.3 Growth rate of e-processes

Suppose that it is the case that $j \to i$, we show that our test statistics in Proposition 15 are such the test *has power*. That is, it suffices to show that the test statistic will *increase* under the alternative, eventually exceed $1/\alpha$, and lead to the rejection of the null hypothesis $H^{i\to j}_0$.

Below, we derive the expected growth rate, which is a standard measure of the power of an e-process test. We note that the growth rate of (the log of) the e-values is edge-specific. It is a function of the edge's causal strength and variance. Also, we note that since the log of the e-values is sub-Gaussian, the test statistic concentrates quickly.

Proposition 16. *Suppose the true edge orientation is actually that $j \to i$ and WLOG $\mu^i(j) > 0$. By setting $\lambda_t = b/\sigma_i^2$ for $S_t^{i\to j}(i)$ and $\lambda_t = b/\sigma_j^2$ for $S_t^{i\to j}(i)$, we have the following growth rates:*

1. $\mathbb{E}[\log S_t^{i\to j,+}(j) \mid \mathcal{F}_{t-1}] = b(\mu_i(j) - b/2)/\sigma_i^2$
2. $\mathbb{E}[\log S_t^{i\to j,+}(i) \mid \mathcal{F}_{t-1}] = \mathbb{E}[\log S_t^{i\to j,-}(i) \mid \mathcal{F}_{t-1}] = b^2/(2\sigma_j^2)$

5.3.4 Robust Testing

In practical settings, the graph skeleton provided may contain mis-oriented edges. In what follows, we show that it is possible to detect and correct incorrect edges in the graph skeleton.

Specifically, we observe that by using only the test statistic $S_t^{i\to j}(j)$, our tests will be robust to spurious edges. The proof is simply that, if neither nodes have an effect on each other, the shift in mean is zero. Thus, both test statistics have expectation at most 1, and are thus e-processes. From Proposition 11, we then know that neither tests will reject w.h.p. And so, we *will not* mistakenly orient an edge w.h.p, when there is none there.

On top of this, we can then use the non-conclusiveness of both tests, after *sufficiently* many rounds, to correct an incorrectly specified edge. Indeed, when there is an edge, we should expect one of the two tests to reject within a bounded number of rounds with high probability. Thus, if we know a lower bound for the edge size, then we can use the non-rejection of both tests after sufficiently many rounds to determine that the edge is spurious. Indeed, if there is an edge, one of the two tests should have rejected w.h.p.

We now derive this bound as follows. For a sequence of sequential e-variables (S_t), define $\tau_\alpha := \min\{t \in \mathbb{N} \cup \{\infty\} : \prod_{k=1}^t S_k \geq \alpha^{-1}\}$ to be the first time $t \in \mathbb{N}$ where the product of S_t exceeds α^{-1} for any $\alpha \in [0, 1]$ (or ∞ if S_t never exceeds α^{-1}).

Proposition 17. . *If the edge $j \to i$ is the true orientation in G, then each of the the following statements hold true with probability $1 - \beta$ for each $\beta \in [0, 1]$:*

1. *For $(S_t^{i\to j,+}(j))$, we have that $\tau_\alpha \leq \frac{\sigma_i^2 \log(\alpha^{-1}\beta^{-1})}{b(\mu_i(j)-b)}$.*
2. *For $(S_t^{i\to j,\pm}(i))$, we have that $\tau_\alpha \leq \frac{\sigma_j^2 \log(\alpha^{-1}\beta^{-1})}{b^2}$*

Thus, these sample complexity results provide high probability upper bounds on the process corresponding to the product of sequential e-variables.

Please refer to Section 5.8 for the proofs of all results in this section and experiment plots.

5.4 Combining edge e-processes according to propagation rules

In this section, we study the theory of *combining* anytime valid e-processes, developed in the previous section. Recall, these test statistics (as in Proposition 15) were constructed for testing a single edge, in isolation. However, implications of Meek rules can allow us to propagate evidence from other edges to our edge of interest.

Importantly, this means that for testing $i \to j$, it is possible to make use of interventional data from *not just* nodes i, j. As we will show, e-processes can be flexibly combined and allow for propagation rules to be encoded into the test-statistic to take advantage of this structure.

Firstly, we observe that each Meek rule may be viewed as being one of two types of logical implications. Let $i_0 \to j_0, i_1 \to j_1, i_2 \to j_2$ be directed edges in the graph. Meek rules are of two forms:

$$i_1 \to j_1 \Rightarrow i_0 \to j_0 \text{ i.e., propagation of a single edge.} \tag{5.3}$$

$$(j_2 \to i_2 \wedge j_1 \to i_1) \Rightarrow j_0 \to i_0$$
$$\text{i.e., propagation of two edges to a single edge.} \tag{5.4}$$

Taking the contrapositive (CP) of Rule (5.4) results in the following rule: $i_0 \to j_0 \Rightarrow (i_2 \to j_2 \vee i_1 \to j_1)$.

Lemma 33 (Meek rules imply hypothesis conjunction/disjunction)**.** *For any edge orientation hypotheses $H_0^{i_0\to j_0}, H_0^{i_1\to j_1}, H_0^{i_2\to j_2}$, we have that*

$$H_0^{i_0\to j_0} = H_0^{i_0\to j_0} \cap H_0^{i_1\to j_1} \textit{ by Rule } 5.3$$
$$H_0^{i_0\to j_0} = H_0^{i_0\to j_0} \cap (H_0^{i_1\to j_1} \cup H_0^{i_2\to j_2}) \textit{ by CP of Rule } 5.4$$

This is useful, because under Rule 5.3 for example, testing $i_0 \to j_0$ is equivalent to testing $i_0 \to j_0$ *and* $i_1 \to j_1$. Thus, we can use evidence from $i_1 \to j_1$ to reject $i_0 \to j_0$, which increases the power of testing $i_0 \to j_0$.

In light of this observation, it is useful to enumerate $i_0 \to j_0$'s implications, to to obtain additional evidence for testing. Intuitively, the more implications an edge (hypothesis) has (due to propagation rules), the more ways there are to verify this hypothesis, since it only takes one false implication to reject a hypothesis. In the next subsection, we develop an algorithm that recursively enumerates these implications.

5.4.1 Enumeration of implications of an edge orientation

In this subsection, we develop an algorithm, Algorithm 13, for enumerating the "extended hypothesis" implied by the original hypothesis corresponding to the edge orientation of interest, $i \to j$. This algorithm allows us to operationalize the Meek rules and enumerate edges that are implied by the null hypothesis, $i \to j$.

In the algorithm, a tree of edges is recursively expanded to enumerate all the edges implied by the root edge. To emphasize, the tree we refer to in this section does not refer to the causal graph (which need not be a tree), but rather a representation of the logical implications that are implied by the root edge.

Let $T^{i\to j}$ be the tree constructed by applying Algorithm 13. A *path* in a tree T is the set of edges encountered by traversing T from its root to a *leaf* node.

Definition 28. *For a tree T, define the logical implications represented by T as follows:*

$$H_0(T) := \bigcup_{P\in\mathcal{P}(T)} \bigcap_{i'\to j'\in P} H_0^{i'\to j'}.$$

Proposition 18. *Algorithm 13 satisfies the following properties:*

- *(Soundness) Algorithm 13 is sound and does terminate.*

Algorithm 13 Enumerating edges implied by Meek rule for a given edge orientation

```
Require: Essential graph $G$, hypothesized orientation $i \to j$.
Initialize empty tree $T$, insert edge $i \to j$ as root.
while exists root to leaf path $P$ such that the oriented edges in $P$ imply new edge via a Meek rule in $G$ do
    if Meek rule of the form (5.3) or (5.4) propagates a single new edge $i' \to j'$ not in $P$ then
        Append $i' \to j'$ to the leaf node of $P$.
    if Meek rule of the form (5.4) propagates two new edges $i_1 \to j_1$, $i_2 \to j_2$ both not in $P$ then
        Add $i_1 \to j_1$ and $i_2 \to j_2$ as children of the leaf node of $P$. ▷ do not include the pair if at least one of the edges is on the path
Return tree $T$ where each path from root to leaf $P$ is a set of edges that are implied by $i \to j$.
```

- *(Correctness) Let $T^{i \to j}$ be the resultant tree of Algorithm 13, then:*

$$H_0^{i \to j} = \bigcup_{P \in \mathcal{P}(T^{i \to j})} \bigcap_{i' \to j' \in P} H_0^{i' \to j'}.$$

Remark 15. *As we prove in Lemma 36, we can stop the tree expansion in Algorithm 13 after any number of application of Meek rules. The corresponding tree T would still be valid ($H_0(T) = H_0^{i \to j}$). That is, we need not exhaust all implications based on the Meek rules. This is useful because one can trade off between the power of the test, and the time/space complexity of a more complicated tree/test statistic. Testing with fewer implications results in lower power, but has the benefit of being easier to track and evaluate.*

5.4.2 Conversion of expanded hypothesis into an e-process

Having enumerated other edge orientations implied by the original edge orientation, in this subsection, we show how to convert these logical relationships into an "extended" e-process useful for testing.

Given a logical tree $T^{i \to j}$, we first design an e-process corresponding to a particular path $P \in T^{i \to j}$. Let V be the set of all nodes in the graph. Let Δ^d denote the probability simplex on d-dimensions. Let $P(i') \coloneqq \{i \to j : i \to j \in P, i = i' \vee j = i'\}$ be the set of edges on path P with one its vertex node i'. We can now construct a corresponding e-process which is defined as follows.

Proposition 19. *Let $(E_t^{i \to j}(i))$ be an e-process w.r.t. $(\mathcal{F}_t^i)$ under $H_0^{i \to j}$. For a path P, define:*

$$E_t^P \coloneqq \exp\left(\sum_{i' \in V} \max_{i \to j \in P(i')} \log E_t^{i \to j}(i') \right.$$
$$\left. - \frac{|P(i')| - 1}{2} \cdot \log(2|T_{i'}(t)| - 2)\right),$$

Then (E_t^P) is an e-process w.r.t. $(\mathcal{F}_t)$ under H_0^P.

Having defined the e-process corresponding to some path $P \in T^{i \to j}$, we may now define the e-process corresponding the full tree $T^{i \to j}$ as follows.

Proposition 20 (Correctness of combined e-process)**.** *Define:*

$$E_t^{i \to j} := \min_{P \in \mathcal{P}(T^{i \to j})} E_t^P.$$

Then, $(E_t^{i \to j})$ *is an e-process when* $H_0^{i \to j}$ *is true.*

Theorem 22. *For any sequence of interventions* (I_t) *predictable w.r.t.* $(\mathcal{F}_t)$*, let* $\widehat{G}_t$ *be the partially oriented DAG where the test for each orientation is defined as follows:*

$$\varphi_t^{i \to j} = \mathbb{1}\{E_t^{i \to j} \geq |\overline{G}|/\alpha\}.$$

Then, $(\widehat{G}_t)$ *is anytime-valid orientation (as defined in* (5.11)*).*

5.4.3 Additional power in combined test statistics

We note that Proposition 20 applies to any expanded $T^{i \to j}$ tree, which includes the tree without any expansion i.e. $T^{i \to j} = (i \to j)$. So does an expanded tree lead to higher power? We note that an increase in power depends on the graph: for instance, if a graph comprises of only isolated edges, no additional power can be gained from propagation. Below, we present one instance where we can prove that the power of the test is sizably larger, thus providing a concrete example showing the value of combining evidence.

Proposition 21. *Consider an uniform intervention policy over nodes* $[n]$*. There exists a graph and edge* $i \to j$*, such that the expected growth rate (i.e. power) of* $\log E_t^{i \to j}$ *under the fully expanded tree* $T^{i \to j}$ *is* $\Omega(|\overline{G}|)$ *times that of* $\log E_t^{i \to j}$ *under the non-expanded tree (i.e. just the single edge* $i \to j$*).*

Please see Figure 5.15 of Section 5.12 for an illustration of a simple n-node chain graph, wherein additional power can be obtained due to Meek rules. Please refer to Section 5.9 for the proofs of all results in this section as well as time complexity analysis of the proposed algorithms.

In closing, we note that this approach to test statistic combination can apply more broadly to other structured hypothesis testing settings, wherein there are logical relationships (Meek rules in this case) relating the hypotheses.

5.5 Experiments on fixed-time versus anytime methods

To illustrate the usefulness of anytime valid tests, we compare our anytime-valid test statistic (as in Section 5.3.2) against a fixed-time test statistic across a variety of graphs.

Graph Setups: We consider two classes of graphs. (1) Erdos-Renyi graphs with varying number of nodes and density $(n, p) \in \{10, 20, 30\} \times \{0.3, 0.5\}$ (2) tree graphs with $n \in \{10, 20, 50, 100\}$. These are used to generate the graph skeleton. The SCM of the graphs is linear Gaussian; the edge strengths are randomly sampled from $\max(U[0, k], b)$, where k is the upper bound.

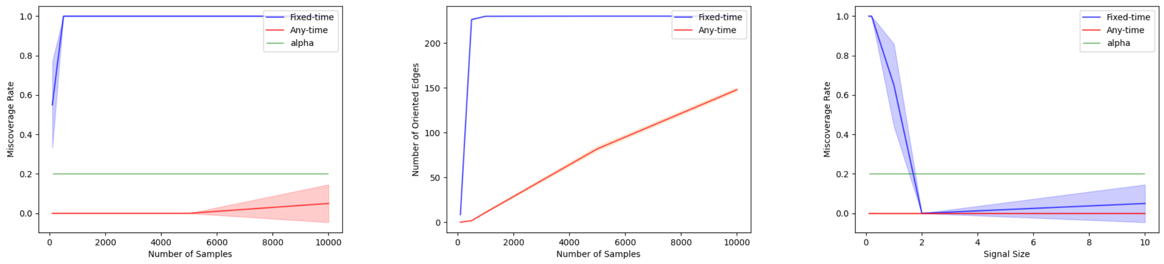


Figure 5.1: (Left) Number of samples vs Miscoverage rate (Middle) Number of samples vs Number of oriented edges (Right) Edge signal size k vs Miscoverage rate.

Fixed-time Baseline: We consider the following p-value that corresponds to the two-sided z-test for edge $i \to j$. The hypothesis test involves checking if the test statistic is below the acceptance threshold $\frac{\alpha}{2|\overline{G}|}$ (from union bound).

Let $\widehat{\mu}_t^{j|\mathrm{do}(i)} := \sum_{k \in T_i(t)} X_k^j, \widehat{\mu}_t^{i|\mathrm{do}(j)} := \sum_{k \in T_j(t)} X_k^i$. Let $T_i(t)$ be the number of times we have intervened on i at time t. Define the fixed-time p-value baseline as:

$$P_t^{i \to j} = 2\left(1 - \Phi\left(\frac{b \cdot |T_i(t)| - |\widehat{\mu}_t^{j|\mathrm{do}(i)}| + |\widehat{\mu}_t^{i|\mathrm{do}(j)}|)}{\sqrt{|T_j(t)| \operatorname{var} X_i + |T_i(t)| \operatorname{var} X_j}}\right)\right)$$

where Φ is the Gaussian CDF function.

Proposition 22 ($P_t^{i\to j}$ is a p-value)**.** *$P_t^{i\to j}$ satisfies $\mathbb{P}(P_t^{i\to j} \leq s) \leq s$ for all $s \in [0,1]$ and $t \in \mathbb{N}$ under $H_0^{i\to j}$.*

Experiment Configurations: In the experiment, we fix $b = 0.1$, variance 1 and the interventional value $\nu = 1$. We vary the number of interventional samples $\in \{100, 500, 1000, 5000, 10000\}$, tolerated error rate $\alpha \in \{0.1, 0.2\}$ and edge strength $k \in \{0.1, 0.2, 1, 2, 10\}$, all of which affect hypotheses testing (i.e. number of orientations). Fixing a particular setting, we simulate 20 trials to compute the mean and standard deviation.

We plot two metrics. The most important is the mis-coverage rate, which is defined to be the number of trials wherein the test statistic returns at least one falsely oriented edge. That is, the percentage of time that an update step that uses this test statistic is wrong. Alongside miscoverage, we also plot the number of oriented edges. This indicates the informativeness of a test statistic, as indeed a test that never rejects can trivially achieve 0 miscoverage rate.

Comparing anytime vs fixed-time: In the interest of space, we present results under the ER graph with $(n, p, \alpha) = (30, 0.5, 0.2)$ in Figure 5.1. Overall, we observe the following trends in our experiments.

Miscoverage: In every setting, we find that our testing framework achieves miscoverage rate below α (line in green), thus validating our theoretical anytime guarantee. On the other hand, in a number of settings, we observe that the fixed-time statistic leads to high miscoverage rate.

Number of Orientations: The reason for the high miscoverage seems to be that the fixed time test statistic is not conservative enough to control the error rate. The anytime test is more conservative in orienting fewer number of edges, so as to attain error control. Note that this control is important in preventing spurious edge orientations, which would then be fed back into the query step as an erroneous representation of the partially oriented graph.

Comparing combined e-values vs base e-values: We also conduct an experiment comparing the combined e-values (Section 5.4) against the base e-values (Section 5.3) in a chain graph, where we expect the combined e-values to be helpful. We find that combining e-values is more useful in large data/graph regimes, while the light-weight, base e-values are more effective in small data/graph regimes.

Please refer to Section 5.10 for all experimental results.

5.6 Optimizing test statistic for causal verification

Once we have an anytime valid test framework that correctly implements the update step, we can turn to designing query strategies that minimize sample complexity under this framework. Towards this goal, we consider the task of causal verification, which acts as a stepping-stone towards causal discovery. Knowing how to optimally intervene to verify a known graph is an useful building block for understanding how to optimally intervene to learn an unknown graph. In this section, we develop a novel querying algorithm with provable guarantees that we believe can be be a stepping stone to more practical algorithms. To do so, we highlight a connection between finite-sample causal verification and the structured bandit literature [23], by demonstrating that causal verification reduces to multi-constraint bandit optimization.

To recap, the goal of active verification is: given knowledge of the true graph, verify the edge orientations, while minimizing the *expected* number of samples needed to conclude that each edge orientation is oriented as in the graph w.h.p.

Problem Setup: Formally, construct an intervention policy that (adaptively) intervenes on nodes $I_1, ..., I_\tau$ such that the the *expected* stopping time $\mathbb{E}[\tau]$ is minimized, where τ is defined as the earliest time step such that every hypothesis corresponding to incorrect orientation $j \to i$ is rejected. That is, $\forall j \to i$, $E_\tau^{j \to i} \geq |\overline{G}|/\alpha$.

5.6.1 Construction of test statistic for causal verification

Since the SCM is known in verification, all edge strengths are known. This allows us to construct a more simplified test-statistic than that of Proposition 19.

Consider some incorrect orientation $j \to i$. Let its logical tree be $T^{j \to i}$. With full information, it is natural to construct a test statistic for $j \to i$ by including only the e-value with the *highest expected growth* rate. Define $S_t^*(P, I_t) = S_t^{e^*, s^*}(I_t)$ for $e^*, s^* = \text{argmax}_{e \in P(I_t), s \in \{\pm\}} \mathbb{E}[S^{e,s}(I_t)]$. This represents the edge and sign e-value with the largest expected growth-rate under intervention I_t, out of all the possible e-values of edges in $P(I_t)$.

With this, we may define a test statistic with the highest expected growth rate under intervention i' as $E_t^{*j \to i}(i') = \prod_{k: I_k = i'}^t S_k^*(P, i')$, path test statistic as $E_t^{*P} := \exp\left(\sum_{i' \in V} \log E_t^{*j \to i}(i')\right)$ and full test statistic as $E_t^{*j \to i} = \min_{P \in \mathcal{P}(T^{j \to i})} E_t^{*P}$. In what follows, we will make the assumption that X_i is a bounded r.v. with b, ν such that $\log S_k^*(P, I)$ is positive (as arm rewards are usually assumed to be positive in bandits literature).

5.6.2 Reduction to multi-constraint bandit optimization

Having defined E_t^{*P}, causal verification then corresponds to choosing an apt intervention policy that jointly optimizes $E_t^{*j \to i}$ for every incorrect orientation $j \to i$, and only insofar as to have $E_t^{*j \to i}$ *exceed a threshold*, $|\overline{G}|/\alpha$. To solve this problem, we observe that causal verification reduces to multi-constraint bandit optimization, defined as follows.

Multi-constraint bandit optimization: An instance is parameterized by n arms, m constraints and budget b:

- There are T rounds for T a specified time horizon.
- At round i, the algorithm may pull an arm x_i, yielding a "gain" *vector*, where $r_{x_i} \sim D_{x_i}$ for $r_{x_i} \in [0, M]^m$.
- There is a known threshold $b \in \mathbb{R}^+$ on the aggregate gain of each constraint.
- The interaction terminates at the earliest round τ, when $\sum_{t=1}^{\tau} r_{x_t} \geq b \cdot 1$ (aggregate gain of every constraint exceeds b), or at the end of the Tth round.

The goal of the algorithm is to minimize the total *expected* cost $\sum_{i=1}^{\tau} c_{x_i}$ (node intervention cost c_{x_i} is set to 1).

Reduction to multi-constraint bandits: We observe that the test statistic for each path $P \in \mathcal{P}(T^{j \to i})$ grows *additively* in the log of e-values $\log S_k^*(P, I_k)$:

$$\begin{aligned} E_t^{*j \to i} \geq |\overline{G}|/\alpha &\Leftrightarrow \forall P \in \mathcal{P}(T^{j \to i}), E_t^{*P} \geq |\overline{G}|/\alpha \\ &\Leftrightarrow \forall P \in \mathcal{P}(T^{j \to i}), \sum_{k=1}^{t} \log S_k^*(P, I_k) \geq \log(|\overline{G}|/\alpha) \end{aligned}$$

Thus, given a causal verification instance, we may instantiate a multi-constraint bandit instance as follows:

1. Arms: define $n = |V|$ arms, each corresponding to a node intervention in the graph.
2. Constraint: define a constraint corresponding to every (P, i') pair, for path $P \in T^{j \to i}$ and intervention $i' \in V$. Thus, the gain of pulling arm $i' \in V$ corresponds to a vector of realizations of random variable $\log S^*(P, i')$ of every path $P \in T^{j \to i}$ of every tree $T^{j \to i}$.
3. Set the threshold $b = \log(|\overline{G}|/\alpha)$.

Guarantee: We develop Algorithm 14 that attains provable guarantees in the multi-constraint bandit setting, which applies immediately to the causal verification setting via the reduction. Let OPT be the expected total number of interventions needed by the optimal dynamic policy. Let REW_{tot} be the algorithm performance of Algorithm 14, which is the expected number of interventions such that every incorrect orientation test statistic exceeds b. Then, we have that:

Algorithm 14 Causal Verification as multi-constraint bandits

```
Require: threshold $b$; time horizon $T$; for each node $x$, known expected gain vector $\bar{r}_x \in [0, M]^m$
    ▷ for node $x$, this vector's entries are the expected growth rates under intervention on node $x$ ($\mathbb{E}[\log S^*(P, x)]$ of every path $P$ of every logical tree $T^{j\to i}$)
In the first $n$ rounds, intervene on each node once
Initialize $v_1 = 1 \in [0, 1]^m$
Set $\epsilon = \sqrt{\frac{M \ln m}{b+M}}$
while [ donot all tests have concluded, since not all test statistics have exceeded $b$]$\sum_{i=1}^{t} r_{x_i} < b \cdot 1$ and $t < T$
    for node $x \in [n]$ do
        Set weighted total gain $g_x = \bar{r}_x \cdot v_t$
    Intervene on node $x_t = \text{argmax}_{x \in X}\, g_x$ with the highest weighted gain
    Receive vector $r_x$, whose entries are realizations of random variables $\log S^*(P, x)$ of every path $P \in T^{j\to i}$ of every tree $T^{j\to i}$
    Update $v_t$ entry-wise with normalized $r_x$, where its $i$th entry changes as follows:
        $$v_{t+1}(i) = v_t(i)(1-\epsilon)^{\ell}, \ell = r_x(i)/M$$
```

Theorem 23. *The regret of Algorithm 14 is:*

$$\text{REW}_{tot} - \text{OPT} \leq \tilde{O}\left(\frac{M}{b} + \sqrt{\frac{(b+M)M}{b}}\right)\text{OPT} + \tilde{O}\left(\frac{M\sqrt{T}}{b} + nM\right).$$

To provide some intuition, in Algorithm 14, one may view v as a varying, weighting over each constraint. Each round, the algorithm greedily pulls the arm whose sum of weighted expected gain is the largest. After a round, if a constraint has seen a sizable increase, then its weighting in v is reduced. This adaptive re-balancing then allows for an arm selection that focuses more on increasing other constraints, which are further away from exceeding the threshold b. Please refer to Section 5.11 for the proofs of all results in this section.

5.7 More Related Works

Finite-Sample Considerations in Causality: The three papers most similar in motivation to that of ours are: Greenewald et al. [125], Wadhwa and Dong [287] and Acharya et al. [3]. Like Greenewald et al. [125], our paper is similarly motivated by finite-sample considerations that exist in real-world settings, where the collection of interventional data (e.g. RCTs) is much more difficult and costly than that the collection of observational data. As such, we also assume that infinitely many observational samples are available, while only finitely many interventional samples can be obtained. [287] is concerned with the sample complexity of causal discovery, albeit that of learning the equivalence class, and not the actual graph, given only observational data only. Finally, [3] is also concerned with finite-sample causal discovery via testing. They study the two node setting, and assume both finite interventional and observational data, which are contrasted in the paper.

While our paper's goal of studying finite-sample causal discovery is the same as those of Acharya et al. [3], Greenewald et al. [125], our paper differs in focusing primarily on the update step. Additionally, our testing framework is applicable in general graphs, going beyond the two-node or tree settings. Different from [125], we study soft interventions instead of hard interventions, thus introducing the need to consider the strength of edges, as edges with weak causal strength require more samples to orient. Different from [3], we study how to propagate edge orientations in hypothesis testing, which is needed when the graph comprises of more than two nodes and a single edge.

Causal Verification: Causal Verification is a well-known task in causal discovery. Besides having practical applications (e.g. verifying a scientific conjecture corresponding to some causal graph structure), it has the theoretical benefit of better understanding the lower bound that underlies any active causal discovery algorithm [70, 232, 266].

Bayesian causal discovery: There has also been a line of work in Bayesian causal discovery, wherein one uses interventional data to update the posterior over all graphs [6, 277, 279]. Since the set of all graphs in the MEC may be prohibitively large, approximation methods are used to sample from the posterior, making less clear what provable guarantees one may be able to provide about such methods.

Functional Causal Discovery: Further afield, there has been a sizable number of paper that leverage specific functional forms of graphs for orientation, using observational data only. Examples of such methods include [141, 260, 321]. Interested readers may refer to for example [120] for a more complete survey of this line of work.

Bandit Multiple-Testing: The closest type of methods in the bandit literature are those dealing with multiple testing [154, 301]. Current work on bandit multiple testing differs from the methods in this paper in two significant ways: (1) bandit multiple testing is primarily focused on controlling the false discovery rate (FDR) and (2) methods lie in the typical hypothesis testing problem setting where one can only reject a hypothesis — in the causal discovery setting, each unoriented edge will be one of two directions, and the negation of one implies the other — hence the relationships between the hypotheses require methods that will derive a certain conclusion for each unoriented edge.

Necessity of Non-Negative Martingales: [234] proves that under a suitable definition of admissibility, all admissible constructions of test statistics for any-time sequential inference must necessarily utilize nonnegative martingales. This shows that the martingale test statistic we construct is in some sense of the "right form".

5.8 Deferred Proofs from Section 5.3

Lemma 34. *Let* $M_t := \prod_{k=1}^{t} S_k$. *Then,* (M_t) *is an NSM w.r.t. filtration* $(\mathcal{F}_t)$ *under* H_0.

Proof. The conditional expectation of M_t is as follows:

$$\mathbb{E}[M_t \mid \mathcal{F}_{t-1}] = \mathbb{E}[S_t \mid \mathcal{F}_{t-1}] \cdot \prod_{k=1}^{t-1} E_k = \mathbb{E}[S_t \mid \mathcal{F}_{t-1}] \cdot M_{t-1} \leq M_{t-1},$$

where the inequality is by definition of S_t being a sequential e-value for each $t \in \mathbb{N}$. □

5.8.1 Deferred Proofs from Section 5.3.1

Proposition 23. $(\varphi_t^{i\to j})$ *is an* anytime-valid test, *that is, the procedure in* (5.2) *ensures that*

$\mathbb{P}(H_0^{i\to j}$ *is rejected* $) = \mathbb{P}($*exists* $t \in \mathbb{N} : \varphi_t^{i\to j}(\alpha) = 1) \leq \alpha$ *when* $H_0^{i\to j}$ *is true for all* $\alpha \in [0, 1]$.

Proof.

$$\mathbb{P}(\text{exists } t \in \mathbb{N} : \varphi_t^{i\to j}(\alpha) = 1 | H_0^{i\to j}) = \Pr(\text{exists } t \in \mathbb{N} : M_t^{i\to j} \geq 1/\alpha | H_0^{i\to j}) \qquad \text{(by definition of } \varphi_t^{i\to j}(\alpha))$$

$$\leq \alpha \qquad \text{(Ville's inequality, because under } H_0^{i\to j} \text{ is true} \Rightarrow (M_t^{i\to j}) \text{ is an e-process)}$$

□

Proposition 24. *Given an anytime-valid test* $(\varphi_t^{i\to j})$*, orient edge* $i \to j$ *in* $\widehat{G}_t$ *the first time* $\varphi_t^{j\to i}(\alpha/|\overline{G}|) = 1$*. Then,* $(\widehat{G}_t)$ *is an anytime-valid partially oriented graph.*

Proof. Let the final oriented graph be $\hat{G}$.

$$\begin{aligned}
&\mathbb{P}\left(\text{exists } t \in \mathbb{N} : \text{exists oriented edge in } \widehat{G}_t \text{ not in } G^*\right) \\
&\leq \sum_{i\to j \in \hat{G}} \mathbb{P}\left(\text{exists } t \in \mathbb{N} : \text{orient edge } i \to j \wedge j \to i \text{ in } G^*\right) \\
&= \sum_{i\to j \in \hat{G}} \Pr(\text{exists } t \in \mathbb{N} : \phi_t^{j\to i}(\alpha/|\overline{G}|) = 1 \wedge j \to i \text{ in } G^*) \\
&\leq \sum_{i\to j \in \hat{G}} \alpha/|\bar{G}| \qquad \text{(by Proposition 11)} \\
&= \alpha \qquad (5.5)
\end{aligned}$$

□

5.8.2 Deferred Results from Section 5.3.2

Proposition 25. *For any sequence* (λ_t) *that is predictable w.r.t.* $(\mathcal{F}_t^j)$*,* $S_t^{i\to j,+}(j)$ *and* $S_t^{i\to j,-}(j)$ *are both sequential e-values under* $H_0^{i\to j}$ *w.r.t. filtration* $(\mathcal{F}_t^j)$*.*

Proof. At time t with $I_t = j$, under $H_0^{i\to j}$, we have that $\pm X_t^i \mid \mathcal{F}_{t-1}^j$ is a mean 0, σ_i^2-sub-Gaussian random variable. We work through the X_t^i case, and the $-X_t^i$ case follows analogously. From definition, its MGF is such that:

$$\mathbb{E}[\exp(\lambda X_t^i)] \leq \exp\left(\frac{\lambda^2\sigma_i^2}{2}\right) \Leftrightarrow \mathbb{E}[S_t^{i\to j,+}(j) \mid \mathcal{F}_{t-1}^j] = \mathbb{E}[S_t^{i\to j,+}(j)] = \exp\left(\lambda X_t^i - \frac{\lambda^2\sigma_i^2}{2}\right) \leq 1$$

□

Proposition 26. *Under the minimal causal effect condition, we have the following:*
Under $H_0^{i\to j,+}$, $S_t^{i\to j,+}(i)$ are sequential e-values w.r.t. filtration $(\mathcal{F}_t^i)$.
Under $H_0^{i\to j,-}$, $S_t^{i\to j,-}(i)$ are sequential e-values w.r.t. filtration $(\mathcal{F}_t^i)$.

Proof. We prove the first statement, and the second follows analogously. WLOG $\mu_j(i) = \mathbb{E}[X_t^j(i)] \geq b$. We have that:

$$\mathbb{E}\left[\exp\left(\lambda(b - X_t^j(i)) - \lambda^2\sigma_j^2/2\right) \mid \mathcal{F}_{t-1}\right] \leq 1 \Leftrightarrow \mathbb{E}[S_t^{i\to j,+}(i) \mid \mathcal{F}_{t-1}^i] = \mathbb{E}[S_t^{i\to j,+}(i)] \leq 1$$

since $b - X_t^j(i) \mid \mathcal{F}_{t-1}$ is a σ_i^2-sub-Gaussian with nonpositive mean $b - \mathbb{E}[X_t^j(i)]$.

□

Proposition 27. *Under $H_0^{i\to j}$, the following processes are e-processes w.r.t. to filtration $(\mathcal{F}_t^j)$, filtration $(\mathcal{F}_t^i)$ respectively:*

$$E_t^{i\to j}(j) = \frac{1}{2}\left(\prod_{k:I_k=j}^{t} S_k^{i\to j,-}(j) + \prod_{k:I_k=j}^{t} S_k^{i\to j,+}(j)\right), E_t^{i\to j}(i) = \min\left(\prod_{k:I_k=i}^{t} S_k^{i\to j,-}(i), \prod_{k:I_k=i}^{t} S_k^{i\to j,+}(i)\right)$$

Proof.

- $(E_t^{i\to j}(j))$ is the average of two processes

$$M_t^+(j) = \prod_{k\in T_j(t)} S_k^{i\to j,+}(j), \qquad M_t^-(j) = \prod_{k\in T_j(t)} S_k^{i\to j,-}(j).$$

By Proposition 13, each of these processes are the product of sequences of sequential e-values (w.r.t. to filtration $(\mathcal{F}_t^j)$) under $H_0^{i\to j}$, i.e., $(S_k^{i\to j,-})$ and $(S_k^{i\to j,+})$. This implies that they are NSMs by Lemma 34, and hence also e-processes w.r.t. to filtration $(\mathcal{F}_t^j)$.
To show that the average of these two e-processes is an e-process, we introduce the notion of a stopping time, and note the following e-process equivalence.

Definition 29. *A* stopping time $\tau \in \mathbb{N}$ *w.r.t. a filtration $(\mathcal{F}'_t)_{t\in\mathbb{N}}$ is a random variable that where $\mathbb{1}\{\tau = t\}$ is measurable w.r.t. $\mathcal{F}_t$.*

Further, we use the following fact about e-processes from [234].

Fact 4 (Item (vi) from Lemma 6 of Ramdas et al. 234)**.** *(E_t) is a an e-process w.r.t. to a filtration $(\mathcal{F}'_t)$ iff it is nonnegative and $\mathbb{E}[E_\tau] \leq 1$ for all stopping times τ defined w.r.t. $(\mathcal{F}'_t)$.*

Now, we get that, for any stopping time τ defined w.r.t. to filtration $(\mathcal{F}_t^j)$:

$$\mathbb{E}[E_\tau^{i\to j}(j)] = \frac{1}{2}(\mathbb{E}[M_\tau^+(j)] + \mathbb{E}[M_\tau^-(j)]) \leq 1,$$

where the last inequality is by $(M_t^+(j)), (M_t^-(j))$ being NSMs defined w.r.t. to filtration $(\mathcal{F}_t^j)$.

- Now, we will prove $(E_t^{i\to j}(i))$ is also an e-process. Since $H_0^{i\to j} \Rightarrow H_0^{i\to j,+} \cup H_0^{i\to j,-}$, if $H_0^{i\to j}$ is true, one of $H_0^{i\to j,+}$ or $H_0^{i\to j,-}$ holds. Without loss of generality, let $H_0^{i\to j,+}$ be true. Here, the processes under consideration are now:

$$M_t^+(i) = \prod_{k\in T_j(t)} S_k^{i\to j,+}(i), \qquad M_t^-(i) = \prod_{k\in T_j(t)} S_k^{i\to j,-}(i).$$

We will show that $M_t^+(i)$ is an NSM w.r.t. to filtration $(\mathcal{F}_t^i)$, which implies that $M_t^{i\to j}$ is an e-process since $M_t^{i\to j} \leq M_t^+(i)$ for all $t \in \mathbb{N}$ almost surely.
When $I_t = i$, by Proposition 14, $S_t^{i\to j,+}(i)$ is an e-value and so:

$$\mathbb{E}[M_t^+(i)|\mathcal{F}_{t-1}] = \mathbb{E}[S_t^{i\to j,+}(i)|\mathcal{F}_{t-1}] \cdot M_{t-1}^+ \leq M_{t-1}^+.$$

Finally, we check that when $I_t \neq i$, we have that:

$$\mathbb{E}[M_t^+(i)|\mathcal{F}_{t-1}] = M_{t-1}^+ \leq M_{t-1}^+.$$

And we note that at the base case $t = 1$, for the NSM, we have that:

$$\mathbb{E}[M_t^+(i)] = \mathbb{E}[S_1^{i\to j,+}(i)] \leq 1 \text{ or } \mathbb{E}[M_t^+(i)] = 1$$

□

5.8.3 Deferred Results from Section 5.3.3

Proposition 28. *Suppose the true edge orientation is actually that $j \to i$ and WLOG $\mu^i(j) > 0$. By setting $\lambda = b/\sigma_i^2$ for $S_t^{i\to j}(i)$ and $\lambda = b/\sigma_j^2$ for $S_t^{i\to j}(i)$, we have the following growth rates:*

1. $\mathbb{E}[\log S_t^{i\to j,+}(j) \mid \mathcal{F}_{t-1}] = b(\mu_i(j) - b/2)/\sigma_i^2$
2. $\mathbb{E}[\log S_t^{i\to j,+}(i) \mid \mathcal{F}_{t-1}] = \mathbb{E}[\log S_t^{i\to j,-}(i) \mid \mathcal{F}_{t-1}] = b^2/(2\sigma_j^2)$

Proof. We analyze the growth rates of each case separately:

1.

$$\begin{aligned}\mathbb{E}[\log S_t^{i\to j}(j) \mid \mathcal{F}_{t-1}] &= \lambda\left(\mathbb{E}\left[X_t^i \mid \mathcal{F}_{t-1}\right]\right) - \frac{\lambda^2\sigma_i^2}{2}\\ &= \frac{b}{\sigma_i^2}\mu_i(j) - \frac{b^2}{2\sigma_i^2}\\ &= \frac{b(\mu_i(j) - b/2)}{\sigma_i^2}\end{aligned}$$

2. We have that:

$$\begin{aligned}\mathbb{E}[\log S_t^{i\to j,\pm}(i) \mid \mathcal{F}_{t-1}] &= \lambda(b \pm \mathbb{E}[X_t^j \mid \mathcal{F}_{t-1}]) - \lambda^2\sigma_j^2/2 \\ &= \lambda b - \lambda^2\sigma_j^2/2 \\ &= \frac{b^2}{2\sigma_j^2}.\end{aligned}$$

□

Remark 16. *We note that* $\mathrm{var}(X_i)$ *in any interventional distribution is identified, and the same as* $\mathrm{var}_{\mathcal{D}_0}(X_i)$. *This allows us to put in the exact multiplier for* $\lambda^2/2$ *in the the NSM.*

From Linear Graphs to Additive Graphs: We note that our setting may be generalized to additive graphs, when given an upper bound on the variance of variables in the interventional.

This is because, to set the appropriate λ for sequential e-values, we only need to have knowledge of b and an upper bound on the variance interventional distribution. With this, we could set a rate such that the growth rate is positive as in the power analysis above.

Proposition 29. *If the edge* $j \to i$ *is the true orientation in G, then each of the the following statements hold true with probability* $1-\beta$ *for each* $\beta \in [0,1]$*:*

1. *For* $(S_t^{i\to j,+}(j))$, *we have that* $\tau_\alpha \leq \frac{\sigma_i^2 \log(\alpha^{-1}\beta^{-1})}{b(\mu_i(j)-b)}$.
2. *For* $(S_t^{i\to j,\pm}(i))$, *we have that* $\tau_\alpha \leq \frac{\sigma_j^2 \log(\alpha^{-1}\beta^{-1})}{b^2}$

Proof. We prove this explicitly for $S_t^{i\to j,+}(j)$ and other results for $(S_t^{i\to j,\pm}(i))$ follow similarly.

Let $M_t \coloneqq \prod_{k=1}^{t} S_k^{i\to j,+}(j)$ as follows.

$$\begin{aligned}M_t &= \exp\left(\sum_{k=1}^{t} \lambda X_t^i - \frac{\lambda^2\sigma_i^2}{2}\right) \\ &= \exp(t(\lambda\mu_i(j) - \lambda^2\sigma_i^2)) \cdot \exp\left(\sum_{k=1}^{t} \lambda(X_t^i - \mu_i(j)) + \frac{\lambda^2\sigma_i^2}{2}\right).\end{aligned}$$

Now, we note that $\exp\left(\sum_{k=1}^{t} -\lambda(X_t^i - \mu_i(j)) - \frac{\lambda^2\sigma_i^2}{2}\right)$ is a nonnegative supermartingale since $X_t^i - \mu_i(j)$ are i.i.d. σ_i^2-sub-Gaussian random variables with mean 0. As a result, we know that

$$M_t \geq \exp(t(\lambda\mu_i(j) - \lambda^2\sigma_i^2)) \cdot \beta$$

for all $t \in \mathbb{N}$ with probability $1-\beta$ by Ville's inequality. If we set $\lambda = b/\sigma_i^2$. We get that

$$\frac{\sigma_i^2 \log(\alpha^{-1}\beta^{-1})}{b(\mu_i(j)-b)} \leq t.$$

implies $M_t \geq \alpha^{-1}$ with probability $1-\beta$. This concludes our desired result. □

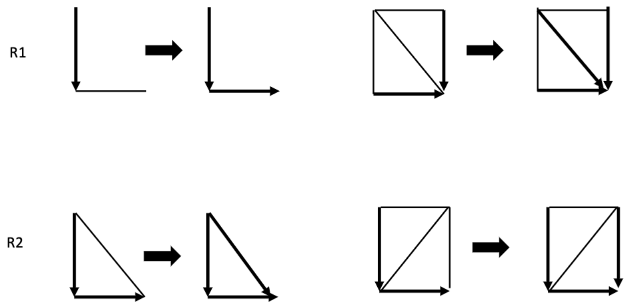


Figure 5.2: The four meek rules for propagating oriented edges.

5.9 Deferred Proofs from Section 5.4

Lemma 35 (Meek rules imply hypothesis conjunction/disjunction (general))**.** *For any edge orientation hypotheses $H_0^{i\to j}, H_0^{i_1\to j_1}, H_0^{i_2\to j_2}$, we have that*

$$\begin{aligned}
&H_0^{i\to j} = H_0^{i\to j} \cap H_0^{i_1\to j_1} && \textit{if } i \to j \Rightarrow i_1 \to j_1 \\
&H_0^{i\to j} \cap H_0^{i_1\to j_1} = H_0^{i\to j} \cap H_0^{i_1\to j_1} \cap H_0^{i_2\to j_2} && \textit{if } i \to j \wedge i_1 \to j_1 \Rightarrow i_2 \to j_2 \\
&H_0^{i\to j} = H_0^{i\to j} \cap (H_0^{i_1\to j_1} \cup H_0^{i_2\to j_2}) && \textit{if } i \to j \Rightarrow i_1 \to j_1 \vee i_2 \to j_2
\end{aligned}$$

Proof. The results follow from an application of the logical rule that if $A \Rightarrow B$, then $A = A \cap B$.

For the first rule, we get the following implications:

$$H_0^{i\to j} \Leftrightarrow i \to j \text{ in } G^* \Rightarrow i_1 \to j_1 \text{ in } G^* \Leftrightarrow H_0^{i_1\to j_1}.$$

For the second rule, we can show its true by the following derivation.

$$H_0^{i\to j} \cap H_0^{i\to j} \Leftrightarrow i \to j \text{ and } i_1 \to j_1 \text{ in } G^* \Rightarrow i_2 \to j_2 \text{ in } G^* \Leftrightarrow H_0^{i_2\to j_2}.$$

For the last rule, we can derive the implication as follows:

$$\begin{aligned}
H_0^{i\to j} &\Leftrightarrow i \to j \text{ in } G^* \\
&\Rightarrow i_1 \to j_1 \text{ in } G^* \vee i_2 \to j_2 \text{ in } G^* \\
&\Leftrightarrow H_0^{i_1\to j_1} \cup H_0^{i_2\to j_2}.
\end{aligned}$$

□

5.9.1 Deferred Results from Section 5.4.1

Let $\mathcal{P}(T)$ denote the set of paths in T. The following lemma proves the correctness of the "extended hypothesis" generated by Algorithm 13.

Lemma 36. *Given some tree T, let T' be the tree that results from applying a single Meek rule to T, i.e., through either Line 4 or Line 6 in Algorithm 13. Then, $H_0(T) = H_0(T')$.*

Proof. We perform a case analysis depending on the Meek rule (as defined in (5.3), (5.4), (5.4)) that is applied to T. Let the path in T that is expanded be $\hat{P}$.

1. In the case of (5.3) or (5.4), there exists a single path $P' = \hat{P} \cup \{i' \to j'\} \in \mathcal{P}(T')$ such that

$$\mathcal{P}(T') = \mathcal{P}(T) \setminus \{\hat{P}\} \cup \{P'\},$$

i.e., the only difference between T and T' is that path $\hat{P}$ gained a child $i' \to j'$ to become P'. We have that:

$$\bigcap_{i\to j \in \hat{P}} H_0^{i\to j} = \left(\bigcap_{i\to j \in \hat{P}} H_0^{i\to j} \right) \cap H_0^{i'\to j'} = \bigcap_{i\to j \in P'} H_0^{i\to j}. \tag{5.6}$$

where the first equality is from Lemma 33. Hence, we get

$$\begin{aligned} H_0(T') = \bigcup_{P \in \mathcal{P}(T')} \bigcap_{i\to j \in P} H_0^{i\to j} &= \left(\bigcup_{P \in \mathcal{P}(T') \setminus \{P'\}} \bigcap_{i\to j \in P} H_0^{i\to j} \right) \cup \left(\bigcap_{i\to j \in P'} H_0^{i\to j} \right) \\ &\overset{\text{(a)}}{=} \left(\bigcup_{P \in \mathcal{P}(T) \setminus \{\hat{P}\}} \bigcap_{i\to j \in P} H_0^{i\to j} \right) \cup \left(\bigcap_{i\to j \in \hat{P}} H_0^{i\to j} \right) \\ &= H_0(T). \end{aligned}$$

where equality (a) is by $\mathcal{P}(T') \setminus \{P'\} = \mathcal{P}(T) \setminus \{\hat{P}\}$ and (5.6).

2. In the case of (5.4), we know that there exist two paths $P'_1, P'_2 \in \mathcal{P}(T')$ such that $P'_1 = \hat{P} \cup \{i'_1 \to j'_1\}$ and $P'_2 = \hat{P} \cup \{i'_2 \to j'_2\}$, where $\hat{P} \in \mathcal{P}(T)$ and $\mathcal{P}(T') = \mathcal{P}(T) \setminus \{\hat{P}\} \cup \{P'_1, P'_2\}$. Further, by Lemma 33, we know that:

$$\bigcap_{i\to j \in \hat{P}} H^{i\to j} \Rightarrow H^{i'_i \to j'_1} \cup H^{i'_2 \to j'_2}.$$

Hence, we get the following equality (using the logical relation $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$):

$$\bigcap_{i\to j \in \hat{P}} H_0^{i\to j} = \left(\bigcap_{i\to j \in \hat{P}} H_0^{i\to j} \right) \cap \left(H_0^{i'_1 \to j'_1} \cup H_0^{i'_2 \to j'_2} \right) = \left(\bigcap_{i\to j \in P'_1} H_0^{i\to j} \right) \cup \left(\bigcap_{i\to j \in P'_2} H_0^{i\to j} \right). \tag{5.7}$$

From this, we obtain

$$\begin{aligned}
H_0(T') &= \left(\bigcup_{P \in \mathcal{P}(T) \setminus \{P_1', P_2'\}} \bigcap_{i \to j \in P} H^{i \to j} \right) \cup \left(\bigcap_{i \to j \in P_1'} H^{i \to j} \right) \cup \left(\bigcap_{i \to j \in P_2'} H^{i \to j} \right) \\
&= \left(\bigcup_{P \in \mathcal{P}(T) \setminus \hat{P}} \bigcap_{i \to j \in \hat{P}} H^{i \to j} \right) \cup \left(\bigcap_{i \to j \in P_1'} H^{i \to j} \right) \cup \left(\bigcap_{i \to j \in P_2'} H^{i \to j} \right) \\
&\qquad\qquad (\text{since } \mathcal{P}(T') \setminus \{P_1', P_2'\} = \mathcal{P}(T) \setminus \{\hat{P}\}) \\
&= \left(\bigcup_{P \in \mathcal{P}(T) \setminus \hat{P}} \bigcap_{i \to j \in \hat{P}} H^{i \to j} \right) \cup \left(\bigcap_{i \to j \in \hat{P}} H^{i \to j} \right) \qquad (\text{by } (5.7)) \\
&= H_0(T).
\end{aligned}$$

□

Proposition 30. *Algorithm 13 satisfies the following properties.*

- *(Soundness) Algorithm 13 is sound and does terminate.*
- *(Correctness) Let $T^{i \to j}$ be the resultant tree of Algorithm 13, then:*

$$H_0^{i \to j} = \bigcup_{P \in \mathcal{P}(T^{i \to j})} \bigcap_{i' \to j' \in P} H_0^{i' \to j'}.$$

Proof. **Soundness:** We note that a Meek rule cannot introduce a novel edge to a path in the tree if the path is of length $|E|$ and already contains an orientation for each possible edge in G. And so, the algorithm must terminate since each root to leaf path's length is bounded. This in turn means that so is the depth of the final tree $T^{i \to j}$.

Correctness: This follows from Lemma 36 that each added edge(s) maintains the invariant that the logical expression corresponding to the tree is equal to $H_0^{i \to j}$.

□

5.9.2 Deferred Proofs from Section 5.4.2

Proposition 31. *Let $(E_t^{i \to j}(i))$ be an e-process w.r.t. $(\mathcal{F}_t^i)$ under $H_0^{i \to j}$. For any path $P \subseteq \mathfrak{E}$ define*

$$E_t^P := \exp\left(\sum_{i' \in V} \max_{i \to j \in P(i')} \log E_t^{i \to j}(i') \right.$$
$$\left. - \frac{|P(i')| - 1}{2} \cdot \log(2|T_{i'}(t)| - 2) \right),$$

Then (E_t^P) is an e-process w.r.t. $(\mathcal{F}_t)$ under H_0^P.

We are able to justify that (E_t^P) is an e-process by constructing an NSM that upper bounds E_t^P. We begin with the following fact.

Fact 5 (Theorem 2 of Cover and Ordentlich [79])**.** *Define $\mathbf{x}_t \in \mathbb{R}_+^d$ to be a d-dimensional nonnegative real vector for each $t \in \mathbb{N}$. Then, there exists a sequence of weight vectors, $(\mathbf{w}_t)$, where $\mathbf{w}_t \in \Delta^d$ and $\mathbf{w}_t$ is solely a function of $(\mathbf{x}_k)_{k\in[t-1]}$ for each $t \in \mathbb{N}$, such that*

$$\log\left(\prod_{k=1}^{t} \mathbf{w}_k^\top \mathbf{x}_t\right) \geq \max_{\mathbf{w}\in\Delta^d} \log\left(\prod_{k=1}^{t} \mathbf{w}^\top \mathbf{x}_t\right) - \frac{d-1}{2} \cdot \log(2(t+1)) \textit{ for all } t \in \mathbb{N}.$$

Proof. For each e-process $E_t^{i\to j}(i')$, let $M_t^{i\to j}(i')$ be the corresponding $(\mathcal{F}_t^{i'})$-NSM (under $H_0^{i\to j}$) such that $E_t^{i\to j}(i') \leq M_t^{i\to j}(i')$ for all $t \in \mathbb{N}$ almost surely. Now, define

$$\Delta M_t^{i\to j}(i') := \begin{cases} M_t^{i\to j}(i') & \text{if } t = 1 \\ 1 & \text{if } M_{t-1}^{i\to j}(i') = 0 \\ \frac{M_t^{i\to j}(i')}{M_{t-1}^{i\to j}(i')} & \text{otherwise} \end{cases}.$$

For $t \geq 2$, we have that:

$$\mathbb{E}[\Delta M_t^{i\to j}(i') | \mathcal{F}_{t-1}^{i'}] = \frac{\mathbb{E}[M_t^{i\to j}(i') | \mathcal{F}_{t-1}^{i'}]}{M_{t-1}^{i\to j}(i')} \leq 1$$

as a result of $M_t^{i\to j}(i')$ being an NSM. And so, $(\Delta M_t^{i\to j}(i'))$ is a sequence of sequential e-values with respect to the filtration $\mathcal{F}_t^{i'}$. And $\mathbb{E}[M_1^{i\to j}(i')] \leq 1$ by Definition 25.

Furthermore, we use the following lemma.

Lemma 37. *$M_t^{i\to j}(i')$ is a NSM and $(\Delta M_t^{i\to j}(i'))$ is a sequence of sequential e-values under $(\mathcal{F}_t)$ as well.*

Proof. The filtration $(\mathcal{F}_t^{i'})$ is important here, since this implies that

$$M_t^{i\to j}(i') = M_{t-1}^{i\to j}(i') \text{ and } \Delta M_t^{i\to j}(i') = 1 \text{ if } I_t \neq i', \tag{5.8}$$

as $M_t^{i\to j}(i')$ is $\mathcal{F}_t^{i'}$-measurable (i.e., a function of samples from i') for each $t \in \mathbb{N}$.

Note that for each $t \in \mathbb{N}$,

$$X_t \perp\!\!\!\perp \mathcal{F}_{t-1} \mid I_t. \tag{5.9}$$

We will now show that $\mathbb{E}[\Delta M_t^{i\to j} \mid \mathcal{F}_{t-1}] \leq 1$, i.e., is a sequential e-value under $(\mathcal{F}_t)$. This is trivially true if $I_t \neq i'$, so we consider the case where $I_t = i'$.

$$\begin{aligned}\mathbb{E}[\Delta M_t^{i\to j} \mid I_t = i', \mathcal{F}_{t-1}] &= \mathbb{E}[\Delta M_t^{i\to j} \mid \mathcal{F}_{t-1}^{i'}, I_t = i, \bigcup_{j\in V, j\neq i'} \mathcal{F}_{t-1}^j] \\ &= \mathbb{E}[\Delta M_t^{i\to j} \mid \mathcal{F}_{t-1}^{i'}, I_t = i'] \leq 1.\end{aligned}$$

The first equality is because $\mathcal{F}_t = \mathcal{F}_{t-1}^{i'} \cup \bigcup_{j\in V, j\neq i'} \mathcal{F}_{t-1}^j$. The last line is by (5.9) and $\Delta M_t^{i\to j}$ being a sequential e-value under $\mathcal{F}_{t-1}^{i'}$.

□

Let $\Delta\mathbf{M}_t(i')$ be the vector of $\Delta M_t^{i\to j}(i')$ indexed for each $i \to j \in P(i')$. Now, we utilize the following regret bound from Fact 5, which implies that there exists a sequence of weights $(\mathbf{w}_t)$ predictable w.r.t. $(\mathcal{F}_t)$ such that we can define the following process:

$$
\begin{aligned}
M_t^P &:= \prod_{k=1}^{t} \mathbf{w}_k^\top \Delta\mathbf{M}_k(I_k) = \exp\left(\sum_{k=1}^{t} \log(\mathbf{w}_k^\top \Delta\mathbf{M}_k(I_k))\right) \\
&= \exp\left(\sum_{i'\in V} \log\left(\prod_{k\in T_{i'}(t)} \mathbf{w}_k^\top \Delta\mathbf{M}_k(i')\right)\right) \qquad \text{(collecting terms across } I_k \in V\text{)} \\
&\overset{(a)}{\geq} \exp\left(\sum_{i'\in V} \max_{\mathbf{w}\in\Delta^{|P(i')|}} \log\left(\prod_{k\in T_{i'}(t)} \mathbf{w}^\top \Delta\mathbf{M}_k(I_k)\right) - \frac{|P(i')|-1}{2}\cdot\log(2(|T_{i'}(t)|-1))\right) \\
&\overset{(b)}{=} \exp\left(\sum_{i'\in V} \max_{\mathbf{w}\in\Delta^{|P(i')|}} \log\left(\prod_{k\in[t]} \mathbf{w}^\top \Delta\mathbf{M}_k(I_k)\right) - \frac{|P(i')|-1}{2}\cdot\log(2(|T_{i'}(t)|-1))\right) \\
&\overset{(c)}{\geq} \exp\left(\sum_{i'\in V} \max_{i\to j\in P(i')} \log M_t^{i\to j}(i') - \frac{|P(i')|-1}{2}\cdot\log(2(|T_{i'}(t)|-1))\right) \\
&\geq \exp\left(\sum_{i'\in V} \max_{i\to j\in P(i')} \log E_t^{i\to j}(i') - \frac{|P(i')|-1}{2}\cdot\log(2(|T_{i'}(t)|-1))\right)
\end{aligned}
$$

Inequality (a) is a result of Fact 5. For equality (b), we note that $\Delta\mathbf{M}_k(i') = \mathbf{1}$ (i.e., the vector of ones) for each $k \notin T_{i'}(t)$, as a result of (5.8). Consequently, is we can change the index of the product from $T_{i'}(t)$ to $[t]$, since multiplying by $\mathbf{w}^\top\mathbf{1} = 1$ does not change the product. Inequality (c) is because the elementary bases is a subset of $\Delta^{|A_{i'}|}$ and $\prod_{k\in T_{i'}(t)} \Delta M_k^{i\to j}(i') = M_t^{i\to j}(i')$ due to telescoping product. The last inequality is by definition of $M_t^{i\to j}(i') \geq E_t^{i\to j}(i')$ for all $t \in \mathbb{N}$.

Now, we only need to show that M_t^P is an NSM w.r.t. $(\mathcal{F}_t)$. Recall $M_t^P = \prod_{k\in T_{i'}(t)} \mathbf{w}_k^\top \Delta\mathbf{M}_k(I_k)$. We have that:

$$\mathbb{E}[M_t^P|\mathcal{F}_{t-1}] = \mathbb{E}[\mathbf{w}_t^\top \Delta\mathbf{M}_t(I_t) \mid \mathcal{F}_{t-1}]M_{t-1}^P$$

Thus, it suffices to show the following:

$$\mathbb{E}[\mathbf{w}_t^\top \Delta\mathbf{M}_t(I_t) \mid \mathcal{F}_{t-1}] \leq 1 \text{ under } H_0^P. \tag{5.10}$$

We know the following is true under H_0^P:

$$\mathbb{E}[\mathbf{w}_t^\top \Delta\mathbf{M}_t(I_t) \mid \mathcal{F}_{t-1}] = \sum_{i\to j\in P_{I_t}} w_t^{i\to j}\mathbb{E}[\Delta M_t^{i\to j}(I_t) \mid \mathcal{F}_{t-1}] \leq \sum_{i\to j\in P_{I_t}} w_t^{i\to j} = 1.$$

The inequality is by definition of $\Delta M_t^{i\to j}$ of being a sequential e-value (under $(\mathcal{F}_t)$) under $H_0^{i\to j}$, which holds as $i \to j \in P$ and H_0^P holds by assumption. The last equality is by $\mathbf{w}_t \in \Delta^{|A_{I_t}|}$.

Thus, we have shown (5.10) and proven our desired result.

□

Proposition 32 (Correctness of combined e-process)**.** *Define*

$$E_t^{i\to j} := \min_{P\in\mathcal{P}(T^{i\to j})} E_t^P.$$

Then, $(E_t^{i\to j})$ is an e-process when $H_0^{i\to j}$ is true.

Proof. By the definition of $\mathcal{P}(T^{i\to j})$:

$$H_0^{i\to j} = H_0(T^{i\to j}) = \bigcup_{P\in\mathcal{P}(T^{i\to j})} H_0^P$$

Thus, if $H_0^{i\to j}$ is true, then there exists $P \in \mathcal{P}(T^{i\to j})$ such that H_0^P is true.

(E_t^P) is an e-process by Proposition 19. Since, $E_t^{i\to j} \le E_t^P$ for all $t \in \mathbb{N}$ almost surely, $(E_t^{i\to j})$ is an e-process, and we have shown our desired result. □

Theorem 24. *For any sequence of interventions (I_t) predictable w.r.t. $(\mathcal{F}_t)$, let $\widehat{G}_t$ be the partially oriented DAG where the test for each orientation is defined as follows:*

$$\varphi_t^{i\to j} = \mathbb{1}\{E_t^{i\to j} \ge |\overline{G}|/\alpha\}.$$

Then, $(\widehat{G}_t)$ is anytime-valid orientation (as defined in (5.11)*).*

Proof. Let the final oriented graph be $\hat{G}$.

$$\begin{aligned}
&\mathbb{P}\left(\text{exists } t \in \mathbb{N} : \text{exists oriented edge in } \widehat{G}_t \text{ not in } G^*\right)\\
&\le \sum_{i\to j\in\hat{G}} \mathbb{P}\left(\text{exists } t \in \mathbb{N} : \text{orient edge } i \to j \wedge j \to i \text{ in } G^*\right)\\
&= \sum_{i\to j\in\hat{G}} \Pr(\text{exists } t \in \mathbb{N} : E_t^{j\to i} \ge |\overline{G}|/\alpha \wedge j \to i \text{ in } G^*)\\
&\le \sum_{i\to j\in\hat{G}} \alpha/|\bar{G}| && \text{(by Proposition 20)}\\
&= \alpha && (5.11)
\end{aligned}$$

□

Proposition 33 (Additional power of combining test statistics)**.** *Consider an uniform set of interventions over nodes $[n]$. There exists a graph and edge $i \to j$, such that the expected growth rate (i.e. power) of $\log E_t^{i\to j}$ under the fully expanded tree $T^{i\to j}$, is $\Omega(|\bar{G}|)$ times that of $\log E_t^{i\to j}$ under the non-expanded tree (i.e. just the single edge $i \to j$).*

Proof. Consider a chain graph $\bar{G} = X_1 - X_2 - ... - X_n$ (generalizable to trees where the root has only one child), where the underlying graph is such that $X_1 \leftarrow X_2 ... \leftarrow X_n$. Such a setting allows for a simple, closed-form expression for the test statistic.

Suppose we are interested in testing $H_0^{1\to 2}$. Suppose there are m interventions, which means m/n interventions of each node.

In this setting, we assume that edge causal effects and variance are equal for fair comparisons. Thus $\forall i, j, \mathbb{E}[\log E_t^{i\to i+1}(i+1)] = \mathbb{E}[\log E_t^{j\to j+1}(j+1)]$. Certainly, if $E_t^{j\to j+1}(j+1)] > E_t^{1\to 2}(2)]$, then gain in power will be even more pronounced.

Using Proposition 19, we have that:

$$\begin{aligned}
&\mathbb{E}[\log E_t^{1\to 2}] \\
&= \mathbb{E}\left[\log E_t^{1\to 2}(1) + \sum_{i=2}^{n-1}[\max(\log E_t^{i-1\to i}(i), \log E_t^{i\to i+1}(i)) - 1/2(\log(2|T_i(t)| - 2))] + \log E_t^{n-1\to n}(n)\right] \\
&\geq \mathbb{E}\left[\log E_t^{1\to 2}(1) + \sum_{i=2}^{n-1}\log E_t^{i-1\to i}(i) + \log E_t^{n-1\to n}(n) - 1/2(n-2)\log(2m/n - 2)\right] \\
&\geq \sum_{i=2}^{n-1}\mathbb{E}[\log E_t^{i-1\to i}(i)] - 1/2(n-2)\log(2m/n-2) \\
&= (n-2)\cdot\mathbb{E}[\log E_t^{1\to 2}(2)] - \tilde{O}(n) \\
&\geq (n-2)/2\cdot\mathbb{E}[\log E_t^{1\to 2}(2) + \log E_t^{1\to 2}(1)] - \tilde{O}(n) \\
&\qquad\qquad (\text{for any edge } i\to j, \mathbb{E}[\log S_t^{i\to j,+}(j) \mid \mathcal{F}_{t-1}] \geq \mathbb{E}[\log S_t^{i\to j,+}(i) \mid \mathcal{F}_{t-1}]) \\
&\geq C\cdot\mathbb{E}[\log E_t^{1\to 2}(2) + \log E_t^{1\to 2}(1)] \qquad (\text{we assume that } \log E_t^{1\to 2}(2) = \Omega(m) >> \tilde{O}(n))
\end{aligned}$$

for constant $C = \Omega(n)$.

□

Remark 17. *Note that the combination of evidence is such that we need not reject any of $X_i \to X_{i+1}$ to reject $X_1 \to X_2$. The cumulative evidence is enough, despite the data not being conclusive for any of the downstream edges!*

5.9.3 Time complexity analysis of algorithms

Algorithm 13: Each path in the loop contains at most $|E|$ edges. Each round in the while loop requires examining at most $|E|$ to see if there is a new edge that is implied via Meek's rule. Thus, if the algorithm is run for T rounds, the time complexity is $T \cdot |E|$. We note that how much the tree is expanded out, as a function of T, is an user choice.

Algorithm 15: First, we consider the time complexity of updating test-statistic given a new intervention i at time t. It suffices to update E_t^P for every path $P \in T^{i\to j}$, which is pre-computed. Using the definition of E^P, it suffices to just re-compute $\log E_t^{i\to j}(i')$ to incorporate the new interventional data, and then take the minimum.

If one re-computation is taken to require one unit of computation, then there are $|P(i')|$ many $\log E_t^{i\to j}(i')$ re-computations. Using the definition of $|P(i')|$, we know it is upper bounded by

Algorithm 15 Anytime Testing for Updates in Finite-Sample Causal Discovery

```
Require: Input: pre-compute logical tree $T^{i\to j}$ for each hypothesized edge orientation for edge
    $i - j$ in skeleton (via Algorithm 13)
Require: Sample from intervention distribution $(x_1^t, ..., x_n^t) \sim X_1, ..., X_n | do(X^t)$
for node $X_i$ adjacent to $X_{I_t}$ do
    if edge $X_i - X_{I_t}$ unoriented then
        Update $E_t^{i\leftarrow I_t}$, $E_t^{i\to I_t}$
        Test $E_t^{i\leftarrow I_t} \geq |\overline{G}|/\alpha, E_t^{i\to I_t} \geq |\overline{G}|/\alpha$    ▷ Test if we can conclude $i \not\leftarrow I_t$ or $i \not\rightarrow I_t$
    w.h.p.
for [ doPropagation]hypothesized orientation edge $i' \to j'$; $i' - j'$ unoriented, $i', j' \neq I_t$
    if exists edge $i \leftarrow I_t$ or $i \to I_t$ in $T^{i'\to j'}$ then
        Update $E_t^{i'\to j'}$ using updated $E_t^{i\leftarrow X^t}$ or $E_t^{i\to X^t}$
        Test $E_t^{i'\to j'} \geq |\overline{G}|/\alpha$    ▷ Test if we can conclude $i' \not\rightarrow j'$ w.h.p.
```

the degree of i'. Thus, if the max degree of the graph is $deg(G)$, then the update to each edge test statistic requires at most $deg(G) \cdot |P \in P(T^{i\to j})|$ updates. In total, updating this for all edge hypotheses is upper bounded by: $2|E| \cdot deg(G) \cdot |P \in P(T^{i\to j})| = O(|V||E| \max_{i\to j} |P \in P(T^{i\to j})|)$. Finally, if there are T rounds with T interventions, the total number of updates comes out to: $O(T \cdot |V||E| \max_{i\to j} |P \in P(T^{i\to j})|)$.

Note that this characterizes the time-complexity of the e-process updates in as being polynomial (more precisely linear) in terms of the graph parameters and the size of the implication trees. As we previously note, the size of this implication tree (that is pre-computed) is an user-based choice. The more implications that are enumerated in the tree, the higher the power of the test. However, this in turn increases the time-complexity (and memory), which we can observe above.

5.10 Experiments

5.10.1 Fixed-time test statistic construction

Proposition 34. *$P_t^{i\to j}$ satisfies $\mathbb{P}(P_t^{i\to j} \leq s) \leq s$ for all $s \in [0, 1]$ and $t \in \mathbb{N}$ under $H_0^{i\to j}$.*

Proof. For $c \in \{\pm 1\}$, define

$$P_t^{+}(c) := 1 - \Phi\left(\frac{b \cdot T_i(t) - \widehat{\mu}_t^{j|\text{do}(i)} + c \cdot \widehat{\mu}_t^{i|\text{do}(j)}}{\sqrt{|T_j(t)| \operatorname{var} X_i + |T_i(t)| \operatorname{var} X_j}}\right),$$

$$P_t^{-}(c) := 1 - \Phi\left(\frac{\widehat{\mu}_t^{j|\text{do}(i)} + b \cdot T_i(t) + c \cdot \widehat{\mu}_t^{i|\text{do}(j)}}{\sqrt{|T_j(t)| \operatorname{var} X_i + |T_i(t)| \operatorname{var} X_j}}\right).$$

Note that, for any choice of c, $P_t^{+}(c)$ and $P_t^{-}(c)$ are z-test p-values under $H_0^{i\to j,+}$ and $H_0^{i\to j,-}$ respectively. As a result, $\max(P_t^{+}, P_t^{-})$ is a p-value under $H_0^{i\to j}$.

Now, we can see that the following is true:

$$P_t^{i\to j} = 2\min(\max(P_t^+(1), P_t^-(1)),\\ \max(P_t^+(-1), P_t^-(-1))).$$

Since taking double the minimum of any two p-values is still a valid p-value by union bound, we get our desired result that $P_t^{i\to j}$ is a p-value.

□

Remark 18. *Note the difference in qualifiers from the anytime guarantee, wherein correctness is guaranteed across time t, and not only at some fixed point t in time.*

5.10.2 Comparing fixed-time vs anytime test statistics

Experiment Configurations: In experiments, we fix $b = 0.1$, variance as 1 and interventional value $\nu = 1$. Each setting is run for 20 trials to evaluate the mean and standard deviation.

We plot two metrics: (1) the mis-coverage rate (number of trials wherein the test statistic returns at least one falsely oriented edge) (2) the number of oriented edges (indeed an uninformative test that never rejects can trivially achieve 0 miscoverage rate).

To assess the guarantee of anytime approaches across a number of settings, we have the following experiments:

- Figures 5.3 and 5.5: Varying the number of interventional samples $\{100, 500, 1000, 5000, 10000\}$ (fixing $k = 0.2$) in ER graphs with number of nodes $\in \{10, 20, 30\}$, $\alpha \in \{0.1, 0.2\}$ and $p = 0.3$.
- Figures 5.4 and 5.6: Varying the number of interventional samples $\{100, 500, 1000, 5000, 10000\}$ (fixing $k = 0.2$) in ER graphs with number of nodes $\in \{10, 20, 30\}$, $\alpha \in \{0.1, 0.2\}$ and $p = 0.5$.
- Figures 5.7 to 5.10: Varying the number of interventional samples $\{100, 500, 1000, 5000, 10000\}$ (fixing $k = 0.2$) in tree graphs with number of nodes $\in \{10, 20, 50, 100\}$ and $\alpha \in \{0.1, 0.2\}$.
- Figures 5.11 and 5.12: Varying edge causal strength $k \in \{0.1, 0.2, 1, 2, 10\}$ (fixing number of samples at 1000) in ER graphs $(n, p) \in \{10, 20, 30\} \times (0.3, 0.5)$ and $\alpha = 0.2$.

In all these settings, in terms of miscoverage, we find that the anytime approach has controlled error rate below that of α (in green), although the miscoverage rate is not always 0. On the other hand, the fixed time approach can attain sizable error rate and introduce spuriously oriented edges. This trend seems consistent across two classes of graphs (ER and trees), as well as in ER graphs with varying SCM parameters edge strength k.

In terms of number of orientations, we observe that the number of orientations increases with sample complexity (as expected). However, the anytime test statistic orients conservatively at a (much) lower pace than does the fixed time approach. In exchange, this provides the error control and guarantees a high probability of only correct orientations.

5.10.3 Understanding the effectiveness of combining test statistics

To examine the effectiveness of propagating evidence from test statistics, we have the following experiment:

- Figure 5.13: Varying the number of interventional samples $\{100, 500, 1000, 5000, 10000\}$ (fixing $k = 0.2$) and plot the number of oriented edges in a chain graph with the number of nodes in $\{5, 10, 20, 50\}$ and edges have alternating causal strength in $\{0.1, 10\}$.

Chain graphs are an example where edges may benefit from propagation effects. We set up the causal strength to vary such that some edge orientations (those with low edge strength) will benefit from other edges (those with high edge strength). In the experiment, we compare the number of oriented edges at a fixed sample size by base e-values (as in Section 5.3) against the combined e-values (as in Section 5.4). Note that we also check for miscoverage rate to ensure correctness (in order to have a fair comparison); we do find that the miscoverage rate under both are 0.

In the plot, we observe that combining test statistics *may* help. It orients more edges than base e-values, when one has a sizable number of samples. Interestingly, we find that the base e-values is better in lower sample regimes. Moreover, the number of data points after which the combined test statistic is more effective increases with graph size.

We believe that this happens, because the combined test statistic, while having higher mean, also has higher variance. Thus, it is most effective when there are more samples. Overall, this suggests we should favor base e-values in smaller data and/or graph regimes, and combined test statistics in big data/graph regimes. The lighter-weight base e-values can be surprisingly effective. Verily, an interesting future work would be to develop testing methods that adapt the test statistic to the (unknown) SCM parameters and the test parameters (e.g. number of budgeted samples).

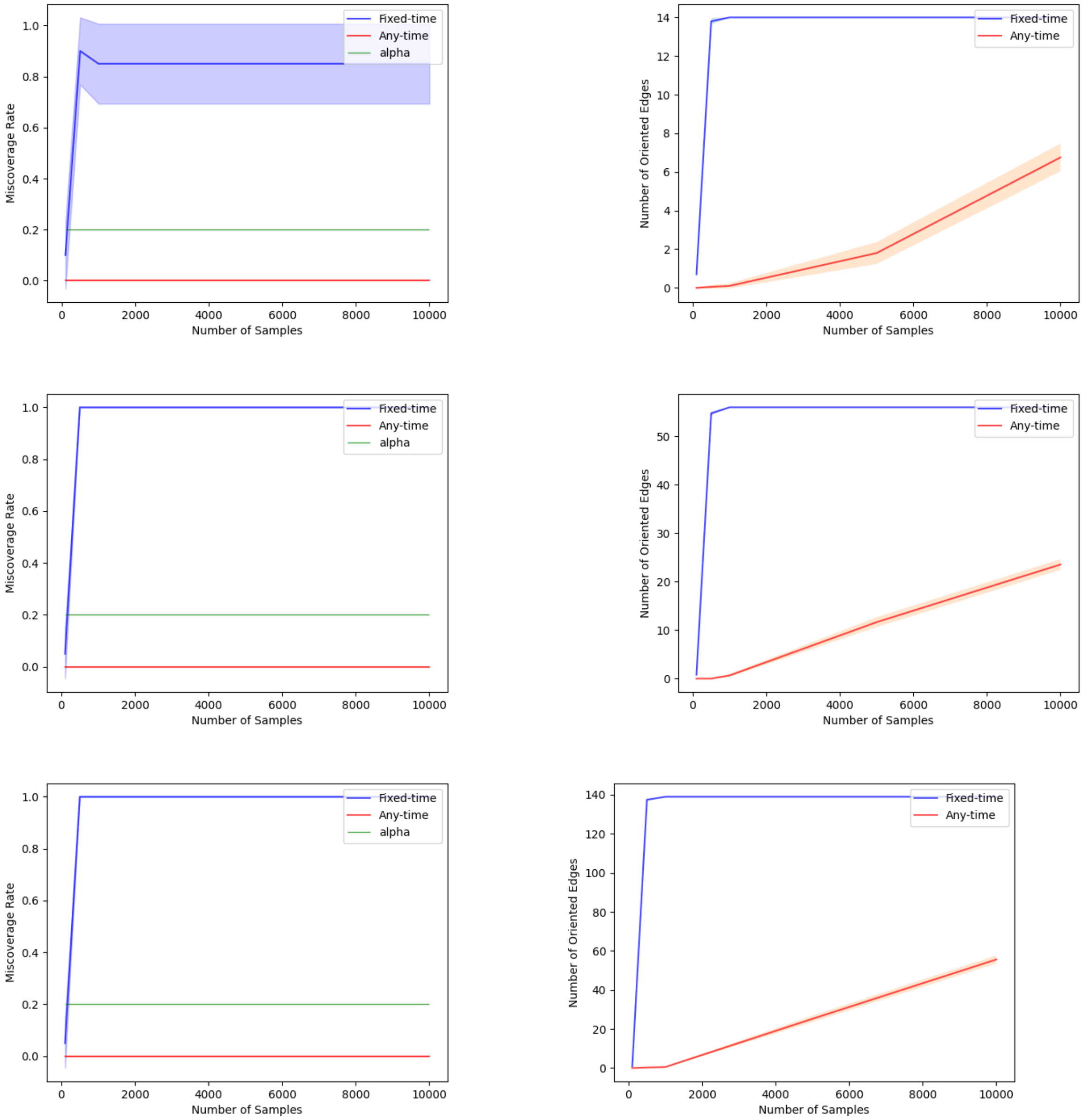


Figure 5.3: Plotting miscoverage rate and number of orientations in Erdos-Renyi Graphs with $\alpha = 0.2, p = 0.3$. First Row: $(n, p) = (10, 0.3)$; Second Row: $(n, p) = (20, 0.3)$; Third Row: $(n, p) = (30, 0.3)$.

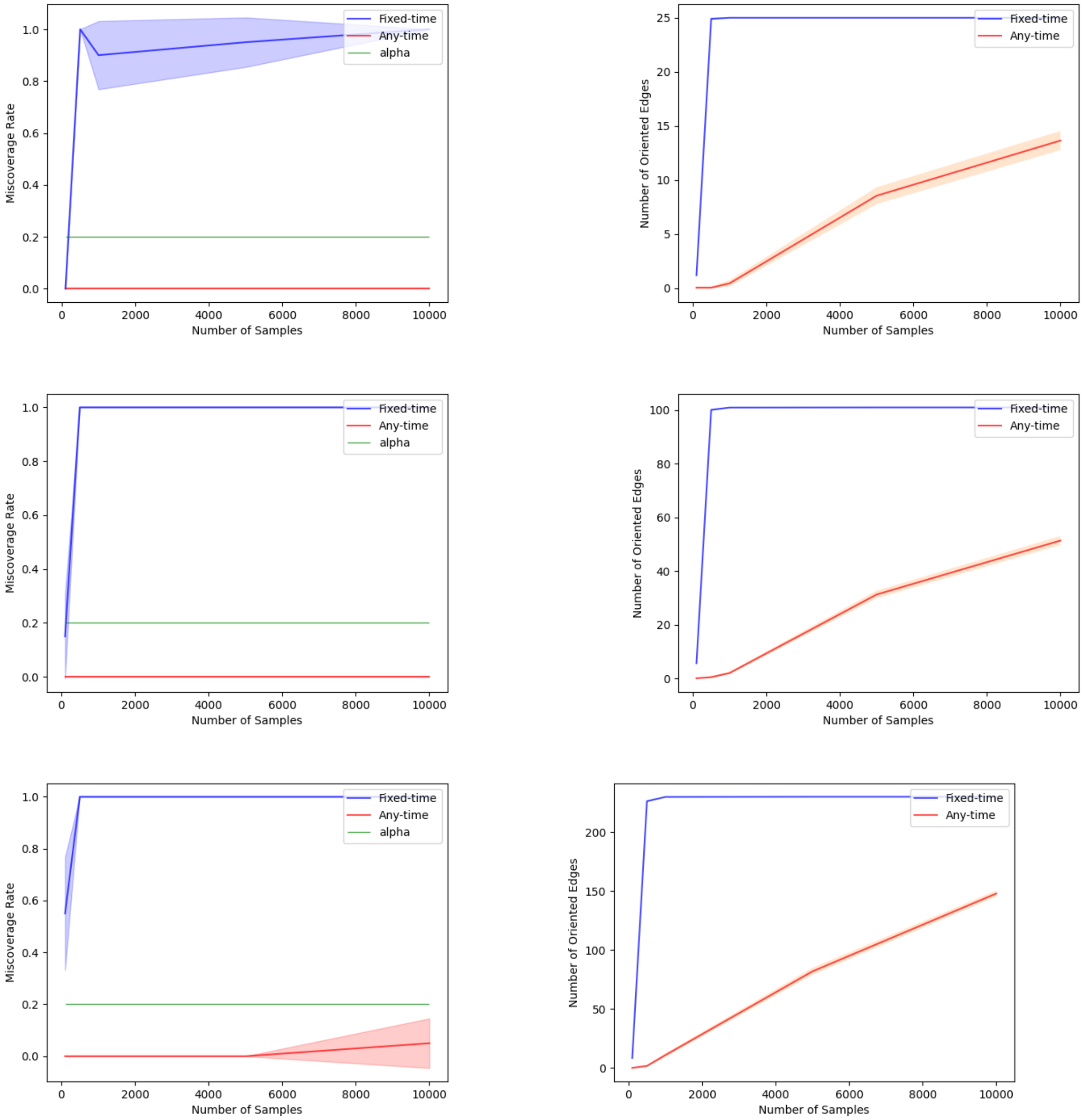


Figure 5.4: Plotting miscoverage rate and number of orientations in Erdos-Renyi Graphs with $\alpha = 0.2, p = 0.5$. First Row: $(n, p) = (10, 0.5)$; Second Row: $(n, p) = (20, 0.5)$; Third Row: $(n, p) = (30, 0.5)$.

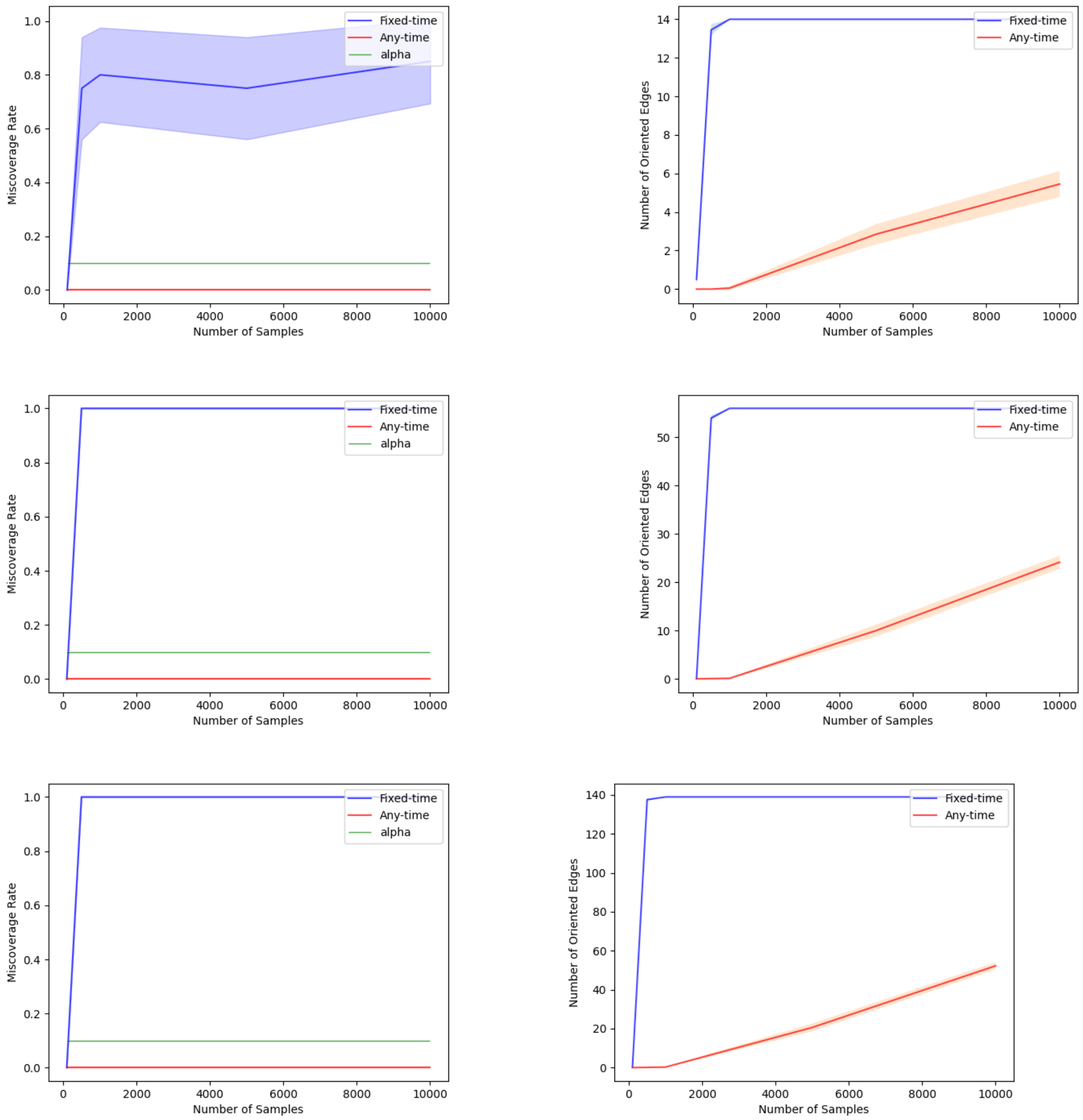


Figure 5.5: Plotting miscoverage rate and number of orientations in Erdos-Renyi Graphs with $\alpha = 0.1, p = 0.3$. First Row: $(n, p) = (10, 0.3)$; Second Row: $(n, p) = (20, 0.3)$; Third Row: $(n, p) = (30, 0.3)$.

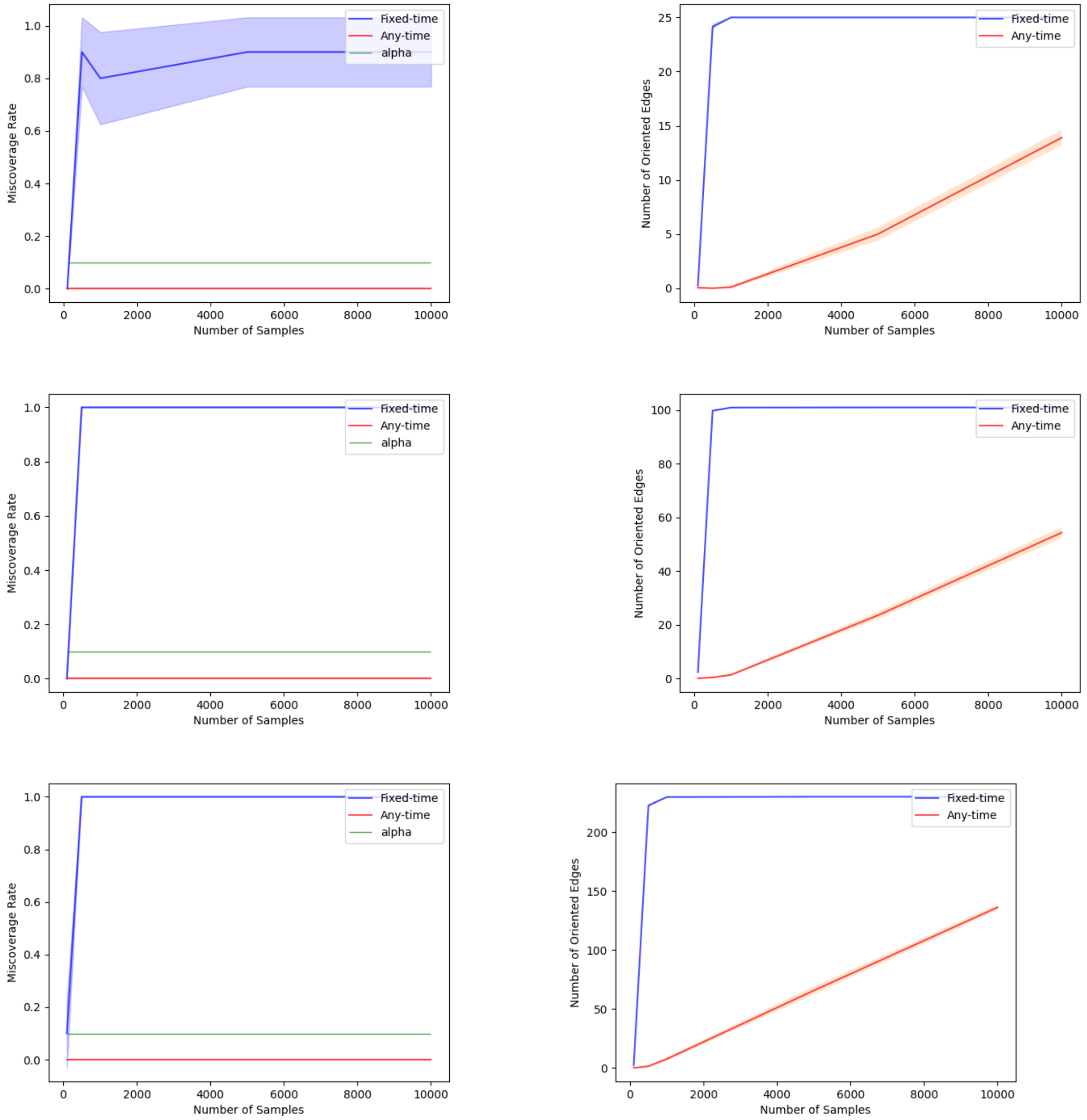


Figure 5.6: Plotting miscoverage rate and number of orientations in Erdos-Renyi Graphs with $\alpha = 0.1, p = 0.5$. First Row: $(n, p) = (10, 0.5)$; Second Row: $(n, p) = (20, 0.5)$; Third Row: $(n, p) = (30, 0.5)$.

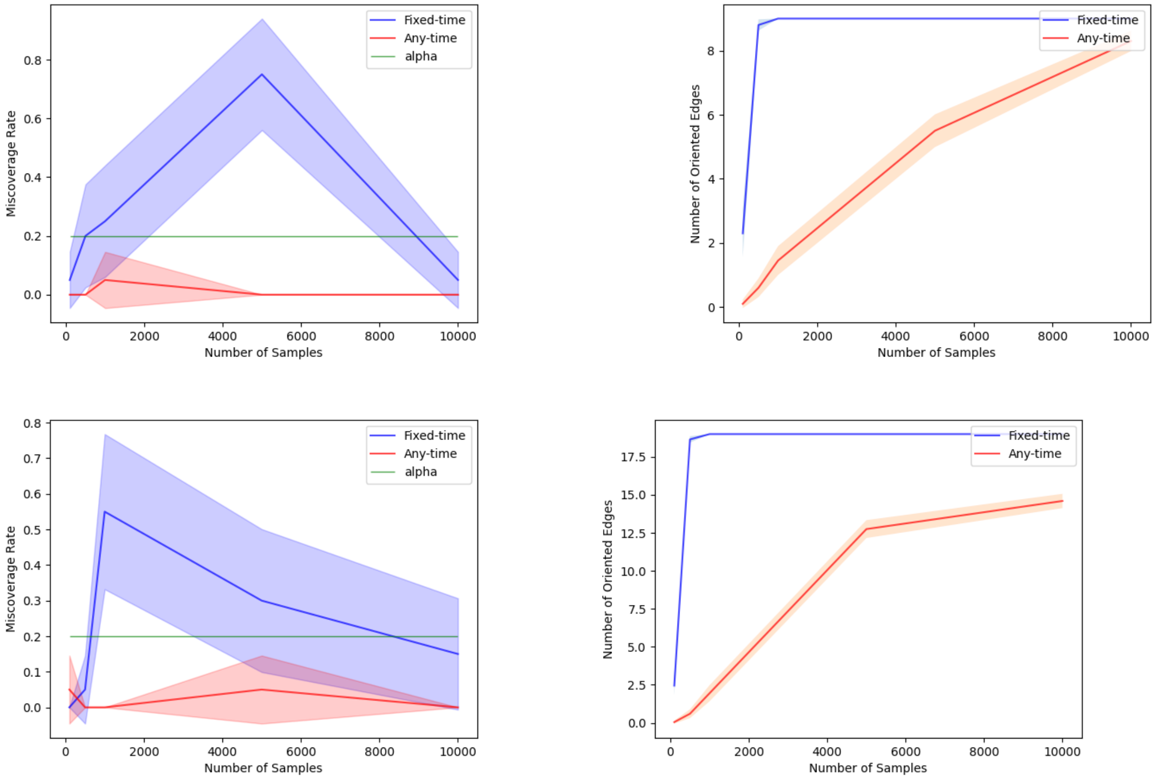


Figure 5.7: Plotting miscoverage rate and number of orientations in tree graphs with $\alpha = 0.2, n \in \{10, 20\}$. First Row: $n = 10$; Second Row: $n = 20$.

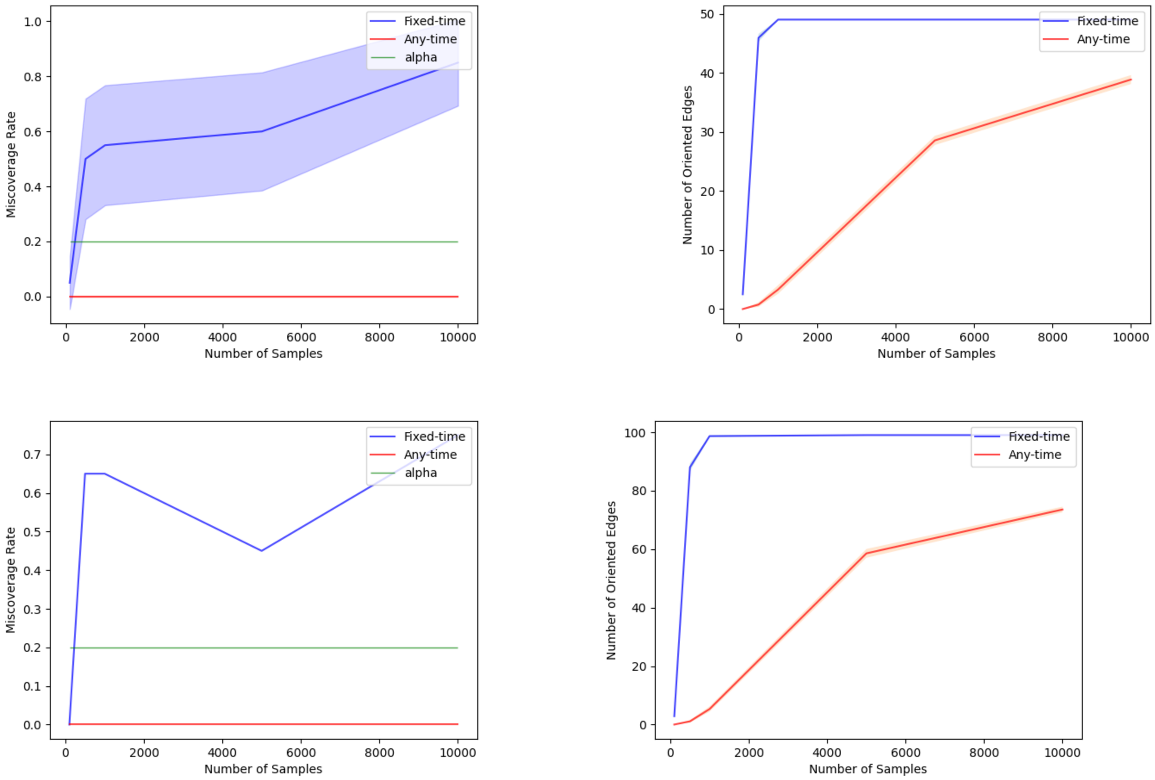


Figure 5.8: Plotting miscoverage rate and number of orientations in tree graphs with $\alpha = 0.2, n \in \{50, 100\}$. First Row: $n = 50$; Second Row: $n = 100$.

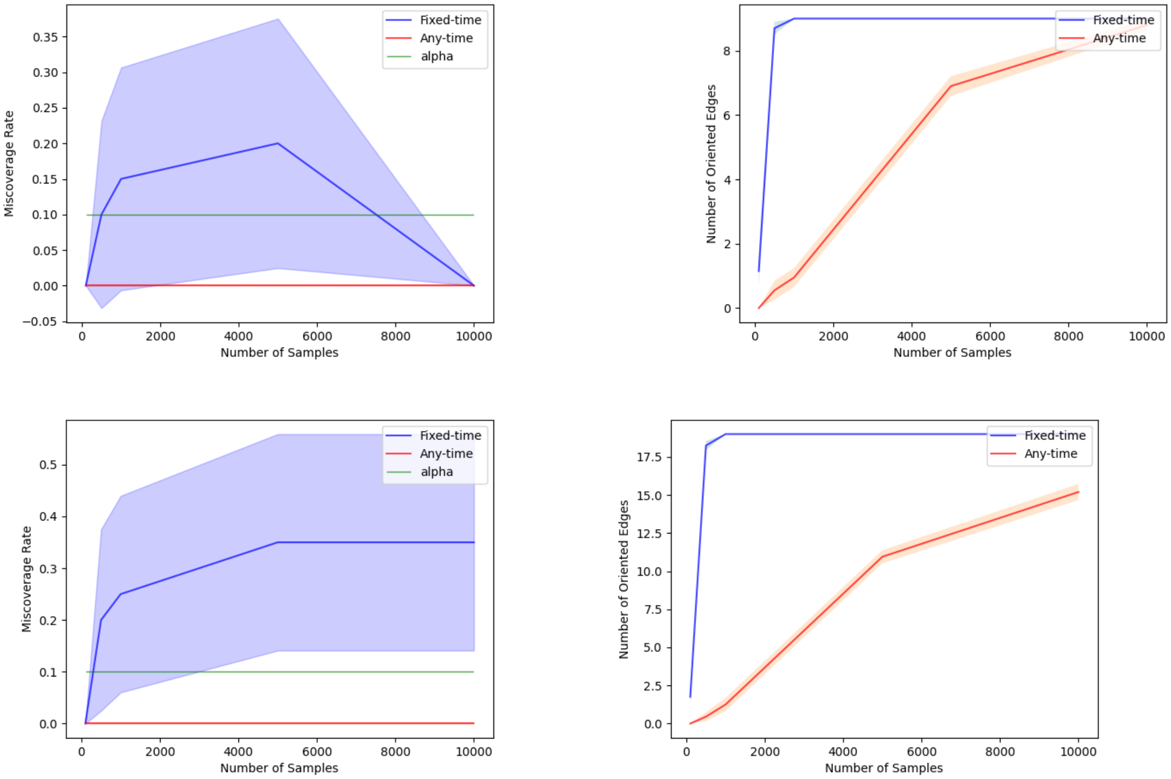


Figure 5.9: Plotting miscoverage rate and number of orientations in tree graphs with $\alpha = 0.1, n \in \{10, 20\}$. First Row: $n = 10$; Second Row: $n = 20$.

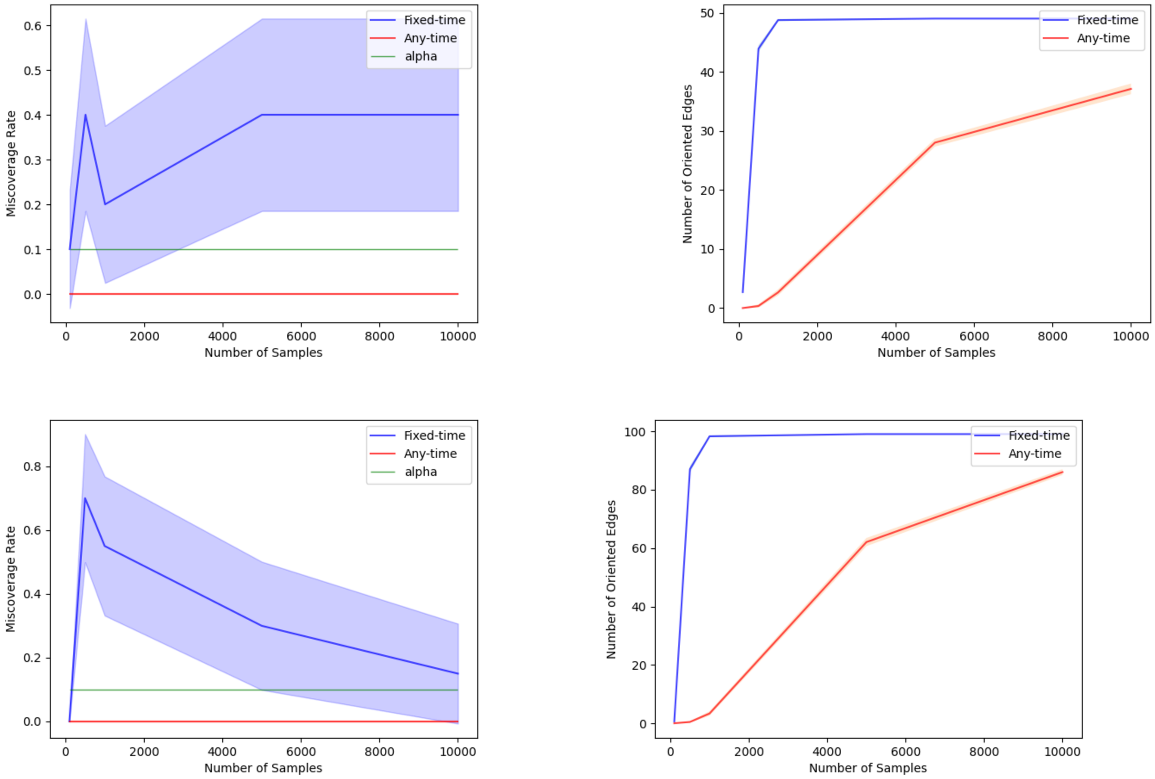


Figure 5.10: Plotting miscoverage rate and number of orientations in tree graphs with $\alpha = 0.1, n \in \{50, 100\}$. First Row: $n = 50$; Second Row: $n = 100$.

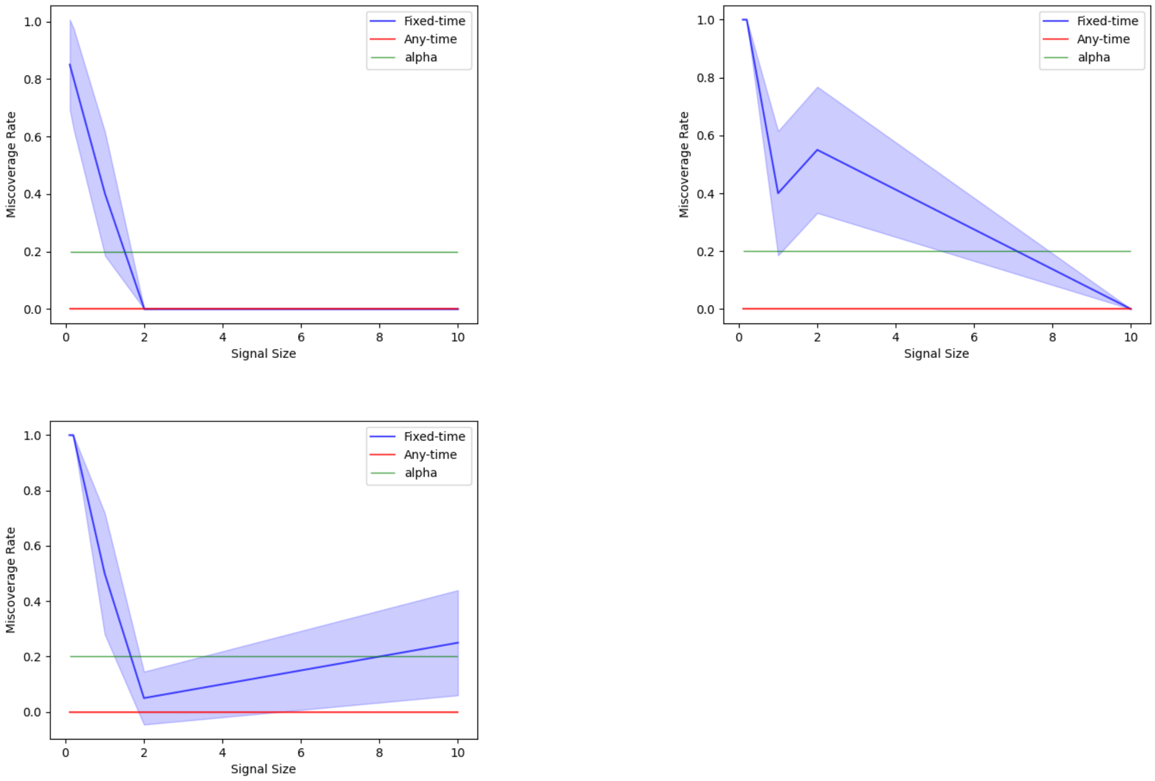


Figure 5.11: Plotting SCM parameter (edge strength k) vs miscoverage rate in ER graphs with $\alpha = 0.2, p = 0.3$. First Row: $n = 10$ (left) and $n = 20$ (right); Second Row: $n = 30$.

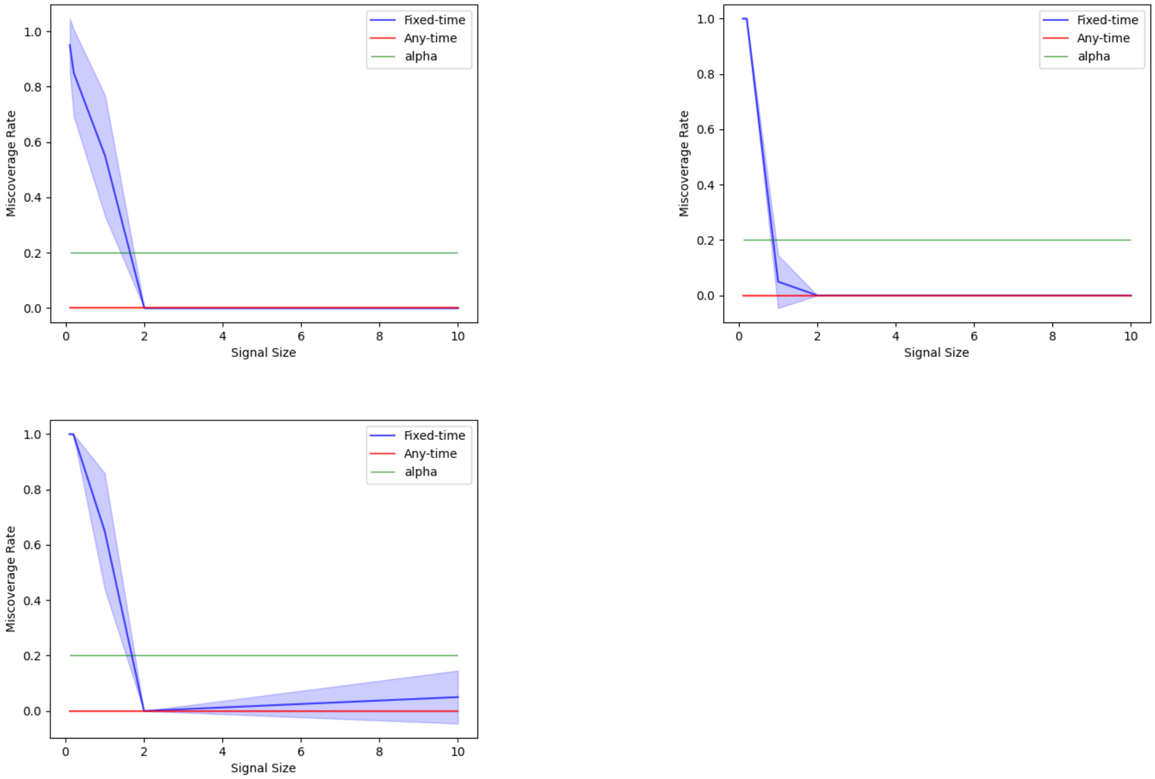


Figure 5.12: Plotting SCM parameter (edge strength k) vs miscoverage rate in ER graphs with $\alpha = 0.2, p = 0.5$. First Row: $n = 10$ (left) and $n = 20$ (right); Second Row: $n = 30$.

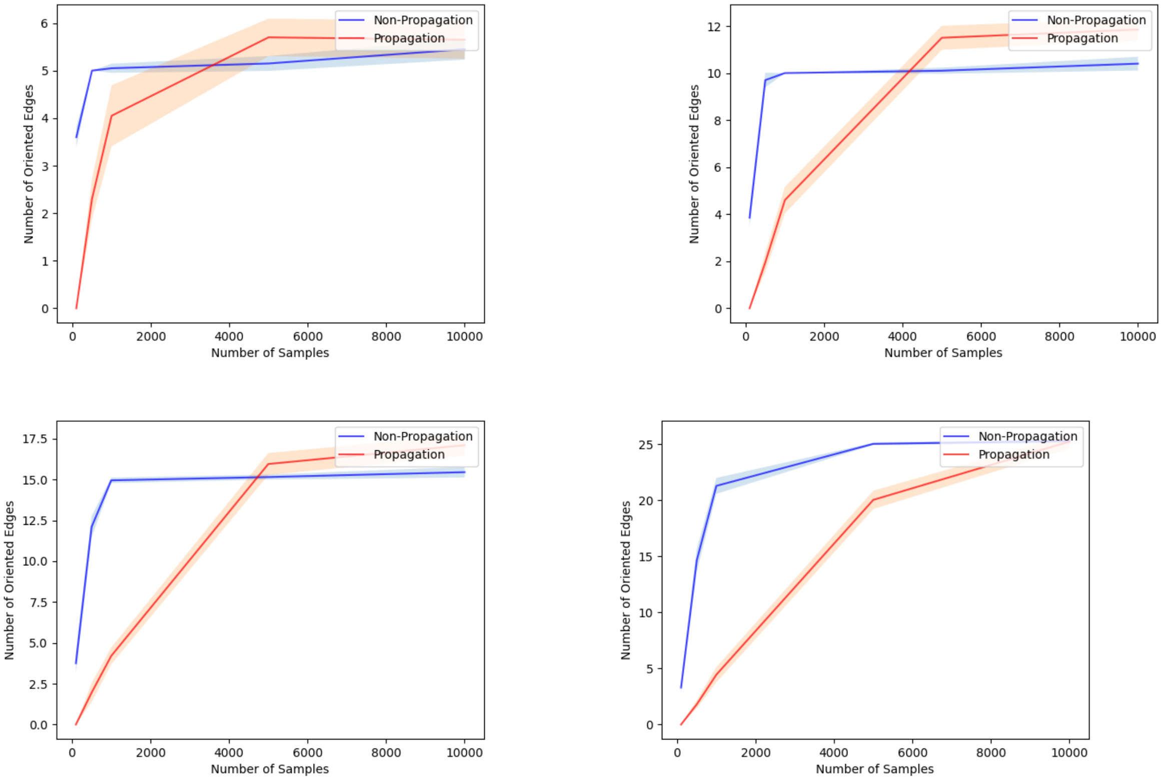


Figure 5.13: Comparing number of orientations of combined e-values vs those of base e-values in chain graphs with $\alpha = 0.2$. First Row: $n = 10$ (left) and $n = 20$ (right); Second Row: $n = 30$ (left) and $n = 50$ (right).

5.10.4 Evaluating Derived Upper Bounds on Stopping Time useful for Robust Testing

In Subsection 5.3.4, we derive a set of upper bounds on the number of samples needed for testing. One important implication is that this allows one to have an upper bound estimate on the amount of interventional data that one needs to collect to do the test. The other implication of this, useful for robust testing, is that one can use the non-conclusiveness of the test after this number of samples to detect spurious, non-edges.

In this subsection, we empirically verify this claim by evaluating the sample complexity needed for testing the orientation of some edge. Please refer to Figure 5.14 for a plot of the results.

- Firstly, we verify that with high probability, the bounds derived in Proposition 17 holds.

  In the experiments, we vary one parameter and fix the rest, checking the number of times the number of samples needed for testing is *below* the derived upper bound, out of 100 trials for each setting.

  We vary $\alpha = \{\text{1e-6}, \text{1e-5}, \text{1e-4}, \text{1e-3}, \text{1e-2}, \text{1e-1}, \text{5e-1}\}$, while setting $\mu_i(j) = 1.0, \sigma = 1.0, b = 0.1, \beta = 0.1$.

  We vary $\sigma = \{\text{1e-6}, \text{1e-3}, \text{1e-2}, \text{1e-1}, 1, 10\}$, while setting $\alpha = 0.01, \mu_i(j) = 1.0, b = 0.1, \beta = 0.1$.

  We vary $\mu_i(j) = \{\text{5e-1}, 1, 5, 10, 100, 1000\}$, while setting $\alpha = 0.01, \sigma = 1.0, b = 0.1, \beta = 0.1$.

- Secondly, we verify that when there is no edge between the two edges, then with high probability the test statistic *does not reject* before the derived number of samples, thus allowing us to use the contrapositive of Proposition 17 to detect spuriously oriented edges.

  In this set of experiments, we use the same parameter setting as above, with the only difference that there is no causal effect from node i to node j. We check the number of times the number of samples needed for test conclusion is *below* the derived upper bound, out of 100 trials for each setting.

  We vary $\alpha = \{\text{1e-6}, \text{1e-5}, \text{1e-4}, \text{1e-3}, \text{1e-2}, \text{1e-1}, \text{5e-1}\}$, while setting $\sigma = 1.0, b = 0.1, \beta = 0.1$. Here, $\mu_i(j) = 0.0$.

  We vary $\sigma = \{\text{1e-6}, \text{1e-3}, \text{1e-2}, \text{1e-1}, 1, 10\}$, while setting $\alpha = 0.01, b = 0.1, \beta = 0.1$. Here, $\mu_i(j) = 0.0$.

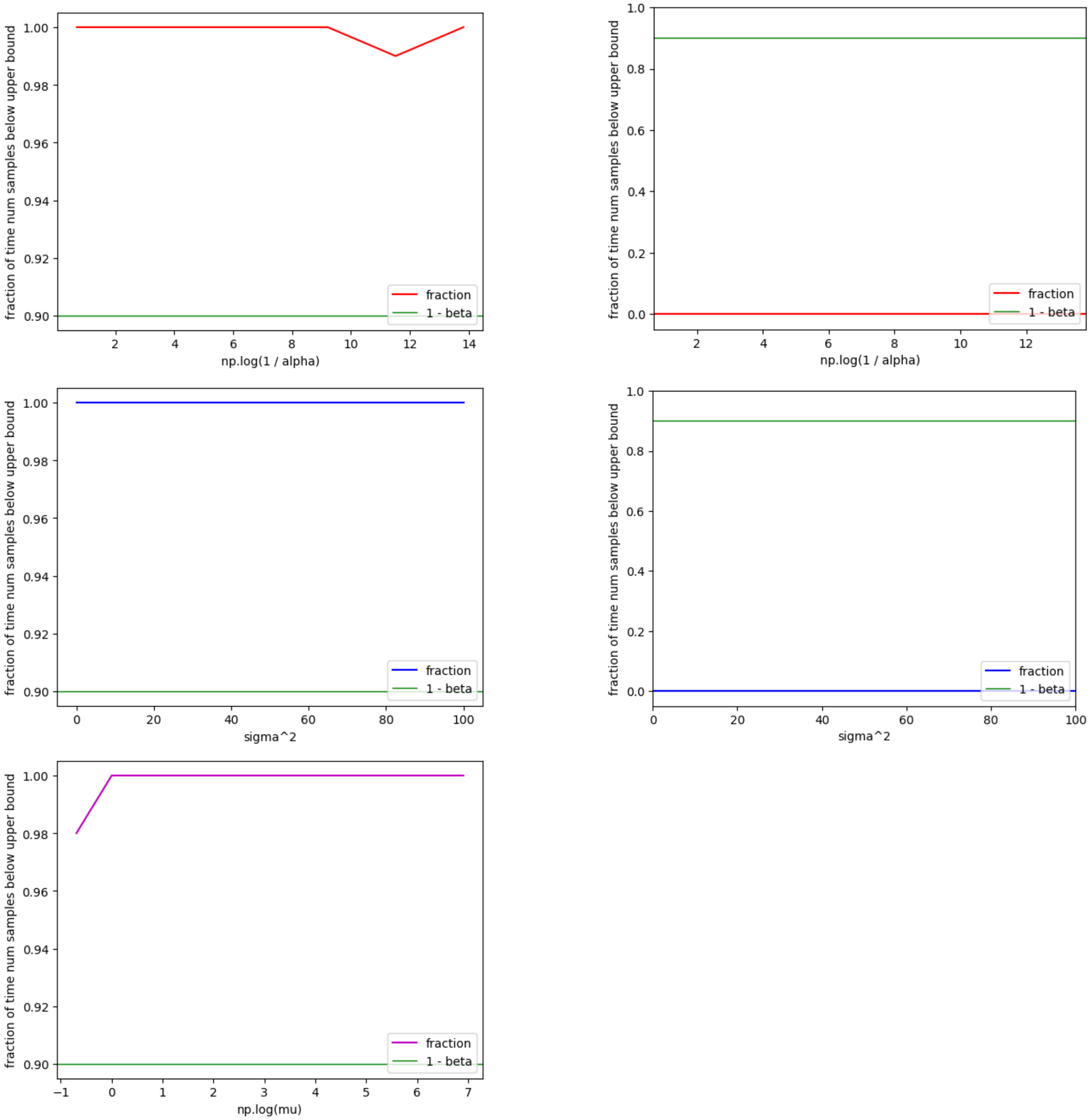


Figure 5.14: (Left Column) the fraction of 100 trials where the needed number of samples to conclude the test is below that of the derived upper bound (Right Column) the fraction of 100 trials where the needed number of sample complexity is below that of the derived upper bound, when there is no edge between the two nodes (i.e. $\mu_j(i) = 0$).

5.11 Multi-constraint Bandit Optimization

Causal verification is a well-known and important task in causal discovery [70, 232, 266]. Besides having practical applications (e.g. verifying a scientific conjecture corresponding to a causal graph structure), it has the theoretical benefit of understanding the lower bound that underlies any causal discovery algorithm.

The key challenge that arises in this problem is that an intervention policy needs to choose nodes I_t in order to grow the test statistic of all n edges *simultaneously*. Moreover, it only needs to optimize each test statistic only to the extent that the test statistic exceeds a threshold. In this section, we develop a novel, *multi-constraint* bandit algorithm needed for verification. Our key observation is that the causal verification setting reduces to the dual of the Bandits with Knapsack (BwK) setup [23].

To this end, we develop Algorithm 14 that attains provable guarantees in the multi-constraint bandit setting, and applies immediately to the causal verification setting using the reduction. As observed in [23], OPT, the expected total number of interventions needed by the optimal dynamic policy is difficult to characterize. In fact, even evaluating the expected number of interventions needed by *a given* time invariant, intervention policy is difficult. This is due to the difficulty of characterizing the *random stopping time* τ, when every test statistic exceeds the threshold. Thus, an algorithmic approach is taken where the proposed algorithm is shown to attain provable guarantees with respect to OPT.

5.11.1 Problem Statement

Setup: An instance of multi-constraint bandit optimization is parameterized by n arms, m constraints (henceforth "resources"), gain vector distributions $\mathcal{D}_i$ for arm i and budget b:

- There are n arms and m resources.
- Time proceeds in T rounds, where T is a finite time horizon given as input into the algorithm.
- Each round t, the learning algorithm picks some arm $x_t \in X$.
- Pulling arm x incurs deterministic cost c_x.
- The algorithm receives a gain vector, $R_{x_t} \in [0, M]^m$ where $R_{x_t} \sim D_{x_t}$ (some known distribution).
- There is a threshold $b \in \mathbb{R}^+$ on the total gain of each resource.
- The interaction terminates the first time $\sum_{t=1}^{\tau} R_{x_t} \geq b \cdot 1$.
- The goal of the algorithm is to minimize the total *expected* cost $\sum_{i=1}^{\tau} c_{x_i}$.

5.11.2 Reducing causal verification to multi-constraint bandits

We show that the causal verification problem corresponds to an instance of the multi-constraint bandit optimization problem. This algorithm is needed as our choice of intervention affects the e-processes of all edges.

We observe that bandit optimization is possible, because for any $j \to i$, the test statistic grows *additively* in the log of e-values:

$$\begin{aligned} E_t^{*j\to i} \geq 1/\alpha &\Leftrightarrow \forall P \in \mathcal{P}(T^{j\to i}), E_t^{*P} \geq 1/\alpha \\ &\Leftrightarrow \forall P \in \mathcal{P}(T^{j\to i}), \sum_{i' \in V} \log E_t^{*j\to i}(i') \geq \log(1/\alpha) \\ &\Leftrightarrow \forall P \in \mathcal{P}(T^{j\to i}), \sum_{k=1}^{t} \log S_k^*(P, I_k) \geq \log(1/\alpha) \end{aligned}$$

Reduction: Thus, given a causal verification instance, we may reduce to a multi-constraint bandit instance as follows:

1. Arms: define $n = |V|$ arms, each corresponding to a node intervention in the graph.

   Set $c_i = 1$ for all $i \in V$, as we only care about the total number of interventions. However, we note that our algorithm can handle differing node intervention costs.
2. Resources: define a resource corresponding to each (P, i') pair each path $P \in T^{j\to i}$ and intervention $i' \in V$.

   Accordingly, define the gain of pulling arm $i' \in V$ (i.e. intervening on node i') as a random draw of $\log S^*(P, i')$.
3. Define budget $b = \log 1/\alpha$.

In the causal verification setting, node intervention cost is set as $c_i = 1$ for all $i \in V$, since the objective of interest is the total number of interventions. Note however that Algorithm 14 can handle varying node intervention costs.

In the analysis below, we assume that X_i is a bounded r.v. with b, ν such that $\log S_k^*(P, I)$ is positive (as arm rewards are usually assumed to be positive in bandits literature). While this represents a subset of all SCM instances, as we will see, the query strategy design already results in solving an involved and novel multi-constraint bandit problem.

Finally, we note that one needs to manually specified the horizon T as in the multi-constraint bandit setting. This is a common assumption in BwK literature [23], which has proven to be difficult to remove. T in the causal verification setting may be viewed as the maximum number of experiments a scientist can run, or a known upper bound on the number of experiments needed to verify the graph. Verily, an exciting future direction is understanding how to remove the need to specify T, and develop a

5.11.3 Algorithm Guarantee:

The goal is to compete with the optimal dynamic policy given all the latent information. That is, OPT is the expected total number of steps of the optimal dynamic policy, given foreknowledge of the distribution of outcome vectors.

Since OPT is difficult to analyze, consider the fractional relaxation of this problem in which the number of rounds in which a given arm is selected (and also the total number of rounds) can be fractional, and the reward and resource consumption per unit time are deterministically equal to the corresponding expected values in the original instance.

$$\begin{aligned} \min_{k_1,\dots,k_n} \quad & c_1k_1 + \dots + c_nk_n \\ \text{s.t.} \quad & \sum_{j=1}^{n} r_{ji}k_i \geq b \text{ for each resource } i \in [m] \\ & k_i \geq 0 \end{aligned}$$

where k_i is the the fractional relaxation for the number of rounds in which a given arm i is selected.

This is a bounded LP, because $\sum_{i=1}^{n} k_i \leq T$ by definition. The optimal value of this LP is denoted by $\mathrm{OPT}_{\mathrm{LP}}$. We may construct the dual program:

$$\begin{aligned} \max_{v_1,\dots,v_m} \quad & b(v_1 + \dots + v_m) \\ \text{s.t.} \quad & \sum_{i=1}^{m} r_{ji}v_i \leq c_j \text{ for each arm } j \in [n] \\ & v_j \geq 0 \end{aligned}$$

The dual variables v_i can be interpreted as a unit gain for the corresponding resource i.

Lemma 38. $\mathrm{OPT}_{\mathrm{LP}}$ *is a lower bound on the value of the optimal dynamic policy:* $\mathrm{OPT}_{\mathrm{LP}} \leq \mathrm{OPT}$.

Proof. Let v^* be the optimal solution to the dual program. We note that by strong duality, $b\sum_{i=1}^{m} v_i^* = \mathrm{OPT}_{\mathrm{LP}} = \sum_{j=1}^{n} c_ik_i^*$.

Let Z_t denote the potential function: sum of costs incurred in optimal dynamic policy, plus total gain of the remaining resource endowment after round t.

At the start, the total gain of the remaining (all the) resource endowment is $Z_0 = b\sum_{i=1}^{m} v_i^*$.

We have that $Z_t = Z_{t-1} + c_{x_it} - \sum_{i=1}^{m} r_{x_ti}v_i$ from arm pull x_t at time t.

From dual feasibility, we have that $c_j - \sum_{i=1}^{m} r_{ji}v_i \leq 0$. Then, it follows that the stochastic process $Z_0, Z_1, ..., Z_T$ is a submartingale.

Let τ be the stopping time of the optimal dynamic algorithm, i.e. the total number of rounds.

Thus, $Z_{\tau-1}$ equals the algorithm's total cost, plus the gain of the remaining (non-negative) resource supply at the start of round τ.

By Doob's optional stopping theorem, we have that $Z_0 \leq \mathbb{E}[Z_{\tau-1}] \leq \mathrm{OPT}$. □

Let us $\mathrm{REW}_{tot} = \sum_{t=1}^{\tau} c_t$.

The algorithmic approach will make use of dual vectors, computed as follows.

Learning the dual variable: We use the multiplicative weights update method to learn the optimal dual vector. This method raises the cost of a resource exponentially as it is consumed, which ensures that heavily demanded resources become costly, and thereby promotes balanced resource consumption.

Algorithm 16

```
In the first $n$ rounds, pull each arm once
For each arm $x$, define known expected gain vector $R_x \in [0, M]^m$
$v_1 = 1 \in [0, 1]^m$  ▷ $v_t \in [0, 1]^m$ is the round-$t$ estimate of the optimal solution to the dual $v^*$
Set $\epsilon = \sqrt{\frac{M \ln m}{b+M}}$
for rounds $t = n + 1, ..., \tau$ do
    for arm $x \in X$ do
        Set expected gain $g_x = R_x \cdot v_t$
    Pull arm $x = x_t \in X$ that maximizes $g_x / c_x$
    Observe realized reward for each resource $r_x \in [0, M]$
    Update estimated unit gain for each resource $i$ with normalized gain $r_x(i)/M$:  ▷ Cost-based MWU
        $v_{t+1}(i) = v_t(i)(1 - \epsilon)^{\ell}, \ell = r_x(i)/M$
```

Scaled-Hedge: This update scheme is such that for any τ and asequence of vectors $\pi_1, ..., \pi_\tau \in [0, M]^m$, feed in normalized $\pi_1/M, ..., \pi_\tau/M$ vectors into the hedge algorithm and obtain guarantee:

$$\forall y \in \Delta[m], \sum_{t=1}^{\tau} y_t^T \pi_t \leq (1 + \epsilon) \sum_{t=1}^{\tau} y^T \pi_t + \frac{M \ln m}{\epsilon}$$

5.11.4 Algorithm Analysis under Known Arm Means

Let $\hat{R}_t \in [0, M]^{m \times n}$ be the actual gain matrix for round t. The (i, x) entry is the realized gain of resource i in round t if arm x were chosen in this round.

Suppose it holds with probability at least $1 - 1/T$ that the confidence interval for every latent parameter, in every round of execution, contains the true value of that latent parameter. We call this high-probability event a "clean execution".

The regret guarantee will hold almost surely assuming that a clean execution takes place. The regret can be at most $T \cdot M$ when a clean execution does not take place, and since this event has probability at most $1/T$ it contributes only $O(M)$ to the regret. We will henceforth assume a clean execution.

Claim 2. *The Algorithm total cost is such that:*

$$\mathrm{REW}_{tot} - \mathrm{OPT}_{\mathrm{LP}} \leq \left[\tilde{O}\left(\frac{M}{b} + \sqrt{\frac{(b+M)M}{b}} \right) \mathrm{OPT}_{\mathrm{LP}} + nM \right] + \tilde{O}\left(\frac{1}{b} \right) \mathrm{OPT}_{\mathrm{LP}} \| \sum_{t=n+1}^{\tau} E_t z_t \|_\infty$$

where $E_t = R - \hat{R}_t$ under Algorithm 14.

Proof. Let k^* be the optimal solution to the LP-Primal with $\mathrm{OPT}_{\mathrm{LP}} = \sum_{j=1}^{n} c_j k_j^*$. For any realized gains by the algorithm policy, we have the following analysis.

Let $\hat{y} = e_i$, where resource i is (one of) the last resources, whose gain exceeds b:

$$\hat{y}^T \left(\sum_{t=1}^{\tau-1} \hat{R}_t z_t \right) \leq b \Rightarrow \hat{y}^T \left(\sum_{t=n+1}^{\tau-1} \hat{R}_t z_t \right) \leq b$$

Let the total cost after exploration be $\mathrm{REW} = \sum_{t=n+1}^{\tau} c_t$ and define:

$$\bar{y} = \frac{1}{\mathrm{REW}} \sum_{t=n+1}^{\tau-1} c_t y_t$$

Under Algorithm 14, we have at time t, by our choice of x_t, the corresponding z_t must be such that:

$$z_t \in \operatorname*{argmax}_{z \in \Delta(X)} \frac{y_t^T R z}{c^T z}$$

$$\begin{aligned}
b &\leq \bar{y}^T R k^* && \text{(from primal feasibility, } Rk^* \geq b1\text{)} \\
&= \frac{1}{\mathrm{REW}} \sum_{t=n+1}^{\tau-1} c_t y_t^T R k^* && \text{(plug in definition of } \bar{y}\text{)} \\
&= \frac{c^T k^*}{\mathrm{REW}} \sum_{t=n+1}^{\tau-1} c_t y_t^T R \frac{k^*}{c^T k^*} \\
&\leq \frac{\mathrm{OPT_{LP}}}{\mathrm{REW}} \sum_{t=n+1}^{\tau-1} y_t^T R z_t && \text{(since } \tfrac{y_t^T R z_t}{c^T z_t} \geq \tfrac{y_t^T R k^*}{c^T k^*}\text{)} \\
&\leq \min_y \frac{\mathrm{OPT_{LP}}}{\mathrm{REW}} \left[(1+\epsilon) \sum_{t=n+1}^{\tau-1} y^T R z_t + M \ln m / \epsilon \right] \\
& && \text{(since this holds for all } y \in \Delta[m] \text{ using hedge)} \\
&< (1+\epsilon) \frac{\mathrm{OPT_{LP}}}{\mathrm{REW}} \min_y \left[y^T \sum_{t=n+1}^{\tau-1} \hat{R}_t z_t + y^T \sum_{t=n+1}^{\tau-1} E_t z_t + M \ln m / \epsilon \right] && \text{(pull out } (1+\epsilon)\text{)} \\
&\leq (1+\epsilon) \frac{\mathrm{OPT_{LP}}}{\mathrm{REW}} \left[\hat{y}^T \sum_{t=n+1}^{\tau-1} \hat{R}_t z_t + \hat{y}^T \sum_{t=n+1}^{\tau-1} E_t z_t + M \ln m / \epsilon \right] && \text{(choose } y = \hat{y}\text{)} \\
&\leq (1+\epsilon) \frac{\mathrm{OPT_{LP}}}{\mathrm{REW}} \left[b + \hat{y}^T \sum_{t=n+1}^{\tau-1} E_t z_t + M \ln m / \epsilon \right] && \text{(since } \hat{y}^T \left(\textstyle\sum_{t=n+1}^{\tau-1} \hat{R}_t z_t \right) \leq b\text{)}
\end{aligned}$$

From this we get that by setting $\epsilon = \sqrt{\frac{M \ln m}{b+M}}$:

$$\text{REW} \leq \text{OPT}_{\text{LP}} \left((1+\epsilon) + \frac{1+\epsilon}{b} \left[\hat{y}^T \sum_{t=n+1}^{\tau-1} E_t z_t \right] + \frac{1+\epsilon}{b} \frac{M \ln m}{\epsilon} \right)$$

$$\Leftrightarrow \text{REW} - \text{OPT}_{\text{LP}} \leq \text{OPT}_{\text{LP}} \left(\epsilon + \frac{1+\epsilon}{b} \frac{M \ln m}{\epsilon} + \frac{1+\epsilon}{b} \left[\hat{y}^T \sum_{t=n+1}^{\tau-1} E_t z_t \right] \right)$$

$$\Leftrightarrow \text{REW}_{tot} - \text{OPT}_{\text{LP}} \leq \left(\epsilon + \frac{1+\epsilon}{b} \frac{M \ln m}{\epsilon} \right) \text{OPT}_{\text{LP}} + \sum_{t=1}^{n} c_t + \frac{1+\epsilon}{b} \text{OPT}_{\text{LP}} \left[\hat{y}^T \sum_{t=n+1}^{\tau-1} E_t z_t \right]$$

$$\Leftrightarrow \text{REW}_{tot} - \text{OPT}_{\text{LP}} \leq \left(\sqrt{\frac{M \ln m}{b+M}} + \frac{M \ln m}{b} + \frac{\sqrt{(b+M) M \ln m}}{b} \right) \text{OPT}_{\text{LP}}$$

$$+ nM + \frac{1+\epsilon}{b} \text{OPT}_{\text{LP}} \left[\hat{y}^T \sum_{t=n+1}^{\tau-1} E_t z_t \right]$$

$$\Leftrightarrow \text{REW}_{tot} - \text{OPT}_{\text{LP}} \leq \tilde{O} \left(\frac{M}{b} + \sqrt{\frac{(b+M)M}{b}} \right) \text{OPT}_{\text{LP}} + nM + \tilde{O} \left(\frac{1}{b} \right) \text{OPT}_{\text{LP}} \left[\hat{y}^T \sum_{t=n+1}^{\tau-1} E_t z_t \right]$$

□

Remark 19. *This roughly leads to a M factor larger than when $M = 1$, which should yield a $O(\sqrt{\frac{\ln m}{b}} + \frac{\ln m}{b})$ multiplier.*

5.11.4.1 Known R Concentration

For Error Analysis, it remains to bound the error term $\| \sum_{t=n+1}^{\tau} E_t z_t \|_\infty$.

In this case, we observe that each entry of E_t is a mean-zero random variable bounded in $[0, M]$. We may then use Hoeffding and union bound across all m resources to get that:

$$\Pr \left(\| \sum_{t=n+1}^{\tau} E_t z_t \|_\infty \leq (\tau - n - 1)\kappa \right) \geq 1 - m \cdot \left(2 \exp(-2(\tau - n - 1)\kappa^2 / M^2) \right)$$

Setting $1/T = m \cdot \left(2 \exp(-2(\tau - n - 1)\kappa^2 / M^2) \right)$, we obtain that:

$$\kappa = \sqrt{\frac{M^2 \log 2mT}{2(\tau - n - 1)}} \Rightarrow (\tau - n - 1)\kappa = O(M\sqrt{T \log 2mT}).$$

5.11.4.2 Regret Guarantee

Theorem 25. *Algorithm 14 with parameter $\epsilon = \sqrt{\frac{M \ln m}{b+M}}$ attains total regret:*

$$\mathrm{REW}_{tot} - \mathrm{OPT}_{\mathrm{LP}} \leq \tilde{O}\left(\frac{M}{b} + \sqrt{\frac{(b+M)M}{b}}\right)\mathrm{OPT}_{\mathrm{LP}} + \tilde{O}\left(\frac{M\sqrt{T}}{b} + nM\right)$$

Proof. We have that:

$$\begin{aligned}\mathrm{REW}_{tot} - \mathrm{OPT}_{\mathrm{LP}} &\leq \tilde{O}\left(\frac{M}{b} + \sqrt{\frac{(b+M)M}{b}}\right)\mathrm{OPT}_{\mathrm{LP}} + nM + \tilde{O}\left(\frac{1}{b}\right)\left[\hat{y}^T \sum_{t=n+1}^{\tau-1} E_t z_t\right] \\ &\leq \tilde{O}\left(\frac{M}{b} + \sqrt{\frac{(b+M)M}{b}}\right)\mathrm{OPT}_{\mathrm{LP}} + \tilde{O}\left(\frac{\|\sum_{t=n+1}^{\tau} E_t z_t\|_\infty}{b} + Mn\right) \\ &\leq \tilde{O}\left(\frac{M}{b} + \sqrt{\frac{(b+M)M}{b}}\right)\mathrm{OPT}_{\mathrm{LP}} + \tilde{O}\left(\frac{M\sqrt{T}}{b} + Mn\right)\end{aligned}$$

□

Remark 20. *The regret dependence on the number of resources m is $O(\ln m)$.*

We note that Theorem 23 follows from that $\mathrm{OPT}_{\mathrm{LP}} \leq \mathrm{OPT}$.

5.12 Worked through Examples

We work out in close form the test statistic of simple graphs to illustrate our test statistic construction and illustrate how it draws on power from its implications. Having already seen the chain graph example, we turn to the triangle example.

Three-node Triangle Graph: Consider testing $X_1 \to X_3$ in triangle graph $X_1 - X_2 - X_3$.

Since $X_3 \to X_2 \wedge X_2 \to X_1 \Rightarrow X_3 \to X_1, \therefore X_1 \to X_3 \Rightarrow X_2 \to X_3 \vee X_2 \to X_1$.

We have that $T^{1\to3} = (X_2 \to X_3 \wedge X_1 \to X_3) \vee (X_1 \to X_2 \wedge X_1 \to X_3)$.

For path $P = X_2 \to X_3 \wedge X_1 \to X_3$, we have that:

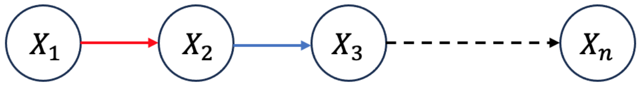
$$E_t^P = \exp\left(\log E_t^{1\to3}(1) + \log E_t^{2\to3}(2) + \max(\log E_t^{1\to3}(3), \log E_t^{2\to3}(3)) - 1/2(\log(2|T_3(t)| - 2))\right)$$

For path $P' = X_1 \to X_2 \wedge X_1 \to X_3$, we have that:

$$E_t^{P'} = \exp\left(\max(\log E_t^{1\to2}(1), \log E_t^{1\to3}(1)) - 1/2(\log(2|T_1(t)| - 2)) + \log E_t^{1\to2}(2) + \log E_t^{1\to3}(3)\right)$$

Thus we see that, asymptotically (ignoring log factors in t), $E_t^{1\to3} = \min(E_t^P, E_t^{P'})$ has *strictly* higher power. This is because both E_t^P and $E_t^{P'}$ have higher power the single-edge e-process corresponding $E_t^{1\to3}(1)E_t^{1\to3}(3) = \exp(\log E_t^{1\to3}(1) + \log E_t^{1\to3}(3))$.

Here, we can also observe that there are two possible updates when node X_3 is intervened upon, which corresponds to a choice when optimizing the test statistic. This naturally later motivates the use of bandit optimization.



$$T^{1\to 2} = (X_1 \to X_2) \wedge (X_2 \to X_3) ... \wedge (X_{n-1} \to X_n)$$

$$E_t^{1\to 2} = \exp\left(\log E_t^{1\to 2}(1) + \sum_{i=2}^{n-1} \max(\log E_t^{i-1\to i1}(i), \log E_t^{i\to i+1}(i)) - 1/2(\log(2|T_i(t)| - 2)) + \log E_t^{n-1\to n}(n)\right)$$

Figure 5.15: Consider testing $X_1 \to X_2$ in the n-node chain graph $X_1 - X_2 - ... - X_n$. This is a graph, where the propagation of edge orientation is crucial for minimizing interventional complexity. We have that $X_1 \to X_2 \Rightarrow X_i \to X_{i+1}$, with which we can derive $T^{1\to 2}$ and $E_t^{1\to 2}$ explicitly. We note that, asymptotically (ignoring log factors in t), $E_t^{1\to 2}$ has much higher power than the e-process of $\exp(\log E_t^{1\to 2}(1) + \log E_t^{1\to 2}(2))$ (from Section 5.3), under non-expanded tree $(1 \to 2)$. $E_t^{1\to 2}$ leverages evidence from for example hypothesis $X_2 \to X_3$ (blue).

Remark 21. *In Figure 5.15, another interesting observation of note is that, due to the combination of evidence, we need not reject any of $X_i \to X_{i+1}$ to reject $X_1 \to X_2$. The cumulative evidence across all $n-1$ edges may be enough for $E_t^{1\to 2}$ to exceed $1/\alpha$, and lead to the rejection of the null. This is despite the data being inconclusive for any of the downstream edges (i.e. $E_t^{i\to i+1}$ need not exceed $1/\alpha$).*

Chapter 6

Strategic Auditing

6.1 Introduction

With the growing usage of artificial intelligence across industries, governance efforts are increasingly ramping up. A key challenge in regulatory efforts is the problem of scalability. Even for well-resourced countries like Norway, which is a pioneer in AI governance, regulators are only able to monitor and engage with a "small fraction of the companies" [198]. This growing issue calls for a better understanding of *efficient* algorithms that can audit machine learning (ML) models. We are particularly interested in the strategic auditing case, where the company whose model is under audit is aware of the auditing strategy. Towards understanding this process, we begin by formalizing the problem of auditing.

Problem Formulation: A regulatory institution is interested in auditing an unknown model $h^* : \mathcal{X} \to \{-1, 1\}$ held by a company (e.g. a lending company in the finance sector), where $\mathcal{X}$ is the feature space (e.g. of all information supplied by users). We assume that the regulatory institution only has knowledge of the hypothesis class $\mathcal{H}$ where h^* comes from (e.g. the family of linear classifiers), and it would like to estimate $\mu(h^*)$ for a function μ that measures the model property of interest. To this end, the institution is allowed to send black-box queries to the model h^*, i.e. send the company a query example x and receive $h^*(x)$. The regulatory institution's goal is to *efficiently* estimate $\mu(h^*)$ to within an error of at most $\epsilon > 0$.

We measure an algorithm's *efficiency* in terms of both its *query complexity* and *computational complexity*. Having an auditing algorithm with low query and computational complexity naturally helps to address the scalability challenge: greater efficiency means that each audit may be processed faster and more audits may be processed at a time.

Property of Interest: While which properties μ to assess is still heavily debated by regulators, we initiate the study of algorithms that audit fairness, a mainstay in regulatory efforts. In particular, we will take μ to be Demographic Parity (DP)[1]: given distribution D_X over $\mathcal{X} \times \{0, 1\}$ (where feature x and sensitive attribute x_A are jointly drawn from), $\mu_{D_X}(h) = \Pr_{(x,x_A)\sim D_X}(h(x) = 1|x_A = 1) - \Pr_{(x,x_A)\sim D_X}(h(x) = 1|x_A = 0)$. For brevity, when it is clear from context, we abbreviate $\Pr_{D_X}, \mu_{D_X}$ as $\Pr, \mu$, respectively. DP measures the degree of disparate treatment of model h on the two sub-populations $x \mid x_A = 0$ and $x \mid x_A = 1$, which we assume are non-

[1] While fairness is the focus of our work, our algorithm may be adapted to any μ which is a function of $\mathcal{X}$ and h^*.

negligible: $\underline{p} := \min(\Pr(x_A = 1), \Pr(x_A = 0)) = \Omega(1)$. Achieving a small Demographic Parity may be thought of as a stronger version of the US Equal Employment Opportunity Commission's "four-fifths rule".[2]

To focus on query complexity, we will abstract away the difficulty of evaluating μ by assuming that D_X is known, which means that for any h we may evaluate $\mu(h)$ to arbitrary precision; for instance, this may be achieved with the availability of an arbitrarily large number of (unlabeled) samples randomly drawn from $x \mid x_A = 0$ and $x \mid x_A = 1$. Our main challenge is that we do not know h^* and only want to query h^* *insofar as to be able to accurately estimate* $\mu(h^*)$.

Guarantees of the Audit: In this chapter, we investigate algorithms that can provide two types of guarantees. The first is the natural, *direct estimation accuracy*: the estimate returned by the algorithm should be within ϵ of $\mu(h^*)$.

The second is that of *manipulation-proof* (MP) estimation. Audits can be very consequential to companies as they may be subject to hefty penalties if caught with violations. Not surprisingly, there have been *strategic* attempts in the past to avoid being caught with violations [e.g. 140] by "gaming" the audit. We formulate our notion of manipulation-proofness in light of one way the audit may be strategically gamed, which we now describe. Note that all the auditor knows about the model used by the company is that it is consistent with the queried labels in the audit. So, while our algorithm may have estimated $\mu(h^*)$ accurately during audit-time, nothing stops the company from changing its model *post-audit* from h^* to a different model $h_{\text{new}} \in \mathcal{H}$ (e.g to improve profit), so long as h_{new} is still consistent with the queries seen during the audit. With this, we also look to understand: given this post-hoc possibility of manipulation, can we devise an algorithm that nonetheless ensures the algorithm's estimate is within ϵ of $\mu(h_{\text{new}})$?

Indeed, a robust set of audit queries would serve as a *certificate* that no matter which model the company changes to after the audit, its μ-estimation would remain accurate. Given a set of classifiers V, a classifier h, and a unlabeled dataset S, define the version space [207] induced by S to be $V(h, S) := \left\{h' \in V : h'(S) = h(S)\right\}$. An auditing algorithm is ϵ-manipulation-proof if, for any h^*, it outputs a set of queries S and estimate $\hat{\mu}$ that guarantees that $\max_{h \in \mathcal{H}(h^*, S)} \left|\mu(h) - \hat{\mu}\right| \leq \epsilon$.

Baseline: i.i.d Sampling: One natural baseline that comes to mind for the direct estimation is i.i.d sampling. We sample $O(1/\epsilon^2)$ examples i.i.d from the distribution $x \mid x_A = i$ for $i \in \{0, 1\}$, query h^* on these examples and take the average to obtain an estimate of $\Pr(h^*(x) = +1 \mid x_A = i)$. Finally, we take the difference of these two estimates as our final DP estimate. By Hoeffding's Inequality, with high probability, this estimate is ϵ-accurate, and this estimation procedure makes $O(1/\epsilon^2)$ queries.

However, i.i.d sampling is not necessarily MP. To see an example, let there be $2n$ points in group $x_A = 1$ with $n = 1/\epsilon^2$ that are shattered by $\mathcal{H}$ and D_X is uniform over these points. Suppose that all points in group $x_A = 0$ are labeled the same: $\Pr_{D_X}(h(x) = 1 | x_A = 0) = 0, \forall h \in \mathcal{H}$. Then, μ-estimation reduces to estimating the proportion of positives in group $x_A = 1$. i.i.d sampling will randomly choose n of these data points to see, and it will produce an ϵ-accurate estimate of $\mu(h^*)$. However, we do not see the other n points. Since the $2n$ points are shattered by $\mathcal{H}$, *after* the queried points are determined, we see that the company can increase or decrease

[2] The "selection rate for any race, sex, or ethnic group [must be at least] four-fifths (4/5) (or eighty percent) of the rate for the group with the highest rate."

DP by up to $1/2$ by switching to a different model.

To perform both direct and MP estimation, it seems promising then to examine algorithms that make use of *non-iid* sampling. Moreover, for MP, we observe that the auditing algorithm should leverage knowledge of the hypothesis class as well, which i.i.d sampling is agnostic to.

Baseline: Active Learning: *PAC active learning* [135] (where PAC stands for Probably Approximately Correct [282]) algorithms are a set of algorithms that can achieve both direct and MP estimation accuracy. PAC active learning algorithms guarantee that, with high probability, $\hat{h}$ in the resultant version space is such that $\mathbb{P}(\hat{h}(x) \neq h^*(x)) \leq \underline{p}\epsilon = O(\epsilon)$. With this, we have $\left|\mu(\hat{h}) - \mu(h^*)\right| \leq \epsilon$ (see Lemma 41 in Appendix 6.7 for a formal proof).

To mention a setting where learning is favored over i.i.d sampling, learning homogeneous linear classifiers under certain well-behaved unlabeled data distributions requires only $O(d \log 1/\epsilon)$ queries [e.g. 30, 82] and would thus be far more efficient than $O(1/\epsilon^2)$ for low-dimensional learning settings with high auditing precision requirements.

Still, as our goal is only to estimate the μ values of the induced version space, it is unclear if we always need to go as far as to learn the model itself. In this chapter, we investigate whether, and if so when, it may be possible to design adaptive approaches to efficiently directly and MP estimate $\mu(h^*)$ using knowledge of $\mathcal{H}$.

To the best of our knowledge, we are the first to theoretically investigate active approaches for direct and MP estimation of $\mu(h^*)$. Our first exploration of active fairness estimation seeks to provide a more complete picture of the theory of auditing machine learning models. Our hope is that our theoretical results can pave the way for subsequent development of practical and efficient algorithms.

Our Contributions: Our main contributions are on two fronts, MP and direct estimation of $\mu(h^*)$:

- For the newly introduced notion of manipulation-proofness, we identify a statistically optimal, but computationally intractable deterministic algorithm. We gain insights into its query complexity through comparisons to the two baselines, i.i.d sampling and PAC active learning.
- In light of the computational intractability of the optimal deterministic algorithm, we design a randomized algorithm that enjoys *oracle efficiency* [e.g. 84]: it has an efficient implementation given access to a mistake-bounded online learning oracle, and an constrained empirical risk minimization oracle for the hypothesis class $\mathcal{H}$. Furthermore, its query performance matches that of the optimal deterministic algorithm up to $\mathrm{polylog}|\mathcal{H}|$ factors.
- Finally, on the direct estimation front, we obtain bounds on information-theoretic query complexity. We establish that MP estimation may be more expensive than direct estimation, thus highlighting the need to develop separate algorithms for the two guarantees. Then, we establish the usefulness of randomization in algorithm design and develop an optimal, randomized algorithm for linear classification under Gaussian subpopulations. Finally, to shed insight on auditing in general settings, we develop distribution-free lower bounds for direction estimation under general VC classes. This lower bound charts the query complexity that any optimal randomized auditing algorithms must attain.

6.1.1 Additional Notations

We now introduce some additional useful notation used throughout the chapter. Let $[m]$ denote $\{1, ..., m\}$. For an unlabeled dataset S, and two classifiers h, h', we say $h(S) = h'(S)$ if for all $x \in S$, $h(x) = h'(x)$. Given a set of classifiers V and a labeled dataset T, define $V[T] := \left\{ h \in V : \forall (x,y) \in T, h(x) = y \right\}$. Furthermore, denote by $V_x^y = V\left[\left\{(x,y)\right\}\right]$ for notational simplicity. Given a set of classifiers V and fairness measure μ, denote by $\mathrm{diam}_\mu(V) := \max_{h,h' \in V} \mu(h) - \mu(h')$ the *μ-diameter* of V. Given a set of labeled examples T, denote by $\Pr_T(\cdot)$ the probability over the uniform distribution on T; given a classifier h, denote by $\mathrm{err}(h,T) = \Pr_T(h(x) \neq y)$ the empirical error of h on T.

Throughout this chapter, we will consider active fairness auditing under the membership query model, similar to membership query-based active learning [13]. Specifically, a deterministic active auditing algorithm $\mathcal{A}$ with label budget N is formally defined as a collection of $N+1$ (computable) functions $f_1, f_2, \ldots, f_N, g$ such that:

1. For every $i \in [N]$, $f_i : (\mathcal{X} \times \mathcal{Y})^{i-1} \to \mathcal{X}$ is the label querying function used at step i, that takes into input the first $(i-1)$ labeled examples $\langle (x_1, y_1), \ldots, (x_{i-1}, y_{i-1}) \rangle$ obtained so far, and chooses the i-th example x_i for label query.
2. $g : (\mathcal{X} \times \mathcal{Y})^N \to \mathbb{R}$ is the estimator function that takes into input all N labeled examples $\langle (x_1, y_1), \ldots, (x_N, y_N) \rangle$ obtained throughout the interaction process, and outputs $\hat{\mu}$, the estimate of $\mu(h^*)$.

When $\mathcal{A}$ interacts with a target classifier h, let the resultant queried unlabeled dataset be $S_{\mathcal{A},h} = \langle x_1, \ldots, x_N \rangle$, and the final μ estimate be $\hat{\mu}_{\mathcal{A},h}$.

Similar to deterministic algorithms, a randomized active auditing algorithm $\mathcal{A}$ with label budget N and B bits of random seed is formally defined as a collection of $N+1$ (computable) functions $f_1, \ldots, f_N, g$, where $f_i : (\mathcal{X} \times \mathcal{Y})^{i-1} \times \{0,1\}^B \to \mathcal{X}$ and $g : (\mathcal{X} \times \mathcal{Y})^N \times \{0,1\}^B \to \mathbb{R}$. Note that each function now take as input a B-bit random seed; as a result, when $\mathcal{A}$ interacts with a fixed h^*, its output $\hat{\mu}$ is now a random variable. Note also that under the above definition, a randomized active auditing algorithm $\mathcal{A}$ that uses a fixed seed b may be viewed as a deterministic active auditing algorithm $\mathcal{A}_b$.

We will be comparing our algorithms' query complexities with those of disagreement-based active learning algorithms [75, 135]. Given a classifier h and $r > 0$, define $B(h,r) = \left\{ h' \in \mathcal{H} : \Pr_{D_X}\left(h'(x) \neq h(x)\right) \leq r \right\}$ as the disagreement ball centered at h with radius r. Given a set of classifiers V, define its disagreement region $\mathrm{DIS}(V) = \left\{ x \in \mathcal{X} : \exists h, h' \in V : h(x) \neq h'(x) \right\}$. For a hypothesis class $\mathcal{H}$ and an unlabeled data distribution D_X, an important quantity that characterizes the query complexity of disagreement-based active learning algorithm is the *disagreement coefficient* $\theta(r)$, defined as

$$\theta(r) = \sup_{h \in \mathcal{H}, r' \geq r} \frac{\Pr_{D_X}(x \in \mathrm{DIS}(B(h,r')))}{r'}.$$

6.2 Related Work

Our work is most related to the following two lines of work, both of which are concerned with estimating some property of a model without having to learn the model itself.

Sample-Efficient Optimal Loss Estimation: Dicker [93], Kong and Valiant [167] propose U-statistics-based estimators that estimate the optimal population mean square error in d-dimensional linear regression, with a sample complexity of $O(\sqrt{d})$ (much lower than $O(d)$, the sample complexity of learning optimal linear regressor). Kong and Valiant [167] also extend the results to a well-specified logistic regression setting, where the goal is to estimate the optimal zero-one loss. Our work is similar in focusing on the question of efficient $\mu(h^*)$ estimation without having to learn h^*. Our work differs in focusing on fairness property instead of the optimal MSE or zero-one loss. Moreover, our results apply to arbitrary $\mathcal{H}$, and not just to linear models.

Interactive Verification: Goldwasser et al. [123] studies verification of whether a model h's loss is near-optimal with respect to a hypothesis class $\mathcal{H}$ and looks to understand when verification is cheaper than learning. They prove that verification is cheaper than learning for specific hypothesis classes and is just as expensive for other hypothesis classes. Again, our work differs in focusing on a different property of the model, fairness.

Our algorithm also utilizes tools from active learning and machine teaching, which we review below.

Active Learning and Teaching: The task of learning h^* approximately through membership queries has been well-studied [e.g. 13, 83, 131, 132, 138]. Our computationally efficient algorithm for active fairness auditing is built upon the connection between active learning and machine teaching [121], as first noted in Hanneke [132], Hegedűs [138]. To achieve computational efficiency, our work builds on recent work on black-box teaching [85], which implicitly gives an efficient procedure for computing an approximate-minimum specifying set; we adapt Dasgupta et al. [85]'s algorithm to give a similar procedure for approximating the minimum specifying set that specifies the μ value.

In the interest of space, please see discussion of additional related work in Appendix 6.5.

6.3 Manipulation-Proof Algorithms

6.3.1 Optimal Deterministic Algorithm

We begin our study of the MP estimation of $\mu(h^*)$ by identifying an optimal deterministic algorithm based on dynamic programming. Inspired by a minimax analysis of exact active learning with membership queries [131], we recursively define the following value function for any version space $V \subseteq \mathcal{H}$:

$$\mathrm{Cost}(V) = \begin{cases} 0, & \mathrm{diam}_\mu(V) \leq 2\epsilon \\ 1 + \min_x \max_y \mathrm{Cost}(V[(x,y)]), & \text{otherwise} \end{cases}$$

Note that $\mathrm{Cost}(V)$ is similar to the minimax query complexity of exact active learning [131], except that the induction base case is different – here the base case is $\mathrm{diam}_\mu(V) \leq 2\epsilon$, which

Algorithm 17 Minimax optimal deterministic auditing

```
Require: Finite hypothesis class $\mathcal{H}$, target error $\epsilon$, fairness measure $\mu$
Ensure: $\hat{\mu}$, an estimate of $\mu(h^*)$
1: Let $V \leftarrow \mathcal{H}$
2: while $\mathrm{diam}_\mu(V) > 2\epsilon$ do
3:     Query $x \in \mathrm{argmin}_x \max_y \mathrm{Cost}\left(V_x^y\right)$, obtain label $h^*(x)$
4:     $V \leftarrow V(h^*, \{x\})$
   return $\frac{1}{2}\left(\max_{h\in V}\mu(h) + \min_{h\in V}\mu(h)\right)$
```

implies that subject to $h^* \in V$, we have identified $\mu(h^*)$ up to error ϵ. In contrast, in exact active learning, Hanneke [131]'s induction base case is $|V| = 1$, where we identify h^* through V.

The value function Cost also has a game-theoretic interpretation. Imagine that a learner plays a multi-round game with an adversary. The learner makes sequential queries of examples to obtain their labels, and the adversary reveals the labels of the examples, subject to the constraint that all labeled examples shown agree with some classifier in $\mathcal{H}$. The version space V encodes the state of the game: it is the set of classifiers that agrees with all the labeled examples shown so far in the game. The interaction between the learner and the adversary ends when all classifiers in V has μ values 2ϵ-close to each other. The learner would like to minimize its total cost, which is the number of rounds. $\mathrm{Cost}(V)$ can be viewed as the minimax-optimal future cost, subject to the game's current state being represented by version space V.

Based on the notion of Cost, we design an algorithm, Algorithm 17, that has a worst-case label complexity at most $\mathrm{Cost}(\mathcal{H})$. Specifically, it maintains a version space $V \subset \mathcal{H}$, initialized to $\mathcal{H}$ (line 1). At every iteration, if the μ-diameter of V, $\mathrm{diam}_\mu(V) = \max_{h,h'\in V} \mu(h) - \mu(h')$, is at most 2ϵ, then since $\mu(h^*) \in I = [\min_{h\in V}\mu(h), \max_{h\in V}\mu(h)]$ returning the midpoint of I gives us an ϵ-accurate estimate of $\mu(h^*)$ (line 4). Otherwise, Algorithm 17 makes a query by choosing the x that minimizes the worst-case future value functions (line 3). After receiving $h^*(x)$, it updates its version space V (line 4). By construction, the interaction between the learner and the labeler lasts for at most $\mathrm{Cost}(V)$ rounds, which gives the following theorem.

Theorem 26. *If Algorithm 17 interacts with some $h^* \in \mathcal{H}$, then it outputs $\hat{\mu}$ such that $\left|\hat{\mu} - \mu(h^*)\right| \leq \epsilon$, and queries at most $\mathrm{Cost}(\mathcal{H})$ labels.*

By the minimax nature of Cost, we also show that among all deterministic algorithms, Algorithm 17 has the optimal worst-case query complexity:

Theorem 27. *If $\mathcal{A}$ is a deterministic algorithm with query budget $N \leq \mathrm{Cost}(\mathcal{H}) - 1$, there exists some $h^* \in \mathcal{H}$, such that $\hat{\mu}$, the output of $\mathcal{A}$ after querying h^*, satisfies $\left|\hat{\mu} - \mu(h^*)\right| > \epsilon$.*

The proofs of Theorems 26 and 27 are deferred to Appendix 6.8.1.

6.3.1.1 Comparison to Baselines

To gain a better understanding of $\mathrm{Cost}(\mathcal{H})$, we relate it to the label complexity of the two baselines, i.i.d sampling and active learning. To establish the comparison, we prove that we can derandomize existing i.i.d sampling-based and active learning-based auditing algorithms with a small overhead on label complexity.

Our first result is that the label complexity of Algorithm 17 is within a factor of $O(\ln |\mathcal{H}|)$ of the label complexity of i.i.d sampling.

Proposition 35. $\mathrm{Cost}(\mathcal{H}) \leq O\left(\frac{1}{\epsilon^2} \ln |\mathcal{H}|\right)$.

Our second result is that the label complexity of Algorithm 17 is always no worse than the distribution-dependent label complexity of CAL [75, 135], a well-known PAC active learning algorithm. We believe that similar bounds comparing $\mathrm{Cost}(\mathcal{H})$ to the complexity of generic active learning algorithms can also be shown; these algorithms include the Splitting Algorithm [82] or the confidence-based algorithm of Zhang and Chaudhuri [318], through suitable derandomization procedures.

Proposition 36. $\mathrm{Cost}(\mathcal{H}) \leq O\left(\theta(\epsilon) \cdot \ln |\mathcal{H}| \cdot \ln \frac{1}{\epsilon}\right)$*, where* θ *is the disagreement coefficient of* $\mathcal{H}$ *with respect to* D_X *(recall Section 6.1.1 for its definition).*

Proof sketch. We present Algorithm 18, which is a derandomized version of the Phased CAL algorithm [142, Chapter 2]. To prove this proposition, using Theorem 27, it suffices to show that Algorithm 18 has a deterministic label complexity bound of $O\left(\theta(\epsilon) \cdot \ln |\mathcal{H}| \cdot \ln \frac{1}{\epsilon}\right)$. We only present the main idea here, and defer a precise version of the proof to Appendix 6.8.3.

We first show that for every n, the optimization problem in line 7 is always feasible. To see this, observe that if we draw S_n, a sample of size m_n, drawn i.i.d from D_X, we have:

1. By Bernstein's inequality, with probability $1 - \frac{1}{4}$,

$$\mathrm{Pr}_{S_n}(x \in \mathrm{DIS}(V_n)) \leq 2 \, \mathrm{Pr}_{D_X}(x \in \mathrm{DIS}(V_n)) + \frac{\ln 8}{m_n},$$

2. By Bernstein's inequality and union bound over $h, h' \in \mathcal{H}$, we have with probability $1 - \frac{1}{4}$,

$$\begin{aligned} \forall h, h' \in \mathcal{H}: \quad & \mathrm{Pr}_S(h(x) \neq h'(x)) = 0 \\ \implies & \mathrm{Pr}_{D_X}(h(x) \neq h'(x)) \leq \frac{16 \ln |\mathcal{H}|}{m_n}. \end{aligned}$$

By union bound, with nonzero probability, the above two condition hold simultaneously, showing the feasibility of the optimization problem.

We then argue that for all n, $V_{n+1} \subseteq B(h^*, \frac{16 \ln |\mathcal{H}|}{m_n})$. This is because for each $h \in V_{n+1}$, h and h^* are both in V_n and therefore they agree on $S_n \setminus T_n$; on the other hand, h and h^* agree on T_n by the definition of of V_{n+1}. As a consequence, $\mathrm{Pr}_{S_n}(h(x) \neq h^*(x)) = 0$, which implies that $\mathrm{Pr}_{D_X}(h(x) \neq h^*(x)) \leq \frac{16 \ln |\mathcal{H}|}{m_n}$. As a consequence, for all $h \in V_{N+1}$, $\mathrm{Pr}(h(x) \neq h^*(x)) \leq \underline{p}\epsilon$, which, combined with Lemma 41, implies that $\left|\mu(h) - \mu(h^*)\right| \leq \epsilon$.

Finally, to upper bound Algorithm 18's label complexity:

$$\begin{aligned} \sum_{n=1}^{N} |T_n| =& \sum_{n=1}^{N} m_n \cdot (2 \, \mathrm{Pr}_{D_X}(x \in \mathrm{DIS}(V_n)) + \frac{\ln 8}{m_n}) \\ \leq& \sum_{n=1}^{N} m_n \cdot (2\theta(\epsilon) \frac{16 \ln |\mathcal{H}|}{m_n} + \frac{\ln 8}{m_n}) \\ \leq& O\left(\theta(\epsilon) \cdot \ln |\mathcal{H}| \cdot \ln \frac{1}{\epsilon}\right). \end{aligned}$$

□

Algorithm 18 Derandomized Phased CAL for Auditing

```
Require: Hypothesis class $\mathcal{H}$, target error $\epsilon$, minority population proportion $p_{\text{minor}}$, fairness measure $\mu$
Ensure: $\hat{\mu}$, an estimate of $\mu(h^*)$
Let $N = \lceil \log_2 \frac{16 \ln |\mathcal{H}|}{p_{\text{minor}} \epsilon} \rceil$
Let $V_1 \leftarrow \mathcal{H}$
for $n = 1, \ldots, N$ do
    Let $m_n = 2^n$
    Find (the lexicographically smallest) $S_n \in \mathcal{X}^{m_n}$ such that:
        $\Pr_{S_n}(x \in \text{DIS}(V_n)) \leq 2 \Pr_{D_X}(x \in \text{DIS}(V_n)) + \frac{\ln 8}{m_n}$
        and $\forall h, h' \in \mathcal{H}$: if $\Pr_{S_n}(h(x) \neq h'(x)) = 0$ then $\Pr_{D_X}(h(x) \neq h'(x)) \leq \frac{16 \ln |\mathcal{H}|}{m_n}$
    Query $h^*$ for the labels of examples in $T_n := S_n \cap \text{DIS}(V_n)$
    $V_{n+1} \leftarrow V_n(h^*, T_n)$
return $\mu(h)$ for an arbitrary $h \in V_{N+1}$
```

6.3.1.2 Computational Hardness of Implementing Algorithm 17

Although Algorithm 17 attains the optimal label complexity of deterministic algorithms, we show in the following proposition that, under standard complexity-theoretic assumptions (NP $\not\subseteq$ $\text{TIME}(n^{O(\log \log n)})$), even approximating $\text{Cost}(\mathcal{H})$ is computationally intractable.

Proposition 37. *If there is an algorithm that can approximate* $\text{Cost}(\mathcal{H})$ *to within* $0.3 \ln |\mathcal{H}|$ *factor in* $\text{poly}(|\mathcal{H}|, |\mathcal{X}|, 1/\epsilon)$ *time, then* $\text{NP} \subseteq \text{TIME}(n^{O(\log \log n)})$.

We remark that the constant 0.3 can be improved to a constant arbitrarily smaller than 1. The main insight behind this proposition is a connection between $\text{Cost}(\mathcal{H})$ and optimal-depth decision trees (see Theorem 30). Using the hardness of computing an approximately-optimal-depth decision tree [173] and taking into account the structure of μ, we establish the intractability of approximating $\text{Cost}(\mathcal{H})$.

Owing to the intractability of Algorithm 17, in the next section, we turn to the design of a computationally efficient algorithm whose label complexity nears that of Algorithm 17 (i.e. $\text{Cost}(\mathcal{H})$).

6.3.2 Efficient Randomized Algorithm with Competitive Guarantees

We present our efficient algorithm in this section, which also serves as a first upper bound on the statistical complexity of computationally tractable algorithms. Our algorithm, Algorithm 19, is inspired by the exact active learning literature [132, 138], based on a connection between machine teaching [121] and active learning.

Algorithm 19 takes into input two oracles, a mistake-bounded online learning oracle $\mathcal{O}$ and an constrained empirical risk minimization (ERM) oracle C-ERM, defined below.

Definition 30. *An* online-learning oracle $\mathcal{O}$ *is said to have a mistake bound of* M *for hypothesis class* $\mathcal{H}$*, if for any classifier* $h^* \in \mathcal{H}$*, and any sequence of examples* $x_1, x_2, \ldots$*, at every round* $t \in \mathbb{N}$*, given historical examples* $(x_s, h^*(x_s))_{s=1}^{t-1}$*, outputs classifier* $\hat{h}_t$ *such that* $\sum_{t=1}^{\infty} I(\hat{h}_t(x_t) \neq h^*(x_t)) \leq M$.

Well-known implementations of mistake bounded online learning oracle include the halving algorithm and its efficient sampling-based approximations [43] as well as the Perceptron / Winnow algorithm [40, 189].

For instance, if $\mathcal{O}$ is the halving algorithm, a mistake bound of $M = \log_2 |\mathcal{H}|$ may be achieved.

We next define the constrained ERM oracle, which has been previously used in a number of works on oracle-efficient active learning [84, 134, 145].

Definition 31. *An* constrained ERM oracle *for hypothesis class $\mathcal{H}$,* C-ERM*, is one that takes as input labeled datasets A and B, and outputs a classifier $\hat{h} \in \operatorname{argmin}\left\{\operatorname{err}(h, A) : h \in \mathcal{H}, \operatorname{err}(h, B) = 0\right\}$.*

The high-level idea of Algorithm 19 is as follows: at every iteration, it uses the mistake-bounded online learning oracle to generate some classifier $\hat{h}$ (line 3); then, it aims to construct a dataset T of small size, such that after querying h^* for the labels of examples in T, one of the following two happens: (1) $\hat{h}$ disagrees with h^* on some example in T; (2) for all classifiers in the version space $V = \left\{h \in \mathcal{H} : \forall x \in T, h(x) = h^*(x)\right\}$, we have $\operatorname{diam}_\mu(V) \leq 2\epsilon$. In case (1), we have found a counterexample for $\hat{h}$, which can be fed to the online learning oracle to learn a new model, and this can happen at most M times; in case (2), we are done: our queried labeled examples ensure that our auditing estimate is ϵ-accurate, and satisfies manipulation-proofness. Dataset T of such property is called a (μ, ϵ)*-specifying set* for $\hat{h}$, as formally defined in Definition 34 in Appendix 6.8.5.

Another view of the μ-specifying set is a set T such that for all h, h' with $\mu(h) - \mu(h') > 2\epsilon$, there exists some $x \in T$, such that $h(x) \neq \hat{h}(x)$ or $h'(x) \neq \hat{h}(x)$. The requirements on T can be viewed as a set cover problem, where the universe U is $\left\{(h, h') \in \mathcal{H}^2 : \mu(h) - \mu(h') > 2\epsilon\right\}$, and the set system is $\mathcal{C} = \{C_x : x \in \mathcal{X}\}$, where (h, h') is in C_x if $h(x) \neq \hat{h}(x)$ or $h'(x) \neq \hat{h}(x)$.

This motivates us to design efficient set cover algorithms in this context. A key challenge of applying standard offline set cover algorithms (such as the greedy set cover algorithm) to construct approximate minimum (μ, ϵ)-specifying set is that we cannot afford to enumerate all elements in the universe U as U can be exponential in size.

In face of this challenge, we draw inspiration from online set cover literature [9, 85] to design an oracle-efficient algorithm that computes $O(\log |\mathcal{H}| \log |\mathcal{X}|)$*-approximate* minimum (μ, ϵ)-specifying sets, which avoids enumeration over U.

Our key idea is to simulate an online set cover process. We build the cover set[3] T iteratively, starting from $T = \emptyset$ (line 5). At every inner iteration, we first try to find a pair (h_1, h_2) in U not yet covered by the current T. As we shall see next, this step (line 9) can be implemented efficiently given the constrained ERM oracle C-ERM. If such a pair (h_1, h_2) can be found, we use the online set cover algorithm implicit in [85] to find a new example that covers this pair, add it to T, and move onto the next iteration (lines 14 to 17). Otherwise, T has successfully covered all the elements in U, in which case we break the inner loop (line 11).

To see how line 9 finds an uncovered pair in U, we note that it can be also written as:

$$(h_1, h_2) = \operatorname*{argmax}_{h, h' \in \mathcal{H}} \left\{\mu(h) - \mu(h') : h(T) = h'(T) = \hat{h}(T)\right\}$$

Thus, if $\mu(h_1) - \mu(h_2) > 2\epsilon$, then the returned pair (h_1, h_2) corresponds to a pair in universe U that is not covered by T. Otherwise, by the optimality of (h_1, h_2), T covers all elements in U.

[3] When it is clear from context, we slightly abuse notations and say "x covers (h, h')" if $(h, h') \in C_x$.

Furthermore, we note that optimization problems (8) and (9) can be implemented with access to C-ERM. We show this for program (8) and the reasoning for program (9) is analogous. Observe that maximizing $\mu(h)$ from $h \in \mathcal{H}$ subject to constraint $h(T) = \hat{h}(T)$ is equivalent to minimizing (a weighted) empirical error of $h \in \mathcal{H}$ on dataset $\left\{(x, +1) : x \in \mathcal{X}, x_A = 0\right\} \cup \left\{(x, -1) : x \in \mathcal{X}, x_A = 1\right\}$, subject to h having zero error on $\{(x, \hat{h}(x)) : x \in T\}$.

We are now ready to present the label complexity guarantee of Algorithm 19.

Algorithm 19 Oracle-efficient Active Fairness Auditing

```
Require: Hypothesis class $\mathcal{H}$, online learning oracle $\mathcal{O}$ with mistake bound $M$, constrained ERM
    oracle C-ERM, target error $\epsilon$, fairness measure $\mu$.
Ensure: $\hat{\mu}$, an estimate of $\mu(h^*)$
Initialize $S \leftarrow \emptyset$
while True do
    $\hat{h} \leftarrow \mathcal{O}(S)$
    Let $T \leftarrow \emptyset$
                                  ▷ Computing an approximate minimum $(\mu, \epsilon)$-specifying set for $\hat{h}$
    Initialize weights $w(x) = \frac{1}{|\mathcal{X}|}$ and threshold $\tau_x \sim \mathrm{Exponential}(\ln(|\mathcal{H}|^2 M/\delta))$ ▷ random
initialization of thresholds
    while true do
        Use C-ERM to solve program:                    ▷ $T$ is an $(\mu, \epsilon)$-specifying set for $\hat{h}$
         $h_1 \leftarrow$ find $\max_{h \in \mathcal{H}} \mu(h)$, s.t. $h(T) = \hat{h}(T)$
        Use C-ERM to solve program:
         $h_2 \leftarrow$ find $\min_{h \in \mathcal{H}} \mu(h)$, s.t. $h(T) = \hat{h}(T)$
        if $\mu(h_1) - \mu(h_2) \leq 2\epsilon$ then
            break
        else
                     ▷ Add examples to $T$ to cover $(h_1, h_2)$, using the online set cover algorithm
implicit in [85]
Determine $\Delta(h_1, h_2) = \{x \in \mathcal{X} : h_1(x) \neq \hat{h}(x)$ or $h_2(x) \neq \hat{h}(x)\}$
            while $\sum_{x \in \Delta(h_1,h_2)} w(x) \leq 1$ do
                Double weights $w(x)$ for all $x$ in $\Delta(h_1, h_2)$
                Update $T \leftarrow \left\{x \in \mathcal{X} : w(x) \geq \tau_x\right\}$
    Query $h^*$ on $T$
    $S \leftarrow S \cup T$
    if $\hat{h}(T) = h^*(T)$ then return $\frac{1}{2}(\mu(h_1) + \mu(h_2))$
```

Theorem 28. *If the online learning oracle $\mathcal{O}$ makes a total of M mistakes, then with probability $1 - \delta$, Algorithm 19 outputs $\hat{\mu}$ such that $\left|\hat{\mu} - \mu(h^*)\right| \leq \epsilon$, with its number of label queries bounded by:*

$$O\left(\mathrm{Cost}(\mathcal{H}) M \log \frac{|\mathcal{H}| M}{\delta} \log |\mathcal{X}|\right).$$

The proof of Theorem 28 is deferred to Appendix 6.8.5. In a nutshell, it combines the following observations. First, Algorithm 19 has at most M outer iterations using the mistake bound guarantee of oracle $\mathcal{O}$. Second, for each $\hat{h}$ in each inner iteration, its minimum (μ, ϵ)-specifying set has size at most $\mathrm{Cost}(\mathcal{H})$; this is based on a nontrivial connection between the optimal deterministic query complexity and (μ, ϵ)-extended teaching dimension (see Definition 36), which we present in Lemma 44. Third, by the $O\left(\log \frac{|\mathcal{H}|M}{\delta} \log |\mathcal{X}|\right)$-approximation guarantee of the online set cover algorithm implicit in [85], each outer iteration makes at most $O\left(\mathrm{Cost}(\mathcal{H}) \log \frac{|\mathcal{H}|M}{\delta} \log |\mathcal{X}|\right)$ label queries.

Remark 22. *Via an argument similar to that in Proposition 37, we can show that, for computationally-efficient algorithms, the approximation factor in constructing an approximately-minimum (μ, ϵ)-specifying set for $\hat{h}$ cannot be significantly improved to, say,* $0.3 \ln |\mathcal{H}|$.

6.4 Statistical Limits of Estimation

In this section, we turn to direct estimation, the second of the two main guarantees one may wish to have for auditing. In particular, we focus on the statistical limits of direct estimation, which involves designing an efficient auditing algorithm that can output $\hat{\mu}$ such that $\left|\hat{\mu} - \mu(h^*)\right| \leq \epsilon$ with a small number of queries.

6.4.1 Separation between Estimation with and without Manipulation-proofness

To start, it is natural to contrast the guarantee of ϵ-manipulation-proofness against ϵ-direct estimation accuracy. Indeed, if the two guarantees are one and the same, we may simply use the MP estimation algorithms for direct estimation as well.

More specifically, we look to answer the question of whether achieving MP is strictly harder, and we answer this question in the affirmative. Indeed, following simple example suggests that MP estimation can sometimes require a much higher label complexity than direct estimation.

Example 1. *Let* $\epsilon = \frac{1}{4}$ *and* $n \gg 1$. $\mathcal{X} = \{0, 1, \ldots, n\}$, *and* $x \mid x_A = 0 \sim \mathrm{Uniform}(\{0\})$, *and* $x \mid x_A = 1 \sim \mathrm{Uniform}(\{1, \ldots, n\})$. *Let* $\mathcal{H} = \left\{h : \mathcal{X} \to \{-1, +1\}, h(0) = -1\right\}$.

First, as $\epsilon = \frac{1}{4}$, *the iid sampling baseline makes* $O(1)$ *queries and ensures that it estimates* $\mu(h^*)$ *with error at most* ϵ *with probability* ≥ 0.9.

However, for manipulation-proof estimation, at least $\Omega(n)$ *labels are needed to ensure that the queried dataset* S *satisfies* $\mathrm{diam}_\mu(\mathcal{H}(h^*, S)) \leq \epsilon$. *Indeed, let* $h^* \equiv -1$. *For any unlabeled dataset* S *of size* $\leq n/2$, *by the definition of* $\mathcal{H}$, *there always exist* $h, h' \in \mathcal{H}(h^*, S)$, *such that for all* $x \in \{1, \ldots, n\} \setminus S$, $h(x) = -1$ *and* $h'(x) = +1$. *As a result,* $\mu(h) = \frac{0}{n} - \frac{0}{1} = 0$, *and* $\mu(h') = \frac{\left|\{1,\ldots,n\} \setminus S\right|}{n} - \frac{0}{1} \geq \frac{1}{2}$, *which implies that* $\mathrm{diam}_\mu(\mathcal{H}(h^*, S)) \geq \frac{1}{2} > \epsilon$. □

6.4.2 Randomized Algorithms for Direct Estimation

The separation result above suggests that different algorithms may be needed if we are *only* interested in efficient direct estimation. Motivated by our previous exploration, a first question to answer is whether randomization should be a key ingredient in algorithm design. That is, can a randomized algorithm achieve query complexity smaller than that of the optimal deterministic algorithm? Through the example below, we answer this question in the affirmative.

Example 2. *Same as the setting of Example 1; recall that iid sampling, a randomized algorithm, estimates $\mu(h^*)$ with error at most $\epsilon = \frac{1}{4}$ with probability ≥ 0.9; it has a query complexity of $O(1)$.*

In contrast, consider any deterministic algorithm $\mathcal{A}$ with label budget $N \leq \frac{n}{2}$; we consider its interaction history with classifier $h_0 \equiv -1$, which can be summarized by a sequence of unlabeled examples $S = \langle x_1, \ldots, x_N \rangle$. Now, consider an alternative classifier h_1 such that $h_1(x) = -1$ on $S \cup \{0\}$, but $h_1(x) = +1$ on $\{1, \ldots, n\} \setminus S$. By an inductive argument, it can be shown that the interaction history between $\mathcal{A}$ and h_1 is also S, which implies that when the underlying hypotheses $h^* = h_0$ and $h^* = h_1$, $\mathcal{A}$ must output the same estimate $\hat{\mu}$ (see Lemma 40 in Appendix 6.6 for a formal proof); however, $\mu(h_0) - \mu(h_1) \geq \frac{1}{2}$, implying that under at least one of the two hypotheses, we must have $\left|\hat{\mu} - \mu(h^*)\right| \geq \frac{1}{4} = \epsilon$.

In summary, in this setting, a randomized algorithm has a query complexity of $O(1)$, much smaller than $\Omega(n)$, the optimal query complexity of deterministic algorithms. □

6.4.3 Case Study: Non-homogeneous Linear Classifiers under Gaussian Populations

In this subsection, we identify a practically-motivated setting where we are able to comprehensively characterize the minimax (randomized) active fairness auditing query complexity up to logarithmic factors. Specifically, we present a positive result in the form of an algorithm that has a query complexity of $\tilde{O}\left(\min(d, \frac{1}{\epsilon^2})\right)$ as well as a matching lower bound that shows any (possibly randomized) algorithm must have a query complexity of $\Omega\left(\min(d, \frac{1}{\epsilon^2})\right)$.

Example 3. *Let $d \geq 2$ and $\mathcal{X} = \mathbb{R}^d$. $x \mid x_A = 0 \sim \mathrm{N}(m_0, \Sigma_0)$, whereas $x \mid x_A = 1 \sim \mathrm{N}(m_1, \Sigma_1)$. Let hypothesis class $\mathcal{H}_{lin} = \left\{h_{a,b}(x) := \mathrm{sign}(\langle a, x \rangle + b) : a \in \mathbb{R}^d, b \in \mathbb{R}\right\}$ be the class of non-homogenenous linear classifiers.*

Recall that i.i.d sampling has a label complexity of $O\left(\frac{1}{\epsilon^2}\right)$. On the other hand, through a membership query-based active learning algorithm (Algorithm 22 in Appendix 6.9.2), we can approximately estimate $\mu(h^*)$ (up to scaling) by doing d-binary searches, using active label queries. This approach incurs a total label complexity of $\tilde{O}(d)$. Choosing the better of these two algorithms gives an active fairness auditing strategy of label complexity $\tilde{O}\left(\min(d, \frac{1}{\epsilon^2})\right)$.

We only present the main idea of Algorithm 22 here, with its full analysis deferred to Appendix 6.9.2. Its core component is Algorithm 20 below, which label-efficiently estimates $\gamma(h^*) = \mathbb{P}_{x \sim \mathrm{N}(0, I_d)}(h^*(x) = +1)$, with black-box label queries to $h^*(x) = \mathrm{sign}(\langle a^*, x \rangle + b^*)$. Algorithm 20 is based on the following insights. First, observe that $\gamma(h^*) = \Phi\left(\frac{b^*}{\|a^*\|_2}\right) =: \Phi(sr)$, where Φ is the standard normal CDF, $s := \mathrm{sign}(b^*)$, and $r := \sqrt{\frac{1}{\sum_{i=1}^{d} m_i^{-2}}}$, for $m_i := -\frac{b^*}{a_i^*}$. On the one hand, s can be easily obtained by querying h^* on $\mathbf{0}$ (line 2). On the other hand, estimating r can be reduced to estimating each m_i. However, some m_i's can be unbounded, which makes their estimation challenging. To get around this challenge, we prove the following lemma, which shows that it suffices to accurately estimate those m_i's that are not unreasonably large (i.e. m_i's for $i \in S$, defined below):

Lemma 39. *Let $\alpha := \sqrt{2d \ln \frac{1}{\epsilon}}$ and $\beta := 2d^{\frac{5}{2}}(\ln \frac{1}{\epsilon})^{\frac{3}{4}}(\frac{1}{\epsilon})^{\frac{1}{2}}$. Suppose $r \leq \alpha$. If there is some $S \subset [d]$, such that:*

1. *for all $i \notin S, |m_i| \geq \beta$,*
2. *for all $i \in S$, $|\hat{m}_i - m_i| \leq \epsilon$;*

then, $\left| \sqrt{\frac{1}{\sum_{i \in S} \hat{m}_i^{-2}}} - r \right| \leq 2\epsilon$.

Algorithm 20 carefully utilizes this lemma to estimate r. First, it tests whether for all i, $h^*(\alpha e_i) = h^*(-\alpha e_i)$; if yes, for all i, $|m_i| \geq \alpha$, and $r \geq \sqrt{\ln \frac{1}{\epsilon}}$, and $\gamma(h^*)$ is ϵ-close to 0 or 1 depending on the value of s (line 4). Otherwise, it must be the case that $r \leq \alpha$. In this case, we go over each coordinate i, first testing whether $|m_i| \leq \beta$ (line 7); if no, we skip this coordinate (do not add it to S); otherwise, we include i in S and estimate m_i to precision ϵ using binary search (line 10). By the guarantees of Lemma 39, we have $|s\hat{r} - sr| \leq 2\epsilon$, which, by the $\frac{1}{\sqrt{2\pi}}$-Lipschitzness of Φ, implies that $\left|\hat{\gamma} - \gamma(h^*)\right| \leq \epsilon$. The total query complexity of Algorithm 20 is $1 + 2d + 2d + d \log_2 \frac{\beta}{\epsilon} = \tilde{O}(d)$.

Algorithm 20 A label efficient estimation algorithm for $\gamma(h^*)$ for non-homogeneous linear classifiers

```
Require: query access to $h^* \in \mathcal{H}_{\text{lin}}$, target error $\epsilon$.
Ensure: $\hat{\gamma}$ such that $\left|\hat{\gamma} - \gamma(h^*)\right| \leq \epsilon$.
 1: Let $\alpha = \sqrt{2d \ln \frac{1}{\epsilon}}$, $\beta = 2d^{\frac{5}{2}}(\ln \frac{1}{\epsilon})^{\frac{3}{4}}(\frac{1}{\epsilon})^{\frac{1}{2}}$.
 2: $s \leftarrow$ Query $h^*$ on $\mathbf{0}$
 3: Query $h^*$ on $\left\{\rho \alpha e_i : \rho \in \{\pm 1\}, i \in [d]\right\}$
 4: if for all $i \in [d]$, $h^*(\alpha e_i) = h^*(-\alpha e_i)$ then
        return 1 if $s = +1$, 0 if $s = -1$
                                        ▷ Otherwise, $r \leq \alpha = \sqrt{2d \ln \frac{1}{\epsilon}}$
 5: $S \leftarrow \emptyset$
 6: for $i = 1, \ldots, d$ do
 7:     Query $h^*$ on $\beta e_i$ and $-\beta e_i$
 8:     if $h^*(\beta e_i) \neq h^*(-\beta e_i)$ then
 9:         $S \leftarrow S \cup \{i\}$
                ▷ Use binary search to obtain $\hat{m}_i$, an estimate of $m_i = -\frac{b^*}{a_i^*}$ with precision $\epsilon$
10:         $\hat{m}_i \leftarrow$ Binary-Search$(i, \beta, \epsilon)$ (Algorithm 21)
11: $\hat{r} \leftarrow \sqrt{\frac{1}{\sum_{i \in S} \hat{m}_i^{-2}}}$                    ▷ $\hat{r}$ is an estimate of $r$
12: return $\Phi(s\hat{r})$
```

For the lower bound, we formulate a hypothesis testing problem, such that under hypotheses H_0 and H_1, the $\mu(h^*)$ values are approximately ϵ-separated. This is used to show that any active learning algorithm with label query budget $\leq \Omega\left(\min(d, \frac{1}{\epsilon^2})\right)$ cannot effectively distinguish H_0 and H_1. Our construction requires a delicate analysis on the KL divergence between the observation distributions under the two hypotheses, and we refer the readers to Theorem 32 for details. □

Algorithm 21 BINARY-SEARCH

```
Require: $i, \beta$ such that $h^*(\beta e_i) \neq h^*(-\beta e_i)$, precision $\epsilon$
Ensure: $m$, an $\epsilon$-accurate estimate of $m_i = -\frac{b}{a_i}$
$u \leftarrow \beta, l \leftarrow -\beta$
while $u - l \geq \epsilon$ do
    $m \leftarrow \frac{u+l}{2}$
    Query $h^*$ on $me_i$
    if $h^*(me_i) = h^*(le_i)$ then
        $l \leftarrow m$
    else
        $u \leftarrow m$
return $m$
```

6.4.4 General Distribution-Free Lower Bounds

Finally, in this subsection, we move beyond the Gaussian population setting and derive general query complexity lower bounds for randomized estimation algorithms that audit general hypothesis classes with finite VC dimension d. This result suggests that, when $d \gg \frac{1}{\epsilon^2}$, or equivalently $\epsilon \gg \frac{1}{\sqrt{d}}$, there exists some hard data distribution and target classifier in $\mathcal{H}$, such that active fairness auditing has a query complexity lower bound of $\Omega(\frac{1}{\epsilon^2})$. Put another way, iid sampling is near-optimal.

Theorem 29 (Lower bound for randomized auditing)**.** *Fix $\epsilon \in (0, \frac{1}{40}]$ and a hypothesis class $\mathcal{H}$ with VC dimension $d \geq 1600$. For any (possibly randomized) algorithm $\mathcal{A}$ with label budget $N \leq O(\min(d, \frac{1}{\epsilon^2}))$, there exists a distribution D_X over $\mathcal{X}$ and $h^* \in \mathcal{H}$, such that $\mathcal{A}$'s output $\hat{\mu}$ when interacting with h^*, satisfies:*

$$\mathbb{P}\left(\left|\hat{\mu} - \mu(h^*)\right| > \epsilon\right) > \frac{1}{8}$$

The proof of Theorem 29 can be found at Appendix 6.9.1. The lower bound construction follows from a similar setting as in Example 1, except that we now choose h^* in a randomized fashion.

6.5 Additional Related Works

Property Testing: Our notion of auditing that leverages knowledge of $\mathcal{H}$ is similar in theme to the topic of property testing [31, 45, 46, 47, 122, 243] which tests whether h^* is in $\mathcal{H}$, or h^* is far away from any classifier in $\mathcal{H}$, given query access to h^*. These works provide algorithms with testing query complexity of lower order than sample complexity for learning with respect to $\mathcal{H}$, for specific hypothesis classes such as monomials, DNFs, decision trees, linear classifiers, etc. Our problem can be reduced to property testing by testing whether h^* is in $\left\{h \in \mathcal{H} : \mu(h) \in [i\epsilon, (i+1)\epsilon]\right\}$ for all $i \in \left\{0, 1, \ldots, \lceil\frac{1}{\epsilon}\rceil\right\}$; however, to the best of our knowledge, no such result is known in the context of property testing.

Feature Minimization Audits: Rastegarpanah et al. [236] study another notion of auditing, focusing on assessing whether the model is trained inline with the GDPR's Data Minimization principle. Specifically, this work evaluates the necessity of each individual feature used in the ML model, and this is done by imputing each feature with constant values and checking the extent of variation in the predictions. One commonality with our work, and indeed across all auditing works, is the concern with minimizing the number queries needed to conduct the audit.

Herding for Sample-efficient Mean Estimation: Additionally, the estimation of DP may be viewed as estimating the difference of two means. Viewed in this light, herding [300] offers a way to use non-iid sampling to more efficiently estimate means. However, the key difference needed in herding is that h^*, whose output is $\{-1, 1\}$, may be well-approximated by $\langle w, \phi(x) \rangle$ for some mapping ϕ known apriori.

Comparison with Sabato et al. [245]: Lastly, Sabato et al. [245] also uses the term "auditing" in the context of active learning with outcome-dependent query costs; although the term "auditing" is shared, our problem settings are completely different: [245] focuses on active learning the model h^* as opposed to just estimating $\mu(h^*)$.

6.6 A General Lemma on Deterministic Query Learning

In this section, we present a general lemma inspired by Hanneke [132], which are used in our proofs for establishing lower bounds on deterministic active fairness auditing algorithms.

Lemma 40. *If an deterministic active auditing algorithm $\mathcal{A}$ with label budget N interacts with labeling oracle that uses classifier h_0, and generates the following interaction history: $\langle (x_1, h_0(x_1)), (x_2, h_0(x_2)), \ldots, (x_N, h_0(x_N)) \rangle$, and there exists a classifier h_1 such that $h_1(x) = h_0(x)$ for all $x \in \{x_1, \ldots, x_N\}$. Then $\mathcal{A}$, when interacting with h_1, generates the same interaction history, and outputs the same auditing estimate; formally, $S_{\mathcal{A},h_1} = S_{\mathcal{A},h_0}$ and $\hat{\mu}_{\mathcal{A},h_1} = \hat{\mu}_{\mathcal{A},h_0}$.*

Proof. Recall from Section 6.1.1 that deterministic active auditing algorithm $\mathcal{A}$ can be viewed as a sequence of $N + 1$ functions $f_1, f_2, \ldots, f_N, g$, where $\{f_i\}_{i=1}^N$ are the label query function used at each iteration, and g is the final estimator function. We show by induction that for steps $i = 0, 1, \ldots, N$, the interaction histories of $\mathcal{A}$ with h_0 and h_1 agree on their first i elements.

Base case. For step $i = 0$, both interaction histories are empty and agree trivially.

Inductive case. Suppose that the statement holds for step i, i.e. $\mathcal{A}$, when interacting with both h_0 and h_1, generates the same set of labeled examples

$$S_i = \langle (x_1, y_1), \ldots, (x_i, y_i) \rangle,$$

up to step i.

Now, at step $i + 1$, $\mathcal{A}$ applies the query function f_{i+1} and queries the same example $x_{i+1} = f_{i+1}(S_i)$. By assumption of this lemma, $h_1(x_{i+1}) = h_0(x_{i+1})$, which implies that the $(i + 1)$-st labeled example obtained when $\mathcal{A}$ interacts with h_1, $(x_{i+1}, h_1(x_{i+1}))$ is identical to $(x_{i+1}, h_1(x_{i+1}))$, the $(i + 1)$-st example when $\mathcal{A}$ interacts with h_0. Combined with the inductive hypotheses that the two histories agree on the first i examples, we have shown that $\mathcal{A}$, when interacting with h_0 and h_1, generates the same set of labeled examples

$$S_{i+1} = \langle (x_1, y_1), \ldots, (x_i, y_i), (x_{i+1}, y_{i+1}) \rangle$$

up to step $i + 1$.

This completes the induction.

As the interaction histories $\mathcal{A}$ with h_0 and h_1 are identical, the unlabeled data part of the history are identical, formally, $S_{\mathcal{A},h_1} = S_{\mathcal{A},h_0}$. In addition, as in both interactive processes, $\mathcal{A}$ applies deterministic function g to the same interaction history of length N to obtain estimate $\hat{\mu}$, we have $\hat{\mu}_{\mathcal{A},h_1} = \hat{\mu}_{\mathcal{A},h_0}$. □

6.7 Deferred Materials from Section 6.1

The following lemma formalizes the idea that PAC learning with $O(\epsilon)$ error is sufficient for fairness auditing, given that $\underline{p} = \min\big(\Pr_{D_X}(x_A = 0), \Pr_{D_X}(x_A = 1)\big)$ is $\Omega(1)$.

Lemma 41. *If h is such that $\mathbb{P}(h(x) \neq h^*(x)) \leq \alpha$, then $\big|\mu(h) - \mu(h^*)\big| \leq \frac{\alpha}{\underline{p}}$.*

Proof. First observe that

$$\begin{aligned} &\big|\Pr(h(x) = +1 \mid x_A = 0) - \Pr(h^*(x) = +1 \mid x_A = 0)\big| \\ \leq &\Pr(h(x) \neq h^*(x) \mid x_A = 0) \\ = &\frac{\Pr(h(x) \neq h^*(x), x_A = 0)}{\Pr(x_A = 0)} \\ \leq &\frac{\Pr(h(x) \neq h^*(x), x_A = 0)}{\underline{p}}, \end{aligned}$$

where the first inequality is by triangle inequality; the second inequality is by the definition of $\underline{p}$. Symmetrically, we have $\big|\Pr(h(x) = +1 \mid x_A = 1) - \Pr(h^*(x) = +1 \mid x_A = 1)\big| \leq \frac{\Pr(h(x) \neq h^*(x), x_A = 1)}{\underline{p}}$. Adding up the two inequalities, we have:

$$\begin{aligned} &\big|\mu(h) - \mu(h^*)\big| \\ \leq &\big|\Pr(h(x) = +1 \mid x_A = 0) - \Pr(h^*(x) = +1 \mid x_A = 0)\big| + \\ &\quad \big|\Pr(h(x) = +1 \mid x_A = 1) - \Pr(h^*(x) = +1 \mid x_A = 1)\big| \\ \leq &\frac{\Pr(h(x) \neq h^*(x), x_A = 0)}{\underline{p}} + \frac{\Pr(h(x) \neq h^*(x), x_A = 1)}{\underline{p}} \\ = &\frac{\Pr(h(x) \neq h^*(x))}{\underline{p}} \leq \frac{\alpha}{\underline{p}}. \end{aligned}$$

□

6.8 Deferred Materials from Section 6.3

6.8.1 Proof of Theorems 26 and 27

Proof of Theorem 26. Suppose Algorithm 17 (denoted as $\mathcal{A}$ throughout the proof) interacts with some target classifier $h^* \in \mathcal{H}$.

We will show the following claim: at any stage of $\mathcal{A}$, if the set of labeled examples L shown so far induces a version $V = \mathcal{H}[L]$, then $\mathcal{A}$ will subsequently query at most $\mathrm{Cost}(V)$ more labels before exiting the while loop.

Note that Theorem 26 follows from this claim by taking $L = \emptyset$ and $V = \mathcal{H}$: after $\mathrm{Cost}(\mathcal{H})$ label queries, it exits the while loop, which implies that, the queried unlabeled examples $S_{\mathcal{A},h^*}$ induces version space $V' = \mathcal{H}(h^*, S_{\mathcal{A},h^*})$ with

$$\max_{h \in V'} \mu(h) - \min_{h \in V'} \mu(h) = \mathrm{diam}_\mu(V') \leq 2\epsilon.$$

Also, note that $h^* \in V'$; this implies that $\mu(h^*) \in [\min_{h \in V'} \mu(h), \max_{h \in V'} \mu(h)]$. Combining these two observations, we have

$$\left|\hat{\mu} - \mu(h^*)\right| \leq \frac{1}{2}\left(\max_{h \in V'} \mu(h) - \min_{h \in V'} \mu(h)\right) \leq \epsilon.$$

We now come back to proving this claim by induction on $\mathrm{Cost}(V)$.

Base case. If $\mathrm{Cost}(V) = 0$, then $\mathcal{A}$ immediately exits the while loop without further label queries.

Inductive case. Suppose the claim holds for all V such that $\mathrm{Cost}(V) \leq n$. Now consider a version space V with $\mathrm{Cost}(V) = n + 1$. In this case, first recall that

$$\mathrm{Cost}(V) = 1 + \min_{x \in \mathcal{X}} \max_{y \in \{-1,+1\}} \mathrm{Cost}\left(V_x^y\right),$$

i.e. $\min_{x \in \mathcal{X}} \max_{y \in \{-1,+1\}} \mathrm{Cost}\left(V_x^y\right) = \mathrm{Cost}(V) - 1 = n$. Also, recall that by the definition of Algorithm 17, when facing version space V, the next query example x_0 chosen by $\mathcal{A}$ is a solution of the following minimax optimization problem:

$$x_0 = \operatorname*{argmin}_{x \in \mathcal{X}} \max_{y \in \{-1,+1\}} \mathrm{Cost}\left(V_x^y\right),$$

which implies that $\max_{y \in \{-1,+1\}} \mathrm{Cost}\left(V_x^y\right) = n$. Specifically, this implies that the version space at the next iteration, $V\left(h^*, \{x_0\}\right) = V_{x_0}^{h^*(x_0)}$, satisfies that $\mathrm{Cost}(V\left(h^*, \{x_0\}\right)) \leq n$. Combining with the inductive hypothesis, we have seen that after a total of $1 + \mathrm{Cost}(V\left(h^*, \{x_0\}\right)) \leq n + 1 = \mathrm{Cost}(V)$ number of label queries, $\mathcal{A}$ will exit the while loop.

This completes the inductive proof of the claim. □

Proof of Theorem 27. Fix a deterministic active fairness auditing algorithm $\mathcal{A}$. We will show the following claim: If $\mathcal{A}$ has already obtained an ordered sequence of labeled examples L, and has a remaining label budget $N \leq \mathrm{Cost}(\mathcal{H}[L]) - 1$, then there exists $h \in \mathcal{H}[L]$, such that, $\mathcal{A}$, when interacting with h as the target classifier:

1. obtains a sequence of labeled examples L in the first $|L|$ rounds;
2. has final version space $\mathcal{H}(h, S_{\mathcal{A},h})$ with μ-diameter $> 2\epsilon$.

The theorem follow from this claim by taking $L = \emptyset$. To see why, we let $h \in \mathcal{H}[\emptyset] = \mathcal{H}$ be the classifier described in the claim. First, note that there exists some other classifier $h' \neq h$ in the final version space $\mathcal{H}(h, S_{\mathcal{A},h})$, such that $\big|\mu(h') - \mu(h)\big| > 2\epsilon$. For such h', $h'(S_{\mathcal{A},h}) = h(S_{\mathcal{A},h})$. Therefore, by Lemma 40, $S_{\mathcal{A},h} = S_{\mathcal{A},h'}$ (which we denote by S subsequently), and h and h' have the exact same labeling on S, and $\hat{\mu}_{\mathcal{A},h} = \hat{\mu}_{\mathcal{A},h'}$. This implies that, for $\mathcal{A}$, at least one of the following must be true:

$$\big|\hat{\mu}_{\mathcal{A},h} - \mu(h)\big| > \epsilon \text{ or } \big|\hat{\mu}_{\mathcal{A},h'} - \mu(h')\big| > \epsilon,$$

showing that it does not guarantee an estimation error $\leq \epsilon$ under all target $h \in \mathcal{H}$.

We now turn to proving the above claim by induction on $\mathcal{A}$'s remaining label budget N. In the following, denote by $V = \mathcal{H}[L]$.

Base case. If $N = 0$ and $\mathrm{Cost}(V) \geq 1$, then $\mathcal{A}$ at this point has zero label budget, which means that it is not allowed to make more queries. In this case, $S_{\mathcal{A},h} = L$, and $\mathcal{H}(S_{\mathcal{A},h}, h) = V$. As $\mathrm{Cost}(V) \geq 1$, we know that

$$\max_{h_1,h_2 \in \mathcal{H}(h, S_{\mathcal{A},h})} \big|\mu(h_1) - \mu(h_2)\big| = \max_{h_1,h_2 \in V} \big|\mu(h_1) - \mu(h_2)\big| > 2\epsilon.$$

This completes the proof of the base case.

Inductive case. Suppose the claim holds for all $N \leq n$. Now, suppose in the learning process, $\mathcal{A}$ has a remaining label budget $N = n + 1$, and has obtained labeled examples L such that $V = \mathcal{H}[L]$ satisfies $\mathrm{Cost}(V) \geq n + 2$. Let x be the next example $\mathcal{A}$ queries. By the definition of Cost, there exists some $y \in \{-1, +1\}$, such that

$$\mathrm{Cost}\left(\mathcal{H}\left[L \cup \big\{(x, y)\big\}\right]\right) = \mathrm{Cost}\left(V_x^y\right) \geq \mathrm{Cost}(V) - 1 \geq n + 1,$$

and after making this query, the learner has a remaining label budget of $N - 1 = n$.

By inductive hypothesis, there exists some $h \in \mathcal{H}\left[L \cup \big\{(x, y)\big\}\right]$, such that when $\mathcal{A}$ interacts with h subsequently (with obtained labeled examples $L \cup \big\{(x, y)\big\}$ and label budget $< n$), the final unlabeled dataset $S_{\mathcal{A},h}$ satisfies

$$\mathrm{diam}_\mu\left(\mathcal{H}(h, S_{\mathcal{A},h})\right) = \max_{h_1,h_2 \in \mathcal{H}(h, S_{\mathcal{A},h})} \big|\mu(h_1) - \mu(h_2)\big| > 2\epsilon.$$

In addition, when interacting with h, $\mathcal{A}$ obtains the example sequence $\langle L, (x, y)\rangle$ in its first $|L| + 1$ rounds of interaction, which implies that it obtains the example sequence L in its first $|L|$ rounds of interaction with h. This completes the induction. □

6.8.2 Proof Sketch of Proposition 35

Proof sketch. Let S_1 and S_2 be $O\left(\frac{1}{\epsilon^2}\ln|\mathcal{H}|\right)$ i.i.d samples from $D_X \mid x_A = 1$ and $D_X \mid x_A = 0$, respectively. Define

$$\hat{\mu}(h, S_1, S_2) = \Pr_{x\sim S_1}(h(x) = +1) - \Pr_{x\sim S_2}(h(x) = +1).$$

Hoeffding's inequality and union bound guarantees that with probability at least $\frac{1}{2}$, $\forall h \in \mathcal{H}$, $|\hat{\mu}(h, S_1, S_2) - \mu(h)| \leq \epsilon$. Now consider the following deterministic algorithm $\mathcal{A}$:

- Let $n = O\left(\frac{1}{\epsilon^2}\ln|\mathcal{H}|\right)$;
- Find (the lexicographically smallest) S_1 and S_2 in $\mathcal{X}^n$, such that

$$\forall h \in \mathcal{H},\ \left|\hat{\mu}(h, S_1, S_2) - \mu(h)\right| \leq \epsilon. \tag{6.1}$$

  This optimization problem is feasible, because as we have seen, a random choice of S_1, S_2 makes Equation (6.1) happen with nonzero probability.
- Return $\hat{\mu}(h^*, S_1, S_2)$ with $2n$ label queries to examples in $S_1 \cup S_2$.

By its construction, $\mathcal{A}$ queries $2n = O\left(\frac{1}{\epsilon^2}\ln|\mathcal{H}|\right)$ labels and returns $\hat{\mu}$ that is ϵ-close to $\mu(h^*)$. □

6.8.3 Proof of Proposition 36

Before we prove Proposition 36, we first recall the well-known Bernstein's inequality:

Lemma 42 (Bernstein's inequality)**.** *Given a set of iid random variables $Z_1, \ldots, Z_n$ with mean μ and variance σ^2; in addition, $|Z_i| \leq b$ almost surely. Then, with probability $1-\delta$,*

$$\left|\frac{1}{n}\sum_{i=1}^{n} Z_i - \mu\right| \leq \sqrt{\frac{2\sigma^2 \ln\frac{2}{\delta}}{n}} + \frac{b\ln\frac{2}{\delta}}{3n}.$$

Proof of Proposition 36. We will analyze Algorithm 18, a derandomized version of the Phased CAL algorithm [142, Chapter 2]. To prove this proposition, using Theorem 27, it suffices to show that Algorithm 18 has a deterministic label complexity bound of $O\left(\theta(\epsilon)\cdot\ln|\mathcal{H}|\cdot\ln\frac{1}{\epsilon}\right)$.

We first show that for every n, the optimization problem in line 7 is always feasible. To see this, observe that if we draw $S_n = \{x_1, \ldots, x_{m_n}\}$ as sample of size m_n drawn iid from D_X, we have:

1. By Bernstein's inequality with $Z_i = I(x_i \in \mathrm{DIS}(V_n))$, with probability $1 - \frac{1}{4}$,

$$\begin{aligned}\Pr_{S_n}(x \in \mathrm{DIS}(V_n)) &\leq \Pr_{D_X}(x \in \mathrm{DIS}(V_n)) + \sqrt{\frac{2\Pr_{D_X}(x \in \mathrm{DIS}(V_n))\ln 8}{m_n}} + \frac{\ln 8}{3m_n} \\ &\leq 2\Pr_{D_X}(x \in \mathrm{DIS}(V_n)) + \frac{\ln 8}{m_n}.\end{aligned}$$

   where the second inequality uses Arithmetic Mean-Geometric Mean (AM-GM) inequality.

2. By Bernstein's inequality and union bound over $h, h' \in \mathcal{H}$, we have with probability $1 - \frac{1}{4}$,

$$\forall h, h' \in \mathcal{H}: \; \Pr_{D_X}(h(x) \neq h'(x)) \leq \Pr_{S_n}(h(x) \neq h'(x)) + \sqrt{\frac{4 \Pr_{D_X}(h(x) \neq h'(x)) \ln |\mathcal{H}|}{m_n}} + \frac{4 \ln |\mathcal{H}|}{3m_n}$$

in which,

$$\forall h, h' \in \mathcal{H}: \; \Pr_{S_n}(h(x) \neq h'(x)) = 0 \implies \Pr_{D_X}(h(x) \neq h'(x)) \leq \frac{16 \ln |\mathcal{H}|}{m_n}.$$

By union bound, with nonzero probability, the above two condition hold simultaneously, showing the feasibility of the optimization problem.

We then argue that for all n, $V_{n+1} \subseteq B(h^*, \frac{16 \ln |\mathcal{H}|}{m_n})$. This is because for all $h \in V_{n+1}$, it and h^* are both in V_n and therefore they agree on $S_n \setminus T_n$; on the other hand, h and h^* agree on T_n by the definition of of V_{n+1}. As a consequence, $\Pr_{S_n}(h(x) \neq h^*(x)) = 0$, which implies that $\Pr_{D_X}(h(x) \neq h^*(x)) \leq \frac{16 \ln |\mathcal{H}|}{m_n}$. As a consequence, for all $h \in V_{N+1}$, $\Pr(h(x) \neq h^*(x)) \leq \frac{16 \ln |\mathcal{H}|}{m_N} \leq \underline{p}\epsilon$, implying that $\left|\mu(h) - \mu(h^*)\right| \leq \epsilon$ (recall Lemma 41).

We now turn to upper bounding Algorithm 18's label complexity:

$$\begin{aligned}
\sum_{n=1}^N |T_n| =& \sum_{n=1}^N m_n \cdot (2 \Pr_{D_X}(x \in \mathrm{DIS}(V_n)) + \frac{\ln 8}{m_n}) \\
\leq& \sum_{n=1}^N m_n \cdot (\theta(\epsilon) \cdot \frac{16 \ln |\mathcal{H}|}{m_n} \cdot \frac{2}{\underline{p}} + \frac{\ln 8}{m_n}) \\
\leq& O\left(\theta(\epsilon) \cdot \ln |\mathcal{H}| \cdot \ln \frac{1}{\epsilon}\right),
\end{aligned}$$

where the inequality uses the observation that for every $n \in [N]$,

$$\Pr_{D_X}(x \in \mathrm{DIS}(V_n)) \leq \Pr_{D_X}\left(x \in \mathrm{DIS}(B(h^*, \frac{16 \ln |\mathcal{H}|}{m_n}))\right) \leq \theta(\frac{\underline{p}\epsilon}{2}) \cdot \frac{16 \ln |\mathcal{H}|}{m_n} \leq \theta(\epsilon) \cdot \frac{16 \ln |\mathcal{H}|}{m_n} \cdot \frac{2}{\underline{p}},$$

where the second inequality is from the definition of disagreement coefficient (recall Section 6.1.1), and the last inequality is from a basic property of disagreement coefficient [135, Corollary 7.2]. □

6.8.4 Proof of Proposition 37

We first prove the following theorem that gives a decision tree-based characterization of the $\mathrm{Cost}(\cdot)$ function. Connections between active learning and optimal decision trees have been observed in prior works [e.g. 31, 173].

Definition 32. *An example-based decision tree $\mathcal{T}$ for (instance domain, hypothesis set) pair $(\mathcal{X}, V)$ is such that:*

1. *$\mathcal{T}$'s internal nodes are examples in $\mathcal{X}$; every internal node has two branches, with the left branch labeled as $+1$ and the right labeled as -1.*

2. *Every leaf l of $\mathcal{T}$ corresponds to a set of classifiers $V_l \subset V$, such that all $h \in V_l$ agree with the examples that appear in the root-to-leaf path to l. Formally, suppose the path from the root to leaf l is an alternating sequence of examples and labels $\langle x_1, y_1, \ldots, x_n, y_n \rangle$, then for every $i \in [n]$, $h(x_i) = y_i$.*

Definition 33. *Fix D_X. An example-based decision tree $\mathcal{T}$ is said to (μ, ϵ)-separate a hypothesis set V, if for every leaf l of $\mathcal{T}$, V_l satisfies $\mathrm{diam}_\mu(V_l) \leq 2\epsilon$.*

Theorem 30. *Given a version space V, $\mathrm{Cost}(V)$ is the minimum depth of all decision trees that (μ, ϵ)-separates V.*

Proof. We prove the theorem by induction on $\mathrm{Cost}(V)$.

Base case. If $\mathrm{Cost}(V) = 0$, then $\mathrm{diam}_\mu(V) \leq 2\epsilon$. Then there exists a trivial decision tree (with leaf only) of depth 0 that (μ, ϵ)-separates V, which is also the smallest depth possible.

Inductive case. Suppose the statement holds for any V such that $\mathrm{Cost}(V) = n$. Now consider V such that $\mathrm{Cost}(V) = n + 1$.

1. We first show that there exists a decision tree of depth $n + 1$ that (μ, ϵ)-separates V. Indeed, pick $x = \mathrm{argmin}_{x \in \mathcal{X}} \max_y \mathrm{Cost}(V_x^y)$.
   With this choice of x, we have both $\mathrm{Cost}(V_x^{-1})$ and $\mathrm{Cost}(V_x^{+1})$ are equal to n. Therefore, by inductive hypothesis for V_x^{-1} and V_x^{+1}, we can construct decision trees $\mathcal{T}^-$ and $\mathcal{T}^+$ of depths n that (μ, ϵ)-separate the two hypothesis classes respectively. Now define $\mathcal{T}$ to be such that it has root node x, and has left subtree $\mathcal{T}^+$ and right subtree $\mathcal{T}^-$, we see that $\mathcal{T}$ has depth $n + 1$ and (μ, ϵ)-separates V.
2. We next show that any decision tree of depth n does not (μ, ϵ)-separate V. Indeed, assume for the sake of contradiction that such tree $\mathcal{T}$ exists. Then consider the example x at the root of the tree; by the definition of Cost, one of $\mathrm{Cost}(V_x^{-1})$ and $\mathrm{Cost}(V_x^{+1})$ must be $\geq n$. Without loss of generality, assume that $V' = V_x^{+1}$ is such that $\mathrm{Cost}(V') \geq n$. Therefore, there must exists some subset $V'' \subset V'$ such that $\mathrm{Cost}(V'') = n$. Applying the inductive hypothesis on V'', no decision tree of depth $n - 1$ can (μ, ϵ)-separate V''. This contradicts with the observation that the left subtree of $\mathcal{T}$, which is of depth $n - 1$, (μ, ϵ)-separates V'. □

We now restate a more precise version of Proposition 37. First we define the computational task of computing a $0.3 \ln(|\mathcal{H}|)$-approximation of $\mathrm{Cost}(\mathcal{H})$ by the following problem:

Problem Minimax-Cost (MC):
Input: instance space $\mathcal{X}$, hypothesis class $\mathcal{H}$, data distribution D_X, precision parameter ϵ.
Output: a number L such that $\mathrm{Cost}(\mathcal{H}) \leq L \leq 0.3 \ln(|\mathcal{H}|)\mathrm{Cost}(\mathcal{H})$.

Proposition 38 (Proposition 37 restated)**.** *If there is an algorithm that solves Minimax-Cost in $\mathrm{poly}(|\mathcal{H}|, |\mathcal{X}|, 1/\epsilon)$ time, then $\mathrm{NP} \subseteq \mathrm{TIME}(n^{O(\log\log n)})$.*

Proof of Proposition 38. Our proof takes after [173]'s reduction from set cover (SC) to Decision Tree Problem (DTP). Here, we reduce from SC to the Minimax-Cost problem (MC), i.e. computing

$\mathrm{Cost}(\mathcal{H})$ for a given hypothesis class $\mathcal{H}$, taking into account the unique structure of active fairness auditing. Specifically, the following gap version of SC's decision problem has been shown to be computationally hard[4]:

Problem Gap-Set-Cover (Gap-SC):
Input: a universe $U = \{u_1, ..., u_n\}$ of size n with $n \geq 10$, and a family of subsets $\mathcal{C} = \{C_1, ..., C_m\}$, and an integer k, such that either of the following happens:

- Case 1: $\mathrm{OPT}_{\mathsf{SC}} \leq k$,
- Case 2: $\mathrm{OPT}_{\mathsf{SC}} \geq 0.99k \ln n$,

where $\mathrm{OPT}_{\mathsf{SC}}$ denotes the minimum set cover size of $(U, \mathcal{C})$.
Output: 1 or 2, which case the instance is in.

Specifically, it is well-known that obtaining a polynomial time algorithm for the above decision problem[5] on minimum set cover would imply that $\mathrm{NP} \subseteq \mathrm{TIME}(n^{O(\log \log n)})$ [107], which is believed to be false.

To start, recall that an instance of Gap-SC problem $I_{\mathsf{SC}} = (U, \mathcal{C}, k)$; an instance of the MC problem $I_{\mathsf{MC}} = (\mathcal{H}, \mathcal{X}, D_X, \epsilon)$.

With this, we define a coarse reduction β that constructs a MC-instance from a Gap-SC instance with universe $U = \{u_1, ..., u_n\}$ and sets $\mathcal{C} = \{C_1, ..., C_m\}$, which will be refined shortly:

1. Let $\mathcal{H} = \{h_0, h_1, \ldots, h_n\}$, where $h_0(x) \equiv -1$ always, and for all $j \in [n]$, h_j corresponds to u_j (the definitions of h_j's will be given shortly).
2. Create example x_0 such that for all $h \in \mathcal{H}$, $h(x_0) = -1$.
3. For every $i \in [m]$, create basis example x_i to correspond to C_i such that for every $j \in [n]$, $h_j(x_i) = 1$ iff $u_j \in C_i$.
4. For each set C_i, create $|C_i| - 1$ auxiliary x's as follows: Given set C_i with $|C_i| = s_i$ that corresponds to $\{h_{i1}, .., h_{is_i}\}$, create a balanced binary tree $\mathcal{T}_i$ with each leaf corresponding to a h_{ij}. Create an auxiliary example associated with each internal node in $\mathcal{T}_i$ as follows: for each internal node in the tree, define the corresponding auxiliary sample x such that its label is $+1$ under all the classifiers in the leaves of the subtree rooted at its left child, and its label is -1 under all remaining classifiers in $\mathcal{H}$. The total number of auxiliary x's is $\leq m \cdot (n-1)$.
5. Define $\mathcal{X}$ as the union of the example sets constructed in the above three items, which has at most $N \leq mn+1$ examples. Define D_X to be such that: $x \mid x_A = 0 \sim \mathrm{Uniform}(\mathcal{X} \setminus \{x_0\})$, and $x \mid x_A = 1 \sim \mathrm{Uniform}(\{x_0\})$, and set $\epsilon = 1/(2N)$. With this setting of ϵ, for every $h \in \mathcal{H}$ such that $h \neq h_0$, $\left|\mu(h) - \mu(h_0)\right| = \left|\Pr(h(x) = +1 \mid x_A = 0) - \Pr(h_0(x) = +1 \mid x_A = 0)\right| \geq \frac{1}{N-1} > 2\epsilon$.

Recall that $\mathrm{OPT}_{\mathsf{SC}}$ is defined as the size of an optimal solution for SC instance $(U, \mathcal{C})$; we let $\mathrm{OPT}_{\mathsf{MC}}$ denote the height of the tree corresponding to the optimal query strategy for the MC

[4] The definition of Gap-SC requires that $n \geq 10$, which is without loss of generality: all Gap-SC instances with $n < 10$ are solvable in constant time.

[5] The constant 0.99 can be changed to any constant < 1 [107].

instance I_{MC} obtained through reduction β. We have the following result:

Lemma 43. $\mathrm{OPT}_{\mathsf{SC}} \leq \mathrm{OPT}_{\mathsf{MC}} \leq \mathrm{OPT}_{\mathsf{SC}} + \max_{C \in \mathcal{C}} \log |C|$.

Proof. Let $k = \mathrm{OPT}_{\mathsf{SC}}$. We show the two inequalities respectively.

1. By Theorem 30, it suffices to show that any example-based decision tree $\mathcal{T}$ that (μ, ϵ)-separates $\mathcal{H}$ must have depth at least k. To see this, first note that by item 5 in the reduction β and the definition of (μ, ϵ)-separation, the leaf in $\mathcal{T}$ that contains h_0 must not contain other hypotheses in $\mathcal{H}$. In addition, as $h_0 \equiv -1$, h_0 must lie in the rightmost leaf of $\mathcal{T}$.
   Now to prove the statement, we know that the examples along the rightmost path of $\mathcal{T}$ corresponds to a collection of sets that form a set cover of $\mathcal{C}$. It suffices to show that this set cover has size no greater than the set cover of I_{SC}. This is because the examples along the rightmost path are either x_i's, which correspond to some set in $\mathcal{C}$, or auxiliary examples which correspond to some subset of a set in $\mathcal{C}$. A set cover instance with U and $\mathcal{C}'$ where $\mathcal{C}'$ comprises of sets from $\mathcal{C}$ and subsets of sets from $\mathcal{C}$ will not have a smaller set cover.
   Therefore, the length of the path from the root to the rightmost leaf is at least k, the size of the smallest set cover of the original SC instance I_{SC}.
2. Let an optimal solution for I_{SC} be $G = \{i_1, ..., i_k\}$. Below, we construct an example-based decision tree $\mathcal{T}$ of depth $k + \max_{C \in \mathcal{C}} \log |C|$ that (μ, ϵ)-separates $\mathcal{H}$:
   Let the rightmost path of $\mathcal{T}$ contain nodes corresponding to $x_{i_1}, ..., x_{i_k}$ (the order of these are not important). At level $l = 1, ..., k$, the left subtree of x_{i_l} is defined to be $\mathcal{T}_{i_l}$ as defined in step 4 of reduction β. Note that this may result in $\mathcal{T}$ with potentially empty leaves, in that for some h covered by multiple x_{i_l}'s, it only appears in x_{i_o} where $o = \min \left\{ l : h(x_{i_l}) = +1 \right\}$.
   We will prove that by the above construction, $\mathcal{T}$ (μ, ϵ)-separates $\mathcal{H}$, as every leaf corresponds to a version space V that is a singleton set (and thus has $\mathrm{diam}_\mu(V) = 0 \leq 2\epsilon$):
   (a) For all but the rightmost leaf, this holds by the construction of $\mathcal{T}_{i_l}$'s.
   (b) For the rightmost leaf, we will show that only h_0 is in the version space. Since G is a set cover, we have that $\cup_{l=1}^{k} C_{i_l} = U$. Therefore, $\forall j \in [n]$, $\exists l \in [k]$ such that $u_j \in C_{i_l} \Leftrightarrow h_j(x_{i_l}) = 1$ by construction. This implies that the all zero labeling of $x_{i_1}, ..., x_{i_k}$ can only correspond to h_0. Therefore, the version space at the rightmost leaf V satisfies $|V| = \{h_0\}$.

   Recall from Theorem 30 that the depth of $\mathcal{T}$ upper bounds $\mathrm{OPT}_{\mathsf{MC}}$. $\mathcal{T}$'s maximum root to leaf path is of length at most $k + \max_{C \in \mathcal{C}} \log |C|$. □

Built from β, we now construct an improved gap preserving reduction β', defined as follows. Given any Gap-SC instance $I_{\mathsf{SC}} = (U, \mathcal{C}, k)$ with universe $U = \{u_1, ..., u_n\}$ and sets $\mathcal{C} = \{C_1, ..., C_m\}$:

1. Take constant $z = \log n$. Construct a Gap-SC instance $I_{\mathsf{SC},z} = (U^z, \mathcal{C}^z, kz)$, containing z copies of the original set covering instance: $U^z = \{u_1^1, \ldots, u_n^1, \ldots, u_1^z, \ldots, u_n^z\}$, $\mathcal{C}^z = \{C_1, \ldots, C_{zm}\}$, where $C_{(p-1)m+i} = \{u_{i1}^p, \ldots, u_{is_i}^p\}$ for $p \in [z]$, $i \in [m]$. Note that $\mathrm{OPT}_{\mathsf{SC},z} = k\mathrm{OPT}_{\mathsf{SC}}$.
2. Apply reduction β to obtain $I_{\mathsf{MC},z}$ from $I_{\mathsf{SC},z}$.

Now, we will argue that β' is a gap-preserving reduction:

1. Suppose the original Gap-SC instance $I_{\mathsf{SC}} = (U, \mathcal{C}, k)$ is in case 1, i.e., $\mathrm{OPT}_{\mathsf{SC}} \le k$. Then, $\mathrm{OPT}_{\mathsf{SC},z} \le kz$. By Lemma 43, $\mathrm{OPT}_{\mathsf{MC},z} \le kz + \max_{C \in \mathcal{C}^z} \log|C| \le kz + \log n \le z(k+1) \le 2zk$.
2. Suppose the original Gap-SC instance $I_{\mathsf{SC}} = (U, \mathcal{C}, k)$ is in case 2, i.e., $\mathrm{OPT}_{\mathsf{SC}} \ge 0.99k \ln n$. Then, $\mathrm{OPT}_{\mathsf{SC},z} \ge 0.99zk \ln n$, which by Lemma 43, yields that $\mathrm{OPT}_{\mathsf{MC},z} \ge 0.99zk \ln n$.

Now suppose that there exists an algorithm $\mathcal{A}$ that solves the MC problem in $\mathrm{poly}(|\mathcal{H}|, |\mathcal{X}|, \frac{1}{\epsilon})$ time. We propose the following algorithm $\mathcal{A}'$ that solves the Gap-SC problem in polynomial time, which, as mentioned above, implies that $\mathrm{NP} \subseteq \mathrm{TIME}(n^{O(\log\log n)})$:

Input: $I_{\mathsf{SC}} = (U, \mathcal{C}, k)$.

- Apply β' on I_{SC} to obtain an instance of MC, $I_{\mathsf{MC},z}$
- Let $L \leftarrow \mathcal{A}(I_{\mathsf{MC},z})$. Output 1 if $L \le 0.7zk \ln n$, and 2 otherwise.

Correctness. As seen above, if I_{SC} is in case 1, then $\mathrm{OPT}_{\mathsf{MC},z} \le 2zk$. For $n \ge 10$, by the guarantee of $\mathcal{A}$, $L \le 0.3 \ln|\mathcal{H}| \cdot \mathrm{OPT}_{\mathsf{MC},z} \le 0.6 \ln(n \log n) \cdot zk \le 0.7zk \ln n$, and $\mathcal{A}'$ outputs 1. Otherwise, I_{SC} is in case 2, then $\mathrm{OPT}_{\mathsf{MC},z} \ge 0.99zk \ln n$, and by the guarantee of $\mathcal{A}$, $L \ge 0.99zk \ln n > 0.7zk \ln n$, and $\mathcal{A}'$ outputs 2.

Time complexity. In $I_{\mathsf{MC},z}$, $|\mathcal{X}| \le (mz \cdot nz + 1) = O(mn \log^2 n)$, $|\mathcal{H}| = nz = n \log n$, and $\epsilon = \frac{1}{2N} = \frac{1}{2(mz \cdot nz + 1)} = \Omega(\frac{1}{mn \log^2 n})$. As $\mathcal{A}$ runs in time $O(\mathrm{poly}(|\mathcal{X}|, |\mathcal{H}|, \frac{1}{\epsilon}))$, $\mathcal{A}'$ runs in time $O(\mathrm{poly}(m, n))$. □

6.8.5 Deferred Materials for Section 6.3.2

6.8.5.1 (μ, ϵ)-specifying set, (μ, ϵ)-teaching dimension and their properties

The following definitions are inspired by the teaching and exact active learning literature [132, 138].

Definition 34 ((μ, ϵ)-specifying set)**.** *Fix hypothesis class $\mathcal{H}$ and any function $h : \mathcal{X} \to \mathcal{Y}$,*[6] *a set of unlabeled examples S is said to be a (μ, ϵ)-*specifying set *for h and $\mathcal{H}$, if $\forall h_1, h_2 \in \mathcal{H}(h, S) \centerdot |\mu(h_1) - \mu(h_2)| \le 2\epsilon$.*

Definition 35 ((μ, ϵ)-extended teaching dimension)**.** *Fix hypothesis class $\mathcal{H}$ and any function $h : \mathcal{X} \to \mathcal{Y}$, define $t(h, \mathcal{H}, \mu, \epsilon)$ as the size of the minimum (μ, ϵ)-specifying set for h and $\mathcal{H}$, i.e. it is the optimal solution of the following optimization problem (OP-h):*

$$\min |S|, s.t. \forall h_1, h_2 \in \mathcal{H}(h, S) \centerdot |\mu(h_1) - \mu(h_2)| \le 2\epsilon$$

Definition 36. *We define the μ-extended teaching dimension* $\mathrm{XTD}(\mathcal{H}, \mu, \epsilon) := \max_{h:\mathcal{X} \to \mathcal{Y}} t(h, \mathcal{H}, \mu, \epsilon)$.

[6] Note that h is allowed to be outside $\mathcal{H}$.

The improper teaching dimension is related to $\mathrm{Cost}(\mathcal{H})$ in that:

Lemma 44.

$$\mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \leq \mathrm{Cost}(\mathcal{H}).$$

Proof. Let $h_0 = \mathrm{argmax}_{h:\mathcal{X}\to\mathcal{Y}}\, t(h, \mathcal{H}, \mu, \epsilon)$. Let k denote $t(h_0, \mathcal{H}, \mu, \epsilon) - 1$. It suffices to show that $\mathrm{Cost}(\mathcal{H}) \geq k$. To see this, first note that

$$\begin{aligned}
\mathrm{Cost}(\mathcal{H}) &= 1 + \min_x \max_y \mathrm{Cost}(\mathcal{H}[(x, y)]) \\
&\geq 1 + \min_{x_1 \in \mathcal{X}} \mathrm{Cost}(\mathcal{H}[(x, h_0(x))]) \\
&\geq 2 + \min_{x_1 \in \mathcal{X}} \min_{x_2 \in \mathcal{X}} \mathrm{Cost}(\mathcal{H}[\{(x_1, h_0(x_1)), (x_2, h_0(x_2))\}])
\end{aligned}$$

We can repeatedly unroll the above expression as long as $\mathrm{diam}_\mu(\mathcal{H}[\{(x_1, h_0(x_1)), \ldots, (x_i, h_0(x_i))\}])$ is at least $> 2\epsilon$. After unrolling $k-1$ times where $U_{k-1} = \langle x_1, \ldots, x_{k-1} \rangle$, we have

$$\mathrm{Cost}(\mathcal{H}) \geq k - 1 + \min_{U_{k-1}} \mathrm{Cost}(\mathcal{H}(h_0, U_{k-1})).$$

By the definition of $t(h, \mathcal{H}, \mu, \epsilon)$, for any U with $U \leq k-1$, there exists $h', h'' \in \mathcal{H}(h_0, U)$ such that $|\mu(h') - \mu(h'')| > \epsilon \Rightarrow \mathrm{diam}_\mu(\mathcal{H}(h_0, U)) > \epsilon$. Thus, for any unlabeled dataset U_{k-1} of size $k-1$, $\mathrm{Cost}(\mathcal{H}(h_0, U_{k-1})) \geq 1$. Therefore, $\mathrm{Cost}(\mathcal{H}) \geq k$. □

6.8.5.2 Proof of Theorem 28

Proof. We prove the theorem as follows:

Correctness. Observe that right before Algorithm 19 returns, it must execute lines 11 and 20. Since the condition on line 20 is also satisfied, the dataset T must be such that $\hat{h}(T) = h^*(T)$. Combined with the definitions of optimization problems (8) and (9), this implies that, the h_1 and h_2 used in line 11 right before return satisfy that

$$\mu(h_1) = \min_{h \in \mathcal{H}(h^*, T)} \mu(h), \quad \mu(h_2) = \max_{h \in \mathcal{H}(h^*, T)} \mu(h).$$

Therefore, $\mu(h^*) \in [\min_{h\in\mathcal{H}(h^*,T)} \mu(h), \max_{h\in\mathcal{H}(h^*,T)} \mu(h)] = [\mu(h_1), \mu(h_2)]$. Furthermore, by line 11, $\mu(h_1) - \mu(h_2) \leq 2\epsilon$. Hence, $\hat{\mu}$, the output of Algorithm 19, satisfies that,

$$\left|\hat{\mu} - \mu(h^*)\right| = \left|\frac{1}{2}\left(\mu(h_1) + \mu(h_2)\right) - \mu(h^*)\right| \leq \epsilon.$$

Label complexity. We now bound the label complexity of the algorithm, specifically, in terms of $\mathrm{XTD}(\mathcal{H}, \mu, \epsilon)$.

First, at the end of the t-th iteration of the outer loop, the newly collected dataset T_t must be such that $\exists x \in T_t$ and $\hat{h}(x) \neq h^*(x)$. As $\mathcal{O}$ has a mistake bound of M, the total number of outer loop iterations, denoted by N, must be most M. In addition, by Lemma 45 given below, with probability $1 - \delta/M$, $|T_t| \le O\left(\mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \cdot \log \frac{|\mathcal{H}|M}{\delta} \log |\mathcal{X}|\right)$. Therefore, by a union bound, with probability $1 - \delta$, the total number of label queries made by Algorithm 19 is at most

$$\sum_{t=1}^{N} |T_t| \le O\left(M \cdot \mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \cdot \log \frac{|\mathcal{H}|M}{\delta} \log |\mathcal{X}|\right).$$

Lemma 45. *For every outer iteration of Algorithm 19, with probability* $\ge 1 - \frac{\delta}{M}$, *T, the dataset at the end of this iteration, satisfies*$|T| \le O\left(\mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \cdot \log \frac{|\mathcal{H}|M}{\delta} \log |\mathcal{X}|\right)$.

Proof. The inner loop is similar to the "black-box teaching" algorithm of [85] except that we are teaching $\mu(\hat{h})$ as opposed to $\hat{h}$ itself. Although [85]'s algorithm was originally designed for exact (interactive) teaching, it implicitly gives an oracle-efficient algorithm for approximately computing the minimum set cover; we will use this insight throughout the proof. As the analysis of [85] is only on the *expected* number of teaching examples, we use a different filtration to obtain a high probability bound over the number of teaching examples.

First we setup some useful notations for the proof. let $\mathcal{X} = \{x_1, \ldots, x_m\}$. Recall that $\lambda = \ln \frac{|\mathcal{H}|^2 M}{\delta}$. Let $W_i(x)$ denote the weight of point $x \in \mathcal{X}$ (denoted by $w(x)$ in the algorithm) at the end of round i of the inner loop and let τ_{x_j} be the exponentially-distributed threshold associated with x_j. Define random variable $U_{i,j} = \mathbb{1}\{\tau_{x_j} > W_i(x_j)\}$. Let M_i denotes the number of teaching examples selected in the ith round of doubling; it can be seen that $M_i = \sum_{j \in [m]} U_{i,j}$. Also define $(i, j) \preceq (i', j')$ iff (i, j) precedes (i', j') lexicographically.

Define two filtrations:

1. Let $\mathcal{F}_{i,j}$ be the sigma-field of all indicator events $\{U_{i',j'} : (i', j') \preceq (i, j)\}$. As a convention, $\mathcal{F}_{i,0} := \mathcal{F}_{i-1,m}$.
2. Let $\mathcal{F}_i$ be the sigma-field of all indicator events $\{U_{i',j'} : j' \in [m], 1 \le i' \le i\}$; this is the filtration used by [85]. It can be easily seen that $\mathcal{F}_i = \mathcal{F}_{i,m}$.

Define $Y_{i,j} = \sum_{(i',j') \preceq (i,j)} Z_{i',j'}$, where $Z_{i,j} = U_{i,j} - \mathbb{E}\left[U_{i,j} \mid \mathcal{F}_{i,j-1}\right] \in [-1, +1]$. Then $Y_{i,j}$ is a martingale as $\mathbb{E}[Y_{i,j}|\mathcal{F}_{i,j-1}] = \mathbb{E}[Z_{i,j}|\mathcal{F}_{i,j-1}] + \mathbb{E}[Y_{i,j-1}|\mathcal{F}_{i,j-1}] = Y_{i,j-1}$.

Let N be the total number of rounds, which by item 1 of Lemma 47, is $O(\mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \ln |\mathcal{X}|)$ (Lemma 4 of [85]) with probability 1. We may then apply Freedman's inequality (Lemma 46): since $Y_{i,j} - Y_{i,j-1} = Z_{i,j} \le 1$ almost surely, for any s and any $\sigma^2 > 0$,

$$\Pr\left(\exists n, m, Y_{nm} \ge s, \sum_{(i,j) \preceq (n,m)} \mathbb{E}[Z_{ij}^2 | \mathcal{F}_{i(j-1)}] \le \sigma^2\right) \le \exp\left(-\frac{s^2}{2(\sigma^2 + s/3)}\right) \quad (6.2)$$

Next, we let $\sigma^2 = \lambda(1 + \mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \ln(2|\mathcal{X}|))$; we have for any n, m:

$$\begin{aligned}
&\sum_{(i,j)\preceq(n,m)} \mathbb{E}[Z_{ij}^2|\mathcal{F}_{i(j-1)}] \\
&= \sum_{(i,j)\preceq(n,m)} \mathbb{E}[U_{ij}^2|\mathcal{F}_{i(j-1)}] - \mathbb{E}[U_{ij}|\mathcal{F}_{i(j-1)}]^2 \\
&\le \sum_{(i,j)\preceq(n,m)} \mathbb{E}[U_{ij}^2|\mathcal{F}_{i(j-1)}] \\
&= \sum_{(i,j)\preceq(n,m)} \mathbb{E}[U_{ij}|\mathcal{F}_{i(j-1)}] \\
&= \sum_{i=1}^{n} \mathbb{E}_{\mathcal{F}_{i-1}}\left[M_i\right] \\
&\le \lambda \sum_{x\in\mathcal{X}} W_n(x) && \text{(Lemma 48)} \\
&\le \lambda(1 + \mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \ln(2|\mathcal{X}|)) = \sigma^2. && \text{(Lemma 47)}
\end{aligned}$$

Meanwhile, we choose $s = \frac{1}{6}\log(\frac{1}{\delta}) + \sqrt{2\sigma^2 \log\frac{1}{\delta} + \frac{1}{6}\log(\frac{1}{\delta})} = O\left(\sqrt{\ln\frac{1}{\delta}}\sigma + \ln\frac{1}{\delta}\right)$, which ensures that the right hand side of Eq. (6.2) is at most δ.

Thus, by Equation (6.2), we have with probability $1 - \delta$, for all n, m,

$$Y_{nm} = \sum_{(i',j')\preceq(n,m)} U_{i'j'} - \sum_{i=1}^{n} \mathbb{E}_{\mathcal{F}_{i-1}}\left[M_i\right] \le O\left(\sqrt{\ln\frac{1}{\delta}}\sigma + \ln\frac{1}{\delta}\right).$$

Also, using Lemma 48 and 47, with probability 1, $\sum_{i=1}^{N} \mathbb{E}_{\mathcal{F}_{i-1}}\left[M_i\right] \le \lambda(1+\mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \ln(2|\mathcal{X}|))$.

Therefore, for Y_{Nm} in particular,

$$\begin{aligned}
Y_{Nm} &\le O\left(\lambda(1 + \mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \ln(2|\mathcal{X}|)) + \sqrt{\lambda(1 + \mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \ln(2|\mathcal{X}|)) \ln(1/\delta)} + \ln(1/\delta)\right) \\
&= O\left(\lambda(1 + \mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \ln(2|\mathcal{X}|)) + \ln\frac{1}{\delta}\right) \\
&= O\left(\mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \ln(|\mathcal{X}|) \ln((|\mathcal{H}|M)/\delta)\right). \qquad \square
\end{aligned}$$

Lemma 46 (Freedman's Inequality)**.** *Let martingale $\{Y_k\}_{k=0}^{\infty}$ with difference sequence $\{X_k\}_{k=0}^{\infty}$ be such that $X_k \le R$ a.s for all k and $Y_0 = 0$. Let $W_k = \sum_{j=1}^{k} \mathbb{E}_{j-1}[X_j^2]$. Then, for all $t \ge 0$ and $\sigma^2 > 0$:*

$$\Pr(\exists k \ge 0 : Y_k \ge t \wedge W_k \le \sigma^2) \le \exp\left(-\frac{t^2/2}{\sigma^2 + Rt/3}\right).$$

Lemma 47. *For any outer iteration of Algorithm 19:*

1. *The number of inner loop iterations is at most $\mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \cdot \log(2|\mathcal{X}|)$.*
2. *At any point in the inner loop, we have that, $\sum_{x\in\mathcal{X}} w(x) \le 1 + \mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \cdot \log(2|\mathcal{X}|)$.*

Proof. The proof is very similar to Dasgupta et al. [85, Lemma 4] with some differences; for completeness, we include a proof here.

We first prove the second item. First, note that at any point of the algorithm, for all x, $w(x) \leq 2$. Let $S^*(\hat{h})$ be the optimal solution of optimization problem (OP-$\hat{h}$) - we have $|S^*(\hat{h})| = t(\hat{h}, \mathcal{H}, \mu, \epsilon) \leq \mathrm{XTD}(\mathcal{H}, \mu, \epsilon)$. Note that every time when line 16 is called, by the feasibility of $S^*(\hat{h})$ with respect to (OP-$\hat{h}$), $\Delta(h_1, h_2) \cap S^*(\hat{h}) \neq \emptyset$, therefore, the weight of some element $x \in S^*(\hat{h})$ gets doubled. This implies that the total number of times line 16 is executed is at most $|S^*(\hat{h})| \cdot \log(2|\mathcal{X}|)$. Otherwise, if the number of time line 16 is executed is $\geq |S^*(\hat{h})| \cdot \log(2|\mathcal{X}|) + 1$, by the pigeonhole principle, there must exist some element $x \in S^*(\hat{h})$ whose weight exceeds 1, which is a contradiction.

Finally, note that each weight doubling only increases the total weight by ≤ 1, we have the final total weight is at most

$$1 + 1 \cdot |S^*(\hat{h})| \cdot \log(2|\mathcal{X}|) \leq 1 + \mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \cdot \log(2|\mathcal{X}|).$$

The first item follows since the number of inner iterations is at most the number of weight doublings. □

Lemma 48. *For every inner iteration,* $\mathbb{E}[M_i | \mathcal{F}_{i-1}] \leq \sum_{x \in \mathcal{X}} \lambda(W_i(x) - W_{i-1}(x))$.

Proof. The proof is almost a verbatim copy of Dasgupta et al. [85, Lemma 6], which we include here:

$$\begin{aligned}
\mathbb{E}[M_i | \mathcal{F}_{i-1}] &= \sum_{x \in \mathcal{X}} \Pr(x \text{ chosen in round } i | x \text{ not chosen before round } i, \mathcal{F}_{i-1}) \\
&= \sum_{x \in \mathcal{X}} 1 - \Pr(\tau_x > W_i(x) | \tau_x > W_{i-1}(x)) \\
&= \sum_{x \in \mathcal{X}} (1 - \exp(-\lambda(W_i(x) - W_{i-1}(x)))) \\
&\leq \sum_{x \in \mathcal{X}} \lambda(W_i(x) - W_{i-1}(x)).
\end{aligned}$$

□

6.9 Deferred Materials from Section 6.4

6.9.1 Distribution-free Query Complexity Lower Bounds for Auditing with VC classes

Theorem 31 (Lower bound for randomized auditing)**.** *If hypothesis class* $\mathcal{H}$ *has VC dimension* $d \geq 1600$*, and* $\epsilon \in (0, \frac{1}{40}]$*, then for any (possibly randomized) algorithm* $\mathcal{A}$*, there exists a distribution* D *realizable by* $h^* \in \mathcal{H}$*, such that when* $\mathcal{A}$ *is given a querying budget* $N \leq \Omega(\min(d, \frac{1}{\epsilon^2}))$*, its output* $\hat{\mu}$ *is such that*

$$\mathbb{P}\left(\left|\hat{\mu} - \mu(h^*)\right| > \epsilon\right) > \frac{1}{8}.$$

Proof. We will be using Le Cam's method with several subtle modifications. First, we will reduce the estimation problem to a hypothesis testing problem, where under different hypotheses, the $\mu(h^*)$ will be centered around two $\Omega(\epsilon)$-separated values with high probability. Second, we will upper bound the distribution divergence of the interaction history under the two hypotheses; this requires some delicate handling, as the label on a queried example depends not only on the identity of the example, but also historical labeled examples.

Step 1: the construction. As $\mathrm{VC}(\mathcal{H}) = d$, there exists a set of examples $Z = \{z_0, z_1, \ldots, z_{d-1}\} \subset \mathcal{X}$ shattered by $\mathcal{H}$. Let $Z_+ = \{z_1, \ldots, z_{d-1}\}$. Let D_X be as follows: $x \mid x_A = 0$ is uniform over Z_+, whereas $x \mid x_A = 1$ is the delta mass on z_0.

Let $\tilde{\epsilon} = 10 \max(\epsilon, \frac{1}{\sqrt{d}})$; by the conditions that $d \geq 1600$ and $\epsilon \leq \frac{1}{40}$, we have $\tilde{\epsilon} \leq \frac{1}{4}$. Let label budget $N = \frac{1}{24\tilde{\epsilon}^2} = \Omega\left(\min(d, \frac{1}{\epsilon^2})\right)$.

Consider two hypotheses that choose h^* randomly from $\{-1,+1\}^{Z_+}$, subject to $h^*(z_0) = 0$:

- H_0: choose h^* such that for every $i \in [d-1]$, independently, $h^*(z_i) = \begin{cases} +1, & \text{with probability } \frac{1}{2} - \tilde{\epsilon} \\ -1, & \text{with probability } \frac{1}{2} + \tilde{\epsilon} \end{cases}$
- H_1: choose h^* such that for every $i \in [d-1]$, independently, $h^*(z_i) = \begin{cases} +1, & \text{with probability } \frac{1}{2} + \tilde{\epsilon} \\ -1, & \text{with probability } \frac{1}{2} - \tilde{\epsilon} \end{cases}$

We have the following simple claim that shows the separation of $\mu(h^*)$ under the two hypotheses. Its proof is deferred to the end of the main proof.

Claim 3. $\mathbb{P}_{h^* \sim H_0}\left(\mu(h^*) \leq \frac{1}{2} - \frac{1}{2}\tilde{\epsilon}\right) \geq \frac{15}{16}$*, and* $\mathbb{P}_{h^* \sim H_1}\left(\mu(h^*) \geq \frac{1}{2} + \frac{1}{2}\tilde{\epsilon}\right) \geq \frac{15}{16}$.

Step 2: upper bounding the statistical distance. Next, we show that H_0 and H_1 are hard to distinguish with $\mathcal{A}$ having a label budget of N. To this end, we upper bound the KL divergence of the joint distributions of $\langle (x_1, y_1), \ldots, (x_n, y_n) \rangle =: (x, y)_{\leq n}$ under H_0 and H_1, denoted as $\mathbb{P}_0$ and $\mathbb{P}_1$ respectively. Applying Lemma 56, we have:

$$\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_1) = \sum_{i=1}^{n} \mathbb{E}\left[\mathrm{KL}\left(\mathbb{P}_0(y_i = \cdot \mid (x,y)_{\leq i-1}, x_i)), \mathbb{P}_1(y_i = \cdot \mid (x,y)_{\leq i-1}, x_i)\right)\right]. \tag{6.3}$$

We claim that for every i and $((x,y)_{\leq i-1}, x_i) \in (\mathcal{X} \times \mathcal{Y})^{i-1} \times \mathcal{X}$ on the support of $\mathbb{P}_0$,

$$\mathrm{KL}\left(\mathbb{P}_0(y_i = \cdot \mid (x,y)_{\leq i-1}, x_i)), \mathbb{P}_1(y_i = \cdot \mid (x,y)_{\leq i-1}, x_i)\right) \leq 3\tilde{\epsilon}^2. \tag{6.4}$$

First, observe that if $\langle (x,y)_{\leq i-1}, x_i \rangle$ is in the support of $\mathbb{P}_0$, there must exists some $h^* : Z \to \{-1,+1\}$ such that $h^*(x_j) = y_j$ for all $j \in [i-1]$; in particular, this means there must not exist $j_1 \neq j_2$ in $[i-1]$, such that $x_{j_1} = x_{j_2}$ but $y_{j_1} \neq y_{j_2}$.

Next, we note that, under H_0, conditioned on $(x,y)_{\leq i-1}$, the posterior distribution of h^* is supported over the set $\left\{h \mid h : Z \to \{-1,+1\}, \forall j \in [i-1], h(x_j) = y_j\right\}$, and specifically, for all $x \in Z \setminus \left\{x_j : j \in [i-1]\right\}$, the $h^*(x)$'s are independent conditioned on $(x,y)_{\leq i-1}$, and

$$\mathbb{P}_0\left(h^*(x) = +1 \mid (x,y)_{\leq i-1}\right) = \frac{1}{2} - \tilde{\epsilon}.$$

The same statement holds for H_1 except that for all $x \in Z \setminus \{x_j : j \in [i-1]\}$, we now have $\mathbb{P}_1(h^*(x) = +1 \mid (x,y)_{\leq i-1}) = \frac{1}{2}+\tilde{\epsilon}$. In addition, the conditional distribution of $y_i \mid (x,y)_{\leq i-1}, x_i$, equals the conditional distribution of $h^*(x_i) \mid (x,y)_{\leq i-1}$, under both H_0 and H_1. We now perform a case analysis:

1. If $x_i \in \{x_j : j \in [i-1]\}$, then under both H_0 and H_1, the distributions of $h^*(x_i) \mid (x,y)_{\leq i-1}$ are equal: they both equal to the delta mass supported on the only element of the singleton set $\{y_j : j \in [i-1], x_j = x_i\}$. In this case, $\mathrm{KL}\left(\mathbb{P}_0(y_i = \cdot \mid (x,y)_{\leq i-1}, x_i)), \mathbb{P}_1(y_i = \cdot \mid (x,y)_{\leq i-1}, x_i)\right) = 0 \leq 3\tilde{\epsilon}^2$.
2. Otherwise, $x_i \notin \{x_j : j \in [i-1]\}$. Under H_0, $h^*(x_i) \mid (x,y)_{\leq i-1}$ takes value $+1$ with probability $\frac{1}{2} - \tilde{\epsilon}$, and takes value -1 with probability $\frac{1}{2} + \tilde{\epsilon}$; similarly, under H_1, $h^*(x_i) \mid (x,y)_{\leq i-1}$ takes value $+1$ with probability $\frac{1}{2}+\tilde{\epsilon}$, and takes value -1 with probability $\frac{1}{2}-\tilde{\epsilon}$. In this case, by Fact 6 and that $\tilde{\epsilon} \leq \frac{1}{4}$, $\mathrm{KL}\left(\mathbb{P}_0(y_i = \cdot \mid (x,y)_{\leq i-1}, x_i)), \mathbb{P}_1(y_i = \cdot \mid (x,y)_{\leq i-1}, x_i)\right) = \mathrm{kl}\left(\frac{1}{2} - \tilde{\epsilon}, \frac{1}{2} + \tilde{\epsilon}\right) \leq 3\tilde{\epsilon}^2$.

In summary, in both cases, Equation (6.4) holds, and plugging this back to Equation (6.3) with $n = \frac{1}{24\tilde{\epsilon}^2}$, we have $\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_1) \leq 3n\tilde{\epsilon}^2 \leq \frac{1}{8}$. By Pinsker's inequality (Lemma 54), $d_{\mathrm{TV}}(\mathbb{P}_0, \mathbb{P}_1) \leq \sqrt{\frac{1}{2}\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_1)} \leq \frac{1}{2}$. By Le Cam's Lemma (Lemma 53), for any hypothesis tester $\hat{b}$, we have

$$\frac{1}{2}\mathbb{P}_0\left(\hat{b} = 1\right) + \frac{1}{2}\mathbb{P}_1\left(\hat{b} = 0\right) \geq \frac{1}{2}\left(1 - d_{\mathrm{TV}}(\mathbb{P}_0, \mathbb{P}_1)\right) \geq \frac{1}{4}. \tag{6.5}$$

Step 3: concluding the proof. Given $\mathcal{A}$'s output auditing estimate $\hat{\mu}$, consider the following hypothesis test:

$$\hat{b} = \begin{cases} 0, & \hat{\mu} < \frac{1}{2}, \\ 1, & \hat{\mu} \geq \frac{1}{2}. \end{cases}$$

Plugging into Equation (6.5), we have

$$\frac{1}{2}\mathbb{P}_0\left(\hat{\mu} \geq \frac{1}{2}\right) + \frac{1}{2}\mathbb{P}_1\left(\hat{\mu} < \frac{1}{2}\right) \geq \frac{1}{4}. \tag{6.6}$$

Now, recall Claim 3, and using the fact that $\mathbb{P}(A \cap B) \geq \mathbb{P}(A) - \mathbb{P}(B^C) = \mathbb{P}(A) + \mathbb{P}(B) - 1$, we have

$$\mathbb{P}_0\left(\left|\hat{\mu} - \mu(h^*)\right| \geq \frac{1}{2}\tilde{\epsilon}\right) \geq \mathbb{P}_0\left(\hat{\mu} \geq \frac{1}{2}, \mu(h^*) \leq \frac{1}{2} - \frac{1}{2}\tilde{\epsilon}\right) \geq \mathbb{P}_0\left(\hat{\mu} \geq \frac{1}{2}\right) + \frac{15}{16} - 1 \geq \mathbb{P}_0\left(\hat{\mu} \geq \frac{1}{2}\right) - \frac{1}{16}. \tag{6.7}$$

Symmetrically, we also have

$$\mathbb{P}_1\left(\left|\hat{\mu} - \mu(h^*)\right| \geq \frac{1}{2}\tilde{\epsilon}\right) \geq \mathbb{P}_1\left(\hat{\mu} < \frac{1}{2}, \mu(h^*) \geq \frac{1}{2} + \frac{1}{2}\tilde{\epsilon}\right) \geq \mathbb{P}_1\left(\hat{\mu} < \frac{1}{2}\right) - \frac{1}{16}. \tag{6.8}$$

Combining Equations (6.6), (6.7), and (6.8), we have

$$\frac{1}{2}\mathbb{P}_0\left(\left|\hat{\mu} - \mu(h^*)\right| \geq \frac{1}{2}\tilde{\epsilon}\right) + \frac{1}{2}\mathbb{P}_1\left(\left|\hat{\mu} - \mu(h^*)\right| \geq \frac{1}{2}\tilde{\epsilon}\right) \geq \frac{1}{4} - \frac{1}{16} > \frac{1}{8}.$$

As $\frac{1}{2}\tilde{\epsilon} > \epsilon$, and the left hand side can be viewed as the total probability of $\left|\hat{\mu} - \mu(h^*)\right| > \epsilon$ when h^* is drawn from the uniform mixture distribution of the h^* distributions under H_0 and H_1. By the probabilistic method, there exists some h^* such that $\mathbb{P}_{h^*,\mathcal{A}}\left(\left|\hat{\mu} - \mu(h^*)\right| > \epsilon\right) > \frac{1}{8}$. □

Proof of Claim 3. Without loss of generality, we show the first inequality; the second inequality can be shown symmetrically. Note that under H_0, the random h^*'s DP value satisfies

$$\mu(h^*) = \Pr(h^*(x) = +1 \mid x_A = 0) - \Pr(h^*(x) = +1 \mid x_A = 1) = \frac{1}{d-1}\sum_{i=1}^{d-1} \mathbb{1}\{h^*(z_i) = +1\},$$

where the second equality follows from that $\Pr(h^*(x) = +1 \mid x_A = 1) = 0$ as $h^*(z_0) = -1$ is always true.

Under H_0, $(d-1)\mu(h^*)$ is the sum of $(d-1)$ iid Bernoulli random variables with mean parameter $\frac{1}{2} - \tilde{\epsilon}$. Therefore, by Hoeffding's inequality, we have

$$\mathbb{P}_0\left(\mu(h^*) > \frac{1}{2} - \frac{1}{2}\tilde{\epsilon}\right) \leq \exp\left(-2(d-1)\cdot\left(\frac{1}{2}\tilde{\epsilon}\right)^2\right) \leq \frac{1}{16},$$

where the second inequality uses the fact that $\tilde{\epsilon} = 10\max\left(\epsilon, \frac{1}{\sqrt{d}}\right) \geq \frac{10}{\sqrt{d}}$. □

6.9.2 Query Complexity for Auditing Non-homogeneous Halfspaces under Gaussian Subpopulations

Theorem 32 (Lower bound)**.** *Let $d \geq 6400$ and $\epsilon \in (0, \frac{1}{80}]$. If D_X is such that $x \mid x_A = 0 \sim \mathrm{N}(0_d, I_d)$, whereas $x \mid x_A = 1 \sim \mathrm{N}(0_d, (0)_{d\times d})$ (i.e. the delta-mass supported at 0_d). For any (possibly randomized) algorithm $\mathcal{A}$, there exists h^* in $\mathcal{H}_{lin}$ the class of nonhomogeneous linear classifiers, such that when $\mathcal{A}$ is given a query budget $N \leq \Omega\left(\min(d, \frac{1}{\epsilon^2})\right)$, its output $\hat{\mu}$ is such that*

$$\mathbb{P}_{\mathcal{A},h^*}\left(\left|\hat{\mu} - \mu(h^*)\right| > \epsilon\right) > \frac{1}{8}.$$

Proof. Similar to the proof of Theorem 31, we will use Le Cam's method. In addition to the same challenges in the proof of Theorem 31, in the active fairness auditing for halfspaces setting, we are faced with the extra challenge that the posterior distributions of $h^*(x_i) \mid (x, y)_{\leq i-1}$ deviates significantly from the prior distribution of $h^*(x_i)$, and cannot be easily calculated in closed form. To get around this difficulty, using the chain rule of KL divergence, along with the posterior formula for noiseless Bayesian linear regression with Gaussian prior, we calculate a tight upper bound on the KL divergence between two carefully constructed, well-separated hypotheses.

Step 1: the construction. Let $\tilde{\epsilon} = 40\max(\epsilon, \frac{1}{\sqrt{d}})$; by the assumption that $\epsilon \leq \frac{1}{80}$ and $d \geq 6400$, we have $\tilde{\epsilon} \leq \frac{1}{2}$. Let label budget $N = \frac{1}{64\tilde{\epsilon}^2} = \Omega\left(\min(d, \frac{1}{\epsilon^2})\right)$.

Consider two hypotheses that choose $h^* = h_{a^*,b^*}$, such that $b^* = -1$, and a^* is chosen randomly from different distributions:

- $H_0 : a^* \sim \mathrm{N}(0, \frac{1}{d}(1+\tilde{\epsilon})I_d)$
- $H_1 : a^* \sim \mathrm{N}(0, \frac{1}{d}(1-\tilde{\epsilon})I_d)$

We have the following claim that shows the separation of $\mu(h^*)$ under the two hypotheses. Its proof is deferred to the end of the main proof.

Claim 4. $\mathbb{P}_{h^* \sim H_0}\left(\mu(h^*) > \Phi(-1) + \frac{\tilde{\epsilon}}{36}\right) \geq \frac{15}{16}$*, and* $\mathbb{P}_{h^* \sim H_1}\left(\mu(h^*) < \Phi(-1) - \frac{\tilde{\epsilon}}{36}\right) \geq \frac{15}{16}$*, where* $\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} \, \mathrm{d}z$ *is the standard normal CDF.*

Step 2: upper bounding the statistical distance. Next, we show that H_0 and H_1 are hard to distinguish with $\mathcal{A}$ making $n \leq N$ label queries. To this end, we upper bound the KL divergence of the joint distributions of $(x,y)_{\leq n}$ under H_0 and H_1, denoted as $\mathbb{P}_0$ and $\mathbb{P}_1$ respectively. To this end, define $\tilde{y}_i = \langle a^*, x_i \rangle - 1$ for $i \in [n]$, and $y_i = \mathrm{sign}(\tilde{y}_i)$. Define $\tilde{\mathbb{P}}_0$ and $\tilde{\mathbb{P}}_1$ (resp. $\mathbb{Q}_0$ and $\mathbb{Q}_1$) as the joint distributions of $(x,\tilde{y})_{\leq n}$ (resp. $(x,y,\tilde{y})_{\leq n}$) under H_0 and H_1 respectively. By the chain rule of KL divergence (Lemma 55 with $Z = (x,y)_{\leq n}, W = \tilde{y}_{\leq n}$ and $Z = (x,\tilde{y})_{\leq n}, W = y_{\leq n}$ respectively), we get:

$$\begin{aligned}
&\mathrm{KL}(\mathbb{Q}_0((x,y,\tilde{y})_{\leq n}), \mathbb{Q}_1((x,y,\tilde{y})_{\leq n}) \\
=&\underbrace{\mathrm{KL}(\mathbb{Q}_0((x,y)_{\leq n}), \mathbb{Q}_1((x,y)_{\leq n})}_{\mathrm{KL}(\mathbb{P}_0,\mathbb{P}_1)} + \underbrace{\mathrm{KL}(\mathbb{Q}_0((\tilde{y})_{\leq n} \mid (x,y)_{\leq n}), \mathbb{Q}_1((\tilde{y})_{\leq n} \mid (x,y)_{\leq n}))}_{\geq 0} \\
=&\underbrace{\mathrm{KL}(\mathbb{Q}_0((x,\tilde{y})_{\leq n}), \mathbb{Q}_1((x,\tilde{y})_{\leq n})}_{\mathrm{KL}(\tilde{\mathbb{P}}_0,\tilde{\mathbb{P}}_1)} + \underbrace{\mathrm{KL}(\mathbb{Q}_0((y)_{\leq n} \mid (x,\tilde{y})_{\leq n}), \mathbb{Q}_1((y)_{\leq n} \mid (x,\tilde{y})_{\leq n}))}_{0},
\end{aligned}$$

where the last term is 0 because under both $\mathbb{Q}_0$ and $\mathbb{Q}_1$, $(y)_{\leq n} \mid (x,\tilde{y})_{\leq n}$ is the delta mass supported on $(\mathrm{sign}(\tilde{y}))_{\leq n}$. As a consequence,

$$\mathrm{KL}(\mathbb{P}_0,\mathbb{P}_1) \leq \mathrm{KL}(\tilde{\mathbb{P}}_0,\tilde{\mathbb{P}}_1)$$

Also, note that $\mathcal{A}$ can be viewed as a query learning algorithm that at round i, receives $(x,\tilde{y})_{\leq i-1}$ as input, and choose the next example for query (i.e., it elects to only use the thresholded value y_j's as opposed to the $\tilde{y}_j$'s). Applying Lemma 56, we have:

$$\mathrm{KL}(\tilde{\mathbb{P}}_0,\tilde{\mathbb{P}}_1) = \sum_{i=1}^{n} \mathbb{E}\left[\mathrm{KL}(\mathbb{P}_0(\tilde{y}_i = \cdot \mid (x,\tilde{y})_{\leq i-1}, x_i)), \mathbb{P}_1(\tilde{y}_i = \cdot \mid (x,\tilde{y})_{\leq i-1}, x_i))\right]. \tag{6.9}$$

We claim that for every i and $((x,\tilde{y})_{\leq i-1}, x_i) \in (\mathcal{X} \times \mathcal{Y})^{i-1} \times \mathcal{X}$ on the support of $\tilde{\mathbb{P}}_0$,

$$\mathrm{KL}(\mathbb{P}_0(\tilde{y}_i = \cdot \mid (x,\tilde{y})_{\leq i-1}, x_i)), \mathbb{P}_1(\tilde{y}_i = \cdot \mid (x,\tilde{y})_{\leq i-1}, x_i)) \leq 3\tilde{\epsilon}^2. \tag{6.10}$$

First, by Lemma 49 (deferred to the end of the proof), under H_0, conditioned on $(x,\tilde{y})_{\leq i-1}$ on the support of $\tilde{\mathbb{P}}_0$, the posterior distribution of a^* is the same as $a^* \sim \mathrm{N}(0, \frac{1}{d}(1+\tilde{\epsilon})I_d)$ conditioned on the affine set $S = \left\{a \in \mathbb{R}^d : \langle a, x_l \rangle + 1 = \tilde{y}_l, \forall l \in [i-1]\right\}$. Denote $X_{i-1} = [x_1^\top; x_2^\top; \ldots, x_{i-1}^\top] \in \mathbb{R}^{(i-1)\times d}$, and $\tilde{Y}_{i-1} = (\tilde{y}_1, \ldots, \tilde{y}_{i-1})$; for $(x,\tilde{y})_{\leq i-1}$ on the support of $\tilde{\mathbb{P}}_0$, it must be the case that $S \neq \emptyset$, and as a result, $\hat{a} = X_{i-1}^\dagger(\tilde{Y}_{i-1} - \mathbb{1}_{i-1}) \in S$. Also, denote by $X_{i-1}^\perp$

a matrix whose columns are an orthonormal basis of $\mathrm{span}(x_1, \ldots, x_{i-1})$; such a $X_{i-1}^\perp$ is always well-defined as $i - 1 \leq n - 1 \leq d - 1$. Applying Lemma 57, we have

$$a^* \mid (x, \tilde{y})_{\leq i-1} \sim \mathrm{N}\left(\hat{a}, \frac{1}{d}(1 + \tilde{\epsilon}) X_{i-1}^\perp (X_{i-1}^\perp)^\top\right),$$

with its covariance matrix $\frac{1}{d}(1 + \tilde{\epsilon}) X_{i-1}^\perp (X_{i-1}^\perp)^\top$ being rank-deficient.

Now, observe that $\tilde{y}_i \mid (x, \tilde{y})_{\leq i-1}, x_i$ has the same distribution as $\langle a^*, x_i \rangle + 1 \mid (x, \tilde{y})_{\leq i-1}$, which is $\mathrm{N}\left(\langle \hat{a}, x_i \rangle + 1, \frac{1}{d}(1 + \tilde{\epsilon}) x_i^\top X_{i-1}^\perp (X_{i-1}^\perp)^\top x_i\right)$.

Similarly, under H_1, we have $\tilde{y}_i \mid (x, \tilde{y})_{\leq i-1}, x_i$ has distribution $\mathrm{N}\left(\langle \hat{a}, x_i \rangle + 1, \frac{1}{d}(1 - \tilde{\epsilon}) x_i^\top X_{i-1}^\perp (X_{i-1}^\perp)^\top x_i\right)$. We now prove (6.10) by a case analysis:

1. If $x_i \in \mathrm{span}(x_1, \ldots, x_{i-1})$, then $(X_{i-1}^\perp)^\top x_i = 0$, and under both H_0 and H_1, the posterior distributions of $\tilde{y}_i \mid (x, \tilde{y})_{\leq i-1}, x_i$ are both delta mass on $\langle \hat{a}, x_i \rangle + 1$, and therefore, $\mathrm{KL}(\mathbb{P}_0(\tilde{y}_i = \cdot \mid (x, \tilde{y})_{\leq i-1}, x_i)), \mathbb{P}_1(\tilde{y}_i = \cdot \mid (x, \tilde{y})_{\leq i-1}, x_i)) = 0 \leq 3\tilde{\epsilon}^2$.
2. If $x_i \notin \mathrm{span}(x_1, \ldots, x_{i-1})$, then $(X_{i-1}^\perp)^\top x_i \neq 0$, and under H_0 and H_1, the posterior distributions of $\tilde{y}_i \mid (x, \tilde{y})_{\leq i-1}, x_i$ are $\mathrm{N}(\hat{\mu}_i, (1 + \tilde{\epsilon})\sigma_i^2)$ and $\mathrm{N}(\hat{\mu}_i, (1 - \tilde{\epsilon})\sigma_i^2)$ respectively, where $\hat{\mu}_i = \langle \hat{a}, x_i \rangle + 1$, and $\sigma_i^2 = \frac{1}{d} x_i^\top X_{i-1}^\perp (X_{i-1}^\perp)^\top x_i$. In this case, by Fact 7,

$$\begin{aligned}
&\mathrm{KL}\left(\mathbb{P}_0(\tilde{y}_i = \cdot \mid (x, \tilde{y})_{\leq i-1}, x_i)), \mathbb{P}_1(\tilde{y}_i = \cdot \mid (x, \tilde{y})_{\leq i-1}, x_i))\right) \\
=&\mathrm{KL}\left(\mathrm{N}(\hat{\mu}_i, (1 + \tilde{\epsilon})\sigma_i^2), \mathrm{N}(\hat{\mu}_i, (1 - \tilde{\epsilon})\sigma_i^2)\right) \\
=&\frac{1}{2}\left(\frac{1 + \tilde{\epsilon}}{1 - \tilde{\epsilon}} - 1 + \ln(\frac{1 - \tilde{\epsilon}}{1 + \tilde{\epsilon}})\right) \\
\leq&\frac{1}{2}\left(\frac{2\tilde{\epsilon}}{1 - \tilde{\epsilon}}\right)^2 \\
\leq&8\tilde{\epsilon}^2,
\end{aligned}$$

where the first inequality is by the fact that $\ln(1 + x) \geq x - x^2$ when $x \geq 0$, and taking $x = \frac{2\tilde{\epsilon}}{1 - \tilde{\epsilon}}$, and the second inequality is from $\tilde{\epsilon} \leq \frac{1}{2}$ and algebra.

In summary, in both cases, Equation (6.10) holds, and plugging this back to Equation (6.9) with $n \leq \frac{1}{64\tilde{\epsilon}^2}$, we have $\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_1) \leq 8n\tilde{\epsilon}^2 \leq \frac{1}{8}$. By Pinsker's inequality (Lemma 54), $d_{\mathrm{TV}}(\mathbb{P}_0, \mathbb{P}_1) \leq \sqrt{\frac{1}{2}\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_1)} \leq \frac{1}{2}$. Le Cam's lemma (Lemma 53) implies that, for any hypothesis tester $\hat{b}$, we have

$$\frac{1}{2}\mathbb{P}_0(\hat{b} = 1) + \frac{1}{2}\mathbb{P}_1(\hat{b} = 0) = \frac{1}{2}(1 - d_{\mathrm{TV}}(\mathbb{P}_0, \mathbb{P}_1)) \geq \frac{1}{4}.$$

Step 3: concluding the proof. Given $\mathcal{A}$'s output auditing estimate $\hat{\mu}$, consider the following hypothesis tester:

$$\hat{b} = \begin{cases} 0, & \hat{\mu} > \Phi(-1), \\ 1, & \hat{\mu} \leq \Phi(-1). \end{cases}$$

Plugging into Equation (6.5), we have

$$\frac{1}{2}\mathbb{P}_0\left(\hat{\mu} \leq \Phi(-1)\right) + \frac{1}{2}\mathbb{P}_1\left(\hat{\mu} > \Phi(-1)\right) \geq \frac{1}{4}. \tag{6.11}$$

Now, recall Claim 4, and using the fact that $\mathbb{P}(A \cap B) \geq \mathbb{P}(A) - \mathbb{P}(B^C) = \mathbb{P}(A) + \mathbb{P}(B) - 1$, we have

$$\begin{aligned}
&\mathbb{P}_0\left(\left|\hat{\mu} - \mu(h^*)\right| \geq \frac{1}{36}\tilde{\epsilon}\right) \qquad (6.12)\\
&\geq \mathbb{P}_0\left(\hat{\mu} \leq \Phi(-1), \mu(h^*) > \Phi(1) - \frac{1}{36}\tilde{\epsilon}\right)\\
&\geq \mathbb{P}_0\left(\hat{\mu} \leq \Phi(-1)\right) + \frac{15}{16} - 1\\
&\geq \mathbb{P}_0\left(\hat{\mu} \leq \Phi(-1)\right) - \frac{1}{16}.
\end{aligned}$$

Symmetrically, we also have

$$\mathbb{P}_1\left(\left|\hat{\mu} - \mu(h^*)\right| \geq \frac{1}{36}\tilde{\epsilon}\right) \geq \mathbb{P}_1\left(\hat{\mu} > \Phi(-1)\right) - \frac{1}{16}. \qquad (6.13)$$

Combining Equations (6.11), (6.12), and (6.13), we have

$$\frac{1}{2}\mathbb{P}_0\left(\left|\hat{\mu} - \mu(h^*)\right| \geq \frac{1}{36}\tilde{\epsilon}\right) + \frac{1}{2}\mathbb{P}_1\left(\left|\hat{\mu} - \mu(h^*)\right| \geq \frac{1}{36}\tilde{\epsilon}\right) \geq \frac{1}{4} - \frac{1}{16} > \frac{1}{8}.$$

As $\frac{1}{36}\tilde{\epsilon} \geq \epsilon$, and the left hand side can be viewed as the total probability of $\left|\hat{\mu} - \mu(h^*)\right| \geq \epsilon$ when h^* is drawn from the uniform mixture distribution of the h^* distributions under H_0 and H_1. By the probabilistic method, there exists some $h^* \in \mathcal{H}$ such that $\mathbb{P}_{h^*}\left(\left|\hat{\mu} - \mu(h^*)\right| > \epsilon\right) > \frac{1}{8}$. □

Lemma 49. *Given the same setting above. For any fixed $i \in \mathbb{N}$ and $(x, \tilde{y})_{\leq i}$, the posterior distribution $a^* \mid (x, \tilde{y})_{\leq i}$ is the same as $a^* \mid \{a^* \in U\}$, where $U = \left\{a : \forall j \in [i] : \langle x_j, a\rangle + 1 = \tilde{y}_j\right\}$.*

Proof. We use the Bayes formula to expand the posterior; below $\propto$ denotes equality up to a multiplicative factor independent of a^*.

$$\begin{aligned}
\mathbb{P}(a^* \mid (x, \tilde{y})_{\leq i}) \propto &\mathbb{P}(a^*, (x, \tilde{y})_{\leq i})\\
\propto &\mathbb{P}(a^*)\prod_{j=1}^{i}\mathbb{P}(x_j \mid a^*, (x, \tilde{y})_{\leq j-1})\mathbb{P}(\tilde{y}_j \mid x_j, a^*, (x, \tilde{y})_{\leq j-1})\\
\propto &\mathbb{P}(a^*)\prod_{j=1}^{i}\mathbb{P}(x_j \mid (x, \tilde{y})_{\leq j-1})\mathbb{1}\left\{\tilde{y}_j = \langle x_j, a^*\rangle + 1\right\}\\
\propto &\mathbb{P}(a^*)\prod_{j=1}^{i}\mathbb{1}\left\{\tilde{y}_j = \langle x_j, a^*\rangle + 1\right\}
\end{aligned}$$

where the second equality uses the definition of conditional probability; the third equality uses the fact that for any fixed query learning algorithm $\mathcal{A}$, x_j is independent of a^* conditioned on $(x, \tilde{y})_{\leq j-1}$, and the observation that given x_j and a^*, $\tilde{y}_j = \langle x_j, a^*\rangle + 1$ deterministically. This concludes the proof. □

Proof of Claim 4. For $h^*(x) = \mathrm{sign}(\langle a^*, x\rangle + b^*)$ where $b^* = -1$, it can be seen that,

$$\mathbb{P}_0(h^*(x) = +1 \mid x_A = 1) = 0,$$

On the other hand,

$$\mathbb{P}_0(h^*(x) = +1 \mid x_A = 0) = \mathbb{P}_{z\sim \mathrm{N}(0,I_d)}(\langle a^*, z\rangle \geq 1) = \mathbb{P}_{z\sim \mathrm{N}(0,I_d)}\left(\left\langle \frac{a^*}{\|a^*\|}, z\right\rangle \geq \frac{1}{\|a^*\|}\right) = 1-\Phi\left(\frac{1}{\|a^*\|}\right).$$

Also, note that under H_0, $\frac{d\|a^*\|_2^2}{(1+\tilde{\epsilon})} \sim \chi^2(d)$; Therefore, by Fact 8, we have that with probability $\geq \frac{15}{16}$, $\frac{d\|a^*\|_2^2}{(1+\tilde{\epsilon})} \geq d\cdot(1 - 10\sqrt{\frac{1}{d}})$, which implies that

$$\frac{1}{\|a^*\|} \leq \sqrt{\frac{1}{(1+\tilde{\epsilon})(1-10\sqrt{\frac{1}{d}})}} \leq \sqrt{\frac{1}{(1+\tilde{\epsilon})(1-\frac{\tilde{\epsilon}}{4})}} \leq 1 - \frac{\tilde{\epsilon}}{4}.$$

Therefore, as for every $a, b \in [\frac{3}{4}, 1]$, $\left|\Phi(a) - \Phi(b)\right| \geq \min_{\xi\in[\frac{3}{4},1]} \Phi'(\xi)|a-b| \geq \frac{1}{9}|a-b|$, we have:

$$1-\Phi\left(\frac{1}{\|a^*\|}\right) \geq 1-\Phi\left(1-\frac{\tilde{\epsilon}}{4}\right) \geq 1-(\Phi(1)-\frac{\tilde{\epsilon}}{36}) \geq \Phi(-1)+\frac{\tilde{\epsilon}}{36}.$$

This concludes the proof of the first inequality. The second inequality is proved symmetrically. □

We now present our (deterministic) active fairness auditing algorithm, Algorithm 22 and its guarantees. Algorithm 22 works under the setting when the two subpopulations are Gaussian, whose mean and covariance parameters (m_0, Σ_0), (m_1, Σ_1) are known. It also assumes access to black-box queries to $h^* \in \mathcal{H}_{\text{lin}} = \left\{h_{a,b}(x) := \mathrm{sign}(\langle a, x\rangle + b) : a \in \mathbb{R}^d, b \in \mathbb{R}\right\}$, and aims to estimate $\mu(h^*)$ within precision ϵ. Recall that

$$\mu(h^*) = \mathrm{Pr}_{x\sim D_X}\left(h^*(x) = 1 \mid x_A = 0\right) - \mathrm{Pr}_{x\sim D_X}\left(h^*(x) = 1 \mid x_A = 1\right),$$

it suffices to estimate $\gamma_b := \mathrm{Pr}_{x\sim D_X}\left(h^*(x) = 1 \mid x_A = 0\right)$ within precision $\epsilon/2$, for each $b \in \{0,1\}$. To this end, we note that

$$\gamma_b = \mathrm{Pr}_{x\sim \mathrm{N}(m_b,\Sigma_b)}\left(h^*(x) = 1\right) = \mathrm{Pr}_{\tilde{x}\sim \mathrm{N}(0,I_d)}\left(h^*(m_b + \Sigma_b^{1/2}\tilde{x}) = 1\right);$$

if we define $\tilde{h}_b : \mathbb{R}^d \to \{-1,+1\}$ such that

$$\tilde{h}_b(\tilde{x}) = h^*(m_b + \Sigma_b^{1/2}\tilde{x}), \tag{6.14}$$

γ_b equals to $\gamma(\tilde{h}_b)$, where $\gamma(h) = \mathbb{P}_{\tilde{x}\sim \mathrm{N}(0,I_d)}\left(h(\tilde{x}) = 1\right)$ is the probability of positive prediction of h under the standard Gaussian distribution. Importantly, as h^* is a linear classifier, $\tilde{h}_b$ is also a linear classifier and lies in $\mathcal{H}_{\text{lin}}$.

Recall that procedure ESTIMATE-POSITIVE (Algorithm 20) label-efficiently estimates $\gamma(h)$ for any $h \in \mathcal{H}_{\text{lin}}$, using query access to h. Algorithm 22 uses it as a subprocedure to estimate $\gamma_b = \gamma(\tilde{h}_b)$ (line 3). To simulate label queries to $\tilde{h}_b$ using query access to h^*, according to

Algorithm 22 Active fairness auditing for nonhomogeneous linear classifiers under Gaussian subpopulations

```
Require: Subpopulation parameters $(m_0, \Sigma_0)$, $(m_1, \Sigma_1)$, query access to $h^* \in \mathcal{H}_{\text{lin}}$, target error $\epsilon$.
Ensure: $\hat{\mu}$ such that $\left|\hat{\mu} - \mu(h^*)\right| \leq \epsilon$.
1: for $b \in \{0, 1\}$ do
2:     Define $\tilde{h}_b : \mathbb{R}^d \to \{-1, +1\}$ such that $\tilde{h}_b(\tilde{x}) = h^*(m_b + \Sigma_b^{1/2}\tilde{x})$;   ▷ $\tilde{h}_b \in \mathcal{H}_{lin}$, and each query to $\tilde{h}_b$ can be simulated by one query to $h^*$
3:     $\hat{\gamma}_b \leftarrow$ ESTIMATE-POSITIVE$(\tilde{h}_b, \frac{\epsilon}{2})$
   return $\hat{\gamma}_0 - \hat{\gamma}_1$
```

Equation (6.14), it suffices to apply an affine transformation on the input $\tilde{x}$, obtaining transformed input $m_b + \Sigma_b^{1/2}\tilde{x}$, and query h^* on the transformed input.

Finally, after $\hat{\gamma}_0, \hat{\gamma}_1$, $\epsilon/2$-accurate estimators of γ_0, γ_1 are obtained, Algorithm 22 takes their difference as our estimator $\hat{\mu}$ for $\mu(h^*)$ (line 3).

Theorem 33 (Upper bound). *If $h^* \in \mathcal{H}_{lin}$, D_X is such that $x \mid x_A = 0 \sim \mathrm{N}(m_0, \Sigma_0)$, $x \mid x_A = 1 \sim \mathrm{N}(m_1, \Sigma_1)$. Algorithm 22 outputs $\hat{\mu}$, such that with probability 1, $\left|\hat{\mu} - \mu(h^*)\right| \leq \epsilon$; moreover, Algorithm 22 makes at most $O(d \ln \frac{d}{\epsilon})$ label queries to h^*.*

Proof. As we will see from Lemma 50, for $b \in \{0, 1\}$, the respective calls of ESTIMATE-POSITIVE ensures that

$$|\hat{\gamma}_b - \gamma_b| \leq \frac{\epsilon}{2}.$$

Therefore,

$$\left|\hat{\mu} - \mu(h^*)\right| \leq |\hat{\gamma}_0 - \gamma_0| + |\hat{\gamma}_1 - \gamma_1| \leq \epsilon.$$

Moreover, for every b, Lemma 50 ensures that each call to ESTIMATE-POSITIVE only makes at most $O(d \ln \frac{d}{\epsilon})$ label queries to $\tilde{h}_b$; as simulating each query to $\tilde{h}_b$ takes one query to h^*, for every b, it also makes at most $O(d \ln \frac{d}{\epsilon})$ label queries to h^*. Summing the number of label queries over $b \in \{0, 1\}$, the total number of label queries by Algorithm 22 is $O(d \ln \frac{d}{\epsilon})$. □

We now turn to presenting the guarantee of the key subprocedure ESTIMATE-POSITIVE and its proof. This expands the analysis sketch in Section 6.4.3.

Lemma 50 (Guarantees of ESTIMATE-POSITIVE). *Recall that $\gamma(h) = \Pr_{x \sim \mathrm{N}(0, I_d)}(h(x) = +1)$. ESTIMATE-POSITIVE (Algorithm 20) receives inputs query access to $h^* \in \mathcal{H}_{lin}$, and target error ϵ, and outputs $\hat{\gamma}$ such that*

$$\left|\hat{\gamma} - \gamma(h^*)\right| \leq \epsilon. \tag{6.15}$$

Furthermore, it makes at most $O(d \ln \frac{d}{\epsilon})$ queries to h^*.

Proof. Let $h^*(x) = \mathrm{sign}(\langle a^*, x\rangle + b^*)$ be the target classifier. First, observe that $\gamma(h^*) = \Phi\left(\frac{b^*}{\|a^*\|_2}\right) =: \Phi(sr)$, where Φ is the standard normal CDF, $s := \mathrm{sign}(b^*)$, and $r := \sqrt{\frac{1}{\sum_{i=1}^d m_i^{-2}}}$, for $m_i := -\frac{b^*}{a_i^*}$. Note that line 2 of ESTIMATE-POSITIVE correctly obtains s, as $s = h^*(\mathbf{0}) = \mathrm{sign}(\langle a^*, \mathbf{0}\rangle + b) = \mathrm{sign}(b)$.

Recall that $\alpha = \sqrt{2d \ln \frac{1}{\epsilon}}$ and $\beta = 2d^{\frac{5}{2}}(\ln \frac{1}{\epsilon})^{\frac{3}{4}}(\frac{1}{\epsilon})^{\frac{1}{2}}$. We consider two cases depending on the line in which ESTIMATE-POSITIVE returns:

1. If ESTIMATE-POSITIVE returns in line 4, then it must be the case that for all $i \in [d]$, $h^*(\alpha e_i) = h^*(-\alpha e_i)$. In this case, by Lemma 52, we have that for every $i, |m_i| \geq \alpha$. This implies that $r = \sqrt{\frac{1}{\sum_{i=1}^d m_i^{-2}}} \geq \sqrt{\frac{1}{d\alpha^{-2}}} \geq \sqrt{2 \ln \frac{1}{\epsilon}}$. For the case that $s = -1$, we have that $\gamma(h^*) = \Phi(sr) \leq \epsilon$, where we use the standard fact that $\Phi(x) \leq \exp(-\frac{x^2}{2})$ for $x \leq 0$; in this case $\hat{\gamma} = 0$ ensures Equation (6.15) holds; for the symmetric case that $s = +1$, $\gamma(h^*) = \Phi(sr) \geq 1 - \epsilon$ and $\hat{\gamma} = 1$, which also ensures Equation (6.15).
2. On the other hand, ESTIMATE-POSITIVE returns in line 12, it must be the case that there exists some $i_0 \in [d]$, such that $|m_{i_0}| \leq \alpha$. This implies that $r = \sqrt{\frac{1}{\sum_{i=1}^d m_i^{-2}}} \leq \sqrt{\frac{1}{m_{i_0}^{-2}}} = |m_{i_0}| \leq \alpha$.

   Now, ESTIMATE-POSITIVE must execute lines 5 to 10. The final S it computes has the following properties: for every $i \in S$ added, by the guarantee of procedure BINARY-SEARCH (Algorithm 21), $|\hat{m}_i - m_i| \leq \epsilon$; otherwise, for $i \notin S$, it must be the case that $h^*(\beta e_i) \neq h^*(-\beta e_i)$, which, by Lemma 52, implies that $|m_i| \geq \beta$. Therefore, all the conditions of Lemma 39 are satisfied, and thus, $|\hat{r} - r| \leq 2\epsilon$. This also yields that $|s\hat{r} - sr| \leq 2\epsilon$. Finally, note that Φ is $\frac{1}{\sqrt{2\pi}}$-Lipschitz, we have

$$\left|\hat{\gamma} - \gamma(h^*)\right| = \left|\Phi(s\hat{r}) - \Phi(sr)\right| \leq \frac{1}{\sqrt{2\pi}} \cdot |s\hat{r} - sr| \leq \epsilon.$$

In summary, in both cases, ESTIMATE-POSITIVE outputs $\hat{\gamma}$ such that Equation (6.15) is satisfied.

We now calculate the total query complexity of ESTIMATE-POSITIVE. Line 2 makes 1 label query; line 3 makes $2d$ label queries; for each $i \in [d]$, line 7 makes 2 label queries, and BINARY-SEARCH makes $\log \frac{2\beta}{\epsilon}$ label queries. In summary, the total label query complexity of ESTIMATE-POSITIVE is:

$$1 + 2d + d(2 + \log \frac{2\beta}{\epsilon}) = O\left(d \ln \frac{d}{\epsilon}\right).$$

We now present the proof of Lemma 39, which is key to the proof of Lemma 50.

Proof of Lemma 39. First, by Lemma 51, and the assumption that for all $i \in S, |\hat{m}_i - m_i| \leq \epsilon$, we have

$$\left|\sqrt{\frac{1}{\sum_{i \in S} \hat{m}_i^{-2}}} - \sqrt{\frac{1}{\sum_{i \in S} m_i^{-2}}}\right| \leq \epsilon.$$

It remains to prove that

$$\left|\sqrt{\frac{1}{\sum_{i \in S} m_i^{-2}}} - \sqrt{\frac{1}{\sum_{i=1}^d m_i^{-2}}}\right| \leq \epsilon,$$

which combined with the above inequality, will conclude the proof.

To see this, let $z = \sum_{i=1}^d m_i^{-2}$ and $z_S = \sum_{i\in S} m_i^{-2}$; since for all $i \notin S, |m_i| \geq \beta$, this implies that

$$|z - z_S| \leq \frac{d}{\beta^2} \leq \frac{2\epsilon}{(4d\ln\frac{1}{\epsilon})^{\frac{3}{2}}},$$

Also, note that $\sqrt{\frac{1}{\sum_{i=1}^d m_i^{-2}}} = r \leq \alpha$ implies that $z \geq \frac{1}{\alpha^2} = \frac{1}{2d\ln\frac{1}{\epsilon}}$; therefore, $z_S \geq z - \frac{2\epsilon}{(4d\ln\frac{1}{\epsilon})^{\frac{3}{2}}} \geq \frac{1}{4d\ln\frac{1}{\epsilon}}$. Now, by Lagrange mean value theorem,

$$\left|\frac{1}{\sqrt{z_S}} - \frac{1}{\sqrt{z}}\right| \leq \max_{z' \in (z_S, z)} \frac{1}{2}(z')^{-\frac{3}{2}} \cdot |z_s - z| \leq \frac{1}{2}(z_S)^{-\frac{3}{2}} \cdot |z_s - z| \leq \frac{1}{2}(4d\ln\frac{1}{\epsilon})^{\frac{3}{2}} \cdot \frac{2\epsilon}{(4d\ln\frac{1}{\epsilon})^{\frac{3}{2}}} \leq \epsilon.$$

This concludes the proof. □

Lemma 51. *Let $l \in \mathbb{N}_+$ and $f(m_1, \ldots, m_l) := \sqrt{\frac{1}{\sum_{i=1}^l m_i^{-2}}}$; then f is 1-Lipschitz with respect to $\|\cdot\|_\infty$.*

Proof. First, we show that f is 1-Lipschitz with respect to $\|\cdot\|_\infty$ in each of the orthants of $\mathbb{R}^l$. Without loss of generality, we focus on the positive orthant $R =: \left\{m \in \mathbb{R}^l : m_i \geq 0, \forall i\right\}$. We now check that for any two points $\mathbf{m}$ and $\mathbf{n}$ in $R, \left|f(\mathbf{m}) - f(\mathbf{n})\right| \leq \|\mathbf{m} - \mathbf{n}\|_\infty$. By Lagrange mean value theorem, there exists some $\theta \in \left\{t\mathbf{m} + (1-t)\mathbf{n} : t \in (0,1)\right\}$, such that

$$\left|f(\mathbf{m}) - f(\mathbf{n})\right| = \left|\langle \nabla f(\theta), \mathbf{m} - \mathbf{n}\rangle\right| \leq \|\nabla f(\theta)\|_1 \|\mathbf{m} - \mathbf{n}\|_\infty,$$

where the second inequality is from Hölder's inequalty. Therefore, it suffices to check that for all $\mathbf{m}$ in the $R_0 =: \left\{\mathbf{m} \in \mathbb{R}^l : m_i > 0, \forall i\right\}$ (interior of R), $\|\nabla f(m_1, \ldots, m_l)\|_1 \leq 1$. To see this, note that

$$\nabla f(m_1, \ldots, m_d) = \left(\frac{m_1^{-3}}{(\sum_{i=1}^l m_i^{-2})^{\frac{3}{2}}}, \ldots, \frac{m_l^{-3}}{(\sum_{i=1}^l m_i^{-2})^{\frac{3}{2}}}\right) =: g,$$

Observe that $\sum_{i=1}^l |g_i|^{\frac{2}{3}} = 1$; this implies that for every $i \in [l], |g_i| \leq 1$, and therefore,

$$\|g\|_1 = \sum_{i=1}^l |g_i| \leq 1.$$

Now consider $\mathbf{m}, \mathbf{n} \in \mathbb{R}^l$ that do not necessarily lie in the same orthant. Suppose the line segment $\left\{t\mathbf{m} + (1-t)\mathbf{n} : t \in [0,1]\right\}$ consists of k pieces, where piece i is $\left\{t\mathbf{m} + (1-t)\mathbf{n} : t \in [t_{i-1}, t_i]\right\}$, where $1 = t_0 > t_1 > \ldots > t_k = 0$, where each piece is contained in an orthant. Then we have:

$$\begin{aligned}
\left|f(\mathbf{m}) - f(\mathbf{n})\right| \leq& \sum_{i=1}^k \left|f(t_{i-1}\mathbf{m} + (1-t_{i-1})\mathbf{n}) - f(t_i\mathbf{m} + (1-t_i)\mathbf{n})\right| \\
\leq& \sum_{i=1}^k \|(t_{i-1}\mathbf{m} + (1-t_{i-1})\mathbf{n}) - (t_i\mathbf{m} + (1-t_i)\mathbf{n})\|_\infty \\
=& \sum_{i=1}^k (t_{i-1} - t_i)\|\mathbf{m} - \mathbf{n}\|_\infty \\
=& \|\mathbf{m} - \mathbf{n}\|_\infty,
\end{aligned}$$

where the second inequality uses the Lipchitzness of f within the orthant that contains piece i, for each i in $[k]$. □

Lemma 52. *Given $i \in [d]$ and $\xi > 0$, if $h^*(\xi e_i) = h^*(-\xi e_i)$, then $|m_i| \geq \xi$.*

Proof. Suppose $h^*(\xi e_i) = h^*(-\xi e_i) = +1$; in this case, $-b_i \leq \xi a_i^* \leq b_i$, and therefore, $|\xi a_i^*| \leq b_i$, which implies that $|m_i| \geq \xi$. The case of $h^*(\xi e_i) = h^*(-\xi e_i) = +1$ can be proved symmetrically. □

6.9.3 Auxiliary Lemmas for Query Learning Lower Bounds

In this subsection we collect a few standard and useful lemmas for establishing lower bounds for general adaptive sampling and query learning algorithms, including active fairness auditing algorithms. Throughout, denote by $\mathbb{P}$ the distribution of interaction transcript (the sequence of N labeled examples $\langle (x_1, y_1), \ldots, (x_N, y_N) \rangle$) obtained by the query learning algorithm by interacting with the environment, and use the shorthand $(x, y)_{\leq i}$ to denote $\langle (x_1, y_1), \ldots, (x_i, y_i) \rangle$.

Lemma 53 (Le Cam's Lemma)**.** *Given two distributions $\mathbb{P}_0$, $\mathbb{P}_1$ over observation space $z \in \mathcal{Z}$, and let $\hat{b} : \mathcal{Z} \to \{0, 1\}$ be any hypothesis tester. Then,*

$$\frac{1}{2}\mathbb{P}_0\left(\hat{b}(Z) = 1\right) + \frac{1}{2}\mathbb{P}_1\left(\hat{b}(Z) = 0\right) \geq \frac{1}{2}\left(1 - d_{\mathrm{TV}}(\mathbb{P}_0, \mathbb{P}_1)\right),$$

where $d_{\mathrm{TV}}(\mathbb{P}_0, \mathbb{P}_1)$ denotes the total variation distance between $\mathbb{P}_0$ and $\mathbb{P}_1$.

Lemma 54 (Pinsker's Inequality)**.** *For two distributions $\mathbb{P}$ and $\mathbb{Q}$, $d_{\mathrm{TV}}(\mathbb{P}_0, \mathbb{P}_1) \leq \sqrt{\frac{1}{2}\mathrm{KL}(\mathbb{P}, \mathbb{Q})}$.*

Lemma 55 (Chain rule of KL divergence)**.** *For two distributions $\mathbb{Q}^0(Z, W)$ and $\mathbb{Q}^1(Z, W)$ over $\mathcal{Z} \times \mathcal{W}$, we have*

$$\mathrm{KL}(\mathbb{Q}^0, \mathbb{Q}^1) = \mathrm{KL}(\mathbb{Q}^0_Z, \mathbb{Q}^1_Z) + \mathbb{E}_{z \sim \mathbb{Q}^0_Z}\left[\mathrm{KL}(\mathbb{Q}^0_{W|Z}(\cdot \mid z), \mathbb{Q}^1_{W|Z}(\cdot \mid z))\right].$$

Fact 6. *Let $\mathrm{kl}(\cdot, \cdot)$ denote the binary relative entropy function. For $a, b \in [\frac{1}{4}, \frac{3}{4}]$, $\mathrm{kl}(a, b) \leq 3(b - a)^2$.*

The following lemma is well-known.

Lemma 56 (Divergence decomposition)**.** *For a (possibly randomized) query learning algorithm $\mathcal{A}$ with label budget N, under two hypotheses H_0, H_1 (represented by distributions over the target concept h^*), we have:*

$$\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_1) = \sum_{i=1}^{N} \mathbb{E}\left[\mathrm{KL}(\mathbb{P}_0(y_i = \cdot \mid (x, y)_{\leq i-1}, x_i)), \mathbb{P}_1(y_i = \cdot \mid (x, y)_{\leq i-1}, x_i))\right]$$

Proof. We simplify $\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_1)$ as follows:

$$\begin{aligned}
\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_1) &= \sum_{(x,y)_{\leq N}} \mathbb{P}_0((x,y)_{\leq N}) \ln \frac{\mathbb{P}_0((x,y)_{\leq N})}{\mathbb{P}_0((x,y)_{\leq N})} \\
&= \sum_{(x,y)_{\leq N}} \mathbb{P}_0((x,y)_{\leq N}) \sum_{i=1}^{N} \ln \frac{\mathbb{P}_{\mathcal{A}}(x_i \mid (x,y)_{\leq i-1})}{\mathbb{P}_{\mathcal{A}}(x_i \mid (x,y)_{\leq i-1})} + \ln \frac{\mathbb{P}_0(y_i \mid (x,y)_{\leq i-1}, x_i)}{\mathbb{P}_1(y_i \mid (x,y)_{\leq i-1}, x_i)} \\
&= \sum_{i=1}^{N} \sum_{(x,y)_{\leq i}} \mathbb{P}_0((x,y)_{\leq i}) \ln \frac{\mathbb{P}_0(y_i \mid (x,y)_{\leq i-1}, x_i)}{\mathbb{P}_1(y_i \mid (x,y)_{\leq i-1}, x_i)} \\
&= \sum_{i=1}^{N} \sum_{(x,y)_{\leq i-1}, x_i} \mathbb{P}_0((x,y)_{\leq i-1}, x_i) \cdot \sum_{y_i} \mathbb{P}_0(y_i \mid (x,y)_{\leq i-1}, x_i) \ln \frac{\mathbb{P}_0(y_i \mid (x,y)_{\leq i-1}, x_i)}{\mathbb{P}_1(y_i \mid (x,y)_{\leq i-1}, x_i)} \\
&= \sum_{i=1}^{N} \mathbb{E}\left[\mathrm{KL}(\mathbb{P}_0(y_i = \cdot \mid (x,y)_{\leq i-1}, x_i)), \mathbb{P}_1(y_i = \cdot \mid (x,y)_{\leq i-1}, x_i))\right],
\end{aligned}$$

where the first equality is by the definition of KL divergence; the second equality is from the chain rule of conditional probability; the third equality is by canceling out the conditional probabilities of unlabeled examples given history, as we run the same algorithm $\mathcal{A}$ under two environments; the fourth equality is by the law of total probability; the fifth equality is again by the definition of the KL divergence. □

Fact 7 (KL divergence between Gaussians of the same mean)**.** *If $\mu \in \mathbb{R}$ and $\sigma_1, \sigma_2 > 0$, then,*

$$\mathrm{KL}\left(\mathrm{N}(\mu, \sigma_1^2), \mathrm{N}(\mu, \sigma_2^2))\right) = \frac{\sigma_1^2}{\sigma_2^2} - 1 + \ln \frac{\sigma_2^2}{\sigma_1^2}.$$

Fact 8 (Concentration of χ^2 random variables)**.** *For $d \geq 1$, $Z \sim \chi^2(d)$, and $\delta > 0$,*

$$\mathbb{P}\left(|Z - d| \leq 2\sqrt{d \ln \frac{1}{\delta}} + 2 \ln \frac{1}{\delta}\right) \geq 1 - \delta.$$

Specifically,

$$\mathbb{P}\left(|Z - d| \leq 10\sqrt{d}\right) \geq \frac{15}{16}.$$

The lemma below is a standard fact on normal distribution conditioned on affine subspaces; we include a proof here as we cannot find a reference.

Lemma 57. *Suppose $U = \left\{\theta \in \mathbb{R}^d : X\theta = y\right\}$ is an nonempty affine subspace of $\mathbb{R}^d$, where $X \in \mathbb{R}^{m \times d}$ has rows $x_1, \ldots, x_m \in \mathbb{R}^d$. Let $\dim(\mathrm{span}(x_1, \ldots, x_m)) = l$, and let $W \in \mathbb{R}^{d \times (d-l)}$ be a matrix whose columns form an orthonormal basis of $\mathrm{span}(x_1, \ldots, x_m)^{\perp}$. Consider $Z \sim \mathrm{N}(0, I_d)$; then,*

$$Z \mid \{Z \in U\} \sim \mathrm{N}(X^{\dagger} y, W W^{\top}).$$

Proof. Denote by $\hat{\theta} = X^{\dagger}y$ the least norm solution of equation $X\theta = y$. It is well-known that $\hat{\theta} \in \mathrm{span}(x_1, \ldots, x_m)$. As $U \neq \emptyset$, $X\hat{\theta} = y$. We now claim that U can be equivalently written as $\left\{\hat{\theta} + W\alpha : \alpha \in \mathbb{R}^{d-l}\right\}$:

1. On one hand, for all $\theta = \hat{\theta} + W\alpha$, $X\theta = X\hat{\theta} + XW\alpha = y + 0 = y$.
2. On the other hand, for every $\theta \in U$, as $X\theta = y$, we have $X(\theta - \hat{\theta}) = \mathbf{0}$, which implies that $\theta - \hat{\theta} \in \mathrm{span}(x_1, \ldots, x_m)^{\perp}$. Therefore, there exists some $\alpha \in \mathbb{R}^{d-l}$ such that $\theta = \hat{\theta} + W\alpha$.

Define $V \in \mathbb{R}^{d \times l}$ to be a matrix whose columns form an orthonormal basis of $\mathrm{span}(x_1, \ldots, x_m)$. We also claim that given a vector $z \in \mathbb{R}^d$, $z \in U \Leftrightarrow V^{\top}z = V^{\top}\hat{\theta}$:

1. If $z \in U$, by the previous claim, $z = \hat{\theta} + W\alpha$, and therefore $V^{\top}z = V^{\top}\hat{\theta} + V^{\top}W\alpha = V^{\top}\hat{\theta}$.
2. If $V^{\top}z = V^{\top}\hat{\theta}$, then note that $z = VV^{\top}z + WW^{\top}z = VV^{\top}\hat{\theta} + W(W^{\top}z) = \hat{\theta} + W(W^{\top}z)$, where the last equality follows from that $\hat{\theta} \in \mathrm{span}(x_1, \ldots, x_m)$. Taking $\alpha_z = W^{\top}z \in \mathbb{R}^{d-l}$, we have $z = \hat{\theta} + W\alpha_z$, implying that $z \in U$.

For the rest of the proof, let $\overset{d}{=}$ denote equality in distribution. Consider random variable $Z \overset{d}{=} \mathrm{N}(0, I_d)$. Let $\epsilon_V = V^{\top}Z, \epsilon_W = W^{\top}Z$. Now, note that the matrix $T = \begin{pmatrix} W^{\top} \\ V^{\top} \end{pmatrix} \in \mathbb{R}^{d \times d}$ is a orthonormal matrix,

$$\begin{pmatrix} \epsilon_V \\ \epsilon_W \end{pmatrix} = \begin{pmatrix} V^{\top} \\ W^{\top} \end{pmatrix} Z = TZ \overset{d}{=} \mathrm{N}(0, I_d),$$

Therefore, ϵ_V, ϵ_W are two independent, standard normal random variables with distributions $\mathrm{N}(0, I_l)$ and $\mathrm{N}(0, I_{d-l})$, respectively.

Note from the second claim that the event $\{Z \in U\}$ is equivalent to $\{\epsilon_V = V^{\top}\hat{\theta}\}$; therefore, $\epsilon_W \mid \{Z \in U\} \overset{d}{=} \mathrm{N}(0, I_{d-l})$. As a result,

$$Z \mid \{Z \in U\} \overset{d}{=} V\epsilon_V + W\epsilon_W \mid \{Z \in U\} \overset{d}{=} \hat{\theta} + W\epsilon_W \mid \{Z \in U\} \overset{d}{=} \mathrm{N}(X^{\dagger}y, WW^{\top}).$$

Part II

Machine Learning for and of Multi-Agent Systems

Chapter 7

Multi-agent Attribution via the Shapley Value

7.1 Introduction

Suppose we have a group of individuals out of which we need to select a team to perform a task. Besides maximizing team performance, we also wish to reward individuals fairly for their contributions to the team [208]. This general problem of *multi-agent attribution* is important in many real world contexts: choosing the best athletes for a sports team [186], choosing good workers for a project [246], choosing a subset of classifiers to use in an ensemble [242] etc. In this chapter we ask: how can we use data on past performance to figure out which individuals complement each other? And how can we *fairly* compensate team members accordingly?

Standard game theory (sometimes called 'non-cooperative' game theory) explicitly specifies actions, players, and utility functions. By contrast, cooperative game theory abstracts away from the 'rules of the game' and simply has as primitives the agents and the characteristic function (henceforth CF). The CF measures how much utility a coalition can create. Solution concepts in cooperative game theory have been developed to be 'fair' divisions of the total utility created by the coalition. These solution concepts can be viewed either as prescriptive (i.e. this is what an individual 'deserves' to get given their contribution) or predictive of what will happen in real world negotiations, where the intuition is that coalitions (or individuals) that don't receive fair compensations will opt to leave the game and simply transact amongst themselves.

These tools are useful for answering our main questions. The CF tells us how well a team will perform and the solution concepts will tell us how to divide value across individuals. For the purposes of this chapter, we consider one of the most prominent solution concepts: the Shapley Value (SV). However, there are two hurdles to overcome.

1. The CF is unknown to us, and is combinatorial in nature, thus requiring a sensible parametric model through which we can learn the CF from team performance data.
2. The SV requires an exponential number of operations to compute.

We introduce the cooperative game abstraction (CGA) model that simultaneously addresses *both* of these issues. In addition, CGA models are interpretable so as to aid analysts in understanding group synergy. Our main idea is motivated by a particular decomposition of the CF into an additive series of weights that capture m-way interaction between the n players for $m = 1, ..., n$. When we zero out terms of order order $k + 1$ and higher, this leaves behind an abstraction, a sketched version of the real cooperative game, which we refer to as a kth order CGA.

Our Contribution: To the best of our knowledge, we are the first to estimate characteristic functions with lossy *abstractions* [118] of the true characteristic function using parametric models, and bound the error of the estimated CF and SV. The second order variant of the CGA was first proposed in [92]. We generalize this work to study CGA models of *any order*. Our theoretical contributions are as follows: (i) sample complexity characterization of when a CGA model of order k (for any order k) is identifiable from data (ii) sensitivity analysis of how the estimation error of the characteristic function propagates into the downstream task of estimating the Shapley Value.

Empirically, we first validate the usefulness of CGAs in artificial RL environments, in which we can verify the predictions of CGAs on counterfactual teams. Then, we model real world data from the NBA, for which we do not have ground truth, using CGAs and show that its predictions are consistent with expert knowledge and various metrics of team strength and player value.

7.2 Related Work

Past works on ML for cooperative games have largely been theoretical and focus either on estimating the CF or estimating the Shapley Value directly without the CF. This differs from our goal, which is to model *both* with a provably good model that demonstrates sound performance on real world data. Indeed, our central premise is that the CF is unknown and needs to be learned from data. To the best of our knowledge, we are the first design a compact representation for the CF with *learning from samples in mind*: CGA not only has good learning theoretic properties, but also allows for fast SV computation.

Below, we describe related work in machine learning and cooperative game theory that assume the CF is unknown. In the appendix, we list additional, more distantly related work that assume the CF is known.

Modeling Characteristic Functions: As mentioned previously, [92] is the first to consider what we consider the second order variant of CGA. However, its focus was on the computational complexity of the *exact* computation of the Shapley Value. We consider the generalization of this representation to any order and are concerned with using lower rank CGA as an abstraction of complex games for *computational tractability*. As the low rank CGA is a *lossy* estimator of the true CF, we study and obtain theoretical bounds on the estimation errors of what we aim to compute: the CF and the SV.

A related work is [108] which proposes the MGH model for CFs. While the MGH model is like CGA in that both are complete representations, it contains nonlinearity that makes it harder to optimize and *interpret*. More crucially, the MPH model *does not* admit an easy computation of the SV. On the other hand, there are succinct representation models proposed for CFs that do allow the SVs to be readily computed. These are algebraic decision diagrams [24] and MC-nets [149], which represent CFs with a set of logical rules. However, the key drawback is that these models cannot be readily parameterized in tensor form and optimized using modern auto-grad toolkits, unlike the CGA.

Lastly, there has also been work in learning theory [33] that examines conditions under which a characteristic function can be PAC learned from samples. This work is concerned only with the theoretical learnability of the CF (and not the SV) for *certain classes* of cooperative games. By contrast, we study a concrete, parametric model that can approximate the CF of *any cooperative game*, study how approximation noise propagates into the SV and empirically verify that the model obtains good performance on real data.

Computing the Shapley Value: There has also been work that directly approximates the Shapley Value, without first learning the CF [35]. This differs from our goal in that we are interested in estimating *both* the Shapley and the CF. The latter is needed for applications such as counterfactual team performance prediction and optimal team formation, as we will demonstrate in the experiments.

Team Performance Analysis from Data: We note that all of the work cited above are theoretical and do not test their model on real world data. [186] is one empirical work that does. They model e-Sports team performances using a 2nd order CGA. Our work differs in that i) we generalize their model and study CGA models *any order* to obtain comprehensive sample complexity bounds ii) we are interested in *fair payoff assignment* in addition to team strength. To this end, we show that CGA allows for easy computation of SV and derive noise bounds for the estimated SV.

Abstraction in Games: Abstraction is an idea often used in game theory to make the computation of solution concepts such as the Nash Equilibrium (NE) tractable. One can efficiently solve for the NE of a abstracted game and lift the strategy to the original game. In non-cooperative game theory, the relationship between the quality of abstraction and the quality of the lifted strategy with respect to the original game has been heavily studied [169, 170, 176]. Our analysis characterizes the relationship between the abstractions and the solution concept, here being the Shapley Value, for any cooperative game. To the best of our knowledge, our work is the first to apply abstraction for computational tractability in the context of cooperative games.

7.3 Cooperative Game Theory Preliminaries

We begin with definitions in cooperative game theory.

Definition 37. *A **cooperative game** is defined by:*

1. *A set of agents $A = \{1, \dots, n\}$ with generic element i*
2. *A characteristic function $v : 2^A \to \mathbb{R}$*

We will refer to a subset of agents $C \in 2^A$ for which $v(C)$ measures how much utility a team C can create and divide amongst themselves. A 'fair division' of this value can be given according to the Shapley Value.

Definition 38. *The Shapley Value of an agent i with respect to team A is:*

$$\varphi_i(v) = \sum_{S \subseteq A \setminus i} \frac{|S|!(n - |S| - 1)!}{n!} (v(S \cup \{i\}) - v(S))$$

The Shapley Value is typically justified axiomatically. It is the unique division of total value that satisfies axioms of efficiency (all gains are distributed), symmetry (individuals with equal marginal contribution to all coalition get the same division), linearity (if two games are combined, the new division is the sum of the games' divisions), null player (players with 0 marginal contribution to any coalition receive 0 value). The Shapley value has been widely applied in ML, in domains such as cost-division [214, 273], feature importance [194], and data valuation [158] to name a few.

7.4 Cooperative Game Abstractions

7.4.1 Motivation

To model the characteristic function v, a natural set of abstractions can be derived from the fact that the characteristic function v can be decomposed into a sum of interaction terms across subsets of agents. In what follows, we will denote abstractions of v as $\hat{v}$.

Fact 9. *There exists a set of values ω_S for each $S = \{i_1, \dots, i_k\} \subseteq A$ such that any characteristic function can be decomposed into its interaction form where:*

$$v(C) = \sum_{k=1}^{|C|} \sum_{S \in 2_k^C} \omega_S. \tag{7.1}$$

where 2_k^C is the set of all coalitions of size k.

Note that Fact 9 implies that CGA is a *complete representation*: a CGA model of order n can model *any* set function. Its downside is that it has 2^n parameters to be learned from data. We may elect to truncate higher order terms and use an order k CGA model $\hat{v}$ to model v instead:

Definition 39. *A kth CGA model is parameterized by weight vector ω, which includes a weight ω_C for all coalitions C with $|C| \leq k$. The corresponding $v(C)$ is defined as in equation 7.1.*

A key property of CGA models is that the Shapley Value may be computed from a simple weighted sum of the CGA parameters.

Fact 10. *The Shapley Value of an individual i with respect to players A may be expressed as:*

$$\varphi_i(v) = \sum_{T \subseteq A \setminus \{i\}} \frac{1}{|T| + 1} \omega_{T \cup \{i\}}$$

7.4.2 Learning a CGA

We learn the CGA model from samples of coalition values from v. Given hypothesis class be $\mathcal{H}$, we perform empirical risk minimization (ERM) with criterion: $\min_{\hat{v} \in \mathcal{H}} \sum_{(C, v(C)) \in \mathcal{D}_P} (\hat{v}(C) - v(C))^2$

An important question that immediately follows is: when can a CGA model be identified from data? We define an exact identification notion as below:

Definition 40. *Suppose that we have a set of hypotheses $\mathcal{H}$ from which we will choose $\hat{v}$ via minimization of the criterion above. Suppose the dataset $\mathcal{D}_P$ is actually generated via a true $v^* \in \mathcal{H}$, we say that $\mathcal{D}_P$ identifies v^* if v^* is the unique minimizer of the criterion.*

Identification is an important question for three reasons. We are interested in the parameters of the model since they will be used to (i) predict the performance of unseen teams (ii) compute the Shapley value (iii) understand complementarity and substitutability between team members. If there are multiple sets of parameters consistent with the data, then none of the inferences we perform to answer those questions (e.g the marginal contribution of players) will be well defined.

We now give some sufficiency and necessity conditions on $\mathcal{D}_P$ for v^* to be identified exactly. These results generalize known sample complexity bounds from [254], expanding their bounds for only order 2 to *any* order k.

Theorem 34 (Sufficiency for Identification)**.** *Suppose $\mathcal{H}$ includes all k^{th} order CGAs and v^* is a k^{th} order CGA. If $\mathcal{D}_P$ include performances from all teams of at least k different sizes $s_1, \ldots, s_k \in [k, n-1]$, then $\mathcal{D}_P$ identifies v^*.*

Theorem 35 (Necessity for Identification)**.** *Suppose $\mathcal{H}$ includes all k^{th} order CGAs and v^* is a k^{th} order CGA. If $\mathcal{D}_P$ contains performances of teams of only $m < k$ different sizes, then $\mathcal{D}_P$ does not always identify v^*.*

We relegate the full proofs to the Appendix, as they are quite involved. To provide some intuition for the proof, in the sufficiency result, the argument uses induction to exploit structure in the matrix to arrive at the conditions under which its null space is empty, which implies that the matrix is full rank and the CGA is identifiable. In the complementary necessity result, we offer counterexamples that show even with all subsets of $m < k$ different sizes, the matrix corresponding to the system of linear equations may not be full rank, thus making the CGA model non-identifiable from data.

These results show that if the order $k = O(1)$, identification is possible with poly(n) samples and that if $k = O(n)$, the number of samples becomes exponential in n. Therefore, we suggest that practitioners should focus on the lowest order CGAs that they believe are suitable. For us, we find that low rank, second order CGAs demonstrate good performance in our experiments.

One consideration is that these bounds may be too pessimistic in requiring *exact* recovery of the true v. In the appendix, we provide sample bounds for identification of CGA under a PAC/PMAC [29] framework (Proposition 1). In particular, we have that under the looser, PMAC approximation notion, only $O(n)$ instead of $O(d_k)$ samples are needed for approximate estimation of most coalition values.

7.5 Approximate Shapley Values

With our approximation of the CF $\hat{v}$ in hand, we examine the fidelity of the SV computed from $\hat{v}$. We denote the approximated SV of player i as $\varphi_i(\hat{v})$ and the real SV $\varphi_i(v)$. As is typical in sensitivity analysis, we derive bounds relating the error in v to the error in the Shapley value.

These bounds may be of independent interest since often in ML applications v is stochastic. For instance, SV is widely used in interpretability literature [62, 74, 88, 116, 194], where v is taken to be the model performance. The model performance is typically stochastic, since it is a function of the random samples of data used to train the model and the randomness in the optimization, which can converge to differing local optima due to the nonconvexity of the losses e.g of deep models.

Let $\varphi(v)$ be the vector of Shapley Values. We start with a worst-case error bound for ℓ_2 when the adversary can choose how to distribute a fixed amount of error into v to construct $\hat{v}$.

Theorem 36. *The ℓ_2 norm of the estimation error of the Shapley Values is bounded by:*

$$\|\varphi(v) - \varphi(\hat{v})\|_2^2 \leq \frac{2}{n}\|v - \hat{v}\|_2^2$$

Though this result is *tight*, it assumes a non-smooth, adversarial distribution that places infinite density on the eigenvector corresponding to the largest singular value of the SV operator. Below, we consider average case bounds assuming that the error is of fixed norm and drawn from a *smooth* distribution; this type of assumption is often used in smooth analysis [127].

Theorem 37. *Assuming that $v - \hat{v}$ is drawn from distribution $\mathcal{D}_{B_r}$ with support equal to a sphere and smooth in that $\kappa_0 \leq \Pr_{\mathcal{D}_{B_r}}(x) \leq \kappa_1$ for any point x in its support, then:*

$$\mathbb{E}_{v-\hat{v}\sim\mathcal{D}_{B_r}}[\|\varphi(v) - \varphi(\hat{v})\|_2^2] \leq \frac{6}{n}\frac{\kappa_1}{\kappa_0}\frac{\|v - \hat{v}\|_2^2}{2^n}$$

We can generalize these results to any noise distribution thus:

Corollary 6. *Suppose noise $v - \hat{v} \sim \mathcal{D}_n$ is such that its conditional distribution satisfies $\kappa_0(r) \leq \Pr_{\mathcal{D}_n}(x | \|x\|_2^2 = r^2) \leq \kappa_1(r)$ for all r and x in $\mathcal{D}_n$'s support, then:*

$$\mathbb{E}_{v-\hat{v}\sim\mathcal{D}_n}[\|\varphi(v) - \varphi(\hat{v})\|_2^2] \leq \frac{6}{n}\mathbb{E}_r\left[\frac{\kappa_1(r)}{\kappa_0(r)}\left(\frac{r^2}{2^n}\right)\right]$$

Intuitively, this means that if the error $v - \hat{v}$ is on average spread out in that $\mathcal{D}_n$ is fairly smooth in expectation across concentric spheres in its support, then the ℓ_2 error of the Shapley value is small on average. Indeed, an astute reader may worry that only a $\frac{2}{n}$ reduction in the *aggregate* approximation error $\|v - \hat{v}\|_2^2$ is not large enough since $v - \hat{v} \in \mathbb{R}^{2^n}$. Theorem 37 and Corollary 6 show that the SV actually induces a $\frac{6}{n}\frac{\kappa_1}{\kappa_0}$ scaling of the *average* approximation error.

We also obtain analogous worst and average-case ℓ_1 bounds with scaling factors on the same order. Due to space constraints, please see Theorem 5 and 6 in the appendix for the results.

Lastly, we note that these bounds are general. In the appendix, we obtain a simple derivation of the CGA-specific bias, which can be plugged into these bounds for the SV bias. Note that bias in the estimation of the CF only arises due to model misspecification, i.e if order k is used to model a game of order r for $r > k$. This description covers *all cases* as any CF of a game necessarily corresponds to a CGA model of a certain order (Fact 9) and estimation error only arises due to a smaller order being specified. Certainly, we note that more refined bounds are a natural future extension to this work.

7.6 Experiments

7.6.1 Virtual Teams

We generate team performance data from the OpenAI particle environment [191] [1]. The task in this environment is team-based and requires cooperation: 3 agents are placed in a map and 3 landmarks are marked, and agents have a limited amount of time to reach the landmarks and are scored according to the minimum distance of any agent to any landmark. In addition, negative rewards are incurred for colliding with other agents. Thus, a team which can cooperate well is able to assign a single landmark per agent in real time and spread out to cover them without colliding with each other.

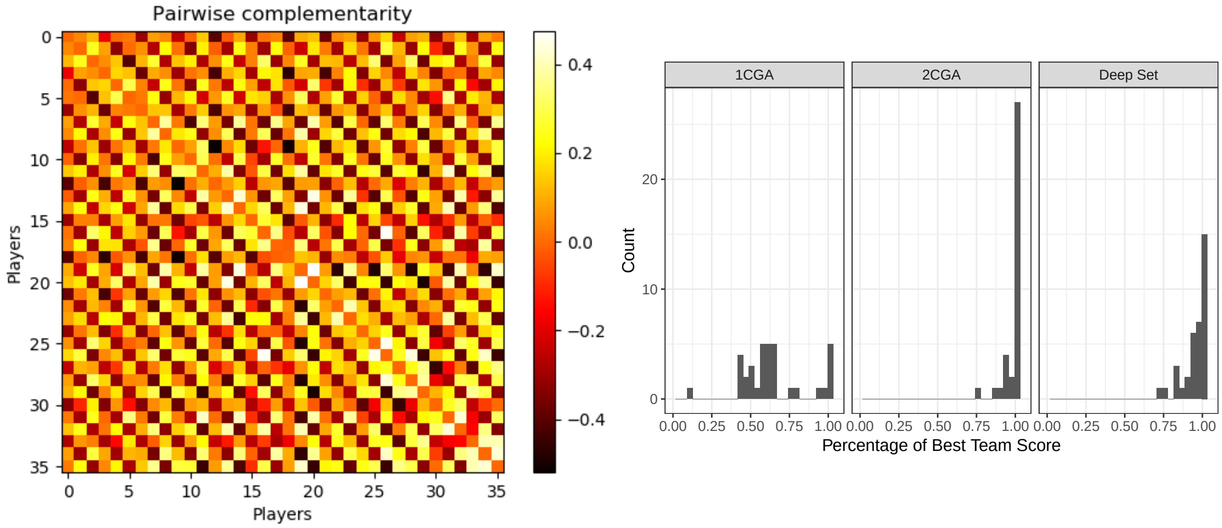


Figure 7.1: Left: Interaction matrix from second order CGA model. Players are clustered by original training team ($\{0, 1, 2\}$ trained together as did $\{3, 4, 5\}$, etc...). We see complex patterns of complementarity and substitutability as well as a clear replication of the well known fact that agents that train together can coordinate much better than agents which are trained separately - this can be seen in the figure by the strong complementarity in the diagonal blocks of size 3 compared to other 3×3 off-diagonal blocks. Right: Histograms of ratios of the score attained by the completed team chosen by the model normalized by the score of the actual best team containing a given initial agent.

We train 12 teams of agents (36 agents total) using the default algorithm and parameters from the OpenAI GitHub repo. We then evaluate all $\binom{36}{3} = 7140$ mixed teams of these agents evaluated over 100000 episodes. The train/validation/test split is 50/10/40. We fit a baseline first order (where team = sum of members) CGA and a second order CGA to predicting the final score of each team. We also compare to a more general, state of the art model for learning set functions, DeepSet [315], which is designed purely for prediction.

[1] `https://github.com/openai/multiagent-particle-envs`

Since DeepSet contains more parameters than the CGA model, we expect it to fit data better. However, unlike the CGA model, Deepset is (i) less easily identified due to the larger sample complexity needed (ii) *not readily interpretable* due to the non-linearity of ϕ (iii) and importantly, one cannot readily compute or estimate the Shapely. To compute the Shapley values exactly, one would have to first compute $v(C)$ for each coalition C, thus requiring 2^n feed-foward passes through the network. Even to approximate the Shapley value, it is known that $O(n \log n)$ evaluations of the model (network) are needed [158]. In contrast, to compute the Shapley with CGA, only one weighted sum of the CGA model parameters is needed and thus takes $O(1)$ number of evaluation.

Prediction: The first order CGA model achieves an test set MSE of of $.79$, the second order model achieves an order of magnitude smaller at $.07$. These results show that in this environment teams are not just sums of their parts. The DeepSet model achieves an MSE of $.042$, showing that we give up some predictive accuracy (but not that much) from using the simpler 2^{nd} order CGA. We emphasize that the goal of this experiment is *not* to find the most predictive model. Rather, it is to show that the much smaller, second order CGA model is roughly comparable to Deepset, all the while conferring the advantages of: 1. being interpretable 2. allowing easy computation of the Shapley Value.

Interpretability: To the first point, we visualize the learned matrix $\widehat{V}$ of the second order CGA in a heatmap (Figure 7.1) that allows us to discern players that complement/substitute each other.

Best Team Formation: For each of the 36 agents we have trained, we ask: what is the best set of 2 agents to add to them to make a team? More generally, this problem of optimal player addition is one often faced by real world sports teams, as they choose new players to draft or sign so as to further bolster their team performance. In this virtual setting, we can evaluate all possible additions to the team so as to gauge the predictive performance of our models.

In our setup, we restrict only to possible teammates which the original agent was not trained with. Figure 7.1 shows the histogram of ratios of the score attained by the completed team, which was selected by the model, normalized by the score of the actual best team. While the first-order CGA fails to construct good teams (since it does not consider any complementarities), the second order CGA and DeepSet model achieve more than $\sim 95\%$ of the possible value. Thus, the complementarity patterns learned via the 2nd order CGA are, in fact, important for this task. We also note that the second order CGA model outperforms DeepSet on this task.

7.6.2 Real World Sports Teams

We now consider a more complex, real world problem: predicting team performance in the NBA. We collect the last 6 seasons of NBA games (a total of 7380 games) from Kaggle along with the publicly available box scores [2]. Unlike in the dataset above, we do not observe absolute team performance, rather we only observe relative performance (who wins). We model matchup outcomes using the Bradley–Terry model. In particular, given the team strengths, the probability of team i winning in a match against team j as:

[2] https://www.kaggle.com/drgilermo/nba-players-stats

$$\Pr[w = 1 \mid \hat{v}, C_i, C_j] = \frac{\exp(\hat{v}(C_i))}{\exp(\hat{v}(C_i)) + \exp(\hat{v}(C_j))}$$

This gives us a well defined negative log likelihood (NLL) criterion of the data $\mathcal{D}$, which we optimize with respect to $\hat{v}$. We set each team in each game to be represented by its starting lineup (5 individuals). Then we learn $\hat{v}$ such that it minimizes the negative log likelihood using standard batch SGD with learning rate 0.001. Because basketball teams are of a fixed size (only one set of sizes), we use L2 regularization to choose one among the many possible set of models parameters.

As with the RL experiment above we compare a first order CGA, a second order CGA, and a DeepSet model. We split the dataset randomly into 80 percent training, 10 percent validation, and 10 percent test subsets. We set hyperparameters by optimizing the loss on the validation set.

7.6.2.1 Results

Prediction: How well does the CGA perform in this task? We begin by studying an imperfect metric: out-of-sample predictive performance. First, we see that the NBA performance can be fit fairly well with just a first order CGA - that is, we can think of most teams roughly as the sum of their parts. The first order CGA yields an out of sample mean negative log likelihood of $-.631$ which is slightly improved to $-.627$ under the second order CGA. We do also experiment with a third order CGA which did not improve over the second order CGA performance. This suggests that the second order is an apt choice for the abstraction. Finally, we observe that the DeepSet model is not able to outperform the CGA yielding an out of sample mean NLL of $-.63$.

Overall, we find that predictive accuracy is low, at only about $\sim 65\%$, as a result of the league being very competitive and teams being fairly evenly matched. Thus, predictive accuracy does not tell the whole story and is not the focus of the experiment. Note that the data at hand is observational and while players do move across teams and starting lineups change due to factors such as injuries, time in the season, etc... who plays with whom is highly correlated across years and starting lineups are endogenous (for example, a coach may not start one of their best players when playing a much weaker team to avoid risking injury). Thus, we cannot evaluate counterfactual teams. Instead, we supplement our the predictive analysis with analyses of the competing models to see if they are truly able to extract insights from the data consistent with NBA analytics experts.

Unseen teams: We consider teams the model has not seen: NBA All Star teams. During each season, fans and professional analysts vote to select ‘superstar’ teams of players that then play each other in an exhibition game, which is not included in our training data. We collect every All Star team from the time period spanning our training set and compare our second-order CGA model scores given to All Star teams with those of 1000 randomly generated, ‘average’ teams.

Recall that in the matchup datasets, the difference in scores between two teams is reflective of the probability that one team will win in a matchup. Thus, there is no natural zero point like when we are predicting v directly and we have chosen one particular normalization where the average score is zero. If our model does generalize well, it should predict that these all star teams are far above average despite never seeing this combination of players in the training set.

We also investigate whether the CGA has learned things about whole teams (e.g “the Cavaliers usually win”) and whether there is sufficient variation in starting lineups that we have learned the disentangled contributions of individual players (e.g the team's success is largely due to Lebron's brilliance). We investigate this by constructing synthetic 'same-team-All-Star' teams where we replace each player in a real All Star team with a randomly selected teammate from their real NBA team from that year.

Figure 7.2 shows the distribution of scores for randomly constructed teams with red lines representing predicted scores for the real All Star teams and blue lines for predicted scores for the 'same-team-All-Star' teams. These results show that the predictive performance of the CGA in win rate prediction comes from meaningful player-level assessment, not just that certain teams usually win (or lose).

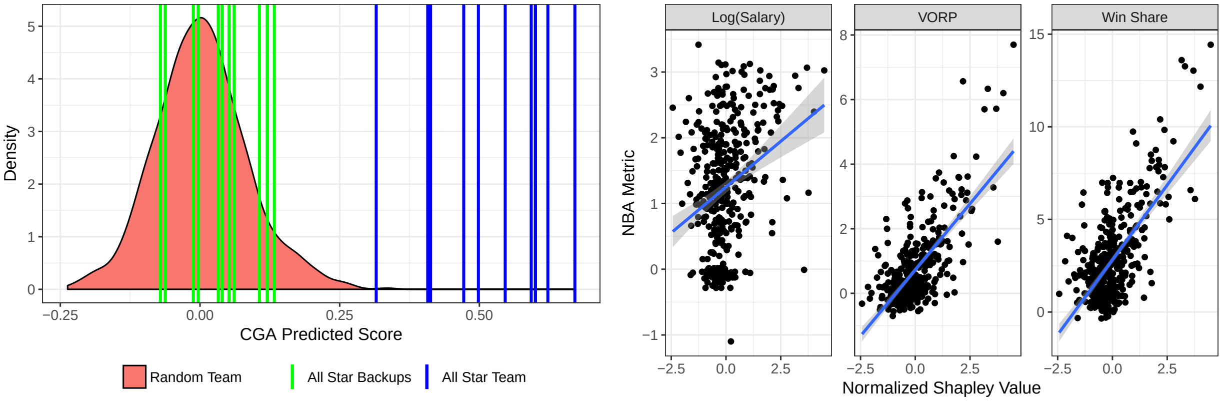


Figure 7.2: Left Panel: The second order CGA predicts that All Star Teams are far above the 99^{th} percentile of random teams. Replacing each All Star with their team-level replacement gives much worse teams. These results show that the predictive performance of the CGA in win rate prediction comes from player-level assessment, and not just memorization of certain teams usually winning or losing. Right Panel: Marginal contributions of individual NBA players, as measured by the Shapley Value from the second order CGA, correlate well with measures of player-level value add used by NBA analysts (VORP, Win-Share) as well as market-level value-add (salary).

Shapley Value as Individual Measure: So far we have asked whether our CGA captures team-level performance. We now turn to asking whether it captures individual-level marginal contribution. For each team, we compute the team members' Shapley Values *with respect to that team*. Since our dataset contains multiple years and individuals move across teams, we average an individual's computed Shapley values across all his teams. We correlate the Shapley Value based contribution scores with real world metrics used to evaluate basketball players' marginal contributions. We consider 3 measures commonly used in NBA analytics.

First, we look at the value-over-replacement metric player[3] (VORP). In basketball analytics, VORP tries to compute what would happen if the player were to be removed from the team and replaced by a random player in their position. Second, we look at win-share[4] (WS). Win-share tries to associate what percent of a team's performance can be attributed to a particular player. Finally, we use individual salaries, which are market measures of individual value add. Of course,

[3] https://www.basketball-reference.com/leaders/vorp_career.html
[4] https://www.basketball-reference.com/about/ws.html

a players' salary reflects much more than an individuals' contribution to team wins and losses (e.g their popularity, scarcity, etc...) and is extremely right tailed in the case of the NBA, so we consider its log. For each of these metrics, for each player, we average their values across the same years as our dataset.

Figure 7.2 plots CGA Shapley values against these measures. We see that there is a strong positive relationship between the CGA predicted Shapley value and other measures of individual contribution. Taken together, these results suggest that CGA indeed learns meaningful individual-level contribution measures, in a way that is consistent with expert knowledge.

Remark: overall, our experiments highlight the computational benefit of CGAs. In many cases like the NBA, team sizes are small relative to the number of players. We show that this structural prior can be encoded in a low rank CGA model, which does just as well (or better) than more complex, agnostic estimators like DeepSet (our main baseline), and is also interpretable to the benefit of users.

7.7 Conclusion

Cooperative game theory is a powerful set of tools. However, the CF is combinatorial and computing solution concepts like the Shapley is difficult. We introduce CGAs as a scalable, interpretable model for approximating the CF, and easily computing the SV. We provide a bevy of theoretical and empirical results so as to guide the application of CGA to model real world data.

Non-cooperative Game Theory has received much attention from the Machine Learning and AI community [54, 177, 178, 181, 261], while Cooperative Game Theory has been less explored. We believe that the intersection of Machine Learning and Cooperative Game Theory is rich with topics ranging from Multi-agent RL to Federated Learning. Our broader hope is that our work provides a springboard for future research in this area.

7.8 Appendix

7.8.1 Identification Theorem Proofs

Theorem 38 (Sufficiency for Identification)**.** *Suppose $\mathcal{H}$ includes all k^{th} order CGAs and v^* is a k^{th} order CGA. If $\mathcal{D}_P$ include performances from all teams of at least k different subset sizes $s_1, \ldots, s_k \in [k, n-1]$, then $\mathcal{D}_P$ identifies v^*.*

Proof. Let $\mathbf{w}$ be the first-through-k'th order weights we seek to learn, with the first n indices corresponding to ω_S such that $|S| = 1$, the next $\binom{n}{2}$ indices corresponding to $|S| = 2$, and so on up through the last $\binom{n}{k}$ terms corresponding to $|S| = k$. Let $\mathbf{v}$ be the corresponding coalitional values we observe.

Finding a k'th-order CGA corresponding to $\mathcal{D}$ can be formulated as finding a solution to $M\mathbf{w} = \mathbf{v}$, where matrix M is a matrix whose rows correspond to the data points and each entry in the matrix $\in \{0, 1\}$. For a given datapoint $(S, v(S))$, the corresponding row has ones in all entries corresponding to interaction terms ω_T such that $T \subseteq S$. Note that we only consider subset sizes $\geq k$, since subsets sizes smaller than k would not exhibit kth order interaction.

To show identifiability, it suffices to show that M has rank equal to the column size, since otherwise the null space is non-empty and there exist multiple $\mathbf{w}$ which satisfies the equation. Equivalently, a full rank matrix ensures that the optimization criterion is strictly convex and that the minimizer is unique. Define matrix M_{ntk} to be the submatrix consisting of all rows from all subsets of size t and columns corresponding only to that of the k'th order weights.

$$M = \begin{array}{l|ccc|}
 & \text{first order weight} & ... & k\text{'th order weight} \\
\text{rows from subsets of size } s_1 & M_{ns_1 1} & ... & M_{ns_1 k} \\
... & ... & ... & ... \\
\text{rows from subsets of size } s_k & M_{ns_k 1} & ... & M_{ns_k k}
\end{array}$$

We will now show that we can perform row reductions on the decomposition into submatrices, such that we end up with all zeroes below the antidiagonal.

First we note that every row in M_{nbk} is a linear combination of rows in M_{nak} for any $a < b$. Consider a row s_b corresponding to subset $\{i_1, ..., i_b\}$. We take all the rows in M_{nak} corresponding to subsets s' where $s' \subseteq \{i_1, ..., i_b\}$ and $|s'| = a$. We sum all $\binom{b}{a}$ of these rows, and denote this row s'_b. Looking at a particular kth order weight, say WLOG corresponding to $\{i_1, ..., i_k\} \subseteq \{i_1, ..., i_b\}$, there is a 1 in this column in s_b. This subset of size k shows up in $\binom{b-k}{a-k}$ subsets of size a. Therefore, the corresponding entry in row s'_b is $\binom{b-k}{a-k}$. And so, we can derive that $\binom{b-k}{a-k}^{-1} s'_b = s_b$ as they both have the same support: every subset of size k in $\{i_1, ..., i_b\}$ can be found in a subset of $\{i_1, ..., i_b\}$ of size a.

Thus we may use an appropriate multiple α of the first row to replace M_{ns_ik} with a zero for any $i > 1$. However, this changes the whole row, and so the j'th order term changes to $M_{ns_ij} - \alpha M_{ns1j}$. But by the same logic as for k, summing the l'th weights gives the same row scaled by $\binom{b-l}{a-l}$, and thus $M_{ns_ij} - \alpha M_{ns1j} = \left(1 - \frac{\binom{b-l}{a-l}}{\binom{b-k}{a-k}}\right) M_{ns_ij}$.

The above shows that we can perform row reduction using the first row of submatrices in order to put zeroes in the last column while retaining all submatrices in other columns (up to rescaling). But now we may apply this logic inductively, by considering only the submatrices corresponding to first through $k-1$'th order weights and rows from subsets of size s_2 or greater, and so on. We get that the matrix M looks as follows after row reduction:

$$\begin{array}{l|cccc|}
 & \text{first order weight} & ... & \text{(k-1)th order weight} & \text{kth order weight} \\
\text{rows from subsets of size } s_1 & M_{ns_1 1} & ... & M_{ns_1(k-1)} & M_{ns_1 k} \\
\text{rows from subsets of size } s_2 & M'_{ns_2 1} & ... & M_{ns_2(k-1)} & 0 \\
... & ... & ... & 0 & 0 \\
\text{rows from subsets of size } s_k & M_{ns_k 1} & ... & 0 & 0
\end{array}$$

where M'_{ns_21} denotes submatrices above the antidiagonal that have been rescaled (note that the first row does not need rescaling). It is then sufficient to show that $M_{ns_1k}, M_{ns_2(k-1)}..., M_{ns_k1}$ (note that each of these submatrices has more rows than columns) are all full rank to show M is full rank.

To do this, we first prove a lemma.

Lemma 58. *When $t \geq k$ and $n = t + k$, the matrix M_{ntk} is full rank.*

Proof. We will proceed by induction on k.

Base case ($k = 1$): Since $k = 1$ each row corresponds to an all-one row with a single zero for the agent left out. Since we have such a row for each agent that can be left out, we get n linearly-independent rows.

Thus the matrix is full rank.

Induction step ($k > 1$):

Assuming this statement holds for orders $1, \ldots, k - 1$. We will prove the statement for when the order is k. To do this we will use induction on n:

Base case ($n = 2k$): $n = 2k \Rightarrow t = k$, and so M_{ntk} is the identity matrix and is thus full rank.

Induction step ($n > 2k$): Assume the matrix is full rank for when number of players is equal to $2k, ..., n$. To prove the matrix is full rank for $n + 1$, consider the following decomposition of $M_{(n+1)(n+1-k)k}$.

$$\begin{array}{rc} & \begin{array}{cc} \text{weights of subsets including 1} & \text{weights of excluding including 1} \end{array} \\ \begin{array}{r} \text{subsets including 1} \\ \text{subsets excluding 1} \end{array} & \left(\begin{array}{cc} A & B \\ 0 & C \end{array} \right) \end{array}$$

Observe that matrix B corresponds to $M_{n(n-k)k}$ and is thus full rank by induction hypothesis. This means we can use linear combinations of rows of B to reduce rows in C. In particular, we can performs row reductions such that we replace C with zeroes: For each row in C corresponding to a team of size of $n + 1 - k$ selected from $[2, ..., n]$, we consider all $n - k$ subsets of this team in B and sum them. For any subset of this team of size k, then we see that it shows up in the sum: $\binom{n+1-k-(k)}{n-k-(k)} = n + 1 - 2k$ times. Therefore, the sum is $n + 1 - 2k$ times the row in C.

Moreover, let D be the $n + 1 - k$ rows from A summed together when performing the row reduction, let the resultant matrix be D. The reduced matrix looks like:

$$M_{(n+1)(n+1-k)k} =$$

$$\begin{array}{rc} & \begin{array}{cc} \text{weights of subsets including 1} & \text{weights of excluding including 1} \end{array} \\ \begin{array}{r} \text{subsets including 1} \\ \text{subsets excluding 1} \end{array} & \left(\begin{array}{cc} A & B \\ D & 0 \end{array} \right) \end{array}$$

Then, we observe that D corresponds to a scaled version of $M_{n(n+1-k)(k-1)}$, which is full rank by the inductive assumption. The scaling factor is calculated as follows: For a subset $\{1, ..., k\}$, the $k - 1$ elements show up in the $n + 1 - k$ subset row of C, then shows up in $\binom{n+1-k-(k-1)}{n-k-(k-1)} = n + 2 - 2k$ of the $n - k$ subsets. And so, D is a $-\frac{n+2-2k}{n+1-2k}$ scaled version of $M_{n(n+1-k)(k-1)}$.

B remains unchanged after the row reduction and is full rank and thus the whole matrix is full rank. □

With this lemma in hand we can return to the main proof. To complete the proof, we will show that for any k, any $n \geq 2k$ and any $t \in [k, n-k]$, M_{ntk} is full rank.

Note that $t \in [k, n-k] \Rightarrow \binom{n}{t} \geq \binom{n}{k}$ (as otherwise number of rows is already fewer than number of columns and the matrix will have rank less than column size).

We will use induction on k.

Base case ($k = 1$): by Lemma 58, the matrix is full rank when $k = 1$ for any team size t and number of players n.

Induction step ($k > 1$): assumes this holds for orders $1, ..., k-1$ and any t and n. For order k, fix some $t \geq k$, we will show the matrix is full rank for all $n \geq t + k$ by induction on n. For the base case $n = t + k$ the matrix is full rank by Lemma 58. For the induction step assume $n > t + k$: assume the matrix is full rank when the number of players is in $\{t + k, ..., n-1\}$. Now when the number of players is n, we may decompose the matrix into columns corresponding to weights of k-size subsets containing player 1, and rows into teams including or excluding player 1.

$$M_{ntk}=$$

$$\begin{array}{rc} & \begin{array}{cc} \text{weights of subsets including 1} & \text{weights of subsets excluding 1} \end{array} \\ \begin{array}{r} \text{subsets including 1} \\ \text{subsets excluding 1} \end{array} & \left(\begin{array}{cc} U_1 & U_3 \\ 0 & U_2 \end{array}\right) \end{array}$$

In doing so, we first observe that U_1 is exactly $M_{(n-1)(t-1)(k-1)}$ and is full rank from the induction hypothesis on k. Secondly, $U_2 = M_{(n-1)tk}$ and is thus full rank by the induction hypothesis on n. Therefore the matrix M_{ntk} is full rank which concludes the inductive step on n. But this also concludes the inductive step on $k > 1$, and thus we we get that all M_{ntk} along the antidiagonal of M are full rank. It follows that M is full rank, and thus $M\mathbf{w} = \mathbf{v}$ has a unique solution.

Finally, because we are choosing k subset sizes from $[k, n-1]$, it's easy to see that if we sort subset size s by $\binom{n}{s}$, then the jth subset size in this sorted order s_i is such that $\binom{n}{s_j} \geq \binom{n}{k-j+1}$, which means the above condition applies. □

Theorem 39 (Necessity for Identification)**.** *Suppose $\mathcal{H}$ includes all k^{th} order CGAs and v^* is a k^{th} order CGA. If $\mathcal{D}_P$ contains performances of teams of only $m < k$ different sizes, then $\mathcal{D}_P$ does not always identify v^*.*

Proof. We will provide an instance when $k = 2$ such that v^* is not identified. In that case $m = 1$, and we may pick teams of size $n-1$. That gives us $\binom{n}{n-1} = n$ rows which is fewer than the number of columns $\binom{n}{2} + \binom{n}{1}$. Thus there will be more than one solution.

Moreover, the conditions specified in Theorem 1 are also tight in the sense that: if we allowed $m = k$ subset sizes, but over a wider interval, then $\mathcal{D}$ does not always identify v^*. To see this, consider $k = 2$ again. Widening the interval means the inclusion of either subset size $k-1$ or n.

If we can pick $k-1$, consider $m = 2$ subset sizes $k-1$ and $n-1$, which together gives $\binom{n}{1} + \binom{n}{n-1}$ rows, which is fewer than the number of columns $\binom{n}{2} + \binom{n}{1}$.

If we can pick n, consider $m = 2$ subset sizes $n-1$ and n, which together gives $\binom{n}{n-1} + \binom{n}{n}$ rows, which is fewer than the number of columns $\binom{n}{2} + \binom{n}{1}$.

□

7.8.2 PAC Analysis

Another natural paradigm through which we may analyze sample complexity of learning a CGA is the PAC framework. Before we proceed, a word about why PAC bounds are not our main focus for sample complexity. One drawback of PAC bounds we considered is that it is only with high probability that *most* coalition values are well approximated. Therefore, it could still be that there is one $\hat{v}(S)$ that is arbitrarily off. Thus, the resultant estimated Shapley value will inherit this large bias. Since we hope to use the estimated Shapley Value for fair credit assignment in practice, we opt for what may be considered more "pessimistic", exact identification guarantees similar to those in [254].

Below, we provide two results based on PAC and PMAC notions of approximation. We prove the result assuming that we have correct CGA order specification. The result follows similarly when a higher order than that of the true CF is specified.

Consider a random sample S of m $(C, v(C))$ data points with C uniformly sampled from 2^A. There are at most m distinct coalitional values in that sample. Call them $\mathbf{v}_{\widehat{\mathbf{S}}}$. We will solve $M^{nk}_{\widehat{S}}\hat{\omega} = \mathbf{v}_{\widehat{\mathbf{S}}}$ where $M^{nk}_{\widehat{S}}$ denotes the matrix consisting of all rows corresponding to coalitions in $\widehat{S}$. This is feasible since there exist ω s.t $M^{nk}\omega = \mathbf{v}$. Note that this step assuming feasibility relies on the CGA model being of order k or higher; if not, $M^{nk}_{\widehat{S}}\hat{\omega} = \mathbf{v}_{\widehat{\mathbf{S}}}$ may not be feasible.

In both parts of the proposition below, we will appeal to uniform convergence results to show that this construction yields a $\hat{\omega}$ and the corresponding $\hat{\mathbf{v}}$ such that it approximates $\mathbf{v}$ with high probability. In all the sample complexity results that follow, let c denote a generic constant.

Proposition 39. *Suppose* $\mathbf{v}$ *is a kth order CGA model with parameter vector* ω *of bounded* ℓ_1 *norm. Then, with a set* $\widehat{S}$ *of* $(C, v(C))$ *data points of size* $m \geq c\left(\frac{d_k+\log(1/\Delta)}{\delta^2}\right)$ *uniformly sampled from* 2^A*, we may compute* $\hat{\omega}$ *and its corresponding* $\hat{\mathbf{v}}$ *as above such that, with probability at least* $1-\Delta$ *over the samples* $\widehat{S}$*:*

$$\Pr_{C\sim 2^A}[\hat{v}(C) = v(C)] \geq 1-\delta$$

Proof. The proof is motivated by the observation that ω may be viewed as a linear classifier with dimension d_k. Indeed, if ω is the true weight, then $M^{nk}\omega = \mathbf{v}$, which is equivalent to:

$$[-M^{nk}_C, v(C)]^T[\omega, 1] \geq 0 \text{ and } [M^{nk}_C, -v(C)]^T[\omega, 1] \geq 0 \text{ for all C}$$

where M^{nk}_C denotes the row of M_{nk} corresponding to coalition C and $[\mathbf{a}, \mathbf{b}], \mathbf{a} \in \mathbb{R}^n, \mathbf{b} \in \mathbb{R}^m$ denotes the $n+m$-dimensional vector obtained by concatenation of $\mathbf{a}$ and $\mathbf{b}$. Thus, if we define a classification task with $2 \cdot 2^N$ data points that have features $[-M^{nk}_C, v(C)]$, $[M^{nk}_C, -v(C)]$ and labels 1 for all the points, we know there exists a classifier $f(\mathbf{x}) = \text{sign}([\omega, 1]^T\mathbf{x})$ which achieves zero loss; here we take the sign of 0 to be 1.

Define data distribution $\mathcal{D}$ to be the uniform distribution over these $2 \cdot 2^N$ data points. A draw of size m from $\mathcal{D}$ may be simulated by sampling coalitions from the uniform distribution over 2^A and then for each chosen coalition C, randomly choosing between $[-M^{nk}_C, v(C)]$ and $[M^{nk}_C, -v(C)]$ with equal probability.

Now, we are ready to prove that the $\hat{v}$ satisfies the statement in Proposition 39. To do this, we use the uniform convergence result below (Theorem 6.8 from [256]):

Lemma 59. *Let $\mathcal{H}$ be a hypothesis class for the classifier, and let f be the true underlying classifier. If $\mathcal{H}$ has VC-dimension d, then with*

$$m \geq c\left(\frac{d + \log\left(\frac{1}{\Delta}\right)}{\delta^2}\right)$$

i.i.d data points $\mathbf{x}_1, ..., \mathbf{x}_m \sim \mathcal{D}$,

$$\delta \geq \left| \Pr_{\mathbf{x} \sim D}[h(\mathbf{x}) \neq f(\mathbf{x})] - \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{h(\mathbf{x}^i) \neq f(\mathbf{x}^i)} \right|$$

for all $h \in \mathcal{H}$ and with probability $1 - \Delta$ over the sampled data points.

By construction, the classifier defined by $\hat{\omega}$, $h(x) = \text{sign}([\hat{\omega}, 1]^T x))$, achieves zero empirical risk on $\hat{S}$ since $h(\mathbf{x}^i) = 1 = f(\mathbf{x}^i)$. So, we apply the uniform convergence result Lemma 59 with $\delta/2$ to get that with probability $1 - \Delta$ over the sampled data points $\mathbf{x}$ from $\mathcal{D}$:

$$\begin{aligned}
\frac{\delta}{2} &\geq \Pr_{\mathbf{x} \sim D}[h(\mathbf{x}) \neq f(\mathbf{x})] \\
&= \Pr_{\mathbf{x} \sim D}[[\hat{\omega}, 1]^T \mathbf{x} < 0)] \\
&= \frac{1}{2}\Pr_{C \sim 2^A}[M_C^{nk}\hat{\omega} > v_C] + \frac{1}{2}\Pr_{C \sim 2^A}[M_C^{nk}\hat{\omega} < v_C] \\
&= \frac{1}{2}\Big(1 - \Pr_{C \sim 2^A}[M_C^{nk}\hat{\omega} = v_C]\Big)
\end{aligned}$$

Therefore, the guarantee for $\hat{\omega}$ over distribution $\mathcal{D}$ translates to the guarantee over the uniform distribution 2^A that $\hat{v} = M^{nk}\hat{\omega}$ can overpredict or underpredict for at most δ percent of all coalitions.

To finish, we note that $[\omega, 1]$ belongs to the hypothesis class of linear classifiers of dimension $d_k + 1$, which is known to have VC Dimension $d_k + 1$. So $\mathcal{H} = \{[\omega, 1] \mid \omega \in \mathbb{R}^{d_k}\}$ has VC dimension $d \leq d_k + 1$. And so, our sample complexity needed for $\hat{\omega}$ to attain small generalization risk using Lemma 59 is $O(\frac{d_k + \log(1/\Delta)}{\delta^2})$.

□

We remark that the sample complexity needed is on the same order as that shown by Theorem 1 in section 7.8.1

Next, we provide a PMAC-like guarantee [29] with much smaller sample complexity.

Proposition 40. *With samples of size $m \geq c\left(\frac{\log(d_k) + \log(1/\Delta)}{\epsilon^2\delta^2}\right)$ uniformly sampled from 2^A, we may compute $\hat{\omega}$ and its corresponding $\hat{\mathbf{v}}$ as above such that, with probability at least $1 - \Delta$ over the samples:*

$$\Pr_{C \sim 2^A}\left[(1 - \epsilon)\hat{v}(C) \leq v(C) \leq (1 + \epsilon)\hat{v}(C)\right] \geq 1 - \delta$$

Proof. The proof follows from combining two known theorems adapted to our setting.

The left hand side of the probabilistic guarantee follows from a straightforward adaptation of the proof of Theorem 5 in [35]. In particular, the only tweak to the proof is that the features of the data points are to be instantiated as $M_C^{nk}/v(C)$ instead of $\mathbb{1}_C/v(C)$. Since ω is bounded by our assumption, we do not need to bound it in terms of values of $v(C)$'s as is done in the proof of [35]. We note the loss function would then be defined as $\ell(\omega, (M_C^{nk}/v(C), y)) = [\frac{M_C^{nk}\omega}{v(C)} - 1]_+$ and $\hat{\omega}$ achieves zero empirical loss because $M_{\widehat{S}}^{nk}\hat{\omega} = \mathbf{v}_{\widehat{\mathbf{S}}} \Rightarrow M_C^{nk}\hat{\omega} = v_C \Rightarrow \frac{M_C^{nk}\hat{\omega}}{v(C)} - 1 = 0$ for all $C \in S$. Altogether, we may arrive at the statement below:

With a set of $m \geq c\left(\frac{\log(d_k)+\log(1/\Delta)}{\epsilon^2\delta^2}\right)$ *coalitions uniformly sampled from* 2^A, $\hat{\omega}$ *constructed as above is such that:*

$$\Pr_{C\sim 2^A}\left[(1-\epsilon)M_C^{nk}\hat{\omega} \leq v(C)\right] \geq 1-\delta$$

with probability at least $1-\Delta$ *over the samples.*

The right hand side follows from a related theorem, Theorem 2 in [305] with the same change in the data features. Again, we can verify that $\hat{\omega}$ achieves zero empirical loss:

With a set of $m \geq c\left(\frac{\log(d_k)+\log(1/\Delta)}{\epsilon^2\delta^2}\right)$ *coalitions uniformly sampled from* 2^A, $\hat{\omega}$ *constructed as above is such that:*

$$\Pr_{C\sim 2^A}[v(C) \leq (1+\epsilon)M_C^{nk}\hat{\mathbf{w}}] \geq 1-\delta$$

with probability at least $1-\Delta$ *over the samples.*

With this, we can initialize both theorems with $\Delta/2$ and $\delta/2$. We first union bound over the random draw of $m-$ size samples to conclude that with probability $\geq 1-\Delta$, both inequalities hold for $\hat{\omega}$, meaning that by union bound again for the random draw of C over 2^A:

$$\Pr_{C\sim 2^A}[(1-\epsilon)M_C^{nk}\hat{\mathbf{w}} \leq v(C) \leq (1+\epsilon)M_C^{nk}\hat{\mathbf{w}}] \geq 1-\delta$$

□

In summary, this means that under an even looser definition of approximability of the CGA model, the sample complexity needed is much smaller: only $O(\log(d_k))$ number of points are needed. Since $d_k \leq 2^n$, this means at most $O(n)$ samples are needed to estimate *most* of the coalition values *approximately* with high probability.

Remark: more generally, we may obtain the above two guarantees under the same sample complexity for any setting where we are looking to estimate solutions $\mathbf{x}$ to large scale linear programs $A\mathbf{x} = \mathbf{b}$, knowing apriori that $\|x\|_1$ is bounded. In such cases, we may obtain a PAC and PMAC-like result by computing $\hat{\mathbf{x}}$ from randomly sampled constraints $\mathbf{a_i}^T\mathbf{x} = b_i$. Notice here that $A \in \mathbb{R}^{2^n \times d_k}$ and the PMAC notion avoids needing the exponential sample complexity that is required to construct $\mathbf{b}$ to compute an exact solution.

This result may be of independent interest.

7.8.3 Shapley Noise Bound Theorem Proofs

Theorem 40 (Shapley noise L2 bound)**.** *The L2 norm of the estimation error of the Shapley values is bounded by:*

$$\sum_{i=1}^{n} \big(\varphi_i(v) - \varphi_i(\hat{v})\big)^2 \leq \frac{2}{n} \sum_{C \in 2^A} \big(v(C) - \hat{v}(C)\big)^2 \tag{7.2}$$

Proof. First we observe that the Shapley value is a linear map $\mathbb{R}^{2^n} \to \mathbb{R}^n$ taking v to $\varphi(v)$. We may describe this map with matrix $S_n \in \mathbb{R}^{n \times 2^n}$ where n is the number of players in the cooperative game. Our work extends a line of work that studies properties of S_n, including [37] that studies its nullspace.

We have that:

$$||\varphi(v) - \varphi(\hat{v})||_2 = ||S_n v - S_n \hat{v}||_2 \leq ||S_n||_{op} ||v - \hat{v}||_2$$

It suffices then to obtain the operator norm of S_n. We know that $||S_n||_{op} = \sqrt{\sigma_{\max}(S_n^T S_n)}$. $S_n^T S_n$ is complicated to analyze, so we opt to analyze $\sigma_{\max}(S_n S_n^T)$ since we know that the nonzero eigenvalues of $S_n^T S_n$ are the same as those of $S_n S_n^T$. $S_n S_n^T$ has nice structure in that all its off-diagonal entries are the same and all its diagonal entries are the same.

Take the ith row of S_n, $(S_n)_i$, we know that the entry in this row corresponding to subset S is:

1. $\frac{1}{n}\binom{n-1}{|S|-1}^{-1}$ if $i \in S$
2. $-\frac{1}{n}\binom{n-1}{|S|}^{-1}$ if $i \notin S$

Therefore, let d_1 denote its diagonal entries, then:

$$\begin{aligned} d_1 &= (S_n)_i^T (S_n)_i \\ &= \sum_{S \in 2^{[n]}, i \in S} (\frac{1}{n}\binom{n-1}{|S|-1}^{-1})^2 + \sum_{S \in 2^{[n]}, i \notin S} (-\frac{1}{n}\binom{n-1}{|S|}^{-1})^2 \\ &= \frac{1}{n^2}\sum_{k=1}^{n} \binom{n-1}{k-1}\binom{n-1}{k-1}^{-2} + \frac{1}{n^2}\sum_{k=0}^{n-1} \binom{n-1}{k}\binom{n-1}{k}^{-2} \end{aligned}$$

Let d_2 denote its off-diagonal entries. Consider the i, jth entry of $S_n S_n^T$, we can characterize the weights in the dot product as follows:

1. $(\frac{1}{n}\binom{n-1}{|S|-1}^{-1})^2$ if $i, j \in S$
2. $(\frac{1}{n}\binom{n-1}{|S|-1}^{-1})(-\frac{1}{n}\binom{n-1}{|S|}^{-1})$ if $i \in S, j \notin S$
3. $(-\frac{1}{n}\binom{n-1}{|S|}^{-1})(\frac{1}{n}\binom{n-1}{|S|-1}^{-1})$ if $i \notin S, j \in S$
4. $(-\frac{1}{n}\binom{n-1}{|S|}^{-1})^2$ if $i, j \notin S$

Therefore, when we sum these together:

$$
\begin{aligned}
d_2 &= (S_n)_i^T (S_n)_j \\
&= \sum_{S\in 2^{[n]}, i,j\in S} (\frac{1}{n}\binom{n-1}{|S|-1}^{-1})^2 - \sum_{S\in 2^{[n]}, i\in S, j\notin S} (\frac{1}{n}\binom{n-1}{|S|-1}^{-1})(-\frac{1}{n}\binom{n-1}{|S|}^{-1}) \\
&\qquad - \sum_{S\in 2^{[n]}, i\notin S, j\in S} (-\frac{1}{n}\binom{n-1}{|S|}^{-1})(\frac{1}{n}\binom{n-1}{|S|-1}^{-1}) + \sum_{S\in 2^{[n]}, i,j\notin S} (-\frac{1}{n}\binom{n-1}{|S|}^{-1})^2 \\
&= \sum_{k=2}^{n} \binom{n-2}{k-2}\binom{n-1}{k-1}^{-2} - 2\sum_{k=1}^{n-1} \binom{n-2}{k-1}\binom{n-1}{k}^{-1}\binom{n-1}{k-1}^{-1} + \sum_{k=0}^{n-2} \binom{n-2}{k}\binom{n-1}{k}^{-2}
\end{aligned}
$$

It's easy to check that $d_1 > d_2$ since $d_2 = (S_n)_i^T(S_n)_j \leq ||(S_n)_i||_2||(S_n)_j||_2 = (S_n)_i^T(S_n)_i = d_1$.

And so, we may write:

$$S_n S_n^T = (d_1 - d_2)I_n + d_2 1_n$$

where 1_n is the all ones matrix.

This allows us to characterize all the eigenvalues of $S_n S_n^T$ and in particular the biggest one.

If the SVD of $1_n = UDU^T$, then we know that D is a diagonal matrix with one entry being n as this is an eigenvalue of 1_n and the rest being 0 since 1_n is only rank 1. And so,

$$S_n S_n^T = U[(d_1 - d_2)I_n + d_2 D]U^T$$

This means that the top eigenvalue is $d_1 - d_2 + n \cdot d_2$ and the rest are all $d_1 - d_2$.

Evaluating $d_1 - d_2 + n \cdot d_2 = d_1 + (n-1)d_2$:

$$
\begin{aligned}
&d_1 + (n-1)d_2 \\
&= \frac{1}{n^2}(\sum_{k=2}^{n-2} \binom{n-1}{k-1}^{-1} + \binom{n-1}{k}^{-1} \\
&\qquad + (n-1)[\frac{k-1}{n-1}\binom{n-1}{k-1}^{-1} - 2\frac{k}{n-1}\binom{n-1}{k-1}^{-1} + \binom{n-2}{k}\binom{n-1}{k}^{-2}]) + \frac{1}{n^2}r \\
&= \frac{1}{n^2}((\sum_{k=2}^{n-2} k\binom{n-1}{k-1}^{-1} + \binom{n-1}{k}^{-1} - 2k\binom{n-1}{k-1}^{-1} + (n-1)\binom{n-2}{k}\binom{n-1}{k}^{-2}) + \frac{1}{n^2}r \\
&= \frac{1}{n^2}((\sum_{k=2}^{n-2} -k\binom{n-1}{k-1}^{-1} + \binom{n-1}{k}^{-1} + (n-1-k)\binom{n-1}{k}^{-1}) + \frac{1}{n^2}r \\
&= \frac{1}{n^2}((\sum_{k=2}^{n-2} -\frac{k!(n-k)!}{(n-1)!} + (n-k)\frac{k!(n-1-k)!}{(n-1)!}) + \frac{1}{n^2}r \\
&= \frac{1}{n^2}r
\end{aligned}
$$

It just remains to evaluate r which are the residual terms from the sums, they are:

$$\begin{aligned}
r &= [1 + 1 + \frac{1}{n-1}] + [1 + \frac{1}{n-1} + 1] \text{ (from the two sums in } d_1\text{)} \\
&\quad + (n-1)(([1 + \frac{n-2}{(n-1)^2}] - 2[\frac{1}{n-1} + \frac{1}{n-1}] + [1 + \frac{(n-2)}{(n-1)^2}]) \text{ (from the three sums in } d_2\text{)} \\
&= 4 + \frac{2}{n-1} + 2n - 2 - 4 + \frac{2(n-2)}{n-1} \\
&= 2n
\end{aligned}$$

To summarize, we get that:

$$\sigma_{\max}(S_n S_n^T) = d_1 + (n-1)d_2 = \frac{1}{n^2}2n = \frac{2}{n}$$

$$\Rightarrow ||S_n||_{op} = \sqrt{\sigma_{\max}(S_n^T S_n)} = \sqrt{\sigma_{\max}(S_n S_n^T)} = \sqrt{\frac{2}{n}}$$

which proves that $||\varphi(v) - \varphi(\hat{v})||_2 \leq \sqrt{\frac{2}{n}}||v - \hat{v}||_2$ (7.2), as desired.

□

The above is a worst case analysis by computing the largest singular value of the Shapley matrix. It turns out, most singular values of the Shapley matrix are very small and won't lead to a large amplification of the noise in the characteristic function.

We perform average case analysis by assuming that the error in the characteristic function is drawn uniformly from a smooth distribution, which is not very "peaky" anywhere, over all noise $v - \hat{v}$ with the same L2 norm.

Lemma 60. *Let $\mathcal{D}_{B_r}$ be a distribution with support equal to a sphere with radius* r *and smooth in that $\kappa_0 \leq \Pr_{\mathcal{D}_{B_r}}(\mathbf{x}) \leq \kappa_1$ for any $\mathbf{x}$ in its support. Consider any matrix $A \in \mathbb{R}^{m_1 \times m_2}$:*

$$\mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{B_r}}[\|A\mathbf{x}\|_2^2] \leq \frac{\kappa_1}{\kappa_0} \frac{\operatorname{Tr}(A^T A)}{m_2} \cdot r^2$$

Proof. Since $A^T A$ is symmetric and thus diagonalizable, consider its m_2 orthonormal eigenvectors $\mathbf{u_1}, .., \mathbf{u_{m_2}}$. We know that $\mathbf{u_1}, .., \mathbf{u_{m_2}}$ forms a basis of $\mathbb{R}^{m_2}$ and we can then write any $\mathbf{x}$ in the support of $\mathcal{D}_{B_r}$ as $\sum_{j=1}^{m_2} \alpha_j \mathbf{u_j}$. Moreover,

$$r^2 = \|\mathbf{x}\|^2 = (\sum_{j=1}^{m_2} \alpha_j \mathbf{u_j})^T (\sum_{j=1}^{m_2} \alpha_j \mathbf{u_j}) = \sum_{j=1}^{m_2} \alpha_j^2$$

since $\mathbf{u_j}^T \mathbf{u_i} = 0$ for $i \neq j$ and $\|\mathbf{u_j}\|_2^2 = 1$.

Define $\mathcal{D}'_{B_r}$ to be the distribution over α that corresponds to each $\mathbf{x}$ drawn from $\mathcal{D}_{B_r}$ and set $S_{D'}$ be its support (which may be characterized as a m_2 dimensional standard simplex as defined by $(\alpha_1^2/r^2, ..., \alpha_{m_2}^2/r^2)$. We abuse notation in letting $x(\alpha)$ be the corresponding x to coefficients vector α. It's a 1-1 correspondence, and so from the smoothness assumption on $\mathcal{D}_{B_r}$, $\Pr_{\mathcal{D}'_{B_r}}(\alpha) = \Pr_{\mathcal{D}_{B_r}}(x(\alpha)) \in [\kappa_0, \kappa_1]$.

Define $k^* = \operatorname{argmax}_{k \in [m_2]} \mathbb{E}_{\alpha \sim \mathcal{D}'_{B_r}}[\alpha_k^2]$, then for any $i \neq k^*$:

$$\begin{aligned}
\mathbb{E}_{\alpha \sim \mathcal{D}'_{B_r}}[\alpha_{k^*}^2] &= \int_{S_{D'}} \alpha_{k^*}^2 \Pr_{\mathcal{D'}_{B_r}}(\alpha) d\alpha \\
&\leq \int_{S_{D'}} \alpha_{k^*}^2 \kappa_1 d\alpha \\
&= \int_{S_{D'}} \alpha_i^2 \kappa_1 d\alpha \\
&\leq \int \alpha_i^2 \frac{\kappa_1}{\kappa_0} \Pr_{\mathcal{D'}_{B_r}}(\alpha) d\alpha \\
&= \frac{\kappa_1}{\kappa_0} \mathbb{E}_{\alpha \sim \mathcal{D}'_{B_r}}[\alpha_i^2]
\end{aligned}$$

where the second equality follows from the symmetry of the support of $\mathcal{D}_{B_r}$, which is a sphere. This implies that:

$$\mathbb{E}_{\alpha \sim \mathcal{D}'_{B_r}}[\alpha_{k^*}^2] \leq \frac{\sum_{j=1}^{m_2} \frac{\kappa_1}{\kappa_0} \mathbb{E}_{\alpha \sim \mathcal{D}'_{B_r}}[\alpha_j^2]}{m_2} = \frac{\kappa_1}{\kappa_0} \frac{\mathbb{E}_{\alpha \sim \mathcal{D}'_{B_r}}[\sum_{j=1}^{m_2} \alpha_j^2]}{m_2} = \frac{\kappa_1}{\kappa_0} \frac{r^2}{m_2}$$

Therefore:

$$\begin{aligned}
\mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{B_r}}[\|A\mathbf{x}\|_2^2] &= \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{B_r}}[\mathbf{x}^T A^T A \mathbf{x}] \\
&= \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{B_r}}[\mathbf{x}^T (\sum_{j=1}^{m_2} \alpha_j \lambda_j \mathbf{u_j})] \\
&= \mathbb{E}_{\alpha \sim \mathcal{D}'_{B_r}}[\sum_{j=1}^{m_2} \lambda_j \alpha_j^2] \\
&= \sum_{j=1}^{m_2} \lambda_j \mathbb{E}_{\alpha \sim \mathcal{D}'_{B_r}}[\alpha_j^2] \\
&\leq \mathbb{E}_{\alpha \sim \mathcal{D}'_{B_r}}[\alpha_{k^*}^2] \sum_{j=1}^{m_2} \lambda_j \\
&\leq \frac{\kappa_1}{\kappa_0} \frac{r^2}{m_2} \sum_{j=1}^{m_2} \lambda_j
\end{aligned}$$

□

Theorem 41. *Assuming that $v - \hat{v}$ is drawn from distribution $\mathcal{D}_{B_r}$ with support equal to a sphere and smooth in that $\kappa_0 \leq \Pr_{\mathcal{D}_{B_r}}(x) \leq \kappa_1$ for any point x in its support, then:*

$$\mathbb{E}_{v - \hat{v} \sim \mathcal{D}_{B_r}}[\|\varphi(v) - \varphi(\hat{v})\|_2^2] \leq \frac{6}{n} \frac{\kappa_1}{\kappa_0} \frac{\|v - \hat{v}\|_2^2}{2^n}$$

Proof. To obtain the bound in the theorem, using this Lemma 60, we can then perform an average case analysis:

$$\mathbb{E}[\|S_n\mathbf{x}\|_2^2] \leq \frac{\kappa_1}{\kappa_0}\frac{\mathrm{Tr}(S_n^T S_n)}{2^n}\|\mathbf{x}\|_2^2 = \frac{\kappa_1}{\kappa_0}\frac{\mathrm{Tr}(S_n S_n^T)}{2^n}\|\mathbf{x}\|_2^2$$

We know that $\mathrm{Tr}(S_n S_n^T) = nd_1$ so the average case multiplier of the noise is:

$$\begin{aligned}
d_1 &= \frac{1}{n^2}\sum_{k=1}^{n}\binom{n-1}{k-1}^{-1} + \frac{1}{n^2}\sum_{k=0}^{n-1}\binom{n-1}{k}^{-1} \\
&= \frac{1}{n^2}(2 + \sum_{k=1}^{n-1}\binom{n-1}{k-1}^{-1} + \binom{n-1}{k}^{-1}) \\
&= \frac{2}{n^2} + \frac{1}{n^2}(\sum_{k=1}^{n-1}\frac{(k-1)!(n-k-1)!(k+n-k)}{(n-1)!}) \\
&= \frac{2}{n^2} + \frac{1}{n(n-1)}(\sum_{k=1}^{n-1}\binom{n-2}{k-1}^{-1}) \\
&= \frac{2}{n^2} + \frac{2}{n(n-1)} + \frac{1}{n(n-1)}(\sum_{k=2}^{n-2}\binom{n-2}{k-1}^{-1}) \\
&\leq \frac{2}{n^2} + \frac{2}{n(n-1)} + \frac{1}{n(n-1)}((n-3)\binom{n-2}{1}^{-1}) \\
&= \frac{2}{n^2} + \frac{3n-7}{n(n-1)(n-2)} \\
&\leq \frac{6}{n^2}
\end{aligned}$$

So, the multiplier is $\frac{\kappa_1}{\kappa_0}\frac{6}{n2^n}$ over the distribution $\mathcal{D}_{B_r}$. □

Next, we can obtain a more general result by integrating across all L2 norms r that $v - \hat{v}$ can take.

Corollary 7. *Suppose noise $v - \hat{v} \sim \mathcal{D}_n$ is such that its conditional distribution satisfies $\kappa_0(r) \leq \mathrm{Pr}_{\mathcal{D}_n}(x|\|x\|_2^2 = r^2) \leq \kappa_1(r)$ for all r and x in $\mathcal{D}_n$'s support, then:*

$$\mathbb{E}_{v-\hat{v}\sim\mathcal{D}_n}[\|\varphi(v) - \varphi(\hat{v})\|_2^2] \leq \frac{6}{n}\mathbb{E}_r\left[\frac{\kappa_1(r)}{\kappa_0(r)}\left(\frac{r^2}{2^n}\right)\right]$$

Proof. This follows from iterated expectation:

$$
\begin{aligned}
\mathbb{E}_{v-\hat{v}\sim\mathcal{D}}[\|\varphi(v)-\varphi(\hat{v})\|_2^2] &= \mathbb{E}_r[\mathbb{E}_{v-\hat{v}\sim\mathcal{D}_{B_r}}[\|\varphi(v)-\varphi(\hat{v})\|_2^2 \mid \|v-\hat{v}\|_2^2 = r^2]] \\
&\leq \mathbb{E}_r\left[\left(\frac{\kappa_1(r)}{\kappa_0(r)}\frac{6}{n2^n}\right)r^2\right] \\
&= \frac{6}{n2^n}\mathbb{E}_r\left[\frac{\kappa_1(r)}{\kappa_0(r)}r^2\right]
\end{aligned}
$$

where the inequality holds by Theorem 37.

□

Remark: Therefore, if $\mathbb{E}_r[\frac{\kappa_1(r)}{\kappa_0(r)}r^2] = c\mathbb{E}_r[r^2]$ for some constant $c = O(1)$, then the error in the Shapley value is fairly small and proportional to $O(1/n)$ of the average L2 error of $v - \hat{v}$.

Theorem 42 (Shapley noise L1 bound)**.** *The sum of absolute errors in Shapley values is bounded by:*

$$
\sum_{i=1}^{n} \left|\varphi_i(v) - \varphi_i(\hat{v})\right| \leq \sum_{C\in 2^A} \left|v(C) - \hat{v}(C)\right| \tag{7.3}
$$

Assuming there is no error in estimating the grand coalition nor the empty set and $n \geq 3$, then we can give a stronger bound on the sum of absolute errors:

$$
\sum_{i=1}^{n} \left|\varphi_i(v) - \varphi_i(\hat{v})\right| \leq \frac{2}{n}\sum_{C\in 2^A} \left|v(C) - \hat{v}(C)\right| \tag{7.4}
$$

Furthermore, assume players are divided into m equal sized teams, $G_1, ..., G_m$, where $|G_i| = N/m$. Then if we compute their Shapley values just with respect to their own teams we get:

$$
\sum_{i=1}^{n} \left|\varphi_i(v) - \varphi_i(\hat{v})\right| \leq \frac{2m}{n}\sum_{C\in 2^A} \left|v(C) - \hat{v}(C)\right| \tag{7.5}
$$

Proof. We can express the difference in Shapley value for i as:

$$
\begin{aligned}
|\varphi_i(v) - \varphi_i(\hat{v})| &= |\frac{1}{n}\sum_{S\subseteq[n]\setminus\{i\}} \binom{n-1}{|S|}^{-1} ([v(S\cup\{i\}) - \hat{v}(S\cup\{i\})] - [v(S) - \hat{v}(S)])| \\
&\leq \frac{1}{n}\sum_{S\subseteq[n]\setminus\{i\}} \binom{n-1}{|S|}^{-1} (|v(S\cup\{i\}) - \hat{v}(S\cup\{i\})| + |v(S) - \hat{v}(S)|) \\
&= \frac{1}{n}\sum_{s=0}^{n-1} \binom{n-1}{s}^{-1} \sum_{S\subseteq[n]\setminus\{i\},|S|=s} (|v(S\cup\{i\}) - \hat{v}(S\cup\{i\})| + |v(S) - \hat{v}(S)|)
\end{aligned}
$$

Thus, for any S of size s:

- If it contains element i, its L1 v error is weighted by $\binom{n-1}{s-1}^{-1}$.
- If it doesn't, it is weighted by $\binom{n-1}{s}^{-1}$.

Observe that the unweighted RHS is equal to:

$$\begin{aligned}
&= \frac{1}{n}\sum_{s=0}^{n-1}\sum_{S\subseteq[n]\backslash\{i\},|S|=s}(|v(S\cup\{i\})-\hat{v}(S\cup\{i\})|+|v(S)-\hat{v}(S)|)\\
&= \frac{1}{n}\sum_{S\subseteq[n]\backslash\{i\}}|v(S\cup\{i\})-\hat{v}(S\cup\{i\})|+\sum_{S\subseteq[n]\backslash\{i\}}|v(S)-\hat{v}(S)|\\
&= \frac{1}{n}||v-\hat{v}||_1
\end{aligned}$$

Therefore, since $\binom{n-1}{s}^{-1}\leq 1$ for $s\in[0,n-1]$:

$$\begin{aligned}
|\varphi_i(v)-\varphi_i(\hat{v})| &\leq \frac{1}{n}\sum_{s=0}^{n-1}\binom{n-1}{s}^{-1}\sum_{S\subseteq[n]\backslash\{i\},|S|=s}(|v(S\cup\{i\})-\hat{v}(S\cup\{i\})|+|v(S)-\hat{v}(S)|)\\
&\leq \frac{1}{n}||v-\hat{v}||_1
\end{aligned}$$

Summing across all i's, this proves inequality (7.3).

Note that $\binom{n-1}{s}^{-1}=1$ holds only for (i) the full set $[n]$ $(s=n-1)$ (ii) the set $\{i\}$ $(s=0)$ (iii) empty set $(s=0)$ (iv) set $[n]\backslash\{i\}=[-i]$ $(s=n-1)$. Thus we can obtain equality if all of the errors in v lie in estimating the full set or the empty set. This makes the bound tight.

We obtain a stronger inequality (7.4) if we assume that there is no error in estimating the empty nor the grand coalition value:

Let $e=||v-\hat{v}||_1$ and $e_i=|v(\{i\})-\hat{v}(\{i\})|+|v([-i])-\hat{v}([-i])|$. Then:

$$\begin{aligned}
|\varphi_i(v)-\varphi_i(\hat{v})| &\leq \frac{1}{n}\sum_{s=0}^{n-1}\binom{n-1}{s}^{-1}\sum_{S\subseteq[n]\backslash\{i\},|S|=s}(|v(S\cup\{i\})-\hat{v}(S\cup\{i\})|+|v(S)-\hat{v}(S)|)\\
&\leq \frac{1}{n}e_i+\frac{1}{n}\sum_{s=1}^{n-2}\binom{n-1}{s}^{-1}\sum_{S\subseteq[n]\backslash\{i\},|S|=s}(|v(S\cup\{i\})-\hat{v}(S\cup\{i\})|+|v(S)-\hat{v}(S)|)\\
&\leq \frac{1}{n}e_i+\frac{1}{n(n-1)}(e-e_i)
\end{aligned}$$

since $\binom{n-1}{s}^{-1}\leq\frac{1}{n-1}$ for $s\in[1,n-2]$.

Summing this across i gives:

$$\begin{aligned}
|\varphi(v)-\varphi(\hat{v})| &\leq \frac{1}{n}\sum_{i=1}^{n} e_i + \frac{1}{n(n-1)}(ne - \sum_{i=1}^{n} e_i) \\
&= \frac{e}{n-1} + \frac{n-2}{n(n-1)}(\sum_{i=1}^{n} e_i) \\
&\leq \frac{e}{n-1} + \frac{n-2}{n(n-1)}e \\
&= \frac{2e}{n}
\end{aligned}$$

since $\sum_{i=1}^{n} e_i \leq e$.

This proves inequality (7.4).

In some case, as is the case with our NBA experimental setup, players are divided into m groups, $G_1, ..., G_m$, and we wish to compute their Shapley values only with respect to their own groups. We can follow a similar analysis as above to derive a bound on the Shapley values. For player $i \in G_j$:

$$\begin{aligned}
|\varphi_i(v)-\varphi_i(\hat{v})| &= |\frac{1}{|G_j|}\sum_{S\subseteq G_j\backslash\{i\}} \binom{|G_j|-1}{|S|}^{-1} ([v(S\cup\{i\}) - \hat{v}(S\cup\{i\})] - [v(S)-\hat{v}(S)])| \\
&\leq \frac{1}{|G_j|}\sum_{S\subseteq G_j\backslash\{i\}} \binom{|G_j|-1}{|S|}^{-1} (|v(S\cup\{i\}) - \hat{v}(S\cup\{i\})| + |v(S)-\hat{v}(S)|) \\
&= \frac{1}{|G_j|}\sum_{s=0}^{|G_j|-1} \binom{|G_j|-1}{s}^{-1} \sum_{S\subseteq G_j\backslash\{i\},|S|=s} (|v(S\cup\{i\}) - \hat{v}(S\cup\{i\})| + |v(S)-\hat{v}(S)|)
\end{aligned}$$

Let $E_j = \sum_{S\subseteq G_j} |v(S)-\hat{v}(S)|$. It's clear that $\sum_{j=1}^{m} E_j \leq e$ since any two teams G_{j_1}, G_{j_2} are disjoint for $j_1 \neq j_2$ and thus don't have any subsets in common; note our assumption that the empty set is estimated without any error by $\hat{v}$.

To maximize the cumulative error, all the errors in e should be placed in subsets S with $S \subseteq G_j$ for some j. So WLOG we can assume that $\sum_{j=1}^{m} E_j = e$. Using inequality (7.4), we get that:

$$\sum_{i\in G_j} |\varphi_i(v)-\varphi_i(\hat{v})| \leq \frac{2E_j}{|G_j|}$$

So the overall bound is:

$$|\varphi(v)-\varphi(\hat{v})| \leq \sum_{j=1}^{m} \frac{2E_j}{|G_j|}$$

Assume $|G_j| = n/m$ for each j, this simplifies to $\frac{2m}{n}e$ and proves inequality 7.5.

□

While the above bounds are tight, the analysis is worst case. For instance, for the first bound we provide, equality holds when all the error in the $\|v - \hat{v}\|_1$ vector is in the coalition value of the grand coalition or the empty set. Below, we provide a simple, average case analysis to show that on average, a randomly drawn error vector leads to a small increase in L1 Shapley error in expectation.

Theorem 43 (Average case Shapley noise L1 bound)**.** *Assuming that the error $\mathbf{v} - \hat{\mathbf{v}}$ is such that the vector $|\mathbf{v} - \hat{\mathbf{v}}|/r$ (where absolute value is coordinate wise) is drawn from distribution $\mathcal{D}_{S_r}$ with support equal to the surface of a 2^n-simplex and smooth in that $\kappa_0 \leq \Pr_{\mathcal{D}_{S_r}}(\mathbf{x}) \leq \kappa_1$ for any point $\mathbf{x}$ in its support, then $\mathbb{E}_{\mathbf{v}-\hat{\mathbf{v}}\sim\mathcal{D}_{S_r}}[\|\varphi(\mathbf{v}) - \varphi(\hat{\mathbf{v}})\|_1] \leq 2\frac{\kappa_1}{\kappa_0}\frac{\|\mathbf{v}-\hat{\mathbf{v}}\|_1}{2^n}$.*

Proof.

$$\begin{aligned}
&\mathbb{E}_{\mathbf{v}-\hat{\mathbf{v}}\sim\mathcal{D}_{S_r}}\big[\big|\varphi_i(\mathbf{v}) - \varphi_i(\hat{\mathbf{v}})\big|\big] \\
&= \mathbb{E}_{\mathbf{v}-\hat{\mathbf{v}}\sim\mathcal{D}_{S_r}}\left[\Big|\frac{1}{n}\sum_{S\subseteq[n]\backslash\{i\}}\binom{n-1}{|S|}^{-1}([v(S\cup\{i\}) - \hat{v}(S\cup\{i\})] - [v(S) - \hat{v}(S)])\Big|\right] \\
&\leq \frac{1}{n}\sum_{s=0}^{n-1}\binom{n-1}{s}^{-1}\sum_{S\subseteq[n]\backslash\{i\},|S|=s}\Big(\mathbb{E}_{\mathbf{v}-\hat{\mathbf{v}}\sim\mathcal{D}_{S_r}}\big[[|v(S\cup\{i\}) - \hat{v}(S\cup\{i\})|\big]+ \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathbb{E}_{\mathbf{v}-\hat{\mathbf{v}}\sim\mathcal{D}_{S_r}}\big[|v(S) - \hat{v}(S)|\big]\Big) \\
&\overset{(1)}{\leq} \frac{1}{n}\sum_{s=0}^{n-1}\binom{n-1}{s}^{-1}\sum_{S\subseteq[n]\backslash\{i\},|S|=s}\left(\frac{\kappa_1}{\kappa_0}\frac{r}{2^n} + \frac{\kappa_1}{\kappa_0}\frac{r}{2^n}\right) \\
&= \frac{2}{n}\frac{\kappa_1}{\kappa_0}\frac{r}{2^n}
\end{aligned}$$

where (1) is due to the following:

Let subset $C^* = \operatorname{argmax}_C \mathbb{E}_{\mathbf{v}-\hat{\mathbf{v}}\sim\mathcal{D}_{S_r}}[[|v(C) - \hat{v}(C)|]$ and subset $C' = \operatorname{argmin}_C \mathbb{E}_{\mathbf{v}-\hat{\mathbf{v}}\sim\mathcal{D}_{S_r}}[[|v(C) - \hat{v}(C)|]$:

$$\begin{aligned}
\mathbb{E}_{\mathbf{v}-\hat{\mathbf{v}}\sim\mathcal{D}_{S_r}}[[|v(C^*) - \hat{v}(C^*)|] &= \int |v(C^*) - \hat{v}(C^*)| \Pr_{\mathcal{D}_{S_r}}(\mathbf{v}-\hat{\mathbf{v}})d(\mathbf{v}-\hat{\mathbf{v}}) \\
&\leq \int |v(C^*) - \hat{v}(C^*)|\kappa_1 d(\mathbf{v}-\hat{\mathbf{v}}) \\
&\overset{(2)}{=} \int |v(C') - \hat{v}(C')|\kappa_1 d(\mathbf{v}-\hat{\mathbf{v}}) \\
&\leq \int |v(C') - \hat{v}(C')|\frac{\kappa_1}{\kappa_0} \Pr_{\mathcal{D}_{S_r}}(\mathbf{v}-\hat{\mathbf{v}})d(\mathbf{v}-\hat{\mathbf{v}}) \\
&= \frac{\kappa_1}{\kappa_0}\mathbb{E}_{\mathbf{v}-\hat{\mathbf{v}}\sim\mathcal{D}_{S_r}}[[|v(C') - \hat{v}(C')|] \\
&\overset{(3)}{\leq} \frac{\kappa_1}{\kappa_0}\Big(\frac{r}{2^n}\Big).
\end{aligned}$$

Here (2) holds by symmetry as the expectation of any two vector coordinates under a uniform distribution over the simplex of vectors is the same. (3) holds because every vector in the support of $\mathcal{D}_{S_r}$ has L1 norm of r, $\sum_C \mathbb{E}_{\mathbf{v}-\hat{\mathbf{v}}\sim\mathcal{D}_{S_r}}[|v(C)-\hat{v}(C)|] = \mathbb{E}_{\mathbf{v}-\hat{\mathbf{v}}\sim\mathcal{D}_{S_r}}[\sum_C |v(C)-\hat{v}(C)|] = r$ and so by our choice of C', $\mathbb{E}_{\mathbf{v}-\hat{\mathbf{v}}\sim\mathcal{D}_{S_r}}[|v(C')-\hat{v}(C')|] \leq \frac{r}{2^n}$.

Summing $\mathbb{E}_{\mathbf{v}-\hat{\mathbf{v}}\sim\mathcal{D}_{S_r}}[|\varphi_i(\mathbf{v})-\varphi_i(\hat{\mathbf{v}})|]$ across all i gives the result.

□

This means that, on average, for a randomly drawn $\varphi(\mathbf{v})-\varphi(\hat{\mathbf{v}})$ with a fixed error budget in L1 error, the L1 error in the Shapley is only proportional to the average error in estimating each coalition. Next, we can obtain a more general bound by integrating across all L1 norms r that $\varphi(\mathbf{v})-\varphi(\hat{\mathbf{v}})$ can take.

Corollary 8. *Suppose noise $v-\hat{v}\sim\mathcal{D}_n$ is such that its conditional distribution satisfies $\kappa_0(r) \leq \Pr_{\mathcal{D}_n}(x|\|x\|_1 = r) \leq \kappa_1(r)$ for all r and x in $\mathcal{D}_n$'s support, then:*

$$\mathbb{E}_{v-\hat{v}\sim\mathcal{D}_n}[\|\varphi(\mathbf{v})-\varphi(\hat{\mathbf{v}})\|_1] \leq 2\mathbb{E}_r\left[\frac{\kappa_1(r)}{\kappa_0(r)}\left(\frac{r}{2^n}\right)\right]$$

Proof. This follows from iterated expectation:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{v}-\hat{\mathbf{v}}\sim\mathcal{D}}[\|\varphi(\mathbf{v})-\varphi(\hat{\mathbf{v}})\|_1]\\
&= \mathbb{E}_r\left[\mathbb{E}_{\mathbf{v}-\hat{\mathbf{v}}\sim\mathcal{D}_{S_r}}\left[\|\varphi(\mathbf{v})-\varphi(\hat{\mathbf{v}})\|_1 \mid \|\mathbf{v}-\hat{\mathbf{v}}\|_1 = r\right]\right]\\
&\leq \mathbb{E}_r\left[\left(\frac{\kappa_1(r)}{\kappa_0(r)}\frac{2}{2^n}\right)r\right]\\
&= 2\mathbb{E}_r\left[\frac{\kappa_1(r)}{\kappa_0(r)}\frac{r}{2^n}\right]
\end{aligned}$$

where the inequality holds by the Theorem above.

□

Remark: Therefore, if $\mathbb{E}_r[\frac{\kappa_1(r)}{\kappa_0(r)}r] = c\mathbb{E}_r[r]$ for some constant $c = O(1)$, then the error in the Shapley value is fairly small and proportional to the average expected L1 error $\frac{\mathbb{E}_r[r]}{2^n}$.

7.8.4 Discussion about CGA-Specific Errors:

Since CGA is a *complete representation*, every game may be expressed as a CGA of some order (see Fact 1). And so, we may plug the CGA-specific bias into the general bounds obtained previously in Theorems 3-6.

Below we derive CGA bias due to model misspecification. Note that the approximation is lossy only when the true game is generated by a CGA model of order r and we model it with a simpler CGA model of order k with $k < r$. When we model the game with a CGA of a higher order than the actual game, it is clear that we can learn a set of weights that would fit the coalition values exactly (since v would be in the columnspace).

Let M^{nk} denote the matrix relating the parameters ω to the coalitional values $\mathbf{v}$. It is a $2^n \times d_k$ matrix of the form:

$$\begin{array}{lc} & \begin{array}{ccc}\text{first order weights} & \text{...} & k\text{th order weights}\end{array} \\ \begin{array}{l}\text{row corresponding to null coalition } \{\} \\ \text{...} \\ \text{row corresponding to grand coalition } A\end{array} & \begin{pmatrix} \text{...} & \text{...} & \text{...} \\ \text{...} & \text{...} & \text{...} \\ \text{...} & \text{...} & \text{...} \end{pmatrix}\end{array}$$

The CGA model parameters $\hat{\omega}$ we learn will be such that:

$$\hat{\omega} = \underset{\mathbf{w}}{\operatorname{argmin}} \, \|M^{nk}\mathbf{w} - M^{nr}\omega^*_{\mathbf{r}}\|_2^2$$

This is just equivalent to projecting vector $M^{nr}\omega^*_{\mathbf{r}}$ onto the columnspace spanned by M^{nk}. Recall from our identification theorem that M^{nk} has enough rows to be full rank, which makes $(M^{nk})^T M^{nk}$ positive definite and invertible; if there are not enough samples, we may instead consider a regularization term that will make the matrix invertible. Define projection matrix P_{nk}:

$$P_{nk} = M^{nk}(((M^{nk})^T M^{nk})^{-1}(M^{nk})^T$$

This means that the misspecification error $e(n, k, r)$ may be expressed as:

$$e(n, k, r) = (I - P_{nk})M^{nr}\omega^*_{\mathbf{r}}$$

which we may plug into our noise bounds for the Shapley value computation.

Unlike the Shapley matrix, the error matrix $(I - P_{nk})M^{nr}$ does not seem to admit a closed form for its trace. Instead, we perform simulations to better understand its properties. In particular, we look to understand if it enjoys the same "averaging-effect" as the Shapley matrix. We compute the max eigenvalue and the average trace norm value sweeping over all n, r, k for $k < r < n$ for $n \in [2, 15]$ (we try these sizes since 15 is the largest possible before the error matrix's size exceeds that permitted by our machine memory). Our simulations suggest that its largest eigenvalue (for the worst case bound) and the average trace value (for the average case bound per Lemma 60) both grow monotonically with n and r (fixing a k). Altogether, this suggests that the ℓ_2 error can grow arbitrarily large with model misspecification.

7.8.5 For Practitioners: How to choose the order of the CGA Model

The order k of the CGA model is dependent on the application at hand. It may be set to be the maximum k-way interaction that the practitioner expects to take place in the team.

Our model is especially useful in settings like the NBA, in which team sizes are small relative to the overall number of players. This structural prior can be encoded in the order of the CGA. As an example, for the NBA, we expect at most 5-way interaction and so it is necessary to only consider compact, low-rank models.

Note that for the time and space complexity of the model, the time to compute the SV is the space complexity of the CGA model: the number of parameters. The complexity of learning a CGA depends on the training method employed to learn $\hat{v}$ (e.g. we use SGD).

7.8.6 Proofs of Facts

For completeness, we provide proofs of the two facts listed.

Fact 11 (Unique decomposition form)**.** *There exists a unique set of values ω_S for each subset $S \subseteq A$ with $|S| \leq k$ such that the characteristic function can be decomposed into its interaction form where*

$$v(C) = \sum_{k=1}^{|C|} \sum_{S \in 2_k^C} \omega_S.$$

Proof. We can show this inductively. For the base case when $|C| = 1$ we have $w_C = v(C)$, which is unique.

Induction step: assume $w_{S'}$ is uniquely determined for $|S'| = 1, ..., m-1$. Then for a particular subset $|S| = m$:

$$v(S) = \sum_{i=1}^{m-1} \sum_{S' \in 2_i^S} w_{S'} + w_S$$

and thus w_S is uniquely determined since we must set it to

$$w(S) = v(S) - \sum_{i=1}^{m-1} \sum_{S' \in 2_i^S} w_{S'}$$

□

Fact 12 (Shapley value expression)**.** *The Shapley Value of an individual i with respect to team A can be expressed as:*

$$\varphi_i(v) = \sum_{T \subseteq A \setminus \{i\}} \frac{1}{|T|+1} \omega_{T \cup \{i\}}$$

Proof. The Shapley value for player i is defined as:

$$\varphi_i(v) = \frac{1}{n} \sum_{S \subseteq A \setminus \{i\}} \binom{n-1}{|S|}^{-1} (v(S \cup \{i\}) - v(S))$$

Plugging in the decomposition form:

$$v(S \cup \{i\}) - v(S) = \sum_{S' \subseteq S} w_{S' \cup \{i\}}$$

Thus, the Shapley value for i is only a function of all w_S where $i \in S$.

Given a subset $T = \{i_1 ... i_t\}$, let us derive the weighted sum of w_T occurrences in $\varphi_{i_1}(v)$. This term only appears if $\{i_2, .., i_t\} \subseteq S$ but $i_1 \notin S$. And so, the weighted sum of occurrences is:

$$\frac{1}{n} \sum_{S \subseteq A \setminus \{i_1\}, \{i_2,..,i_t\} \in S} \binom{n-1}{|S|}^{-1} = \frac{1}{n} \sum_{s=t-1}^{n-1} \binom{n-1}{s}^{-1} \binom{n-t}{s-(t-1)}$$

Similarly, w_T has the same sum of weighted occurrences in expressions for players $i_2, ..., i_t$. And so, by efficiency (since $v(A)$ contains exactly w_T and the sum of Shapley payments equals $v(A)$), they must be assigned equal portions of w_T, i.e $w_T/|T|$. This holds for all subsets T. And so, a player i's Shapley value is the sum of all weights $w_T/|T|$, for all subsets $T \subseteq [n]$ and $i \in T$. □

7.8.7 Relationship to the Core

The main text of the paper has focused on the solution concept of the Shapley Value. Another commonly used solution concept in cooperative game theory is known as the Core [117]. Let n be the number of players in the game, the Core is an allocation $x \in \mathbb{R}^n$ that satisfies:

(i) Efficiency: $\sum_{i \in [n]} x_i = v([n])$

(ii) Stability: for any coalition $C \subset [n]$:

$$\sum_{i \in C} x_i \geq v(C)$$

Intuitively, a payoff vector is in the Core if it incentivizes every coalition C to stay with the grand coalition rather than leave, achieve a value of $v(C)$ and split it amongst themselves in some other way.

The Core of a game may be empty, though an extension known as the Least Core is always guaranteed to exist. The Least Core can be computed by solving the following linear program:

$$\begin{array}{ll} \min_{e,x} & e \\ s.t. & \sum_{i \in [n]} x_i = v([n]) \\ & \sum_{i \in C} x_i \geq v(C) - e \quad \forall S \subset [n] \end{array}$$

Intuitively, the Least Core is the allocation which minimizes the subsidy e required to incentivize all coalitions to stay together. We call the minimum subsidy needed the Least Core value. Unfortunately, [92] show that for any CGA model with order higher than 1, it is NP-Complete to compute the Least Core Value.

One notable allocation in the Least Core is the Nucleolus. For a given allocation x, define deficit function $e_x(C) = v(C) - \sum_{i \in C} x_i$. Order all subsets of $[n]$ according to the deficit function e_x. The nucleolus is defined as the imputation which lexicographically minimizes this ordering of deficits. Intuitively, the Nucleolus is the "inner-most" allocation in the Least Core. In general, the Nucleolus is difficult to compute and requires solving a series of exponential-size linear programs.

Remarkably, [92] prove the following fact:

Fact 13. *Assuming the characteristic function of the underlying game is a second order CGA model, the Shapley Value is in the Least Core (in fact, it is the Nucleoulus).*

Therefore, we can simply compute the Shapley value to obtain a point in the Least Core. All that remains is to approximate the Least Core value. To do this, we establish an approximate notion of the Least Core value by adapting a similar notion from [35] and derive a simple sample complexity bound for estimating this value. The definition goes as follows:

Definition 41. *Given an allocation* $\mathbf{x}$*, a value* e *is a* $\delta-$*probable least core value if:*

$$\mathrm{Pr}_{C\sim 2^A}[\sum_{i\in C} x_i + e \geq v(C)] \geq 1-\delta$$

The least core value is the smallest e^* such that there exist an allocation for which e^* is a $0-$probable least core value.

We will compute a $\delta-$probable least core value by computing the sample least core value on a set of uniformly sampled coalitions $\widehat{S}$. Certainly if $|\widehat{S}| = 2^n$ coalitions, then the sample least core value will be the true least core value exactly. Using standard learning theory tools, we can relate the quality of the estimation of the least core value, in terms of δ, to the size of the samples $\widehat{S}$ needed:

Theorem 44. *Given a set* $\widehat{S}$ *of* $m = O(\frac{\log(1/\Delta)}{\delta^2})$ *coalitions uniformly sampled from* 2^A*, let:*

$$\begin{aligned}\hat{e} \quad &= \operatorname{argmin} e \\ &\textstyle\sum_{i\in C} \varphi_i(\hat{v}) \geq v(C) - e \quad \forall C \subseteq \widehat{S}\end{aligned}$$

then with probability $1-\Delta$ *over the samples,* $\hat{e}$ *is a* $\delta-$*least core value.*

Proof. We prove this through a simple learning theory setup analogous to the proposition above. Define a 2-dimensional linear classifier with weights $\mathbf{w_e} = [e, 1]$. This class of classifier is a subset of all linear classifiers of dimension 2 and thus has VC dimension ≤ 2.

For each of the $2^n - 2$ inequality constraints, construct data point $[1, \sum_{i\in C} \varphi_i(\hat{v}) - v(C)]$ that corresponds to coalition C's constraint. We assign each data point a label of 1. Notice that if classifier $\mathbf{w_e} = [e, 1]$ classifies $[1, \sum_{i\in C} \varphi_i(\hat{v}) - v(C)]$ correctly, then:

$$\mathrm{sign}_{\mathbf{w_e}}([1, \sum_{i\in C} \varphi_i(\hat{v}) - v(C)]) = 1 \Rightarrow [e, 1]^T[1, \sum_{i\in C} \varphi_i(\hat{v}) - v(C)] \geq 0 \Rightarrow \sum_{i\in C} \varphi_i(\hat{v}) \geq v(C) - e$$

Moreover, we know that the classifier we obtain, $\mathbf{w_{\hat{e}}} = [\hat{e}, 1]$, is such that it classifies all the samples in $\widehat{S}$ correctly by construction, and has zero empirical risk. Again, using Lemma 59, we know that this classifier's performance on the samples generalize to all $2^n - 2$ constraints. In particular, if there are at least

$$O(\frac{2 + \log(1/\Delta)}{\delta^2})$$

samples in $\widehat{S}$, then the empirical least core value $\hat{e}$ we compute is such that:

$$\mathrm{Pr}_{C\sim 2^A}[\sum_{i\in C} \varphi_i(\hat{v}) \geq v(C) - \hat{e}] = \mathrm{Pr}_{C\sim 2^A}[\mathrm{sign}_{\mathbf{w_{\hat{e}}}}([1, \sum_{i\in C} \varphi_i(\hat{v}) - v(C)]) = 1] \geq 1-\delta$$

□

Lastly, we remark that for games whose characteristic functions are CGA models of order higher than 2, the Shapley is not the Nucleolus. An interesting extension of this work could be developing faster, sample-based methods for computing the Least Core with higher order CGA models.

7.8.8 Experiments Hyper Parameter Search

In the low rank approximations of $\widehat{V}$ (as suggested by [186, 254]), we represented a team C via a one-hot encoding $\mathbf{x_C}$ and fit a model of the form:

$$\hat{v}(C) = \mathbf{w}^T \mathbf{x_C} + \mathbf{x_C}^T \hat{V} \mathbf{x_C}$$

We tried parameterizing $\hat{V}$ via a low-rank matrix and swept weight decay (l_2 regularization) parameters on our validation set. Here we report the results of the full sweep for both of our experiments.

Table 7.1 shows the MSE (lower is better) of the performance prediction for various parameter values in the OpenAI particle world experiment. Table 7.2 shows the accuracy of the model in predicting wins (higher is better) in the NBA experiment. In both cases we see that a relatively low rank model does very well at capturing structure in our environments. The main text analyzes the models resulting from these parameter choices.

L2 regularization/$\hat{V}$ Rank	1	2	5	10	20	35
0.001	0.256	0.092	**0.066**	0.067	0.068	0.069
0.01	0.261	0.104	0.091	0.090	0.093	0.090
0.1	0.679	0.679	0.652	0.646	0.669	0.664

Table 7.1: Results of hyper-parameter sweep for the second order CGA in the OpenAI particle world experiment. MSE is shown, lower is better.

L2 regularization/$\hat{V}$ rank	5	10	20	50	100
0.001	0.61	0.6214	0.6086	0.5971	0.5929
0.01	0.64	0.6414	0.6414	**0.6429**	0.6429
0.1	0.6214	0.63	0.62	0.6286	0.6242

Table 7.2: Results of hyper-parameter sweep for the second order CGA in the NBA data. Model accuracy is shown, higher is better.

7.9 Additional Related Works in Cooperative Game Theory

In computational, cooperative game theory [59], there has been ample, albeit orthogonal prior work that studies CF representations and Shapley computation. We may classify them as follows:

- Fast methods to compute the SV for certain subclasses of games: [106] (for voting games). By contrast, our representation permits facile Shapley computation for all cooperative games.
- Sample complexity of approximating the SV: [195], [21] (only for simple games), [187] (only for supermodular games). By contrast, our bounds focus on the sample complexity of learning the CF function with the CGA representation, thus drawing upon PAC/PMAC techniques. None of the other works in this category need to nor use learning theoretic methods. They focus only on studying the concentration of estimated SV values via standard concentration inequalities.
- Representation designed to allow for easy computation of the SV: [202] (only for networks), [77]. By contrast, our CGA model has provable, learning theoretic properties (and additionally, practical success on real world data). The provable guarantee is crucial since we need to use the model to learn the unknown CF from data.

Chapter 8

Multi-agent Attribution via the Core

8.1 Introduction

As machine learning systems become more capable, they are increasingly used in our society to automate tasks and generate value. This has lead to a surge in the attention given to the economics of machine learning: how features and data contribute to the performance of ML models. To ensure ML models are functioning as intended, much work has been devoted to studying *feature attribution*: how the features used to represent the data influence the model's predictions [61, 73, 86, 87, 193, 269]. Related to feature attribution is *data valuation* [4, 115, 156, 157, 218], which studies how data points contribute to model performance. With ML models now generating profit for enterprises, this understanding is important in order to fairly compensate data suppliers for their training data.

Similar to the previous chapter, the problem of data valuation may be viewed as a particular instance of the *multi-agent attribution* problem. Indeed, central to feature and data valuation is an equitable means of *credit assignment*.

Virtually all papers, including every single paper cited above, deem the Shapley value (or close variants thereof) to be the "right" way to carry out this credit assignment. The Shapley value is a solution concept from cooperative game theory in which players — in this case features or data points — are assigned payoffs in a way that satisfies four axioms; roughly speaking, a player's payoff is their average marginal contribution to a coalition consisting of other players.

This intense focus on the Shapley value is surprising, however, as — once we have accepted that problems of credit assignment in machine learning can be modeled as cooperative games — there are a plethora of other solution concepts [225]. In particular, there is a seminal solution concept in cooperative game theory that is as prominent as the Shapley value: the core. This solution concept seeks to achieve maximal stability among all possible coalitions of the players in the game — an idea that dates back to the writings of Edgeworth on market equilibrium theory in 1881. It is a solution concept that aims to capture economic feasibility.

Specifically, according to the core, the total payoff to each coalition should be at least its value. When this is not possible, the maximum deficit (difference between value and payoff) of any coalition should be minimized — this is known as the *least core*. The (least) core can be seen as a notion of *group fairness*, in that each group of players (or coalition) gets its dues. Moreover, it is especially apt in the valuation setting, where the data vendors or feature annotators are *paid* in a way that disincentivizes (to the extent possible) any coalition of vendors from choosing to opt out and not contribute; if a coalition S if not paid at least its value $v(S)$ then the coalition would be better off separating from the so-called grand coalition. Thus, the core values may be viewed as *the* set of all *economically plausible* payoffs to participants that compensate them for their contributions.

In this chapter, we provide a much needed comparison of the two solution concepts and show that the (least) core is, practically and conceptually, an attractive alternative to the Shapley value for credit assignment in machine learning. In doing so, we hope to raise awareness of the core as a natural solution concept for fair credit assignment, challenge the wide-ranging usage of the Shapley value and inspire a closer examination of cases where one solution concept should be preferred over the other. It is worth emphasizing that, to the best of our knowledge, we are the first to consider using the core for feature/data valuation in machine learning.

8.1.1 Our Results

Much like the Shapley value, the primary obstacle in applying the concept of least core is computational complexity. Indeed, it is the solution to a linear program whose number of constraints is exponential in the number of players. Nevertheless, we construct a Monte Carlo algorithm that runs in polynomial time and (with given confidence) outputs a payoff allocation in the *δ-probable least core* — a slightly relaxed version of the least core where the payoff constraints may be violated by up to a δ-fraction of coalitions. When the number of players is large, though, this may still be intractable; we therefore show that it is possible to find a solution in the *(ϵ, δ)-probably approximate least core* — whose constraints are additionally relaxed by ϵ each — in time that is polylogarithmic in the number of players.

We also study a well-known refinement of the least core called the *nucleolus*. However, it turns out that results that are analogous to those for the least core are essentially unattainable. Informally, we prove that *any* algorithm would have to require access to the values of an *exponentially* large number of coalitions to compute a payoff allocation in the (ϵ, δ)-probably approximate nucleolus, which again relaxes all relevant constraints by ϵ and allows a δ-fraction of the constraints to be violated. The juxtaposition of the positive computational results for the least core and the negative result for the nucleolus provides a strong endorsement of the former (somewhat coarser) notion over the latter.

In our experiments, we verify these theoretical results and confirm that our algorithm can compute the least core easily and that the nucleolus is difficult to compute. Next, we compare algorithms one would use to compute the Shapley value against our least core algorithm in data valuation tasks. Our results suggest that the least core algorithm compares favorably with those of the Shapley value in low-resource settings that are typical of analysts without access to large-scale computational resources.

8.1.2 Related Work

There is an entire area of algorithmic game theory devoted to the computation of solutions of cooperative games [58]. In particular, a slew of papers have studied the complexity of the core, the least core, and the nucleolus in specific classes of cooperative games [20, 76, 91, 99, 100].

Our work is most closely related to that of Balkanski et al. [34]. They study settings where solutions to cooperative games — specifically, the Shapley value and the core — are learned from samples consisting of coalitions and their values. Like Balcan et al. [28], they are motivated by the observation that in classical applications of cooperative games values of coalitions cannot be accessed via queries; for example, if the game represents company employees working together to complete tasks, it is impossible to know which tasks would be completed had a specific coalition worked alone. Importantly, they do not consider valuation at all. Under the assumption that the underlying game has a nonempty core, Balkanski et al. [34] give bounds on the sample complexity of three approximations of the core.

On a technical level, our definition of approximate notions of least core (Theorems 45 and 46) follow those of Balkanski et al. [34] for the core, by eschewing the assumption that the core is nonempty; our proofs of these results directly build on theirs. Our interpretation of these results is quite different, though, because in our setting coalition values *can* be queried — for example, one can run a black-box predictor with a specific subset of features and measure its accuracy — so we think of our results as guarantees on the performance of Monte Carlo algorithms. Balkanski et al. [34] did not study the nucleolus, so our negative result for the nucleolus (Theorem 47) — which we view as our main theoretical result — is entirely new and has no analog in their work. Finally, the work of Balkanski et al. [34] is purely theoretical, whereas our empirical results study and demonstrate the applicability of the least core to credit assignment in machine learning.

8.2 Preliminaries

A *cooperative game* consists of a set of players $N = \{1, \ldots, n\}$ and a *characteristic function* $v : 2^N \to \mathbb{R}$ which assigns a value to each *coalition* $S \subseteq N$, such that $v(\emptyset) = 0$; we assume that $v(S) \geq 0$ and $v(S) \leq 1$ for all $S \subseteq N$ for ease of exposition. We think of $v(S)$ as the payoff the coalition S could obtain if it went it alone. Given such a game, we are interested in finding a *payoff allocation* (also known as an *imputation*) $\mathbf{x} = (x_1, \ldots, x_n)$, where x_i is the payoff of player $i \in N$. The payoff allocation must be *efficient*, that is,

$$\sum_{i \in N} x_i = v(N).$$

A payoff allocation is in the e-*core* if and only if the total payoff of each coalition is at least its value, up to e:

$$\forall S \subseteq N, \sum_{i \in S} x_i + e \geq v(S).$$

The core itself, by this definition, satisfies these constraints with $e = 0$. Unfortunately, there are cooperative games whose core is empty. But clearly the e-core is nonempty if e is large enough.

The idea behind the *least core* [197] is to choose the smallest e possible. It may be defined as the set of all solutions to the following linear program.

$$\begin{array}{ll} \min & e \\ \text{s.t.} & \sum_{i\in N} x_i = v(N) \\ & \sum_{i\in S} x_i + e \geq v(S) \quad \forall S \subseteq N \end{array} \tag{8.1}$$

One can think of the least core as the set of payoff allocations that require the smallest subsidy $e^\star$ (the value of e in the optimal solution to (8.1)) to each coalition so that, if the payoff to each coalition was boosted by $e^\star$, the allocation would be in the core. The core is nonempty if and only if $e^\star \leq 0$.

We next consider a refinement of the least core, the *nucleolus*, first proposed by [250]. Define the *deficit* of a payoff allocation $\mathbf{x}$ for a coalition $S \subseteq N$ to be $v(S) - \sum_{i\in S} x_i$. The nucleolus is the payoff allocation whose sorted list of deficits across all coalitions lexicographically dominates the list of deficits for any other payoff allocation. That is, the largest deficit (which will be positive if the core is empty) should be as small as possible; subject to that, the second largest deficit should be as small as possible, and so on. Notice that, in particular, the nucleolus minimizes the largest deficit and so its allocation does lie in the least core. In contrast to the least core, which may contain multiple payoff allocations, the nucleolus is known to be unique [250].

8.3 Theoretical Results

Exact computation of the least core and the nucleolus requires solving linear programs with as many constraints as there are coalitions, which would typically be prohibitively expensive. Our strategy, therefore, is to sample a relatively small number of coalitions from an underlying distribution, and compute the desired solution concept on the sampled coalitions — this can be done in time that is polynomial in the number of samples, via the linear program (8.1) for the least core, and via a sequence of such linear programs for the nucleolus [168]. The hope is that this Monte Carlo algorithm would give us a payoff allocation that approximates the desired one with respect to the underlying distribution.

8.3.1 Computing the Least Core

We know from the work of Balkanski et al. [34] that computing the least core exactly is a non-starter — they prove an impossibility even for the core, under the assumption that it is nonempty. We therefore consider approximate versions of the least core.

Given a cooperative game, let $\mathcal{D}$ be a distribution over 2^N, and let $e^\star$ be the subsidy defined by the least core — the optimal solution to Equation (8.1). A payoff allocation $\mathbf{x}$ is in the *δ-probable least core* if and only if

$$\Pr_{S\sim\mathcal{D}}\left[\sum_{i\in S} x_i + e^\star \geq v(S)\right] \geq 1-\delta.$$

That is, the least core constraint is violated with probability at most δ when coalitions are drawn from $\mathcal{D}$.

We have the following result, whose proof appears in Appendix 8.7.

Theorem 45. *Given a cooperative game* (N, v)*, distribution* $\mathcal{D}$ *over* 2^N*, and* $\delta, \Delta > 0$*, solving the linear program* (8.1) *over* $O((n + \log(1/\Delta))/\delta^2)$ *coalitions sampled from* $\mathcal{D}$ *gives a payoff allocation in the* δ*-probable least core with probability at least* $1 - \Delta$.

It may seem surprising that solving the linear program (8.1) with respect to a subset of the coalitions gives a guarantee with respect to the unknown subsidy $e^\star$. But the estimated deficit $\hat{e}$ with respect to a subset of coalitions (that is, a subset of constraints) satisfies $\hat{e} \leq e^\star$ due to monotonicity.

Also note that the choice of $\mathcal{D}$ rests with the algorithm designer. In other words, we can sample coalitions from any distribution $\mathcal{D}$ and compute an allocation in the least core on the sample; the probable least core guarantee would then hold with respect to that same $\mathcal{D}$. In particular, if the uniform distribution over coalitions is used, the guarantee holds with respect to a $(1 - \delta)$-fraction of all coalitions.

While Theorem 45 is encouraging, a potential drawback is that the algorithm's running time is polynomial in the number of players n. While this is an exponential improvement over naïve least core computation, it can still be a nonstarter when the players are features in a high-dimensional space or data points. We therefore define the (ϵ, δ)*-probably approximate least core* to be payoff allocations such that

$$\Pr_{S \sim \mathcal{D}}\left[\sum_{i \in S} x_i + e^\star + \epsilon \geq v(S)\right] \geq 1 - \delta.$$

With this additional relaxation, we can obtain running time that is polynomial in $\log(n)$; the proof is relegated to Appendix 8.8.

Theorem 46. *Given a cooperative game* (N, v)*, distribution* $\mathcal{D}$ *over* 2^N*, and* $\delta, \Delta, \epsilon > 0$*, solving the linear program* (8.1) *over*

$$O\left(\frac{\tau^2\left(\log n + \log\left(\frac{1}{\Delta}\right)\right)}{\epsilon^2 \delta^2}\right)$$

coalitions sampled from $\mathcal{D}$*, where* $\tau = \frac{\max_S v(S)}{\min_{S \neq \emptyset} v(S)}$*, gives a payoff allocation in the* (ϵ, δ)*-probably approximate least core with probability at least* $1 - \Delta$.

We note that τ may be considered a constant in general. For example, in multiclass classification it is no bigger than $\frac{1}{1/m} = m$, where m is the number of classes.

8.3.2 Computing the Nucleolus

The probably approximate least core can be seen as requiring the deficit of "most" coalitions to be approximately at most the maximum deficit $e^\star$ that defines the least core. In the (unique) nucleolus, though, that deficit is associated only with the worst-off coalition. It is natural to ask, instead, that the deficit of "most" coalitions be approximately their *own* deficit under the nucleolus allocation.

Formally, as before fix a cooperative game and a distribution $\mathcal{D}$. Denote by $d^\star(S)$ the deficit of coalition $S \subseteq N$ under the unique nucleolus allocation. A payoff allocation $\mathbf{x}$ is in the

(ϵ, δ)*-probably approximate nucleolus* if and only if

$$\Pr_{S \sim \mathcal{D}} \left[\left| \sum_{i \in S} x_i + d^\star(S) - v(S) \right| \leq \epsilon \right] \geq 1 - \delta.$$

Unfortunately, it turns out that any algorithm that computes the probably approximate nucleolus requires a number of samples that is *exponential* in the number of players n — a doubly exponential increase over the probably approximate least core! — as the following theorem shows.

Theorem 47. *Let* $n \geq 9$*,* $\epsilon < 1/50$*,* $\delta < 1/200$ *and* $\Delta < 4/5$*. Then any deterministic algorithm that for all games* (N, v) *on* n *players, and all distributions* $\mathcal{D}$ *on* N*, computes a payoff allocation in the* (ϵ, δ)*-probably approximate nucleolus with probability at least* $1 - \Delta$ *requires access to the values of* $\Omega(2^{n/3})$ *coalitions sampled from* $\mathcal{D}$*.*

The importance of Theorem 47 lies in the practical guidance it provides. Indeed, the stark contrast between Theorem 46 and 47 suggests that we should focus on approximations of the least core, as natural approximations of the (stronger notion of) nucleolus are essentially beyond reach. Even though the theoretical result is worst-case in nature, we show in Section 8.5 that its implication holds in practice.

We also note that the theorem statement deals with algorithms that are deterministic, up to the random sampling of coalitions from $\mathcal{D}$. However, it is not difficult to extend the theorem to deal with randomized algorithms too, at the cost of complicating the proof further. Moreover, the constants in the theorem statement can certainly be improved, but we do not view their exact values as being important.

8.4 Interlude: A Comparison of the Core and the Shapley Value

Now that we have established that it is viable to compute the least core, we turn to the conceptual part of our argument. Before going into how the least core and the Shapley value differ (we include a comment on when the two are known to coincide in Section 8.5), one thing to note about the least core is that it is a set of solutions, whereas the Shapley value is a point solution concept. To compare the two conceptually (and experimentally as well), we break ties by selecting the payoff allocation in the least core with the smallest ℓ_2 norm. This is known as the *egalitarian least core*.

Axiomatic Properties. The Shapley value has almost always been justified through its four axiomatic properties [61, 73, 87, 193, 269]: (i) efficiency (ii) symmetry (iii) null player (iv) linearity. If we accept this argument, then the egalitarian least core is quite attractive in satisfying all but the last axiom (linearity).

While the least core's lack of linearity is ostensibly a disadvantage, it is unclear to us why it is an essential property for importance scores. The necessity of linearity is commonly justified by defining a cooperative game for each test point with the coalitional value being the model accuracy with respect to that point. And so, one would desire that summing the importance scores across these games would yield the score of the game corresponding to the entire test set [115]. However, in this vein, one can simply define a different game, with the coalitional value being the model accuracy with respect to the entire test set, in the very beginning, thus obviating the need for this property to hold.[1]

By contrast, the stability axiom, which the egalitarian least core does satisfy, is crucial if we are to adopt the economic motivation behind data valuation, as described in data market papers such as that of Ghorbani and Zou [115]. Put another way, if the goal is to output scores that reflect and may be *interpreted* as *economically plausible payments* in a competitive market, then the scores should be such that every coalition is compensated for at least its market value. This is so that the agents in the coalitions, who are rational, do not elect to leave the grand coalition. Contrast this with the Shapley value, which confers only a generic notion of "importance" (where relatively bigger means more "important") and may not necessarily correspond to an economically feasible set of payoffs (as we will see in the experiments).

Behavioral Studies. Studies in behavioral game theory have found the core to be predictive of payment distribution in market settings, suggesting that people perceive the core as a fair scheme for dividing up the total payoffs; by contrast, the Shapley value has received "weaker empirical support" [294]. This is an especially compelling reason to prefer the core over the Shapley value: since the *stakeholders* involved with machine learning are often people, it is imperative to employ a solution concept that is consistent with their behavior and intuition [44, 172]. Indeed, while much is still unclear as to how to assign "importance scores" in interpretability so as to truly aid stakeholders, there exists ample economic literature on how to equitably pay people and the core is one such prominent concept, which we champion as a principled way to assign these scores in the valuation setting.

Negative Computational Results for Shapley. Similar to our negative result for the nucleolus in Theorem 47, prior work has also produced negative results for the computation of the Shapley value. Indeed, the Shapley value is difficult to approximate, not to mention compute exactly. Informally, Bachrach et al. [22] show that no polynomial-time randomized algorithm can build a confidence interval with small accuracy. And Balkanski et al. [34] show that there exist games such that the Shapley value cannot be approximated from samples over the uniform distribution.

In light of these negative results, the latest state of the art algorithms for computing the Shapley value [115, 156] either turn to simpler Monte-Carlo approaches that do not enjoy theoretical guarantees [115] or more complicated algorithms that leverage assumptions such as sparsity to obtain sizable savings in sample complexity [156]. By contrast, we provide a simpler algorithm for computing the approximate least core with probable guarantees.

But do these theoretical results translate into practice? In the next section we show, among other things, that in low-resource settings (where the algorithm has limited computational power) our least core algorithm outperforms state-of-the-art algorithms for the Shapley value, thereby bolstering the computational case in favor of the least core.

[1] We do note that the core satisfies "approximate linearity" in the following sense: An e_1-core under coalition function v_1 and an e_2-core under coalition function function v_2 can be combined into an allocation that satisfies the $(e_1 + e_2)$-core under coalition function $v_1 + v_2$ (though certainly the least core could be better than just summing the least core allocations across the two games).

8.5 Empirical Results

The purpose of this section is twofold. First, we empirically verify our theoretical conclusions about the computability of the least core and nucleolus (which are worst case in nature). Second, we compare the algorithms that one would use to approximate the Shapley value with that for least core.

Our experiments are conducted on feature valuation and data valuation tasks. Following previous work in the area, our primary aim is to use these tasks to confirm that least core values are predictive of importance, albeit in an indirect way (as the ultimate test of human-centered AI must be how the system interacts with people).

8.5.1 Feature Valuation

We choose three smaller-scale UCI datasets [95] that have 10–14 features: this makes it computationally feasible to train a logistic regression classifier on all possible subsets of features and to compute the *exact* Shapley and least core values. To define the cooperative game, the players are the features and the value of a coalition is the test accuracy of a logistic regression classifier that is trained on those features. The three real-world datasets are of different domains: *house* (classifying the party of Congressmen based on their votes on issues), *medical* (predicting the presence of breast cancer based on features of images, and *chemical* (classifying the origin of wine based on chemical analysis).

To empirically verify Theorem 45 from Section 3 (which deals with the probable least core), we sample a small fraction of coalitions uniformly at random from all possible coalitions, and compute the least core by restricting Equation (8.1) to these coalitions. We then determine what fraction of all coalitions satisfy the least core constraints with respect to the true deficit $e^{\star}$ — that gives us *accuracy* $1 - \delta$, which, in turn, leads to δ-probable least core. To obtain error bars, we repeat this ten times. As can be seen in Figure 8.1, even with a small fraction of sampled coalitions, the resultant allocations are δ-probable least core allocations with very small δ.

Theorem 47, by contrast, asserts that many samples are needed to compute the probably approximate nucleolus. Since this is a worst-case result, one may wonder whether it holds in practice. To check this, we apply the same methodology as above. As can be seen in Figure 8.1, even when a sizable fraction of samples are used to compute the (ϵ, δ)-probably approximate nucleolus , most coalitions do not satisfy its constraints.

8.5.2 Data Valuation

Our second set of experiments deals with data valuation. We focus on low-resource settings in which we assume the analyst who is looking to understand data importance has access to limited computational resources (e.g., few cores, no pun intended). We examine the performance of existing algorithms that one would use. To compare, we elect to fix the sample complexity (the number of $v(S)$ queries) that the algorithms are permitted to use. This sidesteps comparing the actual runtimes of the algorithms, which may vary depending on the details of the implementation. The two data valuation Shapley algorithms we compare against are TMC [115] and Group

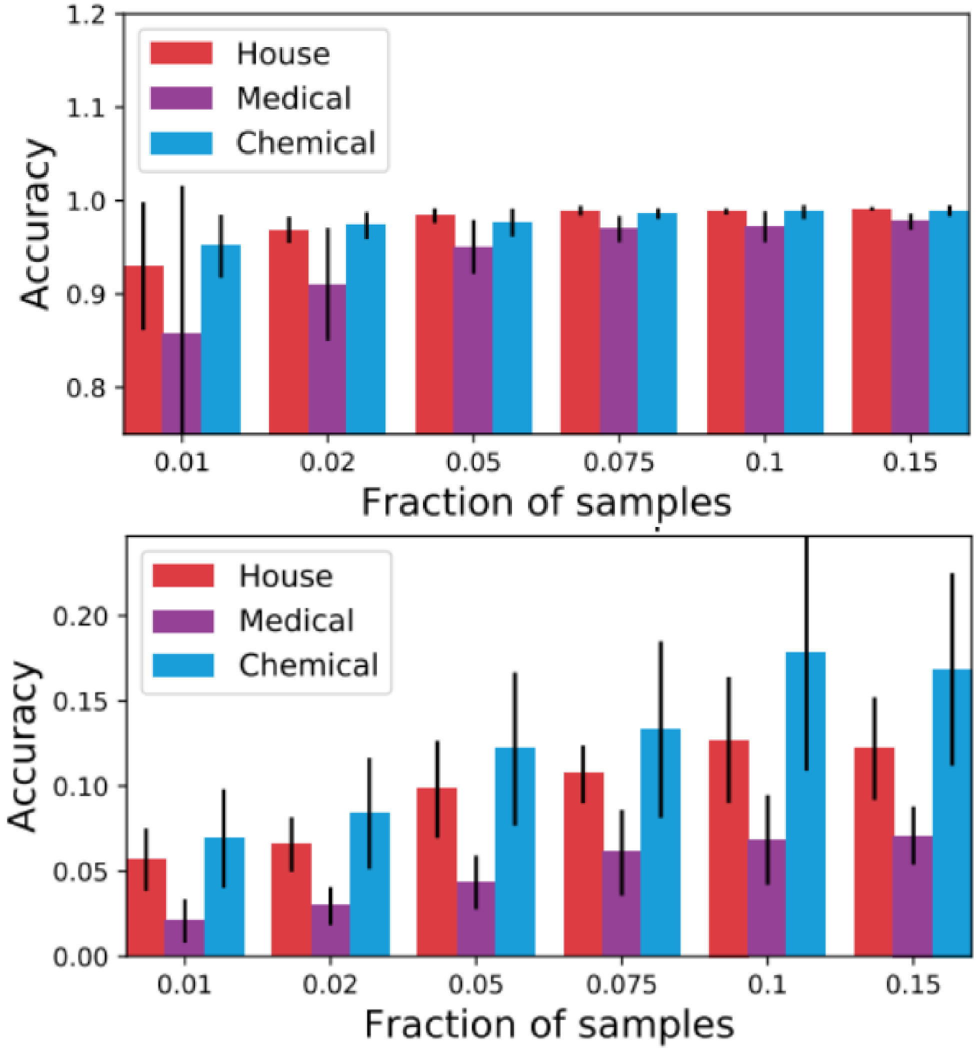


Figure 8.1: Top Panel: Least core accuracy (satisfaction of the core constraint) over coalitions. Bottom Panel: Nucleolus accuracy (satisfaction of the core constraint) over coalitions ($\epsilon = 0.01$).

Testing [156]. Please note that the experiments we conduct below emulate the current, gold standard for evaluating feature or data valuation methods, which is to add or remove features or data as ranked by the method and use the resultant model accuracy as an indicator of the "goodness" of the valuation.

Data Removal. We emulate the data removal experiments as described by Ghorbani and Zou [115] [115]. In this set of experiments, the data is ranked from most valuable to the least valuable using the solution concepts, and the model performance is charted as the most valuable/least valuable five percent of the data is removed at a time. In addition to the two Shapley algorithms we also include two baselines: leave one out (LOO), defined as $v(N) - v(N \setminus \{i\})$ for each player i, and random score assignment.

For the synthetic data generation, we sample 200 data points from 50-dimensional Gaussian, the 50-dimensional parameters are sampled from a uniform distribution and the feature-label relationship is set to be linear. To define the cooperative game, we take the players to be the data and the value of a coalition to be the test accuracy of the model trained only on the data in the coalition. The model used here is logistic regression; we relegate results for neural networks to Appendix 8.11.2. We repeat the procedure 20 times and obtain 95 percent confidence intervals for the mean model performance.

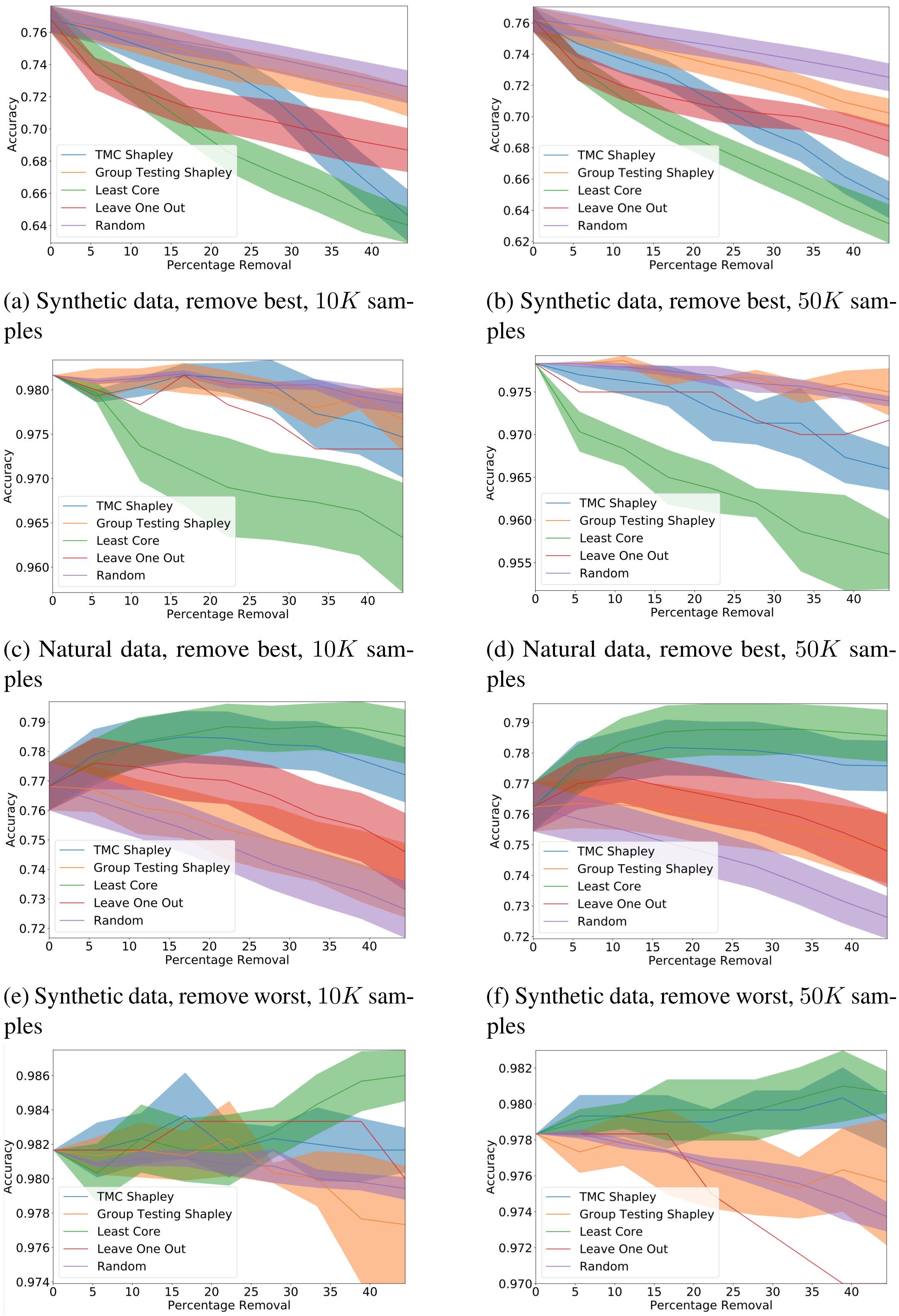
For the natural dataset, we use the dog-vs-fish classification dataset as in the work of Koh and Liang [166] and Ghorbani and Zou [115]. We randomly sample 600 data points and obtain features of the images using Inception network. The model used for training is logistic regression and we vary the budget as before. This entire process is repeated five times to obtain the error bars.

We experiment with a budget of $5K, 10K, 25K, 50K$ for samples as in a low-resource setting. As a point of reference, for the synthetic data experiment, computing the exact least core uses 2^{200} samples. The TMC Algorithm with a stopping threshold of less than one percent change in the estimated Shapley value uses $2.17M$ samples when run until convergence. For the Group Testing Algorithm, using the sample complexity derived, running till convergence uses $11.05M$ samples.

As can be seen in Figure 8.2 (with similar figures for other parameter settings given in Appendix 8.11.2), the least core algorithm compares favorably with the Shapley algorithms in terms of predicting the most and least important (in a sense) data points in these settings. Specifically, the least core's performance is significantly better than the baselines in the synthetic setting, whereas in the natural setting it is slightly better than Shapley value computation via the stronger of the two algorithms.

It is worth pointing out that the formulation of least core is such that it captures a group measure of value, whereas the Shapley value is more of an individual measure. Therefore, this data removal setup should *conceptually* favor Shapley, and yet the least core outperforms it to some degree.

As one more sanity check, we conduct an experiment studying the percentage of utility allocated by the core to noisy data. We divide the dataset into two: a clean portion and a noised portion. We increase the Gaussian noise added to the noised portion and compute the percentage of utility allocated by the core to the clean data. As expected and seen in Figure 8.3, with higher noise, the noised data become less "valuable" and are thus allocated a lower percentage of the overall utility by the core.



(a) Synthetic data, remove best, $10K$ samples

(b) Synthetic data, remove best, $50K$ samples

(c) Natural data, remove best, $10K$ samples

(d) Natural data, remove best, $50K$ samples

(e) Synthetic data, remove worst, $10K$ samples

(f) Synthetic data, remove worst, $50K$ samples

(g) Natural data, remove worst, $10K$ samples

(h) Natural data, remove worst, $50K$ samples

Figure 8.2: Curves of logistic regression test performance when the best and worst data points ranked according to the solution concepts are removed. In (a)–(d) the best data points are removed: the steeper the drop, the better. In (e)–(h) the worst data points are removed: the sharper the rise, the better.

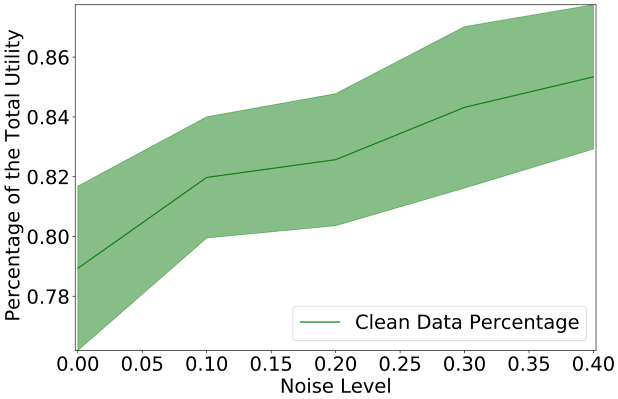


Figure 8.3: Plotting noise level against percentage of total utility assigned to clean data.

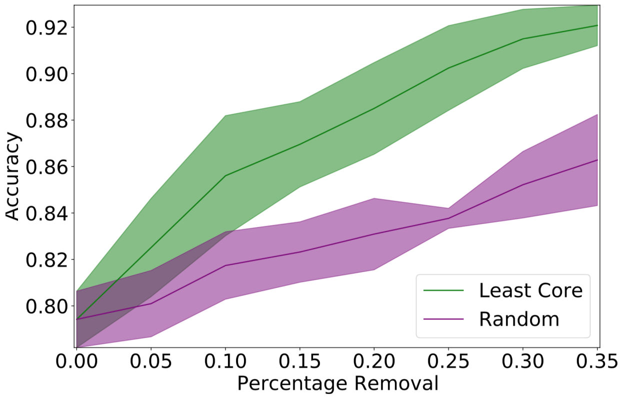


Figure 8.4: Test performance as we correct more and more training data guided by the least core vs. random selection.

Fixing Mislabeled Data. We perform another set of experiments to verify that the magnitude of the least core values strongly correlate with the importance of the data point. In this experiment, we assume we have a dataset with flipped labels and would like to use the importance scores assigned to expedite the correction of "flipped" data points, which should correspond to the lower scores. The specific dataset we use is the Enron Dataset, as in previous work [115, 166]. In total, 1000 data points are used for training a Naive Bayes model which takes as input a bag-of-words representation of emails. We randomly flip the label for twenty percent of the data and allot a budget of 5000 samples for computing the solution concepts. The coalitional values are defined as performance on the validation set, and then the final performance in the plot is assessed on the test set. As can be seen in Figure 8.4, the least core values are much better at picking out lower quality data points than random selection.

Is the Approximate Shapley value in the Approximate Least Core? It is known that the Shapley value coincides with the egalitarian core for convex games, where there is a super-additive effect in players coming together. This effect is not typically present in what we call "supervised-learning" games, in which there are diminishing returns as more and more data or features are added and used. However, in theory it may still be the case that the two solutions usually coincide, which would make it redundant to discuss the core. We therefore test, in the valuation experiments mentioned above, whether approximate Shapley values are close to being in the approximate least core. Our results suggest that this is not the case and therefore the approximate Shapley cannot serve as a proxy for the least core. Details are relegated to Appendix 8.11.2.

8.6 Discussion

In our paper, we provide theoretical and empirical results, along with with conceptual arguments (Section 8.4), that suggest the least core is a principled, alternative means of doing credit assignment in ML. Currently, it appears that virtually all papers on feature and data valuation use the Shapley value for this purpose. In light of the many uses of the core as an economically plausible method of payoff assignment, we introduce this alternative approach to the AI community in the hope invoking further discussion on when and why one solution concept is to be preferred.

Lastly, we wish to note that outside of the comparison of solution concepts, one limitation that is shared by *both* the core and the Shapley value is that they are not suitable for non-additive models [172]. This problem is an artifact of the game setup and not the solution concept. It is another important issue that the community would need to come to a consensus on.

8.7 Proof of Theorem 45

This proof is a direct extension of the proof of Theorem 1 of Balkanski et al. [34]. Like them, we employ the following known lemmas [255].

Lemma 61. *Let $\mathcal{H}$ be a function class from $\mathcal{X}$ to $\{-1, 1\}$, and let f be the true underlying function. If $\mathcal{H}$ has VC-dimension d, then with*

$$m = O\left(\frac{d + \log\left(\frac{1}{\Delta}\right)}{\delta^2}\right)$$

i.i.d. samples $\mathbf{x}^1, ..., \mathbf{x}^m \sim \mathcal{D}$,

$$\left|\Pr_{\mathbf{x}\sim D}[h(\mathbf{x}) \neq f(\mathbf{x})] - \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{h(\mathbf{x}^i)\neq f(\mathbf{x}^i)}\right| \leq \delta$$

for all $h \in \mathcal{H}$ and with probability $1 - \Delta$ over the samples.

Lemma 62. *The function class $\{\mathbf{x} \mapsto sign(\mathbf{w} \cdot \mathbf{x}) : \mathbf{w} \in \mathbb{R}^n\}$ has VC-dimension n.*

We now turn to the proof. Given a coalition S sampled from $\mathcal{D}$, we convert it into a vector $\mathbf{y}^S = (\mathbf{x}^S, -v(S), 1)$ where $x_i^S = 1$ if $i \in S$ and $x_i^S = 0$ otherwise.

Consider a linear classifier h define by $\mathbf{w}^h = (\mathbf{z}, 1, e)$ where $\mathbf{z} \in \mathbb{R}^n$ and $e \in \mathbb{R}$. If sign$(\mathbf{w}^h \cdot \mathbf{y}^S) = 1$ then $\sum_{i\in S} z_i - V(S) + e \geq 0$. And if there exist a linear classifier h that satisfies this property for all coalitions $S \in 2^N$, and in addition $\mathbf{z}$ is efficient, then it represents a payoff allocation in the e-core. This allows us to define a class of functions that contains the e-core for all e. This class is:

$$\mathcal{H} = \left\{ \mathbf{y} \mapsto \text{sign}(\mathbf{w} \cdot \mathbf{y}) : \; \mathbf{w} = (\mathbf{z}, 1, e), \mathbf{z} \in \mathbb{R}^n, e \in \mathbb{R}, \sum_{i=1}^{n} z_i = v(N) \right\}.$$

This class $\mathcal{H}$ is a subset of the class of all linear classifiers of dimension $n + 2$ and thus, by Lemma 62, it has VC-dimension at most $n + 2$.

Now, suppose that we run the linear program (8.1) on our samples $S_1, \ldots, S_m$, which gives us a payoff allocation $\hat{\mathbf{z}}$ and a value $\hat{e}$. Define the corresponding classifier $\hat{h}$; notice that $\hat{h}(\mathbf{y}^{S_i}) = 1$ for all $i = 1, \ldots, m$. In addition, let $\mathbf{z}^\star$ be a payoff allocation in the least core, and $e^\star$ the required subsidy, and define the corresponding classifier $f^\star$. It holds that $f^\star(\mathbf{y}^S) = 1$ for all $S \in 2^N$.

By Lemma 61 we have uniform convergence for all classifiers with probability $1 - \Delta$, and in particular for $\hat{\mathbf{h}}$ it holds that

$$\begin{aligned}
\Pr_{S\sim\mathcal{D}}\left[\sum_{i\in S} \hat{z}_i - v(S) + e^\star \geq 0\right] &\geq \Pr_{S\sim\mathcal{D}}\left[\sum_{i\in S} \hat{z}_i - v(S) + \hat{e} \geq 0\right] \\
&= 1 - \Pr_{S\sim\mathcal{D}}\left[\text{sign}(\mathbf{w}^{\hat{h}} \cdot \mathbf{y}^S) = -1\right] \\
&= 1 - \Pr_{S\sim\mathcal{D}}\left[\hat{h}(\mathbf{y}^S) \neq f^\star(\mathbf{y}^S)\right] \\
&= 1 - \left(\Pr_{S\sim\mathcal{D}}\left[\hat{h}(\mathbf{y}^S) \neq f^\star(\mathbf{y}^S)\right] - \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{\hat{h}(\mathbf{y}^{S_i}) \neq f^\star(\mathbf{y}^{S_i})}\right) \\
&\geq 1 - \delta
\end{aligned}$$

where the first transition holds because $\hat{e} \leq e^\star$ and the fourth transition holds because $\hat{h}$ and $f^\star$ agree on $S_1, \ldots, S_m$. □

8.8 Proof of Theorem 46

This proof directly extends the proof of Theorem 5 of Balkanski et al. [34]. Like them, we use the following result [255].

Lemma 63. *Let $\mathcal{H} = \{\mathbf{w} : ||\mathbf{w}||_1 \leq B\}$ be the hypothesis class, and $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ be the examples domain. Suppose $\mathcal{D}_Z$ is a distribution over $\mathcal{Z}$ s.t $||\mathbf{x}||_\infty \leq R$. Let the loss function $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$ be of the form $\ell(\mathbf{w}, (\mathbf{x}, y)) = \phi(\langle \mathbf{w}, \mathbf{x}\rangle, y)$ and $\phi : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$ is such that for all $y \in \mathcal{Y}$, the scalar function $a \to \phi(a, y)$ is ρ-Lipschitz and such that $\max_{a\in[-BR,BR]} |\phi(a, y)| \leq c$. Then for any $\Delta \in (0, 1)$, with probability of at least $1 - \Delta$ over the choice of an iid sample of size m, $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_m, y_m)$:*

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_Z}[\ell(\mathbf{w}, (\mathbf{x}, y))] \leq \frac{1}{m}\sum_{i=1}^{m} \ell(\mathbf{w}, (\mathbf{x}^i, y^i)) + 2\rho BR\sqrt{\frac{2\log(2d)}{m}} + c\sqrt{\frac{2\log(2/\Delta)}{m}}.$$

for all $\mathbf{w} \in \mathcal{H}$.

We also require the observation that if an (ϵ, δ)-probably approximate least core holds in expectation, then it is likely to hold.

Lemma 64. *For any* $\epsilon > 0$*,* $\delta < 1$ *and* e*-core allocation* $\mathbf{x}$ *computed from samples,*

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}}\left[\left[1 - \frac{\sum_{i\in S} z_i + e}{v(S)}\right]_+\right] \leq \frac{\epsilon\delta}{1+\epsilon} \Rightarrow \Pr_{S\sim\mathcal{D}}\left[\sum_{i\in S} z_i + e^\star + \epsilon \geq v(S)\right] \geq 1 - \delta.$$

Proof. Recall Markov's inequality: for $a > 0$, random variable $X \geq 0$,

$$\Pr[X \leq a] \geq 1 - \frac{\mathbb{E}[X]}{a}.$$

To use it, let $a = \frac{\epsilon}{1+\epsilon}$ and define a nonnegative random variable

$$X = \left[1 - \frac{\sum_{i\in S} z_i + e}{v(S)}\right]_+.$$

Then event $X \leq a$ is such that

$$\begin{aligned}
X \leq a \Leftrightarrow 1 - \frac{\sum_{i\in S} z_i + e}{v(S)} &\leq \frac{\epsilon}{1+\epsilon} \\
\Leftrightarrow \sum_{i\in S} z_i + e &\geq \frac{1}{1+\epsilon} v(S) \\
\Leftrightarrow \sum_{i\in S} z_i + e + \frac{\epsilon}{1+\epsilon} v(S) &\geq v(S) \\
\Rightarrow \sum_{i\in S} z_i + e + \epsilon &\geq v(S) \\
\Rightarrow \sum_{i\in S} z_i + e^\star + \epsilon &\geq v(S)
\end{aligned}$$

where the penultimate step uses $v(S) \leq 1$ for all $S \subseteq N$, and the last step uses that $e^\star \geq e$ since e is the least core value obtained from only a sample of all coalitional constraints.

We conclude that

$$\Pr\left[\sum_{i\in S} z_i + e^\star + \epsilon \geq v(S)\right] \geq \Pr[X \leq a] \geq 1 - \frac{\mathbb{E}[X]}{a} \geq 1 - \frac{\delta a}{a} = 1 - \delta.$$

□

Turning to the theorem's proof, in order to use Lemma 63, we begin by bounding the L_1 norm of every allocation and e in the e-core to obtain B.

Suppose $\mathbf{z}$ is an allocation in the e-core, then $||(\mathbf{z}, e)||_1 = v(N) + e$. This holds because $z_i \geq 0$ for all $i \in N$ and, by efficiency, $||\mathbf{z}||_1 = v(N)$. Therefore:

$$||(\mathbf{z}, e)||_1 = v(N) + e \leq v(N) + \max_S v(S) \leq 2 \max_S v(S)$$

Then, we can take our hypothesis class to be:

$$\mathcal{H} = \left\{ \mathbf{z} \in \mathbb{R}^n : \; ||\mathbf{z}||_1 \leq 2 \max_S v(S) \right\}$$

Given $S \sim \mathcal{D}$, define the corresponding $\mathbf{x}^S = (\frac{\mathbb{1}_{i \in S}}{v(S)}, \frac{1}{v(S)})$ and the label to be $y^S = 1$. This allows us define to $\mathcal{D}_Z$ to be the uniform distribution over all $(\mathbf{x}^S, y^S)$ pairs. Next, suppose we obtain m samples $S_1, \ldots, S_m$ from $\mathcal{D}$, the uniform distribution over all coalitions, we may again run the linear program (8.1) on the m samples, which gives us a payoff allocation $\hat{\mathbf{z}}$ and a value $\hat{e}$. We take our classifier to be of the form $\mathbf{w} = (\hat{\mathbf{z}}, \hat{e})$ and we may define its loss ℓ to be:

$$\begin{aligned}
\ell(\mathbf{w}, (\mathbf{x}^S, y^S)) &= \ell \left((\hat{\mathbf{z}}, \hat{e}), \left(\left(\frac{\mathbb{1}_{i \in S}}{v(S)}, \frac{1}{v(S)} \right), y^S \right) \right) \\
&= \left[y^S - (\hat{\mathbf{z}}, \hat{e}) \cdot \left(\frac{\mathbb{1}_{i \in S}}{v(S)}, \frac{1}{v(S)} \right) \right]_+ \\
&= \left[1 - \frac{\sum_{i \in S} \hat{z}_i + \hat{e}}{v(S)} \right]_+ .
\end{aligned} \tag{8.2}$$

Now, we may utilize Lemma 63 with the remaining variables being $R = \frac{1}{\min_{S \neq \emptyset} v(S)}$, $B = 2 \max_S v(S)$, $\phi(a, y) = [y - a]_+$, $\rho = 1$ and $c = 1 + 2\tau$. This is legal because, ignoring the empty set, by definition of $\mathbf{x}^S$, $||\mathbf{x}^S||_\infty \leq \frac{1}{\min_{S \neq \emptyset} v(S)}$. By definition of the hypothesis class, $||(\mathbf{z}, e)||_1 \leq 2 \max_S v(S)$ for all $(\mathbf{z}, e) \in \mathcal{H}$. $\phi(a, y) = [y - a]_+$ is 1-Lipschitz as:

$$\begin{aligned}
[y - a_1]_+ - [y - a_2]_+ &= \max\{y - a_1, 0\} - \max\{y - a_2, 0\} \\
&= \frac{|y - a_1| + y - a_1}{2} - \frac{|y - a_2| + y - a_2}{2} \\
&= \frac{|y - a_1| - |y - a_2| + a_2 - a_1}{2} \\
&\leq \frac{|y - a_1 - (y - a_2)| + a_2 - a_1}{2} \\
&\leq |a_2 - a_1|
\end{aligned}$$

Lastly, because our example domain $\mathcal{Z}$ is such that $\mathcal{Y} = \{1\}$. We may obtain upper bound c: $c = \max_{a \in [-BR, BR]} |\phi(a, y)| = \max_{a \in [-BR, BR]} [1 - a]_+ \leq (1 - -BR) = 1 + BR = 1 + 2\tau$.

Moreover, since for all S_t in our sample it holds that $\sum_{i \in S_t} \hat{z}_i + \hat{e} \geq v(S)$, Equation (8.2) implies that

$$\frac{1}{m} \sum_{t=1}^{m} \ell \left((\hat{\mathbf{z}}, \hat{e}), \left(\left(\mathbf{x}^{S_t}, \frac{1}{v(S_t)} \right), 1 \right) \right) = 0.$$

Therefore by Lemma 63,

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[l(\mathbf{w},(\mathbf{x},y))] = \mathbb{E}_{S\sim D}\left[\left[1-\frac{\sum_{i\in S}\hat{\mathbf{z}}_i+\hat{e}}{v(S)}\right]_+\right]$$

$$\leq 0+2\cdot 1\cdot 2\tau\sqrt{\frac{2\log(2(n+1))}{m}}+(1+2\tau)\sqrt{\frac{2\log(2/\Delta)}{m}}$$

Using Lemma 64, we need the number of samples m to be such that

$$4\tau\sqrt{\frac{2\log(2(n+1))}{m}}+(1+2\tau)\sqrt{\frac{2\log(2/\Delta)}{m}}\leq\frac{\delta\epsilon}{1+\epsilon},$$

and we get that

$$O\left(\frac{\tau^2\left(\log n+\log\left(\frac{1}{\Delta}\right)\right)}{\epsilon^2\delta^2}\right)$$

samples suffice. □

8.9 Proof of Theorem 47

On a high level, we will construct a set of cooperative games $\mathcal{G}$ over the same set of players N, and a distribution $\mathcal{D}$ over the coalitions, such that no deterministic algorithm can compute a payoff allocation in the (ϵ,δ)-approximate nucleolus with probability $1-\Delta$ using $m\leq\frac{1}{6}\cdot 2^{n/3+1}$ samples with respect to *every* game in $\mathcal{G}$.

The idea of the proof is as follows. We construct the class of games $\mathcal{G}$ in a way that it is likely to observe $v(S_i)=0$ for the coalitions $S_1,\ldots,S_m$ sampled from $\mathcal{D}$. Lemma 65 shows that at least half of the games in our class are consistent with such an observation. But Lemma 67 asserts that *any* payoff allocation would be in the (ϵ,δ)-probably approximate nucleolus of only a small fraction of the games in $\mathcal{G}$. Intuitively, then, when such an input is observed, the algorithm does not have enough information about the underlying game and is likely to violate the (ϵ,δ)-probably approximate nucleolus requirement. In the theorem's proof itself, we formalize this intuition by first assuming that the game itself is drawn from a uniform distribution over $\mathcal{G}$; the theorem statement follows from an averaging argument.

Formally, the class of games $\mathcal{G}$ is defined as follows. Let N be a set of n players; we assume without loss of generality that n is divisible by 3. Let C_1 be a set of 3 players $\{i,j,k\}$. Define C_2,C_3,C_4 to be sets of $n/3-1$ players such that $C_1\cup C_2\cup C_3\cup C_4=N$. Each cooperative game G_{C_1,C_2,C_3,C_4} in our class $\mathcal{G}$ is such that $v(S)=1$ if $\{i,j\}\cup C_2\subseteq S$ or $\{i,k\}\cup C_3\subseteq S$ or $\{j,k\}\cup C_4\subseteq S$; $v(S)=0$ otherwise. The important thing to note is that all coalitions of size $n/3+1$ have value 0, except for exactly three that have value 1: $\{i,j\}\cup C_2$, $\{i,k\}\cup C_3$, and $\{j,k\}\cup C_4$. We call C_1 the *critical set* of game G_{C_1,C_2,C_3,C_4}.

Next, we define the distribution $\mathcal{D}$ to be the uniform distribution over all coalitions of size $n/3+1$.

Lemma 65. *For any m coalitions $S_1, \ldots, S_m$ of size $n/3+1$, at least half of the games in $\mathcal{G}$ satisfy $v(S_i) = 0$ for all $i = 1, \ldots, m$.*

Proof. To count the number of such games, we can count the number of games in which the value of S_i is 1. By symmetry, the number of games in which a coalition S has value 1 is the same for all coalitions S of size $n/3+1$. Moreover, for each game in $\mathcal{G}$ there are three coalitions of size $n/3+1$ with value 1. Therefore, for each S_i, the number of games in $\mathcal{G}$ with $v(S_i) = 1$ is $3|\mathcal{G}|/\binom{n}{n/3+1}$. It follows that the number of games for which it does *not* hold that $v(S_i) = 0$ for all $i = 1, \ldots, m$ is at most $3m|\mathcal{G}|/\binom{n}{n/3+1}$. Since $\binom{n}{n/3+1} \geq 2^{n/3+1}$, by our choice of m this is at most $|\mathcal{G}|/2$. □

We next characterize the nucleolus of games in $\mathcal{G}$.

Lemma 66. *For every game $G_{C_1,C_2,C_3,C_4} \in \mathcal{G}$ and every $S \subseteq N$,*

$$d^*(S) = \begin{cases} 1/3 & S \in \{\{i,j\} \cup C_2, \{i,k\} \cup C_3, \\ & \qquad \{j,k\} \cup C_4\} \\ -\frac{|S \cap \{i,j,k\}|}{3} & \textit{otherwise} \end{cases}$$

Proof. Let us compute the least core first since we know the nucleolus lies within it. Summing the constraints of linear program (8.1) for the coalitions $\{i,j\} \cup C_2$, $\{i,k\} \cup C_3$, $\{j,k\} \cup C_4$, we get that

$$\sum_{t \in N} x_t + (x_i + x_j + x_k) \geq 3 - 3e.$$

Since $1 = \sum_{t \in N} x_t \geq x_i + x_j + x_k$, we have that $2 \geq 3 - 3e$, and hence $e \geq 1/3$. Moreover, $e = 1/3$ is achieved if $x_i = x_j = x_k = 1/3$.

We claim that this payoff allocation is the only one that achieves $e = 1/3$. Indeed, the total payoff to each of the coalitions $\{i,j\} \cup C_2$, $\{i,k\} \cup C_3$, $\{j,k\} \cup C_4$ must be at least $2/3$, which means that the payoff of players at the intersection of each pair of these coalitions must be at least $1/3$. But the intersection of each pair is exactly one of the players i, j, k.

Since the payoff allocation $\mathbf{x}$ is the unique solution to the least core program, it must be the nucleolus. The statement of the lemma directly follows. □

Lemma 66 implies that two games G_{C_1,C_2,C_3,C_4} and $G_{C_1',C_2',C_3',C_4'}$ have the same nucleolus if and only if $C_1 = C_1'$. Let us, therefore, partition $\mathcal{G}$ into *equivalence classes*, where the games in an equivalence class have the same critical set.

Lemma 67. *Any payoff allocation is in the (ϵ, δ)-probably approximate nucleolus for games from at most one equivalence class.*

Proof. Let $\mathbf{x}$ be a payoff allocation. We consider two cases, based on the number of players $i \in N$ with $x_i > \epsilon$.

Case 1: There are at least three players with $x_i > \epsilon$. Let those three players be $\{i, j, k\}$, and consider a game in $\mathcal{G}$ whose critical set is not $\{i, j, k\}$. Then there exists a player ℓ not in the critical set such that $x_\ell > \epsilon$.

Consider all coalitions of size $n/3+1$ containing ℓ but no player from the critical set. By Lemma 66, under the nucleolus of the game, all such coalitions have deficit 0, but under $\mathbf{x}$ they would have a deficit of at at most $-x_{a'} < -\epsilon$. There are $\binom{n-4}{n/3}$ such coalitions, which accounts for the following portion of all coalitions of size $n/3+1$:

$$\frac{\binom{n-4}{n/3}}{\binom{n}{n/3+1}} = \frac{(n/3+1)(2n/3-1)(2n/3-2)(2n/3-3)}{n(n-1)(n-2)(n-3)}$$
$$> (1/3 \cdot 1/2 \cdot 1/2 \cdot 1/2) = \frac{1}{24} \geq \delta.$$

Case 2: There are less than three players with $x_i > \epsilon$.

In this case, for any game in $\mathcal{G}$, $\mathbf{x}$ is such that there exists at least one player in its critical set with allocation at most ϵ. We show that this means $\mathbf{x}$ cannot satisfy the (ϵ, δ)-probably approximate nucleolus property with respect to the game.

Fix a game in $\mathcal{G}$, let the critical set of the game be $\{i, j, k\}$, and let $x_i \leq \epsilon$. Assume for the sake of contradiction that $\mathbf{x}$ satisfies the (ϵ, δ)-probably approximate nucleolus property for this game.

Consider the set of all coalitions of size $n/3+1$ that contain i, j but not k. There are $\binom{n-3}{n/3-1}$ such coalitions. We know by Lemma 66 that all but one of these coalitions have value 0 and deficit $-2/3$. In order for the property

$$\left|\sum_{i \in S} x_i + d^\star(S) - v(S)\right| \leq \epsilon \tag{8.3}$$

to hold for such coalitions, we would need their payoff to be at least $2/3 - \epsilon$.

Overall, there are at least $\binom{n-3}{n/3-1} - \delta\binom{n}{n/3+1} - 1$ many coalitions containing i, j but not k for which Equation (8.3) applies and have value 0. The middle term comes from factoring in that at most a δ fraction of all $\binom{n}{n/3+1}$ coalitions will not satisfy the probably approximate nucleolus property. By summing over the total payoffs of all such coalitions we have

$$\binom{n-3}{n/3-1}(x_i + x_j) + \binom{n-4}{n/3-2}\left(\sum_{t \notin \{i,j,k\}} x_t\right)$$
$$\geq \left(\binom{n-3}{n/3-1} - \delta\binom{n}{n/3+1} - 1\right)(2/3 - \epsilon)$$

since each player that is not i, j or k shows up $\binom{n-4}{n/3-2}$ times. Dividing by $\binom{n-3}{n/3-1}$ and using the fact that $\binom{n-4}{n/3-2}/\binom{n-3}{n/3-1} = 1/3$, we have

$$x_i + x_j + \frac{1}{3}\left(\sum_{t \notin \{i,j,k\}} x_t\right)$$
$$\geq \left(1 - \frac{\binom{n}{n/3+1}}{\binom{n-3}{n/3-1}} \cdot \delta - \frac{1}{\binom{n-3}{n/3-1}}\right)(2/3 - \epsilon).$$

With $n \geq 9$, $\frac{1}{\binom{n-3}{n/3-1}} \leq \frac{1}{15}$ and so we obtain

$$x_i + x_j + \frac{1}{3}\left(\sum_{t \notin \{i,j,k\}} x_t\right) \geq \left(\frac{14}{15} - \frac{\binom{n}{n/3+1}}{\binom{n-3}{n/3-1}}\delta\right)(2/3 - \epsilon).$$

Using efficiency, $\sum_{t \notin \{i,j,k\}} x_t = 1 - x_i - x_j - x_k$, and using the fact that

$$\frac{\binom{n}{n/3+1}}{\binom{n-3}{n/3-1}} = \frac{n(n-1)(n-2)}{(n/3+1)(n/3)(2n/3-1)} \leq 27$$

we get

$$\frac{2}{3}x_i + \frac{2}{3}x_j + \frac{1}{3} - \frac{1}{3}x_k \geq \left(\frac{14}{15} - 27\delta\right)(2/3 - \epsilon).$$

Similarly, by considering the set of all coalitions that contain i, k but not j, we see that

$$\frac{2}{3}x_i + \frac{2}{3}x_k + \frac{1}{3} - \frac{1}{3}x_j \geq \left(\frac{14}{15} - 27\delta\right)(2/3 - \epsilon).$$

Summing both inequalities, we conclude that

$$\frac{4}{3}x_i + \frac{1}{3}(x_j + x_k) + \frac{2}{3} \geq \frac{4}{3} \cdot \frac{14}{15} - 36\delta - \frac{28}{15} \cdot \epsilon + 54\delta\epsilon.$$

Since $x_j + x_k \leq 1$,

$$\frac{4}{3}x_i \geq \frac{11}{45} - 36\delta - \frac{28}{15}\epsilon + 54\delta\epsilon,$$

which is impossible for $x_i \leq \epsilon$ since $\epsilon < 1/50$ and $\delta < 1/200$. □

We are now ready to prove the theorem.

Proof of Theorem 47. Fix the set of players N. Let $\mathcal{U}$ be the uniform distribution over games in $\mathcal{G}$. Since N is fixed, we think of $\mathcal{U}$ as a distribution over characteristic functions and write $v \sim \mathcal{U}$.

Suppose that we draw coalitions $S_1, \ldots, S_m$ from $\mathcal{D}$, and v from $\mathcal{U}$. Let the payoff allocation returned by the given algorithm $\mathcal{A}$ on this input be $\mathcal{A}((S_1, v(S_1)), \ldots, (S_m, v(S_m)))$. Consider the event $\mathcal{E}$ that occurs when $\mathcal{A}((S_1, v(S_1), \ldots, (S_m, v(S_m))$ is in the (ϵ, δ)-probably approximate nucleolus of the game (N, v). We wish to upper-bound the probability of $\mathcal{E}$.

To this end, instead of drawing v from $\mathcal{U}$ directly, it will be useful to use the following generative process. First, decide whether it holds that $v(S_i) = 0$ for all $i = 1, \ldots, m$; call this event $\mathcal{F}$. If $\mathcal{F}$ occurred, condition $\mathcal{U}$ on $\mathcal{F}$ and draw v from this posterior distribution. As we will see shortly, there is no need to explicitly define the process for the case where $\mathcal{F}$ did not occur.

Denoting the complement of $\mathcal{F}$ by $\bar{\mathcal{F}}$, it holds that

$$\begin{aligned}\Pr[\mathcal{E}] &= \Pr[\mathcal{E} \mid \mathcal{F}] \cdot \Pr[\mathcal{F}] + \Pr[\mathcal{E} \mid \bar{\mathcal{F}}] \cdot \Pr[\bar{\mathcal{F}}] \\ &\leq \Pr[\mathcal{E} \mid \mathcal{F}] + \Pr[\bar{\mathcal{F}}].\end{aligned} \tag{8.4}$$

Since for every $S_1, \dots, S_m$, the probability of drawing v from $\mathcal{U}$ such that $\mathcal{F}$ occurs is the same by symmetry, we can compute $\Pr[\mathcal{F}]$ by reversing the coin flips, first drawing v and then $S_1, \dots, S_m$. Only three of the $\binom{n}{n/3+1}$ coalitions of size $n/3+1$ have non-zero value; therefore

$$\Pr[\bar{\mathcal{F}}] = 1 - \left(1 - \frac{3}{\binom{n}{n/3+1}}\right)^m < 1/10, \tag{8.5}$$

where the inequality holds for $n \geq 9$ and $m \leq \frac{1}{6} \cdot 2^{n/3+1}$.

As for $\Pr[\mathcal{E} \mid \mathcal{F}]$, by Lemma 65 at least half of the games in $\mathcal{G}$ (or, equivalently, at least half of the corresponding characteristic functions) are in the support of $\mathcal{U}$ conditioned on $\mathcal{F}$. But by Lemma 67, the payoff allocation $\mathcal{A}((S_1, v(S_1)), \dots, (S_m, v(S_m)))$ can be in the (ϵ, δ)-probably approximate nucleolus of at most one of the $\binom{n}{3}$ equivalence classes. It follows that

$$\Pr[\mathcal{E} \mid \mathcal{F}] \leq \frac{2}{\binom{n}{3}} < 1/10. \tag{8.6}$$

Plugging Equations (8.5) and (8.6) into Equation (8.4), we conclude that $\Pr[\mathcal{E}] < 1/5$.

To recap, when drawing $S_1, \dots, S_m$ from $\mathcal{D}$ and v from $\mathcal{U}$, the probability that the output of $\mathcal{A}$ is in the (ϵ, δ)-probably approximate nucleolus of $G = (N, v) \in \mathcal{G}$ is at most $1/5$. But since this is true for a random game $G \in \mathcal{G}$, there must exist a game $G^\star \in \mathcal{G}$ where the same is true when only drawing $S_1, \dots, S_m$ from $\mathcal{D}$. That is, m samples are insufficient to compute a payoff allocation in the (ϵ, δ)-probably approximate nucleolus with probability at least $1 - \Delta$ for $\Delta < 4/5$. □

8.10 Approximate Least Core Implementation

The approximate least core algorithm works as follows: compute the approximate least core value $\hat{e}$ from the samples via linear program (8.1), then minimize the ℓ_2 norm over all allocations $\mathbf{x}$ s.t $\mathbf{x}$ is in the $\hat{e}-$core:

$$\begin{array}{ll} \min & \|\mathbf{x}\|_2 \\ \text{s.t.} & \sum_{i \in N} x_i = v(N) \\ & \sum_{i \in S} x_i + \hat{e} \geq v(S) \quad \forall S \subseteq N \end{array}$$

This may be easily done with any standard optimization library and it is not hard to argue that the resultant $\mathbf{x}$ satisfies null player and symmetry in addition to efficiency.

8.11 Additional Experimental Results

8.11.1 Feature Valuation

Maximum Deficit. By definition, the maximum deficit $e^\star$ under the least core should be at most as large as that under the Shapley value. However, we wish to verify that the difference is significant in practice. To that end, we compute the least core, the Shapley value, and (as a baseline) equal

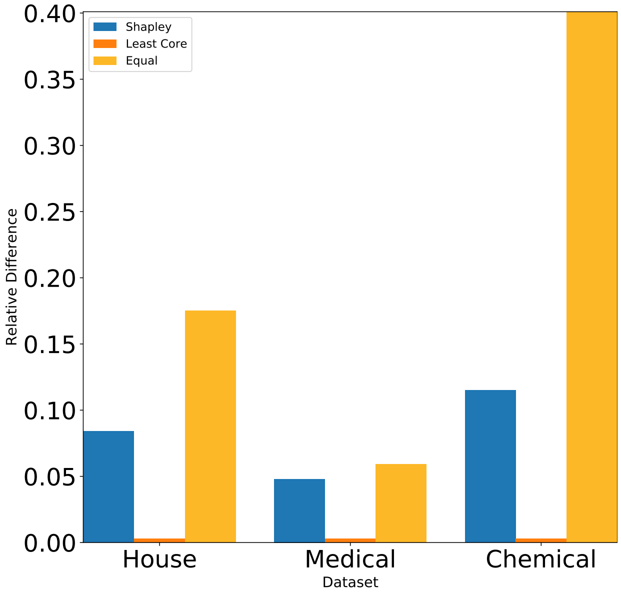


Figure 8.5: Relative difference between different solution concepts' largest deficits and the least core's largest deficit

payoffs on our three datasets. Figure 8.5 shows the difference between the maximum deficit of each of the solution concepts (including the least core itself) and the maximum deficit of the least core. It can be seen that there is a sizable gap between the Shapley value and the least core, considering that the maximum value of any coalition is 1. Note that no sampling (indeed, no randomness) is involved in this experiment.

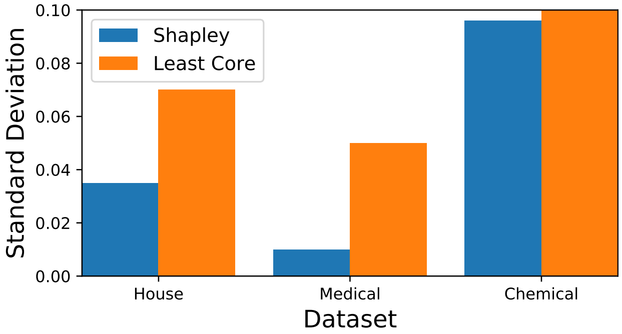


Figure 8.6: Standard deviation of solution concepts

Standard Deviation. On each of our three datasets, we compute the empirical standard deviation of payoff allocations given by the least core and the Shapley value (again no sampling is involved). Interestingly, we observe that the least core has considerably higher standard deviation and may thus be considered more discriminating; see Figure 8.6.

8.11.2 Data Valuation

8.11.2.1 Additional Experimental Details

Below include attach plots for the synthetic and natural experiments that were not included in the main body due to space constraints. We observe that in the synthetic settings, as depicted in Figures 8.7 and 8.8, the approximate least core values are decidedly better than the other importance scores. Under the natural setting, as portrayed by Figure 8.9, it seems that the change in performance is small and the least core and the Shapley value are roughly comparable across all budgets. Lastly, we note that LOO does not have an error bar in the natural experiment since all the runs are based on the same random sample of data points, and so the error bars are only due to the randomness in the sample of $v(S)$'s that are drawn to approximate the solution concepts.

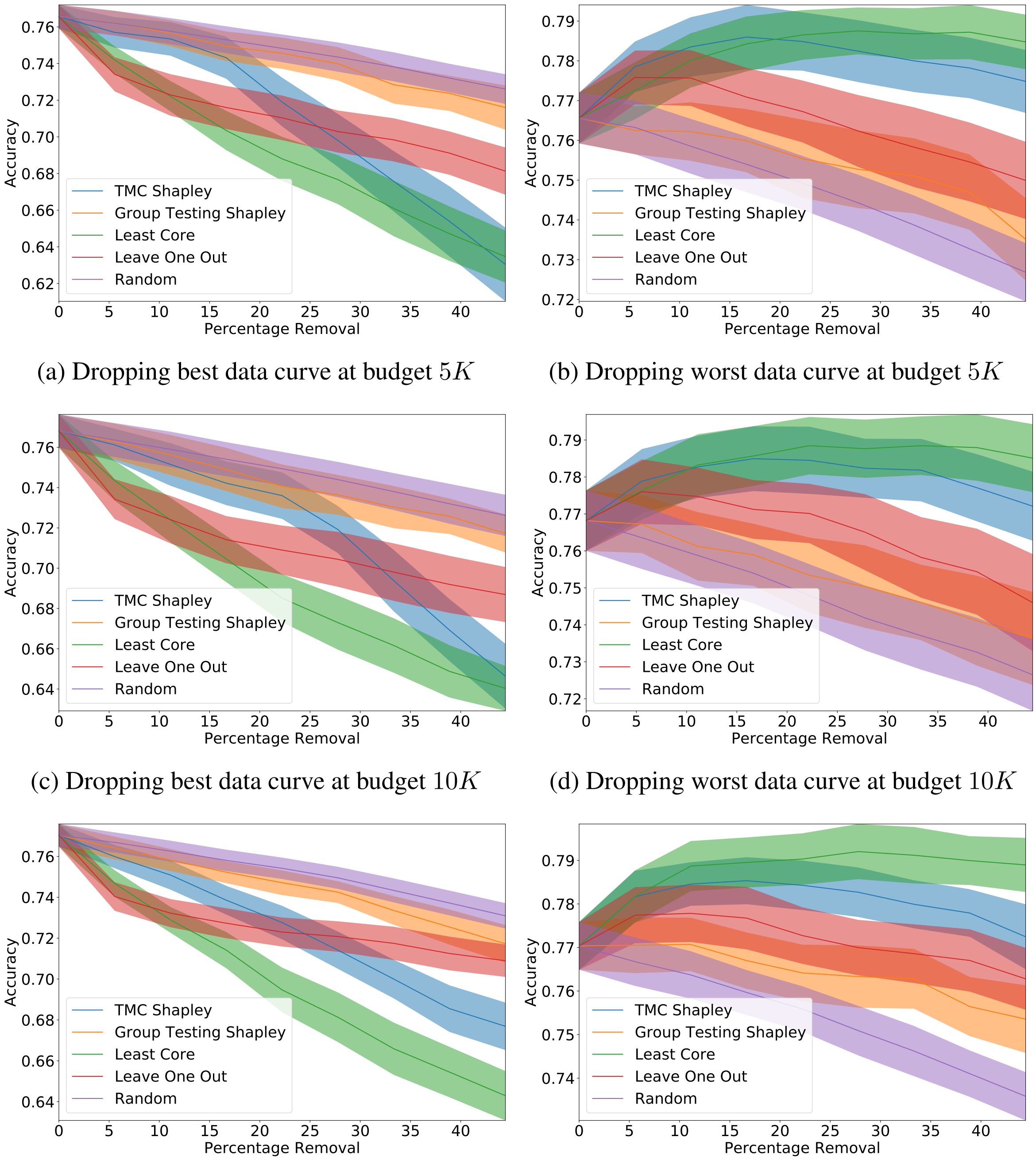
8.11.2.2 Data Quality vs. Score

Lastly, we repeat one more experiment that assesses data quality vs solution concept value. We randomly sample 200 dog-vs-fish data points to form an equally balanced training set. We corrupt 20 percent of train data by adding varying levels of white noise to the features and compute the Least Core value of clean and noisy images. The 5 noise levels are such that it leads to a monotonic decrease in test performance. Then, we plot the percentage of total utility that is assigned to clean scores (since the total utility goes down with noise, using the absolute scale makes it harder to interpret the result). This procedure is repeated 20 times and a budget of 1000 is alloted for approximating the least core.

As can be seen in Figure 8.3, under the no-noise setting, the clean data account for roughly 80 percent of the total utility and with increasing noise added the proportion grows bigger. The slight trend is due to the fact that the test performance does not drop by too much, going from 96.3 to 92.7.

8.11.2.3 Is the Approximate Shapley Value in the Approximate Least Core

Our test procedure is as follows: for each randomly sampled coalition value $v(S)$ used in approximating the least core and estimated Shapley value $\mathbf{x}_S$, we compute $(\sum_{i \in S} x_i + \hat{e})/v(S)$. We count the number of samples for which the ratio is below 0.95. Indeed, if we find one, then the approximate Shapley value $\mathbf{x}$ is not close to being in the approximate least core. Overall, we find that in all the settings we checked, the approximated Shapley does not lie in the approximated least core. For most experiments, at least one percent of all sampled coalitions has its ratio below 0.95. Other trends include that Group Testing tends to produce many more violations than TMC and that the percentage of violations decreases with a larger budget.



(a) Dropping best data curve at budget $5K$

(b) Dropping worst data curve at budget $5K$

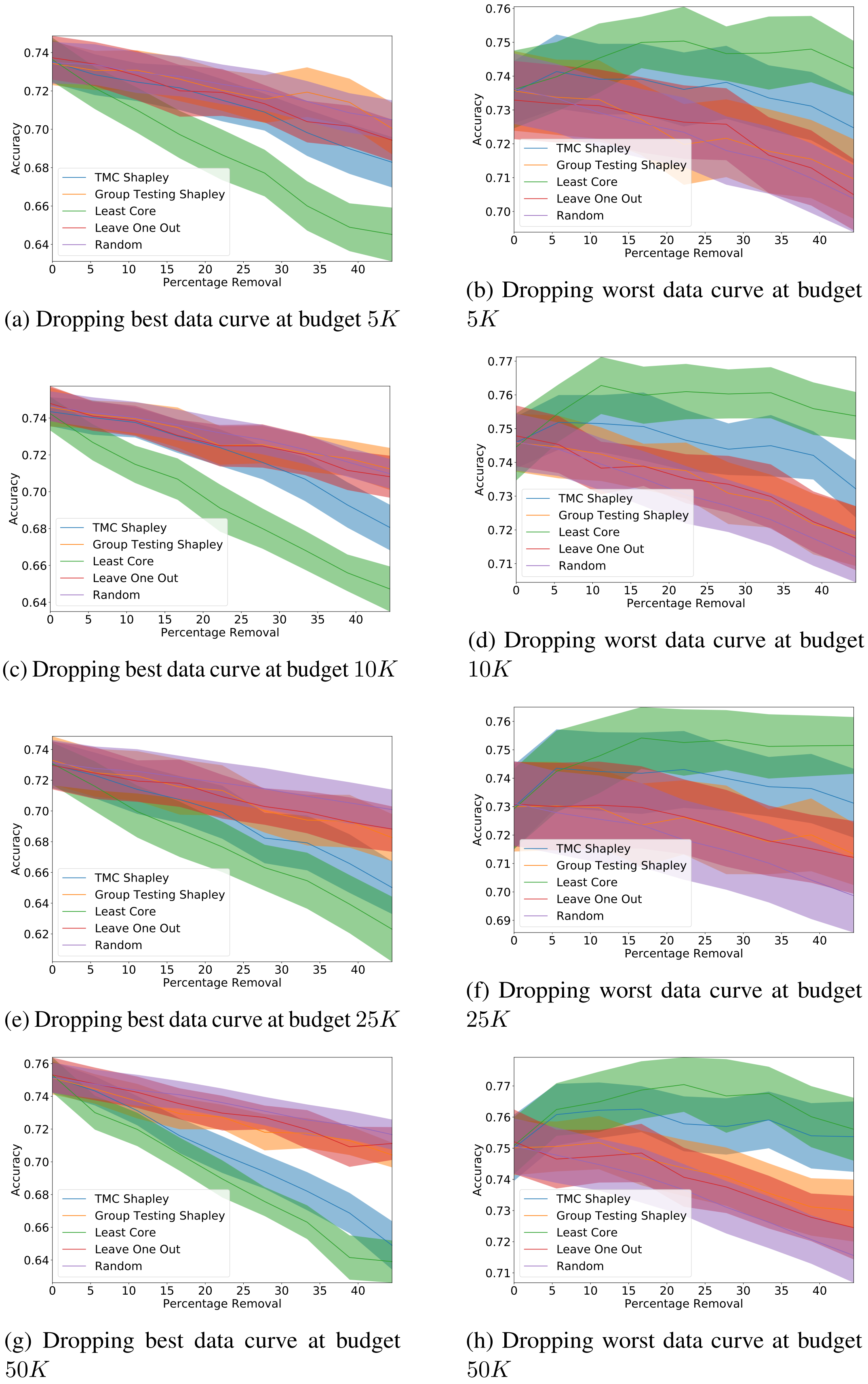
(c) Dropping best data curve at budget $10K$

(d) Dropping worst data curve at budget $10K$

(e) Dropping best data curve at budget $25K$

(f) Dropping worst data curve at budget $25K$

Figure 8.7: Curves of synthetic dataset (under a logistic regression model) test performance when the best and worst data points ranked according to the solution concepts are removed. For the left column, the steeper the drop, the better. For the right column, the sharper the rise, the better.



(a) Dropping best data curve at budget $5K$

(b) Dropping worst data curve at budget $5K$

(c) Dropping best data curve at budget $10K$

(d) Dropping worst data curve at budget $10K$

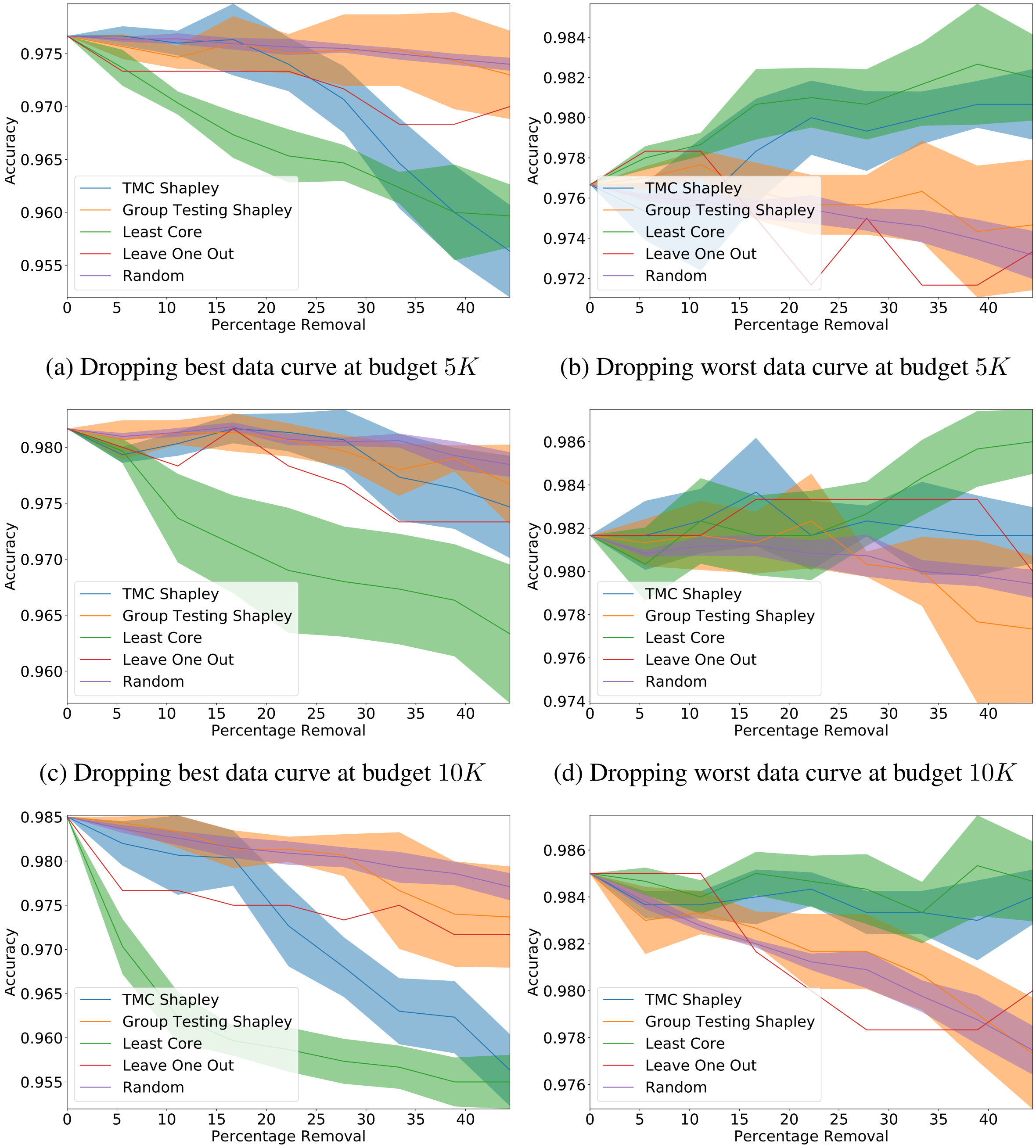
(e) Dropping best data curve at budget $25K$

(f) Dropping worst data curve at budget $25K$

(g) Dropping best data curve at budget $50K$

(h) Dropping worst data curve at budget $50K$

Figure 8.8: Curves of synthetic dataset (under a feedforward neural network model) test performance when the best and worst data points ranked according to the solution concepts are removed. For the left column, the steeper the drop, the better. For the right column, the sharper the rise, the better.



(a) Dropping best data curve at budget $5K$

(b) Dropping worst data curve at budget $5K$

(c) Dropping best data curve at budget $10K$

(d) Dropping worst data curve at budget $10K$

(e) Dropping best data curve at budget $25K$

(f) Dropping worst data curve at budget $25K$

Figure 8.9: Curves of natural, dog-vs-fish dataset (under a logistic regression model) test performance when the best and worst data points ranked according to the solution concepts are removed. For the left column, the steeper the drop, the better. For the right column, the sharper the rise, the better.

Chapter 9

Decentralized Coordination via Outcome-based Payment

9.1 Introduction

Increasingly, we are seeing businesses deploying agents to carry out tasks on their behalf. In the coming agentic era, we will inevitably have multiple, decentralized agents interacting together. An emerging challenge that businesses may have to face is how to incentivize other agents to work alongside its agent. This challenge requires addressing a central difficulty in decentralized multi-agent systems, which is that of differing interests.

In present day commerce, payment is a standard way that two parties use to resolve this challenge and more closely align their business interests. This inspires us to study the overarching question in this chapter: how can we analogously implement such payment schemes in the multi-agent setting and enable *economic alignment*? That is, if I am a business looking to use payment to incentivize another business (and/or its agent) to work with my agent, how can I learn a good policy for my agent along with a payment scheme to go with it?

On a technical level, this setting may be viewed as a Stackelberg Markov game. In this chapter, we study the two-player Stackelberg game, where one player (leader) commits to a policy taking into account the best response to the policy by the other player (follower). We focus on Stackelberg Markov games in particular as agents will be interacting over multiple turns and potentially long horizons. Finally, to model the payment aspect, the leader is able to also increase the reward of the follower in the Markov game, which may be viewed as a form of reward shaping in line with the existing formulation in the literature [41, 48, 152, 248, 297].

In this work, we aim to consolidate the theoretical foundations of Stackelberg learning with payment, as complexity results have yet to be established for two-player Stackelberg Markov games. We focus on a fundamental question: is there an efficient algorithm that can provably compute or learn the optimal policy and payment? Indeed, this is an important question to address as businesses in the future would want payment schemes with *provable guarantees*, so as to ensure that their expenditure is optimal.

Contributions: We analyze the planning and learning setting through both the computational and statistical lens. Please see Table 9.1 for an overview of our results.

	Without Payment	With Payment	
Planning, Learning	DAG	Tree	DAG
Cooperative	✓, × (Theorem 4950)	✓, ✓	✓, ✓ (Theorem 5253)
General Sum	×, × (Proposition 41)	✓, ✓ (Proposition 42)	×, × (Theorem 48)

Table 9.1: Planning & learning settings where computationally *and* statistically efficient algorithms exist.

1. We begin by considering planning in general-sum games. Is there an efficient algorithm that can return the optimal policy and payment? We prove that such a computationally efficient algorithm cannot exist unless NP=P, and identify the structural property of the MDP that results in this hardness. To complement the negative results, we develop an efficient algorithm, applicable when this property is removed.
2. Next, we turn to Cooperative games, which is a broad subclass of Markov games useful for modeling e.g. the interaction between AI service-providers and their users. Moreover, planning is computationally efficient in this setting, making it plausible that efficient learning algorithms may be attainable. As the rewards are already aligned, we begin by considering learning in the Stackelberg game without payment. Surprisingly, however, we find that an efficient algorithm cannot exist, this time in the statistical sense. We identify structural properties of the MDP that result in statistical hardness, and develop an efficient algorithm for when such properties are removed to complement our negative results.
3. Finally, we study learning in Cooperative games with payment. Can payment be used to alleviate the statistical hardness of learning? We answer this in the affirmative by showing that we can adapt existing no-regret RL algorithms to enable sample-efficient learning. In closing, we also use this setting to contrast the two different payment settings we study. We derive matching upper and lower regret bounds for when the leader has to make payments upfront versus on-the-fly, allowing us to quantitatively assess the benefits of being able to make payments on-the-fly.

9.2 Formulation

9.2.1 Stackelberg Markov Game

We consider the standard two-player, episodic finite-horizon Markov game M parameterized by $\langle S, A, B, H, P, \rho, r^L, r^F \rangle$ with state space S, initial state distribution $s_0 \sim \rho$, transitions P and episode length H. The leader has action set A and reward $r^L \in [-1, 1]$, the follower action set B and reward $r^F \in [-1, 1]$. In the case that the game is cooperative, $r^L = r^F$.

In the problem of online learning for Stackelberg Markov games, the learner plays the role of the leader, where apriori the reward functions r^L, r^F and the transitions are unknown to the leader. At each episode $k \in [T]$, the leader commits first to a policy π_k. The follower best responds to π_k with $\mu(\pi_k) \in \operatorname{argmax}_\mu V^{\pi_k,\mu}(s_0; r^F)$. One may view best response as the equilibrium behavior of the follower to the leader policy.

After the episode, the leader and the follower observe the resultant trajectory τ_k realized by the chosen policies in M, where $\tau_k = \left\{(s_i, a_i, b_i, r^L(s_i, a_i, b_i), r^F(s_i, a_i, b_i)\right\}_{i=1}^{H}$ and $a_i \sim \pi_k(s_i), b_i \sim \mu(\pi_k)(s_i)$, $s_{i+1} \sim P_i(\cdot|s_i, a_i, b_i)$.

This trajectory is the outcome of the policies' interaction, which in turn determines the *outcome-based* payment the follower receives.

Leader Payment: Following existing formulations in prior literature, the leader can increase r^F by creating outcome-based payment $b_i^k(s_i, a_i, b_i)$, if state-actions s_i, a_i, b_i are realized during the episode, $s_i, a_i, b_i \in \tau_k$. This results in a modified Markov game where the leader is able to additionally assign payment, with the payment function having signature $b_i^k : S \times A \times B \to \mathbb{R}^+$.

We note that the outcome-based payment need not correspond to direct monetary transfer. For example, we may be interested in modeling the setting where the leader is an AI-service-provider and the follower is a customer user. The leader spends money to improve its agent, and this improved agent adds additional value (e.g. more saved time) for the user during its use. But during this interaction, there is no direct transfer of money from the company to the user.

Thus, to model indirect payments in addition to direct ones, we introduce a final piece of notation, multiplier $\kappa \in \mathbb{R}^+$. $\kappa \cdot b_i^k(s_i, a_i, b_i)$ corresponds to the proportional cost to the leader in creating payment (reward) $b_i^k(s_i, a_i, b_i)$ for the follower. We believe proportionality is a natural assumption to make, and verily $\kappa = 1$ corresponds to direct payment.

9.2.2 Payment Settings

To complete the formulation, we touch on the two types of payment settings considered in this chapter.

Trajectory Payment: The first is the existing payment setting commonly studied in prior literature, which we term trajectory payment. Here, a payment is made by the leader for every state-action on the realized trajectory. This form of payment is considered in principal-agent contracting literature, where the trajectory informing how much the leader will be paying ex-post [96].

Moreover, this form of payment corresponds to the trendy outcome-based pricing model, which is experiencing rapid adoption by several notable SaaS companies due to the rising usage of AI agents [150, 268, 316]. Indeed, this marks a fundamental paradigm shift in software pricing in industry, moving from seat-based subscriptions (traditional SaaS) and usage-based models (cloud infrastructure) to now outcome-based pricing in the agent era [49, 253]. This also makes it imperative then to bolster our theoretical understanding of outcome based pricing, which we study in this chapter.

Upfront Payment: In this chapter, we will also consider a setting that we term upfront payment. As the name suggests, the leader pays for every state-action in the MDP, regardless of the realized trajectory. Note that the follower is still paid based on the realized trajectory. This is more realistic in settings where the leader pays indirectly to the benefit of the follower, and is bound by temporal constraints such that the payment cannot be made on-the-fly.

For a motivating example, consider the AI-service provider setting discussed earlier. The company invests before deployment to improve the agent's functionality, which means that the user (follower) gains added value (reward) on the trajectory realized during the agent's use. However, the key temporal constraint is that the company cannot improve its agent on-the-fly, as the users are using it. Thus, this makes upfront payment a more realistic model of the leader's expenditure. The leader had to invest upfront to improve the agent's capabilities in all states, even though this includes off-trajectory states that are not visited during the interaction with the user. For instance, suppose the agent is a computer-using-agent [14]. The user may use it to handle emails, and the agent would act in states of the computer corresponding to the inbox. However, even though the company had also invested to improve the agent's capabilities in coding, the user may not invoke the agent to do so (perhaps due to excessive risk). And so, the agent would not have acted in other states of the computer corresponding to the codebase.

More generally, there is sizable body of economics contracting literature studying settings where only ex-ante (upfront) payment is possible. Some reasons for this include non-enforceable contracts, where the principal can renege upon observing the outcome [137]. Another cause for this may be non-verifiable outcomes; that is, when outcomes cannot be verified, ex-post contracts become unenforceable as there is no way to condition legally binding payments [5]. Finally, one other reason may simply be that the agent is risk-averse, thus preferring upfront payment in face of stochastic outcomes [174].

Leader Optimization: Putting it all together, we can now write down the resulting Stackelberg game under the two payment settings.

Definition 42. *In Stackelberg Markov games with trajectory payment, the leader optimizes:*

$$\begin{aligned} \max_{\pi, b \geq 0} \quad & V^{\pi,\mu(\pi)}(s_0; r^L - \kappa \cdot b) \\ \textit{s.t.} \quad & \mu(\pi) \in \operatorname*{argmax}_{\mu'} V^{\pi,\mu'}(s_0; r^F + b) \end{aligned}$$

In Stackelberg Markov games with upfront payment, the leader optimizes:

$$\begin{aligned} \max_{\pi, b \geq 0} \quad & \left(V^{\pi,\mu(\pi)}(s_0; r^L) - \kappa \cdot \sum_{s,a,b \in S \times A \times B} b(s,a,b) \right) \\ \textit{s.t.} \quad & \mu(\pi) \in \operatorname*{argmax}_{\mu'} V^{\pi,\mu'}(s_0; r^F + b) \end{aligned}$$

Before moving on, we highlight the generality of the class of games we are studying. The class of Stackelberg Markov games with payment generalizes Stackelberg Markov games. Indeed, constraining the leader to zero payment (i.e. $(\pi, b) = (\pi, 0)$) corresponds to the leader's policy space in Stackelberg Markov Games. Analogously, for Cooperative Stackelberg Markov games with payment studied in the later sections, this class of games generalizes Cooperative Stackelberg Markov games.

9.3 Related Works

As we focus on Stackelberg Markov games with payment, our work is most related to two lines of work. The first is the line of work studying the complexity of Stackelberg policy computation in Markov games. And the second is algorithms for computing optimal payment schemes in

MDPs. We cover both lines of work below, and include a discussion on additional related works in Appendix 9.12.

Stackelberg Optimal Policies in Markov Games without Payment: Due to the wide applicability of the Stackelberg Markov games, there has been a long line of work seeking to understand how to compute optimal leader policies with provable guarantees.

For planning, Conitzer and Sandholm [78], Letchford and Conitzer [182], Letchford et al. [183] study the computational tractability of optimal Stackelberg policy computation in Markov games and subclasses thereof. For stochastic MDPs, they establish that computing the optimal Stackelberg policy is NP-Hard.

For learning, Zhao et al. [322] studies the statistical complexity in cooperative bandit games. Bai et al. [26] studies the statistical complexity in bandit-RL games, a particular subclass of Markov games. Our work differs from this line of work in focusing on Markov games, which are more general than bandit-RL games and have longer horizon than bandit settings. Moreover, the leader is allowed to use payments to shape the follower's rewards. As we will see, this turns out to be crucial for improved exploration during learning in certain settings.

Learning the Optimal Payment Scheme in MDPs: Recently, there has been burgeoning interest in computing optimal payment schemes for contracting agents to act in MDP environments, wherein the leader may increase the follower's rewards as a form of reward shaping to incentivize the follower to play policies desirable to the leader.

The single-agent MDP setting, where only the follower acts in the MDP and the leader incentivizes, is formulated by Ben-Porat et al. [41], Chen et al. [63]. This is followed by a series of interesting work by Bollini et al. [48], Ivanov et al. [152], Wu et al. [297], studying learning under a variety of different payment functions taking as input the state, the state-action or the state-next-state. Our work adds to this line of work by focusing on two-player Markov games, which generalize the single-player setting. Furthermore, while previous works mostly focus on trajectory payment, we also consider upfront payment, applicable in settings where the leader cannot pay on the fly due to temporal constraints. We derive tight regret guarantees to contrast the two differing payment settings.

The paper closest in formulation to that of ours is that by Scheid et al. [248], who considers the same state-action based payment function in the bandit setting. Our work differs in focusing on Markov games, with a longer horizon than that in bandit settings. This in turn introduces difficulty in terms of exploration, and requires a more nuanced optimal payment computation beyond the binary search approach used in [248].

Finally, as payment may be viewed as strategic reward shaping, our analysis is also related to existing RL literature that seeks to theoretically quantify the benefits of reward shaping [213]. Gupta et al. [126] quantifies how statistical sample complexity is improved by reward shaping in the single-agent setting. By contrast, in our work, we study improved sample complexity in two-player cooperative Stackelberg Markov games.

9.4 Planning in General-sum Games

In this section, we ask: is there an efficient algorithm that can compute the optimal policy and payment in general-sum games? We investigate the computational complexity of such an algorithm, starting with the planning setting. Our main finding is that there is no such computationally efficient algorithm unless NP=P. Outcome-based payment does not alleviate the computational intractability of computing the optimal Stackelberg policy, even in planning [78]. We identify that when the MDP has DAG structure, this leads to computational intractability. Later in the section, we complement this negative result with a positive result for when the MDP has tree structure. All proofs in this section may be found in Appendix 9.8.

9.4.1 Hardness Results

We first derive a result showing that it is NP-Hard to compute the optimal leader policy even in deterministic MDPs, without payment. Note that in [78], computational intractability is demonstrated in stochastic MDPs.

Proposition 41. *Under Markov games that are deterministic DAGs, it is NP-Hard to compute the optimal policy:*

$$\begin{aligned} \max_{\pi} \quad & V^{\pi,\mu(\pi)}(s_0; r^L) \\ \textit{s.t.} \quad & \mu(\pi) \in \operatorname*{argmax}_{\mu'} V^{\pi,\mu'}(s_0; r^F) \end{aligned}$$

Helpfully, deterministic MDPs allow us to provide guarantees for both two payment settings. As we show in the proof, the optimal payment scheme pays zero in off-policy states, which can be readily characterized in deterministic MDPs. This result is intuitive as paying in off-policy states only incentivizes the follower to deviate off-policy, which is undesirable and increases leader total payment. With this result, we can derive that the optimal payment scheme is the same under trajectory and upfront payment. Thus, we use same construction, which provides a reduction to the PARTITION problem, to prove computational intractability under both payment settings.

Theorem 48. *Under Markov games that are deterministic DAGs, it is NP-Hard to compute the optimal policy and optimal trajectory payment:*

$$\begin{aligned} \max_{\pi, b \geq 0} \quad & V^{\pi,\mu(\pi)}(s_0; r^L - \kappa \cdot b) \\ \textit{s.t.} \quad & \mu(\pi) \in \operatorname*{argmax}_{\mu'} V^{\pi,\mu'}(s_0; r^F + b) \end{aligned}$$

and it is also NP-Hard to compute the optimal policy and optimal upfront payment:

$$\begin{aligned} \max_{\pi, b \geq 0} \quad & V^{\pi,\mu(\pi)}(s_0; r^L) - \kappa \cdot \sum_{s,a,b \in S \times A \times B} b(s,a,b) \\ \textit{s.t.} \quad & \mu(\pi) \in \operatorname*{argmax}_{\mu'} V^{\pi,\mu'}(s_0; r^F + b) \end{aligned}$$

In closing, we note that the optimal objective value of the subset of Markov games used to reduce to the PARTITION problem is an integral multiple of $1/2$. Due to this, we have that computational intractability in planning implies computational intractability in learning. In more detail, let M^* be the optimal objective value, which is an integer multiple of $1/2$. Suppose by contradiction that we had an algorithm with sublinear regret T^α ($\alpha < 1$). We can then set T large enough such that $T^\alpha/T < 1/2$. This allows us to infer M^* exactly by rounding to the nearest $1/2$, giving us a computationally efficient algorithm for answering the decision version of the PARTITION problem, which is a contradiction.

Algorithm 23 Planning Algorithm for MDP with Deterministic Tree Structure

```
Require: Pre-computed policy $\pi^- \in \text{argmin}_\pi V^{\pi,\mu(\pi)}(s_0; r^F)$ (efficiently computed via Nash-VI)
  for all root to leaf paths $\tau = s_1, a_1, b_1, s_2, a_2, b_2, ..., s_H, a_H, b_H$ do
    Define $\pi(s_i) = a_i$ for $s_i, a_i \in \tau$. In every other state $s_i' \notin \tau$, let $\pi(s_i') = \pi^-(s_i')$.
    Compute $\mu(\pi)$ and compute follower Q-values, $Q^{\pi,\mu(\pi)}(\cdot, \cdot, \cdot)$.
    Solve for the minimal payment scheme using LP:
```

$$\begin{aligned} b^\tau(\pi) = \underset{b}{\text{argmin}} \quad & \sum_{s_i, a_i, b_i \in \tau} b(s_i, a_i, b_i) \\ \text{s.t.} \quad & \sum_{i \geq h, s_i, a_i, b_i \in \tau} r^F(s_i, a_i, b_i) + b(s_i, a_i, b_i) \geq \max_{b_h' \neq b_h} Q^{\pi,\mu(\pi)}(s_h, a_h, b_h'; r^F) \end{aligned}$$

```
  Output the leader policy $\pi$ and payment scheme of the path $\tau$ with maximal return $\sum_{s_i, a_i, b_i \in \tau} r^L(s_i, a_i, b_i) - \kappa \cdot b^\tau(\pi)$.
```

9.4.2 Positive Results

To complement our negative results, we show that positive results are attainable in MDPs without DAG structure. That is, in general-sum games where the MDP has tree structure, there is a polynomial-time algorithm for learning the optimal leader policy and payment. We describe our planning algorithm, Algorithm 23, which forms the crux of our approach to learning in this setting and is applicable under both trajectory and upfront payment.

Proposition 42. *Under Markov games that are deterministic trees, there exists a polynomial-time planning algorithm that computes the optimal policy and payment.*

Remark 23. *To complete the result, we note in Appendix 9.8 that there is a simple exploration strategy using payment for general-sum, deterministic trees, as exploration needs to only recover rewards. This strategy allows us to reduce learning to planning, and then apply Algorithm 23.*

Before moving on, we note that in this general-sum game, the leader behaves in a zero-sum like manner in off-policy states in Algorithm 23. This incentivizes the follower to take the desired policy and allows the leader to minimize the total payment needed to incentivize such policy.

Finally, due to the intractability of computing a global Stackelberg optimum, it is natural to consider computing a local Stackelberg optimum instead, so that the policy and payment scheme does attain some guarantees. Building on existing results on first order methods in Stackelberg games [259], we derive a first order approach to this end. Note that while our work is concerned with global Stackelberg optimality guarantees, we use this to illustrate that a more relaxed solution concept can be computed, if desired.

9.5 Learning in Cooperative Games without Payment

The computational intractability in the general-sum case prompts us to investigate whether efficient algorithms are attainable in significant subclasses of Markov games. Cooperative games are a broad subclass of Markov games useful for modeling e.g. the aforementioned AI-service based setting. Indeed, since the goal of the assistant agent is to aid the user, their rewards are aligned. And so, such settings correspond to a two-player cooperative game, making it an important subclass of Markov games to understand.

Moreover, on a technical level, it seems that there is hope for efficient algorithms as planning is efficient in cooperative games (e.g. via Nash-VI as in [25]). And so, in this section, we study the question: is there an efficient learning algorithm in cooperative games? We delve into this by first considering cooperative games without payment, which has yet to be addressed in the prior literature. Since the rewards are already aligned, we might expect that there are efficient learning algorithms. To our surprise, however, we find that learning in Cooperative Markov games can be prohibitively hard, this time in the statistical sense. All proofs in this section may be found in Appendix 9.9.

Structural properties of MDP: We identify the specific MDP properties under which exploration can be statistically intractable, along with complementary positive results. In a nutshell, we find that if the MDP has deterministic tree structure, then efficient algorithms are possible. However, allowing for stochastic or DAG transitions leads to statistical hardness.

Theorem 49. *There exists a turn-based Stochastic Tree Markov game such that: any (possibly randomized) algorithm that returns the optimal leader policy with probability at least* $1/2$ *requires at least* $\Omega(2^{|S|})$ *number of episodes.*

Theorem 50. *There exists a turn-based Deterministic DAG Markov game such that: any (possibly randomized) algorithm that returns the optimal leader policy with probability at least* $1/2$ *requires at least* $\Omega(2^{|H|})$ *number of episodes.*

Proposition 43. *Under Markov games that are deterministic trees, then there exists a polynomial-time algorithm that can learn a near-optimal leader policy.*

We remark that the statistical intractability results are based on a “needle-in-the-haystack” construction, where only a specific combination of leader actions is optimal. Structural properties of the MDP like stochastic or DAG transitions allow us to embed this construction in the MDP. Combined with the follower best responding instead of coordinating exploration with the leader, we can show that an exponential number of samples is needed by the leader to find the right combination, even if the rewards are already aligned.

Relaxing Follower Best Response behavior: As the statistical hardness is due to both the structural property of the MDP and the best response nature of the follower, a natural question one may ask is: can relaxing the latter alleviate statistical hardness and allow for efficient learning across all MDPs?

The natural way to relax best response is to consider best response under λ-entropy-regularization, which generalizes follower best response (corresponding to when $\lambda = \infty$). This behavior model is often used to model human behavior in human-AI interaction and behavioral economics literature [199, 238, 328]. However, we again find that learning with this follower behavior does not allow for more sample efficient exploration:

Theorem 51. *There exists a turn-based Deterministic DAG Markov game such that: any (possibly randomized) algorithm that outputs the optimal policy given λ-Entropy-regularized best response with probability at least $1/2$ requires at least $\Omega(\exp(\lambda^2 H/8))$ episodes if $\lambda \leq 1$ and $\Omega(\exp(H/8))$ episodes if $\lambda > 1$.*

In closing, we offer a conceptual interpretation of the technical results in this section, using the example of the assistant agent and the user. Our results suggest that the service provider company can have difficulty exploring, due to the user's best response. Indeed, users are simply looking to use the agent wherever it is at its best, and will not use the agent for the sake of its improvement. In particular, this means that users are not willing to use the agent in states that it currently does not currently excel in. Even though, these are precisely the states that the agent needs to obtain more training samples in. And so, this suggests that if the company wants to efficiently explore to learn an even better agent, incentivized exploration is needed.

9.6 Learning in Cooperative Games with Payment

In sum, we know from the previous section that in Stackelberg games, coordinated exploration is necessary for efficient learning. And so, in this section, we study how payment can be used to align the follower and enable efficient leader exploration. Our overall finding is that payment can lead to efficient exploration, and alleviate the statistical hardness in cooperative games without payment. All proofs in this section may be found in Appendix 9.10.

9.6.1 Regret Guarantees in Cooperative Games

We study regret guarantees under the standard reinforcement learning setup with unknown transitions and unknown rewards, which can be stochastic.

Learning protocol: At each episode $k \in [T]$, the leader commits first to a policy π^k and a payment function b^k. The follower best responds to π^k with $\mu(\pi^k) \in \operatorname{argmax}_\mu V^{\pi_k,\mu}(s_0; r^F + b^k)$. After the episode, the leader and the follower observe the resultant trajectory $\tau_k = \{(s_i, a_i, b_i, r^L(s_i, a_i, b_i)\}_{i=1}^H$ realized by the chosen policies in M (recall that $r^L = r^F$). The goal of the learner is to minimize its Stackelberg regret, defined as follows:

Definition 43. *In Stackelberg games with trajectory payment, the Stackelberg regret is defined as:*

$$\mathcal{R}(T) = \sum_{k=1}^{T} V^{\pi^*,\mu(\pi^*;r^F+b^*)}(s_0; r^L - \kappa \cdot b^*) - V^{\pi^k,\mu(\pi^k;r^F+b^k)}(s_0; r^L - \kappa \cdot b^k)$$

The regret under upfront payment regret may be defined analogously.

Towards analyzing Stackelberg regret, we characterize the optimal policy and trajectory payment when $r^L = r^F$; we can analogously show the same result under upfront payment.

Lemma 68. *For any π^*, b^* such that:*

$$\begin{aligned} \pi^*, b^* = \operatorname*{argmax}_{\pi, b} \quad & V^{\pi, \mu(\pi; r^F + b)}(s_0; r^L - \kappa \cdot b) \\ \textit{s.t.} \quad & \mu(\pi; r^F + b) \in \operatorname*{argmax}_{\mu'} V^{\pi, \mu'}(s_0; r^F + b) \end{aligned}$$

If $r^L = r^F$, then we must have $\pi^*, \cdot = \operatorname{argmax}_{\pi,\mu} V^{\pi,\mu}(s_0; r^L)$ and $b^* = 0$.

With this, we have that the optimal payment scheme in any cooperative game must be zero, as one would intuitively expect with already aligned rewards. This allows us to decompose Stackelberg regret into regret due to sub-optimality in policy and regret due to payment used during exploration, which will be responsible for the differing rates between trajectory and upfront payment.

Moreover, we note an interesting contrast due to this result. As we just saw, learning can be prohibitively hard in the absence of payment. Hence, we have that payment is not necessary in planning, but is crucial for learning (efficiently).

The crux of our positive results is that we can apply the canonical optimism under uncertainty principle to achieving sublinear Stackelberg regret. This follows from the observation that payment enables optimism in learning, which the leader can operationalize by setting payments according to its bonuses. This incentivizes the follower to also explore optimistically. A key lemma for bounding the policy regret portion of Stackelberg regret goes as follows.

Lemma 69. *Suppose we can construct an optimistic MDP M_k of the true MDP M. Let the optimal leader policy under M_k be π_k, then:*

$$\sum_{k=1}^{T} V_M^{\pi^*, \mu_M(\pi^*)}(s_0; r^L) - V_M^{\pi^k, \mu_M(\pi^k)}(s_0; r^L) \leq \sum_{k=1}^{T} V_{M_k}^{\pi^k, \mu_{M_k}(\pi^k)}(s_0; r^L) - V_M^{\pi^k, \mu_{M_k}(\pi^k)}(s_0; r^L)$$

Note that because the leader knows M_k, they know the policy $\mu_{M_k}(\pi^k)$ that they would like to incentivize the follower to play. Using this, we show that one can also bound the regret due to the cumulative payment, to obtain the following regret guarantees.

Theorem 52. *UCB-VI-FP (Algorithm 24) incurs $O(T^{1/2})$ regret under trajectory payment. This is tight as there exists a subset of Markov games, where any learning algorithm must incur $\Omega(T^{1/2})$ regret.*

Theorem 53. *There exists an algorithm, leveraging UCB-VI-FP as subroutine, that incurs $O(T^{2/3})$ regret under upfront payment.*

9.6.2 Contrasting Trajectory Payment with Upfront Payment

Finally, as positive results are attainable in Cooperative Markov games, we can analyze the difference in regret rates under the two different payment settings. What is the benefit afforded by settings where the leader can pay on-the-fly? Towards answering this question, we analyze the simple setup of unknown, deterministic rewards. Helpfully, this learning task already a sizable contrast in terms of regret between the two settings. We provide tight bounds on regret guarantees under both payment settings to contrast the two payment settings.

Algorithm 24 UCB-VI with Follower Payment (UCB-VI-FP)

```
Initialize $Q_h(s,a,b) = H$ for all $h \in [H], s,a,b \in S_h \times A \times B$.
for $k = 1, ..., T$ do
    for $h = H, ..., 1; s,a,b \in S_h \times A \times B$ do                    ▷ construct $M_k$
        Compute estimated transitions from data in buffer: $\hat{P}_h(s'|s,a,b) = \frac{N_h^k(s,a,b,s')}{N_h^k(s,a,b)}$
        Compute optimistic rewards of $M_k$ from reward samples in buffer: $\hat{r}_h^k(s,a,b) = \bar{r}_h^k(s,a,b) + c\sqrt{\frac{H^2}{N_h^k(s,a,b)}}$    ▷ standard bonus for stochastic rewards
        $Q_h(s,a,b) = \min(H, \hat{r}_h^k(s,a,b) + \hat{P}_h^k V_{h+1}(s,a,b))$
        $V_h(s) = \max_{a,b} Q_h(s,a,b)$
    Leader commits to Stackelberg policy $\pi^k$: $\pi^k(s_h) = \text{argmax}_a \max_b Q_h(s_h,a,b)$.
    Set outcome-based payment scheme: $\beta_h^k(s_h,a_h,b_h) = 2 \cdot c\sqrt{\frac{H^2|S|}{N_h^k(s,a,b)}}$.
    for $h = 1, ..., H$ do
        Leader plays $a_h^k \sim \pi^k(s_h^k)$, follower plays $b_h^k$ via $\mu(\pi^k)$
        Transition to $s_{h+1}^k \sim P(\cdot|s_h^k, a_h^k, b_h^k)$ and save data $(s_h^k, a_h^k, b_h^k, s_{h+1}^k)$ in buffer
```

Proposition 44. *UCB-VI-FP with indicator bonus incurs constant $O(|S||A||B|)$ regret under trajectory payment, where we designate reward under indicator bonus to be $\hat{r}(s,a,b) = 1\{$if (s,a,b) is unvisited$\}$ and $r(s,a,b)$ o.w.*

As the regret bound is constant in T, we have that the bound must be tight. Next, we derive regret rates under upfront payment, whose regret lower bound requires a significantly nuanced probabilistic argument using Yao's lemma.

Proposition 45. *There exists an algorithm, leveraging UCB-VI-FP with indicator bonus as subroutine, that incurs $O(T^{1/2})$ regret under upfront payment.*

Proposition 46. *There exists a subset of Markov Game instances such that any learning algorithm has to incur $\Omega(T^{1/2})$ regret under upfront payment.*

The construction of the negative result reveals the key difference in two payment schemes. In a nutshell, upfront payment is affected by difficult-to-reach states (ϵ-significant states [159]). On the other hand, trajectory payment is unaffected as the payment is made only if the follower does reach such a state. That is, the leader's payment for actions in that statement is weighted by the visitation probability.

And so, the key difficulty in exploration under upfront payment is that when payment is needed to incentivize the follower to reach insignificant states, a lot of the payment can be wasted even if the follower is aligned, due to the low visitation probability. This is directly responsible for the sizable change in the regret guarantee, going from $O(1)$ to $\Omega(T^{1/2})$. Overall, this suggests that if the leader cannot pay on-the-fly, the payment scheme should factor in the reachability of states.

9.7 Discussion

In this work, we study learning in Stackelberg Markov games with payment. To consolidate the theoretical foundations of this setting, we chart the computational and statistical complexity of both planning and learning.

Future Work: Due to the intractability of general-sum settings, we believe that there is much more work to be done in analyzing more specific subclasses of Markov games. Which other subclasses of Markov games are such that efficient algorithms are attainable?

Limitations: In this chapter, we consider the full information setting. One key underlying assumption then is that the follower's action can be observed by the leader. We believe that this can be realistic for modeling certain digital settings (such as computers), wherein the agent's actions can be readily tracked (computer-using-agent's actions can be logged and monitored) [14, 270]. With that said, handling the case for when the follower's action is not observable is very important, especially in physical environment where monitoring is not possible. And we believe that results from the full information setting we study can serve as a stepping stone towards results in partial information settings with unobserved actions.

Another key underlying assumption is that the leader can readily observe the follower's reward, either directly or through the follower's report. It is conceivable that in cases the leader cannot observe the reward directly, the follower may not report their reward truthfully. In such settings, we note two observations. Let $(\pi^*(r), b^*(r))$ denote an optimal policy under reported follower reward r. Let r^F denote the true reward and r'^F the reported reward.

First, if we are in the cooperative setting, we observe that there is no incentive for the follower to misreport. Because the leader payment is zero, truthful reporting yields the highest return: $V^{\pi^*(r^F),\mu(\pi^*(r^F))}(s_0; r^F) \geq V^{\pi^*(r'^F),\mu(\pi^*(r'^F))}(s_0; r^F)$.

Second, in the general-sum bandit setting with direct payment considered by [248], the payment can now be nonzero but the follower's gain from misreporting is bounded.

Proposition 47. *Suppose the follower can misreport r^F up to Δ, $\|r'^F - r^F\|_1 \leq \Delta$. In the bandit setting, the follower's return can change by at most:*

$$|V^{\pi^*(r^F),\mu(\pi^*(r^F))}(s_0; r^F + b^*(r^F)) - V^{\pi^*(r'^F),\mu(\pi^*(r'^F))}(s_0; r^F + b^*(r'^F))| \leq 2\Delta$$

and the leader's return can change by at most:

$$|V^{\pi^*(r^F),\mu(\pi^*(r^F))}(s_0; r^L - b^*(r^F)) - V^{\pi^*(r'^F),\mu(\pi^*(r'^F))}(s_0; r^L - b^*(r'^F))| \leq 2\Delta$$

However, an open question is whether such a bound carries over to the Markov game case. How much could the follower gain from misreporting r^F up to Δ? Are there algorithms that can induce truthfulness, while still attaining some optimality guarantees? We believe there is a fruitful line of work to be done to handle cases where the leader cannot directly observe and/or verify the follower rewards.

9.8 Proofs for Planning Results in General-sum Games

In [78], it is demonstrated that it is NP-Hard to compute the optimal Stackelberg policy, in absence of payment. But with the ability to pay, we are interested in answering the question: is there a general, efficient algorithm that can compute the optimal policy and payment?

9.8.1 Hardness Results

Proposition 48. *Under Markov games that are deterministic DAGs, it is NP-Hard to compute the optimal policy:*

$$\begin{aligned} \max_{\pi} \quad & V^{\pi,\mu(\pi)}(s_0; r^L) \\ \textit{s.t.} \quad & \mu(\pi) \in \operatorname*{argmax}_{\mu'} V^{\pi,\mu'}(s_0; r^F) \end{aligned}$$

Proof. We show that one can reduce optimal policy computation to the DECISION Knapsack problem.

For a given knapsack instance $\langle \{v_i\}_i, \{w_i\}_i, W \rangle$. Construct the following MDP:

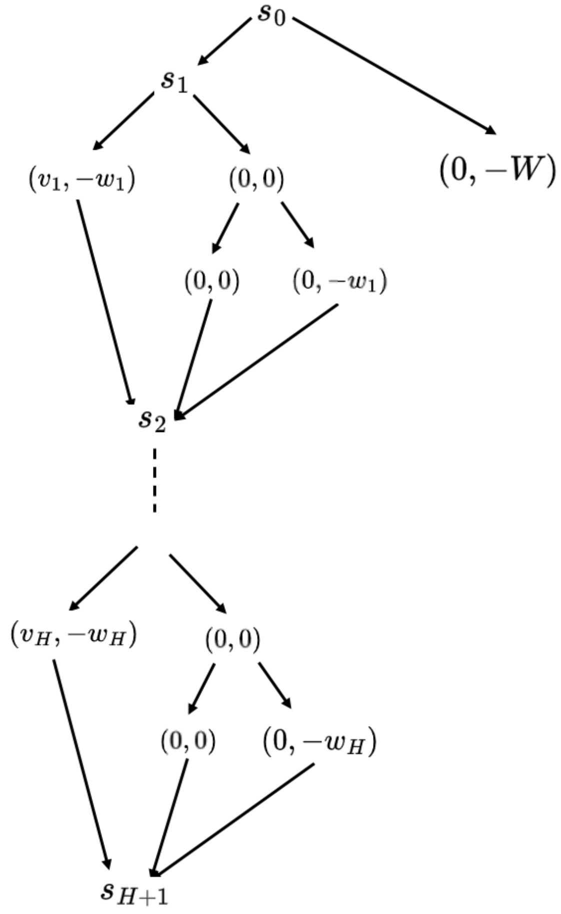


Figure 9.1

At the ith time step, the follower will select one of the two actions with rewards $(v_i, -w_i)$ or $(0, 0)$. The leader influences this through the probability $\pi(s_i) = \pi_i$ of playing $(0, -w_i)$. Here, we assume that the follower plays in favor of the leader in the event of a tie-break. That is, the follower plays the left action with reward $(v_i, -w_i)$ iff the leader plays $(0, -w_i)$ w.p. $\pi_i = 1$.

Therefore, we have that an optimal leader policy π^* must maximize the following objective:

$$\begin{aligned} \max_{\pi_1,\ldots,\pi_H} \quad & \sum_{i=1}^{H} v_i \mathbb{1}\left\{\pi_i = 1\right\} \\ \text{s.t.} \quad & \sum_{i=1}^{H} w_i \pi_i \leq W \\ & 0 \leq \pi_i \leq 1 \end{aligned}$$

since the leader wishes to incentivize the follower to play in the left branch (holds iff $\sum_{i=1}^{H} \pi_i(-w_i) \geq -W$), while maximizing the return in the left branch.

We will show that introducing the constraint $\pi_i \in \{0, 1\}$ is without loss of optimality. Consider some optimal policy π^*. For any i such that $\pi_i^* \in (0, 1)$, let $\pi_i' = 0$ and let $\pi_i' = \pi_i^*$ otherwise. It follows that π_i' is still feasible, while retaining the same objective value.

Hence, given some π^* optimal leader policy, its return matches the optimal objective value of the program:

$$\begin{aligned} \sum_{i=1}^{H} v_i \mathbb{1}\left\{\pi_i^* = 1\right\} = \max_{\pi_1,\ldots,\pi_H} \quad & \sum_{i=1}^{H} v_i \mathbb{1}\left\{\pi_i = 1\right\} \\ \text{s.t.} \quad & \sum_{i=1}^{H} w_i \pi_i \leq W \\ & \pi_i \in \{0, 1\} \end{aligned}$$

Hence, the return of the optimal leader policy can be used to answer the Knapsack Decision problem, making optimal Stackelberg leader policy computation at least as hard as the Knapsack Decision problem.

□

We show that even with payment, it is still NP-Hard to compute the optimal policy in both trajectory and upfront payment settings. Note that in the proof that follows, we can scale the rewards such that $r(\cdot) \in [-1, 1]$ is satisfied, as the optimal policies remain unchanged under scaling.

Theorem 54. *Under Markov games that are deterministic DAGs, it is NP-Hard to compute the optimal policy and optimal trajectory payment:*

$$\begin{aligned} \max_{\pi, b \geq 0} \quad & V^{\pi,\mu(\pi)}(s_0; r^L - \kappa \cdot b) \\ \textit{s.t.} \quad & \mu(\pi) \in \operatorname*{argmax}_{\mu'} V^{\pi,\mu'}(s_0; r^F + b) \end{aligned}$$

and it is also NP-Hard to compute the optimal policy and optimal upfront payment:

$$\begin{aligned} \max_{\pi, b\geq 0} \quad & V^{\pi,\mu(\pi)}(s_0; r^L) - \kappa \cdot \sum_{s,a,b\in S\times A\times B} b(s,a,b) \\ \textit{s.t.} \quad & \mu(\pi) \in \operatorname*{argmax}_{\mu'} V^{\pi,\mu'}(s_0; r^F + b) \end{aligned}$$

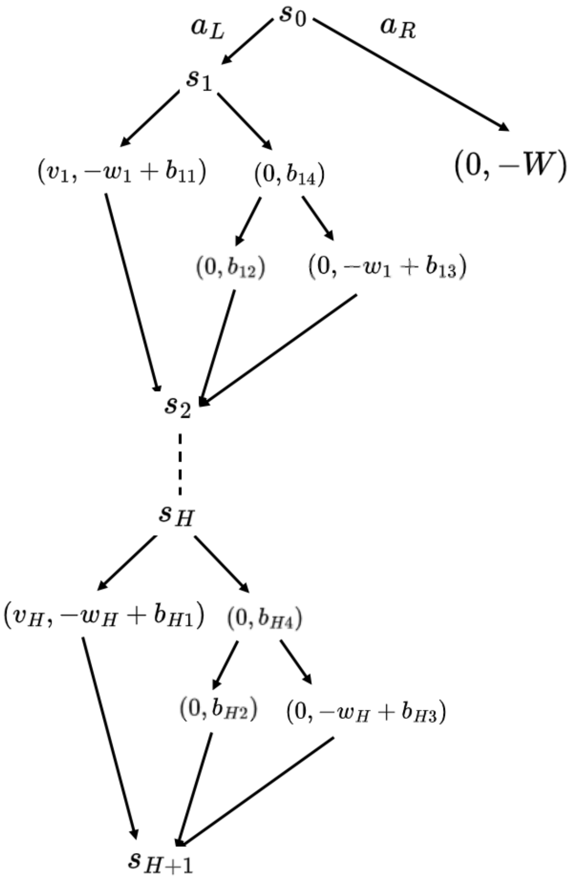


Figure 9.2: Same deterministic MDP with payment

Proof. We will consider the same construction visualized in Figure 9.2 and and for ease of presentation, we will show that the problem is NP-Hard under $\kappa = 1$ (e.g. the direct-payment settings).

Notation-wise, at the ith time step, let b_{i1}, b_{i4} be the leader payment in the left/right branch, b_{i2} be the total payment for the left leader path in the right branch and b_{i3} be the total payment on the right leader path in the right branch. Let $b_0 = b(s_0, a_L)$. And note that clearly any optimal payment is such that $b^*(s_0, a_R) = 0$, since the leader wishes to incentivize the follower to play a_L to obtain a positive return.

With this, we can write down the optimization program under both payment settings.

- Under trajectory payment:

$$
\begin{aligned}
M^*_{trajectory} = \max_{\pi_1,\dots,\pi_H,b} \quad & \sum_{i=1}^{H} v_i x_i - \sum_{i=1}^{H} b_{i1} x_i - \sum_{i=1}^{H}((1-\pi_i)b_{i2} + \pi_i(b_{i3}) + b_{i4})y_i - b_0 \\
\text{s.t.} \quad & \sum_{i=1}^{H}(-w_i + b_{i1})x_i + \sum_{i=1}^{H}((1-\pi_i)b_{i2} + \pi_i(-w_i + b_{i3}) + b_{i4})y_i + b_0 \geq -W \\
& x_i = \mathbb{1}\left\{-w_i + b_{i1} \geq (1-\pi_i)b_{i2} + \pi_i(-w_i + b_{i3}) + b_{i4}\right\}, y_i = 1 - x_i \\
& b_{i1}, b_{i2}, b_{i3}, b_{i4}, b_0 \geq 0
\end{aligned}
$$

- Under upfront payment:

$$
\begin{aligned}
M^*_{upfront} = \max_{\pi_1,\dots,\pi_H,b} \quad & \sum_{i=1}^{H} v_i x_i - \sum_{i=1}^{H}(b_{i1} + b_{i2} + b_{i3} + b_{i4})x_i - \sum_{i=1}^{H}(b_{i1} + b_{i2} + b_{i3} + b_{i4})y_i - b_0 \\
\text{s.t.} \quad & \sum_{i=1}^{H}(-w_i + b_{i1})x_i + \sum_{i=1}^{H}((1-\pi_i)b_{i2} + \pi_i(-w_i + b_{i3}) + b_{i4})y_i + b_0 \geq -W \\
& x_i = \mathbb{1}\left\{-w_i + b_{i1} \geq (1-\pi_i)b_{i2} + \pi_i(-w_i + b_{i3}) + b_{i4}\right\}, y_i = 1 - x_i \\
& b_{i1}, b_{i2}, b_{i3}, b_{i4}, b_0 \geq 0
\end{aligned}
$$

Now, consider any optimal policy π^*, b^*. Let $x_i^*(\pi^*, b^*), y_i^*(\pi^*, b^*)$ denote the follower BR to π^*, b^*. Note that $x_i^*(\pi^*, b^*), y_i^*(\pi^*, b^*) = x_i^*(\pi_i^*, b_{i\cdot}^*), y_i^*(\pi_i^*, b_{i\cdot}^*)$.

Claim 1: We will show that there exists an optimal solution π', b' of the form:

1. If $y_i^*(\pi^*, b^*) = 1$, then $\pi_i' = 0$, $b_{i1}' = b_{i3}' = b_{i4}' = 0$.
2. If $x_i^*(\pi^*, b^*) = 1$, then $\pi_i' = 1$, $b_{i2}' = b_{i3}' = b_{i4}' = 0$.

We will construct a π', b' based on π^*, b^* that satisfy the desired two properties. The construction is as follows:

1. If $y_i^*(\pi^*, b^*) = 1$, we have two cases:
   Case 1: If $(1-\pi_i^*)b_{i2}^* + \pi_i^*(-w_i + b_{i3}^*) + b_{i4}^* \geq 0$, set $\pi_i' = 0$, $b_{i2}' = (1-\pi_i^*)b_{i2}^* + \pi_i^*(-w_i + b_{i3}^*) + b_{i4}^*$, $b_{i1}' = b_{i3}' = b_{i4}' = 0$.
   Firstly, $y_i'(\pi', b') = 1$ still since,

$$
\begin{aligned}
& -w_i + b_{i1}' \\
& \leq -w_i + b_{i1}^* \\
& < (1-\pi_i^*)b_{i2}^* + \pi_i^*(-w_i + b_{i3}^*) + b_{i4}^* \\
& = b_{i2}' = (1-\pi_i')b_{i2}'
\end{aligned}
$$

Thus, feasibly still holds since,

$$
((1-\pi_i^*)b_{i2}^* + \pi_i^*(-w_i + b_{i3}^*) + b_{i4}^*)y_i = ((1-\pi_i')b_{i2}' + \pi_i'(-w_i + b_{i3}') + b_{i4}')y_i
$$

Finally, the trajectory payment objective value increases as total payment for ith step decreases from $(1-\pi_i^*)b_{i2}^* + \pi_i^* b_{i3}^* + b_{i4}^*$ to $(1-\pi_i^*)b_{i2}^* + \pi_i^*(-w_i + b_{i3}^*) + b_{i4}^*$.
The upfront payment objective value increases as total payment for ith step decreases from $b_{i1}^* + b_{i2}^* + b_{i3}^* + b_{i4}^*$ to $(1-\pi_i)b_{i2}^* + \pi_i(-w_i + b_{i3}^*) + b_{i4}^*$.
Case 2: If $(1-\pi_i^*)b_{i2}^* + \pi_i^*(-w_i + b_{i3}^*) + b_{i4}^* < 0$, set $\pi_i' = 0$ and $b_{i1}' = b_{i2}' = b_{i3}' = b_{i4}' = 0$.
$y_i'(\pi', b') = 1$ still since,

$$\begin{aligned}
&-w_i + b_{i1}' \\
&\leq -w_i + b_{i1}^* \\
&< (1-\pi_i^*)b_{i2}^* + \pi_i^*(-w_i + b_{i3}^*) + b_{i4}^* \\
&< 0 = (1-\pi_i')b_{i2}'
\end{aligned}$$

Thus, feasibly still holds since,

$$((1-\pi_i^*)b_{i2}^* + \pi_i^*(-w_i + b_{i3}^*) + b_{i4}^*)y_i < 0 \cdot 1 = ((1-\pi_i')b_{i2}' + \pi_i'(-w_i + b_{i3}') + b_{i4}')y_i$$

Finally, the trajectory payment objective value increases as total payment for ith step decreases from $(1-\pi_i)b_{i2}^* + \pi_i b_{i3}^* + b_{i4}^*$ to 0.
The upfront payment objective value increases as total payment for ith step decreases from $b_{i1}^* + b_{i2}^* + b_{i3}^* + b_{i4}^*$ to 0.

2. If $x_i^*(\pi^*, b^*) = 1$, set $\pi_i' = 1$, $b_{i1}' = b_{i1}^*$ and $b_{i2}' = b_{i3}' = b_{i4}' = 0$.
$x_i(\pi', b') = 1$ still because,

$$\begin{aligned}
&-w_i + b_{i1}' \\
&= -w_i + b_{i1}^* \\
&\geq (1-\pi_i^*)b_{i2}^* + \pi_i^*(-w_i + b_{i3}^*) + b_{i4}^* \\
&\geq \pi_i^* \cdot -w_i \\
&\geq \pi_i' \cdot -w_i
\end{aligned}$$

Feasibility still holds since $(-w_i + b_{i1}^*)x_i = (-w_i + b_{i1}')x_i$.
Finally, the trajectory payment objective value is unchanged as total payment for ith step is still b_{i1}^*.
The upfront payment objective value increases as total payment for ith step decreases from $b_{i1}^* + b_{i2}^* + b_{i3}^* + b_{i4}^*$ to b_{i1}^*.

Simplification of optimization programs:

From the above, we can introduce the constraints that define π', b' in the optimization programs without loss of optimality.

- The trajectory payment program with the constraints as in Claim 1 simplifies to:

$$\begin{aligned} M^*_{trajectory} = \max_{\pi_1,\ldots,\pi_H,b} \quad & \sum_{i=1}^{H} v_i x_i - \sum_{i=1}^{H} b_{i1} x_i - \sum_{i=1}^{H} b_{i2} y_i - b_0 \\ \text{s.t.} \quad & \sum_{i=1}^{H} (-w_i + b_{i1}) x_i + \sum_{i=1}^{H} b_{i2} y_i + b_0 \geq -W \\ & x_i = \mathbb{1}\{\pi_i = 1\}, y_i = \mathbb{1}\{\pi_i = 0\} \\ & b_{i1}, b_{i2}, b_0 \geq 0 \\ & \pi_i \in \{0,1\} \end{aligned}$$

- The upfront payment program with the constraints as in Claim 1 simplifies to:

$$\begin{aligned} M^*_{upfront} = \max_{\pi_1,\ldots,\pi_H,b} \quad & \sum_{i=1}^{H} v_i x_i - \sum_{i=1}^{H} b_{i1} x_i - \sum_{i=1}^{H} b_{i2} y_i - b_0 \\ \text{s.t.} \quad & \sum_{i=1}^{H} (-w_i + b_{i1}) x_i + \sum_{i=1}^{H} b_{i2} y_i + b_0 \geq -W \\ & x_i = \mathbb{1}\{\pi_i = 1\}, y_i = \mathbb{1}\{\pi_i = 0\} \\ & b_{i1}, b_{i2}, b_0 \geq 0 \\ & \pi_i \in \{0,1\} \end{aligned}$$

Reduction to PARTITION:

Since both programs are now the same under the constraints, we will call the optimal objective value $M^* = M^*_{trajectory} = M^*_{upfront}$.

$$\begin{aligned} M^* = \max_{\pi_1,\ldots,\pi_H,b} \quad & \sum_{i=1}^{H} v_i x_i - \sum_{i=1}^{H} b_{i1} x_i - \sum_{i=1}^{H} b_{i2} y_i - b_0 \\ \text{s.t.} \quad & \sum_{i=1}^{H} (-w_i + b_{i1}) x_i + \sum_{i=1}^{H} b_{i2} y_i + b_0 \geq -W \\ & x_i = \mathbb{1}\{\pi_i = 1\}, y_i = \mathbb{1}\{\pi_i = 0\} \\ & b_{i1}, b_{i2}, b_0 \geq 0 \\ & \pi_i \in \{0,1\} \end{aligned}$$

Write $B = \sum_{i=1}^{H} b_{i1} x_i + \sum_{i=1}^{H} b_{i2} y_i + b_0$ to simplify the program to:

$$\begin{aligned} M^* = \max_{x1,\ldots,x_H,B} \quad & \sum_{i=1}^{H} v_i x_i - B \\ \text{s.t.} \quad & \sum_{i=1}^{H} w_i x_i \leq W + B \\ & x_i \in \{0,1\} \\ & B \geq 0 \end{aligned}$$

Now consider Markov game instances where $v_i \in \mathbb{Z}^+$, $w_i = 2v_i$, $W = \sum_{i=1}^H v_i$.

$$\begin{aligned} M^* = \max_{x1,\dots,x_H,B} \quad & \sum_{i=1}^H v_i x_i - B \\ \text{s.t.} \quad & 2\sum_{i=1}^H v_i x_i \leq W + B \\ & x_i \in \{0,1\} \\ & B \geq 0 \end{aligned}$$

Claim 2: We can use this subset of the instances to answer the PARTITION decision problem for any PARTITION instance. We output YES if the objective value M^* of the optimal leader solution computed is exactly $M^* = W/2$ and NO otherwise.

To see this, we will show that $M^* = \frac{W}{2}$ if and only if there is a balanced partition.

$(\Rightarrow)$: Suppose $x_1, \dots, x_H, B$ achieves M^*, we have $\sum_{i=1}^H v_i x_i - B \leq \frac{W}{2} - \frac{B}{2}$ from the feasibility condition. Therefore, for this to be $\frac{W}{2}$, it must be the case that $B = 0$, which implies that $\sum_{i=1}^H v_i x_i = \frac{W}{2}$. And $S = \{i : x_i = 1\}$ gives the balanced partition.

$(\Leftarrow)$: Given some balanced partition S, define $x_i = \mathbb{1}\{i \in S\}$ and $B = 0$. It is feasible because $2\sum_{i=1}^H v_i x_i = 2 \cdot W/2 \leq W + 0$. And it achieves the optimal objective value $\frac{W}{2}$ because $M^* \leq W/2$: from the feasibility constraint, for all x's and B, $\sum_{i=1}^H v_i x_i - B \leq \frac{W}{2} - \frac{B}{2}$.

□

Remark 24. *Scaling all parameters $\langle v_i, w_i, W\rangle$ to $\langle v_i/\kappa, w_i/\kappa, W/\kappa\rangle$ in the reduction to PARTITION allows us to show hardness for all $\kappa > 0$.*

Corollary 9. *There exists no computationally efficient, no-regret learning algorithm.*

Proof. Note that here M^* is an integer multiple of $1/2$. Suppose by contradiction that we had such an algorithm with regret T^α ($\alpha < 1$). We can then set T large enough such that $T^\alpha/T < 1/2$. This allows us to infer M^* exactly, thus giving us a computationally efficient algorithm for answering the PARTITION decision problem.

□

9.8.2 Positive Results

In terms of positive results, we show that there is a polynomial time algorithm for learning the optimal leader policy and payment, even in general-sum games. This holds when the MDP has tree structure.

Proposition 49. *Under Markov games that are deterministic trees, there exists a polynomial-time planning algorithm that computes the optimal policy and payment.*

Proof. We know that the optimal leader policy and payment induce some root to leaf path τ^*. And so, it is sufficient to examine all possible root to leaf paths τ, which is efficient as there are $|S_H|$ root to leaf paths.

Characterizing policies with payment: For each path τ, let the set of leader policies, payment that realize τ be $\Gamma(\tau)$. We have the following characterization.

$\pi, b \in \Gamma(\tau)$ iff:

1. $\pi(s_i) = a_i \quad \forall s_i, a_i \in \tau$
2. $Q^{\pi,BR(\pi)}(s_j, a_j, b_j; r^F + b) \geq \max_{b'_j \neq b_j} Q^{\pi,BR(\pi)}(s_j, a_j, b'_j; r^F + b) \quad \forall s_j, a_j, b_j \in \tau$

The $(\Leftarrow)$ direction is clear. And the $(\Rightarrow)$ direction can be shown by proving the contrapositive. Indeed, if the first condition is not satisfied and $\pi(s_i) \neq a_i$, then $\pi, b \notin \Gamma(\tau)$. Or, if the second condition is not satisfied and $Q^{\pi,BR(\pi)}(s_j, a_j, b_j; r^F + b) < \max_{b'_j \neq b_j} Q^{\pi,BR(\pi)}(s_j, a_j, b'_j; r^F + b)$, then the follower would play $b'_j \neq b_j$ at s_j, which implies $\pi, b \notin \Gamma(\tau)$.

We are interested in the pair of policy, payment $\pi^*(\tau), b^*(\tau) \in \Gamma(\tau)$ that are the optimum of the following optimization program. Note that because $\pi^*(\tau), b^*(\tau) \in \Gamma(\tau)$, they realize τ, thus fixing the leader's return to $r^L(\tau)$. And so, $\pi^*(\tau), b^*(\tau)$ are such that they minimize the total payment needed to realize τ:

$$\begin{aligned}
\pi^*, b^* = \operatorname*{argmin}_{\pi, b \geq 0} \quad & \sum_{s_i \in S_i, a_i \in A, b_i \in B} b(s_i, a_i, b_i) \\
\text{s.t.} \quad & Q^{\pi,BR(\pi)}(s_j, a_j, b_j; r^F + b) \geq \max_{b'_j \neq b_j} Q^{\pi,BR(\pi)}(s_j, a_j, b'_j; r^F + b) \quad \forall s_j, a_j, b_j \in \tau \\
& \pi(s_i) = a_i \quad \forall s_i, a_i \in \tau
\end{aligned}$$

Simplifying Optimization Program: Next, we make two observations that simplify the optimization program:

1. Since $\pi, b \in \Pi(\tau)$ and they realize τ, we have that:
$$Q^{\pi,BR(\pi)}(s_j, a_j, b_j; r^F + b) = r^F(\tau[j:]) + \sum_{i \geq j, s_i, a_i, b_i \in \tau} b(s_i, a_i, b_i)$$
where as the follower is only rewarded payment on the trajectory.
2. Next, we observe that $b^*(s_i, a_i, b_i) = 0$ for $s_i, b_i \notin \tau$.
If not, setting $b(s_i, a_i, b_i) = 0$ maintains feasibility, since it can only reduce the RHS in the constraints, while reducing the objective.

Thus, we can simplify the optimization program to:

$$\begin{aligned}
\pi^*, b^* = \operatorname*{argmin}_{\pi, b \geq 0} \quad & \sum_{s_i \in S_i, a_i \in A, b_i \in B} b(s_i, a_i, b_i) \\
\text{s.t.} \quad & r^F(\tau[h:]) + \sum_{i \geq h, s_i, a_i, b_i \in \tau} b(s_i, a_i, b_i) \geq \max_{b'_h \neq b_h} Q^{\pi,BR(\pi)}(s_h, a_h, b'_h; r^F) \\
& \pi(s_i) = a_i \quad \forall s_i, a_i \in \tau
\end{aligned}$$

Let $\pi^- \in \operatorname{argmin}_\pi V^{\pi,BR(\pi)}(\cdot; r^F)$, which may be computed by Nash-VI in polynomial time. This minimax policy is such that $\pi^- \in \operatorname{argmin}_\pi V^{\pi,BR(\pi)}(s; r^F)$ for any state $s \in S$.

We claim that without loss of optimality, we can set $\pi^*(s_i') = \pi^-(s_i')$ for all states $s_i' \notin \tau$.

Given a pair of optimal solution (π^*, b^*), let π' be such modification of a π^*.

We observe that (π', b^*) achieves the same objective value, while still being feasible. The former holds by construction as the payment remains unchanged.

$\pi'(s_i) = a_i \quad \forall s_i, a_i \in \tau$ holds still by construction. Now, feasibility holds because at any s_i, for any $b_i' \neq b_i$:

$$\begin{aligned}
&Q^{\pi^*,BR(\pi^*)}(s_i, a_i, b_i'; r^F) \\
&= V^{\pi^*,BR(\pi^*)}(s_{i+1}'; r^F) && \text{(state } s_{i+1}' = P(s_i, a_i, b_i') \text{ deterministically)} \\
&\geq V^{\pi^-,BR(\pi^-)}(s_{i+1}'; r^F) && \text{(definition of } \pi^-) \\
&= V^{\pi',BR(\pi')}(s_{i+1}'; r^F) && (\star) \\
&= Q^{\pi',BR(\pi')}(s_i, a_i, b_i'; r^F)
\end{aligned}$$

$(\star)$: Due to the tree structure of the MDP, the set of successor states of s_{i+1}' does not contain any leader states in τ, as they belong to a different branch than the one τ and thus s_{i+1} are in, with the root at state s_i. And so, π''s actions starting at state s_{i+1}' are exactly the same as that of π^-.

And so, π' is feasible because for every s_h,

$$r^F(\tau[h:]) + \sum_{i \geq h, s_i, a_i, b_i \in \tau} b(s_i, a_i, b_i) \geq \max_{b_h' \neq b_h} Q^{\pi^*,BR(\pi^*)}(s_h, a_h, b_h'; r^F) \geq \max_{b_h' \neq b_h} Q^{\pi',BR(\pi')}(s_h, a_h, b_h'; r^F)$$

Minimal Payment LP: Since we have fully determined an optimal policy π^*, the optimal payment may be found by solving the following LP:

$$\begin{aligned}
b^* = \operatorname*{argmin}_{b \geq 0} \quad & \sum_{s_i, a_i, b_i \in \tau} b(s_i, a_i, b_i) \\
\text{s.t.} \quad & r^F(\tau[h:]) + \sum_{i \geq h, s_i, a_i, b_i \in \tau} b(s_i, a_i, b_i) \geq \max_{b_h' \neq b_h} Q^{\pi,BR(\pi)}(s_h, a_h, b_h'; r^F) \\
& \pi(s_i) = a_i \quad \forall s_i, a_i \in \tau \\
& \pi(s_i') = \pi^-(s_i') \quad \forall s_i' \notin \tau
\end{aligned}$$

□

9.8.2.1 General sum learning in deterministic tree

Learning Setting: As a quick recap of the learning setting in the general-sum case, the only unknown is the follower rewards r^F. The MDP is deterministic and so the transitions are known. Also, the leader knows his own reward r^L.

Remark 25. *To complete the result, there is a simple exploration strategy using payment for general-sum, deterministic trees, as exploration needs to only recover rewards. This strategy allows us to reduce learning to planning, and then apply Algorithm 23.*

Explore: To see this, for each root-to-leaf path $\tau = \big\{(s_i, a_i, b_i)\big\}_{i=1}^{H}$, set $b(s_i, a_i, b_i) = H$ for $s_i, a_i, b_i \in \tau$. Then, setting the leader policy to match τ will realize τ, as the follower is incentivized to play actions that realize this path. With this exploration strategy, we obtain estimates of the stochastic reward at every node of the tree $\hat{r}^F$ to precision δ w.h.p. (better than $1 - 1/T$) after $m = \tilde{O}(|S| \cdot 1/\delta^2)$ number of episodes.

Exploit: Now, we bound the instantaneous regret when we plan using $\hat{r}^F$. This regret arises to due to the difference in payment computed by the LP, since r^L is known exactly. The LP admits closed form solution: $b^*(s_h, a_h, b_h) = \max(\max_{b'_h \neq b_h} Q^{\pi^-,\mu(\pi^-)}(s_h, a_h, b'_h; r^F) - \sum_{i \geq h, s_i, a_i, b_i \in \tau} r^F(s_i, a_i, b_i) - \sum_{i \geq h+1, s_i, a_i, b_i \in \tau} b^*(s_i, a_i, b_i), 0)$. Define $\hat{b}(s_h, a_h, b_h)$ analogously under $\hat{r}^F$ and $\hat{\pi}^-$ (computed using $\hat{r}^F$).

Since the function $\max(x, 0)$ is $1-$Lipschitz, we will bound the argument of the function. First, $|\sum_{i \geq h, s_i, a_i, b_i \in \tau} \hat{r}^F(s_i, a_i, b_i) - r^F(s_i, a_i, b_i)| \leq (H - h)\delta$.

Second, for any action b'_h, let $P(s'_{h+1} | s_h, a_h, b'_h) = 1$. We have that:

$$
\begin{aligned}
&Q^{\pi^-,\mu(\pi^-)}(s_h, a_h, b'_h; r^F) - Q^{\hat{\pi}^-,\mu(\hat{\pi}^-)}(s_h, a_h, b'_h; \hat{r}^F) \\
&= V^{\pi^-,\mu(\pi^-)}(s'_{h+1}; r^F) - V^{\hat{\pi}^-,\mu(\hat{\pi}^-)}(s'_{h+1}; \hat{r}^F) \\
&= \max_{\mu} V^{\pi^-,\mu}(s'_{h+1}; r^F) - \max_{\mu} V^{\pi^-,\mu}(s'_{h+1}; \hat{r}^F) + \max_{\mu} V^{\pi^-,\mu}(s'_{h+1}; \hat{r}^F) - \max_{\mu} V^{\hat{\pi}^-,\mu}(s'_{h+1}; \hat{r}^F) \\
&\geq \max_{\mu} V^{\pi^-,\mu}(s'_{h+1}; r^F) - \max_{\mu} V^{\pi^-,\mu}(s'_{h+1}; \hat{r}^F) + 0 \qquad (\hat{\pi}^- \in \operatorname{argmin}_{\pi} \max_{\mu} V^{\pi,\mu}(s'_{h+1}; \hat{r}^F)) \\
&= \max_{\mu} V^{\mu}(s'_{h+1}; \pi^-, r^F) - \max_{\mu} V^{\mu}(s'_{h+1}; \pi^-, \hat{r}^F) \\
&\geq -\delta
\end{aligned}
$$

(due to same visitation probability, any policy μ's return under π^-, r^F vs $\pi^-, \hat{r}^F$ differs by $\leq \delta$)

The other direction follows analogously to get that: $|Q^{\pi^-,\mu(\pi^-)}(s_h, a_h, b'_h; r^F) - Q^{\hat{\pi}^-,\mu(\hat{\pi}^-)}(s_h, a_h, b'_h; \hat{r}^F)| \leq \delta$.

Therefore, we have that $|\hat{b}(s_h, a_h, b_h) - b^*(s_h, a_h, b_h)| \leq O((H - h)^2\delta)$. This means that the instantaneous regret due to $\sum_{s_i, a_i, b_i \in \tau} \hat{b}(s_i, a_i, b_i) - \sum_{s_i, a_i, b_i \in \tau} b^*(s_i, a_i, b_i) \leq O(H^3\delta)$. Choosing $\delta = T^{-1/3}$, we have that the cumulative regret is $O(\delta^{-2}) + O(TH^3\delta)$ respectively from the explore and exploit phase for a total of $O(T^{2/3})$ regret.

9.8.2.2 Computing Local Stackelberg Optimum

The computational hardness result does not preclude algorithms for other solution concepts. Due to the intractability of computing a global Stackelberg optimum, we may be interested in computing instead a local Stackelberg optimum. To this end, we derive a first order approach to this end, illustrating that this looser solution concept can be computed. We focus on trajectory-based payment below as the upfront payment gradient w.r.t. b for f is straightforward.

Value-based Penalty: We can use the existing idea of encoding the BR as Langrangian with value-based penalty [259]. To recap the notation for the policy-based method, leader policy has policy parameters denoted by x and payment b. Follower has policy with parameter y.

$$\min_{x,y,b} \underbrace{-V^{\pi_x,\pi_y}(s_0; r^L - b)}_{f(x,y,b)} + \lambda \underbrace{(-V^{\pi_x,\pi_y}(s_0; r^F + b) + \max_{y'} V^{\pi_x,\pi_{y'}}(s_0; r^F + b))}_{p(x,y,b)}$$

From [259], we have that,

$$\nabla_{x,b} p(x,b,y) = \nabla_{x,b} V^{\pi_y}_{M(x)}(\rho) + \nabla_{x,b} V^{\pi_y}_{M(x)}(\rho)\big|_{\pi=\pi^*(x,b)}$$

where $\pi^*(x,b)$ is the optimal BR to leader policy x and b, and $M(x)$ is the single-agent MDP w.r.t. follower parameterized by leader policy π_x.

We will now describe the gradient component by component, as the overall gradient is the sum of Stackelberg game and reward shaping gradients:

1.
$$\nabla_x f(x,y,b) = -\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t Q^{\pi_x}_{\pi_y}(s_t, a_t; r^L - b) \nabla \log \pi_x(a_t|s_t) | s_0 \sim \rho, \pi_y, \pi_x]$$

2.
$$\nabla_y f(x,y,b) = -\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t Q^{\pi_y}_{\pi_x}(s_t, b_t; r^L - b) \nabla \log \pi_y(b_t|s_t) | s_0 \sim \rho, \pi_x, \pi_y]$$

3.
$$\nabla_b f(x,y,b) = -\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \nabla_b r(s_t, a_t, b_t; r^L - b) | s_0 \sim \rho, \pi_x, \pi_y]$$

4.
$$\begin{aligned}
&\nabla_x p(x,y,b) \\
&= \nabla_x V^{\pi_y}_{\pi_x}(\rho) + \nabla_x V^{\pi_y}_{\pi_x}(\rho)\big|_{\pi=\pi^*(x,b)} \\
&= -\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t (Q^{\pi_x,\pi_y}(s_t, a_t, b_t; r^f + b) - \tau h(\pi_x)) \nabla \log \pi_x(a_t|s_t) | s_0 \sim \rho, \pi_x, \pi_y] \\
&\quad + -\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t (Q^{\pi_x,\pi^*_y(x,b)}(s_t, a_t, b_t; r^f + b) - \tau h(\pi_x)) \nabla \log \pi_x(a_t|s_t) | s_0 \sim \rho, \pi_x, \pi^*_y(x,b)]
\end{aligned}$$

5.
$$\nabla_y p(x,y,b) = \nabla_y V^{\pi_y}_{\pi_x}(\rho) = -\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t Q^{\pi_y}_{\pi_x}(s_t, b_t; r^F + b) \nabla \log \pi_y(b_t|s_t) | s_0 \sim \rho, \pi_x, \pi_y]$$

6.

$$
\begin{aligned}
&\nabla_b p(x, y, b) \\
&= \nabla_b V_{\pi_x}^{\pi_y}(\rho) + \nabla_{x,b} V_{\pi_x}^{\pi_y}(\rho)\big|_{\pi=\pi^*(x,b)} \\
&= -\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \nabla_b r(s_t, a_t, b_t; r^F + b) | s_0 \sim \rho, \pi_x, \pi_y] \\
&+ -\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \nabla_b r(s_t, a_t, b_t; r^F + b) | s_0 \sim \rho, \pi_x, \pi_y^*(x, b)]
\end{aligned}
$$

9.9 Proofs for Learning Results in Cooperative Games without Payment

From the hardness of computing the optimal policy in the general sum setting, we know that this translates to the computaional hardness of learning. On the other hand, we know that planning in cooperative games is efficient. This begs the questions, is learning in cooperative games efficient?

We delve into this question by first considering cooperative games without subsidy, as this has been previously unaccounted for in prior literature. Surprisingly, we find that learning in Cooperative Markov games can be prohibitively hard, statistically.

Lemma 70. *Suppose an algorithm A tries to identify $a^* \in \{a_1, ..., a_n\}$. Each step, it can make a query a_i and receive signal $\mathbb{1}\{a_i = a^*\}$. Then, for any possibly randomized algorithm A with query budget m. If $m \leq n/4$, then:*

$$
\Pr_{a^* \in unif([n]), A}(a^* \notin \{a_1, ..., a_m\}) \geq \frac{1}{2}
$$

Proof. Let a^* be drawn uniformly from $[n]$. Let $a_1, .., a_m$ be queries made by algorithm. And let $Y_i = \mathbb{1}\{a_i = a^*\}$. We will show by induction that for $j \leq m$,

$$
\Pr(Y_{:j} \neq 0) = \Pr(a^* \in \left\{a_1, ..., a_j\right\}) \leq \frac{2j}{n}
$$

Base Case: Since a_1 is independent of a^*, $a^*|a_1$ is uniform over $[n]$. Thus,

$$
\Pr(a^* = a_1) = \frac{1}{n} \leq \frac{2}{n}
$$

Induction Step: Suppose $\Pr(a^* \in \left\{a_1, ..., a_{j-1}\right\}) \leq \frac{2(j-1)}{n}$, we have:

$$
\begin{aligned}
&\Pr(a^* \in \{a_1, ..., a_j\}) \\
&= \Pr(a^* \in \{a_1, ..., a_{j-1}\}) + P(a^* = a_j, a^* \notin \{a_1, ..., a_{j-1}\}) \\
&\leq \frac{2(j-1)}{n} + P(a^* = a_j, a^* \notin \{a_1, ..., a_{j-1}\}) \\
&\leq \frac{2(j-1)}{n} + P(a^* = a_j | a^* \notin \{a_1, ..., a_{j-1}\}) \\
&= \frac{2(j-1)}{n} + \mathbb{E}[P(a^* = a_j | a^* \notin \{a_1, ..., a_{j-1}\}) | a_1, ..., a_{j-1}] \\
&= \frac{2(j-1)}{n} + \frac{1}{n-(j-1)} && (\star) \\
&\leq \frac{2(j-1)}{n} + \frac{2}{n} && (j \leq n/4)
\end{aligned}
$$

$(\star)$: for any fixed $a_1, .., a_{j-1}$, conditioned on $a^* \notin \{a_1, ..., a_{j-1}\}$, a^* is uniform over $[n] \setminus \{a_1, ..., a_{j-1}\}$. Thus,

$$
P(a^* = a_j | a^* \notin \{a_1, ..., a_{j-1}\}) \leq \frac{1}{n-(j-1)}
$$

In conclusion,

$$
P(a^* \in \{a_1, ..., a_m\}) \leq 2j/n \leq 1/2 \Rightarrow P(a^* \notin \{a_1, ..., a_m\}) \geq 1/2
$$

where we use that $j \leq m/4 \Rightarrow 2j/n \leq 1/2$.

□

Theorem 55. *There exists a turn-based Stochastic Tree Markov game such that: any (possibly randomized) algorithm that returns the optimal policy with probability at least* $1/2$ *requires at least* $\Omega(2^{|S|})$ *number of episodes.*

Proof. **Setup:** Consider a two-branch MDP, where the follower chooses first action a_L or a_R, which deterministically transitions to the left and right branch. The leader has return $1 - 1/2|S|$ for the left branch. In the right branch, this transitions with uniform probability $1/|S|$ to one of $|S|$ possible states, each state has two possible actions with reward 0 and 1.

Then, we see that each time step, the leader will choose right branch policy $(\pi_t(s_1), ..., \pi_t(s_m))$, and receive feedback $\mathbb{1}\{\pi_t = \pi^*\} = \mathbb{1}\{\mu(\pi_t) = a_R\}$, as $\mu(\pi_t) = a_R \Leftrightarrow r(\pi_t) \geq 1 - 1/2|S| \Leftrightarrow r(\pi_t) = 1 \Leftrightarrow \pi_t = \pi^*$.

To finish, we may use Lemma 70 to get that any algorithm with budget at most $T \leq |\Pi|/4$ will be s.t. $P_A(\pi^* \in \{\pi_1, .., \pi_T\}) \leq 1/2$. Hence, using the contrapositive, any algorithm such that $P_A(\pi^* \in \{\pi_1, .., \pi_T\}) > 1/2$ must have budget $T \geq |\Pi|/4 + 1 = 2^{|S|}/4 + 1 = \Omega(2^{|S|})$.

□

Theorem 56. *There exists a turn-based Deterministic DAG Markov game such that: any (possibly randomized) algorithm that returns the optimal policy with probability at least* $1/2$ *requires at least* $\Omega(2^{|H|})$ *number of episodes.*

Proof. **Setup:** Consider a two-branch MDP, where the follower chooses first action a_L or a_R, which deterministically transitions to the left and right branch. The leader has return $H - 1/2$ for the left branch. In the right branch, there are two actions at each of the H time steps with reward 0 and 1.

Then, we see that each time step, the leader will choose right branch policy $(\pi_t(s_1), ..., \pi_t(s_n))$, and receive feedback $\mathbb{1}\left\{\pi_t = \pi^*\right\} = \mathbb{1}\left\{\mu(\pi_t) = a_R\right\}$, as $\mu(\pi_t) = a_R \Leftrightarrow r(\pi_t) \geq H - 1/2 \Leftrightarrow r(\pi_t) = H \Leftrightarrow \pi_t = \pi^*$.

To finish, we may use the lemma above to get that any algorithm with budget at most $T \leq |\Pi|/4$ will be s.t. $P_A(\pi^* \in \{\pi_1, .., \pi_T\}) \leq 1/2$. Hence, using the contrapositive, any algorithm such that $P_A(\pi^* \in \{\pi_1, .., \pi_T\}) > 1/2$ must have budget $T \geq |\Pi|/4 + 1 = 2^H/4 + 1 = \Omega(2^H)$.

□

Algorithm 25 Learning Algorithm for Deterministic Tree Markov Game without Payment

```
for all root to leaf paths $\tau = s_1, a_1, b_1, ..., s_H, a_H, b_H$ do
    Define $\pi(s_i) = a_i$ for $s_i, a_i \in \tau$, set $\pi(s)$ to any arbitrary action in states not on $\tau$
    Commit to $\pi$ and observe if $\tau$ is realized by $\mu(\pi)$
    If $\tau$ is realized, apply $\pi$ $m$ times and record estimated return $\hat{r}^L(\tau)$
Return the leader policy that that has realized $\tau$ and has the maximal $\hat{r}^L(\tau)$
```

Learning Setting: As a quick recap of the learning setting in the cooperative case, the only unknown is the leader rewards r^L (the same as that of the follower) The MDP is deterministic and so the transitions are known.

Proposition 50. *Under Markov games that are deterministic trees, then there exists a polynomial-time algorithm that can learn a near-optimal policy.*

Proof. We know that there is some root to leaf path τ^* such that $\tau^* \in \operatorname{argmax}_\tau r^L(\tau)$. Our goal is to search for a leader policy that realizes τ^*. Note that it is sufficient to simply find τ^*, as any leader policy that matches τ^* will induce τ^* as the follower's reward is the same as that of the leader.

To find τ^*, it is sufficient to iterate through all possible root to leaf paths τ. There are $|S_H|$ many policies, and so this can be done in polynomial time. For each path τ, choose any leader policy π with $\pi(s_i) = a_i$ for $s_i, a_i \in \tau$. If τ is not realized by $\pi, \mu(\pi)$, then this means $r^L(\tau)$ must be dominated by a different path's $r^L(\tau')$. And so, $\tau \neq \tau^*$. Thus, after iterating through all possible τ's, we must have observed τ^* among the paths realized.

We can then identify a near-optimal policy by choosing the policy, whose path $\hat{\tau}$ has the maximal $\hat{r}^L(\cdot)$. This gives us a near-optimal policy with high probability (e.g. better than $1 - 1/T$), as after $m = \widetilde{O}(1/\delta^2)$ number of episodes, we have estimated every potential optimal path return $\hat{r}^L(\tau)$ to precision δ w.h.p. And so, our returned policy's return (i.e. $\hat{r}^L(\tau)$) must be sub-optimal by at most δ. Please see Algorithm 25 for the algorithm.

□

Finally, we illustrate that this statistical hardness is surprisingly difficult to overcome, even when we relax the BR nature of the follower.

Theorem 57. *There exists a turn-based Deterministic DAG Markov game such that: any (possibly randomized) algorithm that outputs the optimal policy given λ-Entropy-regularized BR with probability at least $1/2$ requires at least $\Omega(\exp(\lambda^2 H/8))$ episodes if $\lambda \leq 1$ and $\Omega(\exp(H/8))$ episodes if $\lambda > 1$.*

Proof. To show this, we will use Yao's lemma. Let $\mathcal{D}$ be the uniform distribution over all MDP instances $\mathcal{X}$, where the distribution is uniform over which of the two actions at each time step achieves reward 1 (and the other 0). Define $\lambda' = \min(\lambda, 1/\lambda)$ and let:

$$m = \min(\exp(\lambda\lambda' H/8), \exp(H/8))/16$$

Define $cost(A, t, x) = \mathbb{1}(a^* \notin \{a_1, ..., a_t\})$, which is the probability of A not outputting the optimal policy a^* after t episodes.

We will show that for any deterministic algorithm A, if $t \leq m$, with probability at least $\frac{1}{2}$ over the choice of $x \sim \mathcal{D}$, $cost(A, t, x) = 1$. That is, $\min_A \mathbb{E}_{x\sim\mathcal{D}}[cost(A, t, x)] \geq \frac{1}{2}$, which by Yao's lemma means that any randomized algorithms R has $\min_R \max_{x\in\mathcal{X}} \mathbb{E}[cost(R, t, x)] \geq \frac{1}{2}$).

For some optimal code a^*, let the set $S_{H/4}(a^*)$ be all length-H binary combinations with Hamming distance at most $H/4$ or at least $3H/4$ from a^*. Let $S'_{H/4}(a^*)$ be its complement i.e. the set of codes with Hamming distance between $H/4$ and $3H/4$ with respect to a^*.

First, for any a^*, we bound the cardinality of the set $S_{H/4}(a^*)$: $|S_{H/4}(a^*)| = 2(\binom{H}{H} + ... + \binom{H}{3H/4}) = 2(\binom{H}{0} + ... + \binom{H}{H/4})$. And $2(\sum_{i=0}^{1/4\cdot H} \binom{n}{i}) \leq 2 \cdot 2^H \exp(-2H(1/2 - 1/4)^2) \Rightarrow \frac{|S_{H/4}(a^*)|}{2^H} \leq 2\exp(-H/8)$.

In round t, let the leader's chosen binary code be a_t, and we observe one realization of the random variable $b_t \sim \sigma(\lambda \cdot (d_L(a_t, a^*) - 1/2))$, which denotes whether the follower chooses to go left in the BR (1 indicates left). We will use the following lemma, whose proof we defer to the end.

Lemma 71. *For each $t \leq m$, we have that for any chosen code a_t:*

$$\Pr(a_t \notin S_{H/4}(a^*) | b_{:t-1} = 1) \geq 1 - \frac{6}{\exp(H/8)}$$

Then, we have that for any $t \leq m$:

$$\begin{aligned}
&\Pr(b_t = 1 | b_{:t-1} = 1) \\
&\geq \Pr(b_t = 1 | a_t \in S'_{H/4}(a^*), b_{:t-1} = 1) \Pr(a_t \in S'_{H/4}(a^*) | b_{:t-1} = 1) \\
&\geq \frac{1}{1 + \exp(-\lambda[(H - 1/2) - (3H/4 - 1)])} \Pr(a_t \in S'_{H/4}(a^*) | b_{:t-1} = 1) \\
&\geq (1 - \frac{1}{1 + \exp(\lambda H/4)}) \Pr(a_t \in S'_{H/4}(a^*) | b_{:t-1} = 1) \\
&> 1 - \frac{6}{\exp(H/8)} - \frac{1}{1 + \exp(\lambda H/4)} && \text{(using the Lemma above)}
\end{aligned}$$

With this, we have that for any algorithm A with output a_i at episode i:

$$
\begin{aligned}
&\mathbb{E}_{x\sim\mathcal{D}}[cost(A,t,x)] \\
&= P(a^* \notin \{a_1, ..., a_t\}) \\
&\geq P(\bigcup_{i\in[t]} a_i \in S'_{H/4}(a^*)) \\
&\geq P(\bigcup_{i\in[t]} a_i \in S'_{H/4}(a^*), b_{:t} = 1) \\
&= P(b_t = 1 | \bigcup_{i=1}^{t} a_i \in S'_{H/4}(a^*), b_{:t-1} = 1) P(a_t \in S'_{H/4}(a^*) | \bigcup_{i=1}^{t-1} a_i \in S'_{H/4}(a^*), b_{:t-1} = 1) \\
&\qquad P(\bigcup_{i=1}^{t-1} a_i \in S'_{H/4}(a^*), b_{:t-1} = 1) \\
&\geq \frac{1}{1+\exp(-\lambda[(H-1/2)-(3H/4-1)])} P(a_t \in S'_{H/4}(a^*) | \bigcup_{i=1}^{t-1} a_i \in S'_{H/4}(a^*), b_{:t-1} = 1) \\
&\qquad P(\bigcup_{i=1}^{t-1} a_i \in S'_{H/4}(a^*), b_{:t-1} = 1) \quad \text{(using that } a_t \in S'_{H/4}(a^*) \text{ and } b_t \text{ is a function of only } a_t\text{)} \\
&\geq (1 - \frac{1}{1+\exp(\lambda H/4)})(1 - \frac{6}{\exp(H/8)}) P(\bigcup_{i=1}^{t-1} a_i \in S'_{H/4}(a^*), b_{:t-1} = 1) \\
&\qquad\qquad (\Pr(a_t \in S'_{H/4}(a^*) | b_{:t-1} = 1, a_{:t-1}) \geq 1 - \tfrac{6}{\exp(H/8)} \text{ for any fixed sequence } a_{:t-1}) \\
&\geq (1 - \frac{1}{1+\exp(\lambda H/4)} - \frac{6}{\exp(H/8)})^t \qquad \text{(unrolling)} \\
&\geq 1 - \frac{8t}{\min(\exp(H/8), \exp(\lambda H/4))} \\
&\geq 1/2 \qquad (t \leq m)
\end{aligned}
$$

□

Lemma 72. *For each $t \leq m$, we have that for any chosen code a_t:*

$$\Pr(a_t \notin S_{H/4}(a^*) | b_{:t-1} = 1) \geq 1 - \frac{6}{\exp(H/8)}$$

Proof. Let us define $M = \frac{\lambda' H}{8}$ and $T = 1 + \exp(\lambda M)$. We will prove the result holds for any $t \leq T$. This is sufficient as $T \geq \exp(\lambda\lambda' H/8) \geq m$.

We will show a stronger result that for any fixed sequence $a_{:t-1}$, we have that:

$$\Pr(a_t \in S'_{H/4}(a^*) | b_{:t-1} = 1, a_{:t-1}) > 1 - \frac{6}{\exp(H/8)}$$

And so,

$$\Pr(a_t \in S'_{H/4}(a^*)|b_{:t-1} = 1) = \mathbb{E}[\Pr(a_t \in S'_{H/4}(a^*)|b_{:t-1} = 1, a_{:t-1})|a_{:t-1}] > 1 - \frac{6}{\exp(H/8)}$$

Fix any sequence of $a_{:t-1}$ and observed $b_{:t-1} = 1$, first we note that the posterior is of the form:

$$\begin{aligned}
&P(a^* = a|b_{:t-1} = 1, a_{:t-1}) \\
&\propto P(a^* = a) \prod_{i=1}^{t-1} P(b_i = 1|a^* = a, b_{:i-1} = 1, a_{:t-1}) \\
&= P(a^* = a) \prod_{i=1}^{t-1} \sigma(\lambda(d_L(a, a_i) - 1/2)) && (b_i \text{ is a function of only } a^*, a_i)
\end{aligned}$$

Next, since the algorithm is deterministic, a_t is deterministic with given $a_{:t-1}$ and $b_{:t-1}$:

$$\begin{aligned}
&\Pr(a_t \in S'_{H/4}(a^*)|b_{:t-1} = 1, a_{:t-1}) \\
&= \sum_a \Pr(a_t \in S'_{H/4}(a^*)|a^* = a, b_{:t-1} = 1, a_{:t-1}) \Pr(a^* = a|b_{:t-1} = 1, a_{:t-1}) \\
&= \sum_a \mathbb{1}\left\{a \in S'_{H/4}(a_t)\right\} \Pr(a^* = a|b_{:t-1} = 1, a_{:t-1}) \\
&= \sum_{a \in S'_{H/4}(a_t)} \Pr(a^* = a|b_{:t-1} = 1, a_{:t-1})
\end{aligned}$$

It suffices to lower bound the posterior probability mass over $S'_{H/4}(a_t)$: $\sum_{a \in S'_{H/4}(a_t)} P(a^* = a|b_{:t-1} = 1, a_{:t-1})$.

For any chosen code a_i for $i \in [t-1]$, consider all codes up to M hamming distance away from a_i. There are $\sum_{i=0}^{M} \binom{H}{i}$ such codes. This means there are at most $(t-1) \sum_{i=0}^{M} \binom{H}{i}$ codes with at least one posterior update factor less than $\sigma(\lambda(M - 1/2))$.

This implies the remaining set of codes a in $S'_{H/4}(a_t)$ has likelihood factor of at least:

$$\begin{aligned}
&\prod_{i=1}^{t-1} \sigma(\lambda(d_L(a, a_i) - 1/2)) \\
&\geq \prod_{i=1}^{t-1} \sigma(\lambda(M - 1/2)) \\
&\geq (1 - \frac{1}{1 + \exp(\lambda M)})^{t-1} \\
&\geq (1 - \frac{1}{1 + \exp(\lambda M)})^{T} \geq 1/e && (T = 1 + \exp(\lambda M))
\end{aligned}$$

Let N_t be the normalizing factor at episode t. Then, the posterior is such that:

$$
\begin{aligned}
&\sum_{a \in S'_{H/4}(a_t)} \Pr(a^* = a | b_{:t-1} = 1, a_{:t-1}) \\
&\geq \frac{1}{N_t}(2^H - (|S_{H/4}(a_t)| + (t-1) \cdot \sum_{i=1}^{M} \binom{H}{i})) \cdot 1/e \\
&\geq \frac{1}{N_t}(2^H - (|S_{H/4}(a_t)| + T \cdot \sum_{i=1}^{M} \binom{H}{i})) \cdot 1/e \\
&\geq \frac{1}{N_t}(2^H(1 - 2\exp(-H/8) - (1 + \exp(\lambda M))\exp(-2H(1/2 - M/H)^2)) \cdot 1/e \\
&\geq \frac{1}{N_t} \cdot 2^H(1 - 2\exp(-H/8) - 2\exp(\lambda M - 2H(1/2 - 1/8)^2)) \cdot 1/e && (M/H \leq 1/8) \\
&\geq \frac{1}{N_t} \cdot 2^H(1 - 2\exp(-H/8) - 2\exp(-10H/64)) \cdot 1/e && (\lambda M \leq H/8)
\end{aligned}
$$

On the other hand, we also have that:

$$
\begin{aligned}
&\Pr(a_t \in S_{H/4}(a^*) | b_{:t-1} = 1, a_{:t-1}) \\
&= \sum_{a \in S_{H/4}(a_t)} \Pr(a^* = a | b_{:t-1} = 1, a_{:t-1}) \\
&\leq \frac{1}{N_t} \cdot (2^H 2\exp(-H/8)) \cdot 1
\end{aligned}
$$

Therefore, for any $t \leq T$, the probability of selecting a code with return between $H/4$ and $3H/4$ is at least:

$$
\begin{aligned}
&\Pr(a_t \in S'_{H/4}(a^*) | b_{:t-1} = 1, a_{:t-1}) \\
&= 1 - \frac{1}{1 + \Pr(a_t \in S'_{H/4}(a^*) | b_{:t-1} = 1, a_{:t-1}) / \Pr(a_t \in S_{H/4}(a^*) | b_{:t-1} = 1, a_{:t-1})} \\
&\geq 1 - \frac{1}{1 + \frac{1 - 2\exp(-H/8) - 2\exp(-10H/64)}{e[2\exp(-H/8)]}} \\
&\geq 1 - \frac{1}{1 + \exp(H/8)/2e - 2/e} \\
&> 1 - \frac{6}{\exp(H/8)}
\end{aligned}
$$

□

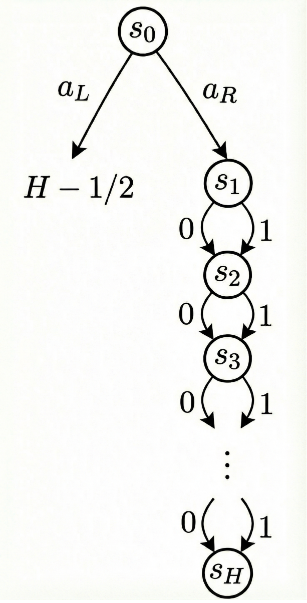


Figure 9.3: Right branch requires getting the right set of H binary actions at states $s_1, ..., s_H$ to exceed the return in the left branch.

9.10 Proofs for Learning Results in Cooperative Games with Payment

Due to the statistical hardness of learning without payment, we now consider whether payment can alleviate this hardness. Note that unlike prior works in bandits [248], we can no longer exhaustively enumerate all possible leader policies, which is feasible only in the bandit setting. We main approach is to adopt the natural idea of setting the outcome-based payments be the bonuses, in order to align incentives during exploration.

9.10.1 General regret guarantees

We first prove two results that are used in all the following regret proofs. The first informs us what is the optimal policy and payment to compete against in the cooperative case.

Lemma 73. *For any π^*, b^* such that:*

$$\begin{aligned} \pi^*, b^* = \operatorname*{argmax}_{\pi,b} \quad & V^{\pi,\mu(\pi;r^F+b)}(s_0; r^L - \kappa \cdot b) \\ s.t. \quad & \mu(\pi; r^F + b) \in \operatorname*{argmax}_{\mu'} V^{\pi,\mu'}(s_0; r^F + b) \end{aligned}$$

If $r^L = r^F$, then we must have $\pi^*, \cdot = \operatorname{argmax}_{\pi,\mu} V^{\pi,\mu}(s_0; r^L)$ and $b^* = 0$.

Proof. We show that for π^* such that it is part of a globally optimal pair $\pi^*, \mu^* = \operatorname{argmax}_{\pi,\mu} V^{\pi,\mu}(s_0; r^L)$.

We claim that $(\pi^*, 0)$ dominates every pair (π, b):

$$
\begin{aligned}
&V^{\pi^*,\mu(\pi^*;r^F)}(s_0;r^L-0)\\
&= V^{\pi^*,\mu^*}(s_0;r^L-0) && (r^F = r^L \Rightarrow \mu^* \text{ is a BR to } \pi^*)\\
&\geq V^{\pi,\mu(\pi;r^F+b)}(s_0;r^L)\\
&\text{(joint optimality } \pi^*,\mu^* = \operatorname{argmax}_{\pi,\mu} V^{\pi,\mu}(s_0;r^L) \text{ implies } \pi^*,\mu^* \text{ dominates } \pi,\mu(\pi;r^F+b))\\
&\geq V^{\pi,\mu(\pi;r^F+b)}(s_0;r^L-\kappa\cdot b)
\end{aligned}
$$

Similarly for upfront payment:

$$
\begin{aligned}
&V^{\pi^*,\mu(\pi^*;r^F)}(s_0;r^L)\\
&= V^{\pi^*,\mu^*}(s_0;r^L) && (r^F = r^L \Rightarrow \mu^* \text{ is a BR to } \pi^*)\\
&\geq V^{\pi,\mu(\pi;r^F+b)}(s_0;r^L)\\
&\text{(joint optimality } \pi^*,\mu^* = \operatorname{argmax}_{\pi,\mu} V^{\pi,\mu}(s_0;r^L) \text{ implies } \pi^*,\mu^* \text{ dominates } \pi,\mu(\pi;r^F+b))\\
&\geq V^{\pi,\mu(\pi;r^F+b)}(s_0;r^L) - \kappa\cdot\sum_{s,a,b} b(s,a,b)
\end{aligned}
$$

□

Lemma 74. *Suppose we can construct an optimistic MDP M_k of the true MDP M. Let the optimal leader policy under M_k be π_k, then:*

$$
\sum_{k=1}^{T} V_M^{\pi^*,\mu_M(\pi^*)}(s_0;r^L) - V_M^{\pi^k,\mu_M(\pi^k)}(s_0;r^L) \leq \sum_{k=1}^{T} V_{M_k}^{\pi^k,\mu_{M_k}(\pi^k)}(s_0;r^L) - V_M^{\pi^k,\mu_{M_k}(\pi^k)}(s_0;r^L)
$$

Proof. By optimality of π_k in M_k and optimism, we have that,

$$
V_{M_k}^{\pi^k,\mu_{M_k}(\pi^k)} \geq V_{M_k}^{\pi^*,\mu_M(\pi^*)} \geq V_M^{\pi^*,\mu_M(\pi^*)}
$$

Therefore, we may bound the instantaneous regret,

$$
\begin{aligned}
&\sum_{k=1}^{T} V_M^{\pi^*,\mu_M(\pi^*)} - V_M^{\pi^k,\mu_M(\pi^k)}\\
&\leq \sum_{k=1}^{T} V_{M_k}^{\pi^k,\mu_{M_k}(\pi^k)} - V_M^{\pi^k,\mu_M(\pi^k)} && \text{(Optimism)}\\
&\leq \sum_{k=1}^{T} V_{M_k}^{\pi^k,\mu_{M_k}(\pi^k)} - V_M^{\pi^k,\mu_{M_k}(\pi^k)} && (\text{BR means } V_M^{\pi^k,\mu_M(\pi^k)} \geq V_M^{\pi^k,\mu_{M_k}(\pi^k)})
\end{aligned}
$$

□

In the proofs that follow, we make use of the classic UCB-VI algorithm as the no-regret RL algorithm [18]. Note that although UCB-VI is applicable when the rewards are in $[0, 1]$, with translation and scaling, regret guarantees of the same order still hold when rewards are in $[-1, 1]$.

Theorem 58. *UCB-VI-FP (Algorithm 24) incurs $O(T^{1/2})$ regret under trajectory payment. This is tight as there exists a subset of Markov games, where any learning algorithm must incur $\Omega(T^{1/2})$ regret.*

Proof. At episode k, we have three MDPs:

1. The ground truth MDP $M = (P, r)$
2. An empirical MDP $\hat{M}^k = (\hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k)$ with $\beta_h^k = 2c\sqrt{\frac{SH^2}{N_h^k(s,a,b)}}$.
3. The subsidy MDP $\tilde{M}^k = (P, r + \beta^k)$

Optimism: We want to show that:

1. $\hat{M}^k$ is optimistic wrt M: for all π, μ,

$$V^{\pi,\mu}(s_0; \hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k) \geq V^{\pi,\mu}(s_0; P, r)$$

2. $\tilde{M}^k$ is optimistic wrt $\hat{M}_k$: for all π, μ,

$$V^{\pi,\mu}(s_0; P, r + \beta^k) \geq V^{\pi,\mu}(s_0; \hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k)$$

To do this, fix some (s, a, b), step h and policies π, μ. Then, we have from total-variation concentration bound for multinomial distribution:

$$\|\hat{P}^k(\cdot \mid s, a, b) - P(\cdot \mid s, a, b)\|_1 \leq c'\sqrt{\frac{|S|}{N_h^k(s, a, b)}}$$

Next, we have

$$(\hat{P}_h^k - P_h)V_{h+1}^{\pi,\mu}(s, a, b; \cdot) = \sum_{s'}(\hat{P}^k(s' \mid s, a, b) - P(s' \mid s, a, b))V_{h+1}^{\pi,\mu}(s'; \cdot).$$

Applying Hölder's inequality (with $p = 1, q = \infty$):

$$|(\hat{P}_h^k - P_h)V_{h+1}^{\pi,\mu}(s, a, b; \dot{})| \leq \|\hat{P}^k(\cdot \mid s, a, b) - P(\cdot \mid s, a, b)\|_1 \, \|V_{h+1}^{\pi,\mu}\|_\infty \leq c'\sqrt{\frac{|S|}{N_h^k(s, a, b)}} \cdot H.$$

Then by Hoeffding, this means there exists c such that:

$$|(\hat{P}_h^k - P_h)V_{h+1}^{\pi,\mu}(s, a, b; \cdot)| + |\bar{r}_h(s, a, b) - r_h(s, a, b)| \leq c\sqrt{\frac{|S|}{N_h^k(s, a, b)}} \cdot H.$$

1. **Optimism of $\hat{M}^k$:**
   Using our preceding bound, we have that for all h, s, a, b:

$$\bar{r}_h^k(s,a,b) + \frac{1}{2}\beta_h^k(s,a,b) \geq r_h(s,a,b) + |(\hat{P}_h^k - P_h)V_{h+1}^{\pi,\mu}(s,a,b;\cdot)|.$$

   We will use this in backward induction on $h \in [H]$ to show optimism:
   Base Case $h = H+1$: $V_h^{\pi,\mu}(s;\hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k) = 0 = V_h^{\pi,\mu}(s;P,r)$.
   Induction Step: suppose we have that for all state s:

$$V_{h+1}^{\pi,\mu}(s;\hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k) \geq V_{h+1}^{\pi,\mu}(s;P,r)$$

   It is sufficient to show that for every s, a, b:

$$Q_h^{\pi,\mu}(s,a,b;\hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k) \geq Q_h^{\pi,\mu}(s,a,b;P,r).$$

   and the result follows from the inequality with $\max_{a,b}$ applied on both sides.
   Using induction hypothesis:

$$\begin{aligned}
Q_h^{\pi,\mu}(s,a,b;\hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k) &= \bar{r}_h^k(s,a,b) + \frac{1}{2}\beta_h^k(s,a,b) + \sum_{s'} \hat{P}_h^k(s'|s,a,b)V_{h+1}^{\pi,\mu}(s';\hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k) \\
&\geq \bar{r}_h^k(s,a,b) + \frac{1}{2}\beta_h^k(s,a,b) + \sum_{s'} \hat{P}_h^k(s'|s,a,b)V_{h+1}^{\pi,\mu}(s';P,r) \\
&\geq r_h(s,a,b) + \sum_{s'} P_h(s'|s,a,b)V_{h+1}^{\pi,\mu}(s';P,r) \\
&= Q_h^{\pi,\mu}(s,a,b;P,r).
\end{aligned}$$

2. **Optimism of $\tilde{M}^k$ w.r.t. $\hat{M}^k$:**
   Using our preceding bound, we have that for all h, s, a, b:

$$\bar{r}_h^k(s,a,b) + \frac{1}{2}\beta_h^k(s,a,b) \geq r_h(s,a,b) + |(\hat{P}_h^k - P_h)V_{h+1}^{\pi,\mu}(s,a,b;\cdot)|.$$

   We will use this in backward induction on $h \in [H]$ to show optimism:
   Base Case $h = H+1$: $V_h^{\pi,\mu}(s;P, r + \beta^k) = 0 = V_h^{\pi,\mu}(s;\hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k)$.
   Induction Step: suppose we have that for all state s:

$$V_{h+1}^{\pi,\mu}(s;P, r + \beta^k) \geq V_{h+1}^{\pi,\mu}(s;\hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k)$$

   It is sufficient to show that for every s, a, b:

$$Q_h^{\pi,\mu}(s,a,b;P, r + \beta^k) \geq Q_h^{\pi,\mu}(s,a,b;\hat{P}^k, \bar{r}^k + \frac{1}{2}\beta^k)$$

   Using induction hypothesis:

$$\begin{aligned}
Q_h^{\pi,\mu}(s,a,b;P,r+\beta^k) &= r_h(s,a,b) + \beta_h^k(s,a,b) + \sum_{s'} P_h(s'|s,a,b) V_{h+1}^{\pi,\mu}(s';P,r+\beta^k) \\
&\geq r_h(s,a,b) + \beta_h^k(s,a,b) + \sum_{s'} P_h(s'|s,a,b) V_{h+1}^{\pi,\mu}(s';\hat{P}^k,\bar{r}^k + \frac{1}{2}\beta^k) \\
&\geq \bar{r}_h^k(s,a,b) + \frac{1}{2}\beta_h^k(s,a,b) + \sum_{s'} \hat{P}_h^k(s'|s,a,b) V_{h+1}^{\pi,\mu}(s';\hat{P}^k,\bar{r}^k + \frac{1}{2}\beta^k) \\
&= Q_h^{\pi,\mu}(s,a,b;\hat{P}^k,\bar{r}^k + \frac{1}{2}\beta^k),
\end{aligned}$$

Bounding Regret: With these, we have

$$\begin{aligned}
&\max_\pi \max_\mu V^{\pi,\mu}(s_0;P,r) \\
\leq &\max_\pi \max_\mu V^{\pi,\mu}(s_0;\hat{P}^k,\bar{r}^k + \frac{1}{2}\beta^k) && \text{(Optimism property 1)} \\
\leq &\max_\mu V^{\pi^k,\mu}(s_0;\hat{P}^k,\bar{r}^k + \frac{1}{2}\beta^k) && \text{(Definition of } \pi^k\text{)} \\
\leq &\max_\mu V^{\pi^k,\mu}(s_0;P,r+\beta^k) && \text{(Optimism property 2)} \\
\leq &V^{\pi^k,\mu^k}(s_0;P,r+\beta^k) && \text{(Follower best responding in subsidized MDP)}
\end{aligned}$$

where in the last line we use the shorthand that $\mu^k := \mu(\pi^k, r+\beta^k))$

This means that the instantaneous regret is bounded by:

$$\begin{aligned}
\text{reg}_k =& \max_\pi \max_\mu V^{\pi,\mu}(s_0;P,r) - V^{\pi^k,\mu^k}(s_0;P,r-\kappa\beta^k) \\
\leq& V^{\pi^k,\mu^k}(s_0;P,r+\beta^k) - V^{\pi^k,\mu^k}(s_0;P,r-\kappa\beta^k) \\
=& V^{\pi^k,\mu^k}(s_0;P,(1+\kappa)\beta^k) && \text{(Linearity of return)}
\end{aligned}$$

In closing, we have the cumulative regret:

$$\begin{aligned}
\sum_{k=1}^T \text{reg}_k &\leq \sum_{k=1}^T V^{\pi^k,\mu^k}(s_0;P,(1+\kappa)\beta^k) \\
&= \sum_{k=1}^T \sum_{i=1}^H (1+\kappa) b(s_i^k,a_i^k,b_i^k) + O(H\sqrt{T}) && \text{(Azuma's inequality)} \\
&\leq O(\sqrt{H^3|S|^2|A||B|T})
\end{aligned}$$

where the last step holds as for any realization of $\{s_i^k, a_i^k, b_i^k\}_{k\in[T],i\in[H]}$, the sum is upper bounded by $O(\sqrt{H^2|S|} \cdot \sqrt{H|S||A||B|T})$.

□

Theorem 59. *There exists an algorithm, leveraging UCB-VI-FP as subroutine, that incurs $O(T^{2/3})$ regret under upfront payment.*

Proof. By Lemma 73, the total regret incurred by a learning algorithm under upfront payment is:

$$\mathcal{R}(T) = \sum_{i=1}^{T} V^{\pi^*,\mu^*} + 0 - (\sum_{i=1}^{T} \mathbb{E}[V^{\pi_i,\mu(\pi_i)}] - \kappa \cdot \sum_{s,a,b \in S \times A \times B} b^i(s,a,b)]))$$

We can establish the $O(T^{2/3})$ with the following algorithm. First, run UCB-VI-FP for $m = T^{2/3}$ iterations. Then, using online to batch conversion, repeat this sequence of policies $\pi_1, ..., \pi_m$ for the remaining $T - m$ steps, with all payments set to zero.

As shown in Theorem 52, the policy regret from the explore phase is $\sum_{i=1}^{m} V^{\pi^*,\mu^*} - \mathbb{E}[V^{\pi_i,\mu(\pi_i)}] \leq O(T^{1/3})$. The payment from the explore phase is $\sum_{i=1}^{m} \kappa \sum_{s,a,b \in S \times A \times B} b^i(s,a,b) \leq m \cdot O(1) = O(T^{2/3})$. The policy regret from the exploit phase is $\sum_{i=m+1}^{T} V^{\pi^*,\mu^*} - \mathbb{E}[V^{\pi_i,\mu(\pi_i)}] \leq O((T - T^{2/3})\frac{\sqrt{T^{2/3}}}{m}) = O(T^{2/3})$. And so, the total regret is $\mathcal{R}(T) = O(T^{2/3})$. □

9.10.2 Contrasting Trajectory Payment with Upfront Payment

Proposition 51. *UCB-VI-FP with indicator bonus incurs constant $O(|S||A||B|)$ regret under trajectory payment.*

Proof. Using Lemma 74, we can reduce Stackelberg learning with payment to single-agent no-regret learning with joint policy class $\pi^{joint} : S \to A \times B$.

For the no-regret learning algorithm, we will again use the classical UCB-VI algorithm [18]. To obtain a tighter bound of the algorithm's expected regret, we construct an optimistic MDP M_k with reward $\hat{r}_h^k(s,(a,b)) := 1$ if (s,a,b) unvisited before step $k-1$ at step h and $r(s,a,b)$ o.w. Since $r \in [-1,1]$, $\hat{r}_h^k(s,a,b) \geq r(s,a,b)$ for all h, k.

Following the UCB-VI proof outline:

$$\begin{aligned}
&\mathbb{E}[V_h^k(s_h^k) - V_h^{\pi_k}(s_h^k)] \\
&= \mathbb{E}[(Q_h^k - Q_h^{\pi_k})(s_h^k,(a_h^k,b_h^k))] \\
&= \mathbb{E}[P_h V_{h+1}^k(s_h^k,(a_h^k,b_h^k)) - P_h V_{h+1}^{\pi_k}(s_h^k,(a_h^k,b_h^k)) + (\hat{r}_h^k - r_h^k)(s_h^k,(a_h^k,b_h^k))] \\
&= \mathbb{E}[V_{h+1}^k(s_{h+1}^k) - V_{h+1}^{\pi_k}(s_{h+1}^k) + b_h^k] \qquad (\text{let } b_h^k = \mathbb{E}[(\hat{r}_h^k - r_h^k)(s_h^k,a_h^k,b_h^k)])
\end{aligned}$$

Unrolling, we have that:

$$\sum_{k=1}^{T} \mathbb{E}[(V_1^* - V_1^{\pi_k})(s_1)] \leq \sum_{k=1}^{T} \mathbb{E}[(V_1^k - V_1^{\pi_k})(s_1)] \leq \mathbb{E}[\sum_{k=1}^{T}\sum_{h=1}^{H} b_h^k]$$

To finish, we observe that for every roll-out, we have that:

$$\sum_{k=1}^{T}\sum_{h=1}^{H} b_h^k = \sum_{s,a,b \in S \times A \times B} \mathbb{1}\left\{N^T(s,a,b) \geq 1\right\}(1 - r(s,a,b)) \geq 2|S||A||B|$$

□

Proposition 52. *There exists an algorithm, leveraging UCB-VI-FP with indicator bonus as subroutine, that incurs $O(T^{1/2})$ regret under upfront payment.*

Proof. We can establish the $O(T^{1/2})$ bound with the following algorithm. First, run UCB-VI-FP with indicator bonus for $m = T^{1/2}$ iterations. Then, using online to batch conversion, repeat this sequence of policies $\pi_1, ..., \pi_m$ for the remaining $T - m$ steps, with all payments set to zero.

As shown in Proposition 45, the policy regret from the explore phase is $\sum_{i=1}^{m} V^{\pi^*,\mu^*} - \mathbb{E}[V^{\pi_i,\mu(\pi_i)}] \leq O(1)$. The payment from the explore phase is $\sum_{i=1}^{m} \kappa \sum_{s,a,b \in S\times A\times B} b^i(s,a,b) \leq m \cdot O(1) = O(T^{1/2})$. The policy regret from the exploit phase is $\sum_{i=m+1}^{T} V^{\pi^*,\mu^*} - \mathbb{E}[V^{\pi_i,\mu(\pi_i)}] \leq O((T - T^{1/2})\frac{1}{m}) = O(T^{1/2})$. And so, the total regret is $\mathcal{R}(T) = O(T^{1/2})$. □

Proposition 53. *There exists a subset of Markov Game instances such that any learning algorithm has to incur $\Omega(T^{1/2})$ regret under upfront payment.*

Proof. **Setup:** We will use Yao's Lemma to show our result. Define cost of algorithm A at the ith episode in instance X as $cost(A,i,x) = \mathbb{E}[\mathbb{1}\{\pi_i \neq \pi^*\}(r(\pi^*) - r(\pi_i)) + S_i]$. Let $\mathcal{D}$ be the uniform distribution over all MDP instances $\mathcal{X}$, where the distribution is uniform over which of the two actions at each time step achieves reward 1 (and the other 0). Our aim is to show $\min_A \mathbb{E}_{x\sim\mathcal{D}}[\sum_{i=1}^{T} cost(A,i,x)] \geq \Omega(\sqrt{T})$. This implies that any randomized algorithms R has $\min_R \max_{x\in\mathcal{X}} \mathbb{E}[\sum_{i=1}^{T} cost(R,i,x)] \geq \Omega(\sqrt{T})$.

Notation: Let the history up until time t be H_t:

$$H_t = (\pi_1, S_1, \tau_1, A_1, W_1', \pi_2, S_2, \tau_2, \ldots, \pi_t, S_t, \tau_t, A_t, W_t')$$

where for example $\tau_1 = (s_1^1, a_1^1, b_1^1, r(s_1^1, a_1^1, b_1^1), \ldots, s_H^1, a_H^1, b_H^1, r(s_H^1, a_H^1, b_H^1))$.

- π_i is the right branch policy at time i
- τ_i is the trajectory in the Markov game at time i.
- S_i is the payment on the right branch at round i, and S_i' the payment on the left branch.
- Define right branch subsidy being at least $\frac{1}{2}$ as $X_i = \mathbb{1}(S_i \geq \frac{1}{2})$.
- $Y_i = \mathbb{1}\{\pi_i = \pi^*\}$.
- A_i denotes if the follower's BR is the right action.

$$\begin{aligned}
&A_i = 1 \\
&\Leftrightarrow r^F(s_0, a_2) = (1-\epsilon)(H - 1/2 + S_i') + \epsilon(S_i + r(\pi_i)) \geq H - 1/2 + S_i' = r^F(s_0, a_1) \\
&\Leftrightarrow \epsilon(S_i + r(\pi_i)) \geq \epsilon(H - 1/2 + S_i')
\end{aligned}$$

And so,

$$A_i = \mathbb{1}(S_i + r(\pi_i) \geq S_i' + H - 1/2)$$

- Let Z_i denote Bernoulli(ϵ) random variable corresponding to the stochastic transition, when the follower chooses action a_2.
  $Z_i = 1$ if the follower transitions to the right branch provided $A_i = 1$.

- Let W'_i denotes if the follower goes into the right branch. We have that:

$$W'_i = A_i \wedge Z_i$$

We are interested in the event $W_t = Y_t \vee X_t Z_t$. It has the useful property that $W_t = 0 \Rightarrow Y_t = 0, X_t Z_t = 0$. Moreover, we claim that $W_t = 0 \Rightarrow W'_t = 0$.

$$\begin{aligned}
& Y_t = 0, X_t Z_t = 0 \\
& \Rightarrow r(\pi_t) \leq H - 1, X_t Z_t = 0 \\
& \Leftrightarrow H - 1/2 - r(\pi_t) \geq 1/2, X_t Z_t = 0 \\
& \Rightarrow X_t \geq A_t, X_t Z_t = 0 \qquad (A_t = 1 \Rightarrow S_t \geq S'_t + H - 1/2 - r(\pi_t) \geq S'_t + 1/2 \Rightarrow X_t = 1) \\
& \Rightarrow A_t Z_t = 0 \qquad (0 = X_t Z_t \geq A_t Z_t) \\
& \Rightarrow W'_t = 0
\end{aligned}$$

We will use this property in the lemma that follows.

Lemma 75.

$$P(Y_t = 0 | W_{:t-1} = 0) \geq 1 - \frac{t-1}{2^H}$$

Proof.

$$\begin{aligned}
& \sum_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0 | H_{t-1}) P(H_{t-1} | W_{:t-1} = 0) \\
&= \sum_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0 | H_{t-1}, \pi_t = \pi') P(H_{t-1} | W_{:t-1} = 0) \\
& \qquad (\pi_t \text{ is a deterministic function of } H_{t-1}) \\
&= \sum_{H_{t-1}: W_{:t-1}=0} P(\pi^* \neq \pi' | H_{t-1}, \pi_t = \pi') P(H_{t-1} | W_{:t-1} = 0) \\
&= \sum_{H_{t-1}: W_{:t-1}=0} P(\pi^* \neq \pi' | Y_{:t-1} = 0, \pi_{:t-1}) P(H_{t-1} | W_{:t-1} = 0) \qquad (\dagger) \\
&\geq 1 - \frac{t-1}{2^H} \qquad (\text{posterior of } \pi^* \text{ is uniform over } \Pi \setminus \pi_{:t-1})
\end{aligned}$$

$(\dagger)$: First, we have that the observed trajectory has the specific functional form: $\tau_i = \tau^{\pi_i} \mathbb{1}\{W'_i = 1\} + \tau_{left}\{W'_i = 0\}$. Thus, conditioned on $W_{:t-1} = 0 \Rightarrow W'_{:t-1} = 0$, we have that $\tau_i = \tau_{left}$ for all $i \in [t-1]$. That is, $\pi^* \perp\!\!\!\perp \tau_i$ in the *conditional* joint distribution.

From this, we have that $\pi^* \perp\!\!\!\perp H_{t-1} \setminus \{Y_{:t-1} = 0, \pi_{:t-1}\} \mid \{Y_{:t-1} = 0, \pi_{:t-1}\}$ by checking D-separation of $\{S_i, A_i, \tau_i, W'_i\}$ with π^*. π^*'s only children are Y_i's, thus conditioning on Y_i's other parents (π_i's) and Y_i's themselves blocks every path to the rest of the random variables.

□

Lemma 76. *For every $t \leq T$:*

$$P(W_{:t} = 0) \geq \frac{1}{2}(1 - \sum_{j=1}^{t} \epsilon P(X_j = 1 | W_{:j-1} = 0))$$

Proof. $W_{:t} = 0 \Leftrightarrow Y_{:t} = 0, X_{:t}Z_{:t} = 0$. Towards bounding the product, we have:

$$\begin{aligned}
&P(Y_t = 0, X_tZ_t = 0 | W_{:t-1} = 0) \\
&= \sum_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0, X_tZ_t = 0 | H_{t-1}) P(H_{t-1} | W_{:t-1} = 0) \\
&\qquad\qquad \text{(where } H_{t-1} \text{ is the history up until } t-1, W_{:t-1} \in H_{t-1}) \\
&= \sum_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0 | H_{t-1}) P(X_tZ_t = 0 | H_{t-1}) P(H_{t-1} | W_{:t-1} = 0) \\
&\qquad\qquad (X_t \text{ deterministic function of } H_{t-1}, Z_t \text{ is independent of } Y_t \Rightarrow Y_t \perp\!\!\!\perp X_tZ_t | H_{t-1}) \\
&\geq \min_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0 | H_{t-1}) \sum_{H_{t-1}: W_{:t-1}=0} (P(X_tZ_t = 0 | H_{t-1}, Z_t = 0) P(Z_t = 0 | H_{t-1}) \\
&\qquad + P(X_t = 0 | H_{t-1}, Z_t = 1) P(Z_t = 1 | H_{t-1})) P(H_{t-1} | W_{:t-1} = 0) \qquad \text{(condition on } Z_t) \\
&= \min_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0 | H_{t-1}) \sum_{H_{t-1}: W_{:t-1}=0} (1 - \epsilon + P(X_t = 0 | H_{t-1}, Z_t = 1)\epsilon) P(H_{t-1} | W_{:t-1} = 0) \\
&= \min_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0 | H_{t-1})(1 - \epsilon + \epsilon \sum_{H_{t-1}: W_{:t-1}=0} P(X_t = 0 | H_{t-1}) P(H_{t-1} | W_{:t-1} = 0) \\
&\qquad\qquad (X_t \text{ is deterministic function of } H_{t-1}, \text{ so independent of } Z_t) \\
&= \min_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0 | H_{t-1})(1 - \epsilon + \epsilon \sum_{H_{t-1}: W_{:t-1}=0} P(X_t = 0 | H_{t-1}, W_{:t-1} = 0) P(H_{t-1} | W_{:t-1} = 0) \\
&= \min_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0 | H_{t-1})(1 - \epsilon + \epsilon P(X_t = 0 | W_{:t-1} = 0)) \\
&= \min_{H_{t-1}: W_{:t-1}=0} P(Y_t = 0 | H_{t-1})(1 - \epsilon P(X_t = 1 | W_{:t-1} = 0))
\end{aligned}$$

From before, for any history H_{t-1} where $W_{:t-1} = 0$:

$$\begin{aligned}
&P(Y_t = 0 | H_{t-1}, W_{:t-1} = 0) \\
&= P(\pi^* \neq \pi' | Y_{:t-1} = 0, \pi_{:t-1}) \qquad \text{(from prior lemma)} \\
&\geq 1 - \frac{t-1}{2^H}
\end{aligned}$$

Putting it together,

$$P(Y_t = 0, X_tZ_t = 0 | W_{:t-1} = 0) \geq (1 - \frac{t-1}{2^H})(1 - \epsilon P(X_t = 1 | W_{:t-1} = 0))$$

and so,

$$\begin{aligned}&P(Y_{:t}=0, X_{:t}Z_{:t}=0)\\&\geq \frac{1}{2}\prod_{i=1}^{t}(1-\epsilon P(X_i=1|W_{:i-1}=0)) && \text{(using that } 2^H >> T\text{)}\\&\geq \frac{1}{2}(1-\sum_{j=1}^{t}\epsilon P(X_j=1|W_{:j-1}=0))\end{aligned}$$

□

Note that the optimal expected return is: $(1-\epsilon)(H-1/2)+\epsilon H$. When the follower chooses a_L, the instantaneous regret is: $(1-\epsilon)(H-1/2)+\epsilon H-(H-1/2)=\epsilon/2$. If the follower chooses a_R and $\pi_t \neq \pi^*$, then the instantaneous regret is at least ϵ. Overall, the instantaneous regret is at least $\epsilon/2$ when $\pi_t \neq \pi^*$. And so, the cumulative regret bound is lower bounded by:

$$\begin{aligned}&\sum_{t=1}^{T}P(\pi_t \neq \pi^*)\epsilon/2+\mathbb{E}[\sum_{t=1}^{T}S_t+S'_t]\\&\geq \sum_{t=1}^{T}P(\pi_t \neq \pi^*)\epsilon/2+1/2\mathbb{E}[\sum_{t=1}^{T}X_t]\\&\geq \epsilon/2\sum_{t=1}^{T}P(\pi_t \neq \pi^*|W_{:t-1}=0)P(W_{:t-1}=0)+1/2\mathbb{E}[\sum_{t=1}^{T}X_t]\\&\geq \epsilon/2\sum_{t=1}^{T}(1-\frac{t-1}{2^H})P(W_{:t-1}=0)+1/2\mathbb{E}[\sum_{t=1}^{T}X_t]\\&\geq \frac{\epsilon}{4}\sum_{t=1}^{T}P(W_{:t-1}=0)+1/2\mathbb{E}[\sum_{i=1}^{T}X_i]\end{aligned}$$

We consider two cases:

1. Case 1: $\mathbb{E}[\sum_{i=1}^{T}X_i]=\sum_{i=1}^{T}P(X_i=1)\geq 1/16\epsilon$
   In this case, the regret is at least $1/32\epsilon$.
2. Case 2: $\sum_{i=1}^{T}P(X_i=1)<1/16\epsilon$
   Then, we claim that $\sum_{j=1}^{T}P(X_j=1|W_{:j-1}=0)\leq 1/4\epsilon$.
   Suppose not and $\sum_{j=1}^{T}P(X_j=1|W_{:j-1}=0)>1/4\epsilon$.
   Then, there must exist $t<T$ such that $\sum_{j=1}^{t}P(X_j=1|W_{:j-1}=0)\in[1/4\epsilon-1, 1/4\epsilon]$.
   This implies:

$$
\begin{aligned}
&\sum_{i=1}^{T} P(X_i = 1) \\
&\geq \sum_{i=1}^{t} P(X_i = 1|W_{:i-1} = 0)P(W_{:i-1} = 0) \\
&\geq \sum_{i=1}^{t} P(X_i = 1|W_{:i-1} = 0)\frac{1}{2}(1 - \epsilon\sum_{j=1}^{i-1} P(X_j = 1|W_{:j-1} = 0)) \\
&\geq \sum_{i=1}^{t} P(X_i = 1|W_{:i-1} = 0)\frac{1}{2}(1 - \epsilon \cdot 1/4\epsilon) \\
&= \frac{3}{8}\sum_{i=1}^{t} P(X_i = 1|W_{:i-1} = 0) \\
&\geq \frac{3}{8}(1/4\epsilon - 1) \\
&> 1/16\epsilon
\end{aligned}
$$

which is a contradiction.
Now, because $\sum_{j=1}^{T} P(X_j = 1|W_{:j-1} = 0) \leq 1/4\epsilon$, the cumulative regret from before is:

$$
\begin{aligned}
&\geq \frac{\epsilon}{4}\sum_{t=1}^{T} P(W_{:t-1} = 0) + \mathbb{E}[\sum_{i=1}^{T} X_i] \\
&\geq \frac{\epsilon}{4}TP(W_{:T-1} = 0) \\
&\geq \frac{\epsilon}{4}T\frac{1}{2}(1 - \epsilon\sum_{j=1}^{T-1} P(X_j = 1|W_{:j-1} = 0)) \\
&\geq \frac{\epsilon}{4}T\frac{1}{2}(1 - \epsilon\frac{1}{4\epsilon}) \\
&\geq \frac{3T\epsilon}{32}
\end{aligned}
$$

In conclusion, the cumulative regret is at least $\min(\frac{1}{32\epsilon}, \frac{3T\epsilon}{32}) = \Omega(T^{1/2})$ when we let $\epsilon = T^{-1/2}$.

□

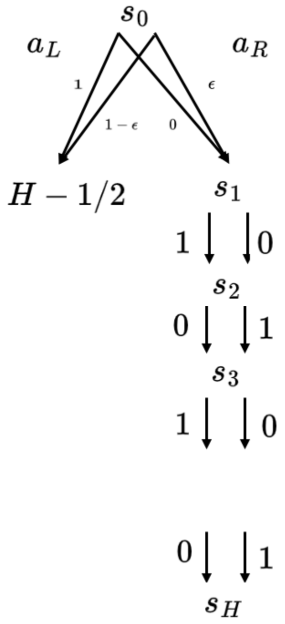


Figure 9.4: Right branch requires getting the right set of H binary actions at states $s_1, ..., s_H$ to exceed the return in the left branch.

9.11 Experiments

Setup: We consider a turn-based Markov Game, where the leader is solving a RL problem and the follower solves a bandit problem in its BR [26]. This class of Markov Games includes the hard instance construction in Section 5. Our goal is to examine whether learning without payment can get stuck even in more "average" (and not worst) case Markov Games.

For experimentation, the leader is learning in a toy MDP with $H = 5$. For the follower, arm a_1 leads to a MDP, whose optimal return is the optimal return in the (turn-based) Markov game. On the other hand, arm a_2 has a deterministic high reward that is α that of the optimal return.

By varying α, we can make the follower get "stuck" in myopically choosing a_2, thus preventing the leader from exploring and learning the actual optimal policy. This is the intuition behind the negative results, Theorem 49 and Theorem 50, where we set α very high.

For the baseline, we use the single-agent learning algorithm UCB-VI in the without payment case, and compare it against UCB-VI-FP in the with payment case. We track the cumulative regret of the two learning algorithms over 40000 episodes and across 20 runs.

Finding: We experiment with different α's, finding that learning without payment can get stuck in the myopic optimum even when α is as low as 0.5. Interestingly, this suggests that there are "non-worst-case" Markov games, where exploration can be difficult without payment. Under $\alpha = 0.5$, Figure 9.5 shows that:

1. In absence of payment, even if the leader is using a (one-sided) no-regret algorithm (UCB-VI), the leader may not be able to explore adequately and incur linear regret.
2. UCB-VI-FP attains sublinear regret, showing the importance of payment in incentivizing

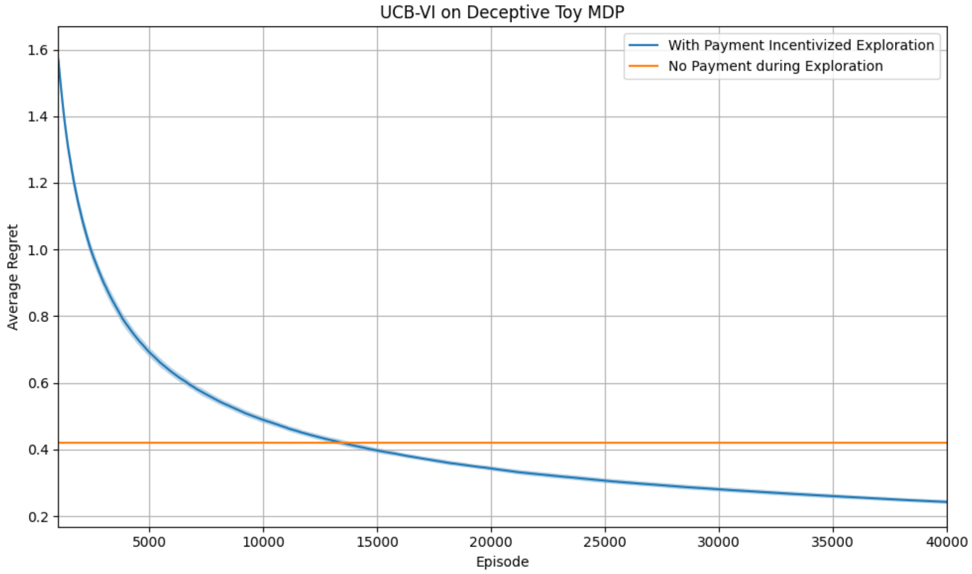


Figure 9.5: Regret plot from episodes 1000 to 40000

exploration needed to learn the optimal policy.

3. UCB-VI-FP initially incurs a higher regret, which we expect due to the additional payment used by the algorithm to incentivize exploration. Over time, its regret improves due to exploration shrinking the policy regret, and reduced incentivization (and thus payment regret). Eventually, UCB-VI-FP's regret dips below that of UCB-VI, with a crossover point at around episode 13500.

9.12 Additional Related Works

Thematically, our paper belongs to the intersection of literature on Stackelberg policy computation in Markov games and literature on contracting through reward shaping in MDPs.

Other Variants of Follower best response in Stackelberg Games: In our paper, we consider the standard assumption of the follower best responding as well as its generalization, the $\lambda-$entropy regularized best response model. Moving further away from this canonical formulation in Stackelberg games, there have also been other formulations of follower behavior. Chen et al. [64], Zhong et al. [324] study learning the optimal policy in face of a myopic follower that greedily best responds. Furthermore, there is also a growing line of work on learning in face of an agent (follower) who is also learning. For example, Guruganesh et al. [128] studies how to contract an agent that is learning. Kao et al. [161] studies learning in face of a follower that is also learning in cooperative games.

Broader Stackelberg games literature: Our paper focuses specifically on Stackelberg Markov games, and adds to the body of work that builds a closer connection with RL theory. Zooming back out, the broader Stackelberg game literature is vast and varied. For example, there is an extensive body of work studying Stackelberg games in normal-form games (horizon-one games), often inspired by security games, as well as empirical methods for Stackelberg games. We mention [32] and [112] as an example of each line of work, in which one may find further relevant references.

Learning the Optimal Payment Scheme in MDPs: It is natural to ask if there are any implications from prior papers on contracting in MDPs[41, 48, 152, 297]. Do papers in the single-agent setting have any implications for the more general two-player Stackelberg Markov game setup we consider?

In the full information setting that we consider, we have already covered the difference between the Markov game setting we consider and the bandit setting studied in [248]. Another paper that studies the full information setting is that of [41], which proves that planning is NP-hard, albeit under a different formulation where the leader aims to maximize the return subject to the payment being capped by some budget. By contrast, similar to previous works by Scheid et al. [248], Wu et al. [297], our paper studies maximizing the return minus the total payment. Hence, it is not immediately clear how the results carry over to our setting.

Besides this, other papers [48, 152, 297] focus on the imperfect information settings, where the payment cannot be a function of the follower's action (hidden at the time of payment). And so, in this case, it is also not clear how results transfer due to differing setups.

As our paper is the first to study Stackelberg Markov games with payment where the leader can set both the policy and the payment, the generality of our setup means that new results arise. For instance, we show hardness results in Cooperative Markov games that do not exist under existing contracting in MDP settings [41, 48, 152, 297]. This finding motivates us to study how to learn efficiently with payment in cooperative games, and we develop no-regret algorithms to this end.

UCB-VI-FP is a notable multi-agent algorithm as the algorithm can only control one player when exploring. Indeed, our results show that even if the leader is using a no-regret learning algorithm, then learning can still be inefficient. And so, a new learning algorithm needs to be developed here, using payment to incentivize *collective* exploration. Finally, we add that we also obtain results under upfront payment, which is a new form of payment that has not been considered in previous contracting in MDP literature.

Bi-level Optimization: While the primary goal of our paper is to study global optimum, we note that bi-level optimizers can tractably compute *local optimum* in planning under known rewards and dynamics [90, 211, 259, 275]. To handle the learning setting with unknown rewards and dynamics, we develop a new algorithm (UCB-VI-FP) for adaptive exploration while minimizing regret.

9.13 Incentive Effects when Follower Reward is Unobservable

Incentive Effects: A key underlying assumption in our setup is that the leader can readily observe the follower's reward and/or trust that the follower has reported their true reward. Truthfulness is important in the partnership, but suppose we allow the follower to misreport all rewards up to Δ, what may happen then? We have the following result in the direct-payment case studied by Scheid et al. [248].

Proposition 54. *Suppose the follower can misreport r^F up to Δ, $\|r'^F - r^F\|_1 \leq \Delta$. In the bandit setting, the follower's return can change by at most:*

$$|V^{\pi^*(r^F),\mu(\pi^*(r^F))}(s_0; r^F + b^*(r^F)) - V^{\pi^*(r'^F),\mu(\pi^*(r'^F))}(s_0; r^F + b^*(r'^F))| \leq 2\Delta$$

and the leader's return can change by at most:

$$|V^{\pi^*(r^F),\mu(\pi^*(r^F))}(s_0; r^L - b^*(r^F)) - V^{\pi^*(r'^F),\mu(\pi^*(r'^F))}(s_0; r^L - b^*(r'^F))| \leq 2\Delta$$

Proof. We show that in the bandit setting, the follower's return differs by at most 2Δ, as does the leader's return.

The bandit Stackelberg setting is such that the leader optimizes:

$$\begin{aligned} \max_i \quad & r_i^L - b_i \\ \text{s.t.} \quad & r_i^F + b_i \geq \max_{j \neq i} r_j^F \end{aligned}$$

The follower may instead report r'^F s.t. $\|r'^F - r^F\|_1 \leq \Delta$. Let i^* be the optimal arm under r^F and arm i' under r'^F.

We observe that in all bandit games with reward r^F, the follower's return is $\max_j r_j^F$. If $i^* = \operatorname{argmax}_j r_j^F$, then it's clear that $b_{i^*} = 0$ as lowering it to zero preserves the follower choosing arm i^*, while increasing the leader's return. In the other case, $r_i^F + b_i \geq \max_j r_j^F$. If this is not tight, then we can lower b_{i^*} s.t. it is tight and preserve the follower choosing arm i^*, while increasing the leader's return.

With this, the return of the follower under truthful reporting is $\max_j r_j^F$. Under r'^F reporting, it's $r_{i'}^F + \max_j r_j'^F - r_{i'}'^F$ (note that it gets the true reward $r_{i'}^F$). The difference is thus:

$$|r_{i'}^F + \max_j r_j'^F - r_{i'}'^F - \max_j r_j^F| \leq |r_{i'}^F - r_{i'}'^F| + |\max_j r_j'^F - \max_j r_j^F| \leq 2\Delta$$

since

$$\max_j r_j'^F \geq r_k' \geq r_k - \Delta = \max_j r_j^F - \Delta$$

and

$$\max_j r_j^F = r_k \geq r_l \geq r_l' - \Delta = \max_j r_j'^F - \Delta$$

Moreover, the leader's return also differs by at most 2Δ:

$$r_{i^*}^L - (\max_j r_j^F - r_{i^*}^F) \geq r_{i'}^L - (\max_j r_j^F - r_{i'}^F) \geq r_{i'}^L - (\max_j r_j'^F - r_{i'}'^F) - 2\Delta$$

and

$$r_{i'}^L - (\max_j r_j'^F - r_{i'}'^F) \geq r_{i^*}^L - (\max_j r_j'^F - r_{i^*}'^F) \geq r_{i^*}^L - (\max_j r_j^F - r_{i^*}^F) - 2\Delta$$

□

Chapter 10

Multi-agent Policy Aggregation via IRL

10.1 Introduction

Another common problem in multi-agent systems and alignment is reconciling the differing policies of agents. That is, we may be interested in inverse reinforcement learning (IRL) from multiple agents (in place of a single agent) [2, 212]. Specifically, suppose we observe n different agents executing policies that are optimal for their individual reward functions. Our goal is to sensibly aggregate these trajectories from these policies into a single policy, through the use of inverse reinforcement learning.

However, if individual agents have wildly divergent reward functions, then the aggregate policy may not represent coherent behavior. In addition, to formally reason about the quality of the optimal policy, we need to relate it to some notion of ground truth. For these reasons, we consider a more specific setting where the agents are *like-minded*, in that individual reward functions are nothing but noisy versions of an underlying reward function. How well would IRL algorithms then fare?

In sum, our research challenge is this:

> *Given observations from policies that are optimal with respect to different reward functions, each of which is a perturbation of an underlying reward function, identify IRL algorithms that can recover a good policy with respect to the underlying reward function.*

We believe that this problem is both natural and general. To further motivate it, let us briefly instantiate it in the context of value alignment in AI safety. One of the prominent approaches in this area is to align the values of the AI system with the values of a human through IRL [129, 244]. Our extension to multiple agents would allow the alignment of the system with the values of *society*.

A compelling aspect of this instantiation is that, if we think of the underlying reward function as embodying a common set of moral propositions, then our technical assumption of like-minded agents can be justified through the *linguistic analogy*, originally introduced by Rawls [237]. It draws on the work of Chomsky [69], who argued that competent speakers have a set of grammatical principles in mind, but their linguistic behavior is hampered by "grammatically irrelevant conditions such as memory limitations, distractions, shifts of attention and interest, and errors.” Analogously, Rawls claimed, humans have moral rules — a common “moral grammar” — in our minds, but, due to various limitations, our moral behavior is only an approximation thereof. Interestingly, this theory lends itself to empirical experimentation, and, indeed, it has been validated through work in moral psychology [203].

Our Model and Results. We start from a common IRL setup: each reward function is associated with a weight vector $\mathbf{w}$, such that the reward for taking a given action in a given state is the dot product of the weight vector and the feature vector of that state-action pair. The twist is that there is an underlying reward function represented by a weight vector $\mathbf{w}^\star$, and each of the agents is associated with a weight vector $\mathbf{w}_i$, which induces an optimal policy π_i. We observe a trajectory from each π_i.

In Section 10.3, we focus on competing with a uniform mixture over the optimal policies of the agents, $\pi_1, \ldots, \pi_n$ (for reasons that we explicate momentarily). We can do this because the observed trajectories are “similar” to the uniform mixture, in the sense that their feature vectors — the discounted frequencies of the features associated with the observed state-action pairs — are close to that of the uniform mixture policy. Therefore, due to the linearity of the reward function, any policy whose feature expectations approximately match those of the observed trajectories must be close to the uniform mixture with respect to $\mathbf{w}^\star$. We formalize this idea in Theorem 60, which gives a lower bound on the number of agents and length of observed trajectories such that any policy that $\epsilon/3$-matches feature expectations is ϵ-close to the uniform mixture. Furthermore, we identify two well-known IRL algorithms, Apprenticeship Learning [2] and Max Entropy [329], which indeed output policies that match the feature expectations of the observed trajectories, and therefore enjoy the guarantees provided by this theorem.

Needless to say, competing with the uniform mixture is only useful insofar as this benchmark exhibits “good” performance. We show that this is indeed the case in Section 10.4, assuming (as stated earlier) that each weight vector $\mathbf{w}_i$ is a noisy perturbation of $\mathbf{w}^\star$. Specifically, we first establish that, under relatively weak assumptions on the noise, it is possible to bound the difference between the reward of the uniform mixture and that of the optimal policy (Theorem 61). More surprisingly, Theorem 62 asserts that in the worst case it is impossible to outperform the uniform mixture, by constructing an MDP where the optimal policy cannot be identified — even if we had an infinite number of agents and infinitely long trajectories! Putting all of these results together, we conclude that directly running an IRL algorithm that matches feature expectations on the observed trajectories is a sensible approach to our problem.

Nevertheless, it is natural to ask whether it is possible to outperform the uniform mixture in typical instances. In Section 10.5 we show that this is indeed the case; in fact, we are able to recover the optimal policy whenever it is identifiable, albeit under stringent assumptions — most importantly, that the MDP has only one state. This leads to a challenge that we call the *inverse multi-armed bandit problem*. To the best of our knowledge, this problem is novel; its study contributes to the (relatively limited) understanding of scenarios where it is possible to outperform teacher demonstrations.

Related work. The most closely related work deals with IRL when the observations come from an agent who acts according to multiple *intentions*, each associated with a different reward function [19, 68]. The main challenge stems from the need to cluster the observations — the observations in each cluster are treated as originating from the same policy (or intention). By contrast, clustering is a nonissue in our framework. Moreover, our assumption that each $\mathbf{w}_i$ is a noisy perturbation of $\mathbf{w}^\star$ allows us to provide theoretical guarantees.

Further afield, there is a body of work on robust RL and IRL under reward uncertainty [119, 239, 240], noisy rewards [323], and corrupted rewards [101]. Of these papers the closest to ours is that of Zheng et al. [323], who design robust IRL algorithms under *sparse* noise, in the sense that only a small fraction of the observations are anomalous; they do not provide theoretical guarantees. Our setting is quite different, as very few observations would typically be associated with a near-perfect policy.

10.2 MDP Terminology

We assume the environment is modeled as an MDP $\{S, A, T, \gamma, D\}$ with an unknown reward function. S is a finite set of states; A is a finite set of actions; $T(s, a, s')$ is the state transition probability of reaching state s' from state s when action a is taken; $\gamma \in [0, 1)$ is the discount factor; and D the initial-state distribution, from which the start state s_0 is drawn for every trajectory.

As is standard in the literature [2], we assume that there is a function $\phi : S \times A \to \mathbb{R}^d$ that maps state-action pairs to their real-valued features. We also overload notation, and say that the feature vector of a trajectory $\tau = \{(s_0, a_0), (s_1, a_1), \dots, (s_L, a_L)\}$ is defined as $\phi(\tau) = \sum_{t=0}^{L} \gamma^t \phi(s_t, a_t)$.

We make the standard assumption that the immediate reward of executing action a from state s is linear in the features of the state-action pair, i.e. $r^{\mathbf{w}}(s, a) = \mathbf{w}^\intercal \phi(s, a)$. This has a natural interpretation: ϕ represents the different factors, and $\mathbf{w}$ weighs them in varying degrees. Note that we assume that ϕ is a sufficiently rich feature extractor that can capture complex state-action-reward behavior; for instance, if one is to view $r^w(s, a)$ as a neural network, ϕ corresponds to all but the last layer of the network.

Let μ denote the feature expectation of policy π, that is, $\mu(\pi) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \phi(s_t, a_t) | \pi]$, where π defines the action a_t taken from state s_t, and the expectation is taken over the transition probabilities $T(s_t, a_t, s_{t+1})$. Therefore, since $R^{\mathbf{w}}(\pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^{\mathbf{w}}(s_t, a_t) \middle| \pi\right]$, the cumulative reward of a policy π under weight $\mathbf{w}$ can be rewritten as:

$$R^{\mathbf{w}}(\pi) = \mathbf{w}^\intercal \cdot \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \phi(s_t, a) \middle| \pi\right] = \mathbf{w}^\intercal \mu(\pi)$$

Let $P_\pi(s, t)$ denote the probability of getting to state s at time t under policy π. Then, the cumulative reward $R^{\mathbf{w}}$ is

$$R^{\mathbf{w}}(\pi) = \sum_{t=0}^{\infty} \gamma^t \sum_{s \in S} P_\pi(s, t) r^{\mathbf{w}}(s, \pi(s)).$$

10.3 Approximating the Uniform Mixture

We consider an environment with n agents $N = \{1, \dots, n\}$. Furthermore, the reward function of each agent $i \in N$ is associated with a weight vector $\mathbf{w}_i$, and, therefore, with a reward function $r^{\mathbf{w}_i}$.

This determines the optimal policy π_i executed by agent i, from which we observe the trajectory τ_i, which consists of L steps. We observe such a trajectory for each $i \in N$, giving us trajectories $\{\tau_1, ..., \tau_n\}$.

As we discussed in Section 10.1, we assume that the reward function associated with each agent is a noisy version of an underlying reward function. Specifically, we assume that there exists a ground truth weight vector $\mathbf{w}^\star$, and for each agent $i \in N$ we let $\mathbf{w}_i = \mathbf{w}^\star + \boldsymbol{\eta}_i$, where $\boldsymbol{\eta}_i$ is the corresponding noise vector; we assume throughout that $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_n$ are i.i.d. Following Abbeel and Ng [2], we also assume in some of our results (when stated explicitly) that $\|\mathbf{w}^\star\|_2 \leq 1$ and $\|\phi(s, a)\|_\infty \leq 1$.

Let us denote by π^u the *uniform mixture* over the policies $\pi_1, \ldots, \pi_n$, that is, the (randomized) policy that, in each trajectory, selects one of these policies uniformly at random and executes it throughout the trajectory.

Our goal in this section is to "approximate" the uniform mixture (and we will justify this choice in subsequent sections). To do so, we focus on IRL algorithms that "match feature expectations." Informally, the property of interest is that the feature expectations of the policy match the (discounted) feature vectors of observed trajectories. This idea is already present in the IRL literature, but it is helpful to define it formally, as it allows us to identify specific IRL algorithms that work well in our setting.

Definition 44. *Given n trajectories $\tau_1, ..., \tau_n$, a (possibly randomized) policy π ϵ-*matches their feature expectations *if and only if* $\|\mu(\pi) - \frac{1}{n}\sum_{i=1}^{n} \phi(\tau_i)\|_2 \leq \epsilon$.

In a nutshell, due to the linearity of the reward function, two policies that have the same feature expectations have the same reward. Therefore, if the observed trajectories closely mimic the feature expectations of π_u, and a policy $\tilde{\pi}$ matches the feature expectations of the observed trajectories, then the reward of $\tilde{\pi}$ would be almost identical to that of π^u. This is formalized in the following theorem, whose proof is relegated to Appendix 10.8.

Theorem 60. *Assume that $\|\phi(s, a)\|_\infty \leq 1$ for all $s \in S, a \in A$. Let $\mathbf{w}^\star$ such that $\|\mathbf{w}^\star\|_2 \leq 1$, fix any $\mathbf{w}_1, \ldots, \mathbf{w}_n$, and, for all $i \in N$, let τ_i be a trajectory of length L sampled by executing π_i. Let $\tilde{\pi}$ be a policy that $\epsilon/3-$matches the feature expectation of these trajectories. If*

$$n \geq \frac{72 \ln\left(\frac{2}{\delta}\right) d}{\epsilon^2 (1-\gamma)^2} \quad \text{and} \quad L \geq \log_{1/\gamma} \frac{3\sqrt{d}}{(1-\gamma)\epsilon}$$

then, with probability at least $1 - \delta$, it holds that $\left|R^{\mathbf{w}^\star}(\tilde{\pi}) - R^{\mathbf{w}^\star}(\pi^u)\right| \leq \epsilon$.

Note that the required number of agents n may be significant; fortunately, we can expect access to data from many agents in applications of interest. For example, Noothigattu et al. [215] built a system that decides ethical dilemmas based on data collected from 1.3 million people.

To apply Theorem 60, we need to use IRL algorithms that match feature expectations. We have identified two algorithms that satisfy this property: the *Apprenticeship Learning* algorithm of Abbeel and Ng [2], and the *Max Entropy* algorithm of Ziebart et al. [329]. For completeness we present these algorithms, and formally state their feature-matching guarantees, in Appendix 10.7.

10.4 How Good is the Uniform Mixture?

In Section 10.3 we showed that it is possible to (essentially) match the performance of the uniform mixture with respect to the ground truth reward function. In this section we justify the idea of competing with the uniform mixture in two ways: first, we show that the uniform mixture approximates the optimal policy under certain assumptions on the noise, and, second, we prove that in the worst case it is actually impossible to outperform the uniform mixture.

10.4.1 The Uniform Mixture Approximates the Optimal Policy

Recall that for all $i \in N$, $\mathbf{w}_i = \mathbf{w}^\star + \boldsymbol{\eta}_i$. It is clear that without imposing some structure on the noise vectors $\boldsymbol{\eta}_i$, no algorithm would be able to recover a policy that does well with respect to $\mathbf{w}^\star$. **Assumptions.** Let us assume, then, that the noise vectors $\boldsymbol{\eta}_i$ are such that the η_{ik} are independent and each η_{ik}^2 is sub-exponential. Formally, a random variable X is *sub-exponential* if there are non-negative parameters (ν, b) such that $\mathbb{E}\left[\exp\left(\lambda(X - \mathbb{E}[X])\right)\right] \le \exp\left(\nu^2\lambda^2/2\right)$ for all $|\lambda| < 1/b$. This flexible definition simply means that the moment generating function of the random variable X is bounded by that of a Gaussian in a neighborhood of 0. Note that if a random variable is sub-Gaussian, then its square is sub-exponential. Hence, our assumption is strictly weaker than assuming that each η_{ik} is sub-Gaussian.

Despite our assumption about the noise, it is *a priori* unclear that the uniform mixture would do well. The challenge is that the noise operates on the coordinates of the individual weight vectors, which in turn determine individual rewards, but, at first glance, it seems plausible that relatively small perturbations of rewards would lead to severely suboptimal policies. Our result shows that this is not the case: π^u is approximately optimal with respect to $R^{\mathbf{w}^\star}$, in expectation.

Theorem 61. *Assume that $\|\phi(s,a)\|_\infty \le 1$ for all $s \in S, a \in A$. Let $\mathbf{w}^\star$ such that $\|\mathbf{w}^\star\|_2 \le 1$, and suppose that $\mathbf{w}_1, ..., \mathbf{w}_n$ are drawn from i.i.d. noise around $\mathbf{w}^\star$, i.e., $\mathbf{w}_i = \mathbf{w}^\star + \boldsymbol{\eta}_i$, where each of its coordinates is such that η_{ik}^2 is an independent sub-exponential random variable with parameters (ν, b). Then*

$$\mathbb{E}[R^{\mathbf{w}^\star}(\pi^u)] \ge R^{\mathbf{w}^\star}(\pi^\star) - O\left(d\sqrt{u} + \nu\sqrt{\frac{d}{u}} + \frac{b}{\sqrt{u}}\right),$$

where $u = \frac{1}{d}\sum_{k=1}^{d} \mathbb{E}\left[\eta_{ik}^2\right]$, and the expectation is taken over the noise.

The exact expression defining the gap between $\mathbb{E}[R^{\mathbf{w}^\star}(\pi^u)]$ and $R^{\mathbf{w}^\star}(\pi^\star)$ can be found in the proof of Theorem 61, which appears in Appendix 10.9; we give the asymptotic expression in the theorem's statement because it is easier to interpret. As one might expect, this gap increases as ν or b is increased (and, in a linear fashion). This is intuitive because a smaller ν or b imposes a strictly stronger assumption on the sub-exponential random variable (and its tails).

Theorem 61 shows that the gap depends linearly on the number of features d. An example given in Appendix 10.10 shows that this upper bound is tight. Nevertheless, the tightness holds in the worst case, and one would expect the practical performance of the uniform mixture to be very good. To corroborate this intuition, we provide (unsurprising) experimental results in Appendix 10.11.

10.4.2 It is Impossible to Outperform the Uniform Mixture in the Worst Case

An ostensible weakness of Theorem 61 is that even as the number of agents n goes to infinity, the reward of the uniform mixture may not approach that of the optimal policy, that is, there is a persistent gap. The example given in Section 10.4.1 shows the gap is not just an artifact of our analysis. This is expected, because the data contains some agents with suboptimal policies π_i, and a uniform mixture over these suboptimal policies must itself be suboptimal.

It is natural to ask, therefore, whether it is generally possible to achieve performance arbitrarily close to $\pi^\star$ (at least in the limit that n goes to infinity). The answer is negative. In fact, we show that — in the spirit of *minimax optimality* [139, 228] — one cannot hope to perform better than π^u itself in the worst case. Intuitively, there exist scenarios where it is impossible to tell good and bad policies apart by looking at the data, which means that the algorithm's performance depends on what can be gleaned from the "average data".

This follows from a surprising[1] result that we think of as "non-identifiability" of the optimal policy. To describe this property, we introduce some more notation. The distribution over the weight vector of each agent i, $\mathbf{w}_i = \mathbf{w}^\star + \boldsymbol{\eta}_i$, in turn induces a distribution over the optimal policy π_i executed by each agent. Denote this distribution by $\mathcal{P}(\mathbf{w}^\star)$.[2] Hence, each agent's optimal policy π_i is just a sample from this distribution $\mathcal{P}(\mathbf{w}^\star)$. In particular, as the number of agents goes to infinity, the empirical distribution of their optimal policies would exactly converge to $\mathcal{P}(\mathbf{w}^\star)$.

Assumptions. For the rest of this section, we make minimal assumptions on the noise vector $\boldsymbol{\eta}_i$. In particular, we merely assume that $\boldsymbol{\eta}_i$ follows a continuous distribution and that each of its coordinates is i.i.d. We are now ready to state our non-identifiability lemma.

Lemma 77 (non-identifiability)**.** *For every continuous distribution $\mathcal{D}$ over $\mathbb{R}$, if η_{ik} is independently sampled from $\mathcal{D}$ for all $i \in N$ and $k \in [d]$, then there exists an MDP and weight vectors $\mathbf{w}_a^\star$, $\mathbf{w}_b^\star$ with optimal policies $\pi_a^\star$, $\pi_b^\star$, respectively, such that $\pi_a^\star \neq \pi_b^\star$ but $\mathcal{P}(\mathbf{w}_a^\star) = \mathcal{P}(\mathbf{w}_b^\star)$.*

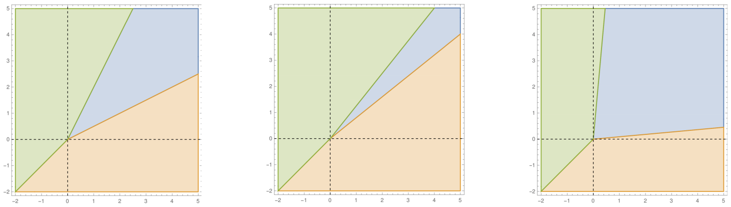
Even if we had an infinite number of trajectories in our data, and even if we knew the exact optimal policy played by each player i, this information would amount to knowing $\mathcal{P}(\mathbf{w}^\star)$. Hence, if there exist two weight vectors $\mathbf{w}_a^\star$, $\mathbf{w}_b^\star$ with optimal policies $\pi_a^\star$, $\pi_b^\star$ such that $\pi_a^\star \neq \pi_b^\star$ and $\mathcal{P}(\mathbf{w}_a^\star) = \mathcal{P}(\mathbf{w}_b^\star)$, then we would not be able to identify whether the optimal policy is $\pi_a^\star$ or $\pi_b^\star$ regardless of how much data we had.

The proof of Lemma 77 is relegated to Appendix 10.12. Here we provide a proof sketch.

Proof sketch of Lemma 77. The intuition for the lemma comes from the construction of an MDP with three possible policies, all of which have probability $1/3$ under $\mathcal{P}(\mathbf{w}^\star)$, even though one is better than the others. This MDP has a single state s, and three actions $\{a, b, c\}$ that lead back to s. Denote the corresponding policies by π_a, π_b, π_c. Let the feature vector be $\phi(s, a) = [0.5, 0.5], \phi(s, b) = [1, -\delta/2], \phi(s, c) = [-\delta/2, 1]$, where $\delta > 0$ is a parameter. Let the ground truth weight vector be $\mathbf{w}^\star = (v_o, v_o)$, where v_o is such that the noised weight vector $\mathbf{w} = \mathbf{w}^\star + \boldsymbol{\eta}$ has probability strictly more than $1/3$ of lying in the first quadrant; an appropriate value of δ always exists for any noise distribution that is continuous and i.i.d. across coordinates s.t the above holds.

[1] At least it was surprising for us — we spent significant effort trying to prove the opposite result!

[2] Note that this distribution does not depend on i itself since the noise $\boldsymbol{\eta}_i$ is i.i.d. across the different agents.



$\delta = 1$ $\delta = 0.25$ $\delta = 10$

Figure 10.1: Regions of each optimal policy for different values of δ. Blue depicts the region where π_a is optimal, orange is where π_b is optimal, and green is where π_c is optimal.

Let us look at weight vectors $\mathbf{w}$ for which each of the three policies π_a, π_b and π_c are optimal. π_a is the optimal policy when $\mathbf{w}^\intercal \mu_a > \mathbf{w}^\intercal \mu_b$ and $\mathbf{w}^\intercal \mu_a > \mathbf{w}^\intercal \mu_c$, which is the intersection of the half-spaces $\mathbf{w}^\intercal(-1, 1+\delta) > 0$ and $\mathbf{w}^\intercal(1+\delta, -1) > 0$. Similarly, we can reason about the regions where π_b and π_c are optimal. These regions are illustrated in Figure 10.1 for different values of δ. Informally, as δ is decreased, the lines separating (π_a, π_c) and (π_a, π_b) move closer to each other (as shown for $\delta = 0.25$), while as δ is increased, these lines move away from each other (as shown for $\delta = 10$). By continuity and symmetry, there exists δ such that the probability of each of the regions (with respect to the random noise) is exactly $1/3$, showing that the MDP has the desired property.

To complete the proof of the lemma, we extend the MDP by adding two more features to the existing two. By setting these new features appropriately (in particular, by cycling the two original features across the arms), we can show that the two weight vectors $\mathbf{w}_a^\star = (v_o, v_o, 0, 0)$ and $\mathbf{w}_b^\star = (0, 0, v_o, v_o)$ lead to $\mathcal{P}(\mathbf{w}_a^\star) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}) = \mathcal{P}(\mathbf{w}_b^\star)$, even though their corresponding optimal policies are π_a and π_b, respectively. □

For the next theorem, therefore, we can afford to be “generous:” we will give the algorithm (which is trying to compete with π^u) access to $\mathcal{P}(\mathbf{w}^\star)$, instead of restricting it to sampled trajectories. Formally, the theorem holds for any algorithm that takes a distribution over policies as input, and returns a randomized policy.

Theorem 62. *For every continuous distribution $\mathcal{D}$ over $\mathbb{R}$, if η_{ik} is independently sampled from $\mathcal{D}$ for all $i \in N$ and $k \in [d]$, then there exists an MDP such that for any algorithm $\mathcal{A}$ from distributions over policies to randomized policies, there exists a ground truth weight vector $\mathbf{w}^\star$ such that $R^{\mathbf{w}^\star}(\mathcal{A}(\mathcal{P}(\mathbf{w}^\star)) \leq R^{\mathbf{w}^\star}(\pi^u) < R^{\mathbf{w}^\star}(\pi^\star)$.*

In words, the constructed instance is such that, even given infinite data, no algorithm can outperform the uniform mixture, and, moreover, the reward of the uniform mixture is bounded away from the optimum. The theorem's proof is given in Appendix 10.13.

10.5 The Inverse Multi-Armed Bandit Problem

In Section 10.4, we have seen that it is impossible to outperform the uniform mixture in the worst case, as the optimal policy is not identifiable. However, it is natural to ask when the optimal policy is identifiable and how it may be practically recovered. In this section we give an encouraging answer, albeit in a restricted setting.

Specifically, we focus on the multi-armed bandit problem, which is an MDP with a single state. Note that the non-identifiability result of Lemma 77 still holds in this setting, as the example used in its proof is an MDP with a single state. Hence, even in this setting of bandits, it is impossible to outperform the uniform mixture in the worst case. However, we design an algorithm that can guarantee optimal performance when the problem is identifiable, under some additional conditions.

Like the general setting considered earlier, there exists a ground truth weight vector $\mathbf{w}^\star$, and for each agent $i \in N$, $\mathbf{w}_i = \mathbf{w}^\star + \boldsymbol{\eta}_i$. For this section, we assume the noise vector $\boldsymbol{\eta}_i$ to be Gaussian and i.i.d. across agents and coordinates. In particular, $\boldsymbol{\eta}_i \sim \mathcal{N}(0, \sigma^2 I_d)$, and independent across i.

The bandit setting is equivalent to a single-state MDP, and hence the components S, T, γ and D are moot. Instead, there are m arms to pull, denoted by $A = \{1, 2, \ldots, m\}$. Similar to our original feature function ϕ, we now have features $\mathbf{x}_j \in \mathbb{R}^d$ associated with arm j, for each $j \in A$. Although in standard stochastic bandit problems we have a reward sampled from a distribution when we pull an arm, we care only about its mean reward in this section. For weight vector $\mathbf{w}$, the (mean) reward of pulling arm j is given by $r^{\mathbf{w}}(j) = \mathbf{w}^\intercal \mathbf{x}_j$. For each agent i (with weight vector $\mathbf{w}_i$), we assume that we observe the optimal arm being played by this agent, i.e., $\tilde{a}_i = \text{argmax}_{j \in A} \mathbf{w}_i^\intercal \mathbf{x}_j$.

We observe the dataset $\mathcal{D} = \{\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_n\}$ which is the set of optimal arms played by the agents. Define $\mathcal{Q}(\mathbf{w}^\star)$ to be the distribution over optimal arms induced when the ground truth weight vector is $\mathbf{w}^\star$. In particular, ground truth weight vector $\mathbf{w}^\star$ induces a distribution over the noised weight vector of each agent (via $\mathbf{w} = \mathbf{w}^\star + \eta$), which in turn induces a discrete distribution over the optimal arm that would be played, which we call $\mathcal{Q}(\mathbf{w}^\star)$ — analogously to the $\mathcal{P}(\mathbf{w}^\star)$ of Section 10.4. Observe that the dataset $\mathcal{D}$ could be rewritten as a distribution over arms, $\tilde{\mathcal{Q}} = (\tilde{\mathcal{Q}}_1, \tilde{\mathcal{Q}}_2, \ldots, \tilde{\mathcal{Q}}_m)$, which is the observed distribution of optimal arms. Moreover, as each agent's optimal arm played is an i.i.d. sample from $\mathcal{Q}(\mathbf{w}^\star)$, the empirical distribution $\tilde{\mathcal{Q}}$ is an unbiased estimate of $\mathcal{Q}(\mathbf{w}^\star)$.

The *inverse multi-armed bandit problem* is to recover $\mathbf{w}^\star$ given the distribution $\tilde{\mathcal{Q}}$, which allows us to identify the optimal arm. In order to achieve this, we aim to find $\mathbf{w}$ such that $\mathcal{Q}(\mathbf{w}) = \tilde{\mathcal{Q}}$, or matches it as closely as possible. Ideally, we would want to find $\mathbf{w}$ such that $\mathcal{Q}(\mathbf{w}) = \mathcal{Q}(\mathbf{w}^\star)$,[3] but since we do not have access to $\mathcal{Q}(\mathbf{w}^\star)$, we use the unbiased estimate $\tilde{\mathcal{Q}}$ in its place.[4] Below, we produce conditions under which the optimal policy is recoverable, and provide a practical algorithm that achieves this for all settings that meet the criteria.

10.5.1 Identifying the Optimal Arm

Since the constraint $\mathcal{Q}(\mathbf{w}) = \tilde{\mathcal{Q}}$ is "far" from being convex in $\mathbf{w}$, we reformulate the problem such that the new problem is convex, and all its optimal solutions satisfy the required constraint

[3] Note that there might be multiple $\mathbf{w}$ such that $\mathcal{Q}(\mathbf{w}) = \mathcal{Q}(\mathbf{w}^\star)$. However, since we care only about the corresponding optimal arm, and identifiability tells us that all weight vectors with the same $\mathcal{Q}$ value have the same optimal arm, we just need to find one such weight vector.

[4] In most cases we will have collected sufficient data such that the optimal arm corresponding to $\tilde{\mathcal{Q}}$ coincides with the optimal arm corresponding to $\mathcal{Q}(\mathbf{w}^\star)$. Although they may not coincide, this probability goes to zero as the size of the dataset $\mathcal{D}$ increases.

(and vice versa). The new objective we use is the cross entropy loss between $\tilde{\mathcal{Q}}$ and $\mathcal{Q}(\mathbf{w})$. That is, the optimization problem to solve is

$$\min_{\mathbf{w}} - \sum_{k \in A} \tilde{\mathcal{Q}}_k \log \mathcal{Q}(\mathbf{w})_k. \tag{10.1}$$

It is obvious that this objective is optimized at points with $\mathcal{Q}(\mathbf{w}) = \tilde{\mathcal{Q}}$, if the original problem was feasible. Otherwise, it finds $\mathbf{w}$ whose $\mathcal{Q}$ is as close to $\tilde{\mathcal{Q}}$ as possible in terms of cross-entropy. Furthermore, this optimization problem is convex under a simple condition, which requires the definition of X_k as an $(m-1) \times d$ matrix with rows of the form $(\mathbf{x}_k - \mathbf{x}_j)^\intercal$, for each $j \in A \setminus \{k\}$.

Theorem 63. *Optimization problem* (10.1) *is convex if $X_k X_k^\intercal$ is invertible for each $k \in A$.*

The proof of the theorem appears in Appendix 10.14. An exact characterization of when $X_k X_k^\intercal$ is full rank is $\text{rank}(X_k X_k^\intercal) = \text{rank}(X_k) = m - 1$, i.e. when X_k is full row rank. For this to be true, a necessary condition is that $d \geq m - 1$ as $\text{rank}(X_k) \leq \min(d, m-1)$. And under this condition, the requirement for X_k to to be full row rank is that the rows $(\mathbf{x}_k - \mathbf{x}_j)^\intercal$ are linearly independent, which is very likely to be the case, unless the feature vectors were set up adversarially. One potential scenario where the condition $d \geq m - 1$ would arise is when there are many features but feature vectors $\mathbf{x}_j$ are sparse.

As the optimization problem (10.1) is convex, we can use gradient descent to find a minimizer. And for this, we need to be able to compute the gradient accurately, which we show is possible; the calculation is given in Appendix 10.15.

Importantly, we can also use our procedure to determine whether the optimal arm is identifiable. Given $\tilde{\mathcal{Q}}$, we solve the optimization problem (10.1) to first find a $\mathbf{w}_o$ such that $\mathcal{Q}(\mathbf{w}_o) = \tilde{\mathcal{Q}}$. Let $\mathbf{w}_o$ have the optimal arm $a_o \in A$. Now, our goal is to check if there exists any other weight $\mathbf{w}$ that has $\mathcal{Q}(\mathbf{w}) = \tilde{\mathcal{Q}}$ but whose corresponding optimal arm is not a_o. To do this, we can build a set of convex programs, each with the exact same criterion (taking care of the $\mathcal{Q}(\mathbf{w}) = \tilde{\mathcal{Q}}$ requirement), but with the constraint that arm $a_i \neq a_o$ is the optimal arm (or at least beats a_o) with respect to $\mathbf{w}$. In particular, the constraint for program i could be $\mathbf{w}^\intercal \mathbf{x}_i > \mathbf{w}^\intercal \mathbf{x}_{a_o}$.[5] As this is a simple affine constraint, solving the convex program is very similar to running gradient descent as before. If any of these convex programs outputs an optimal solution that satisfies $\mathcal{Q}(\mathbf{w}) = \tilde{\mathcal{Q}}$, then the problem is not identifiable, as it implies that there exist weight vectors with different optimal arms leading to the same $\tilde{\mathcal{Q}}$. On the other hand, if none of them satisfies $\mathcal{Q}(\mathbf{w}) = \tilde{\mathcal{Q}}$, we can conclude that a_o is the desired unique optimal arm.

10.5.2 Experiments

We next study the empirical performance of our algorithm for the inverse multi-armed bandit problem. We focus on instances inspired by the counter-example from Lemma 77. The reason for this is that in randomly generated bandit problems, the optimal arm $a^\star$ is very likely to be the mode of $\mathcal{Q}(\mathbf{w}^\star)$, making the mode of $\tilde{\mathcal{Q}}$ a very good estimator of $a^\star$.[6] By contrast, the counterexample allows us to generate "hard" instances.

[5] The strong inequality can be implemented in the standard way via $\mathbf{w}^\intercal \mathbf{x}_i \geq \mathbf{w}^\intercal \mathbf{x}_{a_o} + \epsilon$ for a sufficiently small $\epsilon > 0$ that depends on the program's bit precision.

[6] This is because, for each arm a, the region $\mathcal{R}_a = \{\mathbf{w} : \mathbf{w}^\intercal \mathbf{x}_a \geq \mathbf{w}^\intercal \mathbf{x}_j \text{ for each } j\}$, corresponding to where arm a is optimal, forms a polytope, and the optimal arm's region $\mathcal{R}_{a^\star}$ contains $\mathbf{w}^\star$. Hence, as long as $\mathcal{R}_{a^\star}$ has enough

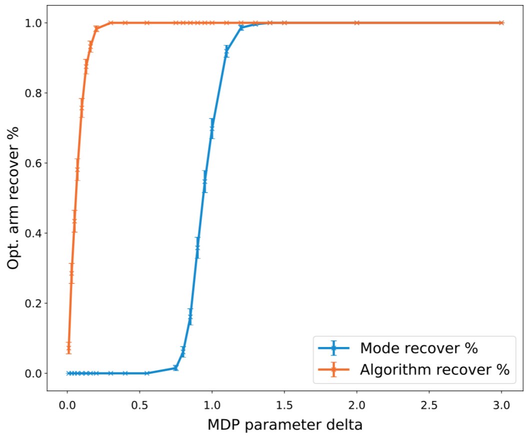


Figure 10.2: Performance as δ is varied.

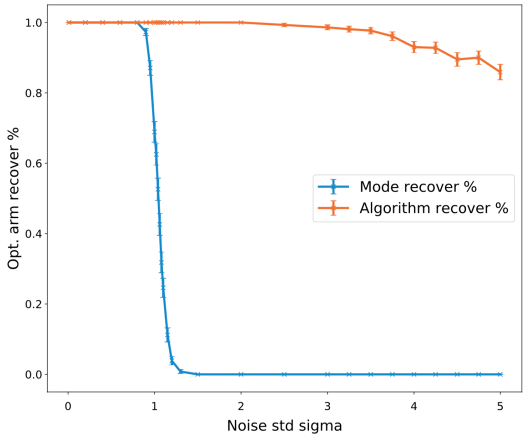


Figure 10.3: Performance as σ is varied.

Specifically, the bandit instances we consider have two features ($d = 2$) and three arms $A = \{1, 2, 3\}$, and their features are defined as $\mathbf{x}_1 = [1, 1]$, $\mathbf{x}_2 = [2, -\delta]$ and $\mathbf{x}_3 = [-\delta, 2]$, where $\delta > 0$ is a positive constant. The ground truth weight vector is given as $\mathbf{w}^\star = [1, 1]$. Hence, for any $\delta > 0$, the optimal arm is arm 1. The noise is $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2)$. Such an instance is very similar to the one of Lemma 77, except that the features are not replicated to extend from two to four features, and hence the problem remains identifiable.

Observe that when the value of δ is small enough, the blue region of Figure 10.1 becomes a sliver, capturing a very small density of the noise $\boldsymbol{\eta}$, and causing arm 1 to not be the mode of $\mathcal{Q}(\mathbf{w}^\star)$. Alternatively, for a given value of δ, if σ is large enough, most of the noise's density escapes the blue region, again causing arm 1 to not be the mode of $\mathcal{Q}(\mathbf{w}^\star)$. In the following experiments, we vary both δ and σ, and show that even when the optimal arm almost never appears in $\mathcal{Q}(\mathbf{w}^\star)$, our algorithm is able to recover it.

Varying parameter δ. In the first set of experiments, we fix the noise standard deviation σ to 1, generate $n = 500$ agents according to the noise $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2)$, and vary parameter δ from 0.01 to 3. Figure 10.2 shows the percentage of times our algorithm and the mode recover the optimal arm 1. This graph is averaged over 1000 runs, and error bars depict 95% confidence intervals.

When δ is extremely close to 0, the optimal arm's region almost vanishes. Hence, small differences between $\tilde{\mathcal{Q}}$ and $\mathcal{Q}(\mathbf{w}^\star)$ could have a substantial effect, and unless $\mathbf{w}^\star$ is numerically recovered within this sliver, the optimal arm would not be recovered. As we move to even slightly larger values of δ, however, the performance of the algorithm improves substantially and it ends up recovering the optimal arm 100% of the time.

By contrast, as δ is varied from 0 to ∞, the density of the noise $\boldsymbol{\eta}$ captured by the blue region increases continuously from 0 to that of the first quadrant. In particular, there is a point where $\mathcal{Q}(\mathbf{w}^\star)$ has probability tied across the three arms, after which arm 1 is always the mode (i.e. mode has 100% performance), and before which arms 2 and 3 are the modes (i.e the mode has 0% performance). This tipping point is evident from the graph and occurs around $\delta = 1$.[7] Observe

volume around $\mathbf{w}^\star$, it would capture a majority of the density of the noise $\boldsymbol{\eta}$, and $a^\star$ would be the mode of the distribution $\mathcal{Q}(\mathbf{w}^\star)$.

[7] The transition in this graph is smoother than a step function because we use the empirical mode from $\tilde{\mathcal{Q}}$ whose

that the performance of the algorithm rises to 100% much before this tipping point, serving as evidence that it can perform well even if the optimal arm barely appears in the dataset. Similar results on when the parameters are set to $\sigma \in \{0.5, 2.0\}$ or $n \in \{250, 1000\}$ may be found in Appendix 10.16.1.

Varying noise parameter σ. Next, we fix the parameter δ to 1 and generate $n = 500$ agents according to noise $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2)$, while varying the noise parameter σ from 0.01 to 5. Figure 10.3 shows the percentage of times our algorithm and the mode recover the optimal arm 1. This graph is also averaged over 1000 runs, and error bars depict 95% confidence intervals.

The results are similar in spirit to Figure 10.2. When σ is extremely large (relative to the ground truth vector $\mathbf{w}^\star = [1, 1]$), the weight space becomes less and less distinguishable with respect to the corresponding $\mathcal{Q}$ values. In particular, small differences between $\tilde{\mathcal{Q}}$ and $\mathcal{Q}(\mathbf{w}^\star)$ again have a substantial effect on the corresponding optimal arms, causing a suboptimal arm to be recovered. At more reasonable levels of noise, however, we can see that the algorithm recovers the optimal arm 100% of the time.

The mode's performance also has a similar flavor to Figure 10.2. For a given value of δ, the regions of Figure 10.1 are completely decided. When σ is close to zero, the noise is almost negligible, and hence the blue region captures most of the density of the noise $\boldsymbol{\eta}$, and the optimal arm is the mode. But as σ is varied from 0 to ∞, the density captured by this region decreases continuously from 1 to a ratio of the volumes of the regions. In particular, we again come across a point where $\mathcal{Q}(\mathbf{w}^\star)$ has probability tied across the three arms, before which arm 1 is always the mode (i.e. mode has 100% performance), and after which arms 2 and 3 are the modes (i.e. the mode has 0% performance). Note that, for $\sigma = 1$, this point is achieved around $\delta = 1$ (Figure 10.2). Hence, when we vary σ while fixing $\delta = 1$, the tipping point is expected to be achieved around $\sigma = 1$, which is indeed the case, as evident from Figure 10.3. Again, observe that the performance of the algorithm is still around 100% significantly after this tipping point. Similar results on when the parameters are set to $\delta \in \{0.5, 2.0\}$ or $n \in \{250, 1000\}$ may be found in Appendix 10.16.2.

10.6 Discussion

We have shown that it is possible to match the performance of the uniform mixture π^u, or that of the average agent. In Section 10.5 we then established that it is possible to learn policies from demonstrations with *superior* performance compared to the teacher, albeit under simplifying assumptions. An obvious challenge is to relax the assumptions, but this is very difficult, and we do not know of existing work that can be applied directly to our general setting. Indeed, the most relevant theoretical work is that of Syed and Schapire [272]. Their approach can only be applied if the sign of the reward weight is known for every feature. This is problematic in our setting as some agents may consider a feature to be positive, while others consider it to be negative. A priori, it is unclear how the sign can be determined, which crucially invalidates the algorithm's theoretical guarantees. Moreover, it is unclear under which cases the algorithm would produce a policy with superior performance, or if such cases exist.

performance varies smoothly as the distance between probabilities of arms 1 and $\{2, 3\}$ changes.

We also remark that, although in the general setting we seek to compete with π^u, we are actually doing something quite different. Indeed, *ex post* (after the randomness has been instantiated) the uniform mixture π^u simply coincides with one of the individual policies. By contrast, IRL algorithms pool the feature expectations of the trajectories $\tau_1, \dots, \tau_n$ together, and try to recover a policy that approximately matches them. Therefore, we believe that IRL algorithms do a much better job of aggregating the individual policies than π^u does, while giving almost the same optimality guarantees.

Finally, we wish to highlight potential extensions to our work. One promising extension is to better understand how notions of social choice could be folded into our framework of aggregation through IRL; we include a short discussion of this point in Appendix 10.17. Another promising direction is to understand what can be gained (e.g., in terms of sample complexity) by going beyond the classical setup of IRL, for instance by allowing for interaction and/or communication between the agents and teachers.

Appendix

10.7 IRL Algorithms

In this appendix we identify two well-known algorithms that match feature expectations.

10.7.1 Apprenticeship Learning

Under the classic Apprenticeship Learning algorithm, designed by Abbeel and Ng [2], a policy $\pi^{(0)}$ is selected to begin with. Its feature expectation $\mu(\pi^{(0)})$ is computed and added to the bag of feature expectations. At each step,

$$t^{(i)} = \max_{\mathbf{w}:\|\mathbf{w}\|_2 \le 1} \min_{j \in \{0,..,i-1\}} \mathbf{w}^\intercal \left(\frac{1}{n} \sum_{i=1}^{n} \phi(\tau_i) - \mu\left(\pi^{(j)}\right) \right)$$

is computed along with the weight $\mathbf{w}^{(i)}$ that achieved this. When $t^{(i)} \le \epsilon$ the algorithm terminates, otherwise the associated optimal policy $\pi^{(i)}$ is computed, and its corresponding feature expectation vector $\mu(\pi^{(i)})$ is added to the bag of feature expectations. The algorithm provides the following guarantee.

Theorem 64 (adapted from Abbeel and Ng [2])**.** *For any $\epsilon > 0$, the Apprenticeship Learning algorithm terminates with $t^{(i)} \le \epsilon$ after a number of iterations bounded by*

$$T = O\left(\frac{d}{(1-\gamma)^2 \epsilon^2} \ln \frac{d}{(1-\gamma)\epsilon} \right),$$

and outputs a mixture over $\pi^{(1)}, ..., \pi^{(T)}$ that ϵ-matches the feature expectations of the observed trajectories.

Note that it is necessary for us to use a randomized policy, in contrast to the case where a single deterministic policy generated all the trajectory samples, as, in our case, typically there is no single deterministic policy that matches the feature expectations of the observed trajectories.

10.7.2 Max Entropy

We next discuss the Max Entropy algorithm of Ziebart et al. [329], which optimizes the max entropy of the probability distribution over trajectories subject to the distribution satisfying approximate feature matching. This is done to resolve the potential ambiguity of there being multiple stochastic policies that satisfy feature matching. Optimizing entropy is equivalent to maximizing the regularized likelihood $L(\mathbf{w})$ of the observed trajectories. Specifically, the objective is

$$L(\mathbf{w}) = \max_{\mathbf{w}} \sum_{i=1}^{n} \log \Pr[\tau_i | \mathbf{w}, T] - \sum_{i=1}^{d} \rho_i \|\mathbf{w}_i\|_1,$$

with

$$\Pr[\tau_i | \mathbf{w}, T] = \frac{e^{\mathbf{w}^\intercal \phi(\tau_i)}}{Z(\mathbf{w}, T)} \prod_{s_t, a_t, s_{t+1} \in \tau_i} T(s_t, a_t, s_{t+1}).$$

The regularization term is introduced to allow for approximate feature matching since the observed empirical feature expectation may differ from the true expectation. Let $\boldsymbol{\rho}$ be an upper bound on this difference, i.e., for all $k = 1, \ldots, d$,

$$\rho_k \geq \left| \frac{1}{n} \sum_{i=1}^{n} \phi(\tau_i)_k - \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^{n} \phi(\tau_i)_k \right] \right|.$$

One may then derive that the gradient of $L(\mathbf{w})$ is the difference between the feature expectation induced $\mathbf{w}$ and the observed feature expectation.

Theorem 65 (adapted from Ziebart et al. [329])**.** *Let $\epsilon > 0$, and assume that the Max Entropy algorithm finds $\mathbf{w}$ such that $|\nabla L(\mathbf{w})| < \epsilon$, then this $\mathbf{w}$ corresponds to a randomized policy that $(\epsilon + \|\boldsymbol{\rho}\|_1)$-matches the feature expectations of the observed trajectories.*

The assumption on the gradient is needed because the above optimization objective is derived only with the approximate feature matching constraint. MDP dynamics is not explicitly encoded into the optimization. Instead, heuristically, the likelihood of each trajectory $\Pr[\tau_i | \mathbf{w}, T]$ is weighted by the product of the transition probabilities of its steps. The follow-up work of Ziebart [328] addresses this by explicitly introducing MDP constraints into the optimization, and optimizing for the causal entropy, thereby achieving unconditional feature matching.

10.8 Proof of Theorem 60

We need to bound the difference between $R^{\mathbf{w}^\star}(\tilde{\pi})$ and $R^{\mathbf{w}^\star}(\pi^u)$. First, recall that $\tilde{\pi}$ $\epsilon/3-$matches the feature expectations of $\tau_1, \ldots, \tau_n$. It holds that

$$\begin{aligned} \left| R^{\mathbf{w}^\star}(\tilde{\pi}) - (\mathbf{w}^\star)^\intercal \left(\frac{1}{n} \sum_{i=1}^{n} \phi(\tau_i) \right) \right| &= \left| (\mathbf{w}^\star)^\intercal \left(\mu(\tilde{\pi}) - \frac{1}{n} \sum_{i=1}^{n} \phi(\tau_i) \right) \right| \\ &\leq \|\mathbf{w}^\star\|_2 \left\| \mu(\tilde{\pi}) - \frac{1}{n} \sum_{i=1}^{n} \phi(\tau_i) \right\|_2 \leq \frac{\epsilon}{3}, \end{aligned} \tag{10.2}$$

where the second transition follows from the Cauchy-Schwarz inequality, and the last from the assumption that $\|\mathbf{w}^{\star}\|_2 \leq 1$. Hence, it is sufficient to demonstrate that, with probability at least $1-\delta$,

$$\left|(\mathbf{w}^{\star})^{\intercal}\left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i)\right) - R^{\mathbf{w}^{\star}}(\pi^u)\right| \leq \frac{2\epsilon}{3}, \tag{10.3}$$

as the theorem would then follow from Equations (10.2), and (10.3) by the triangle inequality.

We note that the difference on the left hand side of Equation (10.3) is due to two sources of noise.

1. The finite number of samples of trajectories which, in our setting, originates from multiple policies.
2. The truncated trajectories τ_i which are limited to L steps.

Formally, let τ_i' denote the infinite trajectory for each i, then the difference can be written as

$$\begin{aligned}\left|(\mathbf{w}^{\star})^{\intercal}\left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i)\right) - R^{\mathbf{w}^{\star}}(\pi^u)\right| \leq \left|(\mathbf{w}^{\star})^{\intercal}\left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i)\right) - (\mathbf{w}^{\star})^{\intercal}\left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i')\right)\right| \\ + \left|(\mathbf{w}^{\star})^{\intercal}\left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i')\right) - R^{\mathbf{w}^{\star}}(\pi^u)\right|\end{aligned}$$

Bounding finite sample noise. We wish to bound:

$$\left|(\mathbf{w}^{\star})^{\intercal}\left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i')\right) - R^{\mathbf{w}^{\star}}(\pi^u)\right| = \left|\frac{1}{n}\left(\sum_{i=1}^{n}(\mathbf{w}^{\star})^{\intercal}(\phi(\tau_i') - \mu(\pi_i))\right)\right|. \tag{10.4}$$

Define random variable $Z_i = (\mathbf{w}^{\star})^{\intercal}(\phi(\tau_i') - \mu(\pi_i))$. Then the right-hand side of Equation (10.4) may be expressed as $|\frac{1}{n}\sum_{i=1}^{n} Z_i|$. Furthermore, Z_i is such that $\mathbb{E}[\phi(\tau_i')_k] = \mu(\pi_i)_k$ for all $k = 1, \ldots, d$. This is because a policy π_i defines a distribution over trajectories, and τ_i' is a draw from this distribution. Using the linearity of expectation, it follows that

$$\mathbb{E}[Z_i] = (\mathbf{w}^{\star})^{\intercal}\mathbb{E}[\phi(\tau_i') - \mu(\pi_i)] = 0.$$

Moreover,

$$|Z_i| \leq \|\mathbf{w}^{\star}\|_2\|\phi(\tau_i')\|_2 + \|\mathbf{w}^{\star}\|_2\|\mu(\pi_i)\|_2 \leq \frac{2\sqrt{d}}{1-\gamma},$$

since $\|\phi(s,\cdot)\|_{\infty} = 1$. Thus, using Hoeffding's inequality, we conclude that

$$\Pr\left[\left|\frac{1}{n}\sum_{i=1}^{n} Z_i\right| > \frac{\epsilon}{3}\right] \leq 2\text{exp}\left(-\frac{2n\left(\frac{\epsilon}{3}\right)^2}{(\frac{4\sqrt{d}}{1-\gamma})^2}\right) \leq \delta,$$

where the last transition holds by our choice of n.

Bounding bias due to truncated trajectories. We wish to bound:

$$\left|(\mathbf{w}^\star)^\intercal \left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i)\right) - (\mathbf{w}^\star)^\intercal \left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i')\right)\right|.$$

For each trajectory τ_i, truncating after L steps incurs a reward difference of:

$$\begin{aligned}\left|(\mathbf{w}^\star)^\intercal\phi(\tau_i') - (\mathbf{w}^\star)^\intercal\phi(\tau_i)\right| &= \left|(\mathbf{w}^\star)^\intercal \sum_{t=L}^{\infty}\gamma^t\phi(\tau_i'(s_t), \tau_i'(a_t))\right| \\ &\leq \sum_{t=L}^{\infty}\gamma^t\|\mathbf{w}^\star\|_2\|\phi(\tau_i'(s_t), \tau_i'(a_t))\|_2 \leq \gamma^L\frac{\sqrt{d}}{1-\gamma} \leq \frac{\epsilon}{3},\end{aligned}$$

where the third transition holds because $\|\phi(\tau_i(s_t), \tau_i(a_t))\|_2 \leq \sqrt{d}$, and the last transition follows from our choice of L. Hence, we obtain

$$\left|(\mathbf{w}^\star)^\intercal \left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i)\right) - (\mathbf{w}^\star)^\intercal \left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i')\right)\right| \leq \frac{1}{n}\sum_{i=1}^{n}\left|(\mathbf{w}^\star)^\intercal\phi(\tau_i) - (\mathbf{w}^\star)^\intercal\phi(\tau_i')\right| \leq \frac{\epsilon}{3}.$$

□

10.9 Proof of Theorem 61

We require a key property of sub-exponential random variables, which is captured by the following well known tail inequality; its proof can be found, for example, in Chapter 2 of Wainwright [288].

Lemma 78. *Let $X_1, \ldots, X_m$ be independent sub-exponential random variables with parameters (ν, b). Then*

$$\Pr\left[\frac{1}{m}\sum_{j=1}^{m}(X_j - u_j) \geq t\right] \leq \begin{cases}\exp\left(-\frac{mt^2}{2\nu^2}\right) \textit{ for } 0 \leq t \leq \frac{\nu^2}{b} \\ \exp\left(-\frac{mt}{2b}\right) \textit{ for } t > \frac{\nu^2}{b}\end{cases},$$

where $u_j = \mathbb{E}[X_j]$.

Turning to the theorem's proof, as π^u is a uniform distribution over the policies $\pi_1, \ldots, \pi_n$, its expected reward is given by

$$R^{\mathbf{w}^\star}(\pi^u) = \frac{1}{n}\sum_{i=1}^{n}R^{\mathbf{w}^\star}(\pi_i). \tag{10.5}$$

Observe that $R^{\mathbf{w}^\star}(\pi_i)$ is a random variable which is i.i.d. across i, as the corresponding noise $\boldsymbol{\eta}_i$ is i.i.d. as well. We analyze the expectation of the difference with respect to $R^{\mathbf{w}^\star}(\pi^\star)$.

First, note that for a weight vector $\mathbf{w}$ and policy π,

$$R^{\mathbf{w}}(\pi) = \sum_{t=0}^{\infty} \gamma^t \sum_{s \in S} P_\pi(s,t) \mathbf{w}^\intercal \phi(s, \pi(s)),$$

where $P_\pi(s,t)$ denotes the probability of being in state s on executing policy π from the start. Hence, for each $i \in N$, we have

$$\begin{aligned}
&R^{\mathbf{w}^\star}(\pi^\star) - R^{\mathbf{w}^\star}(\pi_i) \\
&\quad = \sum_{t=0}^{\infty} \gamma^t \sum_{s \in S} \Big[P_{\pi^\star}(s,t)(\mathbf{w}^\star)^\intercal \phi(s, \pi^\star(s)) - P_{\pi_i}(s,t)(\mathbf{w}^\star)^\intercal \phi(s, \pi_i(s)) \Big] \\
&\quad = \sum_{t=0}^{\infty} \gamma^t \sum_{s \in S} \Big[P_{\pi^\star}(s,t)(\mathbf{w}_i - \boldsymbol{\eta}_i)^\intercal \phi(s, \pi^\star(s)) - P_{\pi_i}(s,t)(\mathbf{w}_i - \boldsymbol{\eta}_i)^\intercal \phi(s, \pi_i(s)) \Big] \\
&\quad = R^{\mathbf{w}_i}(\pi^\star) - R^{\mathbf{w}_i}(\pi_i) + \sum_{t=0}^{\infty} \gamma^t \sum_{s \in S} \Big[- P_{\pi^\star}(s,t) \boldsymbol{\eta}_i^\intercal \phi(s, \pi^\star(s)) + P_{\pi_i}(s,t) \boldsymbol{\eta}_i^\intercal \phi(s, \pi_i(s)) \Big] \\
&\quad \le \sum_{t=0}^{\infty} \gamma^t \sum_{s \in S} \Big[- P_{\pi^\star}(s,t) \boldsymbol{\eta}_i^\intercal \phi(s, \pi^\star(s)) + P_{\pi_i}(s,t) \boldsymbol{\eta}_i^\intercal \phi(s, \pi_i(s)) \Big] \\
&\quad = \sum_{k=1}^{d} \eta_{ik} \left[\sum_{t=0}^{\infty} \gamma^t \sum_{s \in S} \Big[- P_{\pi^\star}(s,t) \phi(s, \pi^\star(s))_k + P_{\pi_i}(s,t) \phi(s, \pi_i(s))_k \Big] \right] \\
&\quad := \sum_{k=1}^{d} \eta_{ik} \alpha_{ik},
\end{aligned}$$

where the inequality holds since $R^{\mathbf{w}_i}(\pi_i) \ge R^{\mathbf{w}_i}(\pi^\star)$, which, in turn, holds because π_i is optimal under $\mathbf{w}_i$.

Using the assumption that $\|\phi(s,a)\|_\infty \le 1$, it holds that $\left|\sum_{s \in S} P_\pi(s,t)\phi(s,a)_k\right| \le 1$ for any policy π. We can therefore bound $|\alpha_{ik}|$ as follows.

$$\begin{aligned}
|\alpha_{ik}| &= \sum_{t=0}^{\infty} \gamma^t \left| \sum_{s \in S} \left[-P_{\pi^\star}(s,t)\phi(s, \pi^\star(s))_k + P_{\pi_i}(s,t)\phi(s, \pi_i(s))_k \right] \right| \\
&\le \sum_{t=0}^{\infty} \gamma^t \left[\left| \sum_{s \in S} P_{\pi^\star}(s,t)\phi(s, \pi^\star(s))_k \right| + \left| \sum_{s \in S} P_{\pi_i}(s,t)\phi(s, \pi_i(s))_k \right| \right] \\
&\le \frac{2}{1-\gamma}.
\end{aligned}$$

Therefore, it holds that

$$\|\boldsymbol{\alpha}_i\|_2 = \sqrt{\sum_{k=1}^{d} \alpha_{ik}^2} \le \sqrt{\sum_{k=1}^{d} \left(\frac{2}{1-\gamma} \right)^2} = \frac{2\sqrt{d}}{(1-\gamma)}.$$

Using this bound along with Equation (10.9), we obtain

$$\begin{aligned}
R^{\mathbf{w}^\star}(\pi^\star) - R^{\mathbf{w}^\star}(\pi_i) &\le \sum_{k=1}^{d} \eta_{ik}\alpha_{ik} \le \|\boldsymbol{\eta}_i\|_2 \|\boldsymbol{\alpha}_i\|_2 \le \frac{2\sqrt{d}}{(1-\gamma)}\sqrt{\sum_{k=1}^{d}\eta_{ik}^2} \\
&= \frac{2d}{(1-\gamma)}\sqrt{\frac{1}{d}\sum_{k=1}^{d}\eta_{ik}^2}.
\end{aligned}$$

Denote $u = \mathbb{E}[\frac{1}{d}\sum_{k=1}^{d}\eta_{ik}^2]$. To compute the expected value of the previous expression (with respect to the randomness of the noise $\boldsymbol{\eta}_i$), we analyze

$$\begin{aligned}
\mathbb{E}\left[\sqrt{\frac{1}{d}\sum_{k=1}^{d}\eta_{ik}^2}\right] &= \int_0^\infty \Pr\left[\sqrt{\frac{1}{d}\sum_{k=1}^{d}\eta_{ik}^2} \ge x\right] dx = \int_0^\infty \Pr\left[\frac{1}{d}\sum_{k=1}^{d}\eta_{ik}^2 \ge x^2\right] dx \\
&= \int_0^{\sqrt{u}} \Pr\left[\frac{1}{d}\sum_{k=1}^{d}\eta_{ik}^2 \ge x^2\right] dx + \int_{\sqrt{u}}^\infty \Pr\left[\frac{1}{d}\sum_{k=1}^{d}\eta_{ik}^2 \ge x^2\right] dx \\
&\le \int_0^{\sqrt{u}} 1\, dx + \int_{\sqrt{u}}^\infty \Pr\left[\frac{1}{d}\sum_{k=1}^{d}\eta_{ik}^2 \ge x^2\right] dx \\
&= \sqrt{u} + \int_0^\infty \Pr\left[\frac{1}{d}\sum_{k=1}^{d}\eta_{ik}^2 \ge u+t\right] \frac{1}{2\sqrt{u+t}} dt \\
&\le \sqrt{u} + \frac{1}{2\sqrt{u}}\int_0^\infty \Pr\left[\frac{1}{d}\sum_{k=1}^{d}\eta_{ik}^2 \ge u+t\right] dt,
\end{aligned}$$

where the fourth transition is obtained by changing the variable using $x = \sqrt{u+t}$. But since each η_{ik}^2 is sub-exponential with parameters (ν, b), from Lemma 78 we have

$$\Pr\left[\frac{1}{d}\sum_{k=1}^{d}\eta_{ik}^2 \ge u+t\right] \le \begin{cases} \exp\left(-\frac{dt^2}{2\nu^2}\right) & \text{for } 0 \le t \le \frac{\nu^2}{b} \\ \exp\left(-\frac{dt}{2b}\right) & \text{for } t > \frac{\nu^2}{b} \end{cases}.$$

Plugging this into the upper bound for the expected value gives us

$$
\begin{aligned}
\mathbb{E}\left[\sqrt{\frac{1}{d}\sum_{k=1}^{d}\eta_{ik}^2}\right] &\leq \sqrt{u} + \frac{1}{2\sqrt{u}}\int_0^{\infty} \Pr\left[\frac{1}{d}\sum_{k=1}^{d}\eta_{ik}^2 \geq u + t\right] dt \\
&\leq \sqrt{u} + \frac{1}{2\sqrt{u}}\left[\int_0^{\frac{\nu^2}{b}} \exp\left(-\frac{dt^2}{2\nu^2}\right) dt + \int_{\frac{\nu^2}{b}}^{\infty} \exp\left(-\frac{dt}{2b}\right) dt\right] \\
&= \sqrt{u} + \frac{1}{2\sqrt{u}}\left[\int_0^{\frac{\nu\sqrt{d}}{b}} \exp\left(-\frac{z^2}{2}\right) \frac{\nu}{\sqrt{d}} dz + \left(-\frac{2b}{d}\right) \exp\left(-\frac{dt}{2b}\right)\Bigg|_{\frac{\nu^2}{b}}^{\infty}\right] \\
&= \sqrt{u} + \frac{1}{2\sqrt{u}}\left[\sqrt{\frac{2\pi}{d}}\nu \int_0^{\frac{\nu\sqrt{d}}{b}} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) dz + \frac{2b}{d}\exp\left(-\frac{d\nu^2}{2b^2}\right)\right] \\
&= \sqrt{u} + \frac{1}{2\sqrt{u}}\left[\sqrt{\frac{2\pi}{d}}\nu \left(\Phi\left(\frac{\nu\sqrt{d}}{b}\right) - \frac{1}{2}\right) + \frac{2b}{d}\exp\left(-\frac{d\nu^2}{2b^2}\right)\right] \\
&= \sqrt{u} + \sqrt{\frac{\pi}{2ud}}\nu \left(\Phi\left(\frac{\nu\sqrt{d}}{b}\right) - \frac{1}{2}\right) + \frac{b}{d\sqrt{u}}\exp\left(-\frac{d\nu^2}{2b^2}\right),
\end{aligned}
$$

where the transition in the third line is obtained by changing the variable using $t = \frac{v}{\sqrt{d}}z$, and Φ denotes the CDF of a standard normal distribution. Hence, taking an expected value for Equation (10.9) and plugging in Equation (10.9), we obtain

$$
\mathbb{E}\left[R^{\mathbf{w}^\star}(\pi^\star) - R^{\mathbf{w}^\star}(\pi_i)\right] \leq \frac{2d}{(1-\gamma)}\left[\sqrt{u} + \sqrt{\frac{\pi}{2ud}}\nu \left(\Phi\left(\frac{\nu\sqrt{d}}{b}\right) - \frac{1}{2}\right) + \frac{b}{d\sqrt{u}}\exp\left(-\frac{d\nu^2}{2b^2}\right)\right].
$$

Rearranging this equation, we have

$$
\mathbb{E}\left[R^{\mathbf{w}^\star}(\pi_i)\right] \geq R^{\mathbf{w}^\star}(\pi^\star) - \frac{2d}{(1-\gamma)}\left[\sqrt{u} + \sqrt{\frac{\pi}{2ud}}\nu \left(\Phi\left(\frac{\nu\sqrt{d}}{b}\right) - \frac{1}{2}\right) + \frac{b}{d\sqrt{u}}\exp\left(-\frac{d\nu^2}{2b^2}\right)\right].
$$

Taking an expectation over Equation (10.5) gives us $\mathbb{E}\left[R^{\mathbf{w}^\star}(\pi^u)\right] = \mathbb{E}\left[R^{\mathbf{w}^\star}(\pi_i)\right]$, and the theorem directly follows. □

We remark that Theorem 61 can easily be strengthened to obtain a high probability result (at the cost of complicating its statement). Indeed, the reward of the uniform mixture $R^{\mathbf{w}^\star}(\pi^u)$ is the average of the individual policy rewards $R^{\mathbf{w}^\star}(\pi_i)$, which are i.i.d. Further, each of these rewards is bounded, because of the constraints on $\mathbf{w}^\star$ and ϕ. Hence, Hoeffding's inequality would show that $R^{\mathbf{w}^\star}(\pi^u)$ strongly concentrates around its mean.

10.10 Example for the Tightness of Theorem 61

Assume $\eta_{ik} \sim \mathcal{N}(0, \sigma^2)$ with $\sigma \leq 2/d$ (to avoid violating the constraint $\|\phi(s,a)\|_\infty \leq 1$). Suppose the MDP has just one state and $2^{d-1}+1$ actions. One action has feature vector $(d\sigma/2, 0, \dots, 0)$, and for each subset $S \subseteq \{2, \dots, d\}$, there is an action a_S with a binary feature vector such that it is 1 for coordinates in S and 0 everywhere else. Let $w^\star = (1, 0, ..., 0)$. The optimal policy is to pick the first action which has cumulative reward of $\frac{d\sigma}{2(1-\gamma)}$. As $\eta_{ik} \sim \mathcal{N}(0, \sigma^2)$ for each k, with constant probability, roughly $d/2$ of the coordinates of the noised vector reward $\mathbf{w}_i$ will deviate by roughly $+\sigma$ and the first coordinate will not increase too much. In this case, the action corresponding to the coordinates with positive deviations will have reward on the order of $d\sigma/2$, beating action 1 to become optimal. Hence, this would lead to π_i picking this action and having 0 reward under $\mathbf{w}^\star$. As this occurs with constant probability for a policy in the data, and π^u is simply a mean of their rewards, its expected value would deviate from the optimum by at least a constant fraction of $d\sigma/2$.

10.11 Empirical Results for the MDP setting

As we have seen in Section 10.4.1, the gap between $R^{\mathbf{w}^\star}(\pi^\star)$ and $R^{\mathbf{w}^\star}(\pi^u)$ is upper bounded by $O(d\sqrt{u} + \nu\sqrt{d/u} + b/\sqrt{u})$ when η_{ik}^2 is sub-exponential, or $O(d\sigma)$ when η_{ik} is Gaussian. Further, Section 10.3 shows that a policy $\tilde{\pi}$ that matches feature expectations of the observed trajectories is very close to π^u in terms of cumulative reward $R^{\mathbf{w}^\star}$. In this appendix, we empirically examine the gaps between $\tilde{\pi}$ (obtained by a "feature matching" IRL algorithm), π^u and $\pi^\star$.

10.11.1 Methodology

As our IRL algorithm we use Apprenticeship Learning, which guarantees the feature-matching property (see Section 10.3 and Appendix 10.7). By Theorem 60 we may safely assume that any IRL algorithm that matches feature expectations would have essentially identical rewards, and therefore would show very similar behavior in our experiments.

We perform our experiments in the following two domains.

Grab a Milk. We adapt the "Grab a Milk" MDP, a route planning RL domain [298], to our setting. The MDP is defined by a 10 by 10 grid room, where the agent starts at $(0,0)$ and has to reach a bottle of milk positioned at $(9,9)$. There are also 16 babies in the room, 5 of which are crying for attention. When the agent crosses a crying baby, they can help soothe the baby, but on crossing a non-crying baby, the agent disturbs the baby. Hence, the goal of this task is to minimize the number of steps to the milk, while at the same time soothing as many crying babies as possible along the way and avoiding crossing non-crying babies. This MDP is adapted to our setting, by defining each state (or grid square) to have three features $\phi(s)$.[8] The first feature captures the reward of taking a step, and is set to -1 if the state is non-terminal, whereas it is set to 5 for the terminal state $(9,9)$. The second is a boolean feature depicting whether there is a crying baby

[8] For these MDPs, the rewards depend only on the states and not state-action pairs, and hence the reward function can be defined as $r^{\mathbf{w}}(s,a) = r^{\mathbf{w}}(s) = \mathbf{w}^\intercal \phi(s)$.

in the particular grid square, and similarly the third is a boolean feature depicting whether there is a non-crying baby in the particular grid square. The rewards in the MDP are then defined as $r^{\mathbf{w}^\star}(s) = (\mathbf{w}^\star)^\intercal \phi(s)$ where the ground truth weight vector is given by $\mathbf{w}^\star = [1, 0.5, -0.5]$. Intuitively, this weight vector $\mathbf{w}^\star$ can be interpreted as the weights for different ethical factors, and each member of society has a noised version of this weight.

Sailing. The other domain we use is a modified version of the "Sailing" MDP [165]. The Sailing MDP is also a gridworld domain (we use the same size of 10 by 10), where there is a sailboat starting at $(0, 0)$ and navigating the grid under fluctuating wind conditions. The goal of the MDP is to reach a specified grid square as quickly as possible. We adapt this domain to our setting by removing the terminal state, and instead adding features for each grid square.[9] Now, the goal of the agent is not to reach a certain point as quickly as possible, but to navigate this grid while maximizing (or minimizing) the weighted sum of these features. We use 10 features for each grid square, and these are independently sampled from a uniform distribution over $(-1, 1)$. The ground truth weight vector $\mathbf{w}^\star$, which defines the weights of these features for the net reward, is also randomly sampled from independent Unif$(-1, 1)$ for each coordinate. As before, this weight vector $\mathbf{w}^\star$ can be interpreted as the weights for different bounties, and each member has a noised version of this weight.

Being gridworld domains, in both the MDPs, the agent has four actions to choose from at each state (one for each direction). The transition dynamics are as follows: On taking a particular action from a given state, the agent moves in that direction with probability 0.95, but with a probability of 0.05 it moves in a different direction uniformly at random. We use a discount factor of 0.95 in both domains.

We generate the trajectories $\{\tau_1, \ldots, \tau_n\}$ as described in Section 10.3, and use a Gaussian distribution for the noise. That is, $\boldsymbol{\eta}_i \sim \mathcal{N}(0, \sigma^2 I_d)$. We generate a total of $n = 50$ trajectories, each of length $L = 30$. IRL is then performed on this data and we analyze its reward as σ is varied. A learning rate of 0.001 is used for the Apprenticeship Learning algorithm.

10.11.2 Results

Figures 10.4 and 10.5 show the performance of π^u and the IRL algorithm as σ is varied. We also include the performance of $\pi^\star$ and a purely random policy π^r (which picks a uniformly random action at each step), as references. Each point in these graphs is averaged over 50 runs (of data generation).

For both domains, the first thing to note is that the uniform mixture π^u and the IRL algorithm have nearly identical rewards, which is why the green IRL curve is almost invisible. This confirms that matching feature expectations leads to performance approximating the uniform mixture.

Next, as expected, one can observe that as σ increases, the gap between $R^\star(\pi^\star)$ and $R^\star(\pi^u)$ also increases. Further, for both domains, this gap saturates around $\sigma = 10$ and the $R^\star(\pi^u)$ curve flattens from there (hence, we do not include larger values of σ in either graph). Note that, in both domains, the ground truth weight vector $\mathbf{w}^\star$ is generated such that $\|\mathbf{w}^\star\|_\infty \leq 1$. Hence, a standard deviation of 10 in the noise overshadows the true weight vector $\mathbf{w}^\star$, leading to the large

[9] Intuitively, these features could represent aspects like "abundance of fish" in that grid square for fishing, "amount of trash" in that square that could be cleaned up, "possible treasure" for treasure hunting, etc.

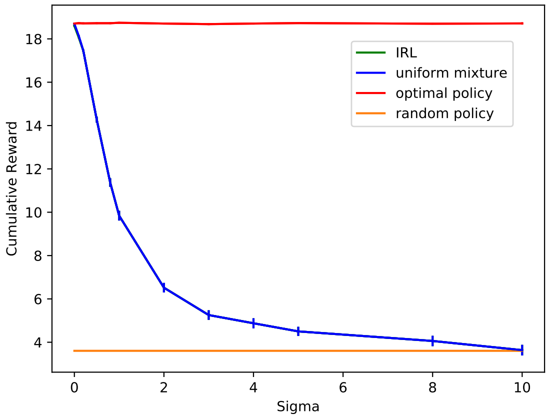


Figure 10.4: Performance on the Sailing MDP. Error bars show 95% confidence intervals.

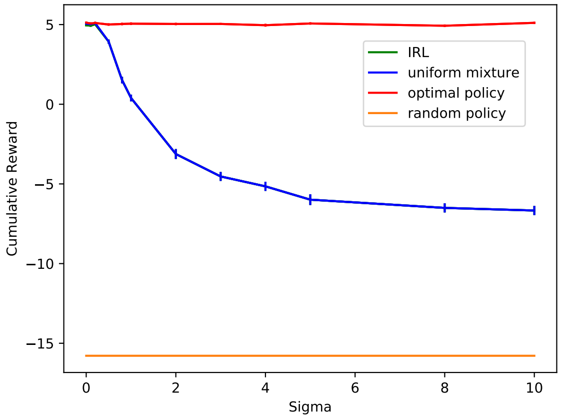


Figure 10.5: Performance on the Grab a Milk MDP. Error bars show 95% confidence intervals.

gap shown in both graphs. Looking at more reasonable levels of noise (with respect to the norm of the weights), like $\sigma \in [0, 1]$, we can see that $R^\star(\pi^u)$ drops approximately linearly, as suggested by Theorem 61. In particular, it is 14.27 at $\sigma = 0.5$ and 9.84 at $\sigma = 1.0$ for Sailing, and it is 3.93 at $\sigma = 0.5$ and 0.39 at $\sigma = 1.0$ for Grab a Milk.

Finally, we compare the performance of π^u with that of the purely random policy π^r. As σ becomes very large, each $\mathbf{w}_i$ is distributed almost identically across the coordinates. Nevertheless, because of the structure of the Grab a Milk MDP, $R^\star(\pi^u)$ still does significantly better than $R^\star(\pi^r)$. By contrast, Sailing has features that are sampled i.i.d. from Unif$(-1, 1)$ for each state, which leads the two policies, π^u and π^r, to perform similarly for large values of σ.

10.12 Proof of Lemma 77

Before proving the lemma, we look at a relatively simple example that we will use later to complete the proof.

10.12.1 Simpler Example

Consider an MDP with a single state s, and three actions $\{a, b, c\}$. Since s is the only state, $T(s, a, s) = T(s, b, s) = T(s, c, s) = 1$, and D is degenerate at s. This implies that there are only three possible policies, denoted by π_a, π_b, π_c (which take actions a, b, c respectively from s). Let the feature expectations be

$$\phi(s, a) = [0.5, 0.5],$$
$$\phi(s, b) = [1, -\delta/2],$$
$$\phi(s, c) = [-\delta/2, 1],$$

where $\delta > 0$ is a parameter. Hence, the feature expectations of the policies $\{\pi_a, \pi_b, \pi_c\}$ are respectively

$$\begin{aligned}
\mu_a &= \frac{1}{2(1-\gamma)}[1, 1], \\
\mu_b &= \frac{1}{2(1-\gamma)}[2, -\delta], \\
\mu_c &= \frac{1}{2(1-\gamma)}[-\delta, 2].
\end{aligned}$$

Let the ground truth weight vector be $\mathbf{w}^\star = (v_o, v_o)$, where v_o is a "large enough" positive constant. In particular, v_o is such that the noised weight vector $\mathbf{w} = \mathbf{w}^\star + \boldsymbol{\eta}$ has probability strictly more than $1/3$ of lying in the first quadrant. For concreteness, set v_o to be such that $\Pr(\mathbf{w} > 0) = 1/2$. Such a point always exists for any noise distribution (that is continuous and i.i.d. across coordinates). Specifically, it is attained at $v_o = -F^{-1}(1 - \frac{1}{\sqrt{2}})$, where F^{-1} is the inverse CDF of each coordinate of the noise distribution. This is because at this value of v_o,

$$\begin{aligned}
\Pr(\mathbf{w} > 0) &= \Pr((v_o, v_o) + (\eta_1, \eta_2) > 0) = \Pr(v_o + \eta_1 > 0)^2 \\
&= \Pr(\eta_1 > -v_o)^2 = \big(1 - F(-v_o)\big)^2 = \left(\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{2}.
\end{aligned}$$

Let us look at weight vectors $\mathbf{w}$ for which each of the three policies π_a, π_b and π_c are optimal. π_a is the optimal policy when $\mathbf{w}^\intercal\mu_a > \mathbf{w}^\intercal\mu_b$ and $\mathbf{w}^\intercal\mu_a > \mathbf{w}^\intercal\mu_c$, which is the intersection of the half-spaces $\mathbf{w}^\intercal(-1, 1+\delta) > 0$ and $\mathbf{w}^\intercal(1+\delta, -1) > 0$. On the other hand, π_b is optimal when $\mathbf{w}^\intercal\mu_b > \mathbf{w}^\intercal\mu_a$ and $\mathbf{w}^\intercal\mu_b > \mathbf{w}^\intercal\mu_c$, which is the intersection of the half-spaces $\mathbf{w}^\intercal(-1, 1+\delta) < 0$ and $\mathbf{w}^\intercal(1, -1) > 0$. Finally, π_c is optimal when $\mathbf{w}^\intercal\mu_c > \mathbf{w}^\intercal\mu_a$ and $\mathbf{w}^\intercal\mu_c > \mathbf{w}^\intercal\mu_b$, which is the intersection of the half-spaces $\mathbf{w}^\intercal(1+\delta, -1) < 0$ and $\mathbf{w}^\intercal(1, -1) < 0$. These regions are illustrated in Figure 10.1 for different values of δ. Informally, as δ is decreased, the lines separating (π_a, π_c) and (π_a, π_b) move closer to each other (as shown for $\delta = 0.25$), while as δ is increased, these lines move away from each other (as shown for $\delta = 10$).

Formally, let R_δ denote the region of $\mathbf{w}$ for which π_a is optimal (i.e. the blue region in the figures), that is,

$$R_\delta = \left\{ \mathbf{w} : \frac{w_1}{1+\delta} < w_2 < w_1(1+\delta) \right\}.$$

This is bounded below by the line $w_1 = (1+\delta)w_2$, which makes an angle of $\theta_\delta = \text{Tan}^{-1}(\frac{1}{1+\delta})$ with the x-axis, and bounded above by the line $w_2 = (1+\delta)w_1$, which makes an angle of θ_δ with the y-axis. We first show that for any value of δ, the regions of π_b and π_c have the exact same probability. The probability that π_b is optimal is the probability of the orange region which is

$$\begin{aligned}
\Pr(\pi_b \text{ is optimal}) &= \int_{-\infty}^{0}\int_{-\infty}^{w_1} \Pr(\mathbf{w})dw_2dw_1 + \int_{0}^{\infty}\int_{-\infty}^{\frac{w_1}{(1+\delta)}} \Pr(\mathbf{w})dw_2dw_1 \\
&= \int_{-\infty}^{0}\int_{-\infty}^{t_2} \Pr(t_2, t_1)dt_1dt_2 + \int_{0}^{\infty}\int_{-\infty}^{\frac{t_2}{(1+\delta)}} \Pr(t_2, t_1)dt_1dt_2 \\
&= \int_{-\infty}^{0}\int_{-\infty}^{t_2} \Pr(t_1, t_2)dt_1dt_2 + \int_{0}^{\infty}\int_{-\infty}^{\frac{t_2}{(1+\delta)}} \Pr(t_1, t_2)dt_1dt_2 \\
&= \Pr(\pi_c \text{ is optimal}),
\end{aligned}$$

where the second equality holds by changing the variables as $t_1 = w_2$ and $t_2 = w_1$, and the third one holds because the noise distribution is i.i.d. across the coordinates. Hence, we have

$$\Pr(\pi_b \text{ is optimal}) = \Pr(\pi_c \text{ is optimal}) = \frac{1 - \Pr(R_\delta)}{2},$$

as R_δ denotes the region where π_a is optimal.

Finally, we show that there exists a value of δ such that $\Pr(R_\delta) = 1/3$. Observe that as $\delta \to 0$, the lines bounding the region R_δ make angles that approach $Tan^{-1}(1) = \pi/4$ and the two lines touch, causing the region to have zero probability. On the other hand, as $\delta \to \infty$, the angles these lines make approach $Tan^{-1}(0) = 0$, so the region coincides with the first quadrant in the limit. Based on our selection of v_o, the probability of this region is exactly $1/2$. Hence, as δ varies from 0 to ∞, the probability of the region R_δ changes from 0 to $1/2$. Next, note that as $\theta_\delta = \text{Tan}^{-1}(\frac{1}{1+\delta})$, this angle changes continuously as δ changes, and hence does the region R_δ. Finally, as the noise distribution is continuous, the probability of this region R_δ also changes continuously as δ is varied. That is, $\lim_{\epsilon\to 0} \Pr(R_{\delta+\epsilon}) = \Pr(R_\delta)$. Coupling this with the fact that $\Pr(R_\delta)$ changes from 0 to $1/2$ as δ changes from 0 to ∞, it follows that there exists a value of δ in between such that $\Pr(R_\delta)$ is exactly $1/3$. Denote this value of δ by δ_o.

We conclude that for $\mathbf{w}^\star = (v_o, v_o)$ and our MDP construction with $\delta = \delta_o$, $\mathcal{P}(\mathbf{w}^\star) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$.

10.12.2 Completing the Proof

Consider the same MDP as in Section 10.12.1. However, for this example, let the feature expectations be

$$\begin{aligned}
\phi(s, a) &= [0.5, 0.5\ \ ,\ -\delta_o/2, 1], \\
\phi(s, b) &= [1, -\delta_o/2,\ \ 0.5, 0.5], \\
\phi(s, c) &= [-\delta_o/2, 1,\ \ 1, -\delta_o/2],
\end{aligned}$$

where δ_o is as defined in Section 10.12.1. Hence, the feature expectations of the policies $\{\pi_a, \pi_b, \pi_c\}$ are respectively

$$\mu_a = \frac{1}{2(1-\gamma)}[1, 1 \quad, -\delta_o, 2],$$
$$\mu_b = \frac{1}{2(1-\gamma)}[2, -\delta_o, \ 1, 1],$$
$$\mu_c = \frac{1}{2(1-\gamma)}[-\delta_o, 2, \ 2, -\delta_o].$$

Consider two weight vectors $\mathbf{w}_a^\star = (v_o, v_o, 0, 0)$ and $\mathbf{w}_b^\star = (0, 0, v_o, v_o)$, where v_o is as defined in Section 10.12.1. Since $\mathbf{w}_a^\star$ completely discards the last two coordinates, it immediately follows from the example of Section 10.12.1 that $\mathcal{P}(\mathbf{w}_a^\star) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Similarly, the same analysis on the last two coordinates shows that $\mathcal{P}(\mathbf{w}_b^\star) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ as well. On the other hand, the optimal policy according to $\mathbf{w}_a^\star$ is π_a while the optimal policy according to $\mathbf{w}_b^\star$ is π_b. Hence, $\pi_a^\star \neq \pi_b^\star$, but we still have $\mathcal{P}(\mathbf{w}_a^\star) = \mathcal{P}(\mathbf{w}_b^\star)$, leading to non-identifiability. □

10.13 Proof of Theorem 62

The proof of this theorem strongly relies on Lemma 77 and the example used to prove it. Consider the MDP as in Section 10.12.2, but now with 6 features instead of just 4. In particular, let the feature expectations of the three policies be

$$\phi(s,a) = [0.5, 0.5 \quad, -\delta_o/2, 1, \ 1, -\delta_o/2],$$
$$\phi(s,b) = [1, -\delta_o/2, \ 0.5, 0.5 \quad, -\delta_o/2, 1],$$
$$\phi(s,c) = [-\delta_o/2, 1, \ 1, -\delta_o/2, \ 0.5, 0.5 \quad].$$

Hence, the feature expectations of the policies $\{\pi_a, \pi_b, \pi_c\}$ are respectively

$$\mu_a = \frac{1}{2(1-\gamma)}[1, 1 \quad, -\delta_o, 2, \ 2, -\delta_o],$$
$$\mu_b = \frac{1}{2(1-\gamma)}[2, -\delta_o, \ 1, 1 \quad, -\delta_o, 2],$$
$$\mu_c = \frac{1}{2(1-\gamma)}[-\delta_o, 2, \ 2, -\delta_o, \ 1, 1 \quad].$$

Consider three weight vectors

$$\mathbf{w}_a^\star = (v_o, v_o, 0, 0, 0, 0),$$
$$\mathbf{w}_b^\star = (0, 0, v_o, v_o, 0, 0),$$
$$\mathbf{w}_c^\star = (0, 0, 0, 0, v_o, v_o).$$

Since $\mathbf{w}_a^\star$ completely discards the last four coordinates, the example of Section 10.12.1 shows that $\mathcal{P}(\mathbf{w}_a^\star) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Similarly, the same analysis on the middle two and last two coordinates shows that $\mathcal{P}(\mathbf{w}_b^\star) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ and $\mathcal{P}(\mathbf{w}_c^\star) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, respectively. However, the optimal policy according to $\mathbf{w}_a^\star$ is π_a, according to $\mathbf{w}_b^\star$ it is π_b, and according to $\mathbf{w}_c^\star$ it is π_c.

Now, consider an arbitrary algorithm $\mathcal{A}$, which takes as input a distribution over policies and outputs a (possibly randomized) policy. Look at the randomized policy $\mathcal{A}(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ returned by $\mathcal{A}$ when the input is $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, and let p_a, p_b, p_c be the probabilities it assigns to playing π_a, π_b and π_c. Let p_i (where $i \in \{a, b, c\}$) denote the smallest probability among the three. Then, $p_i \leq 1/3$. Pick the ground truth weight vector to be $\mathbf{w}_i^\star$. As $\mathcal{P}(\mathbf{w}_a^\star) = \mathcal{P}(\mathbf{w}_b^\star) = \mathcal{P}(\mathbf{w}_c^\star)$, the data generated by $\mathbf{w}_i^\star$ follows the distribution $\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$, and the policy distribution chosen by $\mathcal{A}$ is simply (p_a, p_b, p_c).

Now, with probability $p_i \leq 1/3$, the policy played is π_i leading to a reward of $\mathbf{w}_i^{\star\intercal}\mu_i = \frac{v_o}{(1-\gamma)}$, and with probability $(1 - p_i)$, the policy played is some π_j (where $j \neq i$) leading to a reward of $\mathbf{w}_i^{\star\intercal}\mu_j = \frac{(2-\delta_o)}{2}\frac{v_o}{(1-\gamma)}$ (which is independent of the value of j).[10] Hence, the expected reward of algorithm $\mathcal{A}$ in this case is

$$\begin{aligned} p_i \cdot \frac{v_o}{(1-\gamma)} + (1-p_i) \cdot \frac{(2-\delta_o)}{2}\frac{v_o}{(1-\gamma)} &= \frac{(2-\delta_o)}{2}\frac{v_o}{(1-\gamma)} + p_i \cdot \frac{\delta_o}{2}\frac{v_o}{(1-\gamma)} \\ &\leq \frac{(2-\delta_o)v_o}{2(1-\gamma)} + \frac{\delta_o v_o}{6(1-\gamma)}. \end{aligned}$$

Observe that the uniform mixture π^u in this case is just the input distribution $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Whatever be the chosen $\mathbf{w}_i^\star$, the expected reward of this distribution is exactly

$$\frac{1}{3} \cdot \frac{v_o}{(1-\gamma)} + \frac{2}{3} \cdot \frac{(2-\delta_o)}{2}\frac{v_o}{(1-\gamma)} = \frac{(2-\delta_o)v_o}{2(1-\gamma)} + \frac{\delta_o v_o}{6(1-\gamma)},$$

which is nothing but the upper bound on the expected reward of $\mathcal{A}$. Hence, for any algorithm $\mathcal{A}$ there exists a ground truth weight vector $\mathbf{w}_i^\star$ such that $\mathcal{A}$ has an expected reward at most that of π^u (which in turn is strictly suboptimal). □

10.14 Proof of Theorem 63

To see that this problem is convex, let us analyze the distribution $\mathcal{Q}(\mathbf{w})$.

$$\begin{aligned} \mathcal{Q}(\mathbf{w})_k &= \Pr(\text{Arm } k \text{ is optimal under weight } (\mathbf{w} + \boldsymbol{\eta})) \\ &= \Pr((\mathbf{w} + \boldsymbol{\eta})^\intercal \mathbf{x}_k \geq (\mathbf{w} + \boldsymbol{\eta})^\intercal \mathbf{x}_j \text{ for all j}) \\ &= \Pr((\mathbf{w} + \boldsymbol{\eta})^\intercal (\mathbf{x}_k - \mathbf{x}_j) \geq 0 \text{ for all j}) \\ &= \Pr(X_k(\mathbf{w} + \boldsymbol{\eta}) \geq 0) \\ &= \Pr(-X_k\boldsymbol{\eta} \leq X_k\mathbf{w}). \end{aligned}$$

Since $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 I_d)$, we have

$$-X_k\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 X_k X_k^\intercal).$$

[10] An interesting point to note is that by carefully selecting v_o, one could get the corresponding δ_o to be arbitrarily large, thereby causing the optimal and suboptimal policies to have a much larger gap (equally affecting the uniform mixture π^u as well).

And since $X_k X_k^\intercal$ is invertible, this distribution is non-degenerate and has a PDF. Let us use F_k to denote its CDF. Equation (10.14) then reduces to $\mathcal{Q}(\mathbf{w})_k = F_k(X_k\mathbf{w})$. Plugging this back into our optimization problem (10.1), we have

$$\min_{\mathbf{w}} -\sum_{k\in A} \tilde{\mathcal{Q}}_k \log F_k(X_k\mathbf{w}). \tag{10.6}$$

As F_k corresponds to a (multivariate) Gaussian which has a log-concave PDF, this CDF is also log-concave. Hence, $\log F_k(X_k\mathbf{w})$ is concave in $\mathbf{w}$ for each k, and therefore (10.6) is a convex optimization problem. □

10.15 Gradient Calculation

From Equation (10.6), we know that the objective function of problem (10.1) can be rewritten as $f(\mathbf{w}) = -\sum_{k\in A} \tilde{\mathcal{Q}}_k \log F_k(X_k\mathbf{w})$. Taking the gradient with respect to $\mathbf{w}$, we have

$$\begin{aligned}
\nabla_{\mathbf{w}} f(\mathbf{w}) &= -\sum_{k\in A} \tilde{\mathcal{Q}}_k \nabla_{\mathbf{w}} \log F_k(X_k\mathbf{w}) \\
&= -\sum_{k\in A} \frac{\tilde{\mathcal{Q}}_k}{F_k(X_k\mathbf{w})} \nabla_{\mathbf{w}} F_k(X_k\mathbf{w}) \\
&= -\sum_{k\in A} \frac{\tilde{\mathcal{Q}}_k}{F_k(X_k\mathbf{w})} \left[\sum_{i=1}^{m-1} \left.\frac{\partial F_k(\mathbf{z})}{\partial z_i}\right|_{z=X_k\mathbf{w}} \cdot \nabla_{\mathbf{w}} (X_k\mathbf{w})_i \right] \\
&= -\sum_{k\in A} \frac{\tilde{\mathcal{Q}}_k}{F_k(X_k\mathbf{w})} \left[\sum_{i=1}^{m-1} \left.\frac{\partial F_k(\mathbf{z})}{\partial z_i}\right|_{z=X_k\mathbf{w}} \cdot X_k^{(i)} \right],
\end{aligned}$$

where the third equality holds as $F_k(\mathbf{z})$ has multidimensional input and we're taking the total derivative. Hence, we need to compute $\frac{\partial F_k(\mathbf{z})}{\partial z_i}$. Writing CDF F_k in terms of its PDF p_k (which exists as $X_k X_k^\intercal$ is invertible), we have

$$F_k(\mathbf{z}) = \int_{-\infty}^{z_1} \cdots \int_{-\infty}^{z_{m-1}} p_k(x_1, \ldots, x_{m-1}) dx_1 \ldots dx_{m-1}.$$

We compute partial derivative w.r.t. z_1 first, for simplicity, and generalize it after. In particular,

$$\begin{aligned}
\frac{\partial F_k(\mathbf{z})}{\partial z_1} &= \int_{-\infty}^{z_2} \cdots \int_{-\infty}^{z_{m-1}} \frac{\partial}{\partial z_1} \left[\int_{-\infty}^{z_1} p_k(x_1, \ldots, x_{m-1}) dx_1 \right] dx_2 \ldots dx_{m-1} \\
&= \int_{-\infty}^{z_2} \cdots \int_{-\infty}^{z_{m-1}} p_k(z_1, \ldots, x_{m-1}) dx_2 \ldots dx_{m-1} \\
&= \int_{-\infty}^{z_2} \cdots \int_{-\infty}^{z_{m-1}} p_{k,-1}(x_2, \ldots, x_{m-1} | z_1) p_{k,1}(z_1) dx_2 \ldots dx_{m-1} \\
&= p_{k,1}(z_1) \int_{-\infty}^{z_2} \cdots \int_{-\infty}^{z_{m-1}} p_{k,-1}(x_2, \ldots, x_{m-1} | z_1) dx_2 \ldots dx_{m-1} \\
&= p_{k,1}(z_1) \cdot \Pr_k(Z_2 \leq z_2, \ldots, Z_{m-1} \leq z_{m-1} | Z_1 = z_1) \\
&= p_{k,1}(z_1) \cdot F_{k, Z_{-1}|Z_1 = z_1}(\mathbf{z}_{-1}),
\end{aligned}$$

where $F_{k,Z_{-1}|Z_1=z_1}$ is the conditional CDF of the distribution F_k given the first coordinate is z_1, $p_{k,1}$ is the marginal distribution PDF of this first coordinate, and $p_{k,-1}$ is the PDF of the rest. This derivation holds for the partial derivative w.r.t. any z_i, even though it was derived for z_1. Plugging this into Equation (10.15), the gradient therefore becomes

$$\nabla_{\mathbf{w}} f(\mathbf{w}) = -\sum_{k \in A} \frac{\tilde{Q}_k}{F_k(X_k \mathbf{w})} \left[\sum_{i=1}^{m-1} p_{k,i}((X_k \mathbf{w})_i) \cdot F_{k, Z_{-i}|Z_i = (X_k \mathbf{w})_i}((X_k \mathbf{w})_{-i}) \cdot X_k^{(i)} \right].$$

Note that the conditional distribution $F_{k,Z_{-i}|Z_i=z_i}$ is also a Gaussian distribution with known parameters, and hence it can be estimated efficiently. We conclude that we can use gradient descent updates defined by

$$\mathbf{w}^+ = \mathbf{w} + \alpha \sum_{k \in A} \frac{\tilde{Q}_k}{F_k(X_k \mathbf{w})} \left[\sum_{i=1}^{m-1} p_{k,i}((X_k \mathbf{w})_i) \cdot F_{k, Z_{-i}|Z_i = (X_k \mathbf{w})_i}((X_k \mathbf{w})_{-i}) \cdot X_k^{(i)} \right],$$

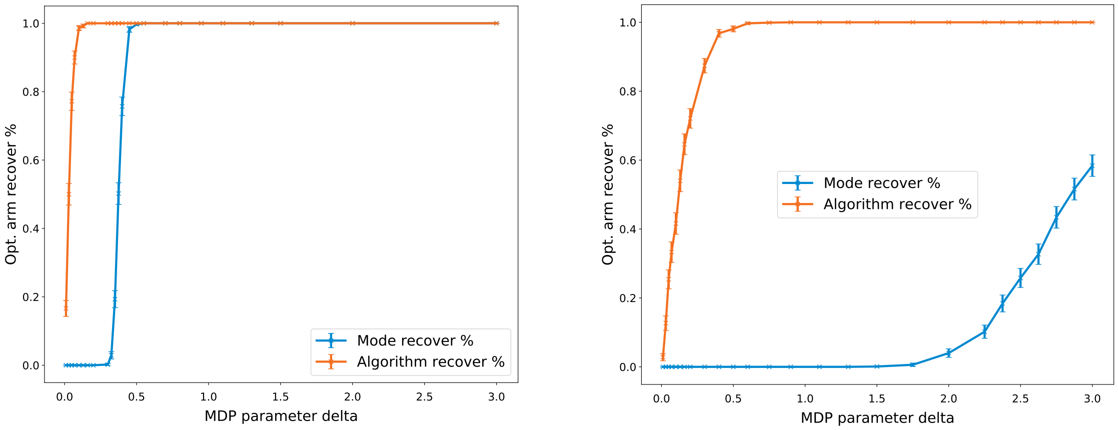
where α is a suitable step size, to find an optimal solution of (10.1).

10.16 Additional Empirical Results for Inverse Bandits

10.16.1 Varying parameter δ

Here, we present the experimental results as δ is varied for additional values of σ and n. All graphs in this section have also been averaged over 1000 runs, and error bars depict 95% confidence intervals. Figure 10.6 shows how the performance varies as δ is varied from 0.01 to 3, when σ is set to 0.5 and 2.0 (while n is still 500). As expected, one can observe that the tipping point (where the mode switches to the blue region corresponding to arm 1) occurs much earlier when $\sigma = 0.5$, and much later when $\sigma = 2$.

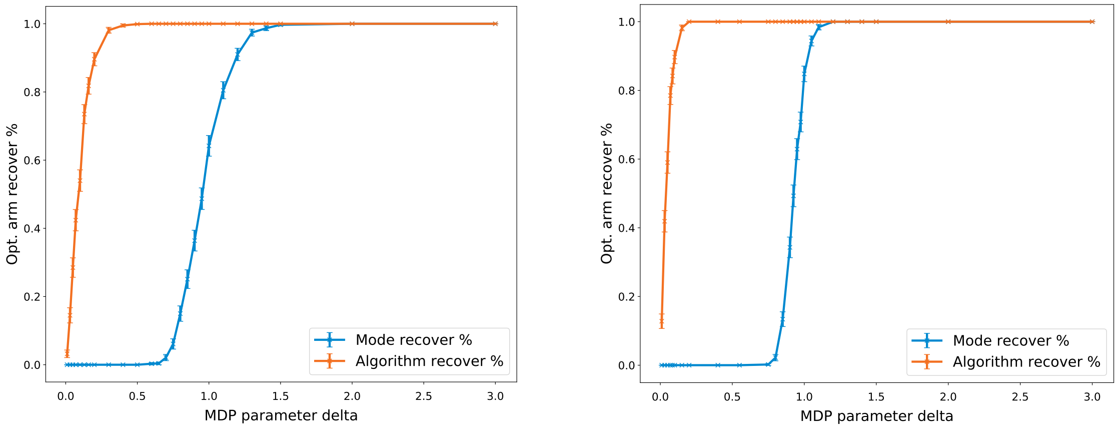
Figure 10.7 shows how the performance varies as δ is varied from 0.01 to 3, when the number of agents n is 250 and 1000 (while σ is still set to 1). First, note that the tipping point (for the



$\sigma = 0.5$

$\sigma = 2.0.$

Figure 10.6: Performance as δ is varied, when σ is fixed to 0.5 and 2.



$n = 250$

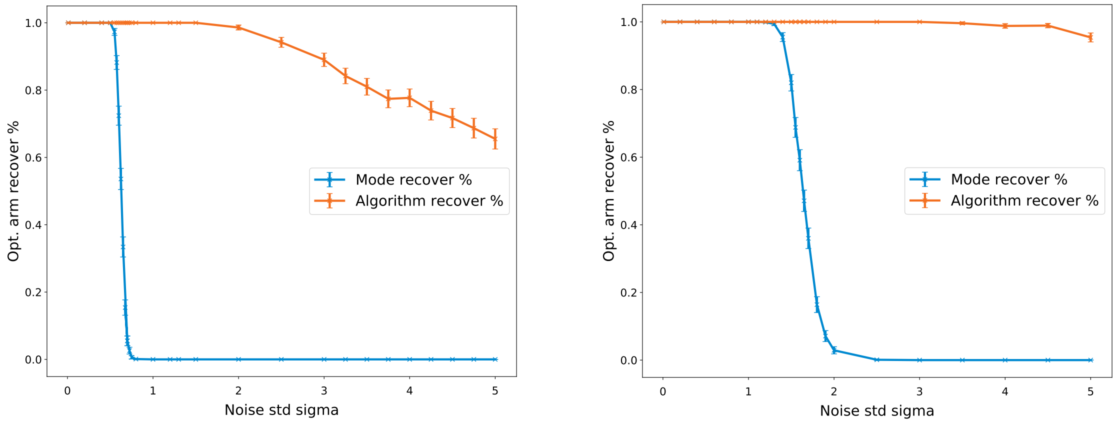
$n = 1000$

Figure 10.7: Performance as δ is varied, when the number of agents is 250 and 1000.

mode switch) only depends on the value of δ and σ, and indeed, we can see from the graphs that the tipping point continues to be around $\delta = 1$ irrespective of the number of the agents. But, the number of agents defines how close $\tilde{\mathcal{Q}}$ is to $\mathcal{Q}(\mathbf{w}^\star)$, and hence determines the sharpness of the transition. In particular, for a larger number of agents, the empirical mode (obtained from $\tilde{\mathcal{Q}}$) is more likely to match the true mode (of $\mathcal{Q}(\mathbf{w}^\star)$). Hence, we can see that when $n = 1000$, the transition of the mode's performance is sharper across the tipping point (because of less noise), while when $n = 250$, the transition is smoother across this tipping point (because of more noise).

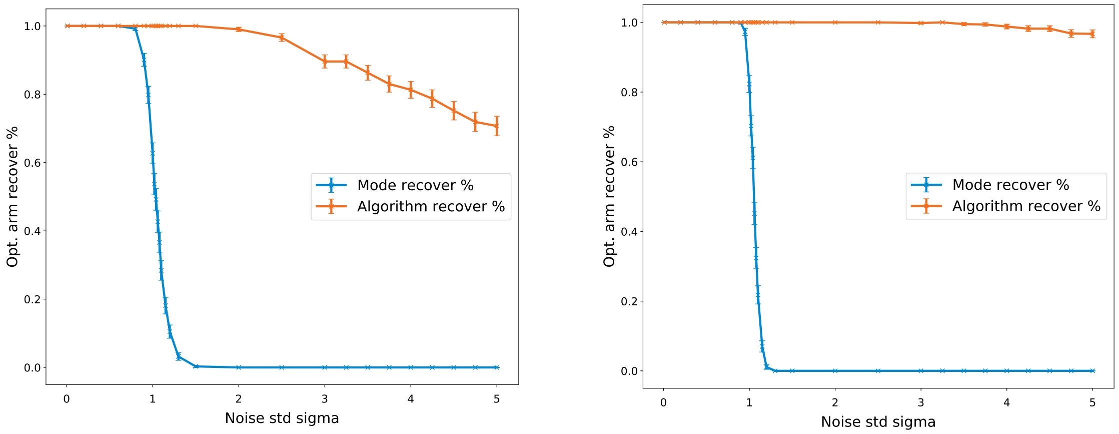
10.16.2 Varying noise parameter σ

Next, we present the experimental results as σ is varied, for additional values of δ and n. All graphs in this section have also been averaged over 1000 runs, and error bars depict 95% confidence intervals. Figure 10.8 shows how the performance varies as σ is varied from 0.01 to 5, when δ is set to 0.5 and 2.0 (while n is still 500). As expected, we can see that the tipping point (where the



$\delta = 0.5$ $\delta = 2.0.$

Figure 10.8: Performance as σ is varied, when δ is fixed to 0.5 and 2.



$n = 250$ $n = 1000$

Figure 10.9: Performance as σ is varied, when the number of agents is 250 and 1000.

mode switches out of the blue region corresponding to arm 1) occurs earlier when $\delta = 0.5$, and much later when $\delta = 2$. Further, at high values of σ, the algorithm's performance is more robust when $\delta = 2$, as the blue region is larger.

Finally, Figure 10.9 shows how the performance varies as σ is varied from 0.01 to 5, when number of agents n is 250 and 1000 (while δ is still set to 1). Again, note that the tipping point of the mode switch occurs at the same point (around $\sigma = 1$) irrespective of the number of agents. And, as Section 10.16.1, when $n = 1000$, the transition of the mode's performance is sharper across the tipping point, while when $n = 250$, the transition is smoother across it. Further, at high values of σ, $n = 1000$ has a much better algorithm performance compared to $n = 500$ (which in turn outperforms that at $n = 250$), showing that even at such high levels of noise, if $\tilde{\mathcal{Q}}$ coincides with $\mathcal{Q}(\mathbf{w}^{\star})$, the algorithm is still able to recover the optimal arm 1.

10.17 Relationship to Social Choice/Welfare

The framework we introduce is also closely related to varying notions of social choice or welfare. One may wonder how one could work varying notions of social choice into this framework. Specifically, suppose that we have learned (via IRL) a reward function and an optimal policy for each agent. Note that this would require a significant amount of data for each agent. Still, how should these policies be aggregated into a single policy? One may cast this as a problem of allocating public goods. A naïve approach would compute each agent's reward for each possible policy, and choose the policy that, say, maximizes social welfare notions such as the Nash social welfare [104]; but this is a pipe dream, due to seemingly insurmountable computational barriers. We believe the discovery of *computationally tractable* methods for this policy aggregation problem may provide attractive alternatives to the approach presented in this paper.

Chapter 11

Learning Multi-agent Hierarchical Systems

11.1 Introduction

Next-generation AI models are poised to produce sophisticated outputs such as long-form texts and videos, and execute complex tasks as agents. To build these AIs responsibly, we need to better our understanding of scalable oversight: the ability to provide *scalable* human feedback to these complex models [11, 50, 71, 180]. An immediate, key challenge to overcome is the size of model outputs, making it time-consuming for humans to parse and provide reliable feedback on, even with AI-assistance [247, 267, 296]. To this end, in this work, we consider human labelers with bounded processing ability such that accurate feedback can only be provided for outputs below some threshold size. We are interested in answering the question:

> How can we scale this limited feedback to supervise a model with outputs *larger* than this limit?

Verily, this task is difficult without further assumptions. If the model output can only be assessed in its entirety, it is impossible for humans to provide reliable feedback. Thus, we investigate a natural architecture that gives us hope to overcome the limitation in feedback: *hierarchical* systems.

Indeed, hierarchical structure exists in many high-dimensional outputs of interest, including long-form texts (books made up of chapters), videos (movies made up of scenes) and code (main functions made up of helper functions). It reflects the way we humans produce many of our most complex creations.

Thus, in this chapter, we study hierarchical learning, wherein we replace a monolithic big model with many small models in a hierarchical multi-agent system. This is so that bounded human supervision can nevertheless be used to supervise the model. Specifically, we will consider learning in the goal-conditioned hierarchical reinforcement learning (HRL) setup and analyze how this can enable scalable oversight.

Goal-oriented RL is a popular approach that has seen sizable success in leveraging state space structure to overcome sparse rewards over long horizons [130, 185, 210]. Our aim in this work differs in using this as an entry-point into understanding how to scale up bounded human feedback to train hierarchical systems. We explore both the challenges of training a multi-model system (in place of one) and the numerous benefits of hierarchical systems, which include more efficient exploration, more efficient credit assignment and the nice property of enabled scalable oversight.

11.1.1 Preliminaries

We consider a finite-horizon, Markov Decision Process (MDP) $\mathcal{M} = \langle S, A, P, r, s_1, H \rangle$, with finite state space S, finite action space A, transition probability $P : S \times A \rightarrow \Delta(S)$, reward $r(s, a) : S \times A \rightarrow [0, 1]$ and finite horizon H. The learner interacts with $\mathcal{M}$ starting at state s_1 and the episode ends after $H = H_h H_l$ time-steps. In this work, policies are trained using human feedback. And so, we assume that a human supervisor is needed to evaluate and provide reward r for trajectories $\tau \sim \pi, P$ generated by policy $\pi : S \rightarrow A$.

Accompanying Example: Consider the task of learning to generate a long-form, argumentation essay. Providing feedback to an end-to-end policy is difficult as labelers would have to read through entire essays to rate the outputs, after which it may be difficult still to assign a single rating to the entire essay. A tractable alternative is to learn a hierarchical model, with a higher-level policy that generates the essay arguments (goals), and lower-level policies that flesh out these points (realize these goals). It would then be easier for the labeler to rate the shorter-length essay content, and also individual fleshed out arguments, in order to generate a rating on the whole. This approach also mirrors existing rubrics for scoring essays [1].

Bounded Feedback: To formalize the difficulty of human supervisors assessing long-form outputs, we assume that reliable feedback can only be provided for trajectories of length at most $\max(H_h, H_l)$. In particular, this means that for the global policy $\pi : S \rightarrow A$, it is infeasible to obtain reliable feedback for its trajectory $\tau \sim \pi, P$, as $|\tau| = H_h H_l$. This thus motivates hierarchical learning, which makes possible the acquisition of reliable feedback in spite of bounded human supervision.

11.1.1.1 Goal-conditioned HRL

Since we are unable to learn a single, monolithic policy, our goal instead will be to learn a set of smaller policies that make up a hierarchical policy. This set consists of a high-level policy $\pi^h : S \rightarrow \Delta(A_h)$ (outputs a high-level action a^h at state $s \in S$), and a set of low-/sub-policies π^l_{s,a^h}: $S^l_{s,a^h} \rightarrow \Delta(A)$, where $S^l_{s,a^h} \subseteq S$ is the set of all states reachable from s after H_l steps.

In a nutshell, the high-level policy designates goals by choosing high level actions. The low-level policies then aim to realize these goals, while also trying to achieve a high intermediate return. Importantly, both such policies act over a shorter horizon of at most $\max(H_h, H_l)$, making it amenable for human supervisors to evaluate.

Goal Function: in the goal-conditioned HRL setting, we assume access to a function g mapping high-level action a^h at state s to a goal-state $g(s, a^h) \in S^l_{s,a^h}$. For example, s is the current content of the essay, a^h is the action (in natural language) "add an argument using X" and $g(s, a^h)$ is the content of the essay with the "argument using X" included.

Goal-conditioned sub-MDP: Given a high level action a^h at state s, this defines the sub-MDP $M(s, a^h)$, which has state space $S^l_{s,a^h} \subseteq S$, action space A (action space of the original $\mathcal{M}$), transition probabilities P restricted to S^l_{s,a^h}, starting state s and finite horizon H^l. The sub-MDP reward r^l will be defined later and as we will see, an apt choice is important for achieving sublinear regret.

High-level MDP: Given a set of low-level policies, π^h may be thought of as operating over a high-level MDP with state space S, action space A^h, starting state s_1 and finite horizon H^h. Importantly, the high-level transition P' of this MDP is a function of the current set of low-level policies $\Pr'(s'|s, a^h) = \Pr(s_{H_l}^{\pi_{s,a^h}} = s')$, which denotes the distribution over the (final) H_lth state that π^l_{s,a^h} reaches. Furthermore, the high-level reward $r^h(s, a^h) = \mathbb{E}_{s_j,a_j\sim\pi_{s,a^h},P}[\sum_{j=1}^{H_l} r(s_j, a_j)|s_1 = s]$ corresponds to the intermediate return of sub-policy π_{s,a^h} in $M(s, a^h)$. Altogether, this gives rise to a key complication in hierarchical learning. This is that both the transitions and rewards in the high-level MDP are non-stationary, as sub-policies π_{s,a^h} are updated over time.

Interaction Protocol: At each time-step t, the high level policy chooses a high level action a_t based on current state s_t. This defines the sub-goal state $g(s_t, a_t)$, along with the corresponding sub-MDP $M(s_t, a_t)$ with finite-horizon H_l, in which sub-policy $\pi^l_{s_t,a_t}$ is used to try to achieve the goal. The overall return of the high level policy π^h and low-level policies $\left\{\pi^l_{s,a}\right\}_{s,a\in S\times A^h}$ is the sum of intermediate returns $r(\pi^l_{s_t,a_t})$ incurred:

$$V^{\pi^h,\pi^l}(s_1) = \mathbb{E}_{a_t\sim\pi^h(s_t),s_{t+1}\sim\Pr(s_{H_l}^{\pi^l_{s_t,a_t}})}[\sum_{t=1}^{H_h} r(\pi^l_{s_t,a_t})|s_{t=1} = s_1].$$

Instantiation in the example: returning to our example, for a cogent essay, the arguments need to be logically related and built on top of each other. This results in a sequential decision making problem corresponding to the one solved by the high level policy π^h. Given an argument $g(s, a^h)$ to flesh out, the low level policy π^l_{s,a^h} generates up to H_l words, whose content aims to realize this argument. Additionally, low-level policies can incur intermediate rewards (return) for eloquent diction and clear structure when fleshing out the argument, all of which add to the essay's persuasiveness.

11.1.1.2 Learning Task

Our aim is to learn a hierarchical policy, whose return is close to that of the optimal, goal-reaching hierarchical policy, which we define as follows. For brevity, from this point on, we will use a^h and a interchangeably to denote high level action.

Assumption 1 (Goal-Reachability). *In every sub-MDP $M(s, a)$, there exists a policy that achieves the goal $g(s, a)$ almost surely. That is,there exists at least one policy $\pi \in \Pi_{s,a}$ in the policy class $\Pi_{s,a}$ such that $\Pr(s^{\pi}_{H_l} = g(s, a)) = 1$.*

In other words, we assume that the goal function g is well-defined in that it designates goals that are feasible to reach from the starting state s (e.g. the argument can be successfully fleshed out in H_l words or less given the essay content thus far). To motivate this assumption, we note there that there are already many settings of interest, where we have prior knowledge of a good goal function. This is because we humans have often (and successfully) taken the hierarchical approach to build up to and produce these long-form creations. So we know what are good goals to set e.g. we write essays by first writing an outline of arguments, then expanding out each point in the outline. Indeed, this approach of explicitly encoding prior knowledge in the hierarchical learning algorithm has been done in both HRL literature (e.g. we know apriori mazes has hierarchical structure in that it consists of rooms [241]) and scalable oversight literature (e.g. we know that books consists of chapters [296]).

With this assumption, there exists constant C large enough such that if $\pi \in \operatorname{argmax}_{\pi \in \Pi_{s,a}} r(\pi) + C \cdot \Pr(s^{\pi}_{H_l} = g(s,a))$, then π is goal-reaching and $\Pr(s^{\pi}_{H_l} = g(s,a)) = 1$.

Definition 45. *Define optimal low-level policies as* $\pi^*_{s,a} \in \operatorname{argmax}_{\pi \in \Pi_{s,a}} r(\pi) + C \cdot \Pr(s^{\pi}_{H_l} = g(s,a))$. *Define optimal high-level policy as* $\pi^* = \operatorname{argmax}_{\pi \in \Pi^h} V^{\pi, \pi^*_{s,a}}(s_1)$.

In words, $\pi^*_{s,a}$ has the highest intermediate return of all goal-reaching policies. Now let π^* be the optimal high-level policy fixing each sub-MDP policy to be $\pi^*_{s,a}$.

Learning Goal: We wish to learn a set of near-optimal high- and low-level polices $(\pi, \{\pi_{s,a}\})$ such that: $V^{\pi^*, \pi^*_{s,a}}(s_1) - V^{\pi, \pi_{s,a}} \leq \epsilon$.

11.1.2 Takeaways

The broad takeaway from this chapter is that hierarchical structure, if it exists, can be provably used to scale up limited human supervision. That is:

> Hierarchical multi-agent systems can enable scalable oversight.

On a more technical level, this chapter studies the challenge of training a set of policies (agents) that work together to form the hierarchical policy (meta-agent). This is the more complicated problem we turn to solve when it is not feasible to train a monolithic policy, due to bounded human supervision. We thus consider learning in the goal-conditioned HRL setup, under both cardinal and ordinal feedback. A key insight that applies in both settings is that an apt sub-MDP reward design (a suitable penalty for non-goal reachability) is needed for bounding regret and controlling the exit state of learned low-level policies. This is so that learned sub-policies do not land at bad states with sizable probability. Doing so would then allow one to compose low-level policies together, and stabilize high-level policy learning in the high-level MDP. More specific takeaways for both types of feedback are as follows:

- Under cardinal feedback, we develop a novel no-regret learning, Algorithm 26, that jointly learns a high-level and a set of low-level policies. Notably, Algorithm 26 only requires low-level feedback. Our main structural result in this setting is that hierarhical RL reduces to multi-task, sub-MDP regret minimization. Thus, the regret from the low-level accumulates additively (instead of say multiplicatively) as speculated upon in [180].
- Under ordinal feedback, we develop a novel hierarchical experiment-design Algorithm 27, building off of existing work on experiment design in preferenced-based RL [317]. A key observation is that in the ordinal case, low-level feedback may not be sufficient and high-level feedback may be needed. This introduces complications in human supervision, as the high-level feedback would need to account for the *current* performance of sub-policies. To this end, we study two natural forms of feedback, requiring differing cognitive loads on the human supervisor. Through the experiment design algorithm we develop, we then analyze the differing sample complexity under the two types of feedback. Finally, we show that high-level feedback should not be used if low-level feedback is sufficient and one form of feedback, with higher cognitive load, leads to better sample complexity.

11.2 Related Works

HRL under cardinal rewards: There has been sizable interest in understanding of the sample complexity of HRL algorithms, which to our knowledge has thus focused on learning from cardinal rewards. On this subject, the two closest papers to that of ours are [241] and [291]. [241] studies goal-conditioned HRL with the key result being a sample complexity lower bound associated with a given hierarchical decomposition. On the upper-bound side, an algorithm (SHQL) is presented, albeit without theoretical guarantees. By contrast, our work presents a learning algorithm with provable guarantees, and further shows that learning in goal-conditioned HRL reduces to multi-task, sub-MDP regret minimization.

[291] studies HRL under the options framework, providing a model-based, Bayesian algorithm with access to a prior distribution over MDPs that is updated over time. It does not adaptively learn sub-policies based on observed returns, computing instead an option for every exit-profile and equivalence class at each time during model-based planning. By contrast, our work does not assume knowledge of the prior nor ability to update posteriors, and does adaptively explore sub-MDPs via the UCB principle. Additionally, [291] demonstrate that when the size of the set of exit ("bottleneck") states is small, learning is efficient. Our work shed further light on this insight by showing that under a suitable sub-MDP reward, we can induce a small set of exit states *with high probability*. Thus, even though the total number of possible exit-states may be high, this condition is sufficient for learning with sublinear-regret.

RL under ordinal rewards: There has also been considerable interest in bandits/RL from preferences [179, 217, 295, 299, 317, 325]. Following the demonstrated success of RLHF [27, 72, 220, 330], there has been great interest in studying offline RL from preference feedback, and particularly experiment design for enhanced sample efficiency [317, 325]. Due to the success of RLHF in alignment, we also consider studying scalable oversight in this setup. Please see the Appendix 11.6 for further discussions on scalable oversight and goal-conditioned RL.

11.3 Learning from Cardinal Feedback

We begin by considering the setting when feedback is in the form of cardinal rewards. As noted before, in HRL, the high-level policy performance is dependent on the low-level policies performance. Thus, a naive approach is to learn near-optimal sub-policies in every sub-MDP $M(s, a)$, and then learn a high-level policy on top. However, a more sample efficient approach is to strategically explore sub-MDPs, and discover sub-policies with high intermediate returns in tandem with a high level policy that visits these "good" sub-MDPs. Please see the Appendix 11.8 for all the proofs. Note that in what follows, for brevity, theoretical statements will contain the phrase "with high probability" and the appendix will contain proofs that formalize this guarantee.

11.3.1 Sub-MDP reward design for Hier-UCB-VI

We are interested in adaptively learning the necessary sub-policies (the useful goals to achieve) and the associated high level policy that invokes these sub-policies. It is natural then to adopt an upper confidence bound approach and construct an exploration bonus that tracks the best/unexplored sub-MDPs. To this end, we develop an adaptation of the classic UCB-VI algorithm [18]. We highlight two key ingredients needed to construct the Hier-UCB-VI Algorithm 26.

Tradeoffs in sub-MDP reward design: Learned sub-policies in HRL have to tradeoff between two objectives. One is high intermediate returns $r(\pi_{s,a})$. The other is that exit-state; sub-policies should not land at "bad" states, as even if the intermediate return is high, $V(s_{H_l}^{\pi_{s,a}}) \approx 0$ means the return from hereon out (and hence the overall return) will be low. Thus, in sub-policy learning, we also need to consider the goodness of the exit-state. But how can we incentivize sub-policies to land at "good" states without being able to calculate V? Luckily, in the goal-conditioned setting, there is a natural answer for a "good" exit-state: $g(s, a)$.

To operationalize this, we design a sub-MDP reward that trades-off between intermediate sub-MDP return and goal-reachability. In sub-MDP $M(s, a)$, at time-step h, sub-MDP reward $r_{l,h}(s', a') = r(s', a') + \kappa \mathbb{1}(h = H_l \wedge s' = g(s, a))$. Crucially, here we set the weighting $\kappa = \max(2H_h H_l, C)$, which corresponds to an upper bound on the regret should we not reach the goal-state.

UCB construction: Next, we wish to obtain an UCB for $r(\pi^*_{s,a})$. Our main observation is that by using a no-regret subroutine for learning in $M(s, a)$, the regret guarantee directly translates to a UCB. Due to our choice of sub-MDP reward r_l, the UCB includes a penalty on non-goal reachability.

Lemma 79 (UCB implied by sub-MDP regret)**.** *Let $UB(\mathcal{R}^n(s, a))$ be an upper bound on sub-MDP $M(s, a)$'s cumulative regret after n rounds. Define $\beta = (\kappa + H_l)2\log(\frac{|\mathcal{C}(S,A)|H_h K}{\delta})$ and bonus,*

$$b_r^{s,a}(n) = \frac{UB(\mathcal{R}^n(s,a)) + \beta\sqrt{n}}{n} - \frac{\kappa}{n}\sum_{i=1}^{n} \mathbb{1}(s_{H_l}^{\pi^i_{s,a}} \neq g(s^h, a^h)).$$

Then, the average reward plus bonus $\bar{r}_n(s, a) + b_r^{s,a}(n)$ is an UCB for $r(\pi^*_{s,a})$ with high probability.

High-level MDP transition stabilization: An additional benefit of incentivizing goal-reachability is that we know the idealized transition probability in the high-level MDP. As mentioned before, another key difficulty with HRL is that the empirically estimated transitions in the high-level MDP drifts over time. In our algorithm, the key stabilization approach is to avoid estimation and set the transition in the upper bound Q_i to be the idealized transition ($g(s, a)$ w.p. 1). This allows us to prove our regret guarantee as described below.

11.3.2 Regret Analysis of Hier-UCB-VI

We start with a definition on clusters of equivalent sub-MDPs. Let there be $\mathcal{C}(S, A^h)$ such clusters. In the most general setting, it is not known apriori if there is any shared structure, in which case each sub-MDP will simply be its own cluster.

Definition 46 (Equivalent sub-MDPs [291])**.** *Two subMDPs $M(s, a)$ and $M(s', a')$ are equivalent if there is a bijection $\mathcal{F}$ between state space, and through $\mathcal{F}$, the subMDPs have the same transition probabilities and rewards.*

Our main structural result is that HRL regret decomposes to multi-task, sub-MDP regret in the cardinal reward setting. This has the implication that only low-level feedback is needed for regret

Algorithm 26 Hierarchical-UCB-VI (Hier-UCB-VI)

```
Initialize $D = \emptyset$, $Q_{H_h+1}(s,a) = H_h H_l \; \forall s, a$, $V_{H_h+1} = 0$, $\kappa = \max(C, 2H_h H_l)$
for episode $k = 1, ..., K$ do
    for timestep $i = H_h, ..., 1$ do
        for $(s,a) \in S \times A^h$ do
            if $(s,a) \in D$ then
                Update UCB: $UB(r^{\pi^*}(s,a)) = \bar{r}_{N^{k,h}(s,a)}(s,a) + b_r^{s,a}(N^{k,h}(s,a))$
                Set:
                    $$Q_i(s,a) = \min(H_h H_l, UB(r^{\pi^*}(s,a)) + V_{i+1}(g(s,a)))$$
    for $s \in S$ do
        $V_i(s) = \max_{a \in A^h} Q_i(s,a)$
    for time step $h = 1, ..., H_h$ do
        Take greedy high-level action $a_h^k = \operatorname{argmax}_{a \in A^h} Q_h(s_h^k, a)$
        Traverse sub-MDP $M(s_h^k, a_h^k)$ with current sub-policy $\pi_{s_h^k, a_h^k}^{N^{k,h}}$ and transition to $s_{h+1}^k$,
    human supervisor provides low-level rewards of the length-$H_l$ roll-out of $\pi_{s_h^k, a_h^k}^{N^{h,k}}$.
        Feed low-level rewards into no-regret RL algorithm $\mathcal{A}$ for sub-MDP $M(s_h^k, a_h^k)$. Set
    the sum of the low-level rewards (the intermediate return of $\pi_{s_h^k, a_h^k}^{N^{h,k}}$ in $M(s_h^k, a_h^k)$) as the
    high-level reward $r(s_h^k, a_h^k) = r(\pi_{s_h^k, a_h^k}^{N^{h,k}})$
        Add to dataset $D = D \cup \left\{(h, s_h^k, a_h^k, r(s_h^k, a_h^k))\right\}$
```

minimization in the cardinal reward case, which as we will see in the ordinal reward case will not always be true.

Theorem 66 (HRL regret minimization reduces to multi-task, sub-MDP regret minimization)**.** *Let* $UB(\mathcal{R}^{N^{K,H_h}(s,a)})$ *be an upper bound on sub-MDP* $M(s,a)$*'s cumulative regret over* $N^{K,H_h}(s,a)$ *visits:*

$$\sum_{k=1}^{K} V_1^{\pi^*}(s_1) - V_1^{\pi^k}(s_1) \leq \tilde{O}\left(\sum_{s,a \in \mathcal{C}(S,A^h)} UB(\mathcal{R}^{N^{K,H_h}(s,a)}) + H^h H^l \sqrt{N^{K,H_h}(s,a)}\right)$$

Proof Sketch. We describe the key regret decomposition. After some manipulation, the regret may decompose into the following form, $\sum_{k=1}^{K} V_1^k(s_1) - V_1^{\pi^k}(s_1) \leq \sum_{k=1}^{K} \sum_{h=1}^{H_h} \rho_h^k + \gamma_h^k + \sigma_h^k + \zeta_h^k$, which may be parsed as follows.

$\rho_h^k = UB(r^{\pi^*}(s,a)) - r(\pi_{s_h^k, a_h^k}^{N^{k,h}})$ captures the regret due to sub-optimal intermediate return, the return of $\pi_{S,a}^*$ versus the return of $\pi_{s_h^k, a_h^k}$.

$\gamma_h^k = (P_h - P^{\pi_{k,h}}) V_{h+1}^{\pi^*}(s_h^k, a_h^k)$, $\sigma_h^k = (P_h - P^{\pi_{k,h}})(V_{h+1}^k - V_{h+1}^{\pi^*})(s_h^k, a_h^k)$ captures the regret due to sub-optimal policies missing goal-reachability. Here P_h is the idealized transition (goal-reaching), while $P^{\pi_{k,h}}$ is the transition induced by the current sub-policy.

$\zeta_h^k = P^{\pi_{k,h}}(V_{h+1}^k - V_{h+1}^{\pi_k})(s_h^k, a_h^k) - (V_{h+1}^k - V_{h+1}^{\pi_k})(s_{h+1}^k)$ is a martingale difference that concentrates via Azuma Hoeffding, and is dominated by the previous three sums.

Focusing on $\sum_{h=1}^{H_h} \rho_h^k + \gamma_h^k + \sigma_h^k + \zeta_h^k$, we observe that $\gamma_h^k, \sigma_h^k \leq 2H_h H_l P^{\pi_{k,h}}(s_{h+1}^k \neq g(s_h^k, a_h^k))$. The key remaining step is to recognize that $\rho_h^k + \gamma_h^k + \sigma_h^k$ resembles the instantaneous regret in $M(s_h^k, a_h^k)$, and the result follows after some further bounding and rearrangement.

□

For a concrete bound, we note that if $\mathcal{A}$ is set as the classic UCB-VI algorithm, then we attain the usual $\tilde{O}(\sqrt{K})$ regret. Furthermore, we note that our bound is flexible in that one can choose more specialized learning algorithms $\mathcal{A}$ to leverage prior knowledge. For instance, if it is known that sub-MDPs are linear, one may choose to invoke multi-task RL algorithms that offer more refined regret bounds for $UB(\mathcal{R}^{N^{K,H_h}(s,a)})$ [144].

Goal Selection: An astute reader will note that the return of the learned hierarchical policy is close to $V_1^*(s_1)$, the return of the optimal hierarchical policy under *goal function* g. In other words, our learned policy is only as good as the goal function g we choose.

One way to relax the assumption that we have a good goal function g is to assume we have access to multiple goal functions to choose from: $g^1, .., g^n$. Then, an useful corollary of the sublinear Hier-UCB-VI regret bound, $\frac{1}{K}[\sum_{k=1}^{K} V_1^{g^i,*}(s_1) - V_1^{g^i,\pi^k}(s_1)] \leq \tilde{O}(\sqrt{K})$, is that it directly implies an UCB on $V_1^{g^i,*}(s_1)$ (optimal return under goal g^i). Hence, we may apply any UCB-based bandit algorithm on top of this to compete with the return of the best goal out of all the candidates $\{g^j\}_{j\in[n]}$.

11.4 Learning from Preference Feedback

In the previous section, we develop an algorithm to efficiently learn a hierarchical policy, purely from low-level, cardinal feedback. Now, we consider learning from ordinal (preferences) feedback. Our first observation is that the low-level feedback is no longer sufficient for learning a good policy.

Proposition 55 (Non-identifiability of ranking among sub-MDP returns)**.** *For any deterministic high-level policy learning algorithm with N_l samples of low-level feedback, there exists a MDP instance that induces regret constant in N_l.*

The intuition for this is simply that low-level, ordinal feedback can only identify rankings of low-level policies specific to a goal (sub-MDP), but not necessarily low level policies *across* differing goals. Thus, no matter how large the low-level sample-size N_l, the regret is non-vanishing in N_l and hence high-level feedback may be needed to learn. Please see Appendix 11.9 for all proofs of results in this section.

11.4.1 Labeler Feedback and Consequences for Reward Modeling

The canonical approach to learning from preferences is reward modeling. Following previous works, we study offline experiment design and assume we have the ability to collect comparison feedback data, in our hierarchical setting both high and low-level data that are then used to learn the reward model [317]. For tractable analysis, we consider the commonly studied linear reward setup [221, 223, 317, 325].

Assumption 2 (Linear Reward Parametrization)**.** *Suppose we have access to some feature map $\phi : S \times A \to \mathbb{R}^d$, $\mathcal{M}$ has linear reward parametrization w.r.t. ϕ if there exists an unknown, reward vector $\theta^* \in \mathbb{R}^d$ such that $r(s,a) = \langle \phi(s,a), \theta^* \rangle$ for all $s, a \in S \times A$.*

Given trajectory $\tau = (s_1, a_1, ..., s_H, a_H)$, we may then define trajectory feature $\phi(\tau) = \sum_{s_i,a_i \in \tau} \phi(s_i, a_i)$, and policy feature expectation under transitions P, $\phi^P(\pi) = \mathbb{E}_{\tau \sim \pi, P}[\phi(\tau)]$.

With known feature map ϕ and unknown reward parameter θ^*, the preference feedback o_t follows the Bradley-Terry-Luce (BTL) model [51].

Assumption 3. *For trajectories τ_1, τ_2: $\Pr(\tau_1 \succ \tau_2) = \sigma((\theta^*)^T(\phi(\tau_1) - \phi(\tau_2)))$.*

With the definitions out of the way, we now describe a *conceptual challenge* that we encounter when learning from high-level feedback, which as we have shown before may be necessary for learning.

What can we assume about the high-level labeler's knowledge?

Consider a high level trajectory $\tau_j = \left\{(s_i^j, a_i^j)\right\}_{i=1}^{H_h}$. $\phi(\tau_j) = \sum_{i \in [H_h]} \phi(s_i^j, a_i^j)$; the key difficulty is that sub-MDP feature expectation $\phi(s_i^j, a_i^j)$ is dependent on the sub-policy deployed in $M(s_i^j, a_i^j)$. Thus, the high level labeler will have to have in mind some sub-policy $\pi_{s,a}$, when making the comparison. We study two natural types of feedback:

1. **Comparisons based on current sub-policy execution:** It is natural to first assume that the labeler envisions $\phi(s_i^j, a_i^j) = \phi(\pi^t_{s_i^j, a_i^j})$ at time t. In words, it is equivalent to asking: "How well does the high level policy do given *current execution* of sub-goals?"

   Current-feedback of this form has the caveat that the labeler will have know about the performance of the current set of sub-policies $\pi^t_{s,a}$ (potentially through AI-assisted means). This knowledge would need to be updated over time as low-level policies $\pi^t_{s,a}$ improve, which introduces a sizable cognitive load.

2. **Comparisons based on idealized sub-policy execution:** To reduce the cognitive load on the labeler, it is natural to fix the sub-policies used in the comparisons. A natural choice then is for the labeler to envision $\phi(s_i^j, a_i^j) = \phi(\pi^*_{s_i^j, a_i^j})$. In words, it is equivalent to asking: "How well does the high level policy do given *perfect execution* of the sub-goals?" Instantiated in some examples, this would be: "how good is the essay if each argument is fleshed out perfectly" or "how good is the code if each helper function is implemented perfectly".

   Idealized-feedback of this form has the caveat that the high-level feedback will be a mismatch of how the current sub-policies actually execute. Although it has the advantage that the labeler is no longer required to (somehow) keep track of low-level sub-policies, thus reducing the cognitive load.

In what follows, we consider both types of feedback, showing that learning from idealized-feedback is possible. As we note, a drawback of idealized-feedback is that it is biased with respect to the realized features (since these are generated under current policies $\pi^t_{s,a}$), while current-feedback is unbiased. We present an upper bound on the bias below.

Lemma 80 (Bias of idealized-feedback)**.** *Suppose there are N_h, N_l high, low-level trajectories, bias b is such that: $\|b\|^2 = \sum_{t=1}^{N_h} |\langle \theta^*, \phi^{\pi^{N_l}}(\pi_1^i) - \phi^{\pi^{N_l}}(\pi_2^i), - \rangle \langle \theta^*, \phi^{\pi^*}(\pi_1^i) - \phi^{\pi^*}(\pi_2^i), | \rangle^2 =$*

$O(N_h/N_l)$.

Proposition 56 (Reward model learning)**.** *Let* $\theta_{MLE} = \text{argmin}_\theta \, \ell_D(\theta)$ *and let* C_b *denote an upper bound on bias* $C_b \geq \|b\|$*, and* γ, B *constants. We have that with high probability:*

$$\|\theta^* - \theta_{MLE}\|_{\hat{\Sigma}^h_{N^h}+\lambda I} \leq C\sqrt{\frac{C_b\sqrt{N_h}}{\gamma^2} + \frac{C_b^2 + d + \log(1/\delta)}{\gamma^2} + \lambda B^2}$$

11.4.2 Hierarchical Preference Learning

We now construct a hierarchical, preference-learning algorithm that invokes REGIME, a contemporary preference-learning algorithm with provable guarantees, as sub-routine for sub-MDP learning [317].

Sub-MDP reward learning: To start, we again need to incentivize goal-reaching in the sub-MDP reward. As such, given original feature ϕ_{orig}, we introduce an additional feature accounting for goal-reachability. For trajectory τ, define $\phi_i(s_i^\tau, a_i^\tau) = [\phi_{orig}(s_i^\tau, a_i^\tau), \mathbb{1}(i = H_l \wedge s_i^\tau = g(s,a))]$ and for policy π, feature expectation $\phi_i(s_i^\pi, a_i^\pi) = [\phi_{orig}(s_i^\pi, a_i^\pi), \mathbb{1}(i = H_l) \Pr(s_{H_l}^\pi = g(s,a))]$.

The corresponding reward vector will also change to become $\theta^* = [\theta^*_{orig}, \kappa]$ for unknown θ^*_{orig}, κ.

Assumption 4. *Through instructions to the labeler,* κ *may be raised beyond a threshold of our choosing.*

That is, we assume we can provide instructions to the labeler, emphasizing goal-reachability such that κ is higher than some given threshold. As before, we take the threshold to be $\max(C, 2H_hH_l)$. And so while κ is unknown, we know that $\kappa \geq \max(C, 2H_hH_l)$. With this set up, we can then bound the regret due to sub-optimal sub-policies, and sub-optimal simulator $P^{\epsilon'}$, both of which are needed in the final regret analysis.

Lemma 81 (Regret due to sub-optimal sub-policies)**.** *For any high-level policy* π*, with high probability:*

$$\langle \phi^{\pi^*,P}(\pi) - \phi^{\pi^{N_l},P}(\pi), \theta^*, \leq \rangle H_h(\frac{C_1}{\sqrt{N_l}} + C_2\epsilon')$$

where this bound makes use of the REGIME guarantee on sub-MDP $M(s,a)$ *that*
$|\langle \phi^P(\pi^*_{s,a}), \theta^*, - \rangle \phi^{P^{\epsilon'}}(\pi^{N_l}_{s,a}), \theta^*| \leq \frac{C_1}{\sqrt{N_l}} + C_2\epsilon'$ *[317].*

Lemma 82 (Regret due to sub-optimal simulator $P^{\epsilon'}$)**.** *Let* $\Phi^{\pi^{N_l},P^{\epsilon'}}(\pi)$ *denote the feature expectation under high level policy* π*, sub-MDP policies* π^{N_l} *and transitions* $P^{\epsilon'}$*. With high probability, for any high level policy* π*:*

$$|\langle \phi^{\pi^{N_l},P}(\pi) - \phi^{\pi^{N_l},P^{\epsilon'}}(\pi), \theta^*, | \rangle \leq O((H_hd^2 + H_h^3H_l^2)\epsilon' + \frac{H_h^2H_l}{\kappa})$$

11.4.3 Hier-REGIME Analysis

Now, we present the Hier-REGIME Algorithm 27. At a high-level description goes as follows. First, we invoke one copy of REGIME across all sub-MDPs with shared exploration (L1-4) and

Algorithm 27 Hierarchical-REGIME (Hier-REGIME)

```
Require: High-level policy class $\Pi^h$, low level-policy classes $\Pi^l_{s,a}$, simulator $P^{\epsilon'}$ with $\epsilon'$-precision
 1: for episode $n = 1, ..., N_l$ do
 2:     $(\pi_1^n, \pi_2^n) \leftarrow \mathrm{argmax}_{\pi_1,\pi_2 \in \bigcup_{s,a} \Pi^l_{s,a}} \|\phi^{P^{\epsilon'}}(\pi_1) - \phi^{P^{\epsilon'}}(\pi_2)\|_{(\hat{\Sigma}^l_n)^{-1}}$   ▷ explore using policy feature expectation across sub-MDPs
 3:     $\hat{\Sigma}^l_{n+1} = \hat{\Sigma}^l_n + (\phi^{P^{\epsilon'}}(\pi_1^n) - \phi^{P^{\epsilon'}}(\pi_2^n))(\phi^{P^{\epsilon'}}(\pi_1^n) - \phi^{P^{\epsilon'}}(\pi_2^n))^T$
 4:     Generate trajectories $\tau_1^n, \tau_2^n$ and acquire comparison feedback $o_n$
 5: Compute MLE $\hat{\theta}^l$ from $\{\tau_1^n, \tau_2^n\}_{n=1}^{N_l}$ and $\{o_n\}_{n=1}^{N_l}$
 6: Compute $\pi^{N_l}_{s,a} = \mathrm{argmax}_{\pi \in \Pi^l_{s,a}} \langle \phi^{P^{\epsilon'}}(\pi), \hat{\theta}^l \rangle$
 7: if $\left\{\phi^{P^{\epsilon'}}(\pi_1) - \phi^{P^{\epsilon'}}(\pi_2) \mid \pi_1, \pi_2 \in \Pi^h, \pi_{s,a} = \pi^{N_l}_{s,a} \ \forall s, a\right\} \subseteq \left\{\phi^{P^{\epsilon'}}(\pi_1) - \phi^{P^{\epsilon'}}(\pi_2) \mid \pi_1, \pi_2 \in \bigcup_{s,a} \Pi^l_{s,a}\right\}$ then   ▷ Check if high-level feedback can lead to better coverage
 8:     for episode $n = 1, ..., N_h$ do
 9:         $(\pi_1^n, \pi_2^n) \leftarrow \mathrm{argmax}_{\pi_1,\pi_2 \in \Pi^h} \|\phi^{\pi^{N_l}, P^{\epsilon'}}(\pi_1) - \phi^{\pi^{N_l}, P^{\epsilon'}}(\pi_2)\|_{(\hat{\Sigma}^h_n)^{-1}}$
10:         $\hat{\Sigma}^h_{n+1} = \hat{\Sigma}^h_n + (\phi^{\pi^{N_l}, P^{\epsilon'}}(\pi_1) - \phi^{\pi^{N_l}, P^{\epsilon'}}(\pi_2))(\phi^{\pi^{N_l}, P^{\epsilon'}}(\pi_1) - \phi^{\pi^{N_l}, P^{\epsilon'}}(\pi_2))^T$
11:         Generate trajectories $\tau_1^n, \tau_2^n$ and acquire comparison feedback $o_n$
12:     Compute MLE $\hat{\theta}^h$ from $\left\{\tau_1^i, \tau_2^i\right\}_{i=1}^{N_h}$ and $\{o_i\}_{i=1}^{N_h}$
13: else
14:     $\hat{\theta}^h = \hat{\theta}^l$.
    return high-level policy $\hat{\pi} = \mathrm{argmax}_{\pi \in \Pi^h} \langle \phi^{\pi^{N_l}, P^{\epsilon'}}(\pi), \hat{\theta}^h \rangle$, low-level policies $\left\{\pi^{N_l}_{s,a}\right\}_{s,a \in S \times A^h}$
```

learned reward (L5). Next, we use the learned reward to compute sub-MDP policies $\pi^{N_l}_{s,a}$ for each sub-MDP $M(s, a)$ (L6). Finally, we invoke one copy of REGIME for the high-level MDP, where the feature function is defined as $\phi^{\pi^{N_l}_{s,a}, P^{\epsilon'}}$ (L8). Next, we note two properties about Algorithm 27.

Hierarchical Exploration: A key aspect of experiment design in offline RL is ensuring sufficient coverage with exploration. The difficulty with coverage in the hierarchical setting is that at first glance, we may need to search for pairs of trajectories over $(\pi_1, \left\{\pi^1_{s,a}\right\}), (\pi_1, \left\{\pi^2_{s,a}\right\}) \in (\Pi^h, \bigtimes_{s,a} \Pi^l_{s,a})$, instead of over $\pi_1, \pi_2 \in \Pi^h$. However, we show that in the goal-HRL case, we can fix the sub-policies to be $\pi^{N_l}_{s,a}$ (for N_l large enough), and this is sufficient to compete with the optimal, hierarchical policy.

Additionally, unlike the tabular setting, sub-MDPs now share a common reward parameter θ^*, thus allowing us to jointly, instead of separately as in tabular case, explore across sub-MDPs.

Sufficiency of low-level feedback: Through the algorithm, we can observe that low- and high-level exploration generates feature expectations set: $\left\{\phi^{P^{\epsilon'}}(\pi_1) - \phi^{P^{\epsilon'}}(\pi_2) \mid \pi_1, \pi_2 \in \bigcup_{s,a} \Pi^l_{s,a}\right\}$ and $\left\{\phi^{P^{\epsilon'}}(\pi_1) - \phi^{P^{\epsilon'}}(\pi_2) \mid \pi_1, \pi_2 \in \Pi^h, \pi_{s,a} = \pi^{N_l}_{s,a} \ \forall s, a\right\}$. Therefore, when coverage of high level policy is subsumed by low-level features already (the latter is a subset of the former), it suffices to explore only using low-level feedback. As shown before in Proposition 56, it is not always sufficient. However, as we will see below, when it is sufficient, using low-level feedback leads to better rates. First, we derive the regret decomposition and then use it evaluate the sample complexity.

Theorem 67. *With high probability, under $N_h > 0$:*

$$\begin{aligned}
&V^{\pi^*,\pi^*} - V^{\hat{\pi},\pi^{N_l}} \\
&\leq \langle \phi^{\pi^*,P}(\pi^*) - \phi^{\pi^{N_l},P}(\pi^*), \theta^*, + \rangle \frac{1}{\sqrt{N_h}}(2d\log(1 + \frac{N_h}{d}))\|\theta^* - \hat{\theta}\|_{\hat{\Sigma}^h_{N_h}} + \\
&|\langle \phi^{\pi^{N_l},P}(\pi^*) - \phi^{\pi^{N_l},P^{\epsilon'}}(\pi^*), \theta^*, | \rangle + |\langle \phi^{\pi^{N_l},P^{\epsilon'}}(\hat{\pi}) - \phi^{\pi^{N_l},P}(\hat{\pi}), \theta^*, | \rangle
\end{aligned}$$

To parse this, the regret decomposes into four terms. The first term is the regret due to sub-optimality in low-level policies π^{N_l}. The remaining three terms are derived from sub-optimality due to high-level policy $\hat{\pi}$, decomposing into the second term on regret due to bias in learned reward $\hat{\theta}$, the third and fourth term on regret due to sub-optimality of simulator $P^{\epsilon'}$.

A main benefit of developing a learning Algorithm 27 is that we can then quantitatively assess the sample complexity associated with the two types of human feedback. As one may expect, there is a tradeoff between better sample complexity and cognitive load, with current-feedback attaining better sample efficiency but also requiring higher cognitive load on the human supervisor.

Corollary 10. *Using Theorem 67, we obtain the following rates in terms of data tradeoffs:*

- ***Idealized-feedback and required high-/low-level feedback:*** *the overall rate comes out to $O(N_l^{-1/4} + N_h^{-1/2})$. While high level trajectories provide additional coverage, it also incurs bias linear in N_h of the bias of the low-level trajectories, thus slowing down the rate (Lemma 80).*
- ***Current-feedback and required high-/low-level feedback:*** *the overall rate comes out to $O(N_l^{-1/2} + N_h^{-1/2})$. The current-feedback is unbiased and results in more efficient reward learning with $\|\theta^* - \hat{\theta}\|_{\hat{\Sigma}^h_{N_h}} = O(1)$ [317].*
- ***Only low-level feedback is required due to sufficiency in coverage:*** *the overall rate comes out to $O(N_l^{-1/2})$. In a nutshell, this is because we can explore with just N_l low-level samples which is unbiased, resulting in $\|\theta^* - \hat{\theta}\|_{\hat{\Sigma}^l_{N_l}} = O(1)$. Hence, both exploration and reward learning is efficient.*

11.5 Discussion

Our work considers scalable oversight in the context of goal-conditioned HRL, in which we show that one can efficiently use hierarchical structure to learn from bounded human feedback.

Limitations & Future Work: In goal-conditioned HRL, our regret guarantees are with respect to the return of the optimal, hierarchical policy, whose performance is dependent on the usefulness of goal function g. Further research is needed to understand on how to learn good goal functions, using limited supervised or unsupervised learning. Additionally, under current-feedback, the labeler providing high-level feedback is somehow made aware of sub-policy performance. An exciting research direction is how one may provide such knowledge through AI-assistance.

11.6 More Related Works

Scalable Oversight: Scalable oversight is a nascent but important topic in the area of AI alignment [11, 50, 71, 180], wherein the goal is to boost the labeler's ability to provide feedback to complex models. Proposed approaches include (recursive) self-critique, summarization, debate, plain model Interaction and market-making, all of which aim to have the model (or auxiliary models) generate interpretable and/or lower-dimensional forms of outputs for the human to parse [50, 147, 151, 180, 247, 267, 296]. Our work studies how one may leverage hierarchical structure as one approach to scaling up feedback.

Goal-conditioned RL: Further afield, there has been a lot of work demonstrating the promise/success of goal-conditioned RL with examples from the likes of [60, 130, 185, 210]. The sub-MDP reward is often set to incentivize *only* goal state reachability, as oftentimes the MDP of interest has sparse rewards, making intermediate returns zero. In our setting, rewards need not be sparse, thus bringing into consideration the tradeoff between intermediate return and goal-reachability. This work initiates the study of scalable oversight in goal-oriented HRL, and owing to the success of goal-oriented HRL in practice, it is our hope that it can be stepping stone towards developing practical scalable oversight techniques.

11.7 Concrete Hierarchical MDP Example

The prototypical example in HRL is the maze, as studied in for instance [210, 241]. A maze consists of rooms with doors. The goal is to get to the exit in as few steps as possible. The MDP may be defined as follows:

- For the global MDP, $S = S^h \times S^l$ where s^h denotes the index of the current room, and s^l denotes the position of the agent in the room. Action set A consists of moving (L, R, U, D, Stay).
- For the High-level MDP, high-level action A^h consists of moving to the (N, S, E, W) door of the room. s is the current location of the agent, and $g(s, a^h)$ maps the goal (door) to its location.
- For the Low-level MDP, it has state space $S^l_{s,a} \subset S$ and the action set A is the same moving (U, D, L, R, Stay).

As noted in the previous section, HRL algorithms can achieve superior statistical sample complexity when there is lots of repeated sub-MDP structure (there are many isomoprhic rooms) and each room has small state-space size [291].

Notation	
$M(s,a)$	sub-MDP at state s with high level action a
$\pi^i_{s,a}$	policy used by sub-MDP $M(s,a)$'s no-regret algorithm during the i-th visit
$\pi^*_{s,a}$	optimal policy in sub-MDP $M(s,a)$
$r(\pi^i_{s,a})$	expected reward of policy $\pi^i_{s,a}$ in sub-MDP $M(s,a)$
$r_{l,h}$	sub-MDP reward definition.
$\hat{r}(\pi^i_{s,a})$	observed reward of policy π in sub-MDP $M(s,a)$
$\bar{r}_n(s,a)$	average observed policy reward $\bar{r}_n(s,a) = \frac{1}{n}\sum_{i=1}^n \hat{r}(\pi^i_{s,a})$
$\mathcal{R}^n(s,a)$	sub-MDP $M(s,a)$ cumulative regret across n steps, $\mathcal{R}^n(s,a) = \sum_{i=1}^n r(\pi^*_{s,a}) - r(\pi^i_{s,a})$
$N^{k,h}(s,a)$	number of times $M(s,a)$ has been visited up until episode k, horizon h
$P^\pi(\cdot \mid s,a)$	distribution over states of policy π after going through subMDP $M(s,a)$
ψ_n	a factor such that $\psi_n = \tilde{O}(\sqrt{n})$, where the $\tilde{O}$ omits up to $\log$ dependence on K

Table 11.1: Table of notation used in this section.

11.8 Proofs for Section 11.3

11.8.1 Sub-MDP Bonus Construction

Sub-MDP Reward Definition: Define the reward in sub-MDP $M(s,a)$ at time step h to be: $r_{l,h}(s',a') = r(s',a') + \kappa\mathbb{1}(h = H_l \wedge s' = g(s,a))$.

Firstly, since by definition $\pi^*_{s,a} \in \operatorname{argmax}_{\pi\in\Pi_{s,a}} r(\pi) + C\cdot\Pr(s^\pi_{H_l} = g(s,a))$, we have that $\pi^*_{s,a} \in \operatorname{argmax}_{\pi\in\Pi_{s,a}} r(\pi) + \kappa\cdot\Pr(s^\pi_{H_l} = g(s,a))$.

Indeed,

$$\begin{aligned}
&r(\pi^*_{s,a}) + \kappa\Pr(s^{\pi^*_{s,a}}_{H_l} = g(s,a)) \\
&= [r(\pi^*_{s,a}) + C\cdot\Pr(s^{\pi^*_{s,a}}_{H_l} = g(s,a))] + (\kappa - C)\Pr(s^{\pi^*_{s,a}}_{H_l} = g(s,a)) \\
&\geq [r(\pi) + C\cdot\Pr(s^{\pi}_{H_l} = g(s,a))] + (\kappa - C)\Pr(s^{\pi}_{H_l} = g(s,a)) \\
&\qquad\qquad (\Pr(s^{\pi^*_{s,a}}_{H_l} = g(s,a)) = 1 \geq \Pr(s^{\pi}_{H_l} = g(s,a))\ \forall\pi)
\end{aligned}$$

Secondly, using the definition of r_l, we have that:

$$r_l(\pi^*_{s,a}) - r_l(\pi^i_{s,a}) = r(\pi^*_{s,a}) + \kappa P(s^{\pi^*_{s,a}}_{H_l} = g(s,a)) - r(\pi^i_{s,a}) - \kappa P(s^{\pi^i_{s,a}}_{H_l} = g(s,a))$$

By the reachability assumption, $P(s^{\pi^*_{s,a}}_{H_l} = g(s,a)) = 1$, this implies that

$$r_l(\pi^*_{s,a}) - r_l(\pi^i_{s,a}) = r(\pi^*_{s,a}) - r(\pi^i_{s,a}) + \kappa P(s^{\pi^i_{s,a}}_{H_l} \neq g(s,a))$$

Therefore, summing this across n visits to $M(s,a)$, we have:

$$
\begin{aligned}
&\mathcal{R}^n(s,a)\\
&=\sum_{i=1}^{n} r_l(\pi^*_{s,a}) - r_l(\pi^i_{s,a})\\
&=\sum_{i=1}^{n} r(\pi^*_{s,a}) - r(\pi^i_{s,a}) + \kappa\sum_{i=1}^{n} P(s_{H_l}^{\pi^i_{s,a}} \neq g(s,a))
\end{aligned}
$$

This statement is useful because we can compute an UCB on $\sum_{i=1}^{n} r(\pi^*_{s,a})$ and, implicitly, a LCB on $\sum_{i=1}^{n} r(\pi^i_{s,a})$ (provided we do not bound $\mathcal{R}^n(s,a)$).

Lemma 83 (Bonus with "penalty" for non-reachability)**.** *Let $UB(\mathcal{R}^n(s,a))$ be any upper bound on the sub-MDP regret, then if we define:*

$$
b_r^{s,a}(n) = \frac{UB(\mathcal{R}^n(s,a)) + (\kappa + H_l)2\log(\frac{|\mathcal{C}(S,A^h)|H_hK}{\delta})\sqrt{n}}{n} - \frac{\kappa}{n}\sum_{i=1}^{n} \mathbb{1}(s_{H_l}^{\pi^i_{s,a}} \neq g(s,a))
$$

Then, $\bar{r}_n(s,a) + b_r^{s,a}(n)$ is an UCB for $r(\pi^*_{s,a})$ with probability $\geq 1 - \frac{\delta}{3|\mathcal{C}(S,A^h)|H_hK}$.
Let the event that the above holds be $\mathcal{E}^n_{s,a}$.

Proof.

$$
\begin{aligned}
&\sum_{i=1}^{n} r(\pi^*_{s,a})\\
&= \mathcal{R}^n(s,a) - \kappa\sum_{i=1}^{n} P(s_{H_l}^{\pi^i_{s,a}} \neq g(s,a)) + \sum_{i=1}^{n} r(\pi^i_{s,a})\\
&\leq \mathcal{R}^n(s,a) - \kappa(\sum_{i=1}^{n} \mathbb{1}(s_{H_l}^{\pi^i_{s,a}} \neq g(s,a)) - \psi_n) + \sum_{i=1}^{n} r(\pi^i_{s,a}) && (\diamond)\\
&= \mathcal{R}^n(s,a) - \kappa\sum_{i=1}^{n} \mathbb{1}(s_{H_l}^{\pi^i_{s,a}} \neq g(s,a)) + \kappa\psi_n + \sum_{i=1}^{n} \hat{r}(\pi^i_{s,a}) + (\sum_{i=1}^{n} r(\pi^i_{s,a}) - \sum_{i=1}^{n} \hat{r}(\pi^i_{s,a}))\\
&\leq UB(\mathcal{R}^n(s,a)) + (\kappa + H_l)\psi_n - \kappa\sum_{i=1}^{n} \mathbb{1}(s_{H_l}^{\pi^i_{s,a}} \neq g(s,a)) + \sum_{i=1}^{n} \hat{r}(\pi^i_{s,a}) && (\kappa' = \kappa + H_l)
\end{aligned}
$$

$(\diamond)$: Here we use two applications of Azuma-Hoeffding:

- With probability higher than $1 - \delta$:

$$
|\sum_{i=1}^{n} P(s_{H_l}^{\pi^i_{s,a}} \neq g(s,a)) - \sum_{i=1}^{n} \mathbb{1}(s_{H_l}^{\pi^i_{s,a}} \neq g(s,a))| \leq \psi_n = 2\sqrt{n}
$$

We have that $\mathbb{E}[P(s_{H_l}^{\pi_{s,a}^i} \neq g(s,a)) - \mathbb{1}(s_{H_l}^{\pi_{s,a}^i} \neq g(s,a))|\mathcal{F}_{i-1}] = 0$.

This is true because $P(s_{H_l}^{\pi_{s,a}^i} \neq g(s,a))$ and $\mathbb{1}(s_{H_l}^{\pi_{s,a}^i} \neq g(s,a)$ are a function of only the transition probability of the MDP at the ith step conditioned on $\mathcal{F}_{i-1}$. Thus, $P(s_{H_l}^{\pi_{s,a}^i} \neq g(s,a)) - \mathbb{1}(s_{H_l}^{\pi_{s,a}^i} \neq g(s,a))$ is a martingale difference. And we can use Azuma-Hoeffding.

- With probability higher than $1-\delta$:

$$|\sum_{i=1}^{n} r(\pi_{s,a}^i) - \sum_{i=1}^{n} \hat{r}(\pi_{s,a}^i)| \leq H_l \psi_n \leq H_l 2\sqrt{n}$$

This again follows from Azuma-Hoeffding on martingale difference $r(\pi_{s,a}^i) - \hat{r}(\pi_{s,a}^i)$, as $\mathbb{E}[r(\pi_{s,a}^i) - \hat{r}(\pi_{s,a}^i)|\mathcal{F}_{i-1}] = 0$. And $|r(\pi_{s,a}^i) - \hat{r}(\pi_{s,a}^i)| \leq H_l$.

Thus,

$$r(\pi_{s,a}^*) \leq \frac{1}{n}\sum_{i=1}^{n} \hat{r}(\pi_{s,a}^i) + b_r^{s,a}(n) \Rightarrow r(\pi_{s,a}^*) - \bar{r}_n(s,a) \leq b_r^{s,a}(n)$$

□

Remark 26. *One choice for $UB(\mathcal{R}^n(s,a)) = H_l^{3/2}\sqrt{|S_{s,a}^l||A|n}$ if we let $\mathcal{A}_{s,a}$ be the standard UCB-VI algorithm [18].*

11.8.2 Optimism Lemma

Lemma 84 (Optimism)**.** *Let V_h^k be the V value as in Algorithm 26 at episode k. Let π^* be the optimal hierarchical policy. For a fixed k and h, if $\forall s, a, n$, $\mathcal{E}_{s,a}^n$ holds, then:*

$$V_h^k(s) \geq V_h^{\pi^*}(s) \quad \forall s$$

Proof. Fix some episode k. We will prove this lemma via induction on $h = H_h + 1, ..., 1$.

Base case: At $h = H_h + 1$, $V_h^k(s) \geq 0 = V_h^{\pi^*}(s)$ for all s.

Induction Step: Suppose this is true for up until $h = H_h + 1, ..., h' + 1$. Now at time step h' and any s, a.

Firstly, if $Q_{h'}^k(s,a) = H_h H_l$ (e.g. if $s, a \notin \mathcal{D}^k$), then $Q_{h'}^k(s,a) \geq Q_{h'}^*(s,a)$. Otherwise, $Q_{h'}^k(s,a) < H_h H_l$ and we have that:

$$\begin{aligned} Q_{h'}^k(s,a) - Q_{h'}^*(s,a) &= [\bar{r}_{N^{k,h}(s,a)}(s,a) + b_r^{s,a}(N^{k,h}(s,a)) + V_{h'+1}^k(g(s,a))] - (r(\pi_{s,a}^*) + P_{h'}V_{h'+1}^{\pi^*}(s,a)) \\ & \qquad \qquad \text{($Q_{h'}^k$ definition as in Equation 7)} \\ &\geq V_{h'+1}^k(g(s,a)) - P_{h'}V_{h'+1}^{\pi^*}(s,a) \\ & \qquad \qquad \text{($\bar{r}_{N^{k,h}(s,a)}(s,a) + b_r^{s,a}(N^{k,h}(s,a))$ is an UCB of $r(\pi_{s,a}^*)$)} \\ &= V_{h'+1}^k(g(s,a)) - V_{h'+1}^{\pi^*}(g(s,a)) \\ & \qquad \qquad \text{($\pi_{s,a}^*$ reaches goal state w.p 1, so $P_{h'}(g(s,a)|s,a) = 1$)} \\ &\geq 0 \qquad \qquad \text{(induction hypothesis)} \end{aligned}$$

Thus, $V_{h'}^k(s) = \max_a Q_{h'}^k(s,a) \geq \max_a Q_{h'}^*(s,a) = V_{h'}^{\pi^*}(s)$.

□

Corollary 11.

$$\sum_{k=1}^{K} V_1^{\pi^*}(s_1) - V_1^{\pi^k}(s_1) \leq \sum_{k=1}^{K} V_1^k(s_1) - V_1^{\pi^k}(s_1)$$

11.8.3 Supporting results needed for regret analysis

Proposition 57.

$$\sum_{k=1}^{K} V_1^k(s_1) - V_1^{\pi^k}(s_1) \leq \sum_{k=1}^{K}\sum_{h=1}^{H_h} \zeta_h^k + \gamma_h^k + \sigma_h^k + \rho_h^k$$

Proof. For any k and h, we consider bounding $V_h^k(s_h^k) - V_h^{\pi_k}(s_h^k)$, which is equal to:

$$\begin{aligned}
V_h^k(s_h^k) - V_h^{\pi_k}(s_h^k) &= (Q_h^k - Q_h^{\pi_k})(s_h^k, a_h^k) \\
&\leq (\bar{r}_{N^{k,h}(s_h^k,a_h^k)}(s_h^k, a_h^k) + b_r^{s_h^k,a_h^k}(N^{k,h}(s_h^k, a_h^k))) - r(\pi_{s_h^k,a_h^k}^{N^{k,h}(s_h^k,a_h^k)}) \\
&+ V_{h+1}^k(g(s_h^k, a_h^k)) - P^{\pi_{k,h}} V_{h+1}^{\pi_k}(s_h^k, a_h^k) && \text{(due to the } \min\text{)} \\
&= \rho_h^k + [V_{h+1}^k(g(s_h^k, a_h^k)) - P^{\pi_{k,h}} V_{h+1}^{\pi_k}(s_h^k, a_h^k)]
\end{aligned}$$

where we set $\rho_h^k = \bar{r}_{N^{k,h}(s_h^k,a_h^k)}(s_h^k, a_h^k) + b_r^{s_h^k,a_h^k}(N^{k,h}(s_h^k, a_h^k))) - r(\pi_{s_h^k,a_h^k}^{N^{k,h}(s_h^k,a_h^k)})$.
Continuing with the original proof and focusing on the second term:

$$\begin{aligned}
&V_{h+1}^k(g(s_h^k, a_h^k)) - P^{\pi_{k,h}} V_{h+1}^{\pi_k}(s_h^k, a_h^k) \\
&= V_{h+1}^k(g(s_h^k, a_h^k)) - P^{\pi_{k,h}} V_{h+1}^k(s_h^k, a_h^k) + P^{\pi_{k,h}}(V_{h+1}^k - V_{h+1}^{\pi_k})(s_h^k, a_h^k) \\
&= (P_h - P^{\pi_{k,h}}) V_{h+1}^k(s_h^k, a_h^k) + P^{\pi_{k,h}}(V_{h+1}^k - V_{h+1}^{\pi_k})(s_h^k, a_h^k)
\end{aligned}$$

(P^h is the transition under optimal sub MDP policy so it takes s_h^k, a_h^k to $g(s_h^k, a_h^k)$ deterministically)

$$\begin{aligned}
&= (P_h - P^{\pi_{k,h}}) V_{h+1}^{\pi^*}(s_h^k, a_h^k) + (P_h - P^{\pi_{k,h}})(V_{h+1}^k - V_{h+1}^{\pi^*})(s_h^k, a_h^k) + P^{\pi_{k,h}}(V_{h+1}^k - V_{h+1}^{\pi_k})(s_h^k, a_h^k) \\
&= \gamma_h^k + \sigma_h^k + P^{\pi_{k,h}}(V_{h+1}^k - V_{h+1}^{\pi_k})(s_h^k, a_h^k)
\end{aligned}$$

where

- $\gamma_h^k = (P_h - P^{\pi_{k,h}}) V_{h+1}^{\pi^*}(s_h^k, a_h^k)$
- $\sigma_h^k = (P_h - P^{\pi_{k,h}})(V_{h+1}^k - V_{h+1}^{\pi^*})(s_h^k, a_h^k)$

In summary,

$$\begin{aligned}
&V_h^k(s_h^k) - V_h^{\pi_k}(s_h^k) \\
&\le \rho_h^k + \gamma_h^k + \sigma_h^k + P^{\pi_{k,h}}(V_{h+1}^k - V_{h+1}^{\pi_k})(s_h^k, a_h^k) \\
&= (V_{h+1}^k - V_{h+1}^{\pi_k})(s_{h+1}^k) + \zeta_h^k + \gamma_h^k + \sigma_h^k + \rho_h^k,
\end{aligned}$$

where we introduce the notation $\zeta_h^k = P^{\pi_{k,h}}(V_{h+1}^k - V_{h+1}^{\pi_k})(s_h^k, a_h^k) - (V_{h+1}^k - V_{h+1}^{\pi_k})(s_{h+1}^k)$. Unrolling the recursion starting at $h = 1$:

$$\begin{aligned}
&V_1^k(s_h^k) - V_1^{\pi_k}(s_h^k) \\
&\le 1(\zeta_h^k + \gamma_h^k + \sigma_h^k + \rho_h^k) + ... + (1)^{H_h}(\zeta_{H_h}^k + \gamma_{H_h}^k + \sigma_{H_h}^k + \rho_{H_h}^k) \\
&= 1 \cdot (\sum_{h=1}^{H_h} \zeta_h^k + \gamma_h^k + \sigma_h^k + \rho_h^k)
\end{aligned}$$

Summing across $k \in [K]$, it suffices to bound:

$$\sum_{k=1}^{K} V_1^k(s_1) - V_1^{\pi^k}(s_1) \le \sum_{k=1}^{K} \sum_{h=1}^{H_h} \zeta_h^k + \gamma_h^k + \sigma_h^k + \rho_h^k$$

□

Remark 27. *We note that there are two sources of sub-optimality in the bound. One is the sub-optimality while executing the sub-MDP policies. This is covered by the per-step high level reward bonus (which is also the UCB on the return of the sub-MDP's return) in ρ_h^k. The other is the sub-optimality of not landing on $g(s_h^k, a_h^k)$, there is covered by γ_h^k, σ_h^k, which affects future reward. The martingale difference ζ_h^k is zero in expectation, so it is not some measure of suboptimality.*

We first bound the ζ's, whose sum is dominated by $\sum_{k=1}^{K} \sum_{h=1}^{H_h} \rho_h^k + \gamma_h^k + \sigma_h^k$.

Lemma 85. *With probability $\ge 1 - \delta/3$:*

$$\sum_{k=1}^{K} \sum_{h=1}^{H_h} \zeta_h^k \le \tilde{O}(H^h H^l \sqrt{H^h K})$$

Let the event that the above inequality hold be $\mathcal{E}^\zeta$.

Proof. The concentration of ζ_h^k follows from Azuma Hoeffding, as the following is a martingale difference.

$$\zeta_h^k = P^{\pi_{k,h}}(V_{h+1}^k - V_{h+1}^{\pi_k})(s_h^k, a_h^k) - (V_{h+1}^k - V_{h+1}^{\pi_k})(s_{h+1}^k)$$

with $\mathbb{E}[\zeta_h^k | F_{k,h}] = 0$, since the expectation is only wrt randomness in s_{h+1}^k. Moreover, this martingale difference is bounded by $4H^h H^l$

□

Next, we simplify the sum of remaining terms.

Lemma 86. *We have that:*

$$\sum_{k=1}^{K}\sum_{h=1}^{H_h}\gamma_h^k \leq H^h H^l \sum_{k=1}^{K}\sum_{h=1}^{H_h} P^{\pi_{k,h}}(s_{h+1}^k \neq g(s_h^k, a_h^k))$$

and

$$\sum_{k=1}^{K}\sum_{h=1}^{H_h}\sigma_h^k \leq H^h H^l \sum_{k=1}^{K}\sum_{h=1}^{H_h} P^{\pi_{k,h}}(s_{h+1}^k \neq g(s_h^k, a_h^k))$$

Proof.

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{h=1}^{H_h}\gamma_h^k \\
&= \sum_{k=1}^{K}\sum_{h=1}^{H_h}(P_h - P^{\pi_{k,h}})V_{h+1}^{\pi^*}(s_h^k, a_h^k) \\
&= \sum_{k=1}^{K}\sum_{h=1}^{H_h} P^{\pi_{k,h}}(s_{h+1}^k \neq g(s_h^k, a_h^k))(V_{h+1}^{\pi^*}(g(s_h^k, a_h^k)) - V_{h+1}^{\pi^*}(s_{h+1}^k)) \\
&\leq H^h H^l \sum_{k=1}^{K}\sum_{h=1}^{H_h} P^{\pi_{k,h}}(s_{h+1}^k \neq g(s_h^k, a_h^k))
\end{aligned}$$

Similarly,

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{h=1}^{H_h}\sigma_h^k \\
&= \sum_{k=1}^{K}\sum_{h=1}^{H_h}(P_h - P^{\pi_{k,h}})(V_{h+1}^k - V_{h+1}^{\pi^*})(s_h^k, a_h^k) \\
&= \sum_{k=1}^{K}\sum_{h=1}^{H_h} P^{\pi_{k,h}}(s_{h+1}^k \neq g(s_h^k, a_h^k))[(V_{h+1}^k - V_{h+1}^{\pi^*})(g(s_h^k, a_h^k)) - (V_{h+1}^k - V_{h+1}^{\pi^*})(s_{h+1}^k)] \\
&\leq H^h H^l \sum_{k=1}^{K}\sum_{h=1}^{H_h} P^{\pi_{k,h}}(s_{h+1}^k \neq g(s_h^k, a_h^k))
\end{aligned}$$

□

Next, we will show the following upper bound and let $\mathcal{E}^{\rho}$ be the event that it holds.

Lemma 87. *With probability* $\geq 1-\delta/3$*:*

$$\sum_{k=1}^{K}\sum_{h=1}^{H_h}\rho_h^k \leq \sum_{s,a\in\mathcal{C}(S,A^h)}\sum_{i=1}^{N^{K,H_h}(s,a)} r(\pi^*_{s,a}) - r(\pi^i_{s,a}) + \sum_{s,a\in\mathcal{C}(S,A^h)}\sum_{i=1}^{N^{K,H_h}(s,a)} \frac{UB(\mathcal{R}^i(s,a)) - \mathcal{R}^i(s,a) + (\kappa''+\kappa)\psi_i}{i}$$

Proof. We first expand the ρ_h^k sum:

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{h=1}^{H_h}\rho_h^k \\
&= \sum_{k=1}^{K}\sum_{h=1}^{H_h} \bar{r}_{N^{k,h}(s_h^k,a_h^k)}(s_h^k,a_h^k) + b_r^{s_h^k,a_h^k}(N^{k,h}(s_h^k,a_h^k))) - r(\pi_{s_h^k,a_h^k}^{N^{k,h}(s_h^k,a_h^k)}) \\
&= \sum_{s,a\in\mathcal{C}(S,A^h)}\sum_{i=1}^{N^{K,H_h}(s,a)} \bar{r}_i(s,a) + b_r^{s,a}(i) - r(\pi^i_{s,a}) \\
&= \sum_{s,a\in\mathcal{C}(S,A^h)}\sum_{i=1}^{N^{K,H_h}(s,a)} \frac{1}{i}\sum_{j=1}^{i}\hat{r}(\pi^j_{s,a}) + \frac{UB(\mathcal{R}^i(s,a)) + \kappa'\psi_i - \kappa\sum_{j=1}^{i}\mathbb{1}(s_{H_l}^{\pi^j_{s,a}} \neq g(s,a))}{i} - r(\pi^i_{s,a}) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(using definition of bonus)} \\
&\leq \sum_{s,a\in\mathcal{C}(S,A^h)}\sum_{i=1}^{N^{K,H_h}(s,a)} \frac{1}{i}\sum_{j=1}^{i} r(\pi^j_{s,a}) + \frac{H_l\psi_i}{i} + \frac{UB(\mathcal{R}^i(s,a)) + \kappa'\psi_i - \kappa\sum_{j=1}^{i}\mathbb{1}(s_{H_l}^{\pi^j_{s,a}} \neq g(s,a))}{i} - r(\pi^i_{s,a}) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(Azume-Hoeffding for concentration of } \hat{r} \text{ around } r\text{)}
\end{aligned}$$

Using the two-sided concentration bound we had before (the other way): $\sum_{j=1}^{i}\mathbb{1}(s_{H_l}^{\pi^j_{s,a}} \neq g(s,a)) + \psi_i \geq \sum_{j=1}^{i} P(s_{H_l}^{\pi^j_{s,a}} \neq g(s,a))$ w.h.p:

$$\begin{aligned}
&\sum_{j=1}^{i} r(\pi^*_{s,a}) - r(\pi^j_{s,a}) \geq \mathcal{R}^i(s,a) - \kappa(\sum_{j=1}^{i}\mathbb{1}(s_{H_l}^{\pi^j_{s,a}} \neq g(s,a)) + \psi_i) \\
&\Rightarrow \sum_{j=1}^{i} r(\pi^*_{s,a}) - \mathcal{R}^i(s,a) + \kappa\psi_i \geq \sum_{j=1}^{i} r(\pi^j_{s,a}) - \kappa\sum_{j=1}^{i}\mathbb{1}(s_{H_l}^{\pi^j_{s,a}} \neq g(s,a))
\end{aligned}$$

We continue our derivation:

$$\sum_{s,a \in \mathcal{C}(S,A^h)} \sum_{i=1}^{N^{K,H_h}(s,a)} \frac{1}{i}(\sum_{j=1}^{i} r(\pi^j_{s,a}) + UB(\mathcal{R}^i(s,a)) + \kappa''\psi_i - \kappa \sum_{j=1}^{i} \mathbb{1}(s^{\pi_j}_{H_l} \neq g(s,a))) - r(\pi^i_{s,a})$$

$$(\kappa'' = \kappa' + H_l)$$

$$\leq \sum_{s,a \in \mathcal{C}(S,A^h)} \sum_{i=1}^{N^{K,H_h}(s,a)} \frac{1}{i}[\sum_{j=1}^{i} r(\pi^*_{s,a}) - \mathcal{R}^i(s,a) + \kappa\psi_i] - r(\pi^i_{s,a}) + \sum_{s,a \in \mathcal{C}(S,A^h)} \sum_{i=1}^{N^{K,H_h}(s,a)} \frac{UB(\mathcal{R}^i(s,a)) + \kappa''\psi_i}{i}$$

(using the identity above)

$$\leq \sum_{s,a \in \mathcal{C}(S,A^h)} \sum_{i=1}^{N^{K,H_h}(s,a)} r(\pi^*_{s,a}) - r(\pi^i_{s,a}) + \sum_{s,a \in \mathcal{C}(S,A^h)} \sum_{i=1}^{N^{K,H_h}(s,a)} \frac{UB(\mathcal{R}^i(s,a)) - \mathcal{R}^i(s,a) + (\kappa'' + \kappa)\psi_i}{i}$$

□

11.8.3.1 Overall Regret Bound

Theorem 68. *Under events* $\bigcap_{s,a,n} \mathcal{E}^n_{s,a} \cap \mathcal{E}^\zeta \cap \mathcal{E}^\rho$, *we have that:* $\sum_{k=1}^{K} \sum_{h=1}^{H_h} \rho^k_h + \gamma^k_h + \sigma^k_h \leq \sum_{s,a \in \mathcal{C}(S,A^h)} (\log(N^{K,H_h}(s,a)) + 1) UB(\mathcal{R}^{N^{K,H_h}(s,a)}) + O(H^h H^l \sqrt{N^{K,H_h}(s,a)})$.

Proof.

$$\sum_{k=1}^{K} \sum_{h=1}^{H_h} \rho^k_h + \gamma^k_h + \sigma^k_h$$

$$\leq \sum_{s,a \in \mathcal{C}(S,A^h)} \sum_{i=1}^{N^{K,H_h}(s,a)} r(\pi^*_{s,a}) - r(\pi^i_{s,a}) +$$

$$\sum_{s,a \in \mathcal{C}(S,A^h)} \sum_{i=1}^{N^{K,H_h}(s,a)} \frac{UB(\mathcal{R}^i(s,a)) - \mathcal{R}^i(s,a) + \kappa\psi_i}{i} + 2H^h H^l \sum_{k=1}^{K} \sum_{h=1}^{H_h} P^{\pi_{k,h}}(s^k_{h+1} \neq g(s^k_h, a^k_h))$$

$$= \sum_{s,a \in \mathcal{C}(S,A^h)} \sum_{i=1}^{N^{K,H_h}(s,a)} \frac{UB(\mathcal{R}^i(s,a)) - \mathcal{R}^i(s,a) + \kappa\psi_i}{i}$$

$$+ \sum_{s,a \in \mathcal{C}(S,A^h)} \sum_{i=1}^{N^{K,H_h}(s,a)} r(\pi^*_{s,a}) - r(\pi^i_{s,a}) + 2H^h H^l \sum_{s,a \in \mathcal{C}(S,A^h)} [\sum_{i=1}^{N^{K,H_h}(s,a)} P(s^{\pi^i_{s,a}}_{H_l} \neq g(s^k_h, a^k_h))]$$

(group third sum by s, a)

$$\leq \sum_{s,a \in \mathcal{C}(S,A^h)} \sum_{i=1}^{N^{K,H_h}(s,a)} \frac{UB(\mathcal{R}^i(s,a)) - \mathcal{R}^i(s,a) + \kappa\psi_i}{i}$$

$$+ \sum_{s,a \in \mathcal{C}(S,A^h)} \sum_{i=1}^{N^{K,H_h}(s,a)} r(\pi^*_{s,a}) - r(\pi^i_{s,a}) + \kappa \sum_{i=1}^{N^{K,H_h}(s,a)} P(s_{H_l}^{\pi^i_{s,a}} \neq g(s^k_h, a^k_h)) \qquad (\kappa \geq 2H_h H_l)$$

$$= \sum_{s,a \in \mathcal{C}(S,A^h)} \sum_{i=1}^{N^{K,H_h}(s,a)} \frac{UB(\mathcal{R}^i(s,a)) - \mathcal{R}^i(s,a) + \kappa\psi_i}{i} + \sum_{s,a \in \mathcal{C}(S,A^h)} \mathcal{R}^{N^{K,H_h}(s,a)}$$

(using the definition for sub-MDP regret)

$$\leq \sum_{s,a \in \mathcal{C}(S,A^h)} \sum_{i=1}^{N^{K,H_h}(s,a)} \frac{UB(\mathcal{R}^i(s,a))}{i} + \mathcal{R}^{N^{K,H_h}(s,a)} + \sum_{s,a \in \mathcal{C}(S,A^h)} \sum_{i=1}^{N^{K,H_h}(s,a)} \frac{\kappa\psi_i}{i}$$

$$\leq \sum_{s,a \in \mathcal{C}(S,A^h)} \sum_{i=1}^{N^{K,H_h}(s,a)} \frac{UB(\mathcal{R}^i(s,a))}{i} + UB(\mathcal{R}^{N^{K,H_h}(s,a)}) + \sum_{s,a \in \mathcal{C}(S,A^h)} O(\kappa\sqrt{N^{K,H_h}(s,a)})$$

(since Azuma-Hoeffding is s.t $\psi_i = O(\sqrt{i})$)

$$\leq \sum_{s,a \in \mathcal{C}(S,A^h)} \sum_{i=1}^{N^{K,H_h}(s,a)} \frac{UB(\mathcal{R}^{N^{K,H_h}(s,a)})}{i} + UB(\mathcal{R}^{N^{K,H_h}(s,a)}) + \sum_{s,a \in \mathcal{C}(S,A^h)} O(H^h H^l \sqrt{N^{K,H_h}(s,a)})$$

(using monotonicity of upper bound $UB(\mathcal{R}^i(s,a))$ in i, assumption that $C = O(H_h H_l)$)

$$= \sum_{s,a \in \mathcal{C}(S,A^h)} (\log(N^{K,H_h}(s,a)) + 1) UB(\mathcal{R}^{N^{K,H_h}(s,a)}) + O(H^h H^l \sqrt{N^{K,H_h}(s,a)})$$

□

Corollary 12 (Regret under $|\mathcal{C}(S, A^h)|$ clusters of isomorphic sub-MDPs [291]). *Let us set UCB-VI to be the sub-MDP learning algorithm, then we have the following regret bound:*

$$\sum_{s,a \in \mathcal{C}(S,A^h)} (\log(N^{K,H_h}(s,a)) + 1)\mathcal{R}^{N^{K,H_h}(s,a)} + O(H^h H^l \sqrt{N^{K,H_h}(s,a)})$$

$$\leq (\log H^h K + 1) \sum_{s,a \in \mathcal{C}(S,A^h)} \mathcal{R}^{N^{K,H_h}(s,a)} + O(H^h H^l \sqrt{|\mathcal{C}(S,A^h)| \cdot H^h K})$$

$(\sum_{s,a \in \mathcal{C}(S,A^h)} N^{K,H_h}(s,a) = H^h K)$

$$\leq (\log H^h K + 1) \sum_{s,a \in \mathcal{C}(S,A^h)} H_l^{3/2} \sqrt{|S^l_{s,a}||A| N^{K,H_h}(s,a)} + O(H^h H^l \sqrt{|\mathcal{C}(S,A^h)| \cdot H^h K})$$

(plug in UCB-VI guarantees)

$$\leq \tilde{O}(H_l^{3/2} \sqrt{\max_{s,a} |S^l_{s,a}||A|} \sqrt{|\mathcal{C}(S,A^h)|(H_h K)} + H_h H_l \sqrt{|\mathcal{C}(S,A^h)| H_h K})$$

$(\sum_{s,a \in \mathcal{C}(S,A^h)} N^{K,H_h}(s,a) = H^h K)$

where we use UCB-VI's guarantee that upper bound $UB(\mathcal{R}^{N^{K,H_h}(s,a)}) = H_l^{3/2}\sqrt{|S^l_{s,a}||A|N^{K,H_h}(s,a)}$.

Remark 28 (High Probability Bound)**.** *For completeness, we show that the regret bound holds with probability greater than* $1-\delta$*. The regret bound holds under* $\bigcap_{s,a,n}\mathcal{E}^n_{s,a}\cap\mathcal{E}^\zeta\cap\mathcal{E}^\rho$*, by union bound:*

$$
\begin{aligned}
&\Pr(\bigcap_{s,a,n}\mathcal{E}^n_{s,a}\cap\mathcal{E}^\zeta\cap\mathcal{E}^\rho)\\
&\geq 1-\sum_{s,a,n}\Pr(\neg\mathcal{E}^n_{s,a})-\Pr(\neg\mathcal{E}^\zeta)-\Pr(\neg\mathcal{E}^\rho))\\
&\geq 1-(|\mathcal{C}(S,A^h)|H_hK)\frac{\delta}{3|\mathcal{C}(S,A^h)|H_hK}-\delta/3-\delta/3\\
&=1-\delta
\end{aligned}
$$

11.9 Proofs for Section 11.4

11.9.1 Low-level Feedback is insufficient for learning

To prove the results below, our approach is to construct two MDP instances with identifical low level feedback such that any deterministic learning algorithm picks the arbitrarily worse high level policy.

Proposition 58 (Non-identifiability of ranking among sub-MDP returns)**.** *For any deterministic high-level policy learning algorithm with* N_l *samples of low-level feedback, there exists a MDP instance that induces regret constant in* N_l*.*

Proof. Consider two-horizon MDP with starting state s_1 with $H_h=1$, $H_l=2$. There are two possible high-level actions a_1 and a_2 at s_1.

For any policy π^1 in sub-MDP $M(s_1,a_1)$, let it have feature expectation $\phi(\pi^1)=[\phi'(\pi^1),1,0]$, and for any π^2 in sub-MDP $M(s_1,a_2)$, $\phi(\pi^2)=[\phi'(\pi^2),0,1]$.

Now, we consider two MDP instances with $\theta^*=[0,0,C']$ and $\theta^*=[0,C',0]$ for some positive constant C'.

Under both instances, we observe identical low-level feedback for trajectories τ,τ' in sub-MDPs $M(s_1,a_j)$, $j\in[2]$: the feedback is Bernoulli with parameter $\sigma(\langle\phi'(\tau)-\phi'(\tau),\theta'\rangle)$.

Consider any deterministic learning algorithm. WLOG it outputs high level policy $\pi^h(s_1)=a_1$ with some set of N_l samples of low-level feedback.

Then, it follows that its regret under $\theta^*=[\epsilon 1,0,C']$ is C', since the reward (and return since $H_h=1$) of $\pi^*_{s_1,a_1}$ is 0, while the reward of the optimal policy which visits $M(s_1,a_2)$ is C'.

□

11.9.2 Hierarchical Experiment Design via REGIME [317]

11.9.2.1 MLE Definition:

We first define the MLE expression; note that the MLE is in terms of trajectories only. Define:

$$f(\{y_i\}_{i=1}^n, \{x_i\}_{i=1}^n) = -\sum_{i=1}^n \log(\mathbb{1}\{y_i = 1\}\,\sigma(\theta^T x_i) + \mathbb{1}\{y_i = 0\}\,(1 - \sigma(\theta^T x_i))$$

$$\ell_D(\theta) = f(\{y_i\}_{i=1}^{N_h}, \{x_i\}_{i=1}^n) + \sum_{s,a} f(\{y_i^{s,a}\}_{i=1}^{N_l}, \{x_i^{s,a}\}_{i=1}^{N_l})$$

- **High-level trajectories:** has realized features,

  $$x_i = \phi^{\pi^{N_l},P}(\tau_1^i) - \phi^{\pi^{N_l},P}(\tau_2^i) = \sum_{j=1}^{H_h} \phi^P(\pi^{N_l}(s_j^{\tau_1^i}, a_j^{\tau_1^i})) - \sum_{j=1}^{H_h} \phi^P(\pi^{N_l}(s_j^{\tau_2^i}, a_j^{\tau_2^i}))$$

  where $\phi^{\pi^{N_l},P}(\tau_j^i)$ is the feature of the high-level trajectory under sub-policy π^{N_l} and transition P (since trajectories are collected from roll-outs in the actual MDP as in [317]).

  On the other hand, under idealized-feedback, the labeler assumes that each goal-conditioned sub-MDP has been executed perfectly (i.e. by $\pi_{s,a}^*$) and so the features correspond to:

  $$x_i^* = \phi^{\pi^*,P}(\tau_1^i) - \phi^{\pi^*,P}(\tau_2^i) = \sum_{j=1}^{H_h} \phi^P(\pi^*(s_j^{\tau_1^i}, a_j^{\tau_1^i})) - \sum_{j=1}^{H_h} \phi^P(\pi^*(s_j^{\tau_2^i}, a_j^{\tau_2^i}))$$

- Comparison y of high level trajectories follows Bernoulli distribution $y_i = \sigma(\theta^* \cdot x_i^*)$.
- **Low-level trajectories:** has realized features,

  $$x_i^{s,a} = \phi(\tau_1^i) - \phi(\tau_2^i) = \sum_{j=1}^{H_h} \phi(s_j^{\tau_1^i}, a_j^{\tau_1^i}) - \sum_{j=1}^{H_h} \phi(s_j^{\tau_2^i}, a_j^{\tau_2^i})$$

  Note that unlike the high level features, low-level features data are always unbiased. Thus, using high level and low-level comparisons has the same bias from the high level.
- Comparison y of low level trajectories follows Bernoulli distribution $y_i = \sigma(\theta^* \cdot x_i^{s,a})$.

11.9.2.2 Requisite Lemmas

Lemma 88 (Lemma 5 of [317]). *Let oracle $P^{\epsilon'}$ be such that with probability $1-\delta/5$, the following holds. Let $d_h^\pi(s,a)$ and $\hat{d}_h^\pi(s,a)$ be the visitation measure of policy π under P and $P^{\epsilon'}$, we have for all $h \in [H]$ and $\pi \in \Pi$:*

$$\sum_{s,a} |d_h^\pi(s,a) - \hat{d}_h^\pi(s,a)| = \sum_s |d_h^\pi(s) - \hat{d}_h^\pi(s)| \leq h\epsilon'$$

This applies across all sub-MDPs $M(s,a)$. Let the event that this expression hold be $\mathcal{E}^{s,a}$.

Lemma 89 (Low-level MLE Bound, Lemma 2 of [317]). *With probability at least* $1-\delta/5$*:*

$$\|\theta^* - \theta^t\|_{\tilde{\Sigma}_n^l} \leq \tilde{O}(1)$$

Let the event that this holds for learning from sub-MDP trajectories be $\mathcal{E}_1^l$*.*

Lemma 90 (Lemma 3 of [317]). *If low-leve trajectories* $\tau_i^{1,2} \sim \pi^i, P^{\epsilon'}$*, then with probability at least* $1-\delta/5$*:*

$$\|\theta^* - \theta^t\|_{\hat{\Sigma}_n^l} \leq \sqrt{2}\|\theta^* - \theta^t\|_{\tilde{\Sigma}_n^l} + O(B\sqrt{d\log 4n/\delta}W)$$

Let the event that this holds for learning from sub-MDP trajectories be $\mathcal{E}_2^l$*.*

11.9.2.3 Bias when using idealized-feedback, high level trajectory data in MLE

Proposition 59 (sub-MDP REGIME guarantee of [317]). *For sub-MDP* $M(s,a)$*, under* $\mathcal{E}^{s,a} \cap \mathcal{E}_1^l \cap \mathcal{E}_2^l$*:*

$$\langle \phi^P(\pi^*), \theta^* \rangle - \langle \phi^P(\pi^{N_l}), \theta^* \rangle \leq \frac{C_1(\delta)}{\sqrt{N_l}} + O(\epsilon')$$

where $C_1(\delta) = O(\sqrt{\log(1/\delta)})$.

Note that for estimation and bias, we have to have both an upper bound and a lower bound (see PbRL example). This requires two-sided bound, where lower bound comes from ϕ^* having higher reward than ϕ and upper bound comes from no-regret. Due to optimality of π^*, we have the lower bound as well:

$$0 \leq \langle \phi^P(\pi^*), \theta^* \rangle - \langle \phi^P(\pi^{N_l}), \theta^* \rangle \leq \frac{C_1}{\sqrt{N_l}} + O(\epsilon')$$

Additionally, we have that:

Lemma 91 (Lemma 6 of [317]). *For any* s_h, a_h*,* $\|v_i\| \leq 2B$*,* $\theta \in \mathbb{R}^d$ *and* $\|\phi\| \leq R$ *under* $\mathcal{E}^{s,a} \cap \mathcal{E}_1^l \cap \mathcal{E}_2^l$*:*

$$|\langle \phi^{P^{\epsilon'}}(\pi^{N_l}(s_h, a_h)) - \phi^P(\pi^{N_l}(s_h, a_h)), v \rangle| \leq BRd^2\epsilon'$$

With this,

$$|\langle \phi^P(\pi^*), \theta^* \rangle - \phi^{P^{\epsilon'}}(\pi^{N_l}), \theta^*| \leq (\frac{C_1}{\sqrt{N_l}} + O(\epsilon')) + BRd^2\epsilon' = \frac{C_1}{\sqrt{N_l}} + C_2\epsilon'$$

Now, we can analyze the bias of including high level trajectory data in the MLE computation:

Lemma 92. *Suppose there are* N_h, N_l *high, low-level trajectories, bias* b *is such that, under* $\bigcap_{s,a} \mathcal{E}^{s,a} \cap \mathcal{E}_1^l \cap \mathcal{E}_2^l$*:*

$$\|b\|^2 = \sum_{t=1}^{T} |\langle \theta^*, x_i \rangle - \langle \theta^*, x_i^* \rangle|^2 \leq 2H_hT(2H_h(\frac{C_1}{\sqrt{N_l}} + C_2\epsilon')^2)$$

Proof.

$$\begin{aligned}
&\sum_{t=1}^{T} |\langle \theta^*, x_i^* \rangle - \langle \theta^*, x_i \rangle|^2 \\
&\leq 2\sum_{t=1}^{T} |\langle \sum_{s,a \in \tau_1^t} \phi^P(\pi^*(s,a)) - \sum_{s,a \in \tau_1^t} \phi^{P^{\epsilon'}}(\pi^{N_l}(s,a)), \theta^* \rangle|^2 + |\langle \sum_{s,a \in \tau_2^t} \phi^P(\pi^*(s,a)) - \sum_{s,a \in \tau_2^t} \phi^{P^{\epsilon'}}(\pi^{N_l}(s,a)), \theta^* \rangle|^2 \\
&\leq 2H_h \sum_{t=1}^{T} \sum_{s,a \in \tau_1^t} |\langle \phi^P(\pi^*(s,a)) - \phi^{P^{\epsilon'}}(\pi^{N_l}(s,a)), \theta^* \rangle|^2 + \sum_{s,a \in \tau_2^t} |\langle \phi^P(\pi^*(s,a)) - \phi^{P^{\epsilon'}}(\pi^{N_l}(s,a)), \theta^* \rangle|^2 \\
&\leq 2H_h T(2H_h(\frac{C_1}{\sqrt{N_l}} + C_2\epsilon')^2)
\end{aligned}$$

Thus,

$$\|b\| = \sqrt{\sum_{t=1}^{T} |\langle \theta^*, x_i \rangle - \langle \theta^*, x_i^* \rangle|^2} \leq 2H_h(\frac{C_1}{\sqrt{N_l}} + C_2\epsilon')\sqrt{T}$$

□

11.9.2.4 MLE Analysis

Under current-feedback, following Lemma 2 of [317], $\|\Delta\|_{\Sigma_n^h + \lambda I} \leq \tilde{O}(1)$. Now, we consider the bias in learned reward under idealized-feedback.

Proposition 60. *Let* $\theta_{MLE} = \operatorname{argmin}_\theta \ell_D(\theta)$ *and let* $C_b \geq \|b\|$. *Then with probability at least* $1 - \delta/5$*:*

$$\|\Delta\|_{\Sigma_n + \lambda I} \leq O\left(\sqrt{\frac{C_b}{\gamma^2\sqrt{n}} + \frac{C_b^2 + d + \log(1/\delta)}{\gamma^2 n} + \lambda B^2}\right)$$

where $\Sigma_n = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T + \lambda I$.

Proof. Define $\Delta = \theta_{MLE} - \theta^*$. As in [325], we have the same convexity result due to $\langle \theta, x_i \rangle \in [-2LB, 2LB]$. Suppose we let $\max_x \|x\| \leq L$ and $\max_{\theta \in \Theta} \|\theta\| \leq B$, then with $\gamma = \frac{1}{2+\exp(-2LB)+\exp(2LB)}$, we have that:

$$\ell(\theta^* + \Delta) - \ell(\theta^*) - \langle \nabla\ell(\theta^*), \Delta \rangle \geq \gamma\|\Delta\|_\Sigma^2$$

And so,

$$\ell(\theta_{MLE}) \leq \ell(\theta^*) \Rightarrow \ell(\theta^* + \Delta) - \ell(\theta^*) - \langle \nabla\ell(\theta^*), \Delta \rangle \leq -\langle \nabla\ell(\theta^*), \Delta \rangle$$

Thus,

$$\gamma\|\Delta\|_\Sigma^2 \leq \|\nabla\ell(\theta^*)\|_{(\Sigma+\lambda I)^{-1}} \|\Delta\|_{(\Sigma+\lambda I)}$$

The key part is bounding $\|\nabla \ell(\theta^*)\|_{(\Sigma+\lambda I)^{-1}}$. We have that:

$$\begin{aligned}
\nabla \ell(\theta^*) &= -\frac{1}{n}\sum_{i=1}^{n}[\mathbb{1}\left\{y_i = 1\right\}\sigma(\langle\theta^*, x_i\rangle) - \mathbb{1}\left\{y_i = 0\right\}(1 - \sigma(\langle\theta^*, x_i\rangle)]x_i \\
&= -\frac{1}{n}X^T(V + b)
\end{aligned}$$

where $v_i = \sigma(\langle\theta^*, x_i^*\rangle)$ w.p $1 - \sigma(\langle\theta^*, x_i^*\rangle)$ and $-(1 - \sigma(\langle\theta^*, x_i^*\rangle))$ w.p $\sigma(\langle\theta^*, x_i^*\rangle)$. And so, entry-wise V is such that $\mathbb{E}[V_i] = 0$ and $|V_i| \le 1$. Note that V_i are independent due to the independence of the random variables Y_i.

Extra term bias is defined as:

$$\begin{aligned}
b_i &= \mathbb{1}\left\{y_i = 1\right\}(\sigma(\langle\theta^*, x_i\rangle) - \sigma(\langle\theta^*, x_i^*\rangle)) - \mathbb{1}\left\{y_i = 0\right\}(1 - \sigma(\langle\theta^*, x_i\rangle - (1 - \sigma(\langle\theta^*, x_i^*\rangle))) \\
&= \sigma(\langle\theta^*, x_i\rangle) - \sigma(\langle\theta^*, x_i^*\rangle)
\end{aligned}$$

By definition, C_b is such that: $\|b\| \le C_b$. As before, define $M = \frac{1}{n^2}X(\Sigma + \lambda I)^{-1}X^T$. We use the fact that $\|M\|_{op} \le 1/n$. Then, we have that:

$$\begin{aligned}
\|\nabla \ell(\theta^*)\|^2_{(\Sigma+\lambda I)^{-1}} &= (V + b)^T M (V + b) \\
&= V^T M V + 2V^T M b + b^T M b \\
&\le C\frac{d + \log(1/\delta)}{n} + 2\|V\|\|Mb\| + b^T M b \\
&\qquad \text{(by Matrix Bernstein, } V^T M V \le C\tfrac{d+\log(10/\delta)}{n} \text{ w.p. } \ge 1 - \delta/10) \\
&\le C\frac{d + \log(1/\delta)}{n} + 2\|V\|\frac{1}{n}\|b\| + \frac{C_b^2}{n} \qquad \text{(using that } \|M\|_{op} \le 1/n) \\
&\le C\frac{d + \log(1/\delta)}{n} + 2(C_2\sqrt{n}\frac{1}{n})C_b + \frac{C_b^2}{n} \\
&\qquad \text{(by Hoeffding } \|V\| \le O(\log(10/\delta)\sqrt{n}) \text{ w.p. } \ge 1 - \delta/10.) \\
&\le O(\frac{C_b}{\sqrt{n}} + \frac{C_b^2 + d + \log(1/\delta)}{n})
\end{aligned}$$

$$\begin{aligned}
\gamma\|\Delta\|^2_{\Sigma+\lambda I} &\le \|\nabla \ell(\theta^*)\|_{(\Sigma+\lambda I)^{-1}}\|\Delta\|_{(\Sigma+\lambda I)} + \lambda(\gamma\|\Delta\|^2) \\
&\le \|\nabla \ell(\theta^*)\|_{(\Sigma+\lambda I)^{-1}}\|\Delta\|_{(\Sigma+\lambda I)} + 4\lambda\gamma B^2
\end{aligned}$$

This implies that with probability $\ge 1 - \delta$:

$$\|\Delta\|_{\Sigma+\lambda I} \le C\sqrt{\frac{C_b}{\gamma^2\sqrt{n}} + \frac{C_b^2 + d + \log(1/\delta)}{\gamma^2 n} + \lambda B^2}$$

□

Corollary 13. *Let* $\theta_{MLE} = \operatorname{argmin}_\theta \ell_D(\theta)$*, then under* $\bigcap_{s,a} \mathcal{E}^{s,a}$*, with probability* $\geq 1 - \delta/5$*:*

$$\|\theta^* - \theta_{MLE}\|_{\tilde{\Sigma}^h_{N^h} + \lambda I} \leq C\sqrt{\frac{1}{\gamma^2\sqrt{N_l}} + \frac{1}{\gamma^2 N_l} + \frac{d + \log(1/\delta)}{\gamma^2 N_h} + \lambda B^2}$$

where $\Sigma_{N_h} = \frac{1}{N_h}\sum_{i=1}^{N_h} x_i x_i^T$*.*
Let the event that this holds for learning from sub-MDP trajectories be $\mathcal{E}_1^h$*.*

Proof. Firstly,

$$\|b\| \leq 2H_h(\frac{C_1}{\sqrt{N_l}} + C_2\epsilon')\sqrt{N_h} = O(\frac{\sqrt{N_h}}{\sqrt{N_l}} + \sqrt{N_h}\epsilon')$$

With this, we have that:

$$\begin{aligned}
&\|\Delta\|_{\tilde{\Sigma}_{N_h} + \lambda I} \\
&= O\left(\sqrt{\frac{C_b}{\gamma^2\sqrt{N_h}} + \frac{C_b^2 + d + \log(1/\delta)}{\gamma^2 N_h} + \lambda B^2)}\right) \\
&= O\left(\sqrt{\frac{\sqrt{N_h/N_l} + \sqrt{N_h}\epsilon'}{\gamma^2\sqrt{N_h}} + \frac{N_h/N_l + N_h\epsilon'^2 + d + \log(1/\delta)}{\gamma^2 N_h} + \lambda B^2}\right)
\end{aligned}$$

□

Hence by choosing $\lambda = \lambda/N_h$:

$$\|\Delta\|_{\tilde{\Sigma}_{N_h} + \lambda I} \leq O\left(\frac{N_h^{1/2}}{N_l^{1/4}} + (N_h\epsilon')^{1/2}\right) + C'$$

11.9.2.5 Relating $\|\theta^* - \theta^n\|_{\hat{\Sigma}_n}$ to $\|\theta^* - \theta^n\|_{\tilde{\Sigma}_n}$

Define:

1. $\Sigma_n = \lambda I + \sum_{i=1}^n(\phi^{\pi^{N_l},P}(\pi_1^i) - \phi^{\pi^{N_l},P}(\pi_2^i))(\phi^{\pi^{N_l},P}(\pi_1^i) - \phi^{\pi^{N_l},P}(\pi_2^i))^T$
2. $\tilde{\Sigma}_n = \lambda I + \sum_{i=1}^n(\phi(\tau_1^i) - \phi(\tau_2^i))(\phi(\tau_1^i) - \phi(\tau_2^i))^T$, where $\tau_i^{1,2} \sim \pi_1^i, \pi^{N_l}, P$.
3. $\hat{\Sigma}_n = \lambda I + \sum_{i=1}^n(\phi^{\pi^{N_l},P^{\epsilon'}}(\pi_1^i) - \phi^{\pi^{N_l},P^{\epsilon'}}(\pi_2^i))(\phi^{\pi^{N_l},P^{\epsilon'}}(\pi_1^i) - \phi^{\pi^{N_l},P^{\epsilon'}}(\pi_2^i))^T$

We wish to relate $\|\theta^* - \theta^n\|_{\hat{\Sigma}_n}$ to $\|\theta^* - \theta^n\|_{\tilde{\Sigma}_n}$.

Lemma 93 (Lemma 3 of [317])**.** *If* $\tau_i^{1,2} \sim \pi_1^i, \pi^{N_l}, P^{\epsilon'}$*, then with probability at least* $1 - \delta/5$*:*

$$\|\theta^* - \theta^t\|_{\hat{\Sigma}_n^h} \leq \sqrt{2}\|\theta^* - \theta^t\|_{\tilde{\Sigma}_n^h} + \tilde{O}(B\sqrt{d\log 4n/\delta}W)$$

Let the event that this holds for learning from sub-MDP trajectories be $\mathcal{E}_2^h$*.*

Lemma 94. *We have that under $\bigcap_{s,a} \mathcal{E}^{s,a} \cap \mathcal{E}_1^l \cap \mathcal{E}_2^l \cap \mathcal{E}_1^h \cap \mathcal{E}_2^h$:*

$$\|\theta^* - \theta^n\|_{\hat{\Sigma}_n} \leq 2\|\theta^* - \theta^n\|_{\tilde{\Sigma}_n} + O(B\sqrt{d \log n/\delta} W) + \sqrt{8n} C(\epsilon', \delta)$$

Proof. Under event $\mathcal{E}_2^h$, as trajectories are sampled from P, we have that:

$$\|\theta^* - \theta^n\|_{\Sigma_n} \leq \sqrt{2}\|\theta^* - \theta^n\|_{\tilde{\Sigma}_n} + O(B\sqrt{d \log n/\delta} W)$$

It remains to upper bound $\|\theta^* - \theta^n\|_{\hat{\Sigma}_n}$ by $\|\theta^* - \theta^n\|_{\Sigma_n}$

We have that under $\bigcap_{s,a} \mathcal{E}^{s,a} \cap \mathcal{E}_1^l \cap \mathcal{E}_2^l$:

$$\begin{aligned}
&|\langle \phi^{\pi^{N_l},P}(\pi) - \phi^{\pi^{N_l},P^{\epsilon'}}(\pi), v\rangle| \leq C(\epsilon', \delta) \\
&\Rightarrow |\langle \phi^{\pi^{N_l},P^{\epsilon'}}(\pi_1^i) - \phi^{\pi^{N_l},P^{\epsilon'}}(\pi_2^i), v\rangle| \leq |\langle \phi^{\pi^{N_l},P}(\pi_1^i) - \phi^{\pi^{N_l},P}(\pi_2^i), v\rangle| + 2C(\epsilon', \delta) \\
&\Rightarrow |\langle \phi^{\pi^{N_l},P^{\epsilon'}}(\pi_1^i) - \phi^{\pi^{N_l},P^{\epsilon'}}(\pi_2^i), v\rangle|^2 \leq 2|\langle \phi^{\pi^{N_l},P}(\pi_1^i) - \phi^{\pi^{N_l},P}(\pi_2^i), v\rangle|^2 + 2(2C(\epsilon', \delta))^2
\end{aligned}$$

Thus,

$$\begin{aligned}
&\|v\|_{\hat{\Sigma}_n}^2 \\
&= v^T(\lambda I + \sum_{i=1}^n (\phi^{\pi^{N_l},P^{\epsilon'}}(\pi_1^i) - \phi^{\pi^{N_l},P^{\epsilon'}}(\pi_2^i))(\phi^{\pi^{N_l},P^{\epsilon'}}(\pi_1^i) - \phi^{\pi^{N_l},P^{\epsilon'}}(\pi_2^i))^T)v \\
&= \lambda\|v\|^2 + \sum_{i=1}^n |\langle \phi^{\pi^{N_l},P^{\epsilon'}}(\pi_1^i) - \phi^{\pi^{N_l},P^{\epsilon'}}(\pi_2^i), v\rangle|^2 \\
&\leq \lambda\|v\|^2 + \sum_{i=1}^n 2|\langle \phi^{\pi^{N_l},P}(\pi_1^i) - \phi^{\pi^{N_l},P}(\pi_2^i), v\rangle|^2 + 8C(\epsilon', \delta)^2 \\
&\leq 2\|v\|_{\Sigma_n}^2 + 8nC(\epsilon', \delta)^2
\end{aligned}$$

Plugging in $v = \theta^* - \theta^n$, we have that:

$$\begin{aligned}
&\|\theta^* - \theta^n\|_{\hat{\Sigma}_n} \\
&\leq \sqrt{2}\|\theta^* - \theta^n\|_{\Sigma_n} + \sqrt{8n} C(\epsilon', \delta) \\
&\leq 2\|\theta^* - \theta^n\|_{\tilde{\Sigma}_n} + O(B\sqrt{d \log n/\delta} W) + \sqrt{8n} C(\epsilon', \delta)
\end{aligned}$$

□

11.9.2.6 High-level policy regret bound

Lemma 95. *For any π, under event $\bigcap_{s,a} \mathcal{E}^{s,a} \cap \mathcal{E}_1^l \cap \mathcal{E}_2^l$:*

$$\langle \phi^{\pi^*,P}(\pi) - \phi^{\pi^{N_l},P}(\pi), \theta^* \rangle \leq H_h(\frac{C_1}{\sqrt{N_l}} + C_2\epsilon')$$

Proof.

$$
\begin{aligned}
&\langle \phi^{\pi^*,P}(\pi) - \phi^{\pi^{N_l},P}(\pi), \theta^* \rangle \\
&= \sum_{h=1}^{H_h} \mathbb{E}_{s_h,a_h \sim \pi,\pi^{N_l},P} \mathbb{E}_{s_{h+1} \sim \pi^{N_l}(s_h,a_h),P} [r(\pi^*(s_h,a_h)) + V_{h+1}^{\pi,\pi^*}(g(s_h,a_h)) - (r(\pi^{N_l}(s_h,a_h)) + V_{h+1}^{\pi,\pi^{N_l}}(s_{h+1}))] \\
&= \sum_{h=1}^{H_h} \mathbb{E}_{s_h,a_h \sim \pi,\pi^{N_l},P} [r(\pi^*(s_h,a_h)) - r(\pi^{N_l}(s_h,a_h)) + P(s_{h+1}^{\pi^{N_l}} \neq g(s_h,a_h))(V_{h+1}^{\pi,\pi^*}(g(s_h,a_h)) - V_{h+1}^{\pi,\pi^{N_l}}(s_{h+1}))] \\
&\leq \sum_{h=1}^{H_h} \mathbb{E}_{s_h,a_h \sim \pi,\pi^{N_l},P} [r(\pi^*(s_h,a_h)) - r(\pi^{N_l}(s_h,a_h)) + P(s_{h+1}^{\pi^{N_l}} \neq g(s_h,a_h)) \kappa H_h H_l] \\
&= \sum_{h=1}^{H_h} \mathbb{E}_{s_h,a_h \sim \pi,\pi^{N_l},P} [r(\pi^*(s_h,a_h)) + P(s_{h+1}^{\pi^*} = g(s_h,a_h)) \kappa H_h H_l - r(\pi^{N_l}(s_h,a_h)) - P(s_{h+1}^{\pi^{N_l}} = g(s_h,a_h)) \kappa H_h \\
&= \sum_{h=1}^{H_h} \mathbb{E}_{s_h,a_h \sim \pi,\pi^{N_l},P} [\langle \phi(\pi^*(s_h,a_h)), \theta^* \rangle - \langle \phi(\pi^{N_l}(s_h,a_h)), \theta^* \rangle] \\
&\leq H_h(\frac{C_1}{\sqrt{N_l}} + C_2 \epsilon')
\end{aligned}
$$

Because for any s_h, a_h, $\langle \phi(\pi^*(s_h,a_h)), \theta^* \rangle - \langle \phi(\pi^{N_l}(s_h,a_h)), \theta^* \rangle \leq \frac{C_1}{\sqrt{N_l}} + C_2 \epsilon'$. □

Lemma 96 (Lower bound on Reachability Probability). *We have that under event $\bigcap_{s,a} \mathcal{E}^{s,a} \cap \mathcal{E}_1^l \cap \mathcal{E}_2^l$:*

$$P(s_{H_l}^{\pi^{N_l}} \neq g(s,a)) \leq \frac{1}{\kappa H_h} + \frac{C_1}{\kappa H_h H_l \sqrt{N_l}} + \frac{C_2 \epsilon'}{\kappa H_h H_l}$$

and

$$P^{\epsilon'}(s_{H_l}^{\pi^{N_l}} \neq g(s,a)) \leq \frac{1}{\kappa H_h} + \frac{C_1}{\kappa H_h H_l \sqrt{N_l}} + \frac{C_2 \epsilon'}{\kappa H_h H_l} + H_l \epsilon'$$

Proof. Due to the regret guarantee, we have that:

$$
\begin{aligned}
&\frac{C_1}{\sqrt{N_l}} + C_2 \epsilon' \\
&\geq \langle \phi^P(\pi^*) - \phi^P(\pi^{N_l}), \theta^* \rangle \\
&= r(\pi^*) + \kappa H_h H_l \cdot 1 - r(\pi^{N_l}) - \kappa H_h H_l \cdot P(s_{H_l}^{\pi^{N_l}} = g(s,a)) \\
&\geq 0 - H_l + \kappa H_h H_l \cdot P(s_{H_l}^{\pi^{N_l}} \neq g(s,a))
\end{aligned}
$$

Thus, we have that:

$$P(s_{H_l}^{\pi^{N_l}} \neq g(s,a)) \leq \frac{1}{\kappa H_h} + \frac{C_1}{\kappa H_h H_l \sqrt{N_l}} + \frac{C_2 \epsilon'}{\kappa H_h H_l}$$

Additionally, we have that from Lemma 5.1:

$$|d_{H_l}^{\pi^{N_l}}(g(s,a)) - \hat{d}_{H_l}^{\pi^{N_l}}(g(s,a))| = |P(s_{H_l}^{\pi^{N_l}} \neq g(s,a)) - P^{\epsilon'}(s_{H_l}^{\pi^{N_l}} \neq g(s,a))| \leq H_l \epsilon'$$

Thus,

$$P^{\epsilon'}(s_{H_l}^{\pi^{N_l}} \neq g(s,a)) \leq \frac{1}{\kappa H_h} + \frac{C_1}{\kappa H_h H_l \sqrt{N_l}} + \frac{C_2 \epsilon'}{\kappa H_h H_l} + H_l \epsilon'$$

□

Lemma 97 (use of the Elliptical Lemma)**.**

$$\langle \phi^{\pi^{N_l}, P^{\epsilon'}}(\pi^*) - \phi^{\pi^{N_l}, P^{\epsilon'}}(\hat{\pi}), \theta^* - \hat{\theta} \rangle \leq \frac{1}{\sqrt{N_h}}(2d \log(1 + \frac{N_h}{d})) \|\theta^* - \hat{\theta}\|_{\hat{\Sigma}_{N_h}}$$

Proof.

$$\begin{aligned}
&\langle \phi^{\pi^{N_l}, P^{\epsilon'}}(\pi^*) - \phi^{\pi^{N_l}, P^{\epsilon'}}(\hat{\pi}), \theta^* - \hat{\theta} \rangle \\
&\leq \|\phi^{\pi^{N_l}, P^{\epsilon'}}(\pi^*) - \phi^{\pi^{N_l}, P^{\epsilon'}}(\hat{\pi})\|_{\hat{\Sigma}_{N_h}^{-1}} \|\theta^* - \hat{\theta}\|_{\hat{\Sigma}_{N_h}} \\
&\leq \frac{1}{N_h} \sum_{i=1}^{N_h} \|\phi^{\pi^{N_l}, P^{\epsilon'}}(\pi^*) - \phi^{\pi^{N_l}, P^{\epsilon'}}(\hat{\pi})\|_{\hat{\Sigma}_i^{-1}} \|\theta^* - \hat{\theta}\|_{\hat{\Sigma}_{N_h}} && (\hat{\Sigma}_{N_h}^{-1} \preceq \hat{\Sigma}_i^{-1}) \\
&\leq \frac{1}{N_h} \sum_{i=1}^{N_h} \|\phi^{\pi^{N_l}, P^{\epsilon'}}(\pi_1^i) - \phi^{\pi^{N_l}, P^{\epsilon'}}(\pi_2^i)\|_{\hat{\Sigma}_i^{-1}} \|\theta^* - \hat{\theta}\|_{\hat{\Sigma}_{N_h}} && (\text{definition of } \pi_{1,2}^i) \\
&\leq \frac{1}{\sqrt{N_h}} \sqrt{\sum_{i=1}^{N_h} \|\phi^{\pi^{N_l}, P^{\epsilon'}}(\pi_1^i) - \phi^{\pi^{N_l}, P^{\epsilon'}}(\pi_2^i)\|_{\hat{\Sigma}_i^{-1}}^2} \|\theta^* - \hat{\theta}\|_{\hat{\Sigma}_{N_h}} \\
&\leq \frac{1}{\sqrt{N_h}}(2d \log(1 + \frac{N_h}{d})) \|\theta^* - \hat{\theta}\|_{\hat{\Sigma}_{N_h}} && (\text{Elliptical Lemma})
\end{aligned}$$

□

Define goal non-reachability probability to be: $\delta = \frac{1}{\kappa H_h} + \frac{C_1}{\kappa H_h H_l \sqrt{N_l}} + \frac{C_2 \epsilon'}{\kappa H_h H_l} + H_l \epsilon'$.

Lemma 98. *Let $\Phi^{\pi^{N_l}, P^{\epsilon'}}(\pi)$ denote the feature expectation under high level policy π, sub-MDP policies π^{N_l} and MDP transitions $P^{\epsilon'}$. Under event $\bigcap_{s,a} \mathcal{E}^{s,a} \cap \mathcal{E}_1^l \cap \mathcal{E}_2^l$, we have that, for any high level policy π: $|\langle \phi^{\pi^{N_l}, P}(\pi) - \phi^{\pi^{N_l}, P^{\epsilon'}}(\pi), \theta^* \rangle| \leq 2H_h B R d^2 \epsilon' + 8H_h^3 H_l \delta$.*

Proof. Let $\mathcal{E}_{reach}$ denote the event that roll-out $\tau \sim \pi, \pi^{N_l}, P$ is such that all high level goals are reached, and similarly event $\mathcal{E}'_{reach}$ for roll-out $\tau' \sim \pi, \pi^{N_l}, P^{\epsilon'}$.

By union bound, $\Pr(\neg\mathcal{E}_{reach}) = \Pr(\exists s_i, a_i, s_{H_l}^{\pi^{N_l}(s_i,a_i)} \neq g(s_i, a_i)) \leq \sum_{i=1}^{H_h} \Pr(s_{H_l}^{\pi^{N_l}(s_i,a_i)} \neq g(s_i, a_i)))) \leq H_h\delta$, and similarly $\Pr(\neg\mathcal{E}'_{reach}) \leq H_h\delta$.

$$|\langle \phi^{\pi^{N_l},P}(\pi) - \phi^{\pi^{N_l},P^{\epsilon'}}(\pi), \theta^*\rangle|$$

$$\leq |\mathbb{E}_{\tau\sim\pi,\pi^{N_l},P}[\langle\phi(\tau),\theta^*\rangle|\mathcal{E}_{reach}]\Pr(\mathcal{E}_{reach}) - \mathbb{E}_{\tau\sim\pi,\pi^{N_l},P^{\epsilon'}}[\langle\phi(\tau),\theta^*\rangle|\mathcal{E}'_{reach}]\Pr(\mathcal{E}'_{reach})|$$

$$+ |\mathbb{E}_{\tau\sim\pi,\pi^{N_l},P}[\langle\phi(\tau),\theta^*\rangle|\neg\mathcal{E}_{reach}]\Pr(\neg\mathcal{E}_{reach}) - \mathbb{E}_{\tau\sim\pi,\pi^{N_l},P^{\epsilon'}}[\langle\phi(\tau),\theta^*\rangle|\neg\mathcal{E}'_{reach}]\Pr(\neg\mathcal{E}'_{reach})|$$

$$\leq |\mathbb{E}_{\tau\sim\pi,\pi^{N_l},P}[\langle\phi(\tau),\theta^*\rangle|\mathcal{E}_{reach}]\Pr(\mathcal{E}_{reach}) - \mathbb{E}_{\tau\sim\pi,\pi^{N_l},P^{\epsilon'}}[\langle\phi(\tau),\theta^*\rangle|\mathcal{E}'_{reach}]\Pr(\mathcal{E}'_{reach})| + 2(H_h\delta)(H_hH_l)$$

(since $|\mathbb{E}_{\tau\sim\pi,\pi^{N_l},P}[\langle\phi(\tau),\theta^*\rangle|\neg\mathcal{E}_{reach}]\Pr(\neg\mathcal{E}_{reach})| \leq (H_h\delta)(H_hH_l)$ and likewise the other term)

$$= |\Pr(\mathcal{E}_{reach})\sum_{h=1}^{H_h}\sum_{s_h,a_h} d(s_h,a_h)\mathbb{E}[\langle\phi^P(\pi^{N_l}(s_h,a_h)),\theta^*\rangle|\mathcal{E}_{reach}]$$

$$- \Pr(\mathcal{E}'_{reach})\sum_{h=1}^{H_h}\sum_{s_h,a_h} d(s_h,a_h)\mathbb{E}[\langle\phi^{P^{\epsilon'}}(\pi^{N_l}(s_h,a_h)),\theta^*\rangle|\mathcal{E}'_{reach}]| + 2H_h^2H_l\delta$$

(under goal reachability, high-level state visitation measure $d(s_h,a_h)$ is the same)

$$\leq \sum_{h=1}^{H_h}\sum_{s_h,a_h} d(s_h,a_h)|\Pr(\mathcal{E}_{reach})\mathbb{E}[\langle\phi^P(\pi^{N_l}(s_h,a_h)),\theta^*\rangle|\mathcal{E}_{reach}]$$

$$- \Pr(\mathcal{E}'_{reach})\mathbb{E}[\langle\phi^{P^{\epsilon'}}(\pi^{N_l}(s_h,a_h)),\theta^*\rangle|\mathcal{E}'_{reach}]| + 2H_h^2H_l\delta$$

$$= \sum_{h=1}^{H_h}\sum_{s_h,a_h} d(s_h,a_h)|\Pr(\mathcal{E}_{reach})\mathbb{E}[\langle\phi^P(\pi^{N_l}(s_h,a_h)),\theta^*\rangle|\mathcal{E}_{s_h,a_h reach}]$$

$$- \Pr(\mathcal{E}'_{reach})\mathbb{E}[\langle\phi^{P^{\epsilon'}}(\pi^{N_l}(s_h,a_h)),\theta^*\rangle|\mathcal{E}'_{s_h,a_h reach}]| + 2H_h^2H_l\delta$$

($\mathcal{E}_{s_h,a_h reach}$ is the event that $g(s_h,a_h)$ is reached under π^{N_l}, P)

$$\leq \sum_{h=1}^{H_h}\sum_{s_h,a_h} d(s_h,a_h)\Pr(\mathcal{E}_{reach})|\mathbb{E}[\langle\phi^P(\pi^{N_l}(s_h,a_h)),\theta^*\rangle|\mathcal{E}_{s_h,a_h reach}] - \mathbb{E}[\langle\phi^{P^{\epsilon'}}(\pi^{N_l}(s_h,a_h)),\theta^*\rangle|\mathcal{E}'_{s_h,a_h reach}]|$$

$$+ |(\Pr(\mathcal{E}_{reach}) - \Pr(\mathcal{E}'_{reach}))\mathbb{E}[\langle\phi^{P^{\epsilon'}}(\pi^{N_l}(s_h,a_h)),\theta^*\rangle|\mathcal{E}'_{s_h,a_h reach}]| + 2H_h^2H_l\delta$$

$$\leq \sum_{h=1}^{H_h}\sum_{s_h,a_h} d(s_h,a_h)(|\mathbb{E}[\langle\phi^P(\pi^{N_l}(s_h,a_h)),\theta^*\rangle|\mathcal{E}_{s_h,a_h reach}] - \mathbb{E}[\langle\phi^{P^{\epsilon'}}(\pi^{N_l}(s_h,a_h)),\theta^*\rangle|\mathcal{E}'_{s_h,a_h reach}]|$$

$$+ (H_h\delta)(H_hH_l)) + 2H_h^2H_l\delta \qquad \text{(since } \Pr(\mathcal{E}'_{reach}), \Pr(\mathcal{E}_{reach}) \in [1 - H_h\delta, 1]\text{)}$$

To finish, we will relate the expression to $|\langle\phi^{P^{\epsilon'}}(\pi^{N_l}(s_h,a_h)) - \phi(\pi^{N_l}(s_h,a_h)),\theta^*\rangle|$.

$$
\begin{aligned}
&\leq \sum_{h=1}^{H_h} \sum_{s_h,a_h} d(s_h,a_h) |\mathbb{E}[\langle \phi^P(\pi^{N_l}(s_h,a_h)), \theta^* \rangle | \mathcal{E}_{s_h,a_h reach}] - \mathbb{E}[\langle \phi^{P^{\epsilon'}}(\pi^{N_l}(s_h,a_h)), \theta^* \rangle | \mathcal{E}'_{s_h,a_h reach}]| + 3H_h^3 H_l \delta \\
&= \sum_{h=1}^{H_h} \sum_{s_h,a_h} d(s_h,a_h) | \frac{1}{\Pr(\mathcal{E}_{s_h,a_h reach})} \Pr(\mathcal{E}_{s_h,a_h reach}) \mathbb{E}[\langle \phi^P(\pi^{N_l}(s_h,a_h)), \theta^* \rangle | \mathcal{E}_{s_h,a_h reach}] \\
&- \frac{1}{\Pr(\mathcal{E}'_{s_h,a_h reach})} \Pr(\mathcal{E}'_{s_h,a_h reach}) \mathbb{E}[\langle \phi^{P^{\epsilon'}}(\pi^{N_l}(s_h,a_h)), \theta^* \rangle | \mathcal{E}'_{s_h,a_h reach}]| + 3H_h^3 H_l \delta \\
&\leq \sum_{h=1}^{H_h} \sum_{s_h,a_h} d(s_h,a_h) \frac{1}{\Pr(\mathcal{E}_{s_h,a_h reach})} | \Pr(\mathcal{E}_{s_h,a_h reach}) \mathbb{E}[\langle \phi^P(\pi^{N_l}(s_h,a_h)), \theta^* \rangle | \mathcal{E}_{s_h,a_h reach}] \\
&- \Pr(\mathcal{E}'_{s_h,a_h reach}) \mathbb{E}[\langle \phi^{P^{\epsilon'}}(\pi^{N_l}(s_h,a_h)), \theta^* \rangle | \mathcal{E}'_{s_h,a_h reach}]| + H_h \left((\frac{1}{1-\delta} - 1) H_h H_l \right) + 3H_h^3 H_l \delta \\
& \qquad (\diamond) \\
&\leq \sum_{h=1}^{H_h} \sum_{s_h,a_h} d(s_h,a_h) \frac{1}{1-\delta} | \Pr(\neg\mathcal{E}_{s_h,a_h reach}) \mathbb{E}[\langle \phi^P(\pi^{N_l}(s_h,a_h)), \theta^* \rangle | \neg\mathcal{E}_{s_h,a_h reach}] \\
&- \Pr(\neg\mathcal{E}'_{s_h,a_h reach}) \mathbb{E}[\langle \phi^{P^{\epsilon'}}(\pi^{N_l}(s_h,a_h)), \theta^* \rangle | \neg\mathcal{E}'_{s_h,a_h reach}]| + \\
&|\mathbb{E}[\langle \phi^{P^{\epsilon'}}(\pi^{N_l}(s_h,a_h)) - \phi^P(\pi^{N_l}(s_h,a_h)), \theta^* \rangle]| + 4H_h^3 H_l \delta \qquad \text{(using that } \tfrac{1}{1-\delta} - 1 \leq 1) \\
&\leq \sum_{h=1}^{H_h} \sum_{s_h,a_h} d(s_h,a_h) \frac{1}{1-\delta} \left(2(\delta)(H_h H_l) + BRd^2\epsilon' \right) + 4H_h^3 H_l \delta \qquad (\diamond\diamond) \\
&\leq \sum_{h=1}^{H_h} \sum_{s_h,a_h} d(s_h,a_h) 2 \left(2H_h H_l \delta + BRd^2\epsilon' \right) + 4H_h^3 H_l \delta \qquad (\tfrac{1}{1-\delta} \leq 2) \\
&\leq 2H_h BRd^2\epsilon' + 8H_h^3 H_l \delta = C(\epsilon', \delta)
\end{aligned}
$$

$(\diamond) : |\frac{\Pr(\mathcal{E}'_{s_h,a_h reach})}{\Pr(\mathcal{E}_{s_h,a_h reach})} - 1| \leq \max(1 - (1-\delta)\frac{1}{1-\delta} - 1)$ since $\Pr(\mathcal{E}'_{s_h,a_h reach}), \Pr(\mathcal{E}_{s_h,a_h reach}) \in [1-\delta, 1]$.

$(\diamond\diamond) : |\langle \phi^{P^{\epsilon'}}(\pi^{N_l}(s_h,a_h)) - \phi^P(\pi^{N_l}(s_h,a_h)), v \rangle| \leq BRd^2\epsilon'$ and $\Pr(\neg\mathcal{E}_{s_h,a_h reach}), \Pr(\neg\mathcal{E}'_{s_h,a_h reach}) \in [0, \delta]$

□

Theorem 69 (Main regret bound). *We have that under event* $\bigcap_{s,a} \mathcal{E}^{s,a} \cap \mathcal{E}_1^l \cap \mathcal{E}_2^l \cap \mathcal{E}_1^h \cap \mathcal{E}_2^h$ *and* $N_h > 0$*:* $V^{\pi^*,\pi^*} - V^{\hat{\pi},\pi^{N_l}} \leq \tilde{O}\left(N_l^{-1/2} + N_h^{-1/2} \|\theta^* - \hat{\theta}\|_{\hat{\Sigma}_{N_h}} \right)$.

Proof.

$$
\begin{aligned}
& V^{\pi^*,\pi^*} - V^{\hat{\pi},\pi^{N_l}} \\
& = \langle \phi^{\pi^*,P}(\pi^*) - \phi^{\pi^{N_l},P}(\hat{\pi}), \theta^* \rangle \\
& = \langle \phi^{\pi^*,P}(\pi^*) - \phi^{\pi^{N_l},P}(\pi^*), \theta^* \rangle + \langle \phi^{\pi^{N_l},P}(\pi^*) - \phi^{\pi^{N_l},P}(\hat{\pi}), \theta^* \rangle \\
& \qquad \text{(first term = sub-MDP sub-optimality; second term = high-level policy sub-optimality)} \\
& \leq H_h(\frac{C_1}{\sqrt{N_l}} + C_2\epsilon') + \langle \phi^{\pi^{N_l},P}(\pi^*) - \phi^{\pi^{N_l},P}(\hat{\pi}), \theta^* \rangle \\
& \leq H_h(\frac{C_1}{\sqrt{N_l}} + C_2\epsilon') + \langle \phi^{\pi^{N_l},P^{\epsilon'}}(\pi^*) - \phi^{\pi^{N_l},P^{\epsilon'}}(\hat{\pi}), \theta^* \rangle \\
& + |\langle \phi^{\pi^{N_l},P}(\pi^*) - \phi^{\pi^{N_l},P^{\epsilon'}}(\pi^*), \theta^* \rangle| + |\langle \phi^{\pi^{N_l},P^{\epsilon'}}(\hat{\pi}) - \phi^{\pi^{N_l},P}(\hat{\pi}), \theta^* \rangle| \\
& \leq H_h(\frac{C_1}{\sqrt{N_l}} + C_2\epsilon') + 2C(\epsilon',\delta) + \langle \phi^{\pi^{N_l},P^{\epsilon'}}(\pi^*) - \phi^{\pi^{N_l},P^{\epsilon'}}(\hat{\pi}), \theta^* - \hat{\theta} \rangle + \langle \phi^{\pi^{N_l},P^{\epsilon'}}(\pi^*) - \phi^{\pi^{N_l},P^{\epsilon'}}(\hat{\pi}), \hat{\theta} \rangle \\
& \qquad \text{(expand out the second term)} \\
& \leq H_h(\frac{C_1}{\sqrt{N_l}} + C_2\epsilon') + 2C(\epsilon',\delta) + \langle \phi^{\pi^{N_l},P^{\epsilon'}}(\pi^*) - \phi^{\pi^{N_l},P^{\epsilon'}}(\hat{\pi}), \theta^* - \hat{\theta} \rangle \\
& \qquad \text{(definition of } \hat{\pi}\text{: } \langle \phi^{\pi^{N_l},P^{\epsilon'}}(\pi^*) - \phi^{\pi^{N_l},P^{\epsilon'}}(\hat{\pi}), \hat{\theta} \rangle \leq 0) \\
& \leq H_h(\frac{C_1}{\sqrt{N_l}} + C_2\epsilon') + 2C(\epsilon',\delta) + \frac{1}{\sqrt{N_h}}(2d\log(1+\frac{N_h}{d}))\|\theta^* - \hat{\theta}\|_{\hat{\Sigma}_{N_h}} \\
& \qquad \text{(use of Elliptical lemma)}
\end{aligned}
$$

□

Data Tradeoff: Using the above bound, we can derive the following rates:

- Under idealized-feedback and requiring both high- and low-level feedback, the overall rate comes out to $O(N_l^{-1/4} + N_h^{-1/2})$.

  This is because $\hat{\Sigma}_{N_h} = O\left(\frac{N_h^{1/2}}{N_l^{1/4}} + 1\right)$. Thus, the dominating factor is the bias of the reward learning.
- Under current-feedback and requiring both high- and low-level feedback, the overall rate comes out to $O(N_l^{-1/2} + N_h^{-1/2})$.

  This is because $\|\theta^* - \hat{\theta}\|_{\hat{\Sigma}_{N_h}} = O(1)$.
- Under only low-level feedback (due to sufficiency in coverage), the overall rate comes out to $O(N_l^{-1/2})$.

  We have that:

$$\begin{aligned}
&\langle \phi^{\pi^{N_l},P^{\epsilon'}}(\pi^*) - \phi^{\pi^{N_l},P^{\epsilon'}}(\hat{\pi}), \theta^* - \hat{\theta}\rangle \\
&\leq \|\phi^{\pi^{N_l},P^{\epsilon'}}(\pi^*) - \phi^{\pi^{N_l},P^{\epsilon'}}(\hat{\pi})\|_{\hat{\Sigma}_{N_l}^{-1}} \|\theta^* - \hat{\theta}\|_{\hat{\Sigma}_{N_l}} && (\hat{\Sigma}_{N_h}^{-1} \preceq \hat{\Sigma}_i^{-1}) \\
&\leq \frac{1}{N_h}\sum_{i=1}^{N_h} \|\phi^{P^{\epsilon'}}(\pi_1^i) - \phi^{P^{\epsilon'}}(\pi_2^i)\|_{\hat{\Sigma}_i^{-1}} \|\theta^* - \hat{\theta}\|_{\hat{\Sigma}_{N_l}} && (\diamond) \\
&\leq \frac{1}{\sqrt{N_l}}(2d\log(1 + \frac{N_l}{d}))\|\theta^* - \hat{\theta}\|_{\hat{\Sigma}_{N_l}}
\end{aligned}$$

$(\diamond)$: since low-level policy feature expectation is a superset of high-level policy expectation, it follows that by choice of low-level policies π_1^i, π_2^i: $\|\phi^{P^{\epsilon'}}(\pi_1^i) - \phi^{P^{\epsilon'}}(\pi_2^i)\|_{\hat{\Sigma}_i^{-1}} \geq \|\phi^{\pi^{N_l},P^{\epsilon'}}(\pi^*) - \phi^{\pi^{N_l},P^{\epsilon'}}(\hat{\pi})\|_{\hat{\Sigma}_{N_l}^{-1}}$
Moreover, since low-level feedback is always unbiased, $\|\theta^* - \hat{\theta}\|_{\hat{\Sigma}_{N_l}} = O(1)$. Thus, the overall rate comes out to $O(N_l^{-1/2})$.

Remark 29 (High Probability Guarantee)**.** *For completeness, we show that the theorem statement holds with probability at least* $1 - \delta$*:*

$$\begin{aligned}
&\Pr(\bigcap_{s,a} \mathcal{E}^{s,a} \cap \mathcal{E}_1^l \cap \mathcal{E}_2^l \cap \mathcal{E}_1^h \cap \mathcal{E}_2^h) \\
&\geq 1 - \Pr(\neg \bigcap_{s,a} \mathcal{E}^{s,a}) - \Pr(\neg\mathcal{E}_1^l) - \Pr(\neg\mathcal{E}_2^l) - \Pr(\neg\mathcal{E}_1^h) - \Pr(\neg\mathcal{E}_2^h) \\
&\geq 1 - \delta/5 - \delta/5 - \delta/5 - \delta/5 - \delta/5 \\
&= 1 - \delta
\end{aligned}$$

11.9.2.7 Additional Guarantees

In addition, we derive requisite conditions on the constants for idealized-feedback (the most interesting case).

Necessary Auxiliary Parameters Bound: We have that,

$$
\begin{aligned}
&H_h(\frac{C_1}{\sqrt{N_l}} + C_2\epsilon') + 2C(\epsilon', \delta) + \frac{1}{\sqrt{N_h}}(2d\log(1 + \frac{N_h}{d}))\|\theta^* - \hat{\theta}\|_{\hat{\Sigma}_{N_h}} \\
&\leq H_h(\frac{C_1}{\sqrt{N_l}} + C_2\epsilon') + 2C(\epsilon', \delta) + N_h^{-1/2} 2d\left(2\|\theta^* - \theta^{N_h}\|_{\tilde{\Sigma}_{N_h}} + O(B\sqrt{d\log N_h/\delta}W) + \sqrt{8N_h}C(\epsilon', \delta)\right) \\
&\leq H_h(\frac{C_1}{\sqrt{N_l}} + C_2\epsilon') + (8d+2)C(\epsilon', \delta) + N_h^{-1/2} 2d\left(\left(\frac{N_h^{1/2}}{N_l^{1/4}} + (N_h\epsilon')^{1/2}\right) + C' + O(B\sqrt{d\log N_h/\delta}W)\right) \\
&\leq (H_hC_1)N_l^{-1/2} + 2dN_l^{-1/4} + C_2H_h\epsilon' + d\epsilon'^{1/2} + 9dC(\epsilon', \delta) + 2dC''N_h^{-1/2} \\
&= (H_hC_1)N_l^{-1/2} + 2dN_l^{-1/4} + C_2H_h\epsilon' + d\epsilon'^{1/2} + 9d\left(2H_hBRd^2\epsilon' + 8H_h^3H_l\delta\right) + 2dC''N_h^{-1/2} \\
&\leq (2d + H_hC_1)N_l^{-1/4} + (C_2H_h + 18d^3H_hBR)\epsilon' + 72dH_h^3H_l\delta + 2dC''N_h^{-1/2}
\end{aligned}
$$

Setting the upper bound to be below ϵ, or each term to be below $\epsilon/4$, we obtain the following bounds:

- $N_l \geq O(\frac{(d+H_hC_1)^4}{\epsilon^4})$.
- $N_h \geq O(\frac{d^2}{\epsilon^2})$.
- $\kappa \geq O(\frac{dH_h^2H_l}{\epsilon})$:
  $72dH_h^3H_l\delta \leq \epsilon/4 \Rightarrow \delta \leq O(\frac{\epsilon}{dH_h^3H_l})$.
  Recall $\delta = \frac{1}{\kappa H_h} + \frac{C_1}{\kappa H_h H_l \sqrt{N_l}} + \frac{C_2\epsilon'}{\kappa H_h H_l} + H_l\epsilon'$.
  This implies that $\kappa \geq O(\frac{dH_h^2H_l}{\epsilon})$ and $\epsilon \leq O(\frac{\epsilon}{dH_h^3H_l^2})$.
- $\epsilon' \leq O(\min(\frac{\epsilon}{dH_h^3H_l^2}, \frac{\epsilon}{d^3H_hBR})$:
  Finally, we also require that $(C_2H_h + 18d^3H_hBR)\epsilon' \leq \epsilon/4 \Rightarrow \epsilon' \leq O(\frac{\epsilon}{d^3H_hBR})$. Thus, we need that $\epsilon' \leq O(\min(\frac{\epsilon}{dH_h^3H_l^2}, \frac{\epsilon}{d^3H_hBR})$.

11.10 Statistical Efficiency of HRL

An useful sanity check for hierarchical RL algorithms is that it achieves improved statisical sample complexity in settings with repeated sub-MDP structure [291]. As in [291], we examine if Algorithm 26 also improves upon algorithms that do not leverage hierarchical structure. We make this comparison with vanilla UCB-VI under the same isomophism assumption.

Corollary 14. *Setting* $\mathcal{A}_{s,a}$ *to be the standard UCB-VI algorithm with* $UB(\mathcal{R}^{N^{K,H_h(s,a)}}) = O(H_l^{3/2}\sqrt{|S_{s,a}^l||A|N^{H_h,K}(s,a)})$*, we have the following bound:*

$$
\begin{aligned}
&\sum_{s,a\in\mathcal{C}(S,A)} UB(\mathcal{R}^{N^{K,H_h(s,a)}}) + H^hH^l\sqrt{N^{K,H_h}(s,a)} \\
&\qquad \leq \tilde{O}(H_l^{3/2}\sqrt{\max_{s,a}|S_{s,a}^l||A|}\sqrt{|\mathcal{C}(S,A^h)|(H_hK)} + H_hH_l\sqrt{|\mathcal{C}(S,A^h)|H_hK})
\end{aligned}
$$

Comparison with vanilla UCB-VI: Standard application of UCB-VI yields the following rate: $\tilde{O}((H_h H_l)^{3/2}\sqrt{|S||A|K})$. Hier-UCB-VI compares favorably to vanilla UCB-VI, if $\max_{s,a}|S^l_{s,a}||\mathcal{C}(S, A^h)| << |S|$. Or in words, there are a lot of repeated/identical sub-MDPs and sub-MDPs have small state space size.

Chapter 12

Discussion and Future Directions

This thesis is devoted towards better understanding the multi-agents in our present world and in our future world. In the first half of this thesis, we study how to design ML models to account for other agents, who are affected by the model's output. In the second half of this thesis, we study how machine learning can facilitate the design of multi-agent systems in both the decentralized and centralized setting. Most of these problems are motivated by contemporary problems in our *present* multi-agent world involving incentives and/or capabilities, which transpired during the course of thesis. Looking to the future, multi-agent based problems abound. This section touches on some prominent problems on the horizon, which I believe to be important in our *future* multi-agentic world.

First, while LLMs have demonstrated tremendous capabilities, much of the present focus has been on enhancing their capabilities in a vacuum. But with the rise of LLMs and the growing promise of agents, we will soon see models interfacing even more closely and frequently with humans and other models (for example representing other businesses). This raises a number of salient problems:

1. **Alignment:** How can we ensure that the model is aligned and not unaligned by bad human actors and/or models [16]? How can we develop comprehensive safeguards?
2. **Incentives:** A less studied topic in current LLM literature is how LLM agents interface with each other. For example, can they soundly do business on the part of the human users just as we humans can do business?

Second, there is an enormous number of problems that we have yet to understand regarding agents. Some key problems are:

1. **Training and Inference:** How can we build better harnesses for orchestrating multiple agents [249]? Multi-agents are now known to be better than single agents in various settings and are verily used in frontier labs [15]. How can we train multiple agents to form a more capable, performant and aligned multi-agent system? How can we develop harnesses for unleashing multi-agent benefits during inference? These are key questions in order to realize AI organizations, as targeted by AGI roadmaps of frontier labs [10].

   Furthermore, with ever growing compute, it seems that data will be the key bottleneck in future ML scaling. A (once) popular belief is that forms of self-play where another agent plays with the agent being trained can generate sufficient synthetic data to enable such scaling [36]. How can we realize this immense possibility?

2. **Decentralized MAS:** Soon, we will have agents carrying out tasks on our behalf on the internet. This means any and every consideration in Responsible AI will need to also have a "multi-agent" counterpart [16].

   For example, how can we ensure multi-agent alignment over the course of multi-turns when the agents have not been trained together? How can we facilitate easy coordination? And, when decentralized agents do collaborate, how will multi-turn attribution work?

In sum, I believe there is a fast growing set of multi-agent problems that we have yet to address! Our present world is inherently multi-agentic, and will soon be populated by more AI agents that interact with us and each other. This makes me ever more bullish about the multi-agent research agenda in our *future* world, which is growing more multi-agentic over time. Thank you very much for reading and I hope you the reader will join me in pursuing this rich and important research agenda!

Chapter 13

Bibliography

[1] AP English Literature Scoring Rubrics. https://apcentral.collegeboard.org/media/pdf/ap-english-literature-and-composition-frqs-1-2-3-scoring-rubrics.pdf. Accessed: 2024-05-15. 11.1.1

[2] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *21st*, pages 1–8, 2004. 10.1, 10.2, 10.3, 10.3, 10.7.1, 64

[3] Jayadev Acharya, Sourbh Bhadane, Arnab Bhattacharyya, Saravanan Kandasamy, and Ziteng Sun. Sample complexity of distinguishing cause from effect. In *International Conference on Artificial Intelligence and Statistics*, pages 10487–10504. PMLR, 2023. 5.7

[4] A. Agarwal, M. A. Dahleh, and T. Sarkar. A marketplace for data: An algorithmic solution. In *20th*, pages 701–726, 2019. 8.1

[5] Philippe Aghion and Richard Holden. Incomplete contracts and the theory of the firm: What have we learned over the past 25 years? *Journal of Economic Perspectives*, 25(2): 181–197, 2011. 9.2.2

[6] Raj Agrawal, Chandler Squires, Karren Yang, Karthikeyan Shanmugam, and Caroline Uhler. Abcd-strategy: Budgeted experimental design for targeted causal structure discovery. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3400–3409. PMLR, 2019. 5.7

[7] Saba Ahmadi, Hedyeh Beyhaghi, Avrim Blum, and Keziah Naggita. The strategic perceptron. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 6–25, 2021. 4.1

[8] Ulrich Aïvodji, Hiromi Arai, Olivier Fortineau, Sébastien Gambs, Satoshi Hara, and Alain Tapp. Fairwashing: the risk of rationalization. In *International Conference on Machine Learning*, pages 161–170. PMLR, 2019. 3.10

[9] Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Joseph Naor. The online set cover problem. *SIAM Journal on Computing*, 39(2):361–370, 2009. 6.3.2

[10] Sam Altman. The intelligence age, sep 2024. URL https://ia.samaltman.com/. Outlines OpenAI's vision for the evolution of AI into personal teams and organizations.

Accessed: 2025-12-06. 1

[11] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016. 4.4, 11.1, 11.6

[12] Christopher Anders, Plamen Pasliev, Ann-Kathrin Dombrowski, Klaus-Robert Müller, and Pan Kessel. Fairwashing explanations with off-manifold detergent. In *International Conference on Machine Learning*, pages 314–323. PMLR, 2020. 3.10

[13] Dana Angluin. Queries and concept learning. *Machine learning*, 2:319–342, 1988. 2.1.1, 2.6, 6.1.1, 6.2

[14] Anthropic. Developing a computer use model, 2024. URL `https://www.anthropic.com/news/developing-computer-use`. 9.2.2, 9.7

[15] Anthropic. How we built our multi-agent research system, jun 2025. URL `https://www.anthropic.com/engineering/multi-agent-research-system`. Accessed: 2025-12-06. 1

[16] Anthropic Alignment Science Team. Recommendations for technical AI safety research directions, 2025. URL `https://alignment.anthropic.com/2025/recommended-directions/`. Accessed: 2025-12-06. 1, 2

[17] Patrick Assouad. Densité et dimension. In *Annales de l'Institut Fourier*, volume 33, pages 233–282, 1983. 3.5

[18] Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International conference on machine learning*, pages 263–272. PMLR, 2017. 9.10.1, 9.10.2, 11.3.1, 26

[19] M. Babeş-Vroman, V. Marivate, K. Subramanian, and M. L. Littman. Apprenticeship learning about multiple intentions. In *28th*, pages 897–904, 2011. 10.1

[20] Y. Bachrach and J. S. Rosenschein. Coalitional skills games. In *7th*, pages 1023–1030, 2008. 8.1.2

[21] Yoram Bachrach, Evangelos Markakis, Ariel D Procaccia, Jeffrey S Rosenschein, and Amin Saberi. Approximating power indices. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*, pages 943–950, 2008. 7.9

[22] Yoram Bachrach, Evangelos Markakis, Ezra Resnick, Ariel D Procaccia, Jeffrey S Rosenschein, and Amin Saberi. Approximating power indices: theoretical and empirical analysis. *Autonomous Agents and Multi-Agent Systems*, 20(2):105–122, 2010. 8.4

[23] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. *Journal of the ACM (JACM)*, 65(3):1–55, 2018. 5.6, 5.11, 5.11.2

[24] R Iris Bahar, Erica A Frohm, Charles M Gaona, Gary D Hachtel, Enrico Macii, Abelardo Pardo, and Fabio Somenzi. Algebric decision diagrams and their applications. *Formal methods in system design*, 10(2-3):171–206, 1997. 7.2

[25] Yu Bai and Chi Jin. Provable self-play algorithms for competitive reinforcement learning.

In *International conference on machine learning*, pages 551–560. PMLR, 2020. 9.5

[26] Yu Bai, Chi Jin, Huan Wang, and Caiming Xiong. Sample-efficient learning of stackelberg equilibria in general-sum games. *Advances in Neural Information Processing Systems*, 34: 25799–25811, 2021. 9.3, 9.11

[27] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022. 11.2

[28] M.-F. Balcan, A. D. Procaccia, and Y. Zick. Learning cooperative games. In *24th*, pages 475–482, 2015. 8.1.2

[29] Maria-Florina Balcan and Nicholas JA Harvey. Learning submodular functions. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 793–802, 2011. 7.4.2, 7.8.2

[30] Maria-Florina Balcan and Phil Long. Active and passive learning of linear separators under log-concave distributions. In *Conference on Learning Theory*, pages 288–316. PMLR, 2013. 6.1

[31] Maria-Florina Balcan, Eric Blais, Avrim Blum, and Liu Yang. Active property testing. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 21–30. IEEE, 2012. 6.5, 6.8.4

[32] Maria-Florina Balcan, Avrim Blum, Nika Haghtalab, and Ariel D Procaccia. Commitment without regrets: Online learning in stackelberg security games. In *Proceedings of the sixteenth ACM conference on economics and computation*, pages 61–78, 2015. 9.12

[33] Maria Florina Balcan, Ariel D Procaccia, and Yair Zick. Learning cooperative games. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015. 7.2

[34] E. Balkanski, U. Syed, and S. Vassilvitskii. Statistical cost sharing. In *31st*, pages 6221–6230, 2017. 8.1.2, 8.3.1, 8.4, 8.7, 8.8

[35] Eric Balkanski, Umar Syed, and Sergei Vassilvitskii. Statistical cost sharing. In *Advances in Neural Information Processing Systems*, pages 6221–6230, 2017. 7.2, 7.8.2, 7.8.7

[36] Trapit Bansal, Jakub Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. Emergent complexity via multi-agent competition. In *International Conference on Learning Representations*, 2018. URL `https://arxiv.org/abs/1710.03748`. 1

[37] Sylvain Béal, Mihai Manea, Eric Rémila, and Philippe Solal. Games with identical shapley values. *Handbook of the Shapley Value*, pages 93–110, 2019. 7.8.3

[38] Yahav Bechavod, Katrina Ligett, Zhiwei Steven Wu, and Juba Ziani. Causal feature discovery through strategic modification. *arXiv preprint arXiv:2002.07024*, 3, 2020. 4.1

[39] Gary S Becker. Crime and punishment: An economic approach. In *The economic dimensions of crime*, pages 13–68. Springer, 1968. 3.3

[40] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *COLT*, volume 3, page 1, 2009. 6.3.2

[41] Omer Ben-Porat, Yishay Mansour, Michal Moshkovitz, and Boaz Taitler. Principal-agent reward shaping in mdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9502–9510, 2024. 9.1, 9.3, 9.12

[42] Thor Benson. Your boss's spyware could train ai to replace you. *Wired*, 2023. URL `https://www.wired.com/story/corporate-surveillance-train-ai/`. 2.1

[43] Dimitris Bertsimas and Santosh Vempala. Solving convex programs by random walks. *Journal of the ACM (JACM)*, 51(4):540–556, 2004. 6.3.2

[44] U. Bhatt, A. Xiang, S. Sharma, A. Weller, A. Taly, Y. Jia, J. Ghosh, R. Puri, J. M. F. Moura, and P. Eckersley. Explainable machine learning in deployment. arXiv:1909.06342, 2019. 8.4

[45] Eric Blais, Renato Ferreira Pinto Jr, and Nathaniel Harms. Vc dimension and distribution-free sample-based testing. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 504–517, 2021. 6.5

[46] Guy Blanc, Neha Gupta, Jane Lange, and Li-Yang Tan. Estimating decision tree learnability with polylogarithmic sample complexity. *Advances in Neural Information Processing Systems*, 33, 2020. 6.5

[47] Avrim Blum and Lunjia Hu. Active tolerant testing. In *Conference On Learning Theory*, pages 474–497. PMLR, 2018. 6.5

[48] Matteo Bollini, Francesco Bacchiocchi, Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Contracting with a reinforcement learning agent by playing trick or treat. *arXiv preprint arXiv:2410.13520*, 2024. 9.1, 9.3, 9.12

[49] Boston Consulting Group. Rethinking B2B software pricing in the agentic AI era. BCG Publications, 2025. URL `https://www.bcg.com/publications/2025/rethinking-b2b-software-pricing-in-the-era-of-ai`. 9.2.2

[50] Samuel R Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamilė Lukošiūtė, Amanda Askell, Andy Jones, Anna Chen, et al. Measuring progress on scalable oversight for large language models. *arXiv preprint arXiv:2211.03540*, 2022. 11.1, 11.6

[51] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. 11.4.1

[52] Gavin Brown, Shlomi Hod, and Iden Kalemaj. Performative prediction in a stateful world. In *International Conference on Artificial Intelligence and Statistics*, pages 6045–6061. PMLR, 2022. 4.1

[53] James N Brown and Robert W Rosenthal. Testing the minimax hypothesis: A re-examination of o'neill's game experiment. *Econometrica: Journal of the Econometric Society*, pages 1065–1081, 1990. 2.11

[54] Noam Brown and Tuomas Sandholm. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018. 7.7

[55] Michael Brückner and Tobias Scheffer. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge*

discovery and data mining, pages 547–555, 2011. 4.1

[56] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023. 4.1.1

[57] Rui M Castro and Robert D Nowak. Minimax bounds for active learning. *IEEE Transactions on Information Theory*, 54(5):2339–2353, 2008. 2.4.2, 2.11.2, 2.14.1

[58] G. Chalkiadakis, E. Elkind, and M. Wooldridge. *Computational Aspects of Cooperative Game Theory*. Morgan & Claypool, 2011. 8.1.2

[59] Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. Computational aspects of cooperative game theory. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(6):1–168, 2011. 7.9

[60] Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. Goal-conditioned reinforcement learning with imagined subgoals. In *International Conference on Machine Learning*, pages 1430–1440. PMLR, 2021. 11.6

[61] J. Chen, L. Song, M. J. Wainwright, and M I. Jordan. L-Shapley and C-Shapley: Efficient model interpretation for structured data. In *7th*, 2019. 8.1, 8.4

[62] Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. L-shapley and c-shapley: Efficient model interpretation for structured data. *arXiv preprint arXiv:1808.02610*, 2018. 7.5

[63] Siyu Chen, Donglin Yang, Jiayang Li, Senmiao Wang, Zhuoran Yang, and Zhaoran Wang. Adaptive model design for markov decision process. In *International Conference on Machine Learning*, pages 3679–3700. PMLR, 2022. 9.3

[64] Siyu Chen, Mengdi Wang, and Zhuoran Yang. Actions speak what you want: Provably sample-efficient reinforcement learning of the quantal stackelberg equilibrium from strategic feedbacks. *arXiv preprint arXiv:2307.14085*, 2023. 9.12

[65] Yiling Chen, Chara Podimata, Ariel D Procaccia, and Nisarg Shah. Strategyproof linear regression in high dimensions. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 9–26, 2018. 2.14.1, 3.2

[66] Yiling Chen, Yang Liu, and Chara Podimata. Learning strategy-aware linear classifiers. *Advances in Neural Information Processing Systems*, 33:15265–15276, 2020. 4.1

[67] Yuansi Chen, Raaz Dwivedi, Martin J Wainwright, and Bin Yu. Fast mcmc sampling algorithms on polytopes. *The Journal of Machine Learning Research*, 19(1):2146–2231, 2018. 3.6.1

[68] J. Choi and K.-E. Kim. Nonparametric Bayesian inverse reinforcement learning for multiple reward functions. In *25th*, pages 314–322, 2012. 10.1

[69] N. Chomsky. *Aspects of the Theory of Syntax*. MIT Press, 1965. 10.1

[70] Davin Choo, Kirankumar Shiragur, and Arnab Bhattacharyya. Verification and search algorithms for causal dags. *Advances in Neural Information Processing Systems*, 35:

12787–12799, 2022. 5.1, 5.2, 5.7, 5.11

[71] Paul Christiano, Buck Shlegeris, and Dario Amodei. Supervising strong learners by amplifying weak experts. *arXiv preprint arXiv:1810.08575*, 2018. 11.1, 11.6

[72] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017. 11.2

[73] S. Cohen, G. Dror, and E. Ruppin. Feature selection via coalitional game theory. *Neural Computation*, 19(7):1939–1961, 2007. 8.1, 8.4

[74] Shay Cohen, Gideon Dror, and Eytan Ruppin. Feature selection via coalitional game theory. *Neural Computation*, 19(7):1939–1961, 2007. 7.5

[75] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine learning*, 15(2):201–221, 1994. 6.1.1, 6.3.1.1

[76] V. Conitzer and T. Sandholm. Complexity of constructing solutions in the core based on synergies among coalitions. *Artificial Intelligence*, 170(6–7):607–619, 2006. 8.1.2

[77] Vincent Conitzer and Tuomas Sandholm. Computing shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. 2004. 7.9

[78] Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*, pages 82–90, 2006. 9.3, 9.4, 9.4.1, 9.8

[79] T.M. Cover and E. Ordentlich. Universal portfolios with side information. *IEEE Transactions on Information Theory*, 42(2):348–363, 1996. 5

[80] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*, 2020. 3.6.2, 3.10

[81] Sanjoy Dasgupta. Analysis of a greedy active learning strategy. *Advances in neural information processing systems*, 17, 2004. 2.6, 2.7.1, 2.14.1

[82] Sanjoy Dasgupta. Coarse sample complexity bounds for active learning. In *NIPS*, volume 18, pages 235–242, 2005. 6.1, 6.3.1.1

[83] Sanjoy Dasgupta, Adam Tauman Kalai, and Claire Monteleoni. Analysis of perceptron-based active learning. In *International conference on computational learning theory*, pages 249–263. Springer, 2005. 2.6, 6.2

[84] Sanjoy Dasgupta, Daniel J Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. *Advances in neural information processing systems*, 20, 2007. 2.3.1, 6.1, 6.3.2

[85] Sanjoy Dasgupta, Daniel Hsu, Stefanos Poulis, and Xiaojin Zhu. Teaching a black-box learner. In *International Conference on Machine Learning*, pages 1547–1555. PMLR, 2019. 6.2, 6.3.2, 13, 6.3.2, 6.8.5.2, 2, 6.8.5.2, 6.8.5.2, 6.8.5.2

[86] A. Datta, A. Datta, A. D. Procaccia, and Y. Zick. Influence in classification via cooperative game theory. In *24th*, pages 511–517, 2015. 8.1

[87] A. Datta, S. Sen, and Y. Zick. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *37th*, pages 598–617, 2016. 8.1, 8.4

[88] Amit Datta, Anupam Datta, Ariel D Procaccia, and Yair Zick. Influence in classification via cooperative game theory. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015. 7.5

[89] Ofer Dekel, Felix Fischer, and Ariel D Procaccia. Incentive compatible regression learning. *Journal of Computer and System Sciences*, 76(8):759–777, 2010. 2.14.1

[90] Stephan Dempe and Alain B Zemkoho. On the karush–kuhn–tucker reformulation of the bilevel optimization problem. *Nonlinear Analysis: Theory, Methods & Applications*, 75(3): 1202–1218, 2012. 9.12

[91] X. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 19(2):257–266, 1994. 8.1.2

[92] Xiaotie Deng and Christos H Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 19(2):257–266, 1994. 7.1, 7.2, 7.8.7

[93] Lee H Dicker. Variance estimation in high-dimensional linear models. *Biometrika*, 101(2): 269–284, 2014. 6.2

[94] Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 55–70, 2018. 3.2, 4.1

[95] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`. 8.5.1

[96] Paul Dutting, Tim Roughgarden, and Inbal Talgam-Cohen. The complexity of contracts. *SIAM Journal on Computing*, 50(1):211–254, 2021. 9.2.2

[97] Frederick Eberhardt. Causation and intervention. *Unpublished doctoral dissertation, Carnegie Mellon University*, page 93, 2007. 5.1, 5.2

[98] Lilian Edwards and Michael Veale. Slave to the algorithm: Why a right to an explanation is probably not the remedy you are looking for. *Duke L. & Tech. Rev.*, 16:18, 2017. 3.1

[99] E. Elkind and D. B. Pasechnik. Computing the nucleolus of weighted voting games. In *20th*, pages 327–335, 2009. 8.1.2

[100] E. Elkind, L. A. Goldberg, P. W. Goldberg, and M. J. Wooldridge. On the computational complexity of weighted voting games. *Annals of Mathematics and Artificial Intelligence*, 56:109–131, 2009. 8.1.2

[101] T. Everitt, V. Krakovna, L. Orseau, and S. Legg. Reinforcement learning with a corrupted reward channel. In *26th*, pages 4705–4713, 2017. 10.1

[102] Tom Everitt, Ryan Carey, Eric D Langlois, Pedro A Ortega, and Shane Legg. Agent incentives: A causal perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11487–11495, 2021. 4.4

[103] Tom Everitt, Esra Kürüm, and Marcus Hutter. Reward tampering. In *Proceedings of the 2021 International Conference on Autonomous Agents and Multiagent Systems*, pages

413–421, 2021. 4.4

[104] B. Fain, K. Munagala, and N. Shah. Fair allocation of indivisible public goods. In *19th*, pages 575–592, 2018. 10.17

[105] Sebastian Farquhar, Ryan Carey, and Tom Everitt. Path-specific objectives for safer agent incentives. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9529–9538, 2022. 4.4

[106] Shaheen S Fatima, Michael Wooldridge, and Nicholas R Jennings. A linear approximation method for the shapley value. *Artificial Intelligence*, 172(14):1673–1699, 2008. 7.9

[107] Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998. 6.8.4, 5

[108] Uriel Feige, Michal Feldman, Nicole Immorlica, Rani Izsak, Brendan Lucier, and Vasilis Syrgkanis. A unifying hierarchy of valuations with complements and substitutes. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. 7.2

[109] Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine learning*, 28:133–168, 1997. 2.6

[110] Drew Fudenberg and Luis Rayo. Training and effort dynamics in apprenticeship. *American Economic Review*, 109(11):3780–3812, 2019. 2.14.1

[111] Luis Garicano and Luis Rayo. Relational knowledge transfers. *American Economic Review*, 107(9):2695–2730, 2017. 2.1.1, 2.14.1

[112] Matthias Gerstgrasser and David C Parkes. Oracles & followers: Stackelberg equilibria in deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 11213–11236. PMLR, 2023. 9.12

[113] Ganesh Ghalme, Vineet Nair, Itay Eilat, Inbal Talgam-Cohen, and Nir Rosenfeld. Strategic classification in the dark. In *International Conference on Machine Learning*, pages 3672–3681. PMLR, 2021. 4.1

[114] AmirEmad Ghassami, Saber Salehkaleybar, Negar Kiyavash, and Elias Bareinboim. Budgeted experiment design for causal structure learning. In *International Conference on Machine Learning*, pages 1724–1733. PMLR, 2018. 5.1, 5.2

[115] A. Ghorbani and J. Zou. Data Shapley: Equitable valuation of data for machine learning. In *36th*, pages 2242–2251, 2019. 8.1, 8.4, 8.5.2, 8.5.2, 8.5.2

[116] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pages 2242–2251. PMLR, 2019. 7.5

[117] D. B. Gillies. 3. solutions to general non-zero-sum games. 1959. 7.8.7

[118] Andrew Gilpin and Tuomas Sandholm. Lossless abstraction of imperfect information games. *Journal of the ACM (JACM)*, 54(5):25–es, 2007. 7.1

[119] R. Givan, S. Leach, and T. Dean. Bounded-parameter Markov decision processes. *Artificial Intelligence*, 122(1–2):71–109, 2000. 10.1

[120] Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based

on graphical models. *Frontiers in genetics*, 10:524, 2019. 5.7

[121] Sally A Goldman and Michael J Kearns. On the complexity of teaching. *Journal of Computer and System Sciences*, 50(1):20–31, 1995. 6.2, 6.3.2

[122] Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998. 6.5

[123] Shafi Goldwasser, Guy N Rothblum, Jonathan Shafer, and Amir Yehudayoff. Interactive proofs for verifying machine learning. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021. 6.2

[124] Daniel Golovin and Andreas Krause. Adaptive submodularity: A new approach to active learning and stochastic optimization. In *COLT*, pages 333–345, 2010. 2.14.1

[125] Kristjan Greenewald, Dmitriy Katz, Karthikeyan Shanmugam, Sara Magliacane, Murat Kocaoglu, Enric Boix Adsera, and Guy Bresler. Sample efficient active learning of causal trees. *Advances in Neural Information Processing Systems*, 32, 2019. 5.1, 5.2, 5.7

[126] Abhishek Gupta, Aldo Pacchiano, Yuexiang Zhai, Sham Kakade, and Sergey Levine. Unpacking reward shaping: Understanding the benefits of reward engineering on sample complexity. *Advances in Neural Information Processing Systems*, 35:15281–15295, 2022. 9.3

[127] Rishi Gupta and Tim Roughgarden. A pac approach to application-specific algorithm selection. *SIAM Journal on Computing*, 46(3):992–1017, 2017. 7.5

[128] Guru Guruganesh, Yoav Kolumbus, Jon Schneider, Inbal Talgam-Cohen, Emmanouil-Vasileios Vlatakis-Gkaragkounis, Joshua Wang, and S Weinberg. Contracting with a learning agent. *Advances in Neural Information Processing Systems*, 37:77366–77408, 2024. 9.12

[129] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. D. Dragan. Cooperative inverse reinforcement learning. In *30th*, pages 3909–3917, 2016. 10.1

[130] Danijar Hafner, Kuang-Huei Lee, Ian Fischer, and Pieter Abbeel. Deep hierarchical planning from pixels. *Advances in Neural Information Processing Systems*, 35:26091–26104, 2022. 11.1, 11.6

[131] Steve Hanneke. The cost complexity of interactive learning. *Unpublished manuscript*, 2006. 2.3, 2.14.1, 6.2, 6.3.1

[132] Steve Hanneke. Teaching dimension and the complexity of active learning. In *International Conference on Computational Learning Theory*, pages 66–81. Springer, 2007. 6.2, 6.3.2, 6.6, 6.8.5.1

[133] Steve Hanneke. *Theoretical foundations of active learning*. Carnegie Mellon University, 2009. 2.6

[134] Steve Hanneke. Rates of convergence in active learning. *The Annals of Statistics*, pages 333–361, 2011. 6.3.2

[135] Steve Hanneke et al. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309, 2014. 2.1.1, 6.1, 6.1.1, 6.3.1.1, 6.8.3

[136] Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, pages 111–122, 2016. 2.14.1, 3.2, 3.3, 4.1

[137] Oliver Hart and John Moore. Incomplete contracts and renegotiation. *Econometrica: Journal of the Econometric Society*, pages 755–785, 1988. 9.2.2

[138] Tibor Hegedűs. Generalized teaching dimensions and the query complexity of learning. In *Proceedings of the eighth annual conference on Computational learning theory*, pages 108–117, 1995. 2.1.1, 2.6, 6.2, 6.3.2, 6.8.5.1

[139] J. L. Hodges Jr and E. L. Lehmann. Some problems in minimax point estimation. *The Annals of Mathematical Statistics*, pages 182–197, 1950. 10.4.2

[140] Russell Hotten. Volkswagen: The scandal explained. *BBC News*, 2015. URL `https://www.bbc.com/news/business-34324772`. 6.1

[141] Patrik Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. *Advances in neural information processing systems*, 2008. 4.6, 4.6, 2, 5.7

[142] Daniel Joseph Hsu. *Algorithms for active learning*. PhD thesis, UC San Diego, 2010. 6.3.1.1, 6.8.3

[143] Huining Hu, Zhentao Li, and Adrian R Vetta. Randomized experimental design for causal graph discovery. *Advances in neural information processing systems*, 27, 2014. 5.1, 5.2

[144] Jiachen Hu, Xiaoyu Chen, Chi Jin, Lihong Li, and Liwei Wang. Near-optimal representation learning for linear bandits and linear rl. In *International Conference on Machine Learning*, pages 4349–4358. PMLR, 2021. 11.3.2

[145] Tzu-Kuo Huang, Alekh Agarwal, Daniel J Hsu, John Langford, and Robert E Schapire. Efficient and parsimonious agnostic active learning. *Advances in Neural Information Processing Systems*, 28, 2015. 6.3.2

[146] Tzu-Kuo Huang, Lihong Li, Ara Vartanian, Saleema Amershi, and Jerry Zhu. Active learning with oracle epiphany. *Advances in neural information processing systems*, 29, 2016. 2.3.2, 2.6, 2.10.4

[147] Evan Hubinger. Ai safety via market making. `https://www.alignmentforum.org/posts/YWwzccGbcHMJMpT45/ai-safety-via-market-making`. Accessed: 2024-05-15. 11.6

[148] Antti Hyttinen, Frederick Eberhardt, and Patrik O Hoyer. Experiment selection for causal discovery. *Journal of Machine Learning Research*, 14:3041–3071, 2013. 5.1, 5.2

[149] Samuel Ieong and Yoav Shoham. Marginal contribution nets: a compact representation scheme for coalitional games. In *Proceedings of the 6th ACM conference on Electronic commerce*, pages 193–202. ACM, 2005. 7.2

[150] Intercom. Pricing AI agents: What does value-based pricing really mean for AI?, May 2025. URL `https://www.intercom.com/blog/pricing-ai-agents/`. 9.2.2

[151] Geoffrey Irving, Paul Christiano, and Dario Amodei. Ai safety via debate. *arXiv preprint*

arXiv:1805.00899, 2018. 11.6

[152] Dima Ivanov, Paul Dütting, Inbal Talgam-Cohen, Tonghan Wang, and David C Parkes. Principal-agent reinforcement learning: Orchestrating ai agents with contracts. *arXiv preprint arXiv:2407.18074*, 2024. 9.1, 9.3, 9.12

[153] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. High accuracy and high fidelity extraction of neural networks. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 1345–1362, 2020. 3.6.2, 3.10

[154] Kevin G Jamieson and Lalit Jain. A bandit approach to sequential experimental design with false discovery control. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, 2018. 5.7

[155] Jeya Vikranth Jeyakumar, Joseph Noor, Yu-Hsi Cheng, Luis Garcia, and Mani Srivastava. How can i explain this to you? an empirical study of deep neural network explanation methods. *Advances in Neural Information Processing Systems*, 33, 2020. 3.1

[156] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. Hynes, N. M. Gürel, B. Li, C. Zhang, D. Song, and C. Spanos. Towards efficient data valuation based on the Shapley value. In *22nd*, pages 1167–1176, 2019. 8.1, 8.4, 8.5.2

[157] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. Hynes, N. M. Gürel, B. Li, C. Zhang, C. Spanos, and D. Song. Efficient task-specific data valuation for nearest neighbor algorithms. *Proceedings of the VLDB Endowment*, 12(11):1610–1623, 2019. 8.1

[158] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gurel, Bo Li, Ce Zhang, Dawn Song, and Costas Spanos. Towards efficient data valuation based on the shapley value. *arXiv preprint arXiv:1902.10275*, 2019. 7.3, 7.6.1

[159] Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pages 4870–4879. PMLR, 2020. 9.6.2

[160] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*, pages 99–127. World Scientific, 2013. 3.3

[161] Hsu Kao, Chen-Yu Wei, and Vijay Subramanian. Decentralized cooperative reinforcement learning with hierarchical information structure. In *International Conference on Algorithmic Learning Theory*, pages 573–605. PMLR, 2022. 9.12

[162] Been Kim, Oluwasanmi Koyejo, Rajiv Khanna, et al. Examples are not enough, learn to criticize! criticism for interpretability. In *NIPS*, pages 2280–2288, 2016. 3.1, 3.6, 3.6.2

[163] Jon Kleinberg and Manish Raghavan. How do classifiers induce agents to invest effort strategically? *ACM Transactions on Economics and Computation (TEAC)*, 8(4):1–23, 2020. 3.2, 4.1

[164] Murat Kocaoglu, Alex Dimakis, and Sriram Vishwanath. Cost-optimal learning of causal graphs. In *International Conference on Machine Learning*, pages 1875–1884. PMLR, 2017. 5.1, 5.2

[165] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer, 2006. 10.11.1

[166] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017. 3.6, 8.5.2, 8.5.2

[167] Weihao Kong and Gregory Valiant. Estimating learnability in the sublinear data regime. *Advances in Neural Information Processing Systems*, 31:5455–5464, 2018. 6.2

[168] A. Kopelowitz. Computation of the kernels of simple games and the nucleolus of N-person games. RM 31, Department of Mathematics, the Hebrew University of Jerusalem, 1967. 8.3

[169] Christian Kroer and Tuomas Sandholm. Extensive-form game abstraction with bounds. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 621–638. ACM, 2014. 7.2

[170] Christian Kroer and Tuomas Sandholm. Imperfect-recall abstractions with bounds in games. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 459–476. ACM, 2016. 7.2

[171] David Krueger, Tegan Maharaj, and Jan Leike. Causal confusion in reinforcement learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5547–5557. PMLR, 13–18 Jul 2020. 4.4

[172] I Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler. Problems with shapley-value-based explanations as feature importance measures. In *37th*, 2020. 8.4, 8.6

[173] Eduardo S Laber and Loana Tito Nogueira. On the hardness of the minimum height decision tree problem. *Discrete Applied Mathematics*, 144(1-2):209–212, 2004. 6.3.1.2, 6.8.4, 6.8.4

[174] Jean-Jacques Laffont and David Martimort. *The theory of incentives: the principal-agent model*. Princeton university press, 2002. 9.2.2

[175] Cassidy Laidlaw, Aditi Singla, and Anca D. Dragan. Correlated proxies: A new definition and improved mitigation for reward hacking. *arXiv preprint arXiv:2403.03185*, 2024. 4.9

[176] Marc Lanctot, Richard Gibson, Neil Burch, Martin Zinkevich, and Michael Bowling. No-regret learning in extensive-form games with imperfect recall. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1035–1042. Omnipress, 2012. 7.2

[177] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 4190–4203, 2017. 7.7

[178] Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. *arXiv preprint arXiv:1612.07182*, 2016. 7.7

[179] Kimin Lee, Laura Smith, and Pieter Abbeel. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. *arXiv preprint arXiv:2106.05091*, 2021. 11.2

[180] Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*, 2018. 11.1, 11.1.2, 11.6

[181] Adam Lerer and Alexander Peysakhovich. Maintaining cooperation in complex social dilemmas using deep reinforcement learning. *arXiv preprint arXiv:1707.01068*, 2017. 7.7

[182] Joshua Letchford and Vincent Conitzer. Computing optimal strategies to commit to in extensive-form games. In *Proceedings of the 11th ACM conference on Electronic commerce*, pages 83–92, 2010. 9.3

[183] Joshua Letchford, Liam MacDermed, Vincent Conitzer, Ronald Parr, and Charles Isbell. Computing optimal strategies to commit to in stochastic games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 1380–1386, 2012. 9.3

[184] Sagi Levanon and Nir Rosenfeld. Generalized strategic classification and the case of aligned incentives. *arXiv preprint arXiv:2202.04357*, 2022. 4.1

[185] Andrew Levy, George Konidaris, Robert Platt, and Kate Saenko. Learning multi-level hierarchies with hindsight. *arXiv preprint arXiv:1712.00948*, 2017. 11.1, 11.6

[186] Yao Li, Minhao Cheng, Kevin Fujii, Fushing Hsieh, and Cho-Jui Hsieh. Learning from group comparisons: exploiting higher order interactions. In *Advances in Neural Information Processing Systems*, pages 4981–4990, 2018. 7.1, 7.2, 7.8.8

[187] David Liben-Nowell, Alexa Sharp, Tom Wexler, and Kevin Woods. Computing shapley value in supermodular coalitional games. In *International Computing and Combinatorics Conference*, pages 568–579. Springer, 2012. 7.9

[188] Erik Lindgren, Murat Kocaoglu, Alexandros G Dimakis, and Sriram Vishwanath. Experimental design for cost-aware learning of causal graphs. *Advances in Neural Information Processing Systems*, 31, 2018. 5.1, 5.2

[189] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988. 6.3.2

[190] Tianqi Liu, Zhaowei Tang, et al. Rrm: Robust reward model training mitigates reward hacking. *arXiv preprint arXiv:2409.13156*, 2024. 4.4

[191] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Neural Information Processing Systems (NIPS)*, 2017. 7.6.1

[192] Zhaozhi Lu, Saptarshi Kumar, Aditya Parnami, Moshe Tennenholtz, and Animesh Kumar. Quantifying and mitigating causal influences in reinforcement learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 14541–14562. PMLR, 17–23 Jul 2022. 4.4

[193] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *30th*, pages 4768–4777, 2017. 8.1, 8.4

[194] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017. 7.3, 7.5

[195] Sasan Maleki, Long Tran-Thanh, Greg Hines, Talal Rahwan, and Alex Rogers. Bounding the estimation error of sampling-based shapley value approximation. *arXiv preprint arXiv:1306.4265*, 2013. 7.9

[196] Charles Marx, Flavio Calmon, and Berk Ustun. Predictive multiplicity in classification. In *International Conference on Machine Learning*, pages 6765–6774. PMLR, 2020. 3.6.2, 3.10

[197] M. Maschler, B. Peleg, and L. S. Shapley. Geometric properties of the kernel, nucleolus, and related solution concepts. *Mathematics of Operations Research*, 4(4):303–338, 1979. 8.2

[198] Dan McCarthy. To regulate ai, try playing in a sandbox. *Emerging Tech Brew*, 2021. URL `https://www.morningbrew.com/emerging-tech/stories/2021/05/26/regulate-ai-just-play-sandbox`. 6.1

[199] Richard D McKelvey and Thomas R Palfrey. Quantal response equilibria for normal form games. *Games and economic behavior*, 10(1):6–38, 1995. 9.5

[200] Christopher Meek. Causal inference and causal explanation with background knowledge. *arXiv preprint arXiv:1302.4972*, 2013. 2

[201] Celestine Mendler-Dünner, Frances Ding, and Yixin Wang. Anticipating performativity by predicting from predictions. In *Advances in Neural Information Processing Systems*. 4.1

[202] Tomasz P Michalak, Karthik V Aadithya, Piotr L Szczepanski, Balaraman Ravindran, and Nicholas R Jennings. Efficient computation of the shapley value for game-theoretic network centrality. *Journal of Artificial Intelligence Research*, 46:607–650, 2013. 7.9

[203] J. Mikhail. *Elements of Moral Cognition: Rawls' Linguistic Analogy and the Cognitive Science of Moral and Legal Judgment*. Cambridge University Press, 2011. 10.1

[204] John Miller, Smitha Milli, and Moritz Hardt. Strategic classification is causal modeling in disguise. In *International Conference on Machine Learning*, pages 6917–6926. PMLR, 2020. 3.10, 4.1

[205] Smitha Milli, Ludwig Schmidt, Anca D Dragan, and Moritz Hardt. Model reconstruction from model explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 1–9, 2019. 3.10

[206] Tom M Mitchell. Version spaces: A candidate elimination approach to rule learning. In *Proceedings of the 5th international joint conference on Artificial intelligence-Volume 1*, pages 305–310, 1977. 3.10

[207] Tom M Mitchell. Generalization as search. *Artificial intelligence*, 18(2):203–226, 1982. 2.2.1, 6.1

[208] Hervé Moulin. *Fair division and collective welfare*. MIT press, 2004. 7.1

[209] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019. 3.1

[210] Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. *Advances in neural information processing systems*, 31, 2018. 11.1, 11.6, 11.7

[211] Roi Naveiro and David Ríos Insua. Gradient methods for solving stackelberg games. In *International conference on algorithmic decision theory*, pages 126–140. Springer, 2019. 9.12

[212] A. Y. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *17th*, pages 663–670, 2000. 10.1

[213] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pages 278–287. Citeseer, 1999. 9.3

[214] Ta Duy Nguyen and Yair Zick. Resource based cooperative games: Optimization, fairness and stability. In *International Symposium on Algorithmic Game Theory*, pages 239–244. Springer, 2018. 7.3

[215] R. Noothigattu, S. S. Gaikwad, E. Awad, S. Dsouza, I. Rahwan, P. Ravikumar, and A. D. Procaccia. A voting-based system for ethical decision making. In *32nd*, pages 1587–1594, 2018. 10.3

[216] Ritesh Noothigattu, Tom Yan, and Ariel D Procaccia. Inverse reinforcement learning from like-minded teachers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9197–9204, 2021. 1.2.2

[217] Ellen Novoseller, Yibing Wei, Yanan Sui, Yisong Yue, and Joel Burdick. Dueling posterior sampling for preference-based reinforcement learning. In *Conference on Uncertainty in Artificial Intelligence*, pages 1029–1038. PMLR, 2020. 11.2

[218] O. Ohrimenko, S. Tople, and S. Tschiatschek. Collaborative machine learning markets with data-replication-robust payments. arXiv:1911.09052, 2019. 8.1

[219] OpenSCHUFA. Openschufa project. 2019. URL `https://openschufa.de/`. 3.1

[220] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 4.1.1, 11.2

[221] Aldo Pacchiano, Aadirupa Saha, and Jonathan Lee. Dueling rl: reinforcement learning with trajectory preferences. *arXiv preprint arXiv:2111.04850*, 2021. 11.4.1

[222] Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. *arXiv preprint arXiv:2201.03544*, 2022. 4.1.1, 4.3.1, 4.9

[223] Jack Parker-Holder, Aldo Pacchiano, Krzysztof M Choromanski, and Stephen J Roberts. Effective diversity in population based reinforcement learning. *Advances in Neural Infor-*

mation Processing Systems, 33:18050–18062, 2020. 11.4.1

[224] Judea Pearl. *Causality*. Cambridge university press, 2009. 4.1, 5.1

[225] B. Peleg and P. Sudhölter. *Introduction to the Theory of Cooperative Games*. Springer, 2nd edition, 2007. 8.1

[226] Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *International Conference on Machine Learning*, pages 7599–7609. PMLR, 2020. 4.1

[227] Javier Perote and Juan Perote-Pena. Strategy-proof estimators for simple regression. *Mathematical Social Sciences*, 47(2):153–176, 2004. 2.14.1

[228] F. Perron and E. Marchand. On the minimax estimator of a bounded normal mean. *Statistics and Probability Letters*, 58:327–333, 2002. 10.4.2

[229] Jonas Peters, Joris M Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal discovery with continuous additive noise models. 2014. 4.6, 4.6, 2

[230] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016. 4.5.1

[231] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017. 4.1.1, 4.2.1

[232] Vibhor Porwal, Piyush Srivastava, and Gaurav Sinha. Almost optimal universal lower bound for learning causal dags with atomic interventions. In *International Conference on Artificial Intelligence and Statistics*, pages 5583–5603. PMLR, 2022. 5.7, 5.11

[233] Nikita Puchkin and Nikita Zhivotovskiy. Exponential savings in agnostic active learning through abstention. In *Conference on Learning Theory*, pages 3806–3832. PMLR, 2021. 2.14.1

[234] Aaditya Ramdas, Johannes Ruf, Martin Larsson, and Wouter Koolen. Admissible anytime-valid sequential inference must rely on nonnegative martingales. *arXiv preprint arXiv:2009.03167*, 2020. 5.7, 5.8.2, 4

[235] Aaditya Ramdas, Peter Grünwald, Vladimir Vovk, and Glenn Shafer. Game-theoretic statistics and safe anytime-valid inference. *Statistical Science (forthcoming)*, 2023. 5.1

[236] Bashir Rastegarpanah, Krishna Gummadi, and Mark Crovella. Auditing black-box prediction models for data minimization compliance. *Advances in Neural Information Processing Systems*, 34, 2021. 6.5

[237] J. Rawls. *A Theory of Justice*. Harvard University Press, 1971. 10.1

[238] Sid Reddy, Anca Dragan, and Sergey Levine. Where do you think you're going?: Inferring beliefs about dynamics from behavior. *Advances in Neural Information Processing Systems*, 31, 2018. 9.5

[239] K. Regan and C. Boutilier. Regret-based reward elicitation for Markov decision processes. In *25th*, pages 444–451, 2009. 10.1

[240] K. Regan and C. Boutilier. Robust policy computation in reward-uncertain MDPs using

nondominated policies. In *24th*, pages 1127–1133, 2010. 10.1

[241] Arnaud Robert, Ciara Pike-Burke, and Aldo A Faisal. Sample complexity of goal-conditioned hierarchical reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 11.1.1.2, 11.2, 11.7

[242] Lior Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39, 2010. 7.1

[243] Dana Ron. *Property testing: A learning theory perspective*. Now Publishers Inc, 2008. 6.5

[244] S. Russell, D. Dewey, and M. Tegmark. Research priorities for robust and beneficial artificial intelligence. *AI Magazine*, 36(4):105–114, 2015. 10.1

[245] Sivan Sabato, Anand D Sarwate, and Nathan Srebro. Auditing: active learning with outcome-dependent query costs. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 1*, pages 512–520, 2013. 2.14.1, 6.5

[246] Eduardo Salas, Dana E Sims, and C Shawn Burke. Is there a ?big five? in teamwork? *Small group research*, 36(5):555–599, 2005. 7.1

[247] William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*, 2022. 11.1, 11.6

[248] Antoine Scheid, Daniil Tiapkin, Etienne Boursier, Aymeric Capitaine, El Mahdi El Mhamdi, Éric Moulines, Michael I Jordan, and Alain Durmus. Incentivized learning in principal-agent bandit games. *arXiv preprint arXiv:2403.03811*, 2024. 9.1, 9.3, 9.7, 9.10, 9.12, 9.13

[249] Erik Schluntz and Barry Zhang. Building effective agents, dec 2024. URL `https://www.anthropic.com/engineering/building-effective-agents`. Accessed: 2025-12-06. 1

[250] D. Schmeidler. The nucleolus of a characteristic function game. *SIAM Journal on Applied Mathematics*, 17(6):1163–1170, 1969. 8.2

[251] Andrew Selbst and Julia Powles. “meaningful information” and the right to explanation. In *Conference on Fairness, Accountability and Transparency*, pages 48–48. PMLR, 2018. 3.1

[252] Lesia Semenova, Cynthia Rudin, and Ronald Parr. A study in rashomon curves and volumes: A new perspective on generalization and model simplicity in machine learning. *arXiv preprint arXiv:1908.01755*, 2019. 3.6.2, 3.10

[253] Sequoia Capital. Pricing in the AI era: From inputs to outcomes, with Paid CEO Manny Medina. Sequoia Capital Podcast, 2025. URL `https://sequoiacap.com/podcast/pricing-in-the-ai-era-from-inputs-to-outcomes-with-paid-ceo-manny-medi` 9.2.2

[254] Arjun Seshadri, Alexander Peysakhovich, and Johan Ugander. Discovering context effects from raw choice data. *ICML 2019*, 2019. 7.4.2, 7.8.2, 7.8.8

[255] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to*

Algorithms. Cambridge University Press, 2014. 8.7, 8.8

[256] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014. 7.8.2

[257] Karthikeyan Shanmugam, Murat Kocaoglu, Alexandros G Dimakis, and Sriram Vishwanath. Learning causal graphs with small interventions. *Advances in Neural Information Processing Systems*, 28, 2015. 5.1, 5.2

[258] Yonadav Shavit, Benjamin Edelman, and Brian Axelrod. Causal strategic linear regression. In *International Conference on Machine Learning*, pages 8676–8686. PMLR, 2020. 4.1

[259] Han Shen, Zhuoran Yang, and Tianyi Chen. Principled penalty-based methods for bilevel reinforcement learning and rlhf. *arXiv preprint arXiv:2402.06886*, 2024. 9.4.2, 9.8.2.2, 9.8.2.2, 9.12

[260] Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(10), 2006. 5.7

[261] Yoav Shoham, Rob Powers, and Trond Grenager. If multi-agent learning is the answer, what is the question? *Artificial intelligence*, 171(7):365–377, 2007. 7.7

[262] Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35: 9460–9471, 2022. 4.9

[263] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020. 3.10

[264] Ruike Song, Jiawen Li, Zhenlian Niu, and Yixin Gu. Causal reward adjustment: Mitigating reward hacking in external reasoning via backdoor correction. *arXiv preprint arXiv:2508.04216*, 2024. 4.4

[265] Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search*. MIT press, 2000. 4.6, 2, 5.1, 5.2

[266] Chandler Squires, Sara Magliacane, Kristjan Greenewald, Dmitriy Katz, Murat Kocaoglu, and Karthikeyan Shanmugam. Active structure learning of causal dags via directed clique trees. *Advances in Neural Information Processing Systems*, 33:21500–21511, 2020. 5.7, 5.11

[267] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020. 11.1, 11.6

[268] Stripe. Outcome-based pricing: A guide for businesses. Stripe Resources, 2025. URL `https://stripe.com/en-br/resources/more/outcome-based-pricing`. 9.2.2

[269] E. Štrumbelj and I. Kononenko. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11:1–18, 2010. 8.1, 8.4

[270] Theodore Sumers, Raj Agarwal, Nathan Bailey, Tim Belonax, Brian Clarke, Jasmine Deng, Evan Frondorf, Kyla Guru, Keegan Hankes, Jacob Klein, Lynx Lean, Kevin Lin, Linda Petrini, Madeleine Tucker, Ethan Perez, Mrinank Sharma, and Nikhil Saxena. Monitoring computer use via hierarchical summarization, 2025. URL `https://alignment.anthropic.com/2025/summarization-for-monitoring`. 9.7

[271] Ravi Sundaram, Anil Vullikanti, Haifeng Xu, and Fan Yao. Pac-learning for strategic classification. In *International Conference on Machine Learning*, pages 9978–9988. PMLR, 2021. 4.1

[272] U. Syed and R. E. Schapire. A game-theoretic approach to apprenticeship learning. In *21st*, pages 1449–1456, 2008. 10.6

[273] X Tan and TT Lie. Application of the shapley value on transmission cost allocation in the competitive power market environment. *IEE Proceedings-Generation, Transmission and Distribution*, 149(1):15–20, 2002. 7.3

[274] Yufei Tao, Hao Wu, and Shiyuan Deng. Cross-space active learning on graph convolutional networks. In *International Conference on Machine Learning*, pages 21133–21145. PMLR, 2022. 1, 2.14.1

[275] Vinzenz Thoma, Barna Pásztor, Andreas Krause, Giorgia Ramponi, and Yifan Hu. Contextual bilevel reinforcement learning for incentive alignment. *Advances in Neural Information Processing Systems*, 37:127369–127435, 2024. 9.12

[276] Jeremy Tien, Jerry Zhi-Yang He, Zackory Erickson, Anca D Dragan, and Daniel S Brown. Causal confusion and reward misidentification in preference-based reward learning. *arXiv preprint arXiv:2204.06601*, 2022. 4.1.1, 4.4

[277] Panagiotis Tigas, Yashas Annadani, Andrew Jesson, Bernhard Schölkopf, Yarin Gal, and Stefan Bauer. Interventions, where and how? experimental design for causal models at scale. *Advances in Neural Information Processing Systems*, 35:24130–24143, 2022. 5.7

[278] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001. (document), 2.3.1, 2.2

[279] Christian Toth, Lars Lorch, Christian Knoll, Andreas Krause, Franz Pernkopf, Robert Peharz, and Julius Von Kügelgen. Active bayesian causal inference. *Advances in Neural Information Processing Systems*, 35:16261–16275, 2022. 5.7

[280] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 601–618, 2016. 3.10

[281] Stratis Tsirtsis and Manuel Gomez-Rodriguez. Decisions, counterfactual explanations and strategic behavior. *arXiv preprint arXiv:2002.04333*, 2020. 3.2

[282] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984. 2.10.4, 6.1

[283] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999. 2.4.1, 2.11.1

[284] Jean Ville. *Étude critique de la notion de collectif.* 1939. 3

[285] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944. 4.2.3

[286] Sandra Wachter, Brent Mittelstadt, and Luciano Floridi. Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *International Data Privacy Law*, 7(2):76–99, 2017. 3.1

[287] Samir Wadhwa and Roy Dong. On the sample complexity of causal discovery and the value of domain expertise. *arXiv preprint arXiv:2102.03274*, 2021. 5.7

[288] M. J. Wainwright. *High-dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, 2019. 10.9

[289] Chaoqi Wang, Zhuokai Zhao, Yifeng Chen, Yiting Li, Yang Yuan, Hao Peng, Heng Ji, et al. Beyond reward hacking: Causal rewards for large language model alignment. *arXiv preprint arXiv:2501.09620*, 2025. 4.4

[290] Francis Rhys Ward, Matt MacDermott, Francesco Belardinelli, Francesca Toni, and Tom Everitt. The reasons that agents act: Intention and instrumental goals. *arXiv preprint arXiv:2402.07221*, 2024. 4.4

[291] Zheng Wen, Doina Precup, Morteza Ibrahimi, Andre Barreto, Benjamin Van Roy, and Satinder Singh. On efficiency in hierarchical reinforcement learning. *Advances in Neural Information Processing Systems*, 33:6708–6718, 2020. 11.2, 46, 11.7, 12, 11.10

[292] WGA. Wga negotiations—status as of may 1, 2023. *Writers Guild of America*, 2023. URL `https://www.wga.org/uploadedfiles/members/member_info/contract-2023/WGA_proposals.pdf`. 2.1

[293] Maranke Wieringa. What to account for when accounting for algorithms: A systematic literature review on algorithmic accountability. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 1–18, 2020. 3.1

[294] M. A Williams. An empirical test of cooperative game solution concepts. *Behavioral Science*, 33(3):224–237, 1988. 8.4

[295] Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017. 11.2

[296] Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*, 2021. 11.1, 11.1.1.2, 11.6

[297] Jibang Wu, Siyu Chen, Mengdi Wang, Huazheng Wang, and Haifeng Xu. Contractual reinforcement learning: Pulling arms with invisible hands. *arXiv preprint arXiv:2407.01458*, 2024. 9.1, 9.3, 9.12

[298] Yueh-Hua Wu and Shou-De Lin. A low-cost ethics shaping approach for designing reinforcement learning agents. In *32nd*, pages 1687–1694, 2018. 10.11.1

[299] Yichong Xu, Ruosong Wang, Lin Yang, Aarti Singh, and Artur Dubrawski. Preference-

based reinforcement learning with finite-time guarantees. *Advances in Neural Information Processing Systems*, 33:18784–18794, 2020. 11.2

[300] Zelai Xu, Tiancheng Yu, and Suvrit Sra. Towards efficient evaluation of risk via herding. *Negative Dependence: Theory and Applications in Machine Learning*, 2019. 6.5

[301] Ziyu Xu, Ruodu Wang, and Aaditya Ramdas. A unified framework for bandit multiple testing. In *Neural Information Processing Systems*, 2021. 5.7

[302] Songbai Yan, Kamalika Chaudhuri, and Tara Javidi. Active learning from noisy and abstention feedback. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1352–1357. IEEE, 2015. 2.6

[303] Songbai Yan, Kamalika Chaudhuri, and Tara Javidi. Active learning from imperfect labelers. *Advances in Neural Information Processing Systems*, 29, 2016. 2.4.2, 2.6, 2.11.2

[304] Tom Yan and Zachary Lipton. A theoretical case-study of scalable oversight in hierarchical reinforcement learning. *Advances in Neural Information Processing Systems*, 37:27295–27339, 2024. 1.2.2

[305] Tom Yan and Ariel D Procaccia. If you like shapley then you'll love the core. 7.8.2

[306] Tom Yan and Ariel D Procaccia. If you like shapley then you'll love the core. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5751–5759, 2021. 1.2.1

[307] Tom Yan and Chicheng Zhang. The human-ai substitution game: active learning from a strategic labeler. In *The Twelfth International Conference on Learning Representations*. 1.1

[308] Tom Yan and Chicheng Zhang. Active fairness auditing. In *International Conference on Machine Learning*, pages 24929–24962. PMLR, 2022. 1.1

[309] Tom Yan and Chicheng Zhang. Margin-distancing for safe model explanation. In *International Conference on Artificial Intelligence and Statistics*, pages 5104–5134. PMLR, 2022. 1.1, 4.1

[310] Tom Yan and Chicheng Zhang. Stackelberg learning with outcome-based payment. 2025. 1.2.1

[311] Tom Yan, Ziyu Xu, and Zachary Chase Lipton. Foundations of testing for finite-sample causal discovery. In *Forty-first International Conference on Machine Learning*. 1.1

[312] Tom Yan, Christian Kroer, and Alexander Peysakhovich. Evaluating and rewarding teamwork using cooperative game abstractions. *Advances in Neural Information Processing Systems*, 33:6925–6935, 2020. 1.2.1

[313] Tom Yan, Shantanu Gupta, and Zachary Lipton. Discovering optimal scoring mechanisms in causal strategic prediction. *arXiv preprint arXiv:2302.06804*, 2023. 1.1

[314] I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480, 2009. 3.6.1

[315] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhut-

dinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, pages 3391–3401, 2017. 7.6.1

[316] Zendesk. Zendesk first in CX industry to offer outcome-based pricing for AI agents. Zendesk Newsroom, 2025. URL `https://www.zendesk.com/newsroom/articles/zendesk-outcome-based-pricing/`. 9.2.2

[317] Wenhao Zhan, Masatoshi Uehara, Wen Sun, and Jason D Lee. Provable reward-agnostic preference-based reinforcement learning. *arXiv preprint arXiv:2305.18505*, 2023. (document), 11.1.2, 11.2, 11.4.1, 11.4.2, 81, 10, 11.9.2, 11.9.2.1, 88, 89, 90, 59, 91, 11.9.2.4, 93

[318] Chicheng Zhang and Kamalika Chaudhuri. Beyond disagreement-based agnostic active learning. *Advances in Neural Information Processing Systems*, 27, 2014. 6.3.1.1

[319] Hanrui Zhang and Vincent Conitzer. Incentive-aware pac learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5797–5804, 2021. 4.1

[320] Jiaqi Zhang, Chandler Squires, and Caroline Uhler. Matching a desired causal state via shift interventions. *Advances in Neural Information Processing Systems*, 34:19923–19934, 2021. 4.6, 1

[321] Kun Zhang and Aapo Hyvarinen. On the identifiability of the post-nonlinear causal model. *arXiv preprint arXiv:1205.2599*, 2012. 5.7

[322] Geng Zhao, Banghua Zhu, Jiantao Jiao, and Michael Jordan. Online learning in stackelberg games with an omniscient follower. In *International Conference on Machine Learning*, pages 42304–42316. PMLR, 2023. 9.3

[323] J. Zheng, S. Liu, and L. M. Ni. Robust Bayesian inverse reinforcement learning with sparse behavior noise. In *28th*, pages 2198–2205, 2014. 10.1

[324] Han Zhong, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Can reinforcement learning find stackelberg-nash equilibria in general-sum markov games with myopically rational followers? *Journal of Machine Learning Research*, 24(35):1–52, 2023. 9.12

[325] Banghua Zhu, Jiantao Jiao, and Michael I Jordan. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. *arXiv preprint arXiv:2301.11270*, 2023. 11.2, 11.4.1, 11.9.2.4

[326] Yinglun Zhu and Robert Nowak. Efficient active learning with abstention. *arXiv preprint arXiv:2204.00043*, 2022. 2.14.1

[327] Simon Zhuang and Dylan Hadfield-Menell. Consequences of misaligned ai. *Advances in Neural Information Processing Systems*, 33:15763–15773, 2020. 4.2.3

[328] B. D. Ziebart. Modeling purposeful adaptive behavior with the principle of maximum causal entropy. Ph.D. thesis, Carnegie Mellon University, 2010. 9.5, 10.7.2

[329] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *23rd*, pages 1433–1438, 2008. 10.1, 10.3, 10.7.2, 65

[330] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences.

arXiv preprint arXiv:1909.08593, 2019. 11.2