Deep Learning for Scalable Neural Interfaces

Joel Ye

September 2026

Neuroscience Institute and Machine Learning Department
Dietrich College of Humanities and Social Sciences
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Robert Gaunt (co-chair)
Leila Wehbe (co-chair)
Jennifer Collinger
Aran Nayebi
Chethan Pandarinath (Emory University)

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in Neural Computation and Machine Learning.

Copyright © 2026 Joel Ye

This work was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, through the Department of Energy Computational Science Graduate Fellowship under Award Number DE-SC0023112; and by the National Institute of Neurological Disorders and Stroke of the National Institutes of Health under Award Numbers UH3NS107714, U01NS108922, U01NS123125, and R01NS121079. This research was also supported in part by the University of Pittsburgh Center for Research Computing (RRID:SCR_022735) through the resources provided, including the H2P cluster supported by National Science Foundation Award Number OAC-2117681.

The content is solely the responsibility of the author and does not necessarily represent the official views of the National Institutes of Health.

Keywords: brain–computer interfaces, intracortical BCI, neural decoding, foundation models, pretraining, transfer learning, motor cortex, spiking activity, benchmarking, intracortical microstimulation, artifact removal

There are more things in heaven and earth, Horatio,
Than are dreamt of in your philosophy.

William Shakespeare, *Hamlet*, I.v

Abstract

The increasing ambition of neuroscience and neurotechnology both supplies and demands vast new quantities of neural data, which creates a critical need for methods that can operate effectively on neural data at scale. This need could be addressed by deep learning. Here, we assess this program for sensorimotor intracortical brain-computer interfaces (iBCI). iBCI systems have traditionally modeled the brain through short datasets to relate a user's neural activity with their bodily state. Although this approach can enable both motor control and sensory feedback, it is brittle, as the models built from a small snapshot of brain data rarely outlive the controlled context in which the data was collected. Real-world iBCI use will demand exquisite reliability and convenience, which in turn requires that iBCI systems evolve to operate on brain data with the texture and scale of everyday use. We approach this in three steps: assembling the groundwork for large-scale neural data models, executing the scaling and evaluating what it yields, and then moving upstream to refine the preprocessing of the neural data that these models are built on. Our first contribution lays two pieces of groundwork: an architecture and an evaluation. For the architecture, we design a Transformer deep network (Neural Data Transformer 2, NDT2) that can be productively trained on data from across time, subjects, and tasks, and use this Transformer to establish promising scaling trends that motivate larger-scale training. For the evaluation, we create a decoding benchmark (FALCON) to measure the consistency of BCI models across sessions under few-shot recalibration. Our second contribution is to execute the scaling, by training and evaluating across a large corpus of intracortical motor data. The resulting model (NDT3) provides a baseline demonstration of the foundation model concept for intracortical motor decoding across settings, but its returns suggest that cross-subject transfer on intracortical data presents a fundamental challenge, motivating careful design of evaluations for neural foundation models. Our third contribution moves from decoding spiking activity to preprocessing and denoising broadband neural data under intracortical microstimulation. Our artifact rejection model (DELETE) casts preprocessing as information compression, treating locally predictable data as artifact and its complement as informative neural signal. This framing allows robust recovery of spiking activity under strong and varied electrical artifacts. The ambition of scaled neural interfaces is immense. For neuroscientific understanding of the brain, scaled models hint at a path to understanding the brain

in its intricate detail. For neurotechnology, scaled models promise BCIs that restore independence as conveniently and personally as consumer electronics. This thesis provides an infrastructure to build towards this goal.

Acknowledgments

I am grateful to my advisors, Rob and Leila, for granting me windows into their worlds and supporting me, as irritable as I can be. I am also grateful to the members of my committee for their time and advice, and more specifically: Chethan, who has always been in my corner from across the country. Jen, for being a beacon of stability and pragmatism on which the BCI team relied. And Aran, for providing proof by example that you really can just do things. Having an academic lineage feels archaic, but I have realized parts of myself through your collective guidance.

I am grateful to my family, mom, dad, and sister, and longtime friends, Zicheng and Kenneth, for humoring me when I ramble and keeping me grounded.

Contents

1 Introduction **1**
1.1 Motivation 1
1.2 Contributions and organization 3

2 Background and Related Work **5**
2.1 Brain–computer interfaces 5
2.1.1 The basics 5
2.1.2 Recording and decoding motor intent 7
2.1.3 Restoring sensation with intracortical microstimulation 8
2.2 Deep learning for neural data 10
2.2.1 Deep networks for BCI data 10
2.2.2 Foundation models for neural data 11

3 Neural Data Transformer 2 **14**
3.1 Introduction 14
3.2 Related Work 16
3.3 Approach 18
3.3.1 Designing Transformers for unsupervised scaling on neural data 18
3.3.2 Datasets 19
3.4 Results 20
3.4.1 NDT2 enables multicontext pretraining 22
3.4.2 NDT2 scaling across contexts 24
3.4.3 Using NDT2 for improved decoding on novel days 26
3.4.4 NDT2 for human cursor control pilots. 27
3.5 Discussion 29

4 The FALCON Benchmark **31**
4.1 Introduction . . . 31
4.1.1 Related work . . . 33
4.2 Benchmark evaluation pipeline and metrics . . . 34
4.2.1 Evaluation strategy and pipeline . . . 34
4.2.2 Supported approaches and benchmark scope . . . 35
4.2.3 Metrics . . . 36
4.3 Datasets . . . 37
4.4 Results . . . 40
4.4.1 FALCON datasets exhibit unstable decoding performance across sessions. . . . 41
4.4.2 FALCON baselines demonstrate the difficulty of improving M1 decoder stability. . . . 43
4.4.3 FALCON baseline performance drops from held-in to held-out datasets. . . . 44
4.5 Discussion . . . 45

5 A Generalist Motor Decoder **49**
5.1 Introduction . . . 49
5.2 Approach . . . 51
5.2.1 Data . . . 51
5.2.2 Model . . . 53
5.2.3 Evaluation Strategy . . . 54
5.3 Results . . . 55
5.3.1 Multi-scale evaluation across motor decoding tasks . . . 56
5.3.2 NDT3 does not generalize to novel input order and output conditions . . . 57
5.3.3 NDT3 does generalize to varied real world distribution shifts. . 60
5.4 Discussion . . . 61

6 DELETE **66**
6.1 Introduction . . . 66
6.2 Methods . . . 68
6.2.1 DELETE approach . . . 68
6.2.2 Datasets . . . 70

6.2.3 Baseline artifact denoising methods . . . 74
6.2.4 Evaluation metrics . . . 76
6.2.5 Neurometric threshold estimation . . . 78
6.3 Results . . . 80
6.3.1 DELETE matches the best baseline across artifact simulations 81
6.3.2 Hyperparameter sensitivity, training, and deployment . . . 84
6.3.3 DELETE preserves artifact-free recordings . . . 87
6.3.4 Evaluation of spike recovery in real-world ICMS datasets . . . 89
6.3.5 ICMS detection threshold correlation validates DELETE denoising 92
6.4 Discussion . . . 95

7 Discussion 98
7.1 Summary . . . 98
7.1.1 Spiking data spanning sessions, subjects, and tasks can be pretrained over as one corpus . . . 98
7.1.2 BCI decoding needs more accessible benchmarks . . . 99
7.1.3 A baseline motor foundation model suggests limits to scaling . 100
7.1.4 Stimulation artifact can be mitigated by learned reconstruction 101
7.2 Future directions . . . 102
7.2.1 Closed-loop deployment . . . 102
7.2.2 Modeling and optimising the neural response to ICMS . . . 104
7.2.3 Somatosensory alignment . . . 105
7.3 Coda . . . 106

A Supplement: NDT2 107
A.1 Supplementary Material . . . 107
A.1.1 Dataset Preparation . . . 107
A.1.2 Methods for online cursor control experiments . . . 109
A.1.3 Compute and Hyperparameter Tuning . . . 109
A.1.4 NDT2 Design Notes . . . 110
A.1.5 Additional exploratory experiments . . . 113
A.1.6 Single-session breakdown of experimental results . . . 114

B Supplement: FALCON 119
B.1 Supplementary Material . . . 119

B.1.1 Computational Resources 119
B.1.2 Interpretation of FALCON Metrics 120
B.1.3 Datasets 120
B.1.4 Baseline Implementation Hyperparameters 130
B.1.5 Evaluation Parameters 138

C Supplement: NDT3 **141**
C.1 Appendix 141
C.1.1 Related works and a proposed taxonomy 142
C.1.2 Supplementary Results 144
C.1.3 Methods 157

D Supplement: DELETE **169**
D.1 Supplementary Material 169
D.1.1 DELETE suppresses highly entrained neural responses 169
D.1.2 Synthetic spike injection and normalized recall 170
D.1.3 Sensitivity of Activity Synchrony to the co-activation threshold 171
D.1.4 Interpreting waveform R^2 discrepancies 171
D.1.5 Neurometric threshold construction and per-method threshold correlations 177

List of Figures

2.1 A sensorimotor BCI control loop and its performance lifetime 5

3.1 NDT2 Approach 18
3.2 Model Training 21
3.3 NDT2 enables pretraining 23
3.4 Scaling of transfer on RTT 24
3.5 Tuning and evaluating pretrained decoders 27

4.1 FALCON Evaluation Design 34
4.2 FALCON datasets span iBCI use cases 37
4.3 Static decoders exhibit decoding instabilities on FALCON datasets . 42
4.4 Baseline model predictions on M1-A 43

5.1 NDT3 Schematic 51
5.2 NDT3 Data and Model Design 52
5.3 NDT3 primary evaluation 63
5.4 NDT3 permutation robustness diagnostic 64
5.5 Generalization of pretraining gains 65

6.1 DELETE schematic 68
6.2 DELETE training and deployment 69
6.3 ICMS datasets 72
6.4 Standardized spike extraction from human ICMS recordings 74
6.5 DELETE evaluation in simulation 82
6.6 DELETE training and deployment 85
6.7 DELETE preservation of non-stimulation datasets 87
6.8 DELETE evaluation on real ICMS data 90

6.9 Neurometric and psychometric detection thresholds 93

7.1 Pretrained and linear controllers are constrained by label quality . . . 103

A.1 Context embedding ablations 111
A.2 Neuron patch size ablation 112
A.3 Sweeps for regularization parameters 113
A.4 NDT2 sensitivity to mask ratio 114
A.5 NDT2 individual dataset breakout 115
A.6 Session context token ablation at small data scales 116
A.7 Individual dataset scaling trend breakout 117
A.8 Single dataset decoding by fine-tuning of NDT2 118

B.1 Sweeps to determine WF history 131
B.2 Continuous vs Trialized Decoding 140

C.1 Pretraining does not impact angular generalization (secondary experiment) 143
C.2 Covariate masking ablation 145
C.3 NDT3 primary ablations 146
C.4 Scaling of unsupervised pretraining 148
C.5 NDT3 fails Seq2Seq handwriting decoding task 149
C.6 NDT3 primary evaluation breakout 151
C.7 Sequential vs joint fine-tuning for permutation diagnostic 152
C.8 Benchmark results with more NDT3 variants 153
C.9 NDT3 pretraining curves 154
C.10 NDT3 fine-tuning training curve 155
C.11 NDT3 fine-tuning with joint neural and behavior objective outperforms behavior only 157
C.12 NDT3 HP sweep 158
C.13 NDT3 vs NDT2 architecture 161
C.14 NDT3 vs POYO comparison 162

D.1 DELETE attenuates artificially synchronized activity 169
D.2 Injected spike recovery 170
D.3 Synchrony threshold sensitivity 172
D.4 Waveform R-squared dataset breakout 174

D.5 Neurometric threshold detailed methods 177

List of Tables

3.1 Neural data pretraining 16
3.2 Human reach intent decoding 28

4.1 FALCON baselines 45

B.1 FALCON movement baselines with zero-shot and static NDT2 132
B.2 Wiener Filter calibration of held-out dataset size 139

C.1 NDT neural prediction metrics on the Neural Latents benchmark . . 157
C.2 NDT2 fine-tuning patience ablation 160
C.3 NDT3 evaluation datasets 167
C.4 Transformer model shapes 168

Chapter 1

Introduction

1.1 Motivation

> In that Empire, the Art of Cartography attained such Perfection that the map of a single Province occupied the entirety of a City, and the map of the Empire, the entirety of a Province. In time, those Unconscionable Maps no longer satisfied, and the Cartographers Guilds struck a Map of the Empire whose size was that of the Empire, and which coincided point for point with it.
>
> — Jorge Luis Borges, "On Exactitude in Science," trans. Andrew Hurley [1]

The brain, in its marvelous detail, has been the subject of mapping and modeling for as long as we have been able to observe it. This endeavor of data collection and analysis constitutes our account of the brain, dating back to the first recordings of giant squid axons and Penfield's pioneering work on mapping of human sensorimotor cortex. Though we have come far from these first experiments, our understanding of the brain remains deeply limited.

Like the Cartographers, though our collective goal might be daunting, we proceed with our daily work in neuroscience in a flurry of data collection. More broadly, our field has an intense focus on advancing our instruments to interface with larger portions of the brain, at greater spatiotemporal resolution. Structural mapping is advancing from whole fly brains toward the larger scales of mouse and human brains [2–4]. For brain function, the number of simultaneously recorded neurons in animal electrophysiology and imaging follows a steady exponential growth [5], and the experiments in which

these recordings occur are increasingly complex and challenging [6].

The implicit premise of these efforts is that with a sufficiently exquisite measure and model, we will advance towards our grand goal of a deeper, unified account of the brain. Criticism within neuroscience of exhaustive data collection, in that it might not provide the understanding we seek, is not uncommon [7–9]. Indeed, Borges continues:

> The following Generations, who were not so fond of the Study of Cartography as their Forebears had been, saw that that vast Map was Useless, and not without some Pitilessness was it, that they delivered it up to the Inclemencies of Sun and Winters. In the Deserts of the West, still today, there are Tattered Ruins of that Map, inhabited by Animals and Beggars.

This thesis' central project is nonetheless to make like the Cartographers and chart a path forward for large-scale brain data modeling. We pursue this goal unabashed by the fable's warning because we have impetus beyond understanding, namely the practical promise of scaled neurotechnology. Fortunately, the devices we build to understand the brain are readily usable for practical purposes. For instance, the very same electrodes that are used to observe neural activity as, say, a monkey reaches for a banana, can be wired to power a brain-computer interface (BCI) that moves a robot hand to grab that same banana. BCIs hold large rehabilitative potential, as they can bypass an individual's native neural pathways that might be damaged from spinal cord injury or neurodegeneration, to allow direct sensorimotor interaction with the world.

As in the rest of neuroscience, BCI data collection is only growing in scale. In particular, BCI systems developed for sensorimotor impairments have achieved enough success in research settings that a number of companies are now competing to commercialize the technology. The models we develop for BCI systems thus must now be compatible with many different potential users at high reliability. More tersely, we need models that are compatible with large-scale BCI data.

This demand is naturally met with deep learning. The past decade has been defined by the explosion of deep learning methods first across natural language processing and computer vision, and then into nearly every data-driven field. This broad adoption of deep learning across domains has cemented a number of narrative expectations. Performance improves with the scale and breadth of the data a model is pretrained on, rather than with the ingenuity of the model's method. The progress of scaling efforts should be measured by pre-established benchmarks, as individual claims are

noisy from the complexity of coupled algorithm design and data pipelines. Finally, deep learning offers a way to rethink the data pipelines themselves.

We now pursue this program for BCI data.

1.2 Contributions and organization

The remainder of this thesis is organized as follows.

- **Chapter 2** briefly reviews background content. We first cover intracortical brain–computer interfaces, enumerating their current capabilities and limitations. We then provide basic priming for deep learning methods and how they have been applied to BCI data.
- **Chapter 3**, reproduced from Ye et al. [10], develops Neural Data Transformer 2 (NDT2), a spatiotemporal Transformer for neural spiking activity, and demonstrates that pretraining can leverage motor BCI datasets spanning sessions, subjects, and experimental tasks. We show that NDT2 enables rapid adaptation to novel contexts in downstream decoding, and that the benefit of pretraining scales differently across context classes.
- **Chapter 4**, reproduced from Karpowicz et al. [11], introduces the FALCON benchmark suite (Few-shot Algorithms for COnsistent Neural decoding), which standardizes evaluation of iBCI robustness. FALCON curates five datasets of neural and behavioral data spanning movement and communication tasks, each with calibration data, optional few-shot recalibration data, and private evaluation data, alongside an evaluation platform and seeded baselines. The resulting comparisons establish that consistent decoding across sessions remains unsolved.
- **Chapter 5**, reproduced from Ye et al. [12], quantifies the foundation model premise directly, pretraining an autoregressive Transformer on 2000 hours of neural population spiking activity paired with diverse motor covariates from over 30 monkeys and humans. The resulting model is broadly useful, benefiting decoding on 8 downstream tasks and generalizing across a variety of neural distribution shifts. But scaling from 200 to 2000 hours already requires increasing model size to 350M parameters to avoid saturation, several downstream datasets scarcely benefit from scale at all, and two targeted demonstrations trace this limit

to variability in the input and output spaces across neural datasets — variability that pretraining alone does not appear to resolve.

- **Chapter 6**, reproduced from Ye et al. [13], turns to the signal itself, presenting DELETE (Denoising Electrical Events with a Transformer Encoder), a reconstruction-based denoiser built on the premise that neural signal is harder to predict than artifact. DELETE matches the best baselines across multiple artifact simulations, preserves neural signal in stimulation-free recordings, and recovers neural activity in intracortical microstimulation datasets where artifacts dwarf the responses of interest.
- **Chapter 7** reviews the contributions of this thesis in the context of a broader roadmap for building large-scale models of neural data. We first discuss what the presented work leaves critically unanswered, and then what the work enables for future directions.

Chapter 2

Background and Related Work

2.1 Brain–computer interfaces

2.1.1 The basics

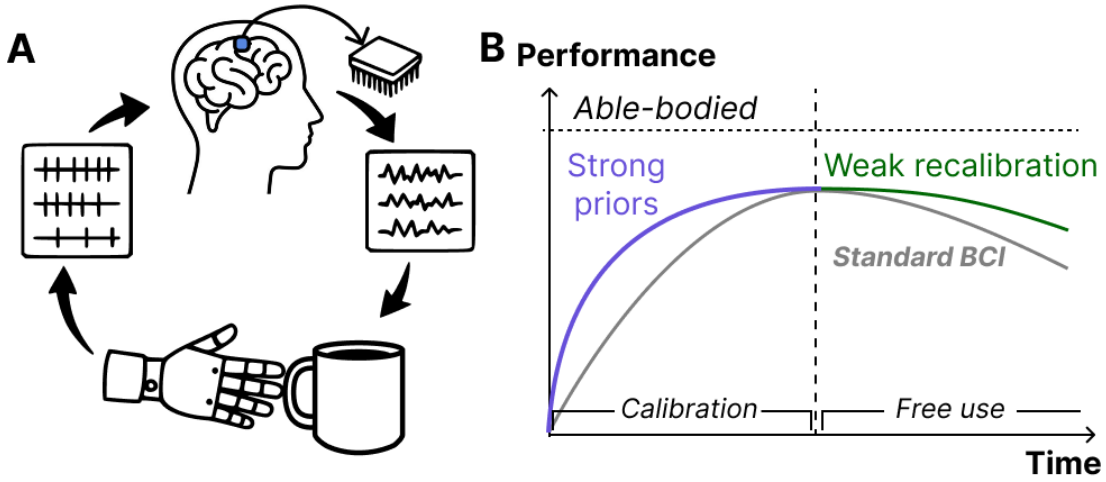

Figure 2.1. A) A sensorimotor BCI-enabled control loop. By recording neural activity from an implanted microelectrode array, this BCI can allow the user to control a robotic hand. When touch sensors on the hand contact a physical object, we can deliver stimulation pulses back to the brain to provide the user with sensory feedback. B) Motor BCI models must be calibrated to relate neural activity at the implant interface with the user's intentions. Performance increases over an explicit calibration phase, which can be accelerated with learned priors, but will decline over time during free use of the system. This decline can be mitigated with varied recalibration methods.

A brain-computer interface (BCI) is a system that allows observation and control of neural activity. In the rehabilitative setting, for example, BCIs can record from

motor cortex to predict movement intentions and can control sensory cortex to evoke sensations. Together, the premise of a sensorimotor BCI is that we might directly relate a user's neural activity with their sensorimotor experience, and thereby restore a degree of sensorimotor function to a paralyzed user (Fig. 2.1A).

Over the last two decades, a number of studies have demonstrated the efficacy of human BCIs based on implanted microelectrode arrays [14, 15]. With implants in motor cortex recording neuronal spiking activity, users have achieved control of digital cursors, robotic arms, and language and speech generation [16–19]. With stimulation of implants in somatosensory cortex, users perceive tactile sensations on their hand, which is valuable as an end in itself but can also be utilized functionally for more skillful motor control [20–23]. BCI algorithms research now unfolds on several axes, naturally including expanding BCI capabilities to new functional milestones [24, 18], but also making the existing repertoire reliably performant across users and time [25, 26], and minimizing user burden for system adoption [15, 27].

Though this thesis focuses on implanted microelectrode arrays, we briefly review the broader landscape of neural interfaces, which span a range of invasiveness, resolution, and coverage. Historically, most BCI demonstrations have been electrode-based, for electrophysiology's high temporal resolution and ready use in stimulation as well as recording. When macroelectrodes are applied to the scalp, we term the modality electroencephalography (EEG); EEG provides whole-brain coverage at minimal signal quality. This has historically provided minimal information rate that could only meaningfully drive clickers or spellers, in which the user selects among options presented on a display and throughput is counted in characters per minute [15, 28]. Electrocorticography (ECoG) and depth electrodes (as in sEEG) use intermediate electrode sizes, and consequently have intermediate spatial coverage and resolution. ECoG has supported long-horizon decoding [29, 30] and is a compelling candidate for the rehabilitative use case that the microelectrodes we study also target.

BCI systems have been prototyped for many more modalities — functional near-infrared spectroscopy, endovascular devices, ultrasound, and even functional magnetic resonance imaging (fMRI). The same holds for writing in: beyond electrical stimulation, optogenetic and chemogenetic tools [31, 32] can address genetically defined populations rather than whatever lies near an electrode, and focused ultrasound offers a route to deep structures without an implant [33]. Despite the wide variety in these modalities, all implied BCI systems promise potential scaled data collection. It is not known, however,

to what extent large-scale data can qualitatively overcome any given modality's classically understood capabilities given by raw signal quality.

2.1.2 Recording and decoding motor intent

Motor decoders and BCI control are enabled by the tuning of motor cortex neurons to movement [34]. Exploiting that different neurons in a population are tuned to different directions of movement, we can build a basic decoder simply by summing preferred directions weighted by normalized firing rates [34, 35]. Optimal linear estimators refine this by weighting each unit's contribution to correct for a non-uniform distribution of preferred directions, while Kalman filters instead add a smoothness prior over the decoded kinematic state. The former enabled high-performance robotic arm control [16, 36], while the latter is a workhorse method for cursor BCIs [37]. We refer the reader to Pandarinath and Bensmaia [15] for a fuller treatment of the progression of motor BCI algorithms.

Innovations in motor BCI control focus on two main axes. The first axis is achieving high-bandwidth and high-degree-of-freedom control. After the first cursor control BCIs were demonstrated in monkeys [38–40] and then in a person with tetraplegia [41], considerable focus was given to improving control bandwidth, as surfaced by the fluency and responsiveness of the cursor [37, 19]. This drove innovation in calibration data annotation strategies without much change in the underlying models. In parallel, research focused on applying motor BCIs to tasks with more degrees of freedom, with the waterline pushed to 10 degrees of freedom in robotic arms [16, 36]. Here the strategy again involved extended calibration and careful protocol development. Further degrees of freedom are challenging due to potential interference, or at least nonlinear interaction, between simultaneously active control dimensions and environmental context [36, 42–44].

The second axis for improvement is maximizing BCI performance over time. A schematic performance curve is shown in (Fig. 2.1B). BCI models are initialized with a calibration protocol where experimenters collect paired neural activity and motor intention labels by prompting the user to behave according to experimental cues. The gray arc shows how BCI performance will increase with this calibration period [45], but then decline over time due to signal nonstationarities that render the calibration data less relevant [46]. Nonstationarity is a blanket term here, as mechanically, a

number of different causes degrade decoders. The recorded signal may degrade as the electrode surface wears down over a characterized lifetime [47–50], and from turnover in the recorded population itself, where over half of units can become unstable from one day to the next [51–53]. The mapping from neural activity to the user's desired motor intention itself may also shift without the recording changing at all, as the same movement is produced under a different task context [44, 54, 55], a different body posture [56], or a different cognitive state [57, 58].

While recalibration can mitigate changes in the neural–behavioral mapping, we seek algorithmic solutions that minimize user burden. Some researchers have begun exploiting the relatively small data drift over time and straightforwardly treat a large corpus of calibration data across multiple days as decoder training. This strategy has been used to provide multiple months of continuous performance in speech BCIs and 2D cursor control BCIs [17, 59, 60, 25]. Multi-day data modeling has also enabled stable reach and grasp performance in an electrocorticographic (ECoG) device [29]. However, even with many days of data to calibrate an initial model, performance eventually degrades. In this event, performance can also be maintained without explicit supervised recalibration by leveraging behavioral priors (Weak recalibration in the figure) [61, 26] as a source of weak learning signal. However, none of these methods have yet been proven for continuous multi-DoF upper limb control, which requires a greater variety of behavior at higher temporal precision and ostensibly poses a greater challenge to model stability.

2.1.3 Restoring sensation with intracortical microstimulation

Somatosensory feedback is integral to motor control, supplying the contact, force, and slip cues that shape an ongoing motor plan [62]. Intracortical microstimulation (ICMS) aims to restore this channel directly, passing small-current stimulation pulses through electrodes implanted in somatosensory cortex to activate a nearby volume of tissue. In doing so, ICMS can emulate touch feedback for neuroprosthetics, and indeed has been shown to improve BCI control of a robotic arm [22]. Currently, ICMS can achieve a variety of perceptual effects through poorly modeled mechanisms. In humans, ICMS in area 1 evokes tactile percepts that localize to the hand in accordance with the known somatotopy [20], and can elicit proprioceptive as well as cutaneous sensations [23]. Two key ICMS parameters, stimulation amplitude and frequency, modulate detectability

and perceived intensity along characteristic psychometric curves [63], but their exact relationships with perception are electrode-dependent [64].

Reflecting the field's relative immaturity in comparison with motor BCIs, sensory BCIs do not particularly struggle with model stability because they lack precise models of evoked sensations to begin with. That is, while any given stimulus's evoked percept location and quality appear relatively stable over time [65], the relationships between precise features of the evoked sensations, such as intensity and duration, and electrode choice and stimulation parameters are difficult to predict [66], and drift slowly over time [67, 68]. The consequence is that sensory BCI capabilities are currently developed by collecting a library of perceptual responses to varied stimulation patterns and curating the electrodes that produce desired percepts for use in functional tasks. Composing stimulation patterns or evoking more naturalistic percepts are difficult research programs as most progress has depended on expensive trial-and-error behavioral assessments, though see [69, 70].

One promising framework for future stimulation design is biomimicry: designing ICMS to evoke neural responses resembling those of natural touch, rather than directly matching stimulation frequencies or amplitudes to physical parameters [71]. Current biomimetic approaches reproduce coarse, subject-general features of natural touch [72], while feedback is otherwise typically delivered under the assumption that neural responses scale proportionally with stimulation frequency or amplitude [65]. These coarsely biomimetic patterns have improved percept naturalness [72], but optimizing ICMS to reproduce natural neural responses more fully will require models of how those responses depend on stimulation [73].

Evoked neural responses have long been investigated as guides to stimulation in cochlear implants and deep brain stimulation [74, 75], and now support closed-loop spinal cord stimulation [76]. For optimizing ICMS, a particular goal is to shape neural responses at millisecond timescales [77]. Electrical artifacts complicate modeling at these timescales, and recent ICMS-optimization frameworks have instead targeted activity measured after stimulation trains have ended [78, 79]. At the same time, recordings between individual stimulation pulses are already possible with appropriate recording hardware and artifact processing [77, 80]. These capabilities open a path toward ICMS control based on the neural responses unfolding during stimulation, rather than only those measured after a train.

2.2 Deep learning for neural data

2.2.1 Deep networks for BCI data

Modern deep learning organizes applications research by the data being modeled, the model architecture, and the optimization objective. We will take this approach to introduce deep networks in BCI.

Data: The neural data recorded from our BCIs are multichannel voltage time series. Traditionally, and throughout most of this thesis, this broadband voltage is filtered and thresholded to extract rapid, transient deflections, which mark putative spikes from a neuron nearby the recording electrode (though see [81]). The typical BCI studied in this thesis will use a small number of Utah arrays [82, 83] implanted in non-human primates (NHPs) or humans, which yield 100 to 200 electrode channels of data with dozens of high-amplitude spiking waveforms, along with many more channels recording multiunit activity that is not easily separable into distinctive waveforms. Our motor datasets will also contain upper-limb covariates, for example in the form of kinematics or electromyography that vary on the timescale of seconds. In NHP datasets, these behaviors generally derive from physical sensors, but as mentioned in Section 2.1, a different strategy must be used to create human BCI datasets since human BCI users have impaired control of their native limb. Human BCI datasets instead assert a certain behavior and experimentally cue the user to attempt the behavior. This artificial association means that the behavior label is likely to be temporally warped and otherwise imprecise [84] relative to the unobserved motor intention of the user. The electrical stimuli used in sensory datasets comprise trains of biphasic current-controlled pulses, where each individual pulse's timing and amplitude can be varied. Perceptual reports to ICMS, though minimally discussed in this thesis, can be given in terms of binary detection, scalar ratings of intensity, or free-form language.

Architecture: The mainstays of broader deep learning architectures that have flourished in the last decade, including multi-layer perceptrons (MLPs) [85], convolutional neural networks (CNNs) [86], recurrent neural networks (RNNs) [87], and Transformers [88], have all been applied to BCI data, and have been compared directly in online decoding [89]. While in practice architectures have since been refined or mixed, many recent works have used either RNNs or Transformers [17, 59, 27]. The former is incumbent because its explicit dynamical system mirrors neural dynamics,

while the latter is empirically dominant outside neuroscience. RNNs process time-series data one timestep at a time, maintaining an internal state that evolves with the input data. An RNN provides an appealing interpretative lens for neural data, as its iterations describe the evolution of the neural dynamical system. The Transformer, in contrast, does not maintain a centralized state but instead learns the relationship between fragments of the data. Each of these fragments is called a token, and they could be, for example, the population activity vector at the start of an experimental trial, abstract metadata about the experiment, or the vertical velocity of a user's arm.

Objective: This thesis focuses on explicit data prediction objectives, namely regression and classification. This will be true despite the variety in the qualitative function of our different models. When the work's objective is to model neural data in isolation, we say we are interested in the model's representation learning. At other times, we say we are building decoders of neural data to predict behavior, or encoders of stimuli to predict neural activity. With the lens of deep networks in particular, these different terms all implementationally overlap, in that the underlying models are operating between one or two data modalities.

Challenges: BCI datasets feature a number of challenges that uniquely interact with deep learning. First, by experimental design, there is often a large component of the neural data that appears strongly connected to the covariate of interest, implying there is not an immediately clear role for nonlinear DNNs. For example, in the motor cortex, we can identify single neurons with firing activity that can be well characterized by a cosine tuning curve with two parameters, a preferred arm direction and depth of firing rate modulation to movement in that direction. To balance this simplicity, BCI datasets will often have nonstationarities [51, 53, 52], as discussed in Section 2.1.2. New cognitive states, neurophysiological changes at the electrode interface, and BCI system hardware state can all introduce shifts in the observed activity. These externalities are difficult to directly observe, so we must build systems that are implicitly robust to these shifts so as to enable stable BCI performance [60].

2.2.2 Foundation models for neural data

A modern trend in deep learning applications research is to create a model that achieves broad competence across tasks in a field, at which point the model can be considered a "foundation model." The term "foundation model" was most deservingly coined in

natural language processing [90], where a large-scale initial training phase (pretraining) across internet text created models that dominated popular benchmarks [91]. Empirical scaling laws characterized predictable reductions in language-model prediction loss with increases in training data, model size, and computation [92], which became broadly interpreted as improvements in data efficiency across all tasks with scale.

Upstream (pretraining) and downstream (target-task) performance can be correlated, but the relationship is nuanced [93] and usually depends on how much target-task data is available. Gains in low-data settings need not persist at the downstream data budgets available in practice. In object detection and segmentation, He et al. [94] showed that models trained from scratch could match models pretrained on ImageNet when given longer training on the target dataset, COCO. Pretraining accelerated convergence but did not necessarily improve final accuracy. Zoph et al. [95] further found that more labeled target data and stronger data augmentation diminished the benefit of ImageNet pretraining. The choice of pretraining source likewise mattered less as fine-tuning data increased [96]. To more comprehensively understand the efficacy of scaling pretraining, we should thus evaluate benefits at a variety of target-data budgets.

The fundamental challenge of replicating the success of pretraining for neural data is the separation of potential pretraining data on distinct axes, which imposes potential transfer learning challenges. For example, datasets vary in sessions, subjects, tasks, brain regions, and recording modalities. Cross-subject transfer learning is particularly important to overcome from the perspective of accelerating performance of devices for new BCI users, but is also particularly challenging, due to individual variability in neural activity, as well as variability in sensor or electrode placement. While this problem is particularly salient for implanted microelectrodes, even non-invasive scaling studies have reported little further benefit from adding subjects [97]. One promising success story for cross-subject transfer, though, is Meta's electromyography-based wristband achieving scaling of zero-shot performance on new users, after training with data from thousands of volunteers [98].

Neural foundation models are pursued for several different goals, depending on whether the model is framed as an unsupervised model over neural data, an encoder, or decoder. Unsupervised representation learning models learn representations from neural data alone, typically with masked reconstruction or contrastive objectives [99–105, 10]. These models have the most direct claim to providing a cohesive account of the

brain, and can perform neural forecasting tasks [106] or cross-region prediction [107]. Encoding models predict neural responses to sensory stimuli, typically pairing a shared model of vision or language with subject- or neuron-specific readouts [108–111]. Such models are increasingly used as in-silico experimental surrogates, that allow generation of stimuli likely to induce targeted neural responses or behavior. These surrogate models accelerate *in vivo* experiments by reducing their role to validation [112]. Finally, there are foundation models focused on BCI-style decoding. Intracortical models pool motor recordings across sessions and subjects, while speech systems have reached high performance by accumulating many labelled hours within a single participant and leaning on a strong downstream language prior [17, 113, 59, 27]. Implanted microelectrodes are a relatively niche domain, and there are larger-scale efforts to build foundation models for EEG [101, 102, 114], and even sEEG [100, 105]. The performance gains promised by foundation models have even motivated BCI decoding in nontraditional modalities such as fMRI [115–117].

Chapter 3

Neural Data Transformer 2: Multi-context Pretraining for Neural Spiking Activity

This chapter is reproduced from "Neural Data Transformer 2: Multi-context Pretraining for Neural Spiking Activity" (Advances in Neural Information Processing Systems, 2023), by Joel Ye, Jennifer L. Collinger, Leila Wehbe, and Robert Gaunt.

3.1 Introduction

Intracortical neural spiking activity contains rich statistical structure reflecting the processing it subserves. For example, motor cortical activity during reaching is characterized with low-dimensional dynamical models [118, 119], and these models can predict behavior under external perturbation and provides an interpretive lens for motor learning [120–122]. However, new models are currently prepared for each experimental context, meaning separate datasets are collected for each cortical phenomena in each subject, for each session. Meanwhile, spiking activity structure is at least somewhat stable across these contexts; for example, dominant principal components (PCs) of neural activity can remain stable across sessions, subjects, and behavioral tasks [123–126]. This structure persists in spite of turnover in recorded neurons, physiological changes in the subject, or task changes required by the experiment [60, 46]. Conserved neural population structure suggests the opportunity for models that span beyond

single experimental contexts, enabling more efficient, potent analysis and application.

In this work we focus on one primary use case: neuroprosthetics powered by intracortical brain computer interfaces (iBCIs). With electrical recordings of just dozens to hundreds of channels of neuronal population spiking activity, today's iBCIs can relate this observed neural activity to behavioral intent, achieving impressive milestones such as high-speed speech decoding [17] and high degree-of-freedom control of robotic arms [36]. Even so, these iBCIs currently require arduous supervised calibration in which neural activity on that day is mapped to behavioral intent. At best, cutting-edge decoders have included training data from across several days, producing thousands of trials, which is still modest by deep learning standards [17]. Single-session models still dominate the Neural Latents Benchmarks (NLB), a primary representation learning benchmark for spiking activity [127]. Thus, despite the scientifically observed conserved manifold structure, there has been little adoption of neural population models that can productively aggregate data from broader contexts.

One possible path forward is deep learning's seemingly robust recipe for leveraging heterogeneous data across domains: a generic model backbone (e.g. a Transformer [128]), unsupervised pretraining over broad data, and lightweight adaptation for a target context (e.g. through fine-tuning). The iBCI community has set the stage for this effort, for example with iBCI dataset releases (Section A.1.1) and Neural Data Transformer (NDT) [88], which shows Transformers, when prepared with masked autoencoding, model single-session spiking activity well. We hereafter refer to NDT as NDT1. Building on this momentum, we report that Transformer pretraining can apply to motor cortical neural spiking activity from iBCIs, and allows productive aggregation of data across contexts.

Contributions: Here, we contribute NDT2, a Transformer that pretrains over broad data sources of motor cortical spiking activity. NDT2 modifies NDT1 to improve scaling across heterogeneous contexts in three ways (Section 3.3): spatiotemporal attention, learned context embeddings, and asymmetric encode-decode [129]. We find positive transfer with data from different data sessions, subjects, and tasks, and quantify their relative value. Once pretrained, NDT2 can be rapidly tuned in novel experimental sessions. We focus on offline evaluation on motor applications, demonstrating NDT2's value in decoding unstructured monkey reaching and human iBCI cursor intent. We also show proof-of-principle real-time cursor control using NDT2 in a human with an iBCI.

3.2 Related Work

Unsupervised neural data pretraining. Unsupervised pretraining's broad applicability is useful in neuroscience; little is common across neural datasets except the neural data itself. Remarkably, pretraining approaches across neural data modalities are similar; of the 4 sampled in Table 3.1, 3 use masked autoencoding as a pretraining objective (EEG uses contrastive learning), and 3 use a Transformer backbone (ECoG uses a CNN). Still, each modality poses different challenges for pretraining; for iBCIs that record spiking activity, the primary challenge is data instability [15]. The high spatial resolution of iBCI microelectrode arrays enables recording of individual neurons and provides the signal needed for high-performance rehabilitation applications, but this fine resolution also causes high sensitivity to shifts in recording conditions [46, 15]. iBCIs typically require recalibration within hours, relative to ECoG-BCIs that may not require recalibration for days [30]. At the macroscopic end, EEG and fMRI can mostly address inter-measurement misalignment through preprocessing (e.g. registration to an atlas).

Table 3.1. Neural data pretraining. NDT2, like contemporary neural data models, are intermediate pretraining efforts, comparable to early-modern large language models like BERT [130]. Neural models vary greatly in task quality and data encoding; implanted devices (Microelectrodes, ECoG, SEEG) severely restrict the subject count available (especially with public data). Volume is estimated as full dataset size / model input size. ECoG = Electrocorticography' SEEG = Stereo electroencephalography; EEG = Electroencephalography; fMRI = Functional magnetic resonance imaging.

Modality	Task	Estimated Pretraining Volume	Subjects
Microelectrodes (ours)	Motor reaching	0.25M trials	∼12
ECoG: LFP [131]	Naturalistic behavior	0.04M trials / 108 days [132]	12
SEEG: LFP [100]	Movie Viewing	3.2M trials / 4.5K electrode-hrs	10
fMRI [103]	Varied (34 datasets)	1.8M trials (12K scans)	1.7K
EEG [99]	Clinical assessment	0.5M trials / (26K runs [133])	11K
BERT [130]	Natural Language	1M 'trials' (3.3B tokens)	-

Data aggregation for iBCI. Data aggregation for iBCI has largely been limited to multi-session aggregation, where a session refers to an experimental period lasting up to several hours. These efforts often combine data through a method called stitching [134]. For context, the extracellular spiking signals recorded on microelectrode arrays are

sometimes "sorted" by the shape of their electrical waveforms, with different shapes attributed to spikes from distinct putative neurons. Sorted data has inconsistent dimensions across sessions, but as mentioned, activity across sessions has been observed to share consistent subspace structure, as e.g. identified by PCA [124]. Stitching aims to extract this stable subspace (and also resolve neuron count differences) by learning readin and readout layers per session. Stitching has been applied to BCI and broader neuroscience applications over half a year [87, 135, 136, 46, 137]. However, even linear layers can add thousands of parameters, which risks overfitting in clinical iBCI data that comprise only dozens of trials.

Alternatively, many iBCI systems simply forgo spike sorting, which appears to have a minor performance impact [60, 15]. Then, multi-session data have consistent dimensions and can feed directly into a single model [60, 138, 139] (even if the units recorded in those dimensions shift [51]). Note that these referenced models also typically incorporate augmentation strategies centered around channel gain modulation, noising, or dropout, emphasizing robustness as a design goal.

Relative to existing efforts, NDT2 aims to aggregate a larger volume of data by considering other subjects and motor tasks (Fig. 3.1B). The fundamental tension we empirically study is whether the increased volume of pretraining is helpful despite the decreased relevance of heterogeneous data for any particular target context.

Domain-adaptive vs. domain-robust decoding. Given little training data, special approaches help enable BCI decoder use in new, shifted contexts. For example, decoders can be designed to be robust to hypothesized variability in recorded populations by promoting invariant representations through model or objective design [139–141]. Alternatively, decoders can be adapted to a novel context with further data collection, which is reasonable especially if only unsupervised neural data are required. Several approaches align data from novel contexts by learning an input mapping that minimizes distributional distance between novel encodings and pretraining encodings explicitly [46, 142, 143]. NDT2 also allows unsupervised adaptation, simply by fine-tuning without additional objectives.

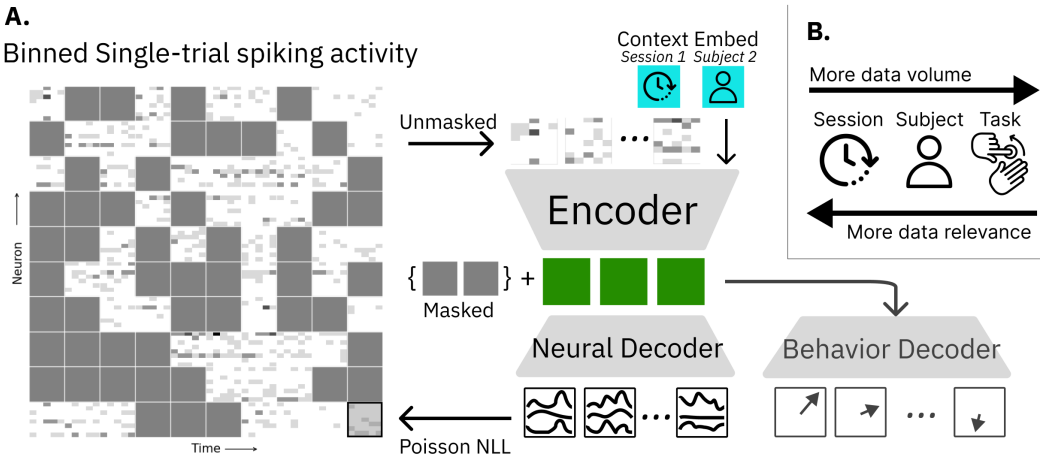

Figure 3.1. A. NDT2 is a spatiotemporal Transformer encoder-decoder in the style of He et al. [129], operating on binned spiking activity. During masked autoencoding pretraining, a spike rate decoder reconstructs masked spikes from encoder outputs; downstream, additional decoders similarly use encoder outputs for behavior prediction (e.g. cursor movement). The encoder and decoders both receive context embeddings as inputs. These embeddings are learned vectors for each unique type of metadata, such as session or subject ID. **B.** NDT2 aims to enable pretraining over diverse forms of related neural data. Neural data from other sessions within a single subject are the most relevant but limited in volume, and may be complemented by broader sources of data.

3.3 Approach

3.3.1 Designing Transformers for unsupervised scaling on neural data

Transformers prepared with masked autoencoding (MAE) [130] are a competitive model for representation learning on spiking neural activity in single contexts, as measured by their performance on the NLB [127]. Cross-domain Transformer investment has also produced vital infrastructure for scaling to large datasets. Thus, we retain Transformer MAE [129], as the unsupervised pretraining recipe. This architecture is reviewed in Fig. 3.1A.

The primary design choice is then data tokenization, i.e. how to decompose the full iBCI spatiotemporal spiking activity into units that NDT2 will compute and predict over. Traditionally, the few hundred neurons in motor populations have been analyzed directly in terms of their population-level temporal dynamics [118]. NDT1 [88] follows this heritage and directly embeds the full population, with one token per timestep. Yet across contexts, the meaning of individual neurons may change, so operations to learn

spatial representations may provide benefits. For example, Le and Shlizerman [144] and Liu et al. [141] add spatial attention to NDT's temporal attention. Yet separate space-time attention can impair performance [145] and requires padding in both space and time when training over heterogeneous data. Still, the other extreme of full neuron-wise spatial attention has prohibitive computational cost. We compromise and group K neurons to a token (padding if needed), akin to pixel patches in ViTs [146]. Neural "patches" are embedded by concatenating projections of the spike counts in the patch. In pilot experiments, we find comparable efficiencies between this patching strategy and factorized attention. We opt for full attention design to easily adopt the asymmetric encoder-decoder proposed in [129]. This architecture first encodes unmasked tokens and uses these encodings to then decode masked tokens. This 2-step approach provides memory savings over the canonical uniform encoder architecture, as e.g. popularized in the language encoder, BERT [130].

We next consider token resolutions. In time, iBCI applications benefit from control rates of 50-100Hz [147]; we adopt 50Hz (20ms bins) and consider temporal context up to 2.5s (i.e. 125 tokens in time). Given a total context budget of 2K tokens (limited by GPU memory), this leaves a few dozen tokens for spatial processing. It is currently common to record from 100-200 electrodes with an interelectrode spacing of 400µm (Blackrock Utah Arrays) [135, 16], so our budget forces 16-32 channels per token. We note that future devices are likely to record from thousands of channels at a time with much higher spatial densities [148], which will warrant new spatial strategies.

NDT2 also receives learned context tokens. These tokens use known metadata to allow cheap specialization, analogous to prompt tuning [149] from language models or environment embeddings [150] from robotics. Specifically, we provide tokens reflecting session, subject, and task IDs.

3.3.2 Datasets

We pretrain models over an aggregation of datasets; the relative volume of different datasets are compared in Fig. 3.2A, with details in Section A.1.1. All datasets contains single-unit (sorted) or multi-unit (unsorted) spiking activity recorded from either monkey or human primary motor cortex (M1) during motor tasks. In particular, we focus evaluation on a publicly available monkey dataset, where the subjects performed self-paced reaching to random targets generated on a 2D screen (Random Target

Task, RTT) [151], and unpublished human clinical BCI datasets. RTT data contains both sorted and unsorted activity from two monkeys over 47 sessions (∼20K seconds per monkey). RTT is ideal for assessing data scaling as it features long, near-hourly sessions of a challenging task where decoding accuracy increases with data [45]. For comparison, in the Maze NLB task, which uses cued preparation and movement periods, decoding performance saturates by 500 trials [127]. Since RTT is continuous, we split each session into 1s trials in keeping with NLB [127].

We also study M1 activity in 3 human participants with spinal cord injury (P2-P4). These participants have limited preserved motor function but can modulate neural activity in M1 using attempted movements of their own limbs. This activity can be decoded to control high degree-of-freedom behavior [36]; we restrict our study to 2D cursor control tasks to be most analogous to RTT, which also restricts targets to a 2D workspace. All experiments conducted with humans were performed under an approved Investigational Device Exemption from the FDA and were approved by the Institutional Review Board at the University of Pittsburgh. The clinical trial is registered at clinicaltrials.gov (ID: NCT01894802) and informed consent was obtained before any experimental procedures were conducted. Details on the implants and clinical trial are described in [16, 36], and similar task data are described in [152].

3.4 Results

We demonstrate the three requirements of a pretrained spiking neural data model for a BCI: a multi-context capable architecture (Section 3.4.1), beneficial scaled pretraining (Section 3.4.2) and practical deployment (Section 3.4.3, Section 3.4.4). We discuss the impact of context tokens in Section A.1.4.

Model preparation and evaluation. Most experiments used a 6-layer encoder (∼3M parameters). NDT2 adds a 2-layer decoder (0.7M parameters) over NDT1 [88]; we ran controls to ensure this extra capacity does not confer benefits to comparison models. To ensure that our models were not bottlenecked by compute or capacity in scaling (Section 3.4.2), models were trained to convergence with early stopping and progressively larger models were trained until no return was observed. We pretrain with causal attention where tokens cannot attend to future timesteps, as would be the case in realtime iBCI use (though bidirectional attention improves modeling). We pretrain with 50% masking and dropout of 0.1. Further hyperparameters were not

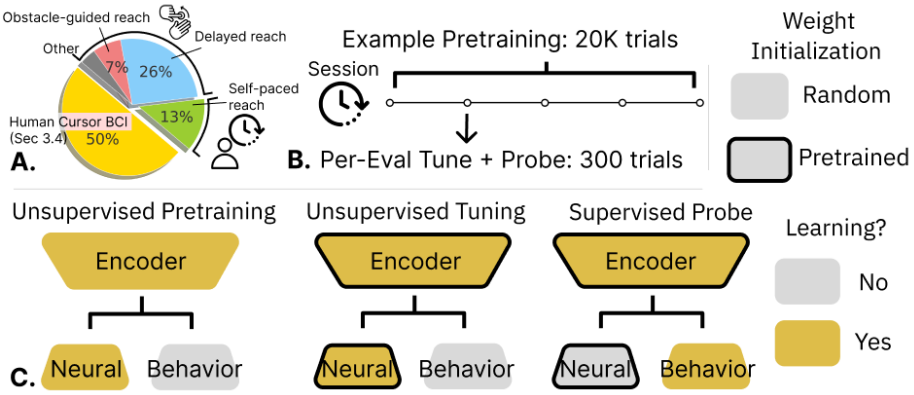

Figure 3.2. Model Training: A. We model neural activity from human and monkey reach. In monkey models, evaluation sessions are drawn from a self-paced reaching dataset [151]; multi-session and multi-subject models pretrain with other sessions in these data. The multi-task model pretrains with the other monkey reach data. Human models use a similar volume of data. **B.** A multi-session model pretrains with data from an evaluation subject, with held-out evaluation sessions. Then, for each evaluation session, we first do unsupervised tuning off the pretrained model, and then train a supervised probe off of this tuned model. **C.** We show which model components are learned (receive gradients) during pretraining and the two tuning stages. For example, supervised probes use an encoder that received both pretraining and tuning on a target session. All tuning is end to end.

swept in general experiments; initial settings were manually tuned in pilot experiments and verified to be competitive against hyperparameter sweeps. Further training details are given in Section A.1.3. We briefly compare against prior reported results, but to our knowledge there is no other work that attempts similar pretraining, so we primarily compare performance within NDT-family design choices.

Model preparation and evaluation follows several steps, as summarized in Fig. 3.2B and C. We evaluate models on randomly drawn held-out test data from select "target" sessions (selection is later detailed per experiment). Separate models are prepared for each target session, either through fine-tuning or by training from scratch for single-context models. For unsupervised evaluation, we use the Poisson negative log-likelihood (NLL) objective, which measures reconstruction performance of randomly masked bins of test trials (Fig. 3.2C middle). For supervised evaluation, we report the R^2 of decoded kinematics, which for these experiments are 2D velocity of the reaching effector (Fig. 3.2C right). These supervised models are separately tuned off the per-evaluation session unsupervised model; so that all supervised scores receive the benefit of unsupervised modeling of target data [153].

3.4.1 NDT2 enables multicontext pretraining

We evaluate on five temporally spaced evaluation sessions of monkey Indy in the RTT dataset, with both sorted and unsorted processing. Both versions are important; sorted datasets contain information about spike identity and are broadly used in neuroscientific analysis, while unsorted datasets are frequently more practical in BCI applications. Velocity decoding is done by tuning all models further with a 2-layer Transformer probe (matching the reconstruction decoder). Here we provide the models with 5 minutes of data (300 training trials) for *each evaluation session*. This quantity is a good litmus test for transfer as it is sufficient to fit reasonable single-session models but remains practical for calibration. A 10% test split is used in each evaluation session (this small % is due to several sessions not containing much more than 300 trials). We pretrain models using approximately 20K trials of data, either with the remaining non-evaluation sessions of monkey Indy (Multi-Session), the sessions from the other monkey (Multi-Subject), or from non-RTT datasets entirely (Multi-Task, see Fig. 3.2).

Prior work in multi-session aggregation either use stitching layers or directly train on multi-day data with consistent unit count. Thus, we use NDT1 with stitching as a baseline for sorted data, and with or without stitching for unsorted data. NDT2 pads observed neurons in any dataset to the nearest patch multiple. All models receive identical context tokens.

We show the performance of these pretrained models for sorted and unsorted data in Fig. 3.3. For context, we show single-session performance achieved by NDT1 and NDT2, and the decoding performance of the nonlinear rEFH model released with the dataset [45]. rEFH scores were estimated by linearly interpolating the 16ms and 32ms scores in Makin et al. [45]. [1] Single session performance for NDT1 and NDT2 is below this baseline. However, consistent with previous findings on the advantage of spatial modeling [144], we find single-session NDT2 provides some NLL gain over NDT1. Underperforming this established baseline is not too unexpected: NDT's performance can vary widely depending on the extent of tuning (Transformers span a wide performance range on the NLB, see also Section A.1.3). Part of the value of pretraining is that it greatly simplifies the tuning needed for model preparation [154].

[1] The rEFH model data splits vary slightly from ours: its data splits are sequential and contiguous in time, whereas we use random draws in keeping with NLB. NDT2 sorted decoding drops from 0.54 to 0.52 when mirroring Makin et al's data splits.

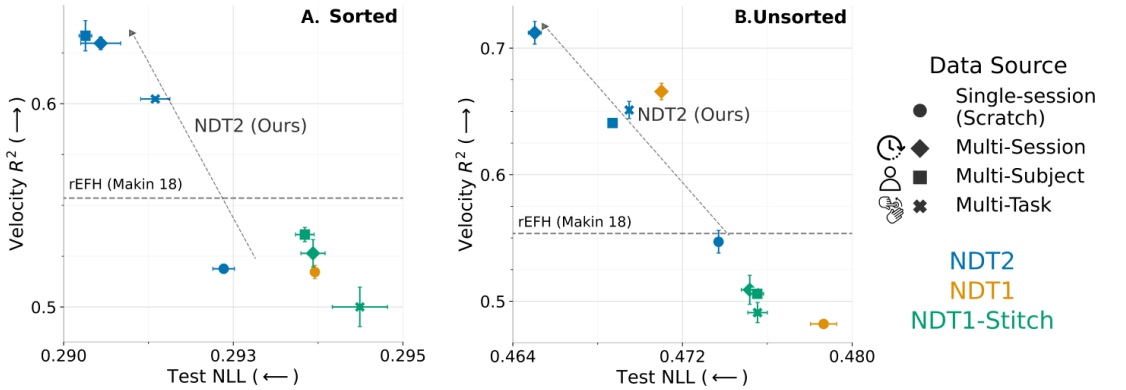

Figure 3.3. NDT2 enables pretraining over multi-session, multi-subject, and multi-task data. We show unsupervised and supervised performance (mean of 5 sessions, SEM intervals of 3 seeds) on sorted **(A)** and unsorted **(B)** spiking activity. Higher is better for R^2, lower is better for negative log-likelihood (NLL). Pretraining data is size-matched at 20Ks, except scratch single-session data. NDT2 improves with pretraining with all data sources, whereas stitching is ineffective. NDT1 aggregation is helpful but does not apply beyond session transfer. A reference well-tuned decoding score from the rEFH model is estimated [45].

However, in sorted data shown in Fig. 3.3A, all pretrained NDT2 models outperform rEFH and single-session baselines, both in NLL and kinematic decoding. Surprisingly, multi-subject data work as well as multi-session data, and multi-task data provide an appreciable improvement as well. NDT-Stitch performs much worse in all cases, and in fact, cross-task pretraining brings NDT-Stitch below the single-session baseline. We expect that stitching is less useful here than in other works because other works initialize their stitching layers by exploiting stereotyped task structure across settings (see PCR in [87]), and we cannot do this because RTT does not have this structure.

The unsorted data has consistent physical meaning (electrode location) across datasets within a subject, which may particularly aid cross-session transfer. Indeed, unsorted cross-session pretraining achieves the best decoding ($> 0.7R^2$) in these experiments (Fig. 3.3B, blue diamond). The consistent dimensionality should not affect cross-task and cross-subject decoding, yet they also improve vs their sorted analogs, indicating the unsorted format benefits decoding in this dataset. Given this, we use unsorted formats in subsequent analysis of RTT. Otherwise, relative trends are consistent with the sorted case. Both analyses indicate that different pretraining distributions all provide some benefit for modeling a new target context, but also that not all types of pretraining data are equivalently useful.

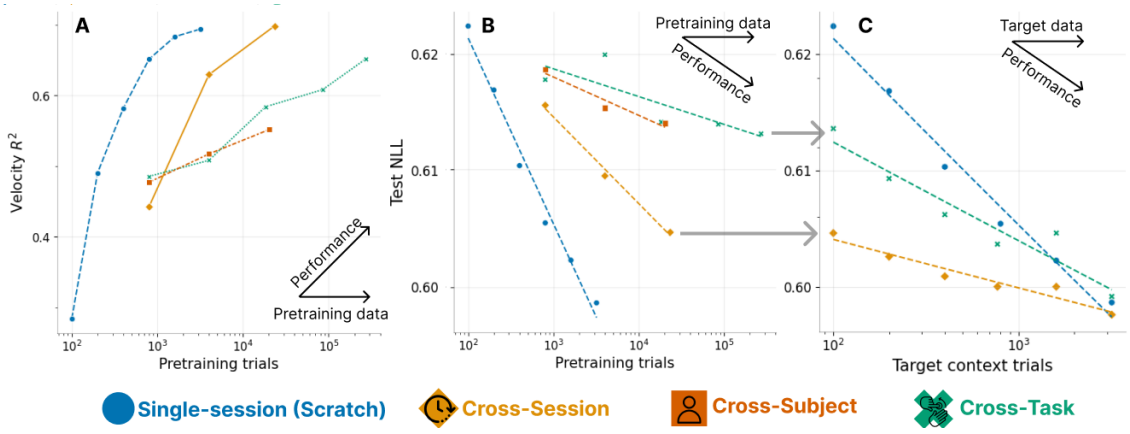

Figure 3.4. Scaling of transfer on RTT. We compare supervised R^2 (**A**) and unsupervised NLL scaling (**B**) as we increase the pretraining dataset size. Each point is a model; non single-session models calibrate to evaluation sessions with 100 trials. All pretraining improves over the leftmost 100-trial single-session from-scratch model, though scaling benefits vary by data sources. **C**. We seek a convergence point between pretraining and training from scratch, as we increase the number of trials we use in our target context. Models converge by 3K trials.

3.4.2 NDT2 scaling across contexts

Naturally, cross-session data are likely the best type of pretraining data for its close relevance, but less relevant data can also occur in much greater volumes. To inform future pretraining data efforts, we perform three analyses to coarsely estimate the data affinity [155] of the three different context classes (cross-session, cross-subject, and cross-task). Previously these relationships have been grounded in shared linear subspaces [123–125]; we now quantify this in the more general generative model encompassed by DNN performance scaling.

Scaling pretraining size. We consider transfer as we scale pretraining size, so as to extrapolate trends that might forecast the utility of progressively larger data. Specifically, we measure performance after tuning varied pretrained models with 100 trials of calibration in a novel context. We do this for both supervised (Velocity R^2) and unsupervised (NLL) metrics in Fig. 3.4A and B respectively. To contextualize performance of cross-context scaling, we measure *in-distribution* scaling of intra-session data (Scratch). For example, the largest cross-session model tuned with 100 trials achieves a NLL similar to 1K trials of intra-session data (Fig. 3.4B, Cross-session vs Scratch). This shows that cross-session data can capture a practically long tail of

single-session neural variance (experiments rarely exceed 1K trials), with nonsaturated benefits in pretraining dataset size. Alternately, cross-session pretraining allows decoding performance similar to the largest single-session model performance (Fig. 3.4A, Cross-session vs Scratch), but this scaling is starting to saturate.

Shallower slopes for cross-subject and cross-task pretraining on both metrics indicate poorer transfer to evaluation data. In the unsupervised case (Fig. 3.4B), cross-subject and cross-task transfer never exceed the NLL achieved with 400 single-session trials. Note that our task scaling may be pessimistic as we mix human data (Table 3.2) with monkey data to prepare the largest model, but the trend before this point is still shallow. However, these limitations do not clearly translate to the supervised metric (note that discrepant scaling metrics are also seen in language models [93]). For example, the decode R^2 achieved by the largest task-pretrained model compares with in-session models at 800 trials; the same model's NLL is most comparable that of a 300 trial single-session model.

Convergence point with from-scratch models. It remains unclear how rapidly pretraining benefits decrease as we increase target context data. We thus study the returns from pretraining as we vary target context calibration sizes [156] (Fig. 3.4C). Both models yield returns up to 3K trials, which represents about 50m of data collection in the monkey datasets, and coincidentally is the size of the largest dataset in [151]. Session pretraining provides larger gains, but task pretraining e.g. also respectably halves the data needed to achieve 0.61 NLL. This indicates pretraining is complementary to scaling target session collection efforts. This need not have been the case: even Fig. 3.4B suggests that task transfer by itself is ineffective at modeling the long tail of neural variance. Note that returns on supervised evaluation are likely similar or better based on Fig. 3.4A/B; we explore a related idea in Section 3.4.3.

Overall, the returns on using pretrained BCI models depends on the use case. If we are interested in best explaining neural variance, pretraining alone underperforms a moderately large in-day data collection effort (single-session model achieves lowest NLL in Fig. 3.4B). However, we do not see *interference* [156] in our experiments, where pretraining then tuning underperforms a from-scratch model. Thus, so long as we can afford the compute, broad pretraining is advantageous; we show these trends are repeated for two other evaluation sessions in Section A.1.6. We reiterate that our pretraining effort is modestly scaled; the largest pretraining only has 2 orders more data than the largest intra-context models. These conclusions may further strengthen

insofar if we are able to better scale curation of pretraining data over individual experimental sessions.

3.4.3 Using NDT2 for improved decoding on novel days

RTT Decoding. A continuously used BCI presents the optimistic setting of having both broad unsupervised data but also multiple sessions worth of supervision for our decoder. To evaluate NDT2 in this case, we follow 1st stage unsupervised pretraining with a 2nd stage of supervised pretraining of a decoder, and finally measure the decoding performance in a novel target session in Fig. 3.5A. We find that given *either* supervised or unsupervised tuning in our target session, beyond smoothly improving over the 0-Shot Pretrained model's performance, achieves decoding performance on par with the best from-scratch models at all data volumes. This is true both in the realistic case where the majority of target-session data are unlabeled (100 Supervised Trials) and in the optimistic case when >1K trials of supervised data are available (Scratch). As expected, though, gains against the Scratch models are largest when target session data are limited. In sum, pretraining and fine-tuning enables practical calibration without explicit domain adaptation designs (as explored e.g. in [46, 142, 143]).

Human BCI evaluation. We next evaluate NDT2 in offline decoding of human motor intent, i.e. open loop trials of 2D cursor control. This shift from actual reach in monkeys is challenging: human sessions often contain few trials (e.g. 40) and intent labels are much noisier than movement recordings (intent labeling is described in Section A.1.1). We now evaluate a temporally contiguous experimental block, but only tune one model over this block, rather than per session, due to the high session count. We also increase test split to 50% to decrease noise from evaluating only a few trials.

We also compare broad pretraining (multi-session, subject, and task) performance (Table 3.2, row 1) with various data ablations. Ablating cross-subject data only has a minor performance impact (row 1 vs 2), while ablating cross-task data, which here indicates removal of data collected during online control and leaving only open-loop observation trials, hurts performance more (row 3 vs 1). We note that NDT2 fails to train using only single-sessions given extremely low trial count; we use linear decoding performance as a single-session reference linear decoding performance. Overall, the plentiful cross-session data likely occludes further gains from cross-subject and cross-

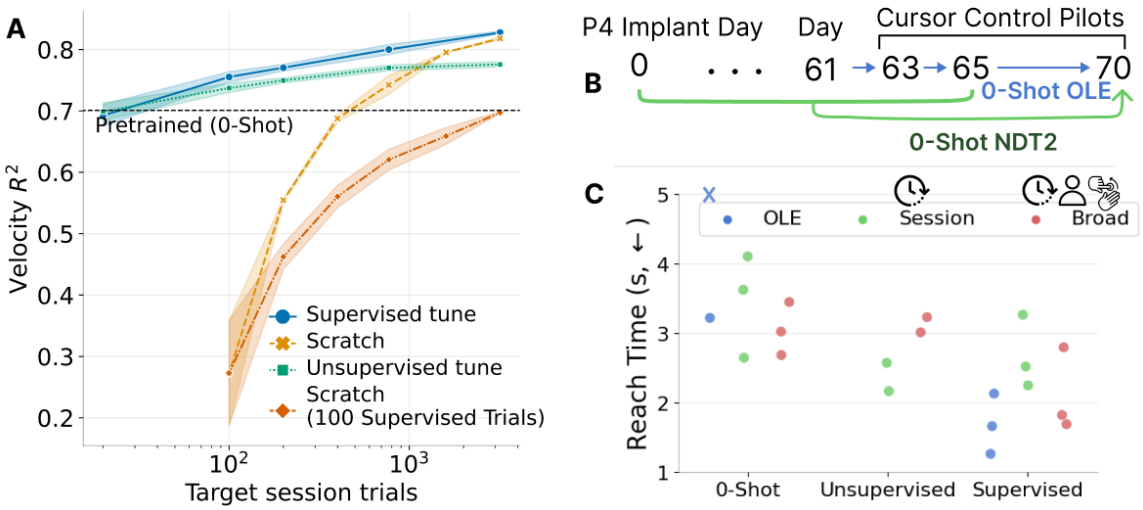

Figure 3.5. Tuning and evaluating pretrained decoders. A. In offline experiments, we compare a multisession pretrained model calibrated to a novel day against current approaches of a pretrained model without adaptation (0-Shot) and from-scratch training (yellow, orange). Both supervised and unsupervised outperforms these strategies. Standard error shown over 3 seeds. **B.** In human control pilot experiments, we evaluated models on 3 test days. "0-Shot" models use no data from test days. **C.** Average reach times in 40 center-out trials is shown for 3 decoders over 2-3 sessions. This average includes trials where target is not acquired in 10s, though this occurs 1-2 times in 40 trials. Session NDT2 uses <250 trials of data, while Broad NDT2 also includes other human and monkey data. Pretrained models provide consistent 0-Shot control while OLE [100] can sometimes fail (shown by X). Control improves with either unsupervised or supervised tuning. However, OLE appears to overtake NDT2 with supervision.

task pretraining, but note that there is no negative transfer.

Next, given reports of monkey to human transfer [157], we assess whether monkey data in either pretraining or decoder preparation improves decoding (Table 3.2, rows 4-7). We find that monkey data, however incorporated, reduces offline decoding performance (row 4-7 < 1). This cross-species result is the first instance of possible harm from broader pretraining, but warrants further study given the potential of transferring able-bodied monkey decoders to humans.

3.4.4 NDT2 for human cursor control pilots.

Finally, to assess NDT2's potential for deployed BCIs, we run realtime, closed-loop cursor control with one person, P4. This person was implanted recently ($\sim$ 2 months prior), and has high signal quality but limited historical data. We compare two NDT2

Table 3.2. Human reach intent decoding. We analyze how pretraining data impacts offline decoding in two people, P2 and P3. Checks (✓) indicate the data used in pretraining beyond task- and subject-specific data; human data totals 100K trials for P2 and 30K trials for P3. Intervals are SEM over 3 fine-tuning seeds. Pretraining transfers across task and somewhat across subject, but there is *no* benefit from monkey data.

	Neural data (Unsup. pretrain)			Behavior (Sup. pretrain)	Velocity R^2 (↑)	
	Cross-Subject	Cross-Task	+130K Monkey	+24K RTT (Monkey)	P2	P3
1)	✓	✓			$0.503_{\pm 0.020}$	$0.515_{\pm 0.008}$
2)		✓			$0.487_{\pm 0.007}$	$0.509_{\pm 0.016}$
3)	✓				$0.444_{\pm 0.007}$	$0.493_{\pm 0.002}$
4)	✓	✓	✓	✓	$0.486_{\pm 0.012}$	$0.472_{\pm 0.019}$
5)	✓	✓	✓		$0.490_{\pm 0.007}$	$0.477_{\pm 0.018}$
6)	✓	✓		✓	$0.474_{\pm 0.009}$	$0.491_{\pm 0.010}$
7)				✓	$0.443_{\pm 0.005}$	$0.455_{\pm 0.013}$
8)	Smoothed spike ridge regression (OLE)				0.077	0.208

models, one of which pretrains with 10 minutes of participant-specific data, and one broadly pretrained on all human and monkey data, along with a baseline linear decoder (indirect OLE [158]). We evaluate the setting where decoders from recent days are available, and can either be used directly (0-shot) or updated with calibration data on the test day (Fig. 3.5B). All models can be supervised with test day data, but NDT2 models can also use unsupervised tuning with only neural data. In a standard center-out BCI-reaching task (methods in Section A.1.2), NDT2 allows consistent 0-shot use, whereas OLE sometimes fails (Fig. 3.5C, blue X). After either form of tuning, both NDT2 models (multisession, broad) improve. Importantly, NDT2 tunes without any additional distribution-alignment priors, as in [46, 140, 139], showing that pretraining and fine-tuning [130] may be a viable paradigm for closed loop motor BCIs.

Perhaps surprisingly, OLE provides the best control given supervised data. The performance gap and large distribution shift from offline analysis to online control is well known [15], though the specific challenge of DNN control only has basic characterization. For example, NDT2 decodes "pulsar" behavior as in [136]. Costello et al. [89] provides a possible diagnosis: these pulses reflect the ballistic reaches of the open loop training data, whereas OLE, due to its limited expressivity, will always provide continuous (but less stable) control that can be helpful in time-based metrics.

Promising approaches to mitigate the pulsar failure mode include further closed-loop tuning [159] or open-loop augmentation [136]; we leave continued study of NDT2 control for future work.

3.5 Discussion

NDT2 demonstrates that broad pretraining can improve models of neural spiking activity in the motor cortex. With simple changes to the broadly used masked autoencoder Transformer, NDT2 at once spans the different distribution shifts faced in spiking data. We find distinct scaling slopes for different context classes, as opposed to a constant offset in effectiveness [92]. For assistive applications of BCI, NDT2's simple recipe for multisession aggregation is promising even if the ideal scenario of cross-species transfer remains unclear. More broadly, we conclude that pretraining, even at a modest 10-100K trials, is useful in realistic deployment scenarios with varied levels of supervised data.

Supervising pretrained BCI decoders. Motor BCIs fundamentally bridge two modalities: brain data and behavior. NDT2 allows neural data pretraining, but leaves open the challenge of decoding of the full range of motor behavior. Without mapping that range, BCIs based on neural data pretraining alone will need continuous supervision. This is still a practical path forward: beyond explicit calibration phases, strategies for supervising BCIs are diverse, and can derive from user feedback [160], neural error signals [161, 162], or task-based estimation of user intent [61, 159, 84]. The ambition of pretrained *BCI* models, with broad paired coverage of the neural-behavior domain will be challenging given the experimental costs of data collection; the field of robotics suggests that scaled offline or simulated learning are important strategies given this expense. Since we lack convincing closed-loop neural data simulators (though see [163, 61]), understanding how to leverage *behavior* from existing heterogeneous datasets is an important next step.

Negative NLB result. NDT2 performance did not exceed current NLB SoTA on motor datasets (RTT, Maze) [164]. We first note the NLB evaluation emphasizes neural reconstruction of held-out neurons, which differs from our primary emphasis on decoding. Second, our unsupervised scaling analysis indicates that modest pretraining (100K trials) provides limited gains for neural reconstruction, especially when the target dataset has many trials (Fig. 3.4C), as in the NLB RTT dataset, so underperformance

does not contradict our results. Revisiting the NLB after further scaling may be fruitful.

Limitations. NDT2's design space is under-explored. For example, we do not claim that full spacetime attention is necessary over factorization. While NDT2 achieves positive transfer, further gains may come from precise mapping of context relationships [155]. Further, it is difficult to extrapolate the benefits of scaling beyond what was explored here, particularly with gains in unsupervised reconstruction appearing very limited. Our evaluation also has a limited scope, emphasizing reach-like behaviors. While these behaviors are more general than previous demonstrations of context transfer [46, 143, 139, 140], evaluating more complex behavior decoding is a practical priority. Further, NDT2 benefits are modest in human cursor control pilots, reiterating the broadly documented challenge in translating gains in offline analyses to online, human control [136, 15]. Finally, design parameters such as masking ratio may affect scaling trends, which we cannot assess due to compute limits.

Broader Impacts. Pretrained iBCI models may great improve iBCI usability. However, DNNs may require further safeguards to ensure that decoded behaviors, especially in real-time control scenarios, operate within reasonable safety parameters. Also, pretraining will require data from many different sources, but the landscape around human neural data privacy is still developing. While there are very few humans involved in these experiments, true deidentification remains difficult, requiring, at a minimum, consented data releases.

Chapter 4

Few-shot Algorithms for Consistent Neural Decoding (FALCON) Benchmark

This chapter is reproduced from "Few-shot Algorithms for Consistent Neural Decoding (FALCON) Benchmark" (Advances in Neural Information Processing Systems, Datasets and Benchmarks Track, 2024), by Brianna M. Karpowicz, Joel Ye, Chaofei Fan, Pablo Tostado-Marcos, and colleagues. B.M.K. and the candidate contributed equally.

4.1 Introduction

Brain-computer interfaces (BCIs) provide a path to restore movement and communication in individuals with paralysis by decoding neuronal population activity to uncover the user's intention. BCIs have recently achieved many promising demonstrations, including high degree of freedom robot arm control [16, 40], computer use and communication [19, 165, 135, 166–168], and speech decoding [113, 17, 59, 169, 170]. A specific class of BCIs known as intracortical BCIs (iBCIs) have enabled many of these impressive technological feats. However, many of these demonstrations have required decoders to be recalibrated daily or even more frequently, interrupting device use and burdening the user. Real-world iBCI deployment will require maintaining high performance over long time periods with minimal recalibration. The challenge here stems from nonstationarities in the neural data that are caused by many factors

acting at multiple timescales, such as shifts in the position of the electrode relative to surrounding tissue, changes in tissue properties in response to electrode implantation, electrode malfunction, or neural plasticity [51, 52]. These nonstationarities result in a changing relationship between neural data and behavior, necessitating frequent decoder recalibration to maintain high performance.

Fortunately, despite these nonstationarities, there are many potential ways to leverage structure in neural or behavioral data to help reduce the burden of recalibration [53]. For example, while the spiking activity recorded on an individual iBCI electrode can change over timescales of hours, neural population activity contains low dimensional structure (manifolds) that shows a consistent relationship with behavior over months to years [87, 171, 124]. Realignment methods that exploit these conserved manifolds [172, 173, 46, 143, 174, 139, 175]can restore decoding without explicit calibration periods. Alternatively, rather than focusing on structure intrinsic to the neural data, another set of approaches attempts to achieve robustness by continually recalibrating decoders using the retrospective analysis of data collected during the subject's normal use of the iBCI [176–178]. A third strategy has centered on supervised deep network training using many sessions to yield decoders that are robust to session-to-session variability [60, 25], which may be extended further to potentially yield universal iBCI decoders that generalize to new subjects or tasks [179, 180, 137]. These diverse and potentially complementary efforts converge around a single problem statement: the real-world iBCI decoding challenge is to maintain high performance on distinct, but related, data distributions, with minimal data from the new setting.

While these diverse approaches have thus far used their own ad-hoc evaluation, standardization could enable rigorous comparison to assess real-world potential and highlight advances upon which future efforts can build. We propose the FALCON benchmark, **F**ew-Shot **Al**gorithms for **Co**nsistent **N**eural Decoding, as a common evaluation for stable, long-term decoding performance. FALCON releases 5 multi-session datasets that span movement and communication tasks relevant to iBCIs: human and monkey reach and grasp behavior (`H1`, `M1`), monkey finger movement (`M2`), human handwriting (`H2`), and birdsong (`B1`). These datasets are divided into held-in and held-out sessions. To evaluate how well few-shot decoders advance iBCI robustness given real-world data constraints, only a small amount of supervised data is released from held-out sessions. Approaches for the more challenging settings of only using neural data on new sessions (unsupervised), or no data from new sessions

(zero-shot), can also be evaluated using the same data splits. This report describes the design of the benchmark, its datasets, and the performance of baseline models. By introducing FALCON, we aim to establish standardized evaluation practices for robust iBCI decoding approaches that can provide researchers with metrics to select methods for in-device use.

4.1.1 Related work

Benchmarks of BCI Decoding. FALCON evaluates iBCI decoding, or the prediction of intention from neural activity. To date, benchmarks of decoding have been uncommon compared to other fields using machine learning. The early BCI competition series [181] and more recent additions of the International BCI Competition [182] and MOABB [183] evaluated decoding in offline (i.e., pre-recorded) noninvasive neural datasets in multiple subjects and highlighted several of the challenges faced in iBCI datasets (multi-session transfer, removal of repeated data structure). More recently, the Brain2Text decoding benchmark [184] evaluates speech decoding in human iBCIs. However, absent a strong benchmarking culture, models across intracortical and noninvasive neural recording modalities [179, 180, 137, 185–189] are still often evaluated on different public or private datasets. This lack of standardization makes comparison across works difficult due to subjectivity in preprocessing, metric choice, and evaluation design.

Benchmarks on Neural Data. Benchmarks for neural data analysis are related but differently motivated than decoding benchmarks. BrainScore [190] evaluates the ability of models to predict brain data when trained on non-neural data tasks. The Sensorium [191] and Algonauts [192] challenges evaluate encoding models that predict brain activity of mouse visual cortex and human fMRI, respectively, given visual stimuli. The Neural Latents Benchmark (NLB) [193] evaluates latent variable models on spiking activity from different brain areas of monkeys. While the NLB has a decoding metric, this metric is computed with ridge regression on inferred latent variables and is not treated as a primary endpoint. FALCON directly evaluates decoding, allows more flexibility in decoder architecture, and more closely aligns with the goal of evaluating the quality of iBCI decoders.

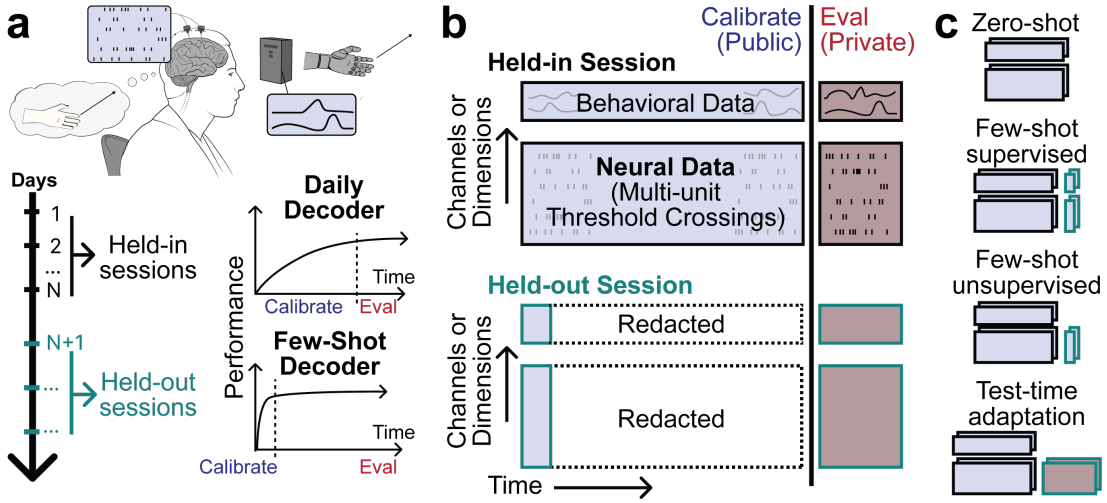

Figure 4.1. FALCON Evaluation Design. **(a)** Top: BCI decoders are prepared by collecting calibration data where a user attempts to perform a cued behavior. This process yields paired examples of behavioral outputs and associated neural data. Bottom: Current practice requires new calibration data to train new decoders. High decoding performance may require substantial data, motivating methods for few-shot decoding. FALCON provides full "held-in" sessions and evaluates few-shot decoding on "held-out" sessions. **(b)** Each session in a FALCON dataset contains multiunit threshold crossings and behavioral data (which are discrete sentences in `H2` and continuous covariates otherwise). Evaluation data is withheld from all sessions. All remaining data is released publicly for held-in sessions. Only a small fraction of data is released for held-out sessions. **(c)** FALCON's design enables comparison of different approaches for consistent decoding. Zero-shot methods use no data from held-out sessions, few-shot methods use the calibration splits from held-out sessions, and test-time adaptive methods can implement behavior-free, unsupervised decoder updates during evaluation.

4.2 Benchmark evaluation pipeline and metrics

4.2.1 Evaluation strategy and pipeline

FALCON evaluates behavioral decoding from iBCI neural activity in five datasets. Each dataset comprises multiple sessions of data divided into two contiguous splits: held-in and held-out (**Fig. 4.1a**). As in standard decoder calibration, held-in sessions provide sufficient data to train a high-performing decoder; held-out sessions are prepared for evaluating few-shot decoder performance and therefore include insufficient data to prepare decoders from scratch (**Fig. 4.1a,b**). All datasets provide multi-unit threshold crossings (detected voltage deflections caused by nearby neuron action potentials) recorded from intracortical electrodes and behavioral data (specified per

task). An evaluation split of the same length is withheld from both held-in and held-out. All remaining data is released for held-in sessions while a small fraction of data is released for held-out sessions. Note that the held-in split provides a standard-data regime iBCI decoding benchmark, but FALCON focuses on few-shot decoding performance in the held-out split.

Submitted decoders are executables that implement an iBCI prediction interface. The evaluation server (EvalAI [194]) requires causal, open-loop predictions to be made on streaming neural data, timestep-by-timestep. The communication datasets make predictions on coarser timescales (per sentence for `H2`, per song motif for `B1`). These formats mimic current iBCI use for their respective tasks. Lack of trial structure in movement tasks is an important training time consideration; decoders trained on trialized data can degrade significantly when evaluated continuously (Section B.1.5). We note that an important limitation of FALCON is that evaluation may be susceptible to promoting models that exploit trial structure implicit in the datasets, even if this does not benefit iBCI control [195].

4.2.2 Supported approaches and benchmark scope

FALCON allows methods with varying data assumptions to be evaluated in a common setting (**Fig. 4.1c**). Decreased data use provides greater reduction of user burden, but can be more challenging.

Zero-shot methods directly predict behavior on new sessions with fixed model parameters. This typically requires using deep networks that train on many sessions (e.g. months) of data. Such methods have enabled high performance cursor control on new days for months into the future [60, 25]. Recent efforts exploring subject generalization [196, 125, 137] and neural data foundation models [179, 180] may alleviate the burden of large scale data collection on individual users. However, as current multi-session zero-shot methods impose large data collection burdens on the user and may still degrade after long-term use, there is a practical need to explore adaptive methods as well.

Few-shot supervised methods assume the collection of limited calibration data for every session of use. For people with paralysis, a high-performance iBCI that requires a short calibration procedure before use may still confer a large advantage over other assistive technologies. Current supervised deep networks typically adapt to short

calibration blocks through fine-tuning of a pretrained model [179, 180, 137]. Few-shot supervision is the least strict setting that can be evaluated with FALCON that still reduces user burden, for which the highest performance is expected.

Few-shot unsupervised methods remove the need for behavioral data on new days, skipping explicit calibration periods by allowing recalibration procedures to be performed using only neural data from normal iBCI use. Due to their lack of reliance on behavioral data, unsupervised approaches are not subject to problems that may arise from behavioral labels, which may be difficult to obtain during iBCI use when guessing a user's intent post-hoc can be unreliable. Unsupervised methods typically assume that the neural activity has an underlying manifold which maintains a stable relationship to behavior over long periods of time [172, 173, 46, 143, 174, 139]. However, the specific context of iBCI use, such as strategy or posture, may lead to a change in the manifold-to-behavior mapping and violate this assumption. FALCON's datasets are drawn from consistent behavior across days, though due to behavioral complexity, not all behavioral conditions will be sampled in the few-shot calibration data.

Test-time adaptation generally leverages behavioral priors to provide model labels on unlabeled data. Currently proposed test-time adaptation methods avoid the collection of any calibration data on test days. Instead, these methods use neural data and inferred behavioral labels collected during normal iBCI use to perform "semi-supervised" decoder recalibration. However, these methods have only been demonstrated for two-dimensional cursor use and language communication [176–178], suggesting open challenges for broad behavioral domains, such as in FALCON's movement datasets.

4.2.3 Metrics

Each dataset uses a standard decoding metric. The movement tasks (`M1`, `M2`, `H1`) require predictions of multi-dimensional motor covariates, such as muscle activity. For these tasks, accuracy is reported using the coefficient of determination (R^2), computed as a variance-weighted average across the R^2 of individual motor covariates. R^2 is useful for interpreting low-dimensional predictions as a constant mean prediction achieves an R^2 of 0 and max R^2 is 1. The handwriting task (`H2`) requires prediction of English characters from a corpus of common sentences; we use word error rate

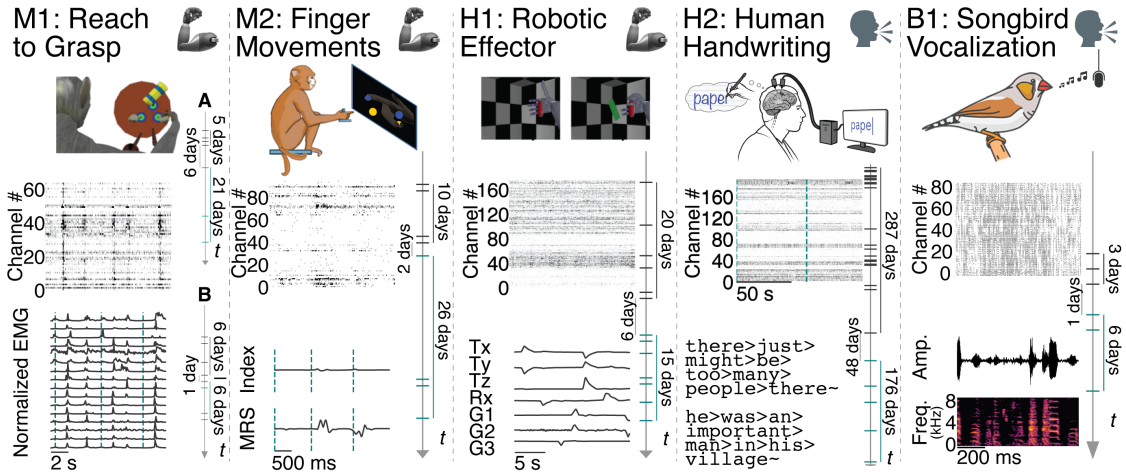

Figure 4.2. FALCON datasets span iBCI use cases. For each column, *top*: task schematic; *middle*: neural activity for all channels over time; *bottom*: example behavioral outputs. Each panel includes a vertical timeline denoting held-in (gray) and held-out (teal) sessions in the dataset. Ticks mark individual sessions, colored vertical bars indicate time elapsed within or between splits. **Tasks:** Movement datasets (mechanical arm) include: monkey reach-to-grasp (`M1`, 2 monkeys (`A`/`B`), 64/96 channels neural data, 16 channels muscle activity), monkey finger movements (`M2`, 96 channels neural data, 2-dimensional finger movements), and human robotic effector (`H1`, 172 channels neural data, 7-dimensional hand and arm velocity outputs). Communication tasks (speaking head) include human handwriting (`H2`, 192 channels neural data) and songbird vocalization (`B1`, 85 channels neural data).

(WER) as a metric, computed as the edit distance between the predicted and expected sequence divided by the length of the intended sequence. Birdsong decoding (`B1`) reports performance as mean squared error (MSE) on the predicted spectrogram; MSE is preferable for evaluating spectrogram predictions as the predictions are much higher-dimensional than in movement tasks. Metrics are computed per session, and across-session mean and standard deviation are reported on EvalAI. Mean and standard deviation are computed separately for held-in and held-out splits.

4.3 Datasets

FALCON aims to provide a comprehensive evaluation of few-shot decoding across contemporary iBCI applications. FALCON datasets span two primary groups of tasks: movement and communication (**Fig. 4.2**). FALCON's movement datasets have either kinematic or muscle outputs, and the communication datasets have either text or vocal outputs. Because most human iBCI study participants have limited independent

movement, human behavioral data are those that the researcher asked the participant to attempt or imagine, while animal behavioral data are recorded from physical actions. All datasets contain electrophysiological voltage recordings collected from intracortical microelectrodes. We extract threshold crossings from the recorded voltages to yield spiking activity, as is standard practice for iBCIs [197]. Detailed descriptions of each dataset and their locations can be found in Section B.1.3.

While the ultimate goal of some iBCI research is applications in humans, we provide animal datasets because animal models are essential to develop iBCI applications and for basic scientific discovery [198]. Using both animal and human data also improves the likelihood of finding models with broad effectiveness, as levels of instability are likely to vary across subjects and species [46, 143, 15].

`M1`: Monkey reach and grasp. The `M1` dataset consists of recordings using Floating Microelectrode Arrays (Microprobes), implanted in the precentral gyrus while two monkeys (`M1-A` and `M1-B`) reached to, grasped, and manipulated an object in a specific location (4 possible objects, 8 possible locations) [199–202]. Intramuscular electromyography (EMG) was recorded from 16 muscles in the right hand and upper extremity. The large number of object/location combinations leads to a wide variety of muscle activations. Unlike higher-level behavioral variables (such as robotic arm endpoint velocities), EMG is a directly measurable output of the motor nervous system, and thus provides a signal that should have a close correspondence to neural activity on a moment-by-moment basis. EMG is also directly relevant to iBCIs that combine with functional electrical stimulation to control paralyzed limbs [203, 204]. Monkey EMG is interesting to iBCI research as human iBCI users with paralysis have limited muscle control and likely lack the ability to produce EMG decoding targets; recent works have proposed cross-species transfer to exploit monkey EMG data for iBCI applications [196].

`M2`: Monkey finger movements. The `M2` dataset consists of Utah array recordings from the precentral gyrus while a monkey made finger movements to control a virtual hand to acquire cued target positions [205, 24]. Finger actuation ranged from full extension to full flexion with cued movements focusing on the index finger and/or the middle-ring-small (MRS) finger group. The goal of including `M2` in the FALCON benchmark is to develop methods that accurately predict individuated finger movements over time. Finger control is a critical aspect of dexterous hand function and is a key target for iBCI control that aims to restore upper limb and hand function

to individuals. Recent work has shown that the encoding of finger behaviors in motor cortex may be compositional [205, 206]; yet, the implications of this finding on iBCI control and decoding stability are unclear.

H1: Human robotic effector. The H1 dataset contains Utah array recordings from the hand and arm motor cortex of a human iBCI participant, collected in a long-term clinical study on iBCIs for sensorimotor control. The participant was cued to attempt to reach and grasp with their right hand. This data was used to calibrate an iBCI for control of a robotic arm in a 7 degree-of-freedom task [16, 36, 22]. These data are open loop, meaning that the participant attempted cued movements but was not directly controlling the output and could not correct errors in real-time. H1 contains a breadth of combinations of robotic arm command variables (3D limb kinematics, 1D rotation, 3D grasp shape) that are often decoding targets for iBCIs. High-dimensional control is particularly burdensome to calibrate, as the large number of possible endpoints demands calibration procedures that are often several minutes long [36]. Developing methods to improve the efficiency of calibration to novel sessions would advance the practical viability of using iBCIs for high-dimensional control.

H2: Human handwriting. The H2 dataset contains neural activity recorded using Utah arrays placed in the "hand knob" area of the dorsal motor cortex of a human iBCI participant, collected as part of the BrainGate2 Clinical Trial. The participant was asked to copy a sentence by attempting to write each letter individually [135]. The H2 dataset falls in the domain of brain-to-text BCIs, which aim to restore communication capabilities. Decoders for this task need to accurately predict the intended character as well as determine when that character was intended to be written, as the task is fully self-paced. This task is therefore not amenable to traditional linear decoders and will require more sophisticated approaches, most canonically RNN decoders with a Connectionist Temporal Classification loss [135, 17, 59, 178]. Additionally, due to the goal of predicting words or sentences, communication iBCIs often use large language models to further refine predictions or build stable decoders [17, 178].

B1: Songbird vocalization. The B1 dataset features neural recordings from a zebra finch songbird using Neuropixels 1.0 probes [207] implanted in the motor brain region robust nucleus of the arcopallium (RA). Alongside neural activity, this dataset includes simultaneous free-behavior audio recordings during awake-singing. Songbird neuroanatomy and vocal behavior have direct parallels to human speech [208], thereby offering a valuable model for exploring neurally-driven speech synthesis applications.

The B1 dataset presents a unique challenge for stable decoding approaches. Neuropixels probes may exhibit nonstationarities that vary significantly in type and timescale compared to traditional microelectrode arrays. Given vocal ground-truth, decoders designed for B1 aim to synthesize high-fidelity continuous amplitude waveforms or spectrogram representations of vocal output. This strategy may better preserve the prosodic elements in reconstructed vocalizations, a significant challenge inherent to current brain-to-text approaches. By developing stable birdsong decoding strategies, we aim to establish baseline methods that can be adapted to human brain-to-speech iBCIs.

4.4 Results

We seed FALCON with representative approaches to provide an initial characterization of the stability challenge. Implementation details on all baselines are provided in Section B.1.4. For all datasets, we provide standard decoders applied to held-out sessions in two ways: (1) trained in a many-shot manner using redacted data ("oracle" decoders) approximately upper-bound performance and (2) applied zero-shot ("static" decoders) to lower-bound performance. For motor datasets (M1, M2, H1), we fit a Wiener Filter (WF; ridge regression with history) on inferred neural firing rates derived from an exponential spike smoothing kernel (see Section B.1.4). A single-session WF is a simple but effective baseline for offline decoding from high quality spiking activity and is a representative default method for closed loop control. We also train a single-session recurrent neural network (RNN) and a multi-session Neural Data Transformer (NDT2 Multi [179]) to establish the performance of higher capacity nonlinear models. For the human handwriting dataset H2, we provide an RNN trained on multiple sessions to predict English letters from neural activity, and a second RNN that additionally uses language models (LMs) as priors to correct RNN outputs and improve accuracy. On B1, we apply the EnSongdec decoder [209] which predicts song embeddings from spiking data using a feedforward network and synthesizes them into continuous birdsong using a pretrained EnCodec model [210].

We also sample state-of-the-art methods for robust decoding to demonstrate how existing approaches perform on FALCON datasets. For movement datasets, we provide two deep-network-based unsupervised few-shot alignment approaches, Nonlinear Manifold Alignment with Dynamics (NoMAD) [46] and Cycle-consistent

Generative Adversarial Network (CycleGAN) [143]. Similar to neural latent variable models [193], NoMAD and CycleGAN use an RNN and an MLP, respectively, to infer neural firing rates through a Poisson firing rate model. These methods then apply distributional alignment to match inferred neural firing distributions on held-out sessions to those of held-in sessions. Different single-session models are trained to provide the different held-in scores. Additionally, we train NDT2 Multi models that only use calibration data. The H2 stability baseline is a test-time adaptive method that uses the LM-corrected outputs as pseudo-labels to iteratively recalibrate the RNN (Continual Online Recalibration with Pseudo-labels; CORP [178]). As vocalization decoding has seen limited development of specific decoder stabilization approaches, we pose B1 as an open question and solicit potential solutions from FALCON submissions.

4.4.1 FALCON datasets exhibit unstable decoding performance across sessions.

We first show that FALCON datasets exhibit qualitative nonstationarities, reflecting the challenges faced in iBCI use. **Figure 4.3a** shows neural spiking activity from all sessions in the M1 dataset. It is clear that neural firing exhibits different properties across sessions. We also visualize this data in 3 dimensions using principal components analysis (PCA) (**Fig. 4.3b**). Under a common projection, we plot average time courses for different reach directions. Directions are clearly separable in both sessions (supporting decoding), but the required decoding map changes between sessions.

Next, we quantify that each session's neural activity can provide good decoding of behavior. We train oracle decoders for each session, which use all non-evaluation data. Specifically, held-in oracle decoders use the data from the calibration split, and held-out oracle decoders use both the calibration split and the redacted data. All oracle decoders are evaluated on respective session evaluation splits. For movement datasets, oracle decoders consist of Wiener Filters with cross-validated history (WFs) or single-session RNN models (**Fig. 4.3c-e**, blue/red). For H2, the oracle decoder is an RNN trained to predict letters from neural data, trained jointly on all held-in calibration splits and incrementally with each session's held-out calibration and redacted splits (**Fig. 4.3f**, blue). For B1, we apply an EnSongdec model [209], which uses neural data to predict song embeddings before reconstructing song spectrograms (**4.3g**, blue). For all datasets, variability in oracle decoding performance is nontrivial

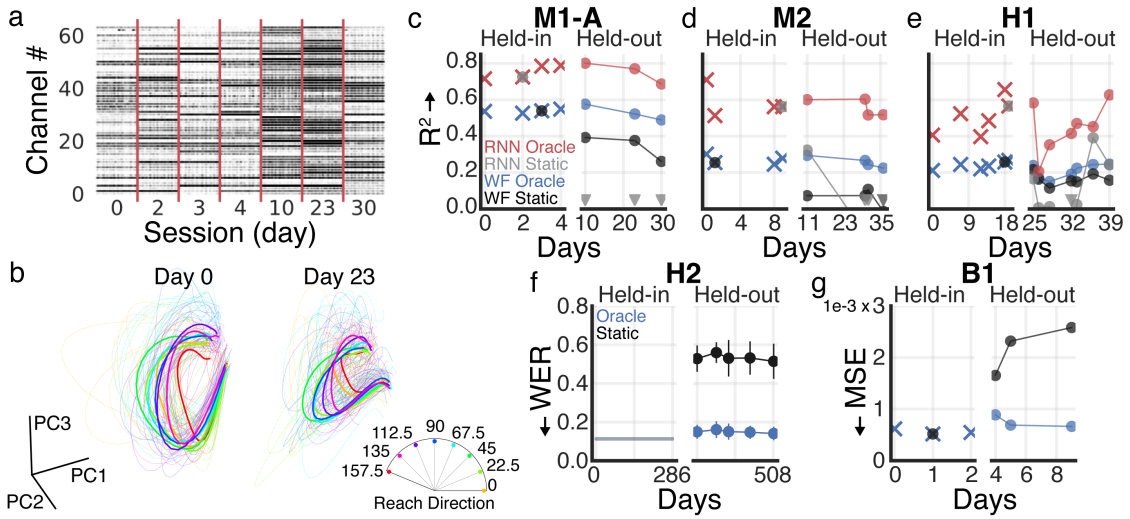

Figure 4.3. Static decoders exhibit decoding instabilities on FALCON datasets. **(a)** Raster plot showing 1 minute of data for each `M1-A` session, separated by red vertical lines. **(b)** Neural trajectories from PCA fit on `M1-A` Day 0 smoothed spiking activity and applied to Day 0 and Day 23 smoothed spiking activity. Colored by reach direction. Thick lines show the average of all reaches in a given direction and thin lines indicate single reaches. **(c-e)** R^2 of oracle decoders (RNN red, WF blue) and static decoders (RNN gray, WF black) for held-in and held-out splits for `M1-A`, `M2`, and `H1`. Higher values indicate more accurate performance. Downward triangles indicate points with negative R^2 otherwise not visible on these axes. Selected static decoders are annotated with a gray or black circle. **(f)** Word error rate (WER) of oracle (blue) and static (black) decoders for `H2`. Lower values indicate more accurate performance. Rather than training one model per held-in session, the held-in decoder is trained using all held-in sessions (performance denoted by horizontal line). Held-out dataset performance reported as mean ± standard deviation across 5 random seeds. **(g)** Mean squared error (MSE) of oracle (blue) and static (black) EnSongdec models for `B1` held-in and held-out splits. Lower values indicate more accurate performance. Static decoder chosen from held-in datasets indicated with black circle.

but small, implying that performance drops from transferring decoders to new sessions are not due to degraded neural data or a lack of correspondence between neural and behavioral data.

Finally, we quantify decoding instabilities in each dataset with zero-shot static decoders. The specific static decoder was chosen from the held-in session oracle decoders as the highest performing on the evaluation split of the other held-in sessions, to approximate good generalization to held-out sessions. We apply the decoder unmodified to the held-out datasets, simulating an iBCI decoder's naive (zero-shot) performance on a new session without recalibration. All datasets showed marked decoding instability (**Fig. 4.3c-g**, black/gray), with drops in WF performance up to

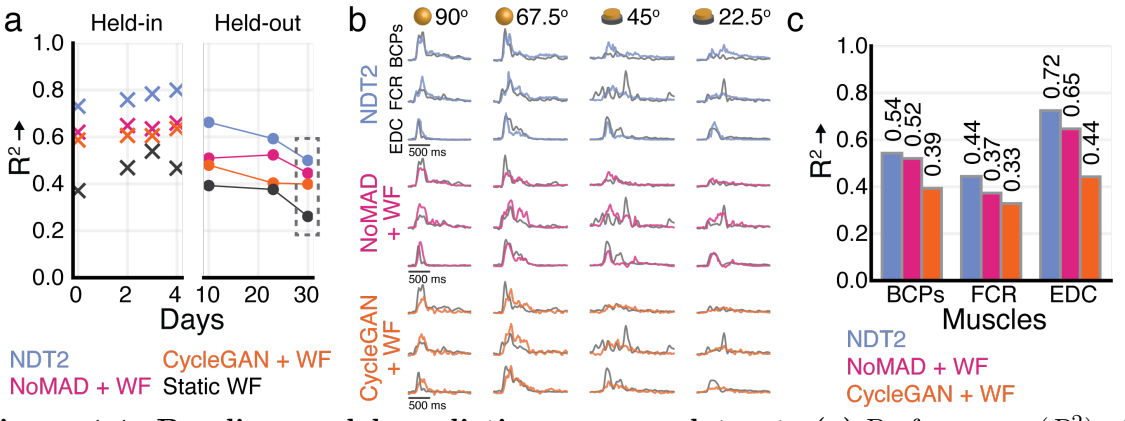

Figure 4.4. Baseline model predictions on `M1-A` dataset. (a) Performance (R^2) of each baseline model on individual held-in and held-out `M1` datasets. Box indicates the dataset that will be elaborated on in later panels. **(b)** Example decoded EMG traces for each baseline approach. Three of the sixteen total muscles shown: biceps (BCPs), flexor carpi radialis (FCR), and extensor digitorum (EDC). Each column is an example trial for one object (sphere or button) and location (angle) pair. Gray traces are the measured EMG for that muscle and experimental condition, and colored traces are the EMG predicted by each decoder-stabilization method on Day 30. **(c)** R^2 values computed for three individual muscles. Together with the remaining muscles, these values comprise the variance-weighted R^2 presented in panel (a).

0.28 R^2 (`M1`), 0.27 R^2 (`M2`), and 0.14 R^2 (`H1`). RNN decoders exhibit more instability on FALCON movement datasets, with drops up to 1.94 R^2 (`M1`), 0.82 R^2 (`M2`) and 0.78 R^2 (`H1`). Communication datasets demonstrate similar trends – error increases as much as 0.40 WER (`H2`) and 9.03e-4 MSE (`B1`).

4.4.2 FALCON baselines demonstrate the difficulty of improving `M1` decoder stability.

We next compare current few-shot approaches applied to `M1-A` in detail. On the held-in datasets, NDT2 yields the highest performance, followed by NoMAD + WF, and CycleGAN + WF (**Fig. 4.4a**). From held-in to held-out sessions, NDT2 dropped by at most 0.30 R^2, NoMAD by at most 0.21 R^2, and CycleGAN by at most 0.24 R^2. Compared to the static WF (drop $\leq$ 0.28 R^2), the baseline approaches show at most a marginal improvement, indicating that stability challenges still affect all approaches applied to `M1`. Model ranking and relative performance are largely preserved across sessions, implying that averaging R^2 across sessions summarizes performance without obscuring gains on specific sessions.

For Day 30, we also present decoded predictions for three key muscles - the biceps (BCPs), flexor carpi radialis (FCR), and extensor digitorum (EDC) - for each baseline approach. In **Fig. 4.4b**, each column is an individual reach for one of the location-object pairs available in the evaluation split on Day 30. Comparing the predicted EMG traces to the measured EMG traces provides context for interpreting the R^2 numbers and understanding which features of the EMG (the baseline, the high frequency features, the magnitude) are predicted well by each method. For example, NDT2 captures more high frequency changes in the muscle activity than other methods, potentially due to its nonlinear decoding. In **Fig. 4.4c**, we show R^2 values for example muscles individually. Per-muscle performance preserves the method ranking shown in **Fig. 4.4a**, providing further confidence that the variance-weighted R^2 over output dimensions is sound.

4.4.3 FALCON baseline performance drops from held-in to held-out datasets.

Baseline results on all datasets are shown in **Table 4.1**. FALCON quantifies notable performance gaps across methods. For example, within oracle decoders, which are by definition trained using the same data, increased model complexity can substantially improve decoding (`M2`: 0.27 vs 0.77 R^2 WF/NDT2 Multi; `H2`: 0.11 vs 0.02 WER RNN Multi/+ LM). Moreover, it is unsurprising that using more data will provide large performance gains (ZS to FSU to FSS in movement datasets, ZS to TTA in `H2`). FALCON encourages the submission of novel approaches in each class of data use.

Given FALCON's flexibility to accommodate many classes of approaches, a method's held-in score may be used to contextualize its own held-out score. Oracle decoders establish the approximate variability in performance between held-in and held-out splits, which appears relatively small (e.g., max difference = 0.04 R^2 for NDT2 Multi on `H1`, 0.04 WER for RNN Multi on `H2`). Yet, all methods show sizable gaps between held-in and held-out scores, far exceeding the expected variability. In absolute terms, all decoders perform well on held-in `M1` (R^2 = 0.46-0.78), but performance drops by 0.12-0.18 R^2 on the held-out split (and the RNN has an extreme failure). `M2` and `H1`, which show lower overall decoding performance, maintain that held-out scores are only a fraction of the potential performance indicated by held-in scores. This is also true for `H2`, where CORP provides a great advance over zero-shot

Table 4.1. FALCON baselines. Metric means and standard deviations over sessions, computed for held-in data and held-out data separately. Standard deviations only shown for the held-out split, for clarity. *Metrics*: R^2 for movement tasks, word error rate (WER) for H2, mean squared error (MSE) for B1. *OR*: oracle models trained with unreleased data on held-out split. *ZS*: Zero-shot/static. *FSU*: Few-shot unsupervised. *FSS*: Few-shot supervised. *TTA*: Test-time adaptive. *Multi*: denotes training with multiple held-in datasets; otherwise models use a single held-in dataset.

Movement (Held-Out R^2 / Held-In R^2 ↑)

	Class	M1-A	M2	H1
Wiener Filter (WF)	OR	$0.53_{\pm 0.04}/0.54$	$0.26_{\pm 0.03}/0.27$	$0.21_{\pm 0.04}/0.24$
RNN	OR	$0.75_{\pm 0.05}/0.75$	$0.56_{\pm 0.04}/0.59$	$0.44_{\pm 0.13}/0.51$
NDT2 Multi	OR	$0.78_{\pm 0.04}/0.77$	$0.58_{\pm 0.04}/0.62$	$0.63_{\pm 0.08}/0.68$
WF	ZS	$0.34_{\pm 0.06}/0.46$	$0.06_{\pm 0.04}/0.15$	$0.16_{\pm 0.03}/0.20$
RNN	ZS	$-.60_{\pm 0.45}/0.52$	$-0.07_{\pm 0.23}/0.20$	$0.09_{\pm 0.18}/0.31$
CycleGAN + WF [143]	FSU	$0.43_{\pm 0.04}/0.61$	$0.22_{\pm 0.06}/0.32$	$0.12_{\pm 0.06}/0.15$
NoMAD + WF [46]	FSU	$0.49_{\pm 0.03}/0.64$	$0.20_{\pm 0.10}/0.35$	$0.13_{\pm 0.10}/0.21$
NDT2 Multi [179]	FSS	$0.59_{\pm 0.07}/0.77$	$0.43_{\pm 0.08}/0.63$	$0.52_{\pm 0.04}/0.62$

Communication (Held-Out Error / Held-In Error ↓)

	Class	H2 (WER)		Class	B1 (MSE $\times 10^{-4}$)
RNN Multi	OR	$0.15_{\pm 0.01}/0.11$	EnSongdec [209]	OR	$7.47_{\pm 0.99}/5.61$
RNN Multi + LM	OR	$0.03_{\pm 0.00}/0.02$			
RNN Multi	ZS	$0.53_{\pm 0.02}/0.11$	EnSongdec	ZS	$21.8_{\pm 3.91}/5.18$
RNN Multi + LM	ZS	$0.37_{\pm 0.01}/0.02$			
CORP [178]	TTA	$0.11_{\pm 0.01}/0.02$			

methods but yields error on the held-out datasets that is nearly 4x higher than that of oracle decoders on average (0.03 vs 0.11 WER). These results indicate that room for improvement remains in the few-shot challenge on FALCON datasets.

4.5 Discussion

FALCON extends previous efforts to benchmark models of neural data by presenting a standardized evaluation procedure for algorithms that improve decoder robustness in iBCI applications. We release datasets from 3 movement and 2 communication

tasks, spanning monkeys, songbirds, and human participants. FALCON is designed to be inclusive of many classes of approaches; we demonstrate standardized comparison of 5 different approaches for movement datasets, 3 different approaches for H2, and 1 approach for B1, each with varying complexity and data-use strategies. These initial models far under-sample the wide design space of methods; we believe further submissions to FALCON will help clarify the value of different training data and priors. We hope that FALCON will encourage new approaches to be developed and adopted for real-world iBCI devices.

We expect that FALCON will enable machine learning researchers to apply cutting-edge approaches to a neuroengineering problem. To this end, we impose minimal restrictions on training strategies: we allow zero-shot, few-shot, or test-time adaptation and provide generous compute for model inference. While the provided baselines train with only the calibration data, FALCON is compatible with foundation models and can be used to assess their efficacy for improving iBCI robustness.

Extensions and limitations FALCON's datasets, except for B1, contain constrained behavior with trial structure, derived from repeated cues to start and stop stereotyped behavior. Model memorization of trial structure can impede closed loop control and has been a major hurdle for adopting deep networks across iBCI settings [195, 86, 211]. Corroborating this narrative, NDT2 models trained on trialized data degraded in FALCON's continuous evaluation (Section B.1.5). To penalize sensitivity to trial structure, FALCON does not provide trial labels in movement decoding tasks. However, FALCON datasets are still inherently structured. Providing datasets with more naturalistic behavior is technically challenging, particularly in humans without intact motor abilities for whom intended behavior must be communicated post-hoc. Nonetheless, future extensions may endeavor to evaluate more free and diverse behaviors, bringing evaluation closer to real-world iBCI use. To more easily aggregate a large number of behaviors, advances in cross-subject or cross-task generalization [196, 179, 125, 123] motivate analogous few-shot benchmarks where users are given restricted data for a new subject or behavior.

An important consideration in interpreting FALCON is that it evaluates open loop prediction, not closed loop iBCI control. Closed loop control introduces shifts in neural data due to sensory feedback [212] and consequent user compensation. Users can correct for certain classes of decoder error, implying that worse decoder predictions

may not yield poor control [213, 214]. The popular robotics paradigm of evaluating control in simulation [215] is challenging for iBCI given the complexity of simulating these considerations. Understanding how to design evaluation that avoids this open-to-closed loop performance gap remains an open problem for the field, and it is important to note that consistent decoding in FALCON may not necessarily yield consistent real-world control. Nonetheless, FALCON solidifies a current community focus on reducing data requirements. Thus, approaches reaching performance saturation in the FALCON benchmark would significantly advance the field.

FALCON datasets all provide multiple sessions of data for individual subjects. While multi-session data is a substantial advance over single dataset benchmarks (e.g. [193]), methods can have variable performance when applied to different subjects [46, 143]. The relatively unique nature of the tasks in FALCON and the cost of intracortical experiments are currently prohibitive to providing data from the high number of subjects needed to support claims of subject generalization. Evaluation of subject generalization will be an important priority for real-world application when these datasets become more common, and FALCON can be easily adapted to support these datasets.

Finally, FALCON baselines exclusively use spiking activity for decoding. While spiking activity is the default input for many iBCIs, the experimental procedure for determining spiking thresholds often involves researcher discretion. Generally, thresholds are set as a multiple of the RMS of voltages recorded during a baseline period, but the precise multiple and protocol for baseline collection varies from dataset to dataset. To encourage research into stability methods that might avoid human variability or thresholding overall, we have additionally released the raw 30kHz broadband activity for `M2` and `B1`.

Ethical considerations Animal datasets were collected with approval by Institutional Animal Care and Use Committees. Human datasets were collected with Institutional Review Board approval, as part of clinical trials conducted under FDA Investigational Device Exemptions. Informed consent was obtained prior to any experimental procedures. Approvals and experimental procedures can be found in the primary references for each dataset.

FALCON focuses on algorithms that solve a problem specific to iBCIs. Such devices are intended to restore function to individuals with disabilities or impairments

resulting from brain injury or disease. However, their widespread adoption raises ethical considerations with respect to the impact of these devices on human identity, privacy, and equity, which are the subject of ongoing study [15, 216].

FALCON also makes use of previously collected animal datasets. Animal models are critical to neuroscientific research that aids in improving our understanding of the brain and develops medical devices for the treatment or assistance of neurological disorders. We hope that by releasing standardized animal datasets, the FALCON benchmarking effort will contribute to the minimization of redundant data collection by allowing researchers to make better use of existing data.

Chapter 5

A Generalist Intracortical Motor Decoder

This chapter is reproduced from "A Generalist Intracortical Motor Decoder," by Joel Ye, Fabio Rizzoglio, Adam Smoulder, and colleagues.

5.1 Introduction

Intracortical neural data collection is growing rapidly. This growth comprises not only larger individual datasets with more neurons and higher behavioral complexity [217, 218], but also an increase in the collective number of datasets. This wealth of data presents an opportunity to develop insights and applications that span datasets, provided we can reconcile their inherent diversity. Large deep networks appear very suitable for this task, so much so that the generic creation of deep networks operating on broad domain data has been termed foundation modeling [90]. Efforts to create foundation models are now proliferating beyond their origins in natural language processing (NLP) and computer vision (CV) into many domains of engineering and science [219]. Here, we evaluate the efficacy of foundation modeling for motor decoding from intracortical spiking activity.

Motor decoding is a valuable domain for characterizing neural data foundation models. Academic and industrial efforts to create iBCIs for neuroprosthetics provide a path for scaling data collection from hundreds to thousands of subject-hours, and also fuel a need for pretrained models that generalize quickly and perform robustly

for new users and settings. Behavior prediction metrics for BCI performance are also more intuitive for benchmarking progress than neural data prediction or the abstract goal of providing scientific insight (e.g. with latent variable models or in silico models) [127, 220]. Finally, recent work has shown that deep networks are able to transfer learn across motor cortical datasets collected at different timepoints, subjects, or tasks [180, 10, 137]. These ingredients provide the motivation and means for scaling neural data modeling.

However, scaling may be constrained by the unique variability present in each neural dataset. This unique variability induces irreducible error, not from the neural data signal to noise, but from what the same neural data means across contexts. To be precise, we distinguish two sources of unique variability introduced by comparing across subjects. The first difference stems from the physical difference in electrodes and recorded neurons, which we term sensor variability. For example, suppose we have toy 2-neuron subjects, where one neuron fires on leftward motion and the other fires on rightward motion. No amount of pretraining will reduce the data needed from *new* subjects to infer each neuron's preferred direction. The second difference is the residual variability in subject behavior and neural dynamics. This is also significant as shared structure between neural datasets, as probed by linear methods, fade quickly beyond the data's top principal components [123, 125]. Since pretraining cannot address the unique variability of each neural dataset, pretraining gains overall may be limited. Neural scaling efforts must therefore be alert to whether performance limits are due to fundamental data barriers or methodological shortcomings.

We combined simple tokenization and a standard autoregressive Transformer to enable pretraining over diverse data and fine-tuning to new tasks without any task-specific parameters (Fig. 5.1A). We pretrained this model, which we call Neural Data Transformer 3 (NDT3), using up to 2000 hours of neural and behavioral data from motor neuroscience experiments with monkeys and clinical intracortical BCI (iBCI) trials with humans. We then evaluated NDT3's decoding performance on eight motor datasets (Section 5.3.1), finding that tuning NDT3 yields models that either improve over task-specific models trained from scratch when task data is under 1.5 hours, and match them beyond this limit (Fig. 5.1B). Further, these gains persist under real-world distribution shifts (Section 5.3.3). These benefits may enable more complex experimental design and decrease the burden of decoder training for people using iBCIs. However, beyond the 1.5 hour limit, we observe that increasing data scale

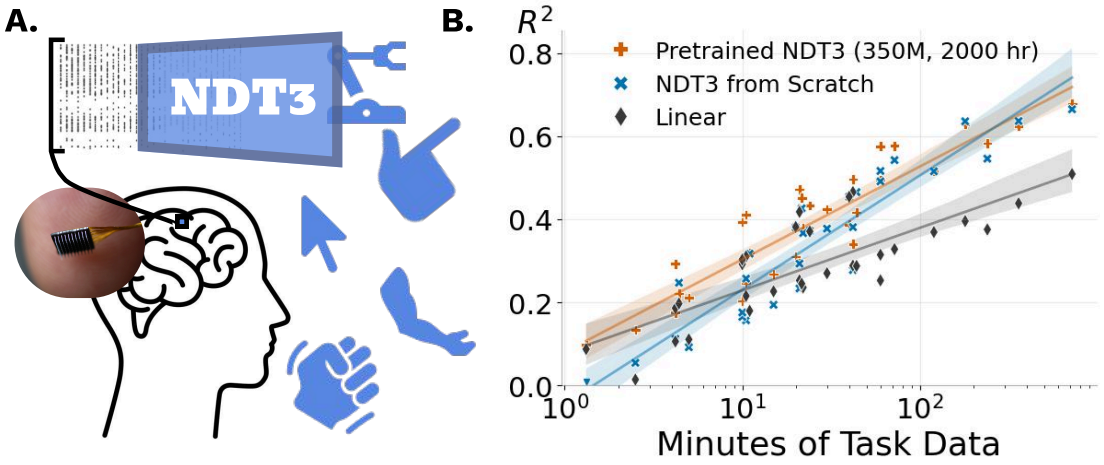

Figure 5.1. A. NDT3 decodes intracortical spiking activity into low-dimensional time series for various motor effectors[1]. **B.** We aggregate decoding performance on tasks with variable amounts of data (from Fig. C.6). Pretrained NDT3 models reliably outperform from-scratch models and linear baselines, up to 1.5 hrs of data.

without increasing model capacity degrades performance, which provides two indicators that data heterogeneity will challenge the productivity of scaling on neural data. To guide future efforts to improve scaling, we analyze two measures of generalization where NDT3 currently fails, in robustness to input sensor variability and output stereotypy (Section 5.3.2).

5.2 Approach

5.2.1 Data

NDT3 models datasets of paired neural spiking activity and behavior (Fig. 5.2). Given our focus on motor decoding, most of the data comes from devices implanted in motor cortex of various monkeys and humans (Fig. 5.2A). These devices are intracortical multielectrode arrays or probes that record 30 kHz extracellular potentials. After bandpass-filtering, spikes are extracted from these potentials when they exceed preset thresholds determined by the researcher. The resulting neural data in our pretraining are diverse (Fig. 5.2B top). Data can have markedly different profiles across electrodes due to physical separation (being on different electrode arrays in the same subject (left)), and the interaction of recording noise and threshold preprocessing can yield

[1]Photo courtesy of Nicho Hatsopoulos and The Chicago Tribune.

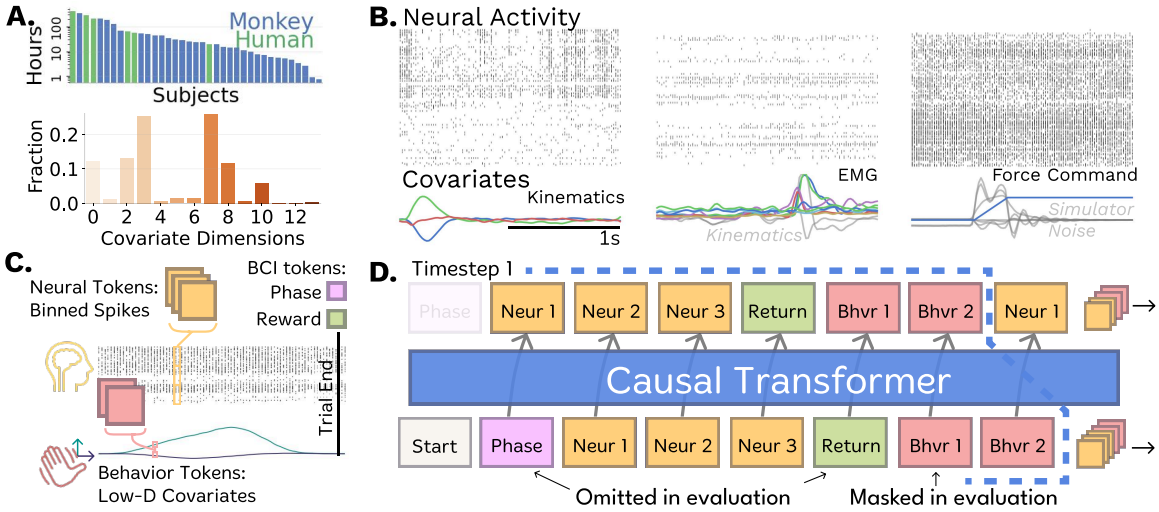

Figure 5.2. NDT3 Data and Model Design: A. NDT3 models paired neural spiking activity and behavioral covariate timeseries. We plot the distribution of 2000 hours of pretraining data by subjects (top) and covariate dimensionality (bottom). **B.** Examples of the neural and behavioral data for each of the three types of behavioral covariates in pretraining: kinematics, EMG (electromyography), or forces. Not all modeled dimensions in data are meaningfully task-related (right, grey behavior). **C.** Neural spiking activity is tokenized in time by binning the number of spikes every 20 ms, and in "space" using patches of channels (usually 32), as in NDT2 [10]. Behavior is low-dimensional in our data, so we use 1 token per behavior dimension, also per 20 ms timestep. NDT3 also pretrains on data from BCI control, which we annotate with two additional tokens. The phase token indicates whether the user is controlling or observing the behavior and the reward token indicates if the BCI task was completed. **D.** NDT3 models tokens in a single flat stream with linear readins and readouts. Every real-world timestep (shown by the blue cutout) yields several tokens, which we order to allow causal decoding at inference-time. At inference, we omit return and phase tokens and zero-mask behavior tokens.

either many inactive channels (middle) or high and noisy firing (right).

The typical behaviors in the pretraining data are reaching and grasping, nearly all from experimental paradigms that consist of short, repeated trials. While neural data were always recorded from microelectrodes, motor covariate signals came from various sensors. In monkey datasets, these sensors measure actual limb activity (e.g., Fig. 5.2B, left: limb kinematics from optical tracking, middle: electromyography (EMG)). In human datasets, physical movements are typically not possible, so the behavioral signals are programmatically generated. These signals are "paired" with the neural data in that they are cued or otherwise instructed to the person, who will attempt or imagine the corresponding behavior, such as grasping at a specified force level (Fig. 5.2B right). This force panel also shows that in pretraining, we cannot

always automatically discern the primary task covariates (e.g., blue line, force, in the panel) from other recorded behavioral variables (grey). Thus, some behavior variables may be unpredictable. Finally, we include closed loop iBCI data, where some behavior is generated by an iBCI decoder (not NDT3, see modeling strategy in Section 5.2.2). The data used for pretraining NDT3 is detailed in Section C.1.3.

We minimize preprocessing of these data to maximize the applicability of our generalist model. Kinematic signals are typically converted to velocities, and all behavior (kinematics, EMG, force) is normalized per dataset such that the maximum absolute value of each variable is one. Data are cut or concatenated into fixed length sequences, without additional annotation of data discontinuity. This strategy, common in language modeling [221], homogenizes the data for improved GPU utilization while maintaining throughput of real data. We used a length of two seconds as it is roughly the timescale of the behavior in our data (Fig. 5.2A). Sequences with no spikes or covariate variability are discarded. In total, this yielded about 3 million sequences, 1 billion neural tokens, or 1750 hours of recorded data. We round this to 2 khrs in text for simplicity.

5.2.2 Model

NDT3 is an autoregressive Transformer with linear readin and readout layers per modality, similar to GATO or TDMs [222–224]. This requires data tokenization (Fig. 5.2C). We tokenize neural data by patching spike counts [10]; each token is a vector of the binned spikes at a fixed temporal resolution (20 ms) and spatial dimension (32 channels). For example, spikes sampled from an electrode array with 100 channels would patch into $4 = \lceil 100/32 \rceil$ 32D neural tokens per 20 ms timestep. As the behavioral variables are already low-dimensional, we simply assign 1 token per dimension at the same temporal resolution as neural data. Finally, we add tokens marking whether the behavior are generated by a BCI system or by physical limb movement. While measured kinematics, EMG, or force will reflect a natural relationship with neural activity, behavioral data from BCI tasks are controlled by a program or learned decoder. We frame BCI-driven behavior as suboptimal control [84] and adopt a scheme from Decision Transformers [225, 226]. In this scheme, we use a Phase token to denote timesteps where behavior is driven by neural activity (i.e. native body movement or BCI control) versus programmatic, open loop control. We also use a

Return token reflecting controller quality based on future task completion. Note that these signals are only considered for pretraining, and are ablated entirely from the model at evaluation. Similarly, input behavior tokens are masked out in inference, so that the model input only indicates how many behavioral dimensions must be predicted. NDT3 is trained with mean-squared error for prediction of behavioral variables, and categorical cross-entropy losses for prediction of neural spike count and return.

All modalities are flattened into a single token stream, with the multiple tokens in each real-world timestep sorted to respect a canonical, causal order required for control (Fig. 5.2D). We use rotary embeddings [227] to track real-world timesteps, and learned, modality-specific position embeddings to distinguish tokens within a timestep. Note that NDT3's autoregressive objective will condition some tokens on other tokens from the same real-world timestep.

Pretraining and Fine-Tuning We pretrain NDT3 models over many scales of pretraining data (1.5 hrs to 2 khrs) and in sizes of 45M and 350M parameters to assess the impact of data and model scaling. Pretraining is stopped early according to validation loss or terminated at a maximum of 400 epochs. The 200 hour, 45M model trains for 480 A100-hours while the 2000 hour (2 khr) 350M model takes 20K A100-hours. Fine-tuning maintains the pretraining objectives and updates all parameters.

5.2.3 Evaluation Strategy

Evaluation datasets and tuning Our main evaluation (Section 5.3.1) uses four human and four monkey datasets sampling varied upper limb movements, which we detail in Section C.1.3. All downstream monkeys are held-out of pretraining, and all humans are held-out of < 2 khr models. That is, we evaluate **scaling of cross-subject generalization**, as it is the most general use case for BCI-oriented foundation models. Note the human data leakage was allowed as significant data came from humans included in pre-determined evaluation datasets, and this leakage did not advantage the 2 khr model (Section C.1.2). Each dataset contains multiple sessions of data, typically from a single monkey or human. We will refer to each such setting as a "task," distinguished from the behavioral procedure performed in each dataset. Each session has unique variability, so one fine-tuning strategy, used in prior work [180, 10, 228], is

to focus evaluations by tuning separate models for each evaluation session. We instead fine-tune NDT3 jointly over data combined from multiple evaluation sessions, which helps manage compute and storage demands, and better reflects that real-world deep network based BCIs typically jointly tune over multiple sessions. Fig. C.7 confirms this joint tuning is comparable to per-session tuning for multi-session data.

Baselines We compare with Wiener filters (WF) and NDT2 [10]. WFs are a conventional linear method for motor iBCI [15]. We implement them as ridge regression with multi-timestep history. NDT2 is a Transformer that uses MAE self-supervision [129] to learn across neural datasets. NDT2 currently leads on FALCON [11], a BCI decoding benchmark that notably includes RNN baselines; we thus omit RNNs in our evaluation. We compare to NDT2 both prepared from scratch on each evaluation dataset (as in FALCON), or tuned from the public pretrained weights from pretraining on 100 hours of human data. We are also aware of two other proposals for scaled Transformers on spiking activity, POYO [180] and NDT-MTM [228]. We omit full comparisons with these models, as our reproduction of POYO underperforms NDT3 on FALCON when both models are trained from scratch, and NDT-MTM is qualitatively similar to NDT (see Section C.1.3).

Downstream Hyperparameters We tune all deep networks (NDT2 and NDT3) over 3 learning rates. This hyperparameter sweep is limited for computational tractability. Importantly, the same sweep is used for all tasks; we list it and show its sufficiency relative to wider sweeps in Section C.1.3. The best learning rate is chosen based on the average validation score over three random seeds, and we report the average score on a separate test split.

5.3 Results

NDT3's pretraining advances intracortical models by an order of magnitude in data and model scale, from 200 to 2000 hours and 10M to 100M+ parameters. In Section 5.3.1, we show that this joint scaling is productive, but only when downstream datasets are < 1.5 hours. To analyze this limit, Section 5.3.2 shows that despite outperforming baselines, NDT3's transfer remains pathologically sensitive to shifts in data input dimensions. Section 5.3.2 also shows that scaling does not improve generalization to new behaviors. Section 5.3.3 concludes by showing that NDT3's pretraining gains, despite their limits, do generalize to novel settings to provide a practical foundation

for motor decoding.

5.3.1 Multi-scale evaluation across motor decoding tasks

To set expectations for how data scale and model size will impact performance, we first measure behavior prediction R^2 computed during pretraining. We report on a test split comprising multiple sessions of 2D reaching mainly from one monkey. Models can learn this as 1.5 hours of data sampled from this task are included in all pretraining datasets. The first three bars in Fig. 5.3A compare a model pretrained on solely these 1.5 hours, or one of two 200 hour datasets: The "Subject" model uses data from 10 other monkeys, while the "Session" model uses 200 hours from the test monkey (from a separate set of experiments with similar behavior). Only the session model provides an improvement, suggesting cross-subject data is too distant to benefit a model that already uses 1.5 hours of task-specific data. The right three bars compare how further scaling to 2 khrs can actually degrade performance, suggesting dissimilar pretraining data can interfere with learning of the test task. This interference is mitigated by increasing model size to 350M parameters, consistent with recommendations to scale model size and dataset size in tandem [146, 154, 229]. However, rescued performance still remains near the 200 hour reference.

In light of previous work showing cross-subject transfer at low downstream data scales [180, 10], our upstream saturation suggests downstream data scale might influence pretraining gain. We thus evaluate downstream at multiple data scales, illustrated in Fig. 5.3B. These two datasets are from a human performing open loop iBCI calibration for bimanual cursor use [136], and a monkey performing self-paced reach to random targets [151]. Both individuals are held-out from pretraining entirely, so the task-specific data are only seen in tuning. In the bimanual task, NDT3 performance improves with increased pretraining data at all downstream data scales. The self-paced reach dataset has more ambiguous results. For example, from-scratch NDT3 is competitive at all but the smallest data scale. These examples raise the concern that pretraining efficacy will depend on not only data scale but also its quality, like the subject or behavior studied. However, such variable efficacy across downstream settings is not uncommon in pretraining [230, 91], and does not preclude an initial aggregate evaluation.

Tuning over 2000 models in 31 evaluation settings, we find that pretraining scale

provides overall benefits (Fig. 5.3C). To begin, from-scratch NDT3 outperforms the WF and NDT2 (pretrained or not). We discuss NDT2's relatively poor performance in Section C.1.2. From-scratch NDT3 performance can be improved with minimal pretraining (1.5 hrs), consistent with findings in computer vision [96]. Performance continues improving up to 2000 hours of pretraining data, but only when paired with increased model size to 350M parameters, as in pretraining. The gain of the 350M 2 khr model over other models is significant under nonparametric Wilcoxon signed rank tests, for all comparisons except the 350M 200 hr model (Fig. 5.3D).

We now return to analyzing the impact of downstream dataset scale and quality on pretraining efficacy. Fig. 5.3E organizes downstream results by data scale, revealing that said scale is the primary factor in determining pretraining efficacy. Specifically, after normalizing by 200-hr model performance to account for dataset variability in absolute performance, the distinction between pretrained models and from-scratch models vanishes by 1.5 hours. This convergence also explains the earlier observation of upstream saturation, as it also used 1.5 hours of calibration. The remaining variability across models at any fixed downstream dataset scale should be attributed to dataset quality. The effect is significant but secondary to data scale, so we detail individual dataset results in Section C.1.2. Moreover, a precise analysis on the impact of dataset quality is beyond the scope of this work, as our evaluation datasets have unique subjects and behaviors, while a controlled analysis would require multi-subject, multi-task datasets. Thus, we conclude provisionally that further scaling of NDT-style neural decoders will mainly be productive at downstream data scales $<$1.5 hours.

5.3.2 NDT3 does not generalize to novel input order and output conditions

Section 5.3.1 showed that pretraining gains saturate at data scales that, while larger than previously measured for neural data models, are practically low for BCI applications, as 1.5 hours can be reached after a few sessions of data collection. Unique variability across neural datasets does imply an eventual ceiling on pretraining gains, but this low threshold raises the concern that our modeling decisions, like tokenization, pretraining hyperparameters, or a lack of post-training, may be at fault. We next begin to dissociate these two possibilities, by highlighting NDT3's sensitivity to the specific neural inputs and behavioral outputs seen in tuning.

Input order sensitivity limits cross-subject transfer. Transfer learning is greatly reduced when using cross-subject neural data relative to cross-session neural data, even when the data are collected in identical experiments and thus controlled for other variables [10]. Limited cross-subject transfer may thus be a more precise cause of limited scaling in pretraining. This hypothesis predicts that pretraining may already account for all feasible cross-subject transfer. We show this by measuring transfer *after* large-scale pretraining, as schematized in Fig. 5.4A. Specifically, we tune NDT3 using calibration data from a single session of monkey self-paced reach [151], optionally preceded by intermediate tuning on additional cross-session or cross-subject data. Fig. 5.4B shows that cross-session intermediate tuning remains beneficial even after pretraining. In contrast, pretrained models do not benefit from cross-subject tuning. Further, cross-subject data does not benefit from-scratch models beyond the pretrained baseline (Test Calib Only). This shows **pretraining occludes further cross-subject transfer, but not cross-session transfer**, and supports the view that scaling is limited by cross-subject variability.

Recall that cross-subject differences can in turn be decomposed into sensor variability and changes in the neural dynamics underlying subject behavior. We can isolate how NDT3 resolves sensor variability while controlling for changes in aggregate neural statistics by performing intermediate tuning on cross-session data with altered input structure. We consider three alterations of decreasing strength: a complete channel shuffle, a half-token shift that preserves channel adjacencies but disrupts the channels in each (multi-channel) token, and a token shuffling (Fig. 5.4B Right). In all settings, intermediate tuning improves the from-scratch model, consistent with NDT's ability to transfer nontrivially across varied datasets. However, as in the cross-subject setting, no gains outperform the pretrained baseline (Test Calib Only). Moreover, the pretrained model only achieves positive transfer from intermediate tuning on token shuffle cross-session data. These results show NDT3's cross-session transfer, and equivalently its downstream scaling, depends on the consistency of neural data dimensions, at least at the granularity of tokens. These conclusions are also supported in extended analysis (Section C.1.2), where we vary intermediate data scales and tuning strategy.

Pretraining does not enable angular extrapolation. Increased data efficiency after pretraining suggests that NDT3 could decode a new subject's behavior without sampling its full dynamic range. For example, center-out tasks require the subject

to make stereotyped ballistic reaches and allow us to hold out reach angles to test decoding generalization. Center-out specifically provides a useful litmus test as the neural activity underlying center-out reach can be projected to a planar subspace [231], guaranteeing that some generalization is possible by construction. It is further known that non-pretrained deep networks fail at such held-out angle generalization [157]. Thus achieving this generalization would be a qualitative milestone for pretraining.

We analyze one session of an isometric monkey center-out dataset from [157], where the monkey exerts forces in one of eight angles and its force level is mapped to cursor position (Fig. 5.4C). We separate this data into 3 held-in and 5 held-out angles, and plot predictions from a Wiener Filter, a from-scratch NDT3, and a pretrained NDT3 (Fig. 5.4D). The WF, when fit to latents constructed through dimensionality reduction (PCA and LDA, see Section C.1.2 for methods), yields predictions that generalize to the held-out angle of leftward reach. Further as expected, from scratch predictions fail to generalize, and qualitatively are constrained to their nearest held-in angle. Pretrained NDT3, however, still fails to generalize, and appears to replicate from-scratch behavior of making predictions constrained to held-in angles. We quantify accuracy in Fig. 5.4E, showing that while pretrained NDT3 improves over from-scratch NDT3 overall, both models far underperform the PCA-LDA-fit WF on held-out generalization. This attractor strategy is undesirable for continuous BCI control [136], but is viable when the data are stereotyped, as is common in neural datasets. We replicate this result in two more settings in Section C.1.2. Importantly, this analysis does not confirm whether pretraining has failed to learn that simple dimensionality reduction can recover planar structure underlying ballistic reaches; plausibly, angular generalization could be achieved with a lightweight post-training protocol that discourages attractors in favor of continuous decoding.

Minimal benchmarks for neural data scaling Both sensor variability and output stereotypy pose fundamental challenges in neural pretraining, rather than trivial shortcomings specific to NDT3. For example, while our analysis raises concerns with NDT's patch-wise tokenization, the alternative of neuron-level tokenization [180] still requires input-channel identification and performs poorly in our hands (Fig. C.14). Moreover, despite enthusiasm for neural foundation models, cross-subject transfer issues persist across neural modalities [97, 232]. Given NDT3's current state-of-the-art performance, these distillations of its current limits serve as concise benchmarks for generalization that future neural foundation models will likely need to overcome.

5.3.3 NDT3 does generalize to varied real world distribution shifts.

Having understood NDT's limitations under systematic synthetic shifts, we next provide examples where its pretraining usefully generalizes under real-world distribution shifts.

Neural distribution shifts Neural data are nonstationary, with shifts rising from a mix of controlled experimental variables to more speculative factors. For example, the firing rate of different channels will evolve over the course of an hour, implying a distribution shift associated with change in time (Fig. 5.5A top left). Shifts also occur between activity in two arm postures or whether finger motion occurs under spring load or not (Fig. 5.5A top middle and right). Since these shifts are common in neural data, pretraining gains should ideally be robust to their effect. We thus tune models on data from one setting (in-distribution, ID) and measure the performance of models in that same setting and the shifted setting (out-of-distribution, OOD). Positive correlation of performance in all cases implies the ID gains conferred by pretraining persist OOD. More practically, these examples suggest that pretraining benefits are not dependent on narrow features specific to the choice of tuning dataset.

Trial structure DNNs have been observed to overfit the temporal structure of experimental data, challenging their use in iBCI control [136, 89]. For example, a DNN might learn there is always no motion before the start of a behavioral trial, independent of the neural activity. To date, these claims have been studied exclusively on un-pretrained deep networks. We next assess how pretraining affects this overfitting in open loop human cursor control data by comparing a continuous and trialized setting. In the continuous setting, we cut random one second intervals of data in training and continuously stream up to one second of data in evaluation. The trialized setting formats data to respect trial boundaries, so the model always sees data aligned to the start of behavior.

In Fig. 5.5B, we show that models using both trialized training and evaluation outperform models with both continuous training and evaluation. This implies NDT3 will learn to exploit clear trial structure in data. However, while trialized from-scratch models become subtrivial under continuous evaluation (solid blue line is off-panel), pretrained models degrade more gracefully. For example, the 350M 2 khr model evaluated continuously only performs slightly worse with trialized tuning than with

continuous tuning. Pretraining NDT3 thus reduces its dependence on trial structure, which should benefit both data analysis and iBCI control. Note, however, the contrast in these results with Fig. 5.4D, which show that DNNs clearly do overfit to tuning data in some cases. These nuances underscore the importance of rigorously evaluating model generalization in future work.

New brain areas In Fig. 5.5C, we return to multiscale fine-tuning to test how NDT3, pretrained on motor cortex, performs in somatosensory cortex (S1) and oculomotor areas (FEF and MT). The S1 data shares the same self-paced reaching behavior of Fig. 5.3B, while the Oculomotor data is an analogous visual pursuit task. The gain from pretraining over from-scratch models is high in S1, but also nontrivial in the Oculomotor dataset. While the former can be attributed to the close interaction of sensorimotor areas, the latter implies NDT3 has learned a broader prior. Although these results are encouraging, our previous results suggest these gains could reflect common experimental artifacts like trial structure, rather than neurophysiological priors (e.g. declining subject focus over time [233]). For example, this Oculomotor dataset contains 4 behavioral conditions, which may benefit from the tendency to learn classifiers shown in Fig. 5.4D rather than a prior on neural dynamics.

5.4 Discussion

Broader Impacts NDT3 is built as a generic pretrained model to accelerate BCI motor decoding workflows. To the extent it can, and with the cost and complexity of pretrained deep networks over simpler decoders considered, such BCI devices are intended to restore function to individuals with disabilities or impairments resulting from brain injury or disease. Human BCIs do however pose risks for data privacy and personal agency [234], which are accentuated by the adoption of pretraining paradigms. NDT3 may also be used in non-human primate BCI research. We believe that the increased data efficiency of pretrained models in those settings can help reduce the need for redundant data collection.

Limitations This work provides an initial assessment of scaling up neural data decoding with a Transformer. Beyond the explicitly analyzed limitations in Section 5.3.2, this work can be expanded on both architectural and data design axes. Architectural choices include the use of different neural features (spike times, bandpower) or metadata features (per-day embeddings). Data design axes are harder to explore, as it is

constrained by the expense of creating intracortical datasets (public or not). They are nonetheless vital to understand, as NDT3's results may poorly predict scaling trends on datasets with more diverse neural activity or behavior, with unconstrained (naturalistic) experimental paradigms, or with many more (1000s) of subjects.

Many fields are now pursuing large-scale deep learning as "a tide that lifts all boats" [235], hoping that improvements in effective pretraining will yield field-wide, downstream improvements. Such a unifying abstraction would be timely for neuroscience, given the increasing volume, diversity, and complexity of modern neural data. Joining other pretraining efforts on varied modalities of neural data (Section C.1.1), we trained NDT3 on 2000 hours of paired neural population activity from motor cortex and behavior, and then conducted a broad downstream decoding evaluation. Consistent with other foundation models, we found the best aggregate performance comes from increasing data scale and model size jointly (Section 5.3.1). We release the 350M 2 khr NDT3 as a strong baseline foundation model for intracortical decoding from spiking activity, with the disclaimer that benefits will likely be minimal by 1.5 hours of downstream data. To move forward, we highlight input order sensitivity and output attractors as issues future foundation models should address.

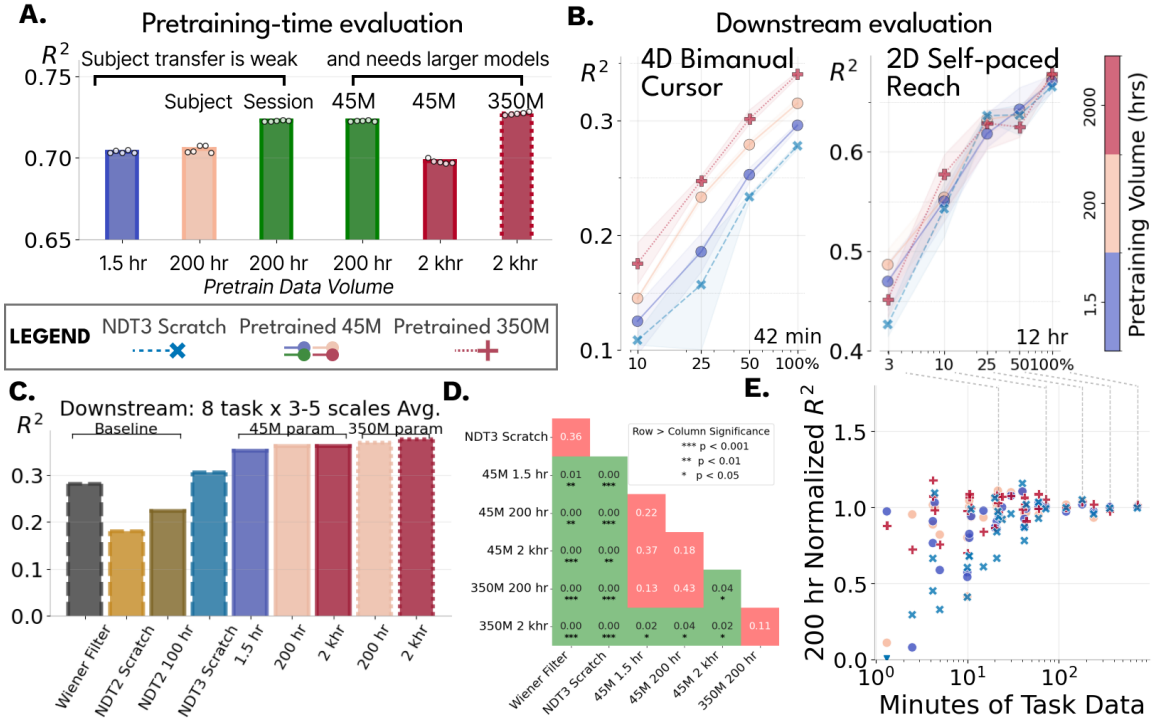

Figure 5.3. Evaluation on diverse motor tasks: A single legend is used throughout. **A.** Pretraining evaluation for 5 models, the center two bars are repeated for clarity. The top 5 evaluations across pretraining checkpoints are averaged and scattered individually. All models' pretraining data include 1.5 hours of calibration data for the test dataset. We compare a model with just this data (1.5 hr) to using 200 hours of additional data from over 10 other monkeys (Subject) vs 200 hours from the test monkey (Session). Only the additional test monkey data improves over the calibration model. The final two models pretrain on our full pretraining dataset (2 khr), which includes the 200 hours from the test monkey. Using 2 khrs underperforms the 200 hr model at 45M parameters, with performance restored by increasing capacity to 350M. **B.** Examples of good and bad data-scaling in downstream multiscale evaluation on two datasets. The bottom right text shows total time used in downstream fine-tuning. The x-axis scales this full dataset down by random subsampling. Shading shows standard deviation on 3 tuning seeds. Increasing pretraining data yields performance gains at all downstream scales in the 4D task, but effects are unclear in the self-paced reach task. **C.** Downstream performance averaged for 31 settings comprised of different downstream datasets and scales, for different NDT3s and baselines. 45M NDT3s improve with data from 1.5 hrs to 200 hrs but saturate at 2 khrs. Increasing model size to 350M parameters enables further gain. **D.** p-values computed from FDR-corrected pairwise t-tests for each pair of models. The 350M 2 khr NDT3 significantly outperforms other pretrained NDT3s, except the 350M 200 hr NDT3, and is the only model to do so. NDT2s omitted for brevity, see Fig. C.8. **E.** Per-task performance, normalized by the 350M 200 hr NDT3 performance, is shown against task time for different NDT3 models. Each vertical band shows models trained on the same evaluation setting, e.g. dashed lines show the evaluations from the self-paced reaching dataset. Model variability vanishes by 1.5 hours, implying scaling will only help below this limit.

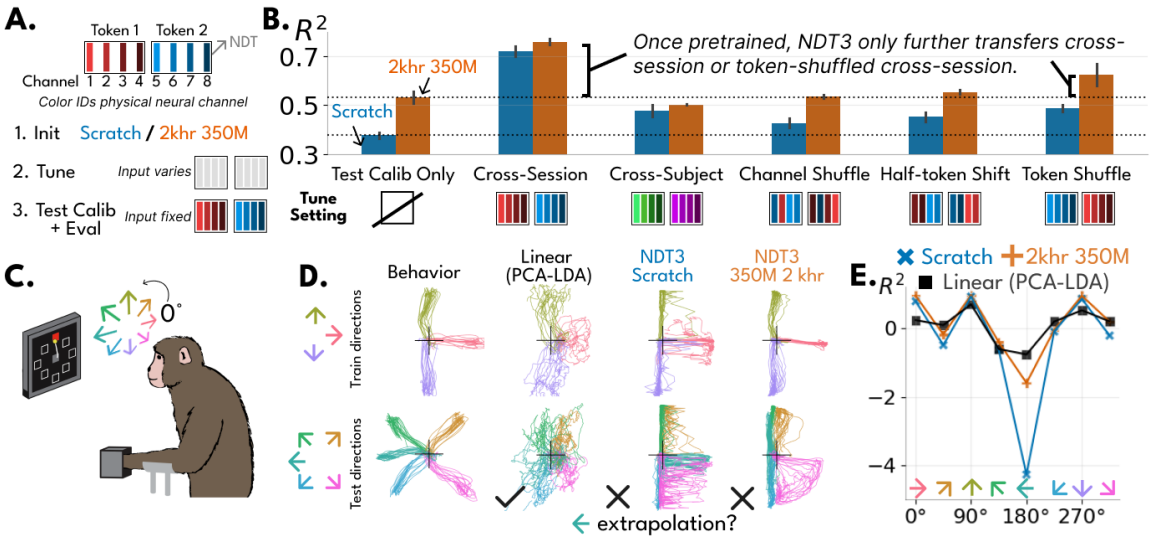

Figure 5.4. Pathological sensitivity to input order and stereotyped outputs. A. Setup for fine-tuning for input analysis. Colored bar marks a constant physical electrode channel of neural data, boxes show how they are tokenized for NDT. Schematic uses only 8 channels and 2 tokens, data has 96 channels and 3 tokens. Using models initialized from scratch or from pretraining (1), we tune and evaluate performance on a self-paced reaching dataset (3). We precede evaluation by training on an intermediate setting (2) with altered neural inputs relative to (3). (3) only uses a minute of data, while (2) uses several hours. **B.** The first three settings are no intermediate training (Test Calib Only), tuning with data Cross-Session data or Cross-Subject data with the same behavior. The latter settings alter cross-session data to emulate how cross-subject data might fail to transfer. Shuffle channel randomly permutes inputs, half-token shift rolls channels so that each channel i uses data from $i + 16$, and shuffle token permutes data tokenwise, while preserving the channels grouped in each token. Gains above the Test Calib Only baseline are only preserved under token-shuffling. **C.** We study angular extrapolation in an isometric monkey dataset where exerted forces are mapped to cursor positions in 8 different angles. We use three held-in and five held-out angles. **D.** Predictions are made either by fitting a Wiener Filter to neural data after dimensionality reduction (Linear), or from NDT3 (Scratch, 350M 2khr). While the linear model extrapolates to held-out angles, NDT3 predictions strictly follow held-in angles. **E.** Pretraining improves over from-scratch in held-in angles, but far underperforms in held-out angles.

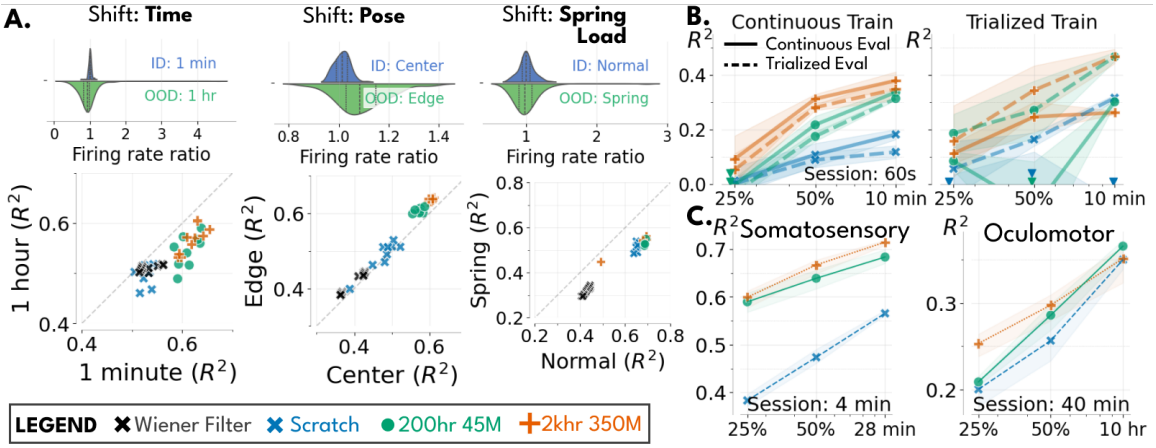

Figure 5.5. Generalization of pretraining gains. A. Models fine-tuned in one distribution of data are evaluated in-distribution (ID) and out-of-distribution (OOD). Top plots show the distribution across channels of neural firing rates from OOD and ID trials, normalized by average ID firing rates. Lower plots scatter OOD vs ID performance, with each point being a single model with different hyperparameters. The **time** shift uses two human cursor datasets collected one hour apart. Models were tuned in each block and were evaluated in the second block. **Pose** shift uses a monkey center-out reach task which was performed with the hand starting in different locations in the workspace. **Spring Load** uses a dataset of monkey 1D finger motion with or without spring force feedback. **B.** Models are evaluated on a human open-loop cursor dataset prepared in two ways. Trialized training receives inputs according to trial boundaries, varying from 2-4 seconds in length. Continuous training receives random 1 second snippets (that can cross trial boundaries). Trialized evaluation matches trialized training, and continuous evaluation is done by streaming up to 1 second of history. ▼ indicates points below 0.0. Continuously trained models perform well in both evaluation settings, while models trained on trialized data fail in continuous evaluation. **C.** Multiscale fine-tuning performance of NDT3 on datasets recorded outside motor cortex, namely S1 (Somatosensory) and FEF/MT (Oculomotor).

Chapter 6

DELETE: Denoising Electrical Events with a Transformer Encoder

This chapter is reproduced from "DELETE: Denoising Electrical Events with a Transformer Encoder," by Joel Ye, Wentao Yang, Taylor Hobbs, Robin Lienkämper, Charles M. Greenspon, Leila Wehbe, and Robert Gaunt.

6.1 Introduction

Every scientist must overcome the challenge of separating signal from noise in their data. In electrophysiology, recordings can be degraded by artifacts from the environment, subject movement, or our own experimental stimulation protocols (Fig. 6.1A). In an ideal world, each noise source has uniquely identifiable symptoms that could be best characterized and mitigated with bespoke solutions. Yet real-world noise, like the real-world data it resides in, can vary in ways that make bespoke treatment expensive. For example, electrophysiological stimulation protocols can vary in amplitude, timing, and electrode selection, resulting in a repertoire of related artifact profiles rather than a single, well-characterized artifact. Further, even if any single noise source can be mitigated, their combinatorial interactions invite additional headaches, resulting in degraded datasets or expensive curation.

To avoid bespoke denoising methods, one practice is to embrace noise as inherent to the data and turn to deep networks for end-to-end processing. However, applying this approach directly to raw, high-bandwidth sensor data is inefficient when the

signals needed for scientific inference are often much sparser. In neurophysiology, broadband electrophysiology is sampled at tens of kilohertz per recording channel, whereas individual neurons on those channels typically fire only tens of spikes per second. This data imbalance is certainly not unique to neuroscience: detectors at the Large Hadron Collider observe collisions at rates approaching 40 million events per second, while online trigger systems retain only a minute fraction for downstream analysis. This disparity has motivated anomaly-detection-based preprocessing that identifies rare events against a vast background of ordinary observations [236]. Inspired by this perspective, we propose an analogous approach to electrophysiological denoising, using local predictability to distinguish informative neural events from background structure.

Here we demonstrate this principle with a Transformer-based autoencoder [128, 129], and term our method DELETE (Denoising Electrical Events with a Transformer Encoder). Though DELETE's formulation is general, we focus our analysis on intracortical microstimulation (ICMS), where, instructively, artifact severity can increase with stimulation-protocol complexity.

As illustrated in Fig. 6.1B, the large stimulation transient and its slow decay visually dominate a small evoked response, but more importantly, their timing is much more precise than that of the physiological evoked response, as the former arise from circuitry while the latter arises from biology. To separate the two, DELETE learns to predict the recording from a local spacetime context broad enough to infer the artifact, but too limited to reliably predict the neural response.

DELETE's learning of artifact regularity resembles existing rejection methods that learn across repetitions or recording channels. DELETE is distinguished, however, by its weaker assumptions: it requires no predefined artifact windows and asserts no parametric form for the artifact. Moreover, a single DELETE model can train across multiple types of spatial or temporal artifacts, including different stimulation paradigms, replacing a toolbox of artifact-specific methods with one general-purpose denoiser.

We build this case in three stages. We first show that DELETE performs comparably to competent baselines for artifact suppression across spatial-noise and stimulation-artifact simulations, and use these simulations to characterize key hyperparameters. We then benchmark DELETE against other denoising methods on real recordings, showing that it largely preserves neural activity in stimulation-free recordings and

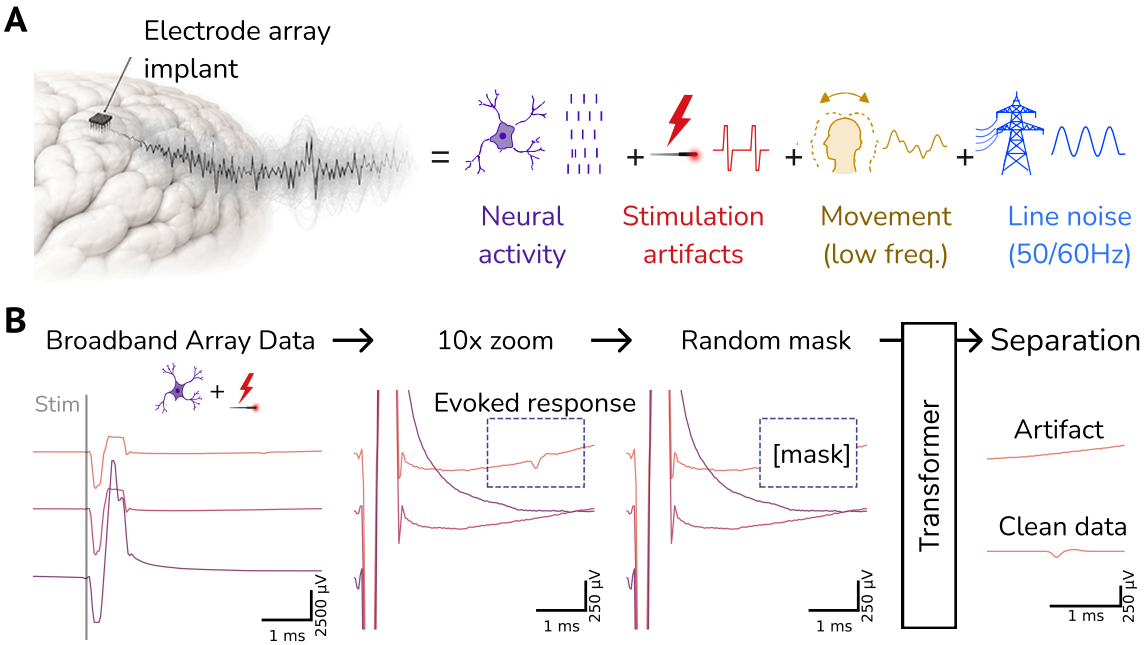

Figure 6.1. DELETE removes predictable artifacts in data to recover neural activity. (A) Electrophysiological recordings reflect neural activity, but are also contaminated by varied real-world noise sources. (B) DELETE operates on multichannel broadband data. The depicted pipeline shows three channels contaminated by a large electrical stimulation artifact. Zooming reveals the target evoked response. DELETE is trained to reconstruct random obfuscated subsets of the data from its surrounding spacetime context. With the assumption that neural signal is less predictable than artifact, DELETE's training will emphasize the predictable slow decay of the stimulation artifact, thus enabling separation of artifact and denoised data.

performs consistently across our challenging human stimulation datasets. Finally, we demonstrate DELETE's utility beyond heuristic metrics, deriving neurometric thresholds from recovered activity that correlate with psychometric detection thresholds.

6.2 Methods

6.2.1 DELETE approach

We summarize DELETE's training and deployment in Fig. 6.2. DELETE is a masked autoencoder (MAE) [130, 129], a deep network trained to reconstruct an input context when portions are masked out. For denoising, DELETE's context is a short, contiguous chunk of broadband electrophysiology. We used a 32-channel by 6 ms window. We use 32 channels as given by the physical electrode count in each of our electrode

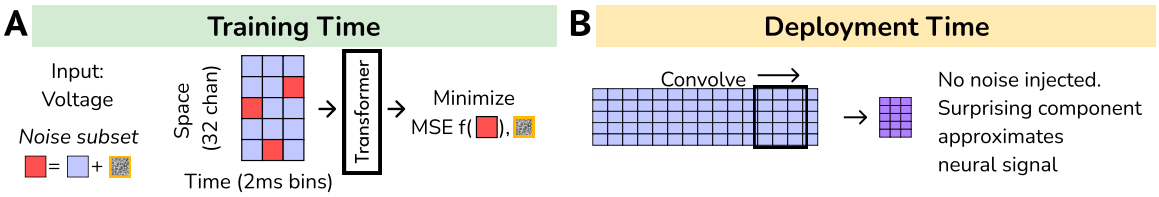

Figure 6.2. DELETE training and deployment. (A) Training: A random subset of the multichannel space × time input is masked by adding Gaussian noise. DELETE is trained to estimate the added noise. (B) Deployment: DELETE is convolved across broadband recording without masking, and its output is used directly as the de-artifacted neural signal.

arrays, while we use 6 ms according to sensitivity analyses in Section 6.3.2. This spatiotemporal input is split into tokens for the Transformer. We used a 2 ms slice from each channel as a token, resulting in 96 tokens. At 30 kHz sampling, each token starts as a 60-dimensional vector. We linearly project this raw data into and out of the Transformer's hidden dimension. This setup thus far closely resembles other autoencoders applied to neural data, including the work of Talukder et al. [131] and DeepInterpolation [237]. Even when oriented around denoising, however, these existing studies focus on what the model can recover. DeepInterpolation, for example, uses these predicted outputs to remove imaging shot noise. In contrast, here we are interested in what DELETE *cannot* predict.

In training (Fig. 6.2A), DELETE receives inputs with a random subset of tokens obfuscated. Due to the large dynamic range of artifacts relative to neural signal, we find it more numerically stable to implement this obfuscation with additive Gaussian noise rather than standard blanking. Further, the training objective is modified to predict not the original, true signal, but instead the injected noise itself. Estimating this noise tasks the model with distinguishing the corrupted input from native structure that can at least partially be reconstructed from the surrounding context. When applied without artificial noise, the same operation instead outputs the naturally unpredictable components of the recording, such as evoked spikes. For models trained on real data, we set the injected-noise amplitude to $\sigma = 100\,\mu$V, relative to the amplifier's data range of 8,000 μV. This value was chosen heuristically to minimize distortion of recovered spike amplitudes; see Section 6.3.2. Models are selected by best validation loss with early stopping.

The training data need not consist of curated artifact windows. For intermittent artifacts like stimulation, including stimulation-free periods in training allows the model to be deployed continuously. At test time, DELETE is applied in sliding windows

across the raw recording, providing a continuous preprocessing layer (Fig. 6.2B).

Our reference 32-channel model uses eight Transformer layers with a hidden dimension of 256 and four attention heads per layer. During training, we independently corrupt each input token with Gaussian noise with probability 0.2 and evaluate mean squared error between the predicted and injected noise only on corrupted tokens. We train this model with the MARS optimizer [238], using a nominal learning rate of 3×10^{-4} and cosine decay.

6.2.2 Datasets

We evaluate DELETE on both simulated and real-world electrophysiology with varied artifact type and severity. Our simulations extend methods used in previous work [239, 240], while our real-world evaluations use human sensorimotor recordings and an open-source macaque reaching dataset. We focus on recovering neural spiking activity, using a 30 kHz sampling rate for simulations and human recordings and the native rate of approximately 24.4 kHz for macaque recordings.

Simulations

We study spatial (cross-channel) artifacts by extending the linear-mixing simulation of Musial et al. [239] to a 64-channel array. Each channel reports the sum of sampled neural spike waveforms, per-channel private bandlimited Gaussian noise, and a linear combination of 32 latent noise sources shared across channels. Spike times are drawn per channel from a 4th-order Gamma inter-spike-interval process, and waveforms are sampled from five units recorded with Neuropixels in mouse brain [233]. To evaluate DELETE's ability to multiplex across artifact regimes, we generate multiple realizations by varying the mixing coefficients of the latent noise sources across channels.

To study temporally structured stimulation artifacts, we adopt the intracortical microstimulation (ICMS) simulation of Wang et al. [240], which was developed to benchmark artifact removal for an ICMS-based visual prosthetic. Trains of stimulation pulses are delivered at 200 Hz with current amplitude swept across 20–80 μA in 20 μA steps on a single channel in a 64-channel array. These pulses produce large, lightly varied artifact templates alongside spontaneous spikes, ICMS-evoked spikes, and spike-triggered field potentials; both the artifact and the evoked response decay with distance from the stimulating channel. We extend this base setup with *Slow Decay*

and *Common Noise* variants to stress different baseline methods. *Slow Decay* extends the exponential decay of the stimulation artifact, which effectively adds DC offsets across pulses and degrades artifact stereotypy. *Common Noise* adds a shared Gaussian noise component to all channels, which disadvantages single-channel curve fitting.

Participants and intracortical recordings

For real-world evaluation, we apply DELETE to stimulation-free (Section 6.3.3) and ICMS (Section 6.3.4) data. These data were primarily collected from five human participants (P2–P5 and C1) with spinal cord injury who were implanted with NeuroPort Electrode arrays (Blackrock Neurotech) in sensorimotor cortex. We also analyzed macaque data from motor-cortex during a random-target reaching task [151]. Each recording channel corresponds to an electrode; we generally use electrode for the physical site and channel for its recorded signal.

Human baseline and ICMS data were collected as part of a longitudinal study of brain-computer interfaces (BCIs) for sensorimotor neuroprosthetics, which was approved with an Investigational Device Exemption from the FDA and by institutional review board protocols at both the University of Pittsburgh and the University of Chicago. The clinical trial is registered at clinicaltrials.gov (ID: NCT01894802) and informed consent was obtained before any experimental procedures. ICMS was delivered to somatosensory cortex using Neuroplex ES stimulation cables (Blackrock Neurotech) and a CereStim C96 stimulator, which together enabled concurrent stimulation and recording. Baseline datasets were collected from participants either at rest or when prompted to imagine moving their upper limb in an unstructured manner. In ICMS datasets, the participants were either prompted to report stimulus detection or were at rest.

We measure preservation of neural activity with the human baseline and macaque data. In the human data from three participants (P2–P4), we assess firing-rate preservation using per-channel Pearson correlations between rates with and without DELETE preprocessing, computed from spike counts binned at 1 ms and smoothed with a Gaussian kernel ($\sigma = 10\,\text{ms}$). We also compute the power spectral density of these baselines using Welch's method with 200 ms windows. We compared decoding of finger velocity under three preprocessing conditions: Raw (no denoiser), LRR, and DELETE. DELETE was fine-tuned on the five target recordings without behavioral labels. Finger velocity was predicted by ridge regression using the current and six

preceding 20 ms bins of exponentially smoothed spike counts ($\tau = 240$ ms). For each preprocessing method, the spike-detection threshold was selected from 4, 4.5, and 5 times the channel-wise standard deviation of the filtered signal to maximize mean validation decoding R^2 across the five sessions; final performance was evaluated on held-out test data.

Intracortical microstimulation protocol

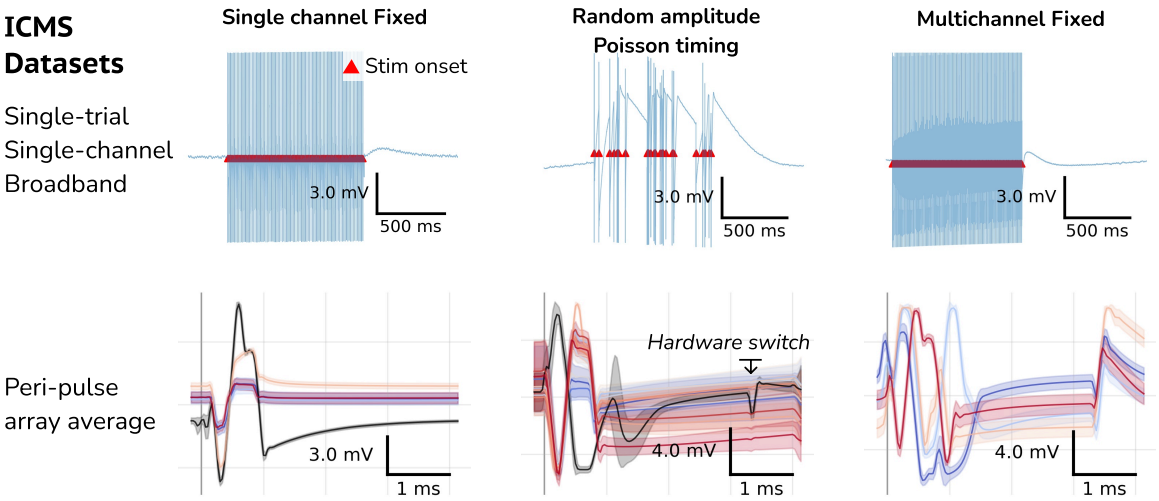

Figure 6.3. ICMS datasets. Broadband recordings under three intracortical microstimulation (ICMS) protocols, ordered by increasing artifact severity. Top: Example trials showing the stimulating channel's broadband activity; red triangles mark stimulation onsets. Bottom: Pulse-averaged peri-pulse activity, shown for sampled channels. Shading shows standard deviation across pulses. The black channel is the stimulated channel for the Single channel Fixed and RAP protocols; all shown channels are stimulated in Multichannel Fixed. The RAP and Multichannel Fixed averages show a secondary transient at a fixed 3.3 ms after each pulse, caused by a hardware switch on our stimulation platform.

In each ICMS experiment, stimulation trains were delivered to one of two sensory arrays. Individual pulses were fixed to a 200 μs cathodal phase, 100 μs interphase interval, and 400 μs anodal phase. For our primary evaluation benchmark, we varied pulse trains in amplitude (2–80 μA), pulse timing, and electrode selection, according to one of three protocols: Single channel Fixed, Random Amplitude Poisson-timing (RAP), and Multichannel Fixed, as illustrated in Fig. 6.3. In Single channel Fixed data, each pulse is delivered to a single electrode at 100 Hz. In RAP data, each pulse is generated according to an inhomogeneous Poisson process, and delivered to a single electrode uniformly drawn from a subset of nine electrodes on the array. The rate of

the generating Poisson process shifts every 200 ms and is randomly drawn from 20 to 100 Hz. In Multichannel Fixed data, stimulation is again performed at 100 Hz, but each pulse stimulates a fixed set of four electrodes on the array. We use five datasets under each protocol to form a benchmark (Section 6.3.4) spanning four participants (P2–P5), with approximately 60,000 pulses across about 15 minutes of stimulation in total.

For our neurometric analysis, we combine detection task data and passive, task-free datasets. These datasets are a superset of the Fixed benchmark datasets, with stimulation on single electrodes, constant frequency, and varied amplitudes across trials. The detection task estimates the amplitude needed to detect ICMS in a two-alternative forced-choice (2AFC) format. The amplitude sampling is determined either by a transformed up-down staircase [241] or by the method of constants, which sweeps predefined amplitudes a fixed number of times [68]. The passive stimulation data perform a similar fixed sweep over predefined amplitudes, but without a report. Most neurometric datasets are 100 Hz, 1 second trains, along with some 100 Hz, 0.3 second passive trains and 50 Hz, 1 second detection task trains. We analyze 35 electrode-days spanning 23 distinct stimulating electrodes across four participants (P3–P5 and C1). We define an electrode-day as recordings collected while stimulating one electrode in one participant on a single day at a fixed stimulation frequency.

Human ICMS dataset processing

We used the human ICMS recordings to benchmark DELETE and several baselines in a standardized pipeline (Fig. 6.4). The pipeline takes broadband recordings from the two electrode arrays (64 channels total) and annotated stimulation pulse times (stage 1). We then apply the selected artifact removal method (stage 2). Our evaluated methods varied in their treatment of the stimulation transient, which often saturated the amplifier's dynamic range and thus left no valid neural signal. Even methods that attempt to remove it (ERAASR and DELETE) generally fail to do so completely. We see this in the residual transient in the artifact removal stage. Thus we unconditionally linearly interpolate the data from 0.1 ms before to 2 ms after each stimulation pulse (stage 3). Some of our ICMS datasets (RAP and Multichannel Fixed, see Fig. 6.3) contain a secondary transient artifact 100 samples (3.3 ms) after the stimulation pulse when the stimulator circuit switches off for device safety. We similarly interpolate an additional 0.67 ms interval at this timepoint when these artifacts are present.

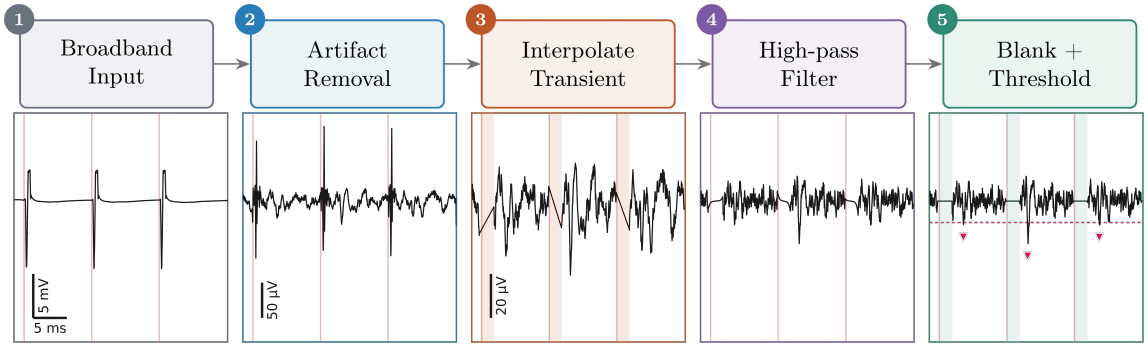

Figure 6.4. Standardized spike extraction from human ICMS recordings. One example channel is shown at successive processing stages, with stimulation pulse times marked by vertical lines. Note that the voltage scale changes across stages. (1) Raw broadband, dominated by the stimulation transient. (2) The chosen method, here DELETE, removes artifact. This is the only stage that varies across evaluated methods. (3) The pulse window (-0.1 to $+2$ ms, shaded) is interpolated unconditionally, as all methods fail to fully remove the stimulation transient at stronger artifacts. (4) A 750 Hz high-pass filter brings the data to spike band. (5) The previously interpolated window is widened by 0.2 ms and zeroed to suppress filter ringing, and spikes are detected as crossings (markers) of a $-4.5\times$ baseline RMS threshold (dashed).

We then apply a bidirectional 750 Hz high-pass Butterworth filter, as in prior work on this recording system [242] (stage 4). The previously interpolated interval is widened by 0.2 ms on both sides and zeroed to account for filter ringing. We compute a baseline activity RMS using pre-stimulation data, and extract spikes as crossings of $-4.5\times$ this baseline RMS (stage 5). Unless otherwise specified, we do not pre-filter channels for their firing rate or spikes for their quality.

6.2.3 Baseline artifact denoising methods

We compare DELETE against several popular baselines. To parse the space of existing software-based approaches for stimulation artifacts [243], we use two axes: the spatiotemporal context they draw on to estimate the artifact, and the data on which that estimate is fit. The simplest methods—blanking or linearly interpolating through the artifact [244, 242]—might best be thought of as data-free: they make no attempt to model the artifact, and recover no signal within the treated window. Next, filtering, polynomial, or exponential curve fits [245, 240, 246, 247] are estimated and applied on the immediate temporal context within each inter-pulse window. These local temporal fits have broad applicability, requiring no knowledge of electrode

arrangement and, beyond pulse timing, little of the stimulation protocol. More sophisticated methods assume and exploit further structure in the recording: spatial regression from simultaneously recorded channels [239], template averaging over repeated pulses [248, 249] or over the successive periods of a periodic artifact [250], a subspace of leading principal components taken across channels, pulses, and trials [251], or structured probabilistic models that fit the artifact's joint spatial, temporal, and amplitude regularities [252]. These recover more signal, but may become unwieldy when addressing a large variety of artifacts, for example from variable, multi-electrode stimulation. At the widest extreme, dictionary learning [253] abandons a fixed parametric form and learns artifact prototypes from the data itself, absorbing more irregular artifacts, though this method is still reliant on aligned artifact windows as input.

Our baselines are selected to span spatial and temporal approaches for both stimulation-free and stimulation-based data. As baseline spatial methods, we use common average reference (CAR), which subtracts the across-channel mean from each channel, and linear regression reference (LRR) [239], which regresses each channel's activity from all other channels at the same timepoint. CAR is only evaluated in simulations.

We consider polynomial [240] and spline curve fitting as pure temporal baselines. These methods fit the inter-pulse window in ICMS data with a polynomial or spline fit. In the benchmarks, we sweep the polynomial order and select the best per dataset, over even orders 8–16 on real ICMS data and orders 3–8 in simulation. The spline baseline is a cubic smoothing spline fit per channel to each post-blank inter-pulse window. Its smoothing parameter is set to $s = 0.001\,n$ for an n-sample window to allow RMS residual of about $32\,\mu$V—an allowance comparable to spike amplitude in our data. As a simple spatiotemporal baseline, we also evaluate applying polynomial fitting followed by LRR (Poly LRR). LRR fitting excludes the blanked transient intervals in stimulation data.

We also evaluate template subtraction [248, 249] and ERAASR [251], which are fit to a fixed tensor of aligned inter-pulse artifact windows. Template subtraction (Template in figures) models the artifact as the average activity in the inter-pulse window for all pulses of a fixed stimulation electrode and amplitude. ERAASR also exploits pulse-level stereotypy, but further models artifact across simultaneously recorded channels and trials. It estimates the artifact as the leading principal components

along each of these axes in turn and regresses them out, reconstructing each channel from the others so that local spiking is spared. In our implementation we use only the channel and pulse axes, as several of our datasets lack trial repetition. Note that their dependence on fixed inter-pulse frames precludes these two methods from applying to ICMS with irregular timing.

Baseline hyperparameters, namely polynomial order and ERAASR component count, are selected per dataset in the benchmarks. In simulation, we select on spike recovery, minimizing (1 − hit rate) + false-positive rate on the training split. For simplicity, we fix the polynomial order at 8 for the downstream analyses in Fig. 6.9. Due to computational expense, ERAASR is set to use 1 channel PC and 4 pulse PCs, as selected on a 20% subset of the full benchmark.

Conceptually, DELETE generalizes these baselines. It omits the assumptions about artifact smoothness or parametric form used by many baselines, but retains the assumption that artifacts and neural signal are distinguished by their variability and predictability. As a deep network, DELETE is likely behaviorally closest to dictionary learning [253], in keeping with interpretations of deep networks as interpolators over their training examples [254, 255], though we did not compare against it, as it shares the aligned-window requirement of template subtraction and ERAASR.

6.2.4 Evaluation metrics

In simulation, where ground truth neurophysiological signal is available, we report broadband deviation and spike hit rate. Both are scored over the same interval: the window between consecutive pulses (5 ms at 200 Hz), excluding the first 0.8 ms after each pulse. This exclusion is sufficient to span the acute stimulation transient and subsequent high-pass filter ringing in simulation, and marks the first sample either metric can meaningfully score. The deviation is the per-pulse mean absolute error between the ground-truth broadband (including evoked spikes and slower LFP) and the denoised output, reported as the mean paired difference across pulses (baseline minus DELETE). The hit rate is the fraction of ground-truth spikes recovered, matching detected spike troughs one-to-one to ground-truth spike times within ±0.1 ms. The false-positive rate is the fraction of detected spikes without a ground-truth match. Both rates are reported against distance from the stimulating electrode.

We quantify single-channel spike isolation with a signal-to-noise ratio (SNR): for

each channel, the signal is the mean spike peak (the most negative filtered voltage within ± 0.4 ms of each detected threshold crossing) and the noise is the standard deviation of the filtered voltage over the inter-spike baseline (crossing windows excluded); SNR is the absolute value of the ratio. For concurrent-stimulation recordings, SNR is computed separately in a pre-stimulation quiet window and the peri-stimulation window (first to last pulse), giving baseline and peri-stimulation SNR.

On real ICMS recordings, where evoked activity has no ground truth, we report four heuristic metrics, illustrated in the ICMS evaluation (Fig. 6.8A–D). The two metrics with per-channel calculations (RMS Inflation and Waveform R^2) are averaged across channels on the stimulated array, excluding stimulating electrodes.

RMS Inflation. The ratio of peri-stimulation to pre-stimulation signal RMS after artifact rejection; ratios far above 1 imply residual artifact that triggers false threshold crossings.

Waveform R^2. Measures whether spikes detected during stimulation still resemble the units seen in quiet (pre-stimulation) periods. For each channel, we cluster its pre-stimulation spikes into a small bank of waveform templates (up to three cluster-median templates via k-means), then match every peri-stimulation spike to its best-fitting template over a ± 0.1 ms time shift and score the match by the coefficient of determination, R^2. The channel-level Waveform R^2 is the mean R^2 across its peri-stimulation spikes. Channels with fewer than two pre-stimulation or two peri-stimulation snippets, or otherwise lacking a valid score, contribute zero to the array average.

Activity Synchrony. Counts samples in which simultaneous threshold crossings occur on more than 25% of the array, reported as synchrony events per second (syncs/s). This conservatively high, single-sample criterion is intended to identify array-wide contamination, such as sudden movement, line noise, or insufficiently mitigated stimulation artifact. Synchronous ICMS-evoked activity could trip this metric, as antidromic spike activation can exhibit less than 0.1 ms (3 samples at 30 kHz) of latency jitter [256], but we see this as unlikely in our setting. Sombeck et al. [247] infer that most spikes observed after 0.7 ms are likely transsynaptic, and further measure that evoked spike jitter increases with latency from the stimulation pulse, to >0.1 ms by 1 ms, well within our blanking period of 2.2 ms. To check sensitivity to the 25% threshold, we show that the metric's qualitative ranking

holds across a range of thresholds in Section D.1.3.

Injection Recall. Injects observed spike waveforms into inter-pulse intervals lacking detected spikes, re-runs denoising, and measures the fraction recovered versus time from pulse onset. We summarize recall as the pooled fraction recovered for injection offsets from 0.6 to 7 ms after pulse onset, shortening this interval as needed for closely spaced pulses. As injected spikes may be low-amplitude and inherently unrecoverable when injected into noisy background, we normalize by recall in non-stimulation periods. We show that recall rapidly increases when injecting synthetic spikes of moderately large amplitude in Section D.1.2.

We emphasize that these metrics can only partially assess denoising performance as they are not based on external truth. For example, RMS Inflation and Activity Synchrony are normative about the expected neurophysiological response. RMS Inflation may legitimately exceed 1 when stimulation evokes sufficiently large spiking or field-potential responses, and genuinely synchronized responses could trigger Activity Synchrony. Thus, both metrics can misclassify physiological activity as artifact. Inversely, instead of rating neural response poorly, waveform R^2 and injection recall can score artifact well. As shown in Fig. 6.8B, Waveform R^2 can remain high when a method preserves line-noise crossings in both periods. Injection Recall can likewise be biased upward for noisy methods because spikes are inserted only into intervals without existing detections, potentially selecting unusually easy inter-pulse intervals. These limitations are inherent to real-world evaluation without ground truth, so conclusions must be made holistically.

6.2.5 Neurometric threshold estimation

To evaluate denoising against a downstream readout of evoked activity, we estimate a neural detection threshold from multiple trials of denoised recordings and compare it to the participant's behavioral threshold. The comparison follows the neurometric calculation of Britten et al. [257], applied to ICMS detection as in Grani et al. [80].

Per-trial neural response. For each experiment and each denoising method, we extract threshold crossings with the pipeline of Section 6.2.2 and compute per-channel firing rates in a 300 ms window beginning at the first pulse of the train, with corresponding windows in no-stimulation intervals. Rates during ICMS are corrected

for blanking by dividing by the fraction of the measurement interval that remains unblanked. For each trial and channel, we subtract the firing rate in the 100 ms immediately preceding the corresponding measurement window. We then z-score these baseline-subtracted responses using each channel's mean and standard deviation pooled across stimulation and no-stimulation intervals, separately for each denoising method. As in Grani et al. [80], we observe decaying modulation with distance to the stimulating electrode, but the exact spatial pattern and extent vary across stimulating electrodes. As a simple summary feature, we average these z-scored responses across channels within a Manhattan distance of 4 from the stimulating electrode, excluding its own channel. To prevent dilution by unresponsive channels, we drop channels whose mean unsubtracted rate over the two highest tested amplitudes falls at or below 2 Hz. This selection is performed separately for each denoising method.

Neurometric threshold. Grani et al. [80] define a neurometric threshold from the neural response curve itself, fitting a sigmoid to response against amplitude and taking its half-maximal point. That construction suits our data less well, as amplitude ranges vary more across stimulating electrodes, and our responses often do not saturate within the delivered range, leaving the sigmoid's half-maximal point poorly constrained. We therefore define the threshold by discriminability instead, as in Britten et al. [257].

With sufficient repeats at each amplitude, discriminability can be estimated directly using the empirical AUROC, which gives the probability that our neural summary feature is higher during stimulation than baseline. However, our dataset has uneven trial counts, with some amplitudes tested only once or twice, making these per-amplitude estimates imprecise.

We therefore use a parametric AUROC, fitting the individual trial responses with a log-sigmoidal amplitude-response curve. Evidence on a stimulation trial at amplitude a is modeled as Gaussian with mean $\mu_0 + f(a)$ and standard deviation $\sigma_s(a)$, against no-stimulation trials with mean μ_0 and standard deviation σ_0. This AUROC reference is drawn from the non-target intervals of detection trials together with no-stimulation catch trials from passive data. The mean evoked response follows

$$f(a) = \beta \cdot \operatorname{expit}\left(\frac{\log a - a_0}{\tau}\right),$$

Here, β is the maximum mean response above the no-stimulation reference, a_0 is

the midpoint on the log-amplitude axis, and τ sets the transition width. We allow variance to scale with stimulation amplitude by setting $\sigma_s(a) = \sigma_{s0} + \kappa f(a)$, where σ_{s0} is the stimulated-response standard deviation at zero current and κ sets its increase with mean response. We estimate all parameters jointly by maximizing the Gaussian likelihood of individual stimulation and no-stimulation trials across amplitudes. Both the log-amplitude and amplitude-dependent variance were used as they modestly improved the fit to empirical response.

Under this model the discriminability at any amplitude follows in closed form,

$$\mathrm{AUROC}(a) = \Phi\left(\frac{f(a)}{\sqrt{\sigma_s(a)^2 + \sigma_0^2}}\right),$$

where Φ is the standard normal cumulative distribution function. The neurometric threshold $\theta(c)$ is the smallest amplitude at which AUROC reaches criterion c; Fig. D.5A illustrates the construction for one stimulating electrode. Because the denominator itself depends on f, we locate θ numerically by bisection. We use $c = 0.75$, matching the criterion of the psychometric fit below. Thresholds above the highest delivered current are clipped to it; those below the lowest are retained without clipping.

As a sanity check, we compared our parametric fit against a Weibull curve fit to empirical AUROCs at each amplitude by inverse-variance-weighted least squares, constrained to chance (0.5) at zero current. Our parametric estimate yields slightly higher thresholds than this empirical estimate (Fig. D.5B).

Psychometric threshold. From the concurrent 2AFC report we fit the probability of choosing the stimulated interval against amplitude with a two-parameter logistic by maximum likelihood, floored at chance. The psychometric threshold is the curve's midpoint, equivalently the 75%-correct point.

6.3 Results

We design our experiments primarily to assess the flexible utility of DELETE, but also to provide intuition for its behavior as a relatively black-box method. In Section 6.3.1's simulations, we show that DELETE recovers linear optima under spatial noise and provides competitive performance in varied ICMS settings. We then use simulation ground truth to characterize key hyperparameters before examining

practical training and deployment on real recordings in Section 6.3.2. Section 6.3.3 assesses DELETE's potential to degrade artifact-free datasets by turning to quality-controlled stimulation-free recordings. There, we show that DELETE largely preserves spike waveforms, timing and firing rates on high-SNR channels, and maintains motor decoding performance. We then reach challenging human ICMS datasets in Section 6.3.4. As ground truth is no longer available, we provide a battery of heuristic metrics to show DELETE is generally competitive across easy and challenging artifact conditions. Finally, in Section 6.3.5, we move beyond heuristic metrics by comparing the activity recovered by each denoising method with stimulation amplitude and ICMS detection behavior.

6.3.1 DELETE matches the best baseline across artifact simulations

Spatial artifacts

Spatial artifacts—shared, cross-channel structure from sources like motion or reference drift—are commonly mitigated in multielectrode recordings by virtual re-referencing. We begin with a minimal sanity check that DELETE should match these strategies by extending the simulation setup from Musial et al. [239]. This simulation generates 64-channel array responses by adding multiple sources of bandlimited noise and templated neural spike waveforms. Raw and cleaned waveforms extracted from the known timing in these simulated data are shown in Fig. 6.5A. Common average reference (CAR) is only modestly helpful as different channels reflect different mixtures of noise sources, but linear regression reference (LRR) [239] cleanly recovers individual spike waveforms. DELETE performs similarly, providing a sanity check that its Transformer-based denoising formulation still behaviorally resembles a familiar linear regression method.

DELETE's nonlinear capacity, however, should enable it to model multiple artifact regimes *in one model*. Fig. 6.5B assesses this claim by comparing DELETE against CAR and LRR on two simulation draws (datasets A and B), where the linear combinations of shared noise sources are resampled. Like LRR, DELETE fails on unseen data, falling a little below trivial averaging behavior (Train Dataset A, Eval Dataset B). However, when trained on both datasets, DELETE achieves near-optimal performance on each, putatively inferring the dataset context and deploying the correct denoising rule, while LRR lacks such flexibility.

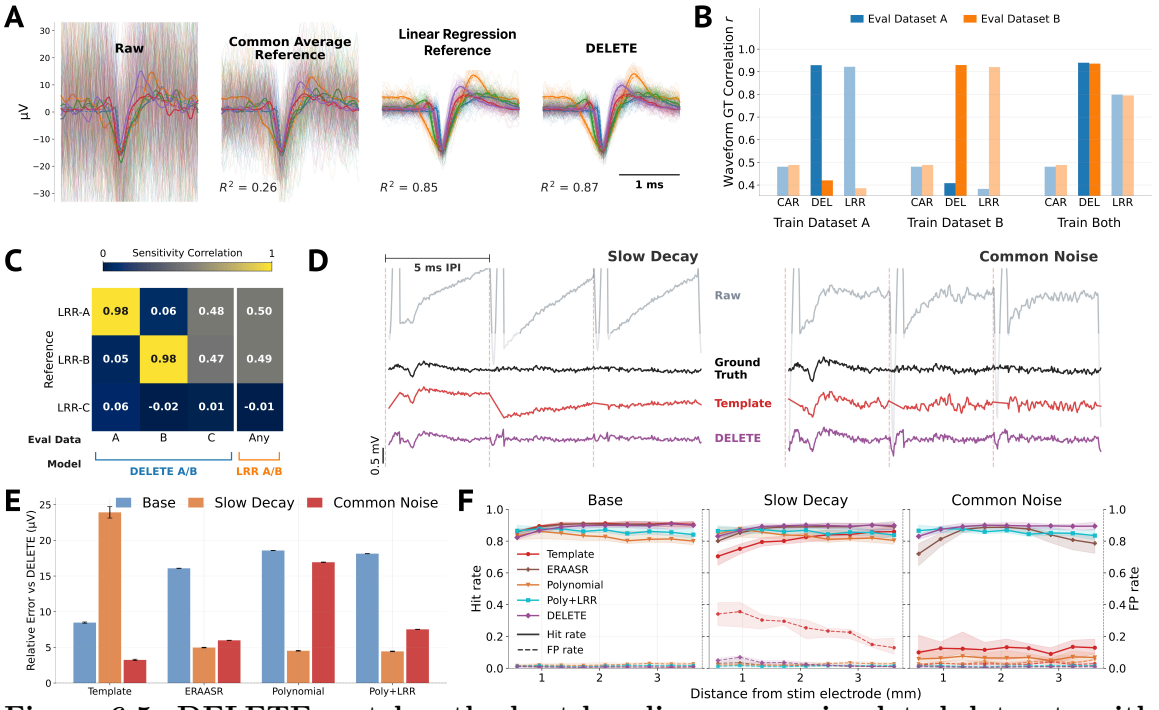

Figure 6.5. DELETE matches the best baseline across simulated datasets with common noise and stimulation artifacts. (A) Spike waveforms in a linear simulation [239], denoised by common average reference (CAR), linear regression reference (LRR), and DELETE. Color indicates spike identity, and spike-triggered averages are shown in bold. DELETE and LRR recover clear spike waveforms, as measured by waveform R^2 against the ground truth. (B) We prepare models with two separate simulated datasets and report waveform–ground-truth correlation for models trained on one or both datasets. One DELETE model trained on both regimes recovers near-optimal performance on each, whereas a single LRR degrades when compromising across regimes. (C) We compare the input–output sensitivity of DELETE and LRR models trained jointly on datasets A and B against the sensitivity of reference LRR filters fit separately on datasets A, B, and a third simulation C. The DELETE model behaves as LRR when applied to dataset A or B, and as the joint LRR when in the third regime C. (D) ICMS simulation traces from two datasets adapted from Wang et al. [240]. Slow Decay accumulates DC offset across pulses, while Common Noise contains channel-shared noise on top of the stimulation artifact. We show the raw data, ground-truth, Template subtraction, and DELETE traces. Raw data are partially lightened for clarity. IPI denotes inter-pulse interval; red dashed lines mark stimulation pulses. (E) Mean paired difference in per-pulse deviation (mean absolute error against the ground-truth broadband), computed as baseline minus DELETE, across the Base, Slow Decay, and Common Noise regimes; positive values indicate larger error than DELETE. Error bars indicate the standard error of the mean paired difference across pulses. (F) Spike hit rate (solid) and false-positive rate (dashed) versus distance from the stimulation electrode, per method, across three ICMS simulations.

To build this interpretation of DELETE as a piecewise linear model, we measure model input–output *sensitivities*. That is, we measure the 64×64 matrix describing how much each channel's prediction changes with respect to any other channel's input. Fig. 6.5C correlates the flattened sensitivity matrices extracted from DELETE and LRR models fit to datasets A and B. DELETE has multiple columns because it behaves differently on datasets A, B, and a third simulation draw C. The highly correlated diagonal for A and B implies that near those datasets, DELETE's best linear approximation is essentially the purpose-fit LRR. Notably, however, when in an unknown third regime C, DELETE relaxes to behaving as the joint LRR, as the DELETE C and LRR columns match. These simulation analyses show that DELETE can behave as a flexible ensemble of LRR models, deploying them selectively based on the data context. We do not intend this comparison to suggest that LRR is typically trained on multiple datasets; in practice, LRR is often updated continuously to avoid unknown regimes, and DELETE should be applied similarly. Rather, we aim to demonstrate that DELETE can model multiple settings at once, which may be useful for data efficiency and generalization when noise profiles change rapidly, as with novel stimulation parameters.

Stimulation artifacts

Stimulation artifacts have exploitable spatial and temporal structure, yet remain challenging because they vary across and within experiments. We extend the ICMS simulation of Wang et al. [240] with two variants that challenge different artifact removal strategies (Fig. 6.5D). In Slow Decay, artifact persists across pulses, producing accumulating offsets that strain template subtraction. Common Noise adds a noise source shared across channels, challenging single-channel curve fitting. We compare DELETE against template subtraction, ERAASR, polynomial curve fitting, and polynomial fitting combined with LRR, using broadband deviation and spike recovery to assess performance.

DELETE achieves the lowest deviation, or absolute error, against ground truth, in all three simulations (Fig. 6.5E). Other methods have mixed performance across settings. We can readily observe that Template fails on Slow Decay, where the offset accumulating across pulses breaks the temporal stereotypy templating relies on. We also see that Polynomial curve fitting fails dramatically on Common Noise, because the added fluctuations are poorly captured by a smooth polynomial fit. In contrast, the

spatiotemporal methods ERAASR and Poly LRR perform more consistently, though still behind DELETE. Further, all methods have a steady performance gap against DELETE on the Base dataset. Interpreting DELETE's superior performance precisely is difficult, as the deviation metric is heavily influenced by slow LFP activity.

Downstream analysis of microelectrode recordings typically discards this band in favor of spiking activity, so we turn to spike recovery in Fig. 6.5F. DELETE again performs consistently well. Poly LRR lags only slightly behind DELETE in all three settings. ERAASR lags slightly in Common Noise. We interpret this as due to the fact that although ERAASR models shared artifact across channels, the common noise is low amplitude relative to the stimulation artifact, so it falls outside the leading principal components that ERAASR removes. Template subtraction also has a saliently high false-positive rate and a low hit rate on Slow Decay, due to inter-pulse artifact variability. These ICMS simulations support the idea that accounting for multiple noise settings requires spatiotemporal modeling, and that DELETE accomplishes this most effectively among the tested baselines.

6.3.2 Hyperparameter sensitivity, training, and deployment

Having established DELETE's performance in simulation, we use the same ground-truth setting to characterize two key hyperparameters before turning to training and deployment on real recordings.

In Fig. 6.6A, we vary the temporal context window and see that spike recovery generally improves with longer context. We stop at 6 ms, where DELETE can match stable baselines without yet imposing an intolerable latency for potential online ICMS control. Shorter-context models ultimately achieve lower training loss but have worse spike recovery, underscoring the need to consider downstream metrics when hyperparameter tuning. In Fig. 6.6B, we vary the amplitude of noise injected during training in a simulation where ground-truth spike amplitudes are known. Recovered spike amplitude varies systematically with this hyperparameter, so it should be tuned to the expected spike scale. The simulated spikes are large (up to 500 μV), whereas many spikes in our human recordings have amplitudes below 50 μV. Because ground truth is unavailable in real recordings, we heuristically set $\sigma = 100\,\mu$V for real-data models.

Until now, we have considered DELETE models trained from scratch on individual

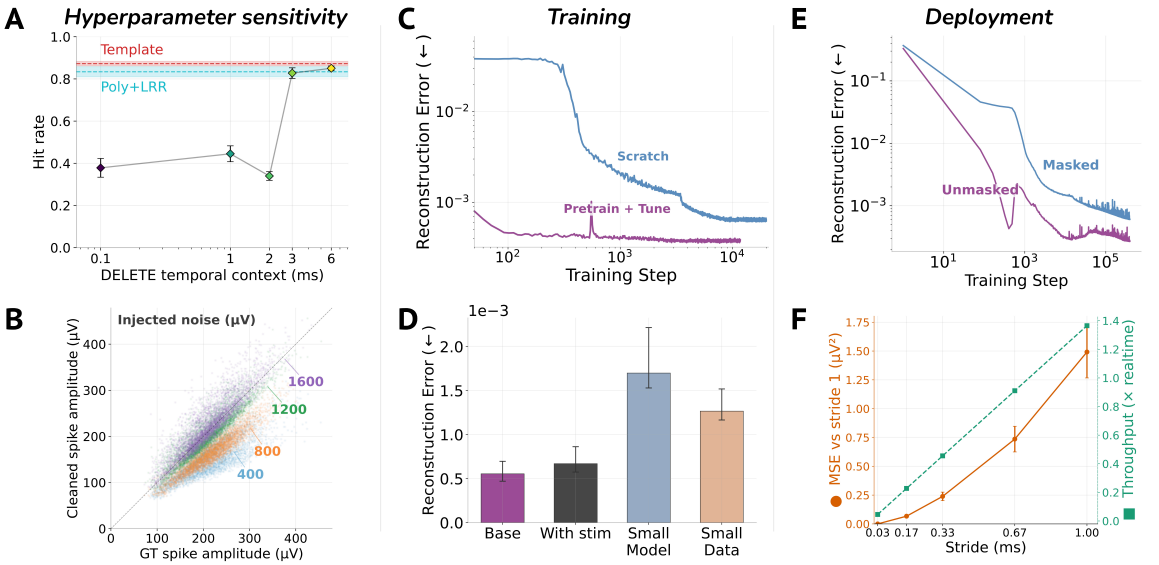

Figure 6.6. DELETE hyperparameter sensitivity, training, and deployment. Columns characterize hyperparameter sensitivity using simulation ground truth (A,B), training and adaptation on real recordings (C,D), and deployment (E,F). (A) Temporal-context sensitivity: spike hit rate against DELETE's temporal context (0.1–6 ms) in the ICMS Base simulation, with Template and Poly LRR shown for reference. Error bars show standard deviations across simulation trials. (B) Training-noise sensitivity: recovered spike amplitude against ground truth in the Slow Decay simulation for DELETE models without explicit stimulation inputs, trained with four injected-noise amplitudes ($\sigma = 400$–$1600\,\mu$V). (C) Evaluation curves across training: a pretrained-and-fine-tuned model (Pretrain+Tune) reaches lower reconstruction error faster than training from scratch. (D) Design ablations. Base is our DELETE model used on real recordings: an 11-million-parameter model pretrained on 18 hours of ICMS broadband data, without explicit stimulation parameters as input. With stim includes stimulation parameters, Small Model uses 0.6 million parameters, and Small Data uses 1 hour of ICMS broadband data. Error bars are 95% bootstrap confidence intervals across 131,072 held-out evaluation examples per variant. (E) Masked (training) versus unmasked (deployment) error over training. (F) Inference-stride tradeoff between output deviation and throughput relative to real time. Deviation is the MSE against one-sample-stride outputs on pre-stimulation segments from three datasets; throughput is measured on an NVIDIA RTX 4090.

simulation datasets. To amortize the training cost for repeated application to human ICMS recordings, we instead pretrain on broadband data from three participants (P2–P4) and fine-tune to each target dataset. As expected, on a new exemplar ICMS dataset, fine-tuning DELETE reaches lower reconstruction error in fewer training steps than training from scratch (Fig. 6.6C). Note that while our pretraining spans multiple participants, we did not provide the model with any identifying annotations.

In Fig. 6.6D, we review basic pretraining ablations of model size, pretraining-data size, and stimulation input. The Base configuration uses an 11-million-parameter model pretrained on 18 hours of ICMS data, without explicit stimulation parameters as input. Reducing the model to 0.6 million parameters or the pretraining data to 1 hour increases reconstruction error, but offers substantially lighter training configurations. Including explicit stimulation parameters (With stim) does not improve reconstruction error. Accordingly, we omit stimulation parameters from our models.

With the training configuration established, we next examine deployment on unmasked data (Fig. 6.2B). DELETE is trained with perturbed inputs, so applying it directly to unperturbed recordings introduces a mismatch between training and deployment. A similar training–deployment mismatch occurs in masked language models [130], which are also trained on masked inputs but often applied directly to unmasked inputs. This discrepancy is tolerable, as Fig. 6.6E shows that DELETE's unmasked behavior is also "trained" by the main objective, though its error curve is more complex than the steadier decline of the masked error. We also confirm that, by the end of training, applying DELETE unmasked produces similar outputs to the training-faithful strategy of tiling the full input with random subsets and passing each to DELETE.

Finally, real-time deployment requires selecting an inference stride, which determines DELETE's computational load and output cadence. Strides shorter than the 6 ms context window yield overlapping predictions for each sample, which are averaged at the cost of throughput. In Fig. 6.6F, output deviation from one-sample-stride inference grows gradually as stride increases, while a 1 ms stride achieves real-time throughput. For our analyses, we use a modest intermediate stride of 0.33 ms, balancing output agreement and computational cost.

These analyses fix the configuration used in all real-data evaluations that follow: a 6 ms context, $\sigma = 100\,\mu$V injected noise, pretraining with per-dataset fine-tuning, and a 0.33 ms inference stride.

6.3.3 DELETE preserves artifact-free recordings

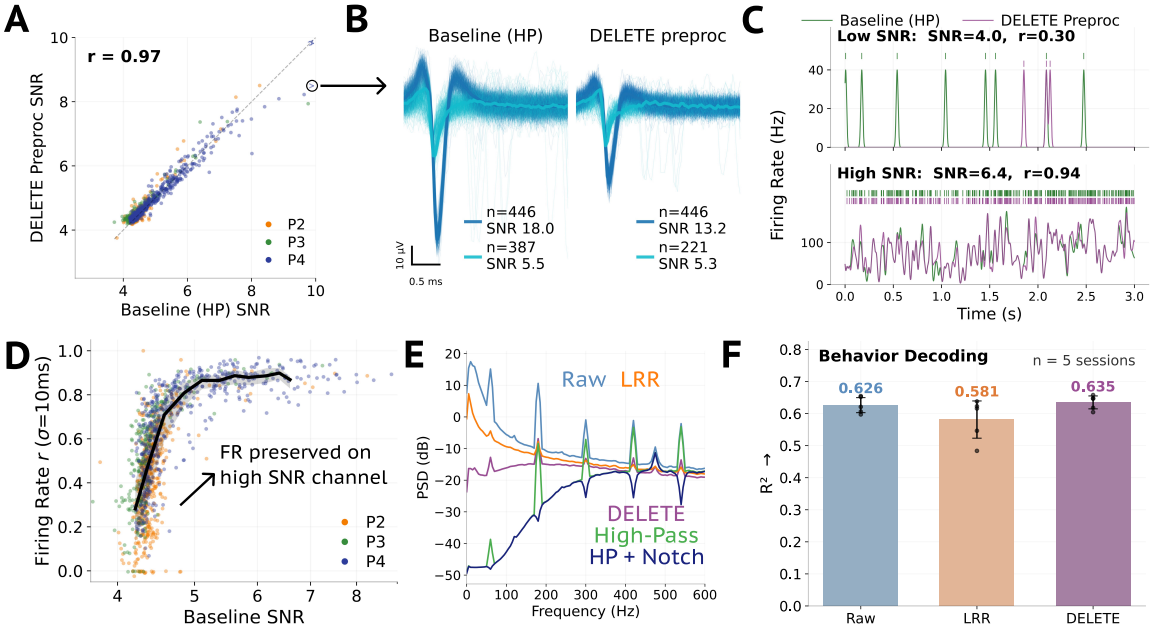

Figure 6.7. DELETE preserves single-channel, population, and behavioral content of non-stimulation electrophysiology. (A) Per-channel spike SNR with or without DELETE preprocessing across three human BCI participants (P2–P4); SNR is preserved ($r = 0.97$). (B) Spike waveforms for the selected outlier channel from (A) under baseline preprocessing and DELETE. DELETE largely retains waveform cluster identifiability but shrinks amplitudes. Small waveform hash can be lost, while large waveforms are preserved in count but can have diminished SNR. (C) Firing-rate traces and correlations for baseline vs DELETE preprocessing for a representative low-SNR (SNR $= 4.0$, $r = 0.30$) and high-SNR channel (SNR $= 6.4$, $r = 0.94$). The ticks at the top of each panel indicate spike crossing times. DELETE preserves the timing of activity on high-quality channels. (D) Firing-rate correlation ($\sigma = 10$ ms) between baseline and DELETE preprocessing compared with channel SNR; agreement saturates at ~ 0.9 for high-SNR channels, with disagreement confined to low-SNR (noise-dominated) channels. The dark line traces the median firing-rate correlation across channels within each baseline-SNR bin (bin width 0.25, pooled over participants; bins with fewer than 10 channels omitted). (E) Power spectral density of Raw, LRR, DELETE, 300 Hz Butterworth High-Pass, and HP+Notch processing. DELETE can be interpreted as a light high-pass combined with LRR spatial filtering. (F) Motor behavior decoding R^2 (O'Doherty RTT [151]; $n = 5$ sessions) under Raw, LRR, and DELETE preprocessing; all conditions share a band-pass filter and threshold-crossing spike detection. Bars show mean decoding R^2; error bars show standard deviation, and points show individual sessions.

DELETE assumes that artifact is the immediately and easily predictable content of broadband data, but whether acting on this premise also hallucinates or degrades neural data is an empirical question. To be thorough, we begin our analysis of real

neural data denoising by evaluating DELETE's impact on relatively clean, artifact-free recordings.

We demonstrate this with human baseline recordings and motor decoding datasets in Fig. 6.7. We begin by examining DELETE's impact on spikes extracted from each channel of these baseline recordings. We process the data either with a standard Butterworth high-pass filter alone, or with DELETE followed by the same filter. In Fig. 6.7A, we observe that the per-channel spike SNR is largely preserved even with DELETE preprocessing. However, preserved SNR does not establish that spike waveforms and counts are unchanged. We therefore examine an outlier channel with two distinct units in Fig. 6.7B. Here, DELETE clearly shrinks the amplitudes of both units, even if on most channels the noise level is proportionally reduced to preserve SNR. Although crossing counts are preserved exactly for the larger unit, this shrinking causes us to lose many crossings for the near-hash unit. This attenuation is consistent with the dependence of recovered spike amplitude on training-noise amplitude (Fig. 6.6B). It is not clear whether the loss of such low-quality spikes is significant for most applications.

Preserved waveform shape and SNR also leave open whether DELETE preserves the timing of detected activity. In Fig. 6.7C, we show the firing rate traces for a high-SNR and low-SNR channel before and after DELETE preprocessing. On the low-SNR channel, crossing count is also low, and FR correlation is low because DELETE does not preserve the crossings. For the high-SNR, high-firing channel, however, crossing times are largely preserved. We summarize this in Fig. 6.7D by plotting the FR correlation versus channel SNR. Consistent with the waveform analysis, firing rates are preserved on high-SNR channels, with disagreement concentrated among low-SNR channels.

DELETE is applied continuously through time and thus can be viewed as a filter itself. To give a spectral characterization for DELETE, we compare its power spectral density to reference methods on an example baseline recording in Fig. 6.7E. The filter references use a second-order 300 Hz Butterworth high-pass filter, alone or followed by notch filters at odd harmonics of 60 Hz. Compared to the high-pass filter alone, DELETE has a much softer rolloff, and appears to have a "cutoff" at $\sim$ 300 Hz, consistent with DELETE's short (6 ms) context window. The same recording also contains visible line noise at odd harmonics of 60 Hz. It is tempting to consider DELETE's mitigation of this line noise as evidence that it behaves as a combination

HP and notch filter, but we see that its suppression is very similar to that of LRR applied to the same data. Consistent with its similarity to LRR established in simulation (Section 6.3.1), DELETE behaves spectrally like a light high-pass combined with LRR-style spatial filtering, removing shared or low-frequency structure and line noise while preserving the spike band.

DELETE's spatial filtering, prominent in its spectral characterization, raises the possibility that we attenuate cross-correlated channels that fire together. We assess this in a controlled simulation in Fig. D.1. There, we conclude that extremely synchronized channels will be erased by DELETE, but slight stochasticity reduces this effect. These results identify a potential cost of spatial filtering. To test whether DELETE nevertheless preserves information useful for behavior decoding, we decode monkey reach velocity in Fig. 6.7F. Specifically, we compare three preprocessing conditions—Raw (no denoiser), LRR, and DELETE—each followed by the same high-pass filter and threshold-crossing spike detection. LRR appears to occasionally degrade decoding performance, but DELETE does not.

These analyses together show that, across single channel and population metrics, in space and time, DELETE provides light-touch preprocessing that preserves the vital content of non-artifacted recordings.

6.3.4 Evaluation of spike recovery in real-world ICMS datasets

Having characterized DELETE's flexibility and reliability in simulation and baseline datasets, we now evaluate DELETE on concurrent ICMS recordings. Instead of simulating different artifact distributions, here we consider multiple ICMS protocols and participants, reflecting challenging, real-world variability. The three protocols, defined in Section 6.2.2 and shown in Fig. 6.3, span increasing artifact difficulty. Single channel Fixed, the most common protocol for human BCI applications, induces the simplest artifact profile: its peri-pulse average shows low variability across pulses, and the acute artifact ends by 1 ms. In RAP, inter-pulse timings are variable and potentially very short (4 ms), accentuating nonlinear amplifier regimes. The resulting traces show increased peri-pulse variability and do not relax to baseline, even on non-stimulating electrodes. Multichannel Fixed contains the most severe artifacts, from frequent amplifier saturation and hardware jitter.

On real-world data, we lack ground truth for evoked activity, so we rely on four

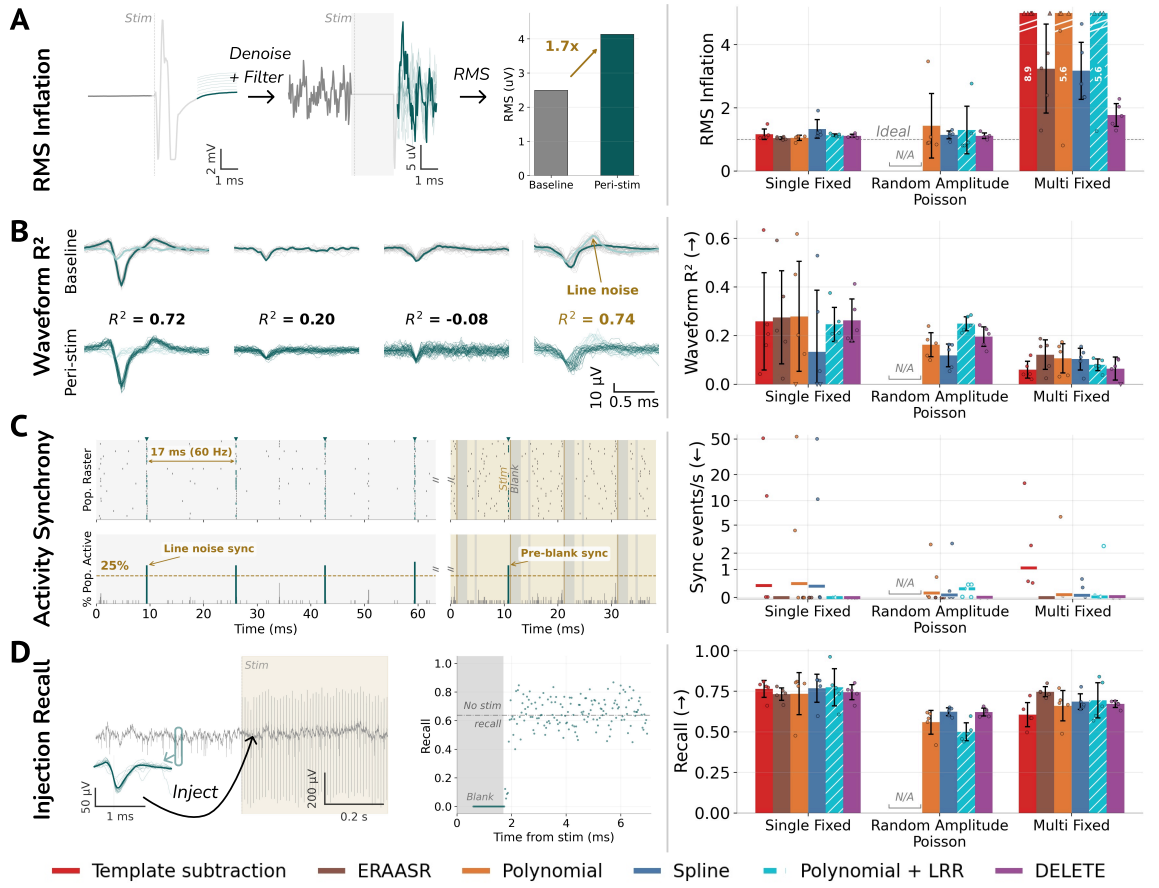

Figure 6.8. DELETE recovers spiking activity with short latency after stimulation across increasingly challenging real ICMS datasets. Each row pairs a metric's definition (left) with method performance across the three protocols (right). Points show performance on individual datasets. In A, B, and D, bars show the mean across datasets and error bars show one standard deviation; in C, horizontal lines show the median. (A) *RMS Inflation*: peri-stimulation signal RMS relative to a pre-stimulation baseline after denoising, filtering, and blanking. The gray span marks the fixed interval blanked around each stimulation pulse. Peri-stimulation inflation is measured post-blank; the example channel shows 1.7× peri-stimulation inflation over baseline. (B) *Waveform* R^2: peri-stimulation spike waveforms compared to pre-stimulation baseline templates, with four example channels spanning high to failed recovery. The rightmost example shows a metric failure mode, as correlation can be high if noise enters the pre-stimulation waveform pool. (C) *Activity Synchrony*: synchronous population events per second. Synchrony is marked when more than 25% of the 32-channel array is active in the same 0.03 ms sample. The examples show population rasters (top) and the fraction of active channels per sample (bottom): periodic line-noise crossings produce synchronous events at 17 ms intervals (left), while stimulation-locked crossings produce synchronous events immediately before the blanking windows (right). Sync rates are plotted on a log-scaled axis, with zero at the axis floor. (D) *Injection Recall*: known spikes are injected into the peri-stimulation window and recovery is measured in a second pipeline pass. Raw recall saturates at a ceiling < 1.0 as we inject small spikes into noisy background.

heuristic metrics defined in Section 6.2.4 and illustrated in Fig. 6.8A–D: RMS Inflation, Waveform R^2, Activity Synchrony, and Injection Recall. We report the performance of DELETE against the baselines defined in Section 6.2.3. Template subtraction and ERAASR are omitted on the RAP data as they require a fixed inter-pulse window to estimate artifact.

By RMS Inflation, DELETE performs consistently across protocols, with its clearest advantage on Multichannel Fixed data. All methods return near baseline RMS on Single channel Fixed data, while on Multichannel Fixed data most baselines leave substantial residual energy, with Template subtraction, Polynomial, and Poly LRR all exceeding the plotted range. ERAASR and splines leave less, and DELETE the least, though none returns fully to baseline. This is partially by design: DELETE is trained to output injected noise of limited amplitude. Nonetheless, the large residual RMS left by several baselines indicates substantial potential contamination of downstream spike extraction.

Waveform R^2 provides a more mixed assessment. DELETE scores comparably to existing methods on Single channel Fixed data, but trails some alternatives on RAP and Multichannel Fixed data. Individual datasets show larger discrepancies, which we examine in Section D.1.4. These examples show that lower waveform agreement can reflect contamination of the baseline waveforms themselves. We therefore regard waveform R^2 as a limited measure to be interpreted alongside the other metrics.

Activity Synchrony most clearly distinguishes DELETE's consistency across datasets. ERAASR and Poly LRR also achieve low synchrony on most datasets, while Template subtraction, Polynomial, and Spline show occasional large failures, indicating the value of using cross-channel context to remove shared contamination. Yet among these spatial methods, DELETE alone maintains near-zero synchrony across all three protocols (Fig. 6.8C).

Injection Recall is a relatively steady metric without drastic outliers. Most methods perform similarly, with Template and Polynomial lagging on the Multichannel Fixed data and DELETE and Splines showing a steady advantage on RAP data. DELETE sits within the range of the baselines on Single channel Fixed data. The differences between methods are smaller than those reported in similar analyses by Negri et al. [246], Wang et al. [240]. We attribute this to the standard blanking in our pipeline, which uniformly suppresses recall for all methods in the immediate post-pulse interval.

DELETE's central advantage is its consistent performance across datasets, artifact

profiles, and metrics. It generally matches the baselines on the simplest Single channel Fixed protocol while remaining effective under the more challenging RAP and multichannel artifact conditions. DELETE also gracefully integrates spatial context, at least as well as a sequential LRR stage appended to Polynomial fitting. Nonetheless, these comparisons still leave uncertainty about true denoising performance because metrics are not grounded in external data. We therefore turn to a downstream task, relating the recovered neural responses to ICMS detection behavior.

6.3.5 ICMS detection threshold correlation validates DELETE denoising

ICMS in sensory cortex elicits perceptual responses, with a local activation mechanism [258] that provides a unique opportunity to relate recovered neural activity to perceptual reports. This approach has its own challenges: artifact can covary with stimulation conditions that also affect behavior, and the local neural response need not fully account for the perceptual report. However, recent work has built the case that the local, intra-ICMS response in visual cortex does correlate with perceptual decisions [80], encouraging our analogous analysis with somatosensory ICMS.

We thus collected detection datasets across stimulating electrodes and participants, which use the simplest protocol, with amplitude and frequency fixed within each train (Fig. 6.9A top). We supplemented some datasets with passive stimulation, which putatively evokes a similar neural response as in the active detection task, but takes less time and participant effort (Fig. 6.9A bottom). This passive paradigm unfortunately reintroduces the secondary hardware transient which complicated the RAP and Multichannel Fixed datasets. While the secondary transient likely degrades potential downstream analysis, we treat it as representative of real-world nuisances that flexible artifact mitigation methods should overcome.

We illustrate how DELETE does overcome this secondary transient more gracefully than a polynomial estimator in Fig. 6.9B and C. The Detection raster plot shows that both estimators recover similar peri-pulse responses in the simpler single blank setting. Both estimators show increasing activity with stimulation amplitude, with reliable responses shortly after the blanking window by 50 μA. The shown Passive dataset sweeps a wider set of amplitudes, but only DELETE preserves this strong response despite the secondary transient, with the estimators agreeing again only

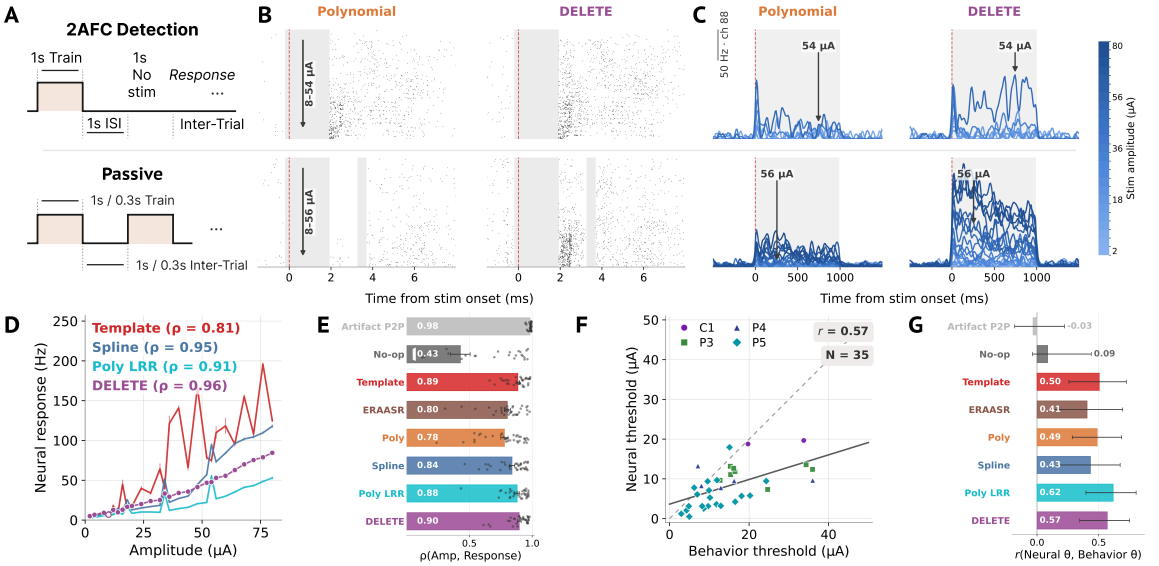

Figure 6.9. Denoising evoked activity from ICMS psychometric detection. (A) Stimulus designs. In the 2AFC detection task, a 1 s ICMS train is delivered in one of two 1 s intervals separated by a 1 s inter-stimulus interval, and the participant reports which interval contained the stimulus. Passive blocks deliver trains without a report, with 1 s or 0.3 s trains and matched inter-trial intervals. (B) Peri-pulse rasters of detected threshold crossings on the stimulated array, aligned to pulse onset (red dashed) and sorted by stimulation amplitude, under Polynomial and DELETE denoising (columns) for a detection and a passive dataset (rows). Stimulation amplitudes span 8–54 μA in the detection example and 2–80 μA in the passive example; raster labels indicate range endpoints rather than individual tested levels (see Section 6.2.2 for amplitude sampling). Gray spans mark blanked intervals: the fixed post-pulse window and, in the passive protocol, the secondary hardware transient at 3.3 ms arising from the firmware it uses. (C) Trial- and population-averaged firing rate over the stimulation train for the same datasets and denoisers, colored by stimulation amplitude. (D) Population response against stimulation amplitude for one dataset under four of the six denoisers (ERAASR and Polynomial omitted for legibility), each annotated with rank correlation ρ between amplitude and response. Given the variable length trains in our protocol (0.3 s and 1 s), we only analyzed the first 0.3 s from all trials. (E) The same amplitude–response correlation for every denoising method and two controls across 35 electrode-days. Artifact P2P uses raw peak-to-peak artifact voltage; No-op uses blanking, high-pass filtering, and threshold crossing without an artifact model. Bars show the mean over electrode-days with standard error, and points are individual electrode-days. Arrows mark electrode-days below the broken axis. (F) Neurometric vs psychometric threshold under DELETE, one point per electrode-day ($n = 35$), with marker shape and color denoting participant. (G) Correlation between neurometric and behavioral threshold for each denoising method and the two controls in E, that is, the relationship in panel F recomputed per method. Error bars are bootstrap confidence intervals across electrode-days; the denoiser scatters are shown in Fig. D.5C.

after about 5 ms post-pulse. The polynomial estimator, forced to fit the ∼1 ms gap between primary and secondary blanking, suppresses almost all potential activity. This suppression is also clear in the trial-level, population-average PSTH (Fig. 6.9C), where it compresses the range of recovered responses across the wider 2–80 μA sweep.

Across stimulating electrodes and denoising estimators, we observe increasing population response strength with stimulation amplitude, consistent with prior reports [80, 247]. Fig. 6.9D shows this trend for one example. The response is graded under every estimator, but several estimators spike irregularly. This likely reflects inconsistent assessment of response strength across protocols, as seen in Fig. 6.9C. We quantify this trend by the rank correlation between amplitude and recovered response in Fig. 6.9E, which provides a simple metric we normatively expect to be high based on monotonic recruitment shown in Sombeck et al. [247]. To this end, all methods report high correlations, but DELETE, Template, and Poly LRR achieve significantly higher correlations than the other denoising baselines (paired Wilcoxon tests, Holm-adjusted $p < 0.05$, except Template versus Spline, $p = 0.147$). These estimators stand in contrast to blanking and high-pass filtering without an artifact model (No-op), whose weak amplitude ordering is consistent with sporadic artifactual threshold crossings across stimulation amplitudes, contaminating the estimated response.

However, a smooth graded response curve is insufficient to distinguish denoisers that mainly differ in depth of modulation. For example, a graded response can arise trivially from artifact: raw peak-to-peak artifact voltage (Artifact P2P) has the highest amplitude correlation of all conditions in Fig. 6.9E. We therefore ask whether the denoisers preserve variation across stimulating electrodes that is relevant to the downstream behavioral report of ICMS detection. Specifically, stimulation-evoked neural activity can be used to define a *neurometric threshold*, which we compare with the psychometric detection threshold. We construct the neurometric threshold using an ideal-observer framework [257], measuring how reliably recovered activity distinguishes stimulation from the no-stimulation reference. The threshold is the amplitude at which this discriminability, quantified by AUROC, reaches 0.75 (Section 6.2.5; Fig. D.5A). Using the same criterion for neural discrimination and behavioral detection allows us to compare their thresholds across stimulating electrodes.

We compare behavioral and neural thresholds under DELETE in Fig. 6.9F. Aggregated across participants, we observe a moderate correlation ($r = 0.57$, $p < 0.001$), consistent with Grani et al. [80]'s interpretation of the local neuronal response to ICMS

as a neural correlate for behavior. Other denoising estimators also show significant correlations (Fig. D.5C and Fig. 6.9G), whereas Artifact P2P and No-op show little correspondence with behavioral threshold ($r = -0.03$ and 0.09, respectively). For Artifact P2P, the artifact is already discriminable at the lowest delivered currents, placing its estimated thresholds largely below the tested range. Paired bootstrap comparisons, Holm-adjusted across seven contrasts, distinguish DELETE from Artifact P2P ($\Delta r = 0.60$, $p = 0.004$), with No-op marginal ($\Delta r = 0.48$, $p = 0.065$), while differences among the denoising estimators are unresolved ($p > 0.05$). The correspondence with behavior thus depends on artifact removal, and is not reproduced by the raw artifact itself. DELETE's strong correspondence with behavior provides convergent evidence that it preserves behaviorally relevant response structure, consistent with its robust performance across the metrics in Section 6.3.4.

Unlike the approximately matched thresholds of tuned neurons responding to natural stimuli [257, 259], the majority of our neural thresholds are lower than their corresponding psychometric thresholds, placing them below unity, sometimes substantially so. This directional offset is expected from the causal position of the measured response. In natural sensation, behavior can pool information across a population that is largely invisible to our current neural recordings. With ICMS, by contrast, the locally evoked neural response is the majority root cause of the subsequent perceptual report. To the extent that our recordings capture the relevant local state, downstream readout may discard stimulus information but cannot create information absent from that state. We therefore expect ICMS to become neurally discriminable before it becomes reliably perceptible, and thresholds to order correspondingly. This expectation can fail given our incomplete recording of the directly evoked response [258], and our scalar summary of what we can observe, but nonetheless suggests a likely fallibility of estimates that place neurometric thresholds above psychometric thresholds. Template subtraction and DELETE yield neurometric thresholds above psychometric thresholds least often (3 and 4 of 35, respectively), compared with 14 for Poly LRR (Fig. D.5C).

6.4 Discussion

Our ambition to understand the brain precisely depends on parsing the data we collect. This requires separating neural signal from the myriad artifacts that contaminate raw

sensor data. The literature is replete with rejection methods that perform well in their motivating settings, but benchmarks continue to find simple baselines competitive across varied settings [243, 240]. This tension implies a practical uncertainty: it remains difficult to know if a method's assumptions will remain appropriate for a new artifact's structure.

Here, we treat generality as an explicit design and evaluation objective. DELETE pairs the simple principle of removing predictable structure with the expressive capacity of a deep network. We evaluate this combination's effectiveness with ground truth in spatial and temporal artifact simulations, heuristic measures on real recordings that mix multiple types of noise, and perceptual reports in ICMS-based detection data. Across these evaluations, DELETE matches strong baselines in simpler settings while recovering neural responses under more complex artifact structures that challenge individual baselines. DELETE thus shifts the focus of artifact rejection from tuning parametric forms to isolate specific noise sources to designing an encompassing input context from which artifacts are predicted.

This flexibility comes with less interpretability than a polynomial fit, and further risks removing valuable neural signal. Indeed, DELETE's objective implies that highly entrained neural responses should be removed along with artifact. This is a fundamental limitation of the objective, however, and applies to prior linear methods like ERAASR, as well. We replicate an analysis from O'Shea and Shenoy [251] in Section D.1.1 showing that DELETE does not qualitatively increase suppression of entrained neural responses beyond ERAASR.

In practical terms, DELETE follows an increasingly widespread deep learning pipeline of pretraining and day-to-day adaptation. DELETE needs no privileged inputs, using only the broadband data to be denoised in both training and deployment, without annotation. Furthermore, DELETE is cheap to run, with a configurable stride that can achieve real-time latency on a consumer GPU (NVIDIA 4090) with minimal performance loss.

DELETE's generality is likely to matter more over time, because stimulation itself is becoming less stereotyped. As noted in Section 6.3.4, spatiotemporally patterned stimulation is an active research direction, and such patterns are increasingly likely to come from an optimizer rather than a pre-specified pulse train [260, 73]. Artifacts generated this way inherit the variability of the patterns that produce them, and are correspondingly poorly served by methods that assume stereotypy across pulses,

channels, or trials. All of our evaluated methods struggled even with stimulation of just four electrodes at fixed timing, suggesting that more complex patterns will pose still greater challenges for artifact removal.

DELETE's initial step towards generalized neural data preprocessing invites several future directions. The most convincing confirmation of DELETE's utility will be through demonstrated benefits in downstream analysis. We take a first step in Section 6.3.5, where detection thresholds derived from DELETE's recovered activity track the participant's own perceptual thresholds. However, this metric is rather coarse for comparing denoisers. Future work should pursue more challenging and impactful tasks, such as decoding rich perceptual reports or controlling ongoing activity, to better evaluate preprocessing quality. Further, DELETE's efficacy on real data has only been established for recovering ICMS-evoked spiking activity. We have not explored strategies to separate artifacts with different spatiotemporal extents, or to recover different types of neural signal like LFP. Finally, developing DELETE into an off-the-shelf foundation model would require scaled pretraining and evaluation. Such a study would inform adoption procedure, like whether new subjects, studies, or hardware would warrant expensive custom pretraining. Such work would likely require a more standardized data ecosystem for artifact rejection, which is challenged by the community's diversity of stimulation hardware and protocols.

Separating signal from artifact will remain an evergreen challenge for scientific analysis. DELETE replaces artifact-specific designs with one model to remove locally predictable content. This simple and general approach is not behaviorally a large change from familiar methods, but provides a framing that will become essential as neuroscientific data collection grows in complexity and scale.

Chapter 7

Discussion

In closing, we revisit the works in this thesis, both to provide additional commentary on their relationship with contemporary work and to highlight analyses that might more tightly connect this thesis to the surrounding literature. We then discuss future directions in closed-loop motor control and sensory stimulation before returning to the broader motivation for modeling neural data at scale.

7.1 Summary

7.1.1 Spiking data spanning sessions, subjects, and tasks can be pretrained over as one corpus

A basic requirement for pretraining is that transfer learning can occur between the datasets to be pretrained on. The NDT2 study (Ch. 3) took dataset pairings previously shown in the literature to share statistical structure, and measured a Transformer's ability to transfer learn between them. We showed that transfer learning was present, scaled with the size of the source pretraining dataset, and was strongest when the source and target datasets only varied in time as opposed to subject or task.

Regrettably, NDT2 did not connect pre-existing analytic characterizations of dataset similarity (via canonical correlation analysis) [123, 124] to transfer learning performance. As is, there is a discrepancy between the qualitative narrative of that prior work, which claims near-equivalence of the top PCs of datasets across time and subjects, and the unequal transfer learning performance shown in NDT2.

7.1.2 BCI decoding needs more accessible benchmarks

FALCON (Ch. 4) is most interesting for its interface, which admits a high degree of flexibility in potential submissions. Its released training dataset is designed to allow methods that want to overcome session nonstationarities in unsupervised or supervised manners, but usefully also supports comparison of putative foundation models. I hope FALCON can serve as an analog of the GLUE benchmark [261] for the BCI community. GLUE is a multi-task language benchmark that critically legitimized pretraining in NLP. While BERT [130] posted dominant scores on GLUE to show the power of Wikipedia-scale pretraining, NDT3 did not outperform NDT2 with FALCON-only pretraining. As the broader neural foundation modeling community pushes ahead on claims of benefits from scaling, FALCON provides a concrete venue to demonstrate gains beyond those achieved by NDT3.

Advances in neural foundation modeling will require a maturing data ecosystem, which ostensibly means pretraining corpora, codebases, and benchmarks. For neural data, benchmarks are currently a particularly salient bottleneck. Codebase releases for the sake of standardization and reproducibility will always remain useful, but their value in reducing the barrier to entry may be less vital as coding AI is adopted. I deem data releases less deserving of attention simply because public data releases are already trending upwards, especially in animal neuroscience [233, 262–264, 106]. Moreover, data releases only weakly enforce the standardization that ML comparisons require. Even in language modeling, large public data corpora are routinely filtered and reprocessed to produce better pretraining [265–269]. There have been commendable efforts to standardize preprocessing and evaluation in EEG and fMRI [183, 270], but the breadth of data and tasks to standardize is large, leaving proper foundation model assessment muddled [271].

Data ecosystem tooling is worth only the adoption it achieves, and FALCON appears to be struggling on this end. Now two years post-release, there have only been a handful of competitive submissions that have nudged numbers slightly on one track or the other, compared to the contemporary Brain2Text benchmarks [272, 273]. While B2T 2024 drew a level of interest and number of submissions similar to FALCON, contemporary high-profile research articles on speech neuroprostheses and a move of the evaluation platform from EvalAI to Kaggle led to orders of magnitude more submissions to B2T 2025. The lesson from FALCON applies to the breadth of meta-scientific work needed for foundation models: these tools earn their value through

uptake, and thus maximizing adoption should be a first priority.

7.1.3 A baseline motor foundation model suggests limits to scaling

Chapter 5 builds a pretraining corpus of thousands of hours, pretrains NDT3, and evaluates it extensively. Our primary finding is that gains are restricted to low downstream data scales. This must be reconciled with the field's enthusiasm for scaling to larger models and data. My resolution is that most other works evaluate at lower downstream scales than NDT3, and so have not yet met the limit. Consider three high-profile works: POYO+ [274], NEDS [275], and Wang et al. [276]. All claim scaling gains, but all fine-tune below NDT3's 90-minute limit: 20 minutes for POYO+ (private correspondence) and 20 for NEDS (computed from released data and the reported train split). Wang et al. [276] tunes on up to 76 minutes, the closest any comes to testing the limit. The point is not exact numerical agreement with NDT3, but that claims of scaling benefit must be situated against pragmatic downstream data scales. In my mind, the implicit campaign promise of foundation models has been to enable a qualitatively new level of BCI functionality. To meet that expectation, I would expect foundation-model-enabled BCIs to improve on hundred-hour-plus subject-specific models, which persons living with disabilities might readily accommodate to regain a degree of independence. This bar has not been met yet.

Regardless of evaluation technicalities, the baseline expectation that neural foundation modeling should at least somewhat succeed obliges us to ask whether NDT3's limited scaling is fundamental or methodological. NDT3 draws two frequent critiques: its pretraining data composition, and its naive patching tokenization. Indeed, NDT3's data is not particularly impressive, combining minimally curated experimental data from several labs with a high volume of impoverished human calibration data that inherently contains labeling errors. Minimal curation can only harm training, and is likely unique to NDT3 among contemporary work. Wang et al. [276], for instance, pretrains on a small (15-hour) but uniformly collected cohort of eight mice. Other projects have typically used external releases from the Allen Institute or the International Brain Laboratory [262, 263], and future industry models are likely to use internally standardized data. Beyond low raw data quality control, NDT3 also uses relatively mundane behavioral tasks, with many hours of center-out reaching trials.

To this limitation, my instinct is that richer or ultimately naturalistic data, of, say, unrestrained monkeys moving through rich habitats, would likely improve scaling trends. Even still, it seems unlikely that the improvement could span the two orders of magnitude between NDT3's 90 minute pretraining-positive regime and the hundred hour-data we'd realistically have for a single user.

NDT3's patch tokenization was chosen as a naive baseline that simplified technical implementation. With coordinate annotations, the tokenization strategy should be exactly as complete in the universal approximation sense as other approaches: per-context adapter layers [87, 275, 276], neuron-level embeddings [180, 274], or in-context learned representations [277–280]. However, we showed that NDT3's performance degrades sharply under input coordinate shifting across patch boundaries, which must be a unique limitation of the naive tokenization strategy. It remains to be seen whether sensor coordinate variability is resolvable by these other strategies, which may incur their own failure modes. Comparison of intracortical scaling against modalities with more stable coordinate systems, like fMRI, should inform whether sensor variability is a fundamental limitation for scaling.

7.1.4 Stimulation artifact can be mitigated by learned reconstruction

Chapter 6 presents a new perspective on artifact rejection through surprisal. DELETE is trained to reconstruct broadband recordings from local spatial and temporal context, with the assumption that neural signal is harder to predict than artifact. It requires no explicit targeting of artifact epochs. It matches the strongest baselines across artifact simulations, preserves neural signal in stimulation-free recordings, and recovers spiking activity within milliseconds of an electrical stimulation pulse. As the rest of the thesis and broader field of neuroscience analyze the spiking activity downstream of DELETE, DELETE's major impact could be in a marginal increase in the quality of a vast amount of raw neural data collection.

DELETE's full realization requires progress in two directions. First, there should be further validation that the recovered responses reflect neurophysiological truth, most likely by a demonstrated gain in perceptual decoding performance from ICMS-evoked responses [79]. For example, the intensity of an ICMS-evoked percept should likely be best decoded when the putative response being decoded is most accurate. Second,

DELETE requires technical maturation of its approach in the vein of AI systems engineering. To apply DELETE online for real-time ICMS control experiments, we would need to solidify its pretraining and continuous fine-tuning strategy, which may be more challenging than the analogous system of linear regression reference fit over a moving buffer [281].

7.2 Future directions

7.2.1 Closed-loop deployment

This thesis' scope has largely been offline evaluation, but it is essential that claims on the benefits of neural foundation models are grounded in improvements in online control. High-degree-of-freedom control currently requires daily recalibration and frequent experimenter intervention, and the calibration burden grows with dimensionality, reaching roughly fifteen minutes for seven degrees of freedom [16] and twenty for ten [36]. A minimal benefit of scaling neural foundation models should be to reduce this calibration burden (whilst we are not in the multi-session, multi-hour data regime).

Translating offline prediction benefits to online control is, however, challenging, as they are fundamentally distinct objectives. As we have seen in NDT3's angular generalization experiments, NDT3 excels at constraining its outputs to seen training directions, at the cost of failing on unseen directions. In preliminary online cursor control experiments, we find NDT3 performs equivalently to a linear baseline (OLE, Fig. 7.1A–B), which suggests online control may be constrained more by label quality than by model capacity. Indeed, all models are affected by insufficient labels from open-loop calibration, which has motivated a number of closed-loop adaptation methods based on providing label corrections during partial or full decoder-based BCI control [37, 212, 282, 283]. We observe that NDT and linear baselines alike improve with this closed-loop adaptation in Fig. 7.1C, but to similar extents.

These observations motivate examining both the demands of the control task and the learning signals used to train the decoder. First, we can attempt to identify a task of sufficient complexity and difficulty that decoder performance does not saturate in a short period of calibration (regardless of label noise). For instance, multi-session-trained deep networks are necessary for the highest performance in most speech BCI settings. Similarly, high-DoF motor control tasks that require tens of minutes of

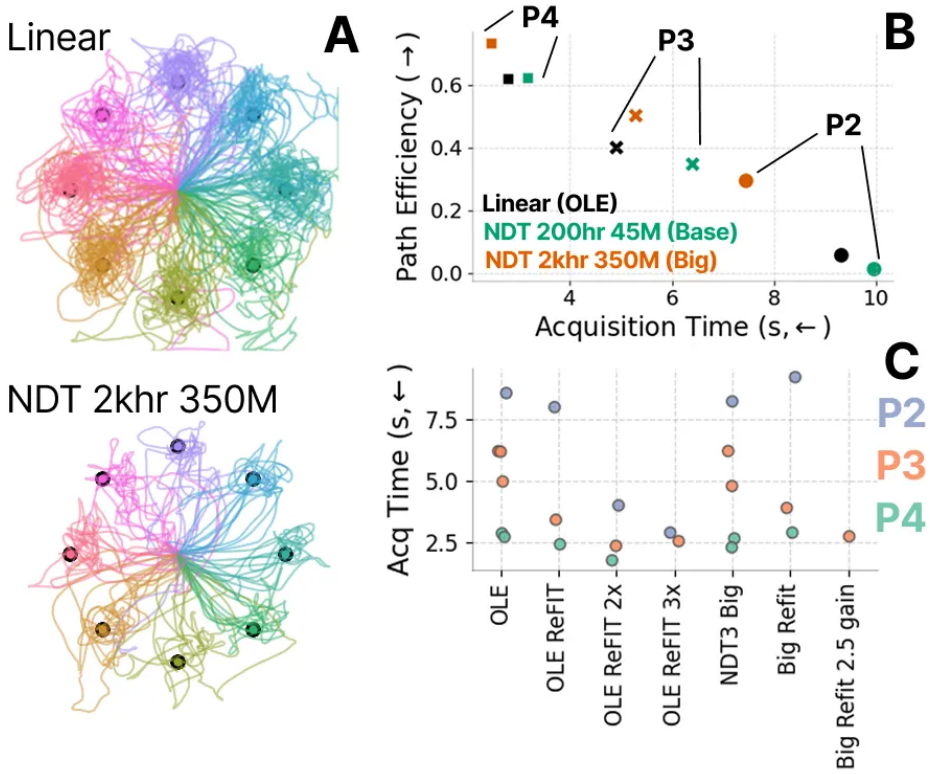

Figure 7.1. Pretrained and linear controllers are constrained by label quality. A. OLE and NDT brain-control trajectories in one human participant. **B.** The largest NDT model matches linear model performance for 2D cursor control when all models are trained on an open-loop calibration block, for 3 participants (P2-P4). **C.** With closed-loop tuning, control improves for both NDT and the linear baseline, to comparable extents.

calibration are more appropriate targets than two-dimensional cursor control. Second, we should consider a curriculum which introduces less prescriptive learning signals for the BCI controller. Speech BCIs benefit from using a Connectionist Temporal Classification (CTC) loss function [284] to learn to transcribe speech without explicit temporal alignment in its labels. In contrast, continuous movement BCIs assert an inevitably incorrect precise movement vector at each time point. One potential direction forward would be to transition BCI calibration from supervised learning to reinforcement learning, where the BCI controller is optimized (in concert with the user) to maximize control performance.

7.2.2 Modeling and optimising the neural response to ICMS

An immediate and important synthesis of this thesis' work on motor foundation models and artifact rejection is a large-scale model of the neural response to ICMS. Current practice for determining sensory BCI parameters is very time-consuming and costly, as it involves testing different ICMS patterns and recording their perceptual response [66, 65, 80]. Consequently, the design space of usable ICMS parameters is highly constrained, and likely precludes the development of ICMS that can reliably evoke more complex sensations. The resulting search burden is fairly intractable so long as it depends on perceptual feedback, even e.g. with Bayesian stimulation parameter optimization [285]. To unlock the wider potential of ICMS, we can leverage the hypothesis of biomimicry [72], and optimize ICMS patterns to target neural responses instead of perceptual responses. This produces two novel challenges: how to learn models of the neural response to ICMS and models of the perceptual response associated with a given pattern of neural activity. We focus on the former first, and address the latter in Section 7.2.3.

Models of the neural response to ICMS may be much more tractable than their perceptual counterparts, and have indeed already been demonstrated to control evoked responses in visual cortex [79]. Their utility in the broader sensory BCI context rests on two assumptions. First, we assume that the controllable variability of ICMS-induced perception is appreciably large and mediated by the locally observed neural response, which the spatiotemporal locality of the immediate response to ICMS makes plausible [258]. This assumption makes modeling the neural response worthwhile in the first place. Second, we assume that the response to passive, task-free stimulation is related to the response during a functional task. This assumes a relatively minor impact of top-down modulation on the population response in ICMS as we currently use it, and is for example supported by observations that phosphene detection thresholds are largely unchanged when a participant is mobile rather than seated [286], and more directly by the finding that representations learned from spontaneous activity alone support decoding of stimulation-evoked perception [287]. This assumption is vital, as passive ICMS can be collected at volume with minimal burden on the user.

Even granted the ability to use passive ICMS, the stimulation parameter space is exponential, so we would certainly need to account for data integrated across many sessions. This research program of mapping the neural response to ICMS would likely benefit from initial surveys of the relationship of responses across electrodes and

stimulus classes. Eventually, a comprehensive ICMS response model could be built through a model-driven active learning loop with batched updates from experimental data, akin to the in silico loop of Ding et al. [112].

7.2.3 Somatosensory alignment

The ICMS response model does defer the burden of modeling perceptual responses, which is ultimately the bottleneck of sensory BCI calibration. To address this, we must seek an additional source of data to ground the mapping from neural activity to reported percepts. Identifying responses for a given percept would not be particularly challenging in an intact participant, as we would simply record the response to an appropriate external stimulus. However, BCI participants typically receive their devices due to impaired sensorimotor pathways, so we cannot generally rely on this approach. Here, we propose two additional potential sources of information. First, we can lean into the foundation model premise, and collect neural responses to a wide variety of stimuli in intact individuals, such as primates. Although there will inevitably be individual variability in the neural response and species-level distribution shifts, our goal is not to eliminate the need for perceptual response collection, but to minimize it. In this framing, the participant burden is minimized because we can tune the response-behavior foundation model on a reduced set of participant-specific perceptual reports, rather than building such a model from scratch.

Second, we can turn to the empirical alignment between task-optimized deep networks and neural data, which has not yet been discussed in this thesis. Concretely, there is a robust literature demonstrating that task-optimized deep networks predict neural activity, most maturely in vision [288, 289, 190], but also in language [290, 108], audition [291], and olfaction [292]. These same task-optimized networks can be tuned on neural activity to produce better predictors of it, and the mapping so learned transfers across participants [293, 294]. Compared head to head against networks fit on neural data alone, the task-optimized prior network mainly buys neural data efficiency rather than ultimate accuracy [295, 296], but this is suitable for our BCI use case. The closest demonstration of task-optimized alignment for somatosensation is in rodents, where networks trained to categorize simulated whisker input come to match somatosensory cortex closely enough to saturate the explainable neural variance, with alignment scaling linearly in task performance [297]. In principle, these task-driven

networks provide a whole new source of priors separate from neural data for scaling neural data foundation models.

7.3 Coda

Every work produced in this thesis was written with the premise that neural data collection is growing to dizzying heights. Against this backdrop, the work here has hinted at the value of scaling and clarified some of the major obstacles to realizing it, giving us both reason to continue the program and a clearer sense of what it requires. We should appreciate the particular coincidence that our ability to interrogate the brain in new orders of magnitude of detail is proceeding in lockstep with deep learning's maturity as a commodity tool to index this new data. Moreover, there is much to gain from productively modeling all this data, as the neural interfaces consequently powered could potentially reshape how we understand ourselves and interact with the world. We are thus Cartographers still, mildly chastened but better equipped to map the most immaculate Empire of all.

Appendix A

Supplement to Chapter 3 (NDT2)

A.1 Supplementary Material

A.1.1 Dataset Preparation

We aimed to perform minimal preprocessing on datasets. For pretraining, we do *not* explicitly filter for successful trial outcome as done in some neuroscientific analyses (though some datasets are released with erratic outcomes pre-filtered). Neither do we (beyond what is provided directly in datasets) filter for cross-correlated channels or low-firing neurons. We also do not z-score neuronal firing, both for simplicity and so as to not remove any potential cross-channel/session information. The one exception to this is that neurons with firing < 0.5Hz are removed in the sorted analysis of the O'Doherty RTT dataset, to reduce the number of spatial channels below 288. As some datasets report single unit activity and some report multi-unit activity, the dynamic range of the input data varied by an order of magnitude, with baseline firing rates varying between 0.1Hz to upwards of 50Hz. We believe additional data curation is likely to improve model quality.

In total, the max number of pretraining trials or pseudo trials was on the order of 100K trials. Each trial lasted up to 2.5s (cropped or chunked if trials were longer), and used all recorded M1 activity, and PMd activity if available.

Decoding targets were either in a standard unit or in z-scores against the dataset mean and standard deviation (not a session specific z-score). Standard units of meters/second were primarily used in most RTT analysis, except when preparing an RTT/human BCI decoder, which used respective z-scores.

Quantifying volume Should neural data pretraining proliferate, it will be important to distinguish pretraining efforts by neural data volume. This is tricky as the most pertinent unit of neural data seems likely to depend on the function of the neural data; we are concerned comparisons on the convenient basis of time may lead to the illusion of standardization. For example, some datasets may contain much more behavior than others in the same time periods. This issue is still present, though less severe, if we presume to only differentiate data up to orders of magnitude: compare experimental behavior vs naturalistic behavior, which may have extended periods of rest. For now, we see it reasonable to quantify the value of data in pretraining with respect to a semantic units of covariate data (experimental trials), and encourage documentation of pretraining composition.

Reaching datasets

- Neural Latents Benchmark motor datasets (`MC_Maze`, `MC_Maze_small`, `MC_Maze_med`, `MC_Maze_large`, `MC_RTT`): ∼3.7K trials.
- Churchland et al., obstacle-guided (maze) reaching, 2 monkeys, 9 sessions / ∼20K trials.
- Nir-Even Chen et al., delayed reaching, 2 monkeys, 12 sessions / ∼ 80K trials total.
- O'Doherty et al., self-paced reaching, 2 monkeys, 47 sessions / ∼40K seconds total.

Isometric manipulandum datasets

- Gallego-Carracedo et al., isometric center-out and hold, 2 monkeys, 12 sessions, ∼2.7K trials total.
- Dyer et al., 2 monkeys (same as above), 4 sessions/∼750 trials total.

Human BCI datasets [Private]

- Human participant data from ongoing clinical trials. Subsetted to 2D cursor control activity, either under observation/attempted activity, partial, or full BCI control. During observation, participants observe a programmatically controlled cursors, which e.g. is performing center out at a steady pace in a trialized fashion. 130K trials. Note only observation trials are used for evaluating decoding.

A.1.2 Methods for online cursor control experiments

We evaluate realtime, online cursor control using a simple Center-Out reaching task. This is a trialized task where the cursor initializes on the center of a screen and the participant reaches to 1 of 8 radially spaced target that was cued at the start of the trial. The trial ends after the cursor is held at the target for 0.25s (preventing fly-bys from ending the trial). During open-loop calibration, the cursor is autopiloted to the target. Velocity labels are produced by assuming that the participant's intention matches the autopilot presentation. Specifically: we take the velocity of the cursor on the screen and apply a boxcar filter of 500ms to remove nonsmooth rendering, using the smoothed result as our velocity label. To span more reach conditions, we use an open-loop calibration that does not reset the cursor at the center after each trial, but instead immediately presents the next target. Our unsupervised tuning uses the neural data, but not the labels, collected in the same open loop calibration.

After each of 8 decoders (0-shot and supervised for OLE, 0-shot; supervised and unsupervised for 2 NDT models) are prepared, we evaluate each decoder in a block of 40 trials. Evaluation blocks proceed without randomization (OLE always began evaluation as it is the fastest to prepare); our results may advantage earlier decoders due to participant fatigue which accumulates over the session.

A.1.3 Compute and Hyperparameter Tuning

The full, uncurated logs of all model training are available at https://wandb.ai/joelye9/context_general_b joelye9; interested users may compare specific hyperparameters logged there against those in the codebase during reproduction attempts.

Basic hyperparameters

1. In both pretraining and fine-tuning, we scale batch size (accumulating batches or using multi-GPU training when necessary) to be roughly proportional to full dataset so that each epoch requires 10-100 steps; we find performance is not too sensitive to batch size within an order of magnitude of this heuristic (especially in pretraining).
2. In pretraining we manually tuned LR to $5e-4$ in initial experiments and hold it fairly constant in pretraining. We swept learning rate in our hyperparameter comparisons below.

3. In pretraining, we use learning rate warm-up for 100 epochs, and decay to ϵ by 2500 epochs. This is a high threshold that is typically not reached: training converges within 100-1K epochs for our manually tuned LR. In fine-tuning, we experimented with similar ramping schedule but settled on fixed small LR (which are typically grid-searched).
4. For RTT, we swept and found that a decoding lag of 120ms worked reasonably well. (This is similar to reports in [127]. For human BCI, we do not use decoding lag.
5. For human offline evaluation, we take the best of two evaluation hyperparameter settings: 10% or 50% masking during target-session tuning. We also report the R^2 only for times where the intent is non-zero; participants are not typically perfectly zero-intent during the majority of non-zero phases (i.e. data are noisy). We do not filter data by putative quality as measured by online performance in the experiment in which the data was collected; thus our calibration data includes several noisy, incomplete trials as well. Evaluation data are restricted to a contiguous set of sessions with non-trivial linear decoding.
6. Hyperparameters were not swept further during realtime pilots.

Compute costs We estimate computational costs with respect to data volume, as model size is held relatively static (6-12 layers, 128-384 hidden size). Most analysis was run on SLURM clusters. Pilot realtime decoding used an NVIDIA 1060/2060, where loop time was <15ms (neural data was binned at 20ms). Model tuning was performed on a remote server and took <5 minutes.

1. Fitting datasets on the order of 1K trials typically requires 20m-1hr on 12G 1080/2080-series NVIDIA GPUs.
2. 10K-20K trial datasets require 2-8 32G-V100 hours.
3. 100K+ datasets require 72 80G-A100 hours.

A.1.4 NDT2 Design Notes

NDT2 introduces two primary design elements: context tokens, and patch size. We first consider the effect of context tokens. NDT2 integrates context tokens directly by adding learned tokens and adding them to the data token sequence (i.e. in-context aggregation). Our pilot experiments found no difference using cross-attention

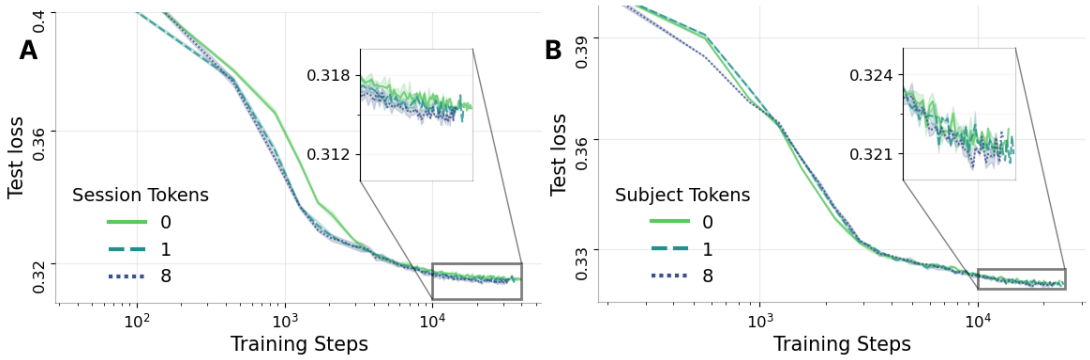

Figure A.1. Context embedding ablations. A. Multi-session model training curves, with varying session context token count (3 seeds). Learning is improved, but additional tokens do not have notable effect. The converged score in only modestly affected. **B**. Subject transfer training curves with similarly varied token budget for subject embedding. Models receive 1 session token. There is no clear additional improvement nor harm.

integration. The training curves of sorted multisession models augmented with context tokens, shown in Fig. A.1A, demonstrate a primary effect in speeding convergence, which can be valuable in large scale pretraining. The benefit to converged NLL (some $1e-3$) is modest but non-negligible, considering the NLL resolution in Fig. 3.3. This trend replicates at smaller data scales (Fig. A.6). Providing one session token and additionally varying the available subject tokens (Fig. A.1B) has much smaller effects. However, given no harm and negligible compute overhead, we hold as a default policy to provide one token for each of session, subject, and task.

Patch size is a less intuitive hyperparameter in NDT2 than in a similar architecture in the image domain, ViT. This is because units within a patch are not meaningfully related, especially across datasets; adjacent units often correspond to different physical locations on an electrode array, and this physical meaning is typically underspecified in open data releases. Smaller patch sizes, ideally at the single neuron level, thus appear to be a better design.

However, in early experiments we explored patch size but quickly found a tradeoff. Smaller patch sizes do appear to incrementally improve neural data models, but are both more expensive computationally (to train) and statistically (to learn decoders off of). We show this in Fig. A.2. Note how the unsupervised NLL is similar or better with smaller patches, but decoding is dramatically worse, regardless of whether we mean pool across the population at each timestep (Pool) or not. Smaller patches may be worth revisiting if we have a high amount of supervised data to train the decoder.

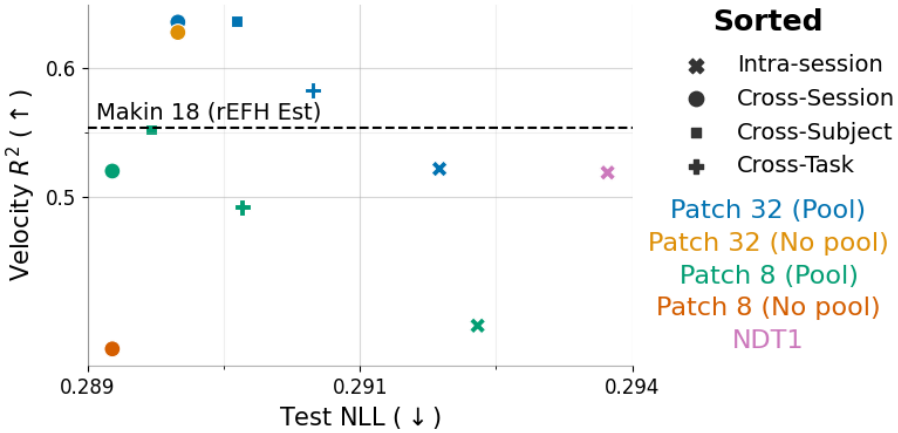

Figure A.2. 32-neuron patches compared against 8-neuron patches. Smaller patches sometimes improve NLL but reliably decode more poorly. (Pool)-ed models average-pool all the neural data features for a given timestep before being given to the decoder, whereas regular decoders simply cross-attend all neural data features.

Architectural details. We refer readers to the codebase for full details, but provide an overview here. For each patch, NDT2 linearly projects input spike counts and concatenates these projections to form the patch embedding, zero-padding if the patch isn't full. Input spikes pass through Transformer encoder and decoder layers, and the final layer's output is then linearly projected to predict Poisson rates or effector velocity. NDT2 used pre-normalization layers but otherwise leave the Pytorch implementation of the Transformer layers untouched.

HP Sweeps.

We briefly show that NDT2 achieves higher performances than comparisons when sweeping across dropout ($[0.1, 0.4]$), weight decay ($[1e-3, 5e-2]$, and hidden size $(128, 256)$. NDT2 does have higher variance, but the main sensitivity is to dropout. We run this sweep and test evaluation in one training stage, Our base NDT2 uses dropout 0.1, hidden size 256, weight decay $1e-2$. In the code, this experiment is configured in `exp/arch/tune_hp, exp/arch/tune_hp_unsort`.

Mask Ratio.

We do not widely explore mask ratios due to compute constraints. In pilots throughout, we do not find that decoding is too sensitive to mask ratio (e.g. Fig. A.4), but reconstruction quality is hard to compare as the inference problem depends on the masking ratio itself. The reasonable effectiveness of high mask ratios is consistent

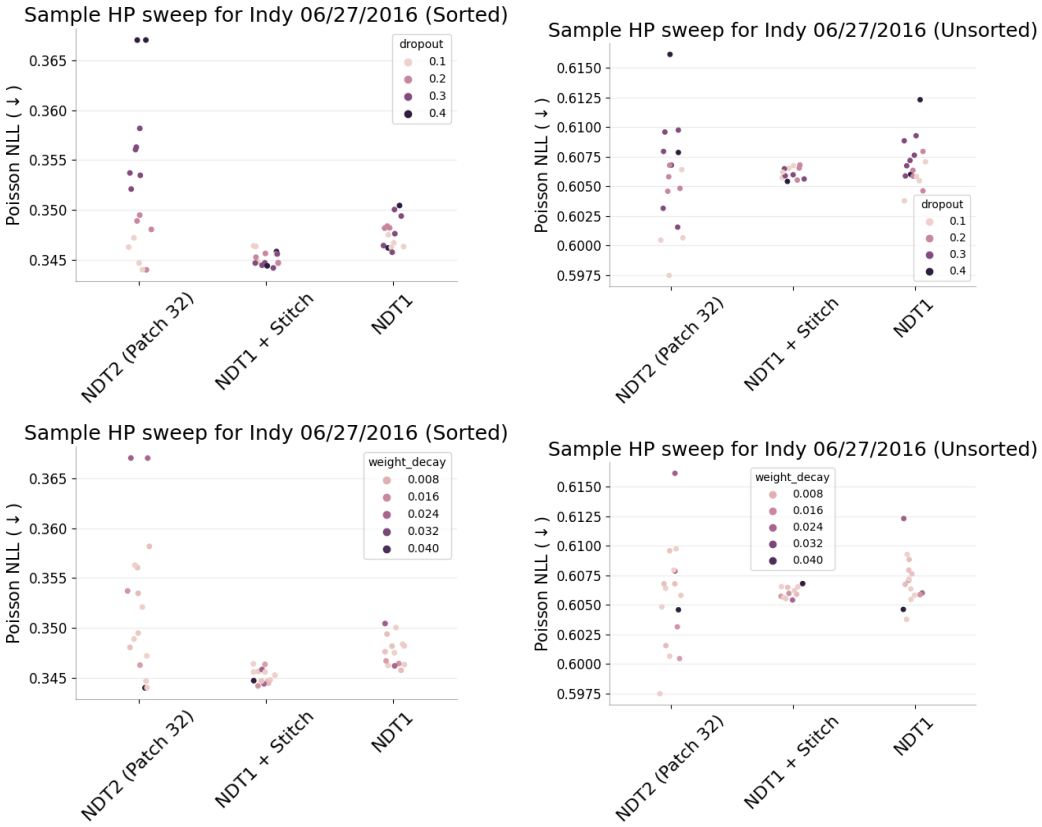

Figure A.3. Sweeps for regularization parameters. NDT2 requires lower dropout.

with general observations of low dimensionality and high redundancy in the code, compared to say, language [298].

A.1.5 Additional exploratory experiments

Stitching design Our stitching implementation randomly intializes a linear readin and readout linear layer. For ease of implementation, we stitch at the output of the network encoder rather than the output of the decoder (the linear-exponential readout layer comes after per-context stitch layer). In unreported pilots, we find stitching into compressed dimensions (e.g. half of readin channels) to reduce the context-specific parameter count, or only including stitching at the readin or readout made no significant difference.

Kinematic decoder design There are three straightforward strategies for building a kinematic decoder with NDT2. In keeping with NLB, we could learn a linear probe on representations at each timestep, or we could use a thin Transformer decoder to

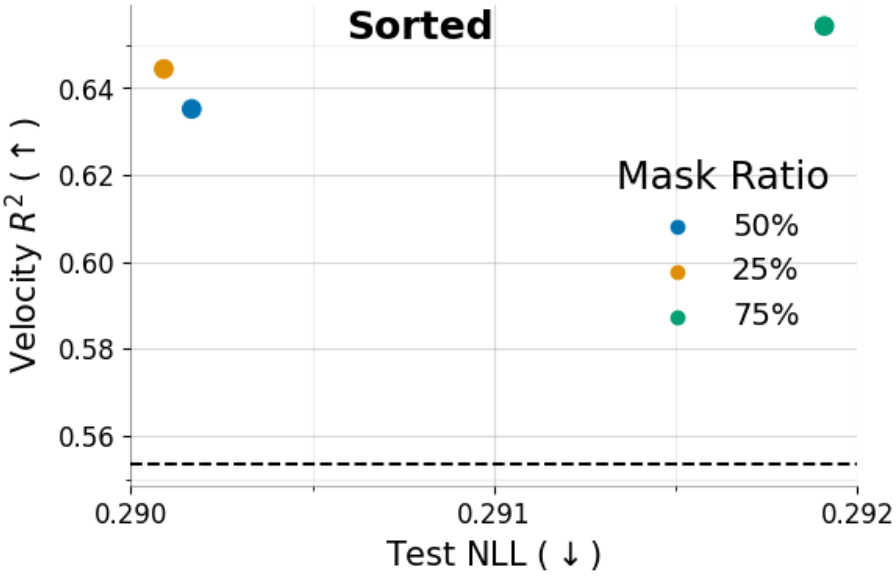

Figure A.4. Mask ratios over 5 datasets. At test time, the given ratio is held out during unsupervised evaluation.

allow information from multiple timesteps to aid the prediction. We experimented with both and chose the latter for minor gains. For simplicity, we run the Transformer decoder in one forward pass for all timesteps, i.e. there is no autoregressive feedback of previous kinematic estimates. We find that cross-attention for decoding queries slightly edges out in-context attention on the higher ends of the pretraining data scales we explore (e.g. 100K), and report with that setting. The primary difference is that cross-attention restricts neural data tokens from attending to the kinematic query tokens, while in-context strategies do not distinguish the two. For decoding probes, where the decoder is prepared on only a few hundred trials, we find it beneficial to mean-pool neural data tokens per timestep, and still use in-context attention (linear decoding directly works similarly).

A.1.6 Single-session breakdown of experimental results

Single-session variability We break open the aggregate results from our primary result figures. The primary takeaways are elaborated in each caption. Overall, we note that single datasets are insufficient to make conclusions on design choices given variability in results.

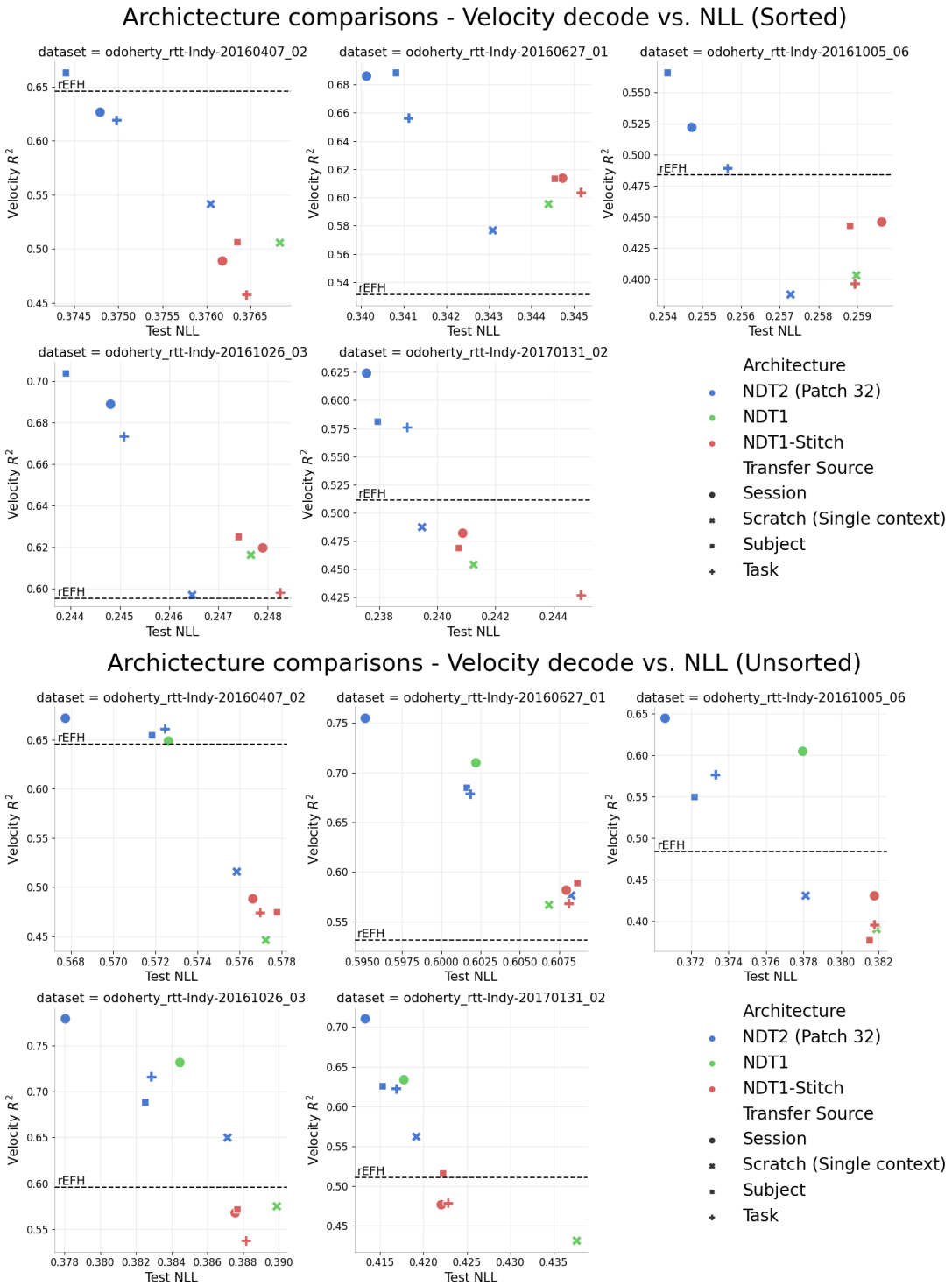

Figure A.5. We breakout Fig. 3.3 into individual datasets (points indicate means on 3 seeds). NDT2 shows consistent improvements over stitching, single session baselines, and rEFH (in most cases), but the ranking between data sources shifts, particularly for decoding scores.

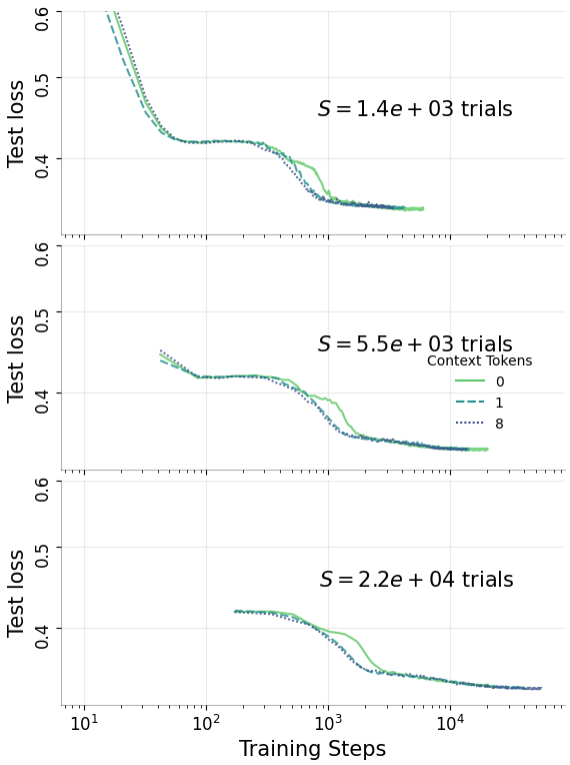

Figure A.6. Session context tokens improve learning, as measured by unsupervised loss in model training curves, though the majority of benefit of realized with 1 token. We compare this for three data scales, annotated by S, where we scale the number of trials available per session. (The increments were 100%, 25%, and 6.25% of the data). We hypothesized that more data per session would make session tokens less relevant, but the primary effect of increasing convergence appears unchanged at these scales.

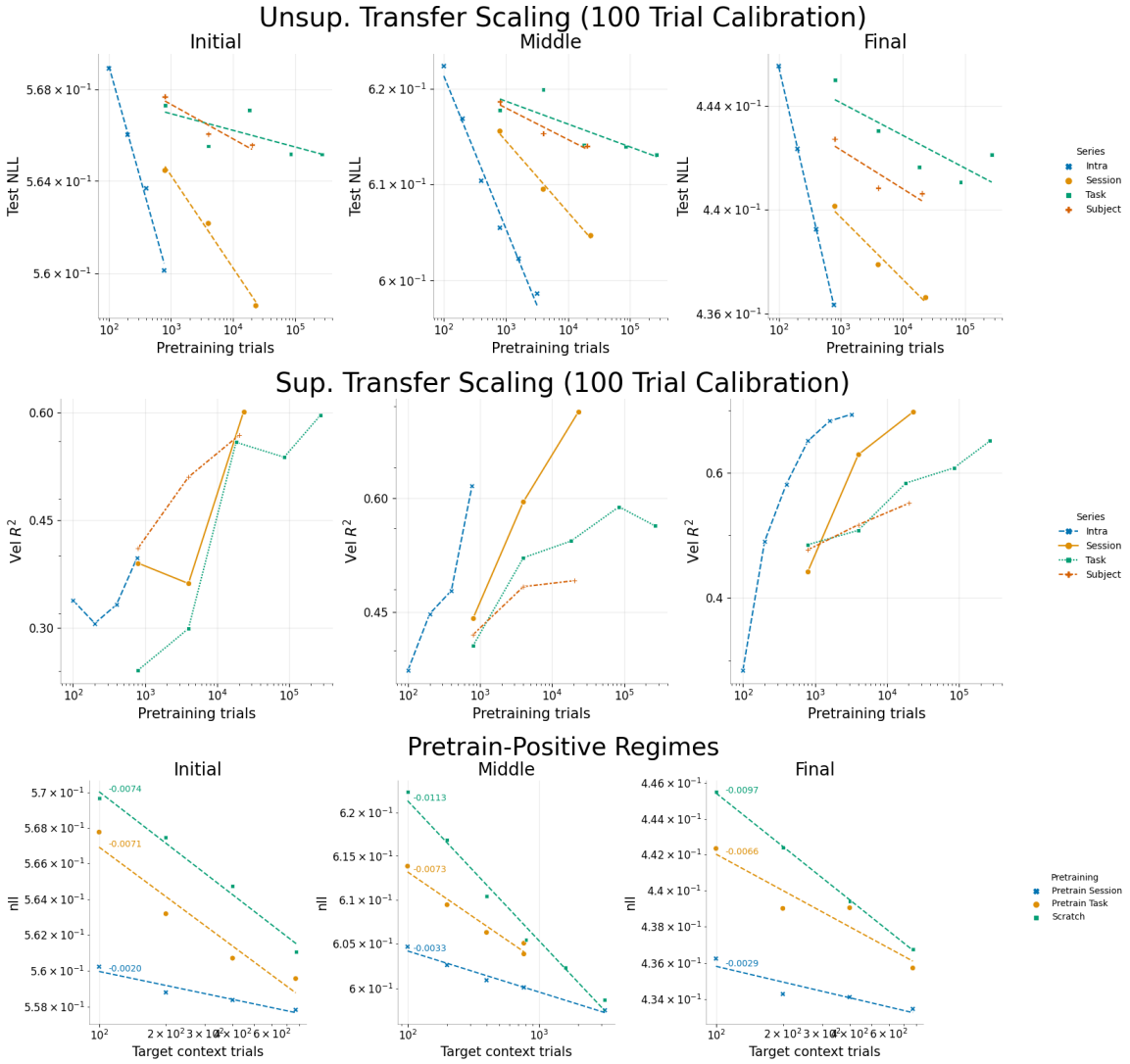

Figure A.7. Individual dataset results for three scaling analyses. We presented the largest middle session in the primary text, here we also show results on the first and final sessions in the dataset. The unsupervised and supervised transfer scaling reiterate the previous conclusions: cross-subject and task transfer provides low returns on scaling for unsupervised reconstruction, and decoding results are much more optimistic than unsupervised results. For convergence analysis (Pretrain-positive regimes), all three trend lines suggest convergence beyond 1K trials.

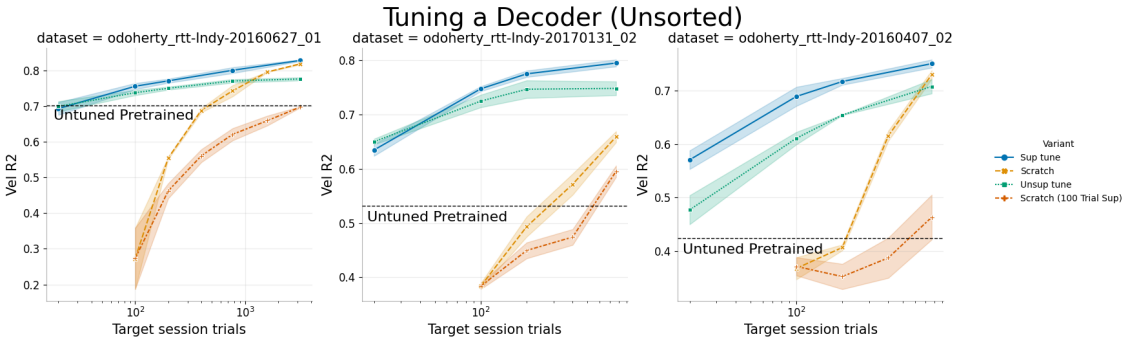

Figure A.8. The same breakout as above for pretrained decoder tuning. Again, we find that decoder tuning reliably outperforms non-adaptive decoders.

Appendix B

Supplement to Chapter 4 (FALCON)

B.1 Supplementary Material

B.1.1 Computational Resources

EvalAI is a platform that allows users to host and participate in AI challenges. It provides a simple interface for the participants to submit their solutions and for the challenge organizers to evaluate them. The link to our FALCON EvalAI page can be found on our challenge website: https://snel-repo.github.io/falcon/

For the FALCON challenge, we provide a container-based image evaluation infrastructure within EvalAI. This environment is primarily based on AWS Elastic Kubernetes Service (Kubernetes Cluster service) but also uses other AWS services like Elastic File System (EFS) to store the ground-truth dataset file(s) on which the evaluation is to be performed. The EFS file system is mounted on the AWS Elastic Compute Cloud (EC2) instance inside the EKS cluster, so the dataset file(s) can be accessed for evaluation.

Once the challenge's participant submits a solution (model) through the EvalAI platform by pushing a Docker container image, an EvalAI agent publishes the solution to the AWS Elastic Container Registry (ECR). The EKS cluster has a single EC2 instance server with pre-defined resources (CPU, memory, storage) which pulls the Docker container image and runs the evaluation script on the submitted solution. The evaluation script is responsible for evaluating the submitted solution and providing a score based on the evaluation criteria. The score is then sent back to the EvalAI platform to report leaderboard metrics.

The EC2 instance type is `g4dn.4xlarge` which has 16 vCPUs, 64 GiB of memory, 1 NVIDIA T4 GPU, and 100 GB of storage. This configuration is sufficient to run the evaluation script on the submitted baseline solutions for the FALCON challenge. EC2 G4dn instances are created to help accelerate machine learning inference and graphics intensive workloads.

B.1.2 Interpretation of FALCON Metrics

One goal of the FALCON benchmark is to standardize the metrics used for evaluation of stable decoding performance for each task. As the decoded outputs are very distinct in nature, different metrics were selected for tasks in each domain, each of which is representative of the most widely used metric in each field. As with all benchmarks, metrics should be interpreted with care, as these metrics alone do not necessarily capture all properties of a predicted output. R^2 has a convenient maximum at 1, but can be arbitrarily negative if the predictions contain more variance than the ground truth variable. R^2 also heavily penalizes predictions that are shifted from the expected center point. WER does not account for how close a given prediction is to the intended word and may overly penalize predictions that are only incorrect by a few characters. MSE can occupy unbounded ranges (i.e., $[0, \infty)$) that can be difficult to contextualize without other relative values. Hence, while models that demonstrate gains in the FALCON metrics will certainly show improved predictions, a poor-scoring model may not necessarily have unreasonable outputs. We recommend visualizing predictions to add additional context to FALCON scores.

B.1.3 Datasets

FALCON datasets come from multiple labs and were often collected as parts of larger experiments. Thus, some datasets included in FALCON may share subject and task with other publicly available data, but we ensure that any such releases exclude the specific held-out sessions used in FALCON.

Data format

All datasets were formatted according to the Neurodata Without Borders (NWB) standard for neurophysiological data. NWB provides open-source Python and Matlab APIs for reading formatted datasets (https://github.com/NeurodataWithoutBorders).

The data format builds upon HDF5 and can also be read using any package in any programming language that can access typical HDF5 files. Additionally, FALCON releases a code package that facilitates reading from and analyzing the converted NWB files (https://github.com/snel-repo/falcon-challenge).

Data hosting and licensing

Datasets are hosted on the Distributed Archives for Neurophysiology Data Integration (DANDI), a platform specifically designed for publishing and sharing neurophysiological data. DANDI generates metadata and identifiers for all uploaded datasets. The datasets we have released on DANDI are distributed under a Creative Commons Attribution 4.0 International license. The authors bear all responsibility in case of violations of rights. The FALCON datasets can be found at the following links:

- `M1-A` - `https://dandiarchive.org/dandiset/000941`
- `M1-B` - `https://dandiarchive.org/dandiset/001209`
- `M2`- `https://dandiarchive.org/dandiset/000953`
- `H1`- `https://dandiarchive.org/dandiset/000954`
- `H2`- `https://dandiarchive.org/dandiset/000950`
- `B1`- `https://dandiarchive.org/dandiset/001046`

Dataset documentation - `M1`

General description This dataset contains unsorted spike times and electromyography (EMG) data from two macaques performing a reach and grasp task. Neural activity was recorded from Floating Microelectrode Arrays (FMAs; Microprobes) implanted in the motor cortex (M1). EMG was recorded from 16 muscles of the right hand and upper extremity. EMG electrodes were comprised of 32-gauge, Teflon-coated, multi-stranded, stainless steel wire. They were implanted in bipolar pairs, separated by 5-10mm along the axis of the muscle. Muscle targeting and separation were performed as described in [299]. Wires were tunneled subcutaneously to exit the skin of the back at the midline in four separate bundles. Each bundle ended in a separate connector sewn into the back of a jacket worn by the monkey.

Our release of `M1-A` consists of 4 held-in datasets spanning 5 days, each with 53-61 minutes of calibration data, and 3 held-out datasets spanning 21 days, which have

only 1.1-2.2 minutes of calibration data available. We also release a second monkey, `M1-B`, for which there are 4 held-in datasets spanning 7 days (with 52-60 minutes of calibration data each) and 4 held-out datasets spanning 7 days (with 0.9-1.3 minutes of calibration data).

Source This dataset was collected by Adam G. Rouse with the support of Marc Schieber. The data was collected for the purpose of to dissociating the effects of location and object during reach-to-grasp behaviors. The experiment and data collection is described and features in a number of papers, including [199–202]. The dataset creators have granted permission to use and distribute the dataset sessions as part of the benchmark.

Intended use This dataset has been curated for evaluating stable decoding approaches as part of the FALCON benchmark. Muscle activations are a target for iBCIs through functional electrical stimulation [204] and are important scientifically for motor control as they are the direct output of commands sent from the central nervous system. The dataset is available on DANDI to allow others to evaluate their approaches on the data.

Experimental design The experimental task is to reach to, grasp, and manipulate 4 different objects at 8 different locations, arranged in a center-out fashion. The 4 objects are separated by 45 degrees with a fifth object in the middle. The center object is a coaxial cylinder, and the four peripheral (target) objects include: a button mounted inside a tube, a sphere, a perpendicular cylinder, and a coaxial cylinder identical to the center object. The trained manipulation schemes are as follows: cylinders are pulled towards the subject, the button is pushed, and the sphere is rotated 45 degrees.

The objects are arranged in a fixed order (perpendicular cylinder, coaxial cylinder, button, sphere) spanning 135 degrees of a circle. Objects were rotated to one of eight orientations in 22.5 degree increments (some positions excluded due to biomechanical or visual constraints). This leads to a total of 8 possible locations per object. Trials begin with the monkey pulling on the center cylinder and holding for 1500-2000ms. A blue light cues a pheripheral object, which the monkey needs to reach to, grasp, and manipulate. For a trial to be successful, the monkey complete these interactions

with the cued object within 1000ms and hold the object in the manipulated state for 1000ms.

Data collection methods The dataset contains neural activity of two rhesus monkeys implanted with 6 FMAs, 16 channels each, electrodes of length 1.5-8mm. We focus on 4 of these arrays (H, I, J, and K) placed in primary motor cortex (m1) that were consistently recorded from throughout the duration of this dataset. Recordings were sampled at 30kHz. Thresholds to extract spiking data were manually set for each channel and may vary from session to session. Intramuscular EMG was recorded from 16 muscles including: anterior deltoid (DLTa), posterior deltoid (DLTp), pectoralis major (PECmaj), short head of biceps (BCPs), lateral head of triceps (TCPlat), flexor carpi radialis (FCR), flexor carpi ulnaris (FCU), extensor carpi radialis brevis (ECRB), extensor carpi ulnaris (ECU), radial and ulnar flexor digitorum profundus (FDPr, FDPu), abductor pollicis longus (APL), extensor digitorum communis (EDC), thenar muscle group (Thenar), first dorsal interosseus (FDI), and hypothenar muscle group (Hypoth).

Processing For all sessions, thresholds crossings were computed in 20ms bins. We apply a standard set of preprocessing for the EMG data decoding target as follows: notch filter the signal at 60Hz and harmonics to remove line noise, high pass filter (acausal Butterworth filter, 4th order) with a 65Hz cutoff, rectify the resulting signal, clip the signal at the 99th quantile, scale the signal at the 95th quantile, resample to 50Hz, rectify again, and finally low pass filter (acausal Butterworth filter, 4th order) with a cutoff of 10Hz. In both held-in and held-out files, the last 40% of the data was reserved for evaluation. Remaining data is released for held-in sessions for calibration. Ten trials were released for each held-out calibration set.

Dataset documentation - `M2`

General description This dataset contains unsorted spike times and finger kinematics from a macaque performing an individuated finger control task. Neural activity was recorded from precentral gyrus. Finger position and finger velocity were also recorded.

`M2` includes 4 held-in datasets over 10 days, with between 5.9-13.3 minutes of calibration data per session available, and 4 held-out datasets over 26 days with

between 0.8-1.7 minutes of calibration data provided.

Source This dataset was collected by Samuel R. Nason-Tomaszewski, Matthew J. Mender, and Cynthia A. Chestek at the University of Michigan. The data was collected to study closed-loop individuated finger control with an iBCI. The experiment and data collection are discussed in [205]. The dataset creators have granted permission to use and distribute the dataset sessions as part of this benchmark.

Intended use This dataset has been curated for evaluating stable decoding approaches as part of the FALCON benchmark. Dexterous hand control is an important behavior that iBCIs aim to restore, and robust decoding approaches that work well with individuated finger movement behaviors may aid in this endeavor. The dataset is available on DANDI to allow others to evaluate their approaches on the data.

Experimental design The monkey is shown a virtual hand whose finger state mirrors that of a manipulandum. The manipulandum was designed to measure the monkey's finger state. Only two independent degrees of freedom are allowed in the manipulandum - the index finger and MRS group, which bundles middle, ring, and small fingers.

In trials, the monkey is shown visual cues (colored dots) to indicate a target finger state, and the monkey moves their fingers to the cued positions. The visual cues are colored corresponding to the colors of the fingers, indicating which finger group should be moved to each target. The monkey is trained to proficiency, and each trial is on the order of a second.

The finger and target positions are bounded to the range [0, 1], where 0 represents full extension of the finger group, and 1 represents full flexion of the finger group. The task begins with both targets at a central position (0.5), and the targets return to the central position every other trial (i.e. a typical center-out-and-back task paradigm). In between central targets, the task randomly selects from a set of target positions that moves one or both groups of fingers away from the central position. Whether one or both groups of fingers moves in a given trial is randomly selected, and the distance to which each group moves is also randomly selected (± 0.2, ± 0.3, or ± 0.4 from the central position). For trials in which both finger groups are instructed to move from the central position, the magnitude of instructed movement for both finger groups is

always equal, but the direction of movement is not always the same. There are no trials in which one group was instructed to move to +0.4 and the other group was instructed to move to -0.4 due to the difficulty in separating fingers that far.

Data collection methods The `M2` dataset contains neural activity of a rhesus macaque with two 64-channel Utah microelectrode arrays placed in the precentral gyrus, of which 96 channels are are provided. Recordings were originally sampled at 30kHz. Finger positions and velocities were recorded at 1kHz using actuator velocities measured by the manipulandum.

Processing For all sessions, we release threshold crossings in 20ms bins. We similarly resample the finger kinematics to 50Hz for consistency. In both held-in and held-out sessions, the last 40% of the trials were reserved for evaluation splits. The preceding 60% of held-in session data is released for calibration. For held-out sessions, the first 10% of each session is released as the calibration split.

Dataset documentation - `H1`

General description This dataset contains unsorted spike times and robotic actuator kinematics from a human iBCI participant during an open-loop attempted reach-to-grasp task. Neural activity was recorded from motor cortex. Behavioral covariates include the 7-degree-of-freedom kinematics corresponding to each cue. The 7 degrees of freedom are: 3 dimensions of translation, 1 dimension of rotation (roll), and 3 dimensions for grasp shaping. The grasp shaping dimensions are: pinching (flexion and extension of thumb/index/ring finger), scooping (flexion and extension of ring and pinky finger), and thumb abduction.

`H1` consists of 6 held-in sessions over 20 days (5-9 minutes of calibration data each) and 7 held-out sessions over 15 days (1.5-1.8 minutes of calibration data provided).

Source This data was collected by Sharlene Flesher, John Downey, Jennifer L. Collinger, and Robert A. Gaunt at the University of Pittsburgh as part of a clinical trial of iBCIs for sensorimotor control. The experiment and data collection protocol are described in [16, 36, 22]. Participant and date-time information have been obfuscated for deidentification. This data was collected under an Investigational Device Exemption from the U.S. Food and Drug Administration and is registered at

ClinicalTrials.gov (NCT01894802). The study was also approved by the Institutional Review Boards at the University of Pittsburgh and the Space and Naval Warfare Systems Center Pacific. Informed consent was obtained before any study procedures were conducted and included permission for data sharing. Dataset collectors granted their permission to use and distribute these sessions as part of this benchmark.

Intended use This data has been curated for evaluating stable decoding approaches as part of the FALCON benchmark. The `H1` dataset provides an example of a high degree-of-freedom behavior, which may pose specific challenges to robust decoding approaches. The dataset is available on DANDI to allow others to evaluate their approaches on the data.

Experimental design As this is an open loop task, the participant is asked to attempt to perform a movement cued with a virtual arm. The virtual arm movement occurs in phases: reach, grasp, carry, release. Each phase begins with a presentation of a combined visual and word cue for a particular movement, so the participant can prepare an imagined movement, and an another cue to execute the imagined movement. The participant has had practice following these cues to calibrate similar decoders before this dataset was collected.

Data collection methods Neural data was collected at 30kHz from two Utah arrays placed in the hand and arm region of motor cortex.

Processing For all sessions, we preprocess the neural data into 20ms bins. Robotic arm and hand kinematics were also downsampled to 50Hz to be consistent with the neural data. For both held-in and held-out sessions, we reserve the last 20% of each file for evaluation. All remaining data is released for held-in sessions, to be used for calibration. The first 20% of each file is released for each held-out session as the calibration split.

Dataset documentation - `H2`

General description `H2` contains unsorted spike times and sentence prompts from a of a BrainGate2 pilot clinical trial participant (referred as T5) during an open-loop attempted handwriting task. Neural activity was recorded from the hand "knob" area

of arrays placed in the precentral gyrus. Behavioral covariates for this task are not continuous but rather the cued sentence for each trial.

`H2` consists of 21 held-in sessions over 287 days (7-45 minutes of calibration data each) and 5 held-out sessions over 176 days (1-2 minutes of calibration data provided).

Source This data was collected by Chaofei Fan, Leigh R. Hochberg, and Jaimie M. Henderson at Stanford University as part of the BrainGate2 Neural Interface System clinical trial (ClinicalTrials.gov Identifier: NCT00912041, registered June 3, 2009) on iBCIs. The experiment and data collection protocol are described in [135, 178]. This pilot clinical trial was approved under an Investigational Device Exemption (IDE) by the US Food and Drug Administration (Investigational Device Exemption G090003). Permission was also granted by the Institutional Review Boards of Stanford University (protocol 20804). Informed consent was obtained prior to any study procedures being conducted. Dataset collectors granted their permission to use and distribute these sessions as part of this benchmark.

Intended use This dataset has been curated for evaluating stable decoding approaches as part of the FALCON benchmark. Restoring communication is a primary high-level goal of iBCIs, and the handwriting task is a prime example of a brain-to-text decoding scheme that is appropriate for participants with arrays placed in motor areas. The dataset is available on DANDI to allow others to evaluate their approaches on this data.

Experimental design On each trial, T5 was prompted to copy a cued sentence by writing individual characters (26 English letters, 4 punctuations, and a special ">" character to represent space). T5 was instructed to attempt to write as if his hand were not paralysed, while imagining that he was holding a pen on a piece of ruled paper. Each trial has a fixed-length delay period and a variable-length go period. During the delay period, T5 reads the cued sentence. Once the go cue is on, T5 starts to copy the sentence letter by letter and uses a verbal cue to indicate he has finished copying. In each session, both open-loop and "pseudo-closed-loop" trials were collected. In open-loop trials, the participant only sees the cued sentence; in pseudo-closed-loop trials, the real-time decoded letters are shown.

Data collection methods Data was collected from two 96-channel intracortical Utah arrays placed in the hand "knob" area of the participant's left hemisphere precentral gyrus. Original neural recordings were collected at 30kHz.

Processing For all sessions, neural data was binned at 20ms. We provide only data from within each trial period, as inter-trial time could be long and lead to unnecessarily large data files. Each trial is accompanied by the cued sentence. For both held-in and held-out sessions, 40% of the trials are reserved for evaluation. For held-in sessions, remaining trials are released for calibration. For held-out sessions, only 3 trials are released for few-shot calibration.

Dataset documentation - `B1`

General description The `B1` dataset contains unsorted spike times, audio recordings, and audio spectrograms from a zebra finch songbird during natural vocal behavior. Neural activity was recorded using Neuropixels 1.0 probes from the motor region robust nucleus of the arcopallium (RA). The dataset includes a precomputed spectrogram derived from the recorded amplitude waveform corresponding to each vocal epoch, which constitutes the decoding target for this dataset. Due to the inherent non-determinism in spectrogram computation, we include the computational function used to derive the spectrograms. Additionally, the raw amplitude waveform is available for use in decoding strategies that might prefer it as a target before conversion to spectrogram form.

`B1` consists of 3 held-in sessions corresponding to 3 consecutive days of recordings, with 7-26 seconds of calibration data per session available, and 3 held-out sessions recorded over the following 6 days, with 2.7 seconds of calibration data per session provided.

Source This dataset was collected by Pablo Tostado-Marcos, Ezequiel Matias Arneodo, Timothy Q. Gentner, and Vikash Gilja. The original experiment and procedures are described in [209]. Data collectors granted their permission to use and distribute these data as part of this benchmark.

Intended use This dataset has been curated for evaluating stable decoding approaches as part of the FALCON benchmark. The dataset offers an alternative to

brain-to-text iBCIs by focusing on the direct reconstruction of audio spectrograms from neural signals, thereby encouraging decoding approaches that preserve the prosodic elements of vocal behavior. The `B1` dataset proposes the songbird model as a proxy for human vocalization, supporting the development of neuroprostheses aimed at restoring communication capabilities and advancing stable-decoding methods. This dataset is available on DANDI to allow others to evaluate their approaches on these data.

Experimental design An adult, male zebra finch songbird was implanted with a single high-density Neuropixels 1.0 probe targeting the telencephalic motor region RA in the right brain hemisphere. Simultaneous neural and behavioral (song) data were collected in a single-housing acoustically-isolated chamber during awake-singing. The bird was allowed to move and sing freely during 120-240 minute-long recording sessions.

Data collection methods Neural data and vocal behavior were collected simultaneously. Voltage signals were recorded by 384 Neuropixels channels, amplified, band-pass filtered (300Hz-10000Hz), multiplexed and digitized at 30kHz on the Neuropixels headstage, and transferred to the data acquisition module. We focus on 85 channels in the `B1`dataset corresponding to region RA. Audio signals were recorded at 25kHz and high-pass filtered (250Hz) for subsequent extraction of spectral features. The stereotypy characteristic of zebra finch song enabled the segmentation of non-overlapping sequences of song syllables, or motifs, composing the bird's own song. These motifs are similar in their syntactic structure but vary in timing, pitch and syllable count across vocal renditions. Custom software was used for extracting song motifs from the audio recordings and for computing the spectrogram representations to be used as decoding targets.

Processing To allow maximum flexibility in decoder design, we provide threshold crossing activity at the original 30kHz resolution. Thresholds were independently set on a per-channel basis and may vary across sessions. The amplitude waveform corresponding to each motif rendition, synchronized to neural data, is also provided at the original 25kHz sampling rate. The audio signals provided were band-pass filtered within the relevant birdsong frequency range (250Hz-8000Hz) and de-noised using the *noisereduce* Python package [300]. The spectrogram corresponding to

each amplitude waveform, which constitutes the ultimate decoding target, is provided at 1kHz resolution. We provide only a window of data around each 700ms-long motif (100ms before motif onset and 100ms after the end of the motif; total epoch length is 900ms). For both held-in and held-out datasets, 40% of the song motifs are reserved for evaluation. On held-in datasets, the remaining 60% of the data is released for calibration. On held-out datasets, 3 motifs are made available for the few-shot calibration split.

B.1.4 Baseline Implementation Hyperparameters

Wiener Filter (WF)

Description For the movement datasets, linear static and oracle decoders are done using a Wiener filter. Wiener filters predict the current value of an output signal using previous timesteps, as defined by:

$$y[t] = \sum_{i=0}^{I-1} w_i x[t-i]$$

where $y[t]$ is the output signal at time t, $x[t]$ is the input signal at time t, w_i is the filter coefficient, and I is the number of previous samples to use for decoding. In our decoder, the input signal x is the smoothed neural data, y is the behavioral output to predict, and I is the number of time bins of history. The weights are fit using a matrix formation of the above equation:

$$W = (X^T X + \lambda I)^{-1} X^T y$$

where W is a matrix of filter coefficients, X represents the predictor data with history and bias, and y represents the output signal. λI represents a diagonal matrix with the L2 regularization constant filling the diagonal. The bias term is not regularized and therefore its diagonal entry is set to zero. The L2 regularization aims to avoid decoder overfitting by penalizing solutions with large individual weights.

Implementation We implement the WF as a Ridge Regression model using the Scikit-learn library [301].

Parameter Optimization L2 regularization values are obtained using 5-fold cross validation. We sweep a range of 20 values spanning 1e-5 to 1e5 in logspace. For each value, we train and test a Wiener filter using 5-fold cross validation, testing the decoder on a held-out fold. The optimal regularization value was selected based on which value yielded the highest performance metric. Final performance was reported on the held out fold.

To determine the number of bins of history to use for each dataset, we swept from $I = 0$ to $I = 30$ by training on all held-in calibration and held-out oracle data splits and reporting performance on the evaluation split for each dataset, separately. We selected the appropriate number of bins by plotting the resulting R^2 for each session and choosing the value that coincided with the elbow for the most sessions within a dataset. These results are shown in **Figure B.1**. We ultimately selected I=30 bins (600ms) for `M1`, I=7 bins (140ms) for `M2`, and I=30 bins (600ms) for `H1`.

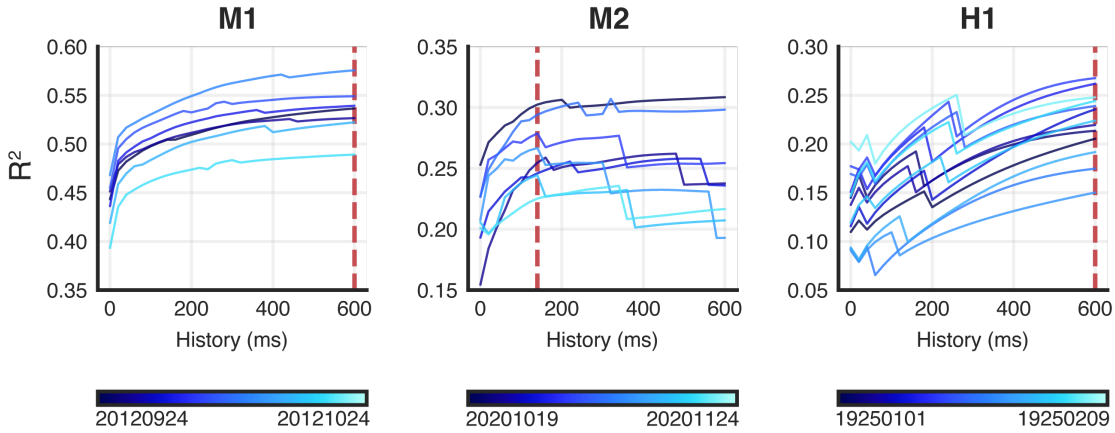

Figure B.1. Sweeps to determine WF history. We train WF decoders on each session (differentiated by shades of blue, see colorbar) of all datasets with history from $I = 0$ms to $I = 600$ms and evaluate the prediction R^2 on the held-out split. We select the history that maximizes performance for most sessions, indicated with the vertical red dashed line. For `M1` and `H1`, performance continued to rise with increased history. We capped history at 600ms to ensure reasonable use for iBCIs [302, 303], but selected this maximal value for these datasets. For `M2`, performance reached an elbow with 140 ms of history.

Code Availability The WF decoder used for FALCON is available on our Github repository at: https://github.com/snel-repo/falcon-challenge/blob/main/decoder_demos/sklearn_decod

RNN Decoder - Movement

Description The RNN baseline uses a 1-layer LSTM followed by a linear readout of behavior. It is a simple supervisd baseline.

Implementation This minimal baseline excludes any multi-session layers, and thus was only trained on single sessions of data to report oracle and zero-shot results. It is implemented within the NDT2 codebase, but uses simple Pytorch layers.

Parameter Optimization We sweep learning rate and model hidden size, though find training varies little with these choices. We sweep 6 parameter combinations per model. Models train very quickly (2-10 minutes on a single 2080 NVIDIA GPU).

Code Availability RNN baseline code is available in the NDT2 codebase: https://github.com/joel99/co

Neural Data Transformer 2 [179]

Further results We present additional NDT2 baselines in **Table B.1**.

Table B.1. FALCON movement baselines with zero-shot and static NDT2. We include NDT2 zero-shot and static results for comparison with the RNN baseline. Oracle single-session models and zero-shot transfer achieve comparable results with the vanilla RNN decoder.

Movement (Held-Out R^2 / Held-In R^2 ↑)				
	Class	M1	M2	H1
Wiener Filter (WF)	OR	$0.53_{\pm 0.04}/0.54$	$0.26_{\pm 0.03}/0.27$	$0.21_{\pm 0.04}/0.24$
RNN	OR	$0.75_{\pm 0.05}/0.75$	$0.56_{\pm 0.04}/0.59$	$0.44_{\pm 0.13}/0.51$
NDT2	OR	$0.71_{\pm 0.06}/0.72$	$0.44_{\pm 0.08}/0.53$	$0.38_{\pm 0.13}/0.44$
NDT2 Multi	OR	$0.78_{\pm 0.04}/0.77$	$0.58_{\pm 0.04}/0.62$	$0.63_{\pm 0.08}/0.68$
WF	ZS	$0.34_{\pm 0.06}/0.46$	$0.06_{\pm 0.04}/0.15$	$0.16_{\pm 0.03}/0.20$
RNN	ZS	$-.61_{\pm 0.48}/0.51$	$0.13_{\pm 0.09}/0.17$	$0.08_{\pm 0.17}/0.29$
NDT2 [179]	ZS	$0.11_{\pm 0.11}/0.55$	$-0.03_{\pm 0.15}/0.28$	$0.10_{\pm 0.10}/0.32$
NDT2 Multi [179]	FSS	$0.59_{\pm 0.07}/0.77$	$0.43_{\pm 0.08}/0.63$	$0.52_{\pm 0.04}/0.62$

Description NDT2 is a Transformer-based deep neural network previously demonstrated to enable multi-context neural data modeling with or without any specific parameters for different datasets. The model tokenizes each timestep of the input

data into contiguous subsets of fixed length along the full channel dimension. As there are multiple tokens per timestep, the multiple input tokens must be merged to produce per-timestep decoding. In this work, the NDT2 baselines use cross-attention for behavior decoding. Additionally, the models perform neural data reconstruction as a secondary objective.

Implementation NDT2 models are prepared as the other single-session baselines, i.e. separate models are trained per session and the model that performs best on held-in data is used for zero-shot transfer. NDT2 Multi models use all available calibration data at once, e.g. for the oracle NDT2 Multi model, a single model was trained with all held-in calibration, held-out calibration, and held-out redacted data; the regular NDT2 Multi model uses all calibration data. Thus NDT2 does use the few-shots of calibration data available in future sessions for predicting an early held-out session; this is an implausible design for real-world use, chosen for simplicity.

As described in Section B.1.5, the training dataloader was modified to provide random fixed length subsets over either training on trialized data or direct splitting of the continuous data. This is required on `M1`and `M2`for the model to be more robust to FALCON's continuous evaluation.

Parameter Optimization NDT2 used a fixed grid search on model parameters and learning rate for each of `M1`, `M2`, `H1`, and trained with early stopping. The checkpoint with best validation score was used to compute the baseline metric. Individual runs (6 runs per sweep) cost up to 3 hours per model (for `M1`data) on a single NVIDIA 2080 GPU.

Code Availability The codebase and checkpoints, along with the specific hyperparameter sweeps, are on the public Github repo: https://github.com/joel99/context_general_bci/.

NoMAD [46]

Description Nonlinear Manifold Alignment with Dynamics (NoMAD) is a manifold alignment decoder stabilization approach that operates in a few-shot unsupervised regime. NoMAD frames the stabilization problem using pairs of datasets consisting of an initial calibration dataset, "Day 0," where both neural and behavioral data are

available, and a later dataset that contains neural nonstationarities with respect to Day 0, termed "Day K," for which only neural data is available.

On Day 0, a dynamics model is trained, and a decoder is trained to map from the inferred dynamics to the behavior. The model and decoder are then frozen. On Day K, using neural calibration data, a feedforward alignment network is trained to map the new neural data onto the fixed dynamics model. This alignment network is trained primarily using a Kullback-Leibler divergence cost between the RNN states of the dynamics model on Day 0 and Day K. After alignment network training, the fixed decoder can be applied to the Day K dynamics to maintain accuracy.

Implementation The Day 0 dynamics model is latent factor analysis via dynamical systems (LFADS) [304, 87]. Unlike the model with the behavioral readout described in [46], the LFADS models here are the standard nonautonomous models described in [87, 305, 306]. Of note, we select only one Day 0 dataset from the held-in sessions for each task; for `M1` , this is the `20120926` session, for `M2`, `2020-10-19-Run2`, and for `H1`, `19250113T120811`. The remaining details are consistent with [46]. The decoder applied is a Wiener Filter with 4 bins of history for `M1`and `M2`; for `H1`, we used 4 bins of history for most held-in sessions and 8 bins of history to evaluate the Day 0 session that NoMAD aligned to as well as all held-out sessions.

Parameter Optimization Day 0 LFADS model parameters were optimized using AutoLFADS [305, 306]. AutoLFADS was trained using 8 NVIDIA GeForce RTX 2080 Ti GPUs with training time less than one hour per held-in dataset. NoMAD alignment network hyperparameters were optimized using a random search, with each model in the random search using one NVIDIA GeForce RTX 2080 Ti GPU (training time approx. 20 minutes). Wiener Filter bins of history were optimized using a grid search. Models and decoders were selected based on those which had the highest accuracy on the held-out calibration data.

Code Availability Because [46] is still under review, the Systems Neural Engineering Lab will not release the code right now. The code will be made available when this paper has been published.

CycleGAN [143]

Description Cycle-Consistent Adversarial Network (CycleGAN) is a decoder stabilization approach based on the use of Generative Adversarial networks (GANs) that, like NoMAD, operates in a few-shot unsupervised regime. Similar to a conventional GAN, CycleGAN architecture consists of a pair of neural networks, a generator and a discriminator. The generator (or aligner) is trained to transform the high-dimensional neuronal firing rates from the calibration "Day K" dataset into a form resembling the initial "Day 0" dataset, which was used to train the fixed neural-to-behavior decoder. The discriminator is trained adversarially to the generator to maximize the distance between the distributions of Day K and Day 0 datasets. Unlike regular GANs, CycleGAN also implements a cycle-consistent loss that regularizes the learning of the Day K to Day 0 mapping function, thereby reducing the search space and making training more stable. This is achieved by adding a second pair of generator and discriminator networks that are trained to learn the opposite transformation (i.e., from Day 0 to Day K). After CycleGAN training, the aligner is used to transform Day K into Day 0 data to maintain the accuracy of a fixed Day 0 decoder.

Implementation Here we select only one Day 0 dataset from the held-in sessions for each task to compute the fixed decoder and the subsequent Day 0 to Day K CycleGAN aligners. (`20120926` session for `M1` , `2020-10-28-Run1` for `M2`, and `19250120` for `H1`). The decoder used is a Wiener Filter with 8 bins of history for all the tasks.

Parameter Optimization Training CycleGAN is computationally efficient since the generator and discriminators pairs are feedforward neural networks. The training process can be completed on any modern CPU in under two minutes. We used the same hyperparameters as described in [143]. The selection of CycleGAN aligners and decoders was based on those that achieved the highest accuracy on the held-out calibration data.

Code Availability A step-by-step tutorial on the use of Cycle-GAN for neural alignment can be found on the public Github repo: `https://github.com/limblab/adversarial_BCI/blob/main/Cycle_GAN_aligner.ipynb`.

RNN Decoder and Language Models - Communication (H2)

Description The baseline RNN decoder for H2consists of a shared Gated Recurrent Unit (GRU) backbone and a set of session-specific affine transform layers. The neural activity first undergoes an affine transformation via the session-specific layer, and then the GRU backbone decodes the transformed neural activity into characters. The GRU and session-specific transforms are trained end-to-end on multiple sessions.

The language model (LM) evaluates a sentence's probability. Given the character probabilities output from RNN, we use beam search and an LM to find the most likely sentence.

Implementation The baseline RNN is a 2-layer GRU with 512 hidden units. A softmax layer maps the GRU outputs to character probabilities. The session-specific layer is implemented as a linear layer with the same number of units as the input neural features.

The baseline RNN is trained jointly on multiple sessions to maximize performance. The zero-shot/static model is trained on calibration splits of all held-in sessions. The oracle model is trained on oracle splits of held-out sessions (not including sessions later than the testing day) plus the calibration splits of all held-in sessions.

We use a 3-gram LM trained on the OpenWebText2 corpus to convert the RNN outputs into words in real-time. To further improve the accuracy, we use GPT2-XL to rescore the outputs from the 3-gram LM. See details in [178].

Parameter Optimization Hyperparameters were optimized by grid search in [178]. No additional optimization is done for this work.

Code Availability Code and pre-trained models are available here: `https://github.com/cffan/CORP`

CORP [178]

Description CORP leverages language models (LMs) for test-time adaptation. It first uses LMs to correct errors due to nonstationarity in decoded sentences. The corrected sentences are used as pseudo-labels to calibrate the RNN. The calibration runs after the user finishes writing a sentence.

Implementation We use the same RNN and LM implementations as in B.1.4. For FALCON, we first trained a seed model on calibration splits of held-in sessions. Then, for each held-out evaluation session, we take the calibrated model from the previous session (seed model for the first evaluation session), use the model to decode trials from the new sessions, and run calibration after decoding.

Parameter Optimization Hyperparameters were optimized by grid search in [178]. No additional optimization is done for this work.

Code Availability Code and pre-trained models are available here: `https://github.com/cffan/CORP`

EnSongdec [209] - `B1`

Description The EnSongdec model is a brain-to-song deep neural network that enables synthesis of an amplitude waveform from input brain activity. The model features a feed-forward neural network (FFNN) trained to predict audio (song) embeddings, z, from each timestep of input neural data. The FFNN is coupled to a Quantizer-Decoder network extracted from a pre-trained, state-of-the-art audio codec (EnCodec) [210]. The quantization layer converts z into a compressed latent representation, z_q, utilizing residual vector quantization (RVQ). The decoder network uses transposed convolutions to reconstruct the time-domain audio signal at its original sampling rate.

Implementation We first used the Encoder network of a pre-trained EnCodec model to extract meaningful embedding representations of birdsong. To prioritize reconstruction quality over data compression and streamability, we opted for minimal EnCodec audio compression settings (24kbps at a 48kHz upsampled input). Next, we optimized custom feed-forward neural networks to translate input neural signals into continuous embedding representations of birdsong. Threshold-crossing inputs were smoothed using a 1-d Gaussian kernel ($\sigma = 30$) and downsampled to match the rate of the target song embeddings (150 samples per second; 7ms bins). 14ms of neural data (2 bins) were used to predict each audio embedding sample. The resulting feed-forward neural network featured an input layer of size $i = N \times history_bins$ (where $N = 85$ denotes the number of Neuropixels channels targeting the brain region

of interest), two 64-unit hidden layers and a 128-unit output layer corresponding to the dimensionality of the embedding space. ELU activation functions were employed and mean square error (MSE) was used as the reconstruction loss to train FFNNs. The Quantizer-Decoder network excised from the aforementioned pre-trained EnCodec model was coupled to the FFNN to synthesize a continuous time-domain song signal. The spectrogram of the reconstructed song was compared to the spectrogram of the original recorded song to evaluate the performance of EnSongdec.

Parameter Optimization Hyperparameters were optimized using a grid search approach based on minimal song reconstruction error. We used Weights & Biases for experiment tracking. Models were trained on a proprietary cluster of servers using NVIDIA GeForce RTX 2080 Ti GPUs.

Code Availability Code to implement EnSongdec can be found in the public Github repository: https://github.com/pabloslash/EnSongdec

B.1.5 Evaluation Parameters

Continuous vs Trialized Evaluation for Motor Tasks

Real-world BCIs will require continuous decoding of user intention. This motivated us to design FALCON evaluation to be continuous, despite the fact that the evaluation data was often collected in a trialized setting. When making this design choice, we identified a sensitivity to training and evaluation context length that varied across datasets. Specifically, we trained trialized NDT2 decoders, where training data is divided into single behavioral trials, and continuous NDT2 decoders, where the continuous training data is divided into fixed length segments. Trialized models used up to 4 seconds of history (`M1` and `H1` had trials in excess of this length, but `M2` trials averaged to about 1 second. Further, continuous models did not directly train on data split into segments of a given length, but rather required augmentation. Data was split into segments longer than the target length; e.g. for 2 seconds of history, data might be split into 4 second segments, and a 2 second slice was drawn at random. Decoders of either kind were evaluated in a trialized setting, where signals about trial change could be used to reset model input, or in a continuous setting, where these signals were not available.

Fig. B.2 illustrates the sensitivity of trialized models to continuous evaluation. Trialized models often performed well in trialized evaluation, and in particular did not degrade with long histories. However, when evaluated in continuous settings, performance dropped precipitously in `M1` and `M2`. In `H1`, models trained on trialized data continue to improve with higher histories, on either trialized or continuous evaluation.

In contrast, models trained on continuous data fail with long histories, but peak performance was often comparable to peak trialized performance. Accordingly, NDT2 baselines for `M1` and `M2` trained with continuous data and with trialized data for `H1`.

The dependence of trialized training on trialized evaluation suggests that models are exploiting trial structure (distinct behavior at the start, middle, and end of trials) to reduce uncertainty about decoding at different timepoints. This is likely to not benefit closed loop control, where decoding should be able to produce flexible behavior at any timepoint. Since continuous models are also able to achieve similar performances, it is possible that continuous models are still exploiting trial structure by inferring the part of the trial that needs to be currently decoded. Moreover, the fact that performance continues to improve with longer context for `H1` remains a particularly concerning edge case that may be exploited in FALCON leaderboards. We do not restrict `H1` context in evaluation for simplicity, but encourage works to report the history they use as input for context. We also encourage work that sheds light on why `H1` behaves differently than `M1` and `M2`.

Determining data volumes for few-shot calibration on held-out days

FALCON aims to enforce the realistic constraint that calibration on new sessions will have limited neural and behavioral data. To ensure that the few-shot problem was well-represented, we established that the held-out calibration splits were insufficient to train new linear decoders on their own. As shown in **Table B.2**, performance of WF decoders on the movement datasets trained on the held-out calibration splits under-performs those trained on the held-out oracle splits.

Table B.2. WF decoder performance on held-out data splits. WF decoders trained on held-out calibration and held-out oracle splits for all movement datasets.

Training Data	`M1`	`M2`	`H1`
Held-out calibration	$0.24_{\pm 0.04}$	$0.14_{\pm 0.05}$	$0.11_{\pm 0.03}$
Held-out oracle	$0.53_{\pm 0.04}$	$0.26_{\pm 0.03}$	$0.21_{\pm 0.04}$

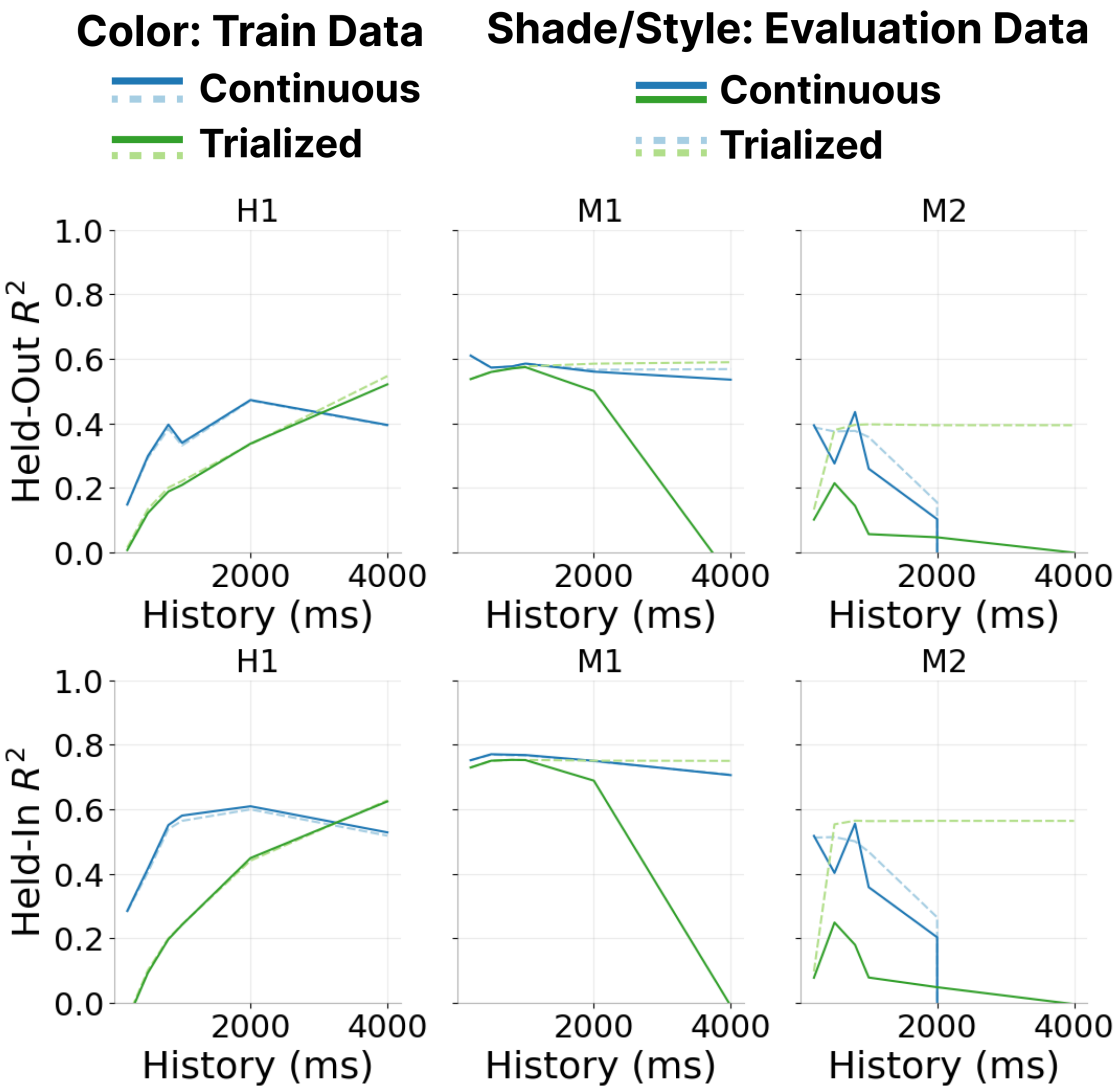

Figure B.2. Continuous vs Trialized Decoding. We train and evaluate NDT2 decoders on combinations of trialized and continuous data. History indicates the length of context the model uses to make predictions during evaluation. These decoders are sensitive to data trialization, with varying effects across datasets. Models trained with trialized data in `M1`and `M2`fail significantly when evaluated in a continuous fashion (indicating dependence on trial structure during decoding). Continuous models can often match the performance of trialized models in either trialized or continuous evaluation, but is sensitive to the length of history used. `H1`, unlike the other datasets, sees continued gains with increased context beyond the explored range even under continuous evaluation, indicating further possibility of model exploitation of trial structure. Held-in metrics show similar trends. FALCON is susceptible to models that exploit these gains.

Appendix C

Supplement to Chapter 5 (NDT3)

C.1 Appendix

Contents

C.1.1 Related works and a proposed taxonomy 142
C.1.2 Supplementary Results 144
Further tests of held-out angle generalization. 144
Ablations 144
Isolated scaling in neural data 147
Pretraining does not benefit FALCON H2 (Handwriting) . . . 149
Multiscale decoding on individual motor tasks 150
Sequential tuning is similar to joint tuning 152
Aggregate performance on all NDT3 models with significance tests 153
Neural vs Behavioral Objectives 154
Downstream probe through pretraining 155
Evaluation on the Neural Latents Benchmark 156
Neural objectives are needed in fine-tuning 156
C.1.3 Methods 157
Metrics and Evaluation 157
Training 158
Baselines and Other Transformers 159
Pretraining and Evaluation datasets 163

Generalization analyses and Further evaluations 164
Architectural Details 165

C.1.1 Related works and a proposed taxonomy

[1]

Neural data is sufficiently diverse so as to support many distinct efforts to train large neural data models. The scale of pretraining is somewhat larger in the non-implanted modalities, where data is more abundant. The largest EEG models have reached a scale of 2.5K [101] to 40K hours of data [102], or higher volumes if also considering non-brain biosignals (EKG) [307, 308]. Current fMRI models operate in the 1K [103] to 7K [104] hour range. The largest models in these studies are in the 0.1B-1B parameter range. Intracranial modalities, including sEEG [100, 105], ECoG [131, 132], and spiking activity [220], have thus far been studied at an order of magnitude smaller scales of data and model size (20-1000 hours, <0.1B parameters).

Direct scaling on neural data modeling should be distinguished from NeuroAI efforts [109] to measure how models of the human sensorimotor experience (e.g. language, vision, audio models) predict neural data [108]. However, as multimodal efforts begin to blur this distinction [110, 111], care will be required to distinguish advances in modeling neural data, embodied data, or their interaction.

Comparing neural data models Current efforts to understand scaling in neural data Simeon et al. [309], Sato et al. [310] will have their reach limited by the specificity of every neural dataset. A meta-challenge for the field is understanding how different parameters (species, brain area, modality, task) impact scaling properties. This would be greatly aided by development of reporting practices for different neural data models. To facilitate comparison, we create a model card in the NDT3 codebase [311]. In addition to the standard model card, we propose reporting an additional taxonomy to aid comparisons across neural data models, using two concepts.

First: neural data models can be conceptualized as modeling slices of the *plenneural function*, inspired by the plenoptic function in vision [312]. The plenoptic function is a model of an idealized eye which parameterizes all possible images with 7 dimensions: 4D to describe the global spacetime of the view, 2D to describe viewing angle (spherical) or coordinate (Cartesian) of the image, and 1D for wavelength. Since neural data

[1] This subsection surveys the landscape as it stood when Chapter 5 was written, and the reported scales have since been overtaken. See Section 2.2.2 for the current picture.

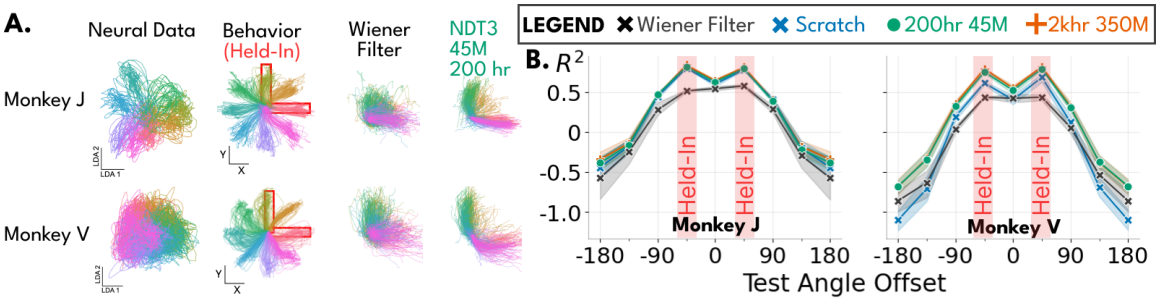

Figure C.1. A. Two monkey 2D center-out datasets with 8 angular conditions. LDA projections show monkey J's data is distinctly more separable than monkey V's. We then test generalization of behavioral decoding to held-out angles after training on 2 of 8 angles (boxed in red). All NDT3s and the WF produce predictions constrained between the held-out angles. The Wiener Filter here, unlike in the main text, is a direct fit on neural data without dimensionality reduction. **B.** We quantify performance for decoding on each angle with respect to distance from the central angle between the held-in angles. We average performance over all 8 central angles.

models are primarily interested in circumscribed systems rather than the physical world, a similar global coordinate system (e.g. 4D for all possible electric potentials) would be uninformative. We thus propose reporting more qualitative coordinates:

1. Identity: The network or individual being recorded.
2. Task: The behavior, stimuli, or other activity the network is reflecting.
3. Spacetime: Coordinates specified in a network-local coordinate frame (e.g. brain area).

Second: The modeled extent of this plenneural function is conveniently discretized in three resolutions in a Transformer-like sequence modeling framework: the token, the sequence, and the full training data. The token is the most granular unit of data being modeled; NDT3 models neural populations 32 neurons at a time, in 20ms bins. At the sequence input level, NDT3 models inputs from single humans or monkeys, across 128-256 neurons in 2 second snippets, while performing effectively one "movement." Finally, NDT3's pretraining spans dozens of individuals, records motor and premotor areas over 2.5K hours, over a variety of arm and hand movements.

C.1.2 Supplementary Results

Further tests of held-out angle generalization.

To further support our single-dataset illustration of attractor structure in pretrained NDT3, we evaluate reach angle generalization across sessions in the isometric, force-based dataset (Monkey J) setting and a second, manipulandum-based (Monkey J) setting with monkey movement. To begin, we visualize the separability of these neural data with LDA (Fig. C.1A Neural Data). We next train decoders on every pair of angles separated by 90 degrees (one shown) and plot predictions on held-out trials from all angles. NDT3 and WFs, here directly fit to high-D data instead of after PCA-LDA, both fail to extrapolate to held-out angles, consistent with Rizzoglio et al. [157], illustrating the necessity of the dimensionality reduction for generalization in the main text. We quantify prediction performance in Fig. 5.4D. These plots again show that held-out generalization is subtrivial, while being entirely consistent with pretraining's overall narrative of improved performance in all conditions.

Methods for PCA-LDA. Our PCA-LDA preparation used in Fig. 5.4C-E adheres closely to our standard data preparation and is directly comparable to the inputs received by the deep networks. To begin, we smooth all spike counts with an exponential kernel, as in our Wiener Filter baseline (Section C.1.3). We then z-score these smoothed spikes, and fit PCA to the high-dimensional (96D) z-scored neural activity at each timestep to extract the top 10 PCs. Finally, we fit linear discriminant analysis (LDA) to reduce this 10D PC space to neural activity to 2D. Example neural data projections are shown in Fig. C.1A, showing clear radial structure that can be exploited for generalization. The training data for PCA and LDA are both restricted to only the held-in angles. Note that LDA uses categorical labels to separate the 3 held-in reach directions but is applied without class labels after fitting. While this train-time labeling is technically not given to NDT3, this does not influence our argument that NDT3 pretraining should be capable of angular generalization, as NDT3 does not confuse the reach angle of held-in predictions. The final Wiener Filter is fit to the 2D data and uses 1 second of history.

Ablations

We ablate the major design decisions made to enable NDT3's large scale pretraining. These ablations give us confidence that NDT3 overcomes the basic challenges we

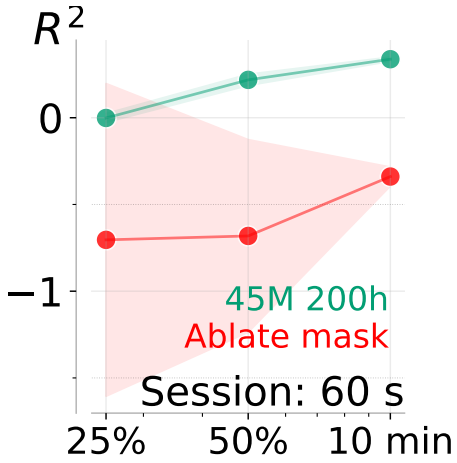

Figure C.2. Ablation of covariate masking on an open 2D Cursor + Click dataset. Covariate inputs are completely masked in inference for the default NDT3, and autoregressively generated in the ablation.

encountered in development, but compute restrictions prevent more exhaustive comparisons or exploration of model design space. We encourage further work exploring the influence of different hyperparameters. In these plots, we distinguish validation split performance and evaluation split performance, which is computed by batch-mode prediction (not the costly streaming evaluation used throughout main experiments).

Covariate dropout We find the default next-step prediction objective fails for learning decoding of highly autocorrelated covariate timeseries, perhaps because simply relying on teacher-forced behavioral inputs provides a severe shortcut that prevents learning of a proper neural to behavior decoding map [313]. Different time-series models have addressed this by adopting convolutional input-output layers [314], tokenizing along temporal dimensions [315], or learning with contrastive objectives [105, 99]. We avoid introducing architectural modifications and instead adopt a simple dropout procedure that masks a portion of covariate inputs some fraction of the time. Specifically, on every training batch, two random numbers are drawn. The first, $M \sim U[0, 1]$, determines what fraction of covariate inputs should be masked. On 90% of batches, we also sample $T \sim U[0, 2]$ seconds, such that the mask is only applied after timestep T. That is, on 90% of batches, the model is provided a prefix-prompt. We do not block losses on this prefix as in prefix-LMs [316]. Pretraining metrics for validation and evaluation are always computed with a prefix and full masking of non-prefix timesteps. In Fig. C.2, we ablate covariate masking (which also removes the prefix logic), and tune on a 2D Cursor + Click task. The ablated model performs subtrivially with

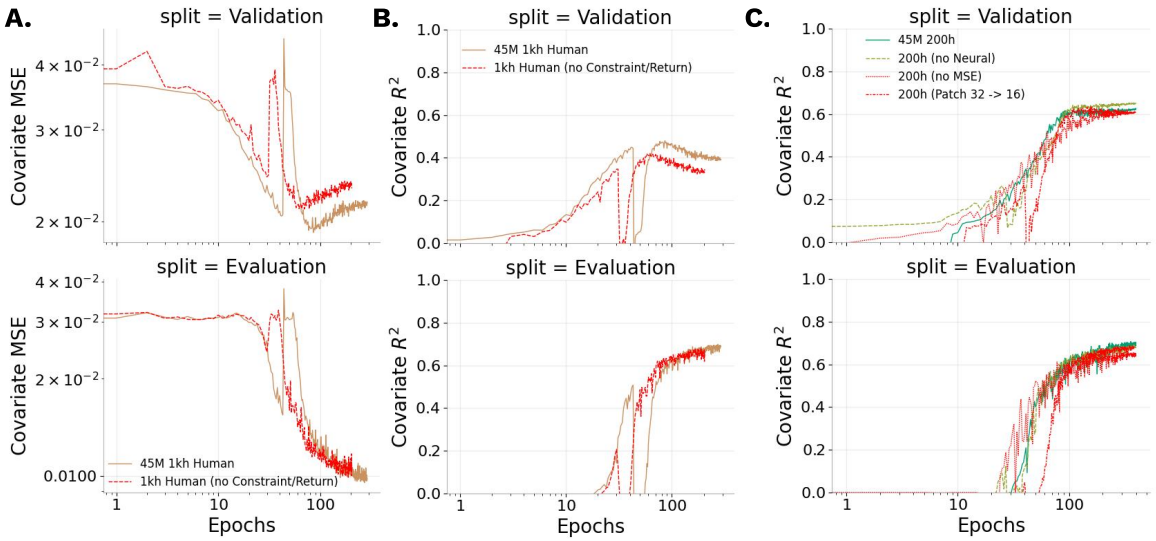

Figure C.3. Ablations evaluated based on upstream evaluation split. **A., B.** Ablation of BCI control tokens **C.** Ablation of neural objective and covariate MSE objective in favor of classification over quantized covariates.

student-forced predictions provided as input at test time. Note that the ablated model performs trivially with masked inputs (not shown).

BCI-phase and return conditioning NDT3's pretraining includes several hundred hours of BCI control data, where the covariates were set by another decoder. We introduced phase and return conditioning tokens to differentiate the several types of BCI control data from recorded behavior. Specifically, in BCI data, NDT3 receives input tokens specifying what fraction of the behavior reflects neural input (BCI control is on) vs programmatic input (BCI control is off, as in open loop BCI calibration). Further, we provide inputs encoding reward (trial success) when trials change, and return (future reward over a 10 second horizon, which crosses data boundaries). This design is intended to evaluate the potential for a Decision-Transformer like offline learning strategy for improved online control, but we do not discuss this in this work. In Fig. C.3A, B, we focus on whether these inputs improves pretraining loss and R^2 in validation splits, which has contains BCI data, and the held-out evaluation split containing only monkey behavior. The figures show that the ablation significantly decreases validation split performance, and causes a slightly earlier stopping point leading to worse evaluation performance. Note both models early before the full training budget of 400 epochs.

Neural reconstruction objective All main NDT3 models used a neural recon-

struction objective inherited from the self-supervised learning pretraining from NDT2. We ablate this choice post-hoc and see it may actually minorly harm pretraining (validation split), though the neural objective doesn't harm evaluation split decoding (Fig. C.3C). Note the scalar weighting of neural vs covariate objectives were set to be roughly balanced in pretraining. Section C.1.2 provides a downstream analysis on the standalone value of the neural reconstruction objective.

MSE over classification In robotics and certain generalist models [223], continuous action spaces are sometimes better decoded and controlled when quantized [317]. This is because MSE is an insufficient objective when the output distribution is multimodal (e.g. one of two possible paths in robotics). While it seems unlikely that the close relationship between movement behavior and motor cortex is multimodal, multimodal behavior may be appropriate when pretrained on heterogeneous data, i.e. when similar neural activity corresponds to different behavior in two datasets. We attempted such a quantization, including HL-Gauss smoothing [318] which we found to help; but this does not recover the performance of the default MSE objective (Fig. C.3C) on the evaluation split. We found this performance gap persisted under fine-tuning (not shown). This suggests that NDT3 is differentiates neural data inputs from different datasets.

Patch size NDT2 and NDT3 both tokenize neural data by patching them into fixed size clusters. It is unclear whether transfer learning might occur for sub-token features, which motivates the use of smaller tokens in larger datasets that might afford it [319]. We change patch size to 16 and show this performs slightly worse in the 45M 200h model (Fig. C.3)C. Smaller patches (and subsequent increased neural tokens) may be more beneficial in the larger scale models, but their benefit must be weighed against their increased compute burden.

Isolated scaling in neural data

Due to NDT3's joint modeling of behavior and neural data, it is difficult to dissociate whether scaling gains in behavior come from improved behavioral or neural priors. To assess whether NDT3 can scale solely from neural data modeling, we pretrain a new set of 45M parameter models up to 200 hours with only causal neural data modeling objective. As before, we then tune to a downstream decoding task, in this case, a Critical Stability Task dataset (CST). From a representation probing perspective, improved downstream performance implies higher quality neural representations. We

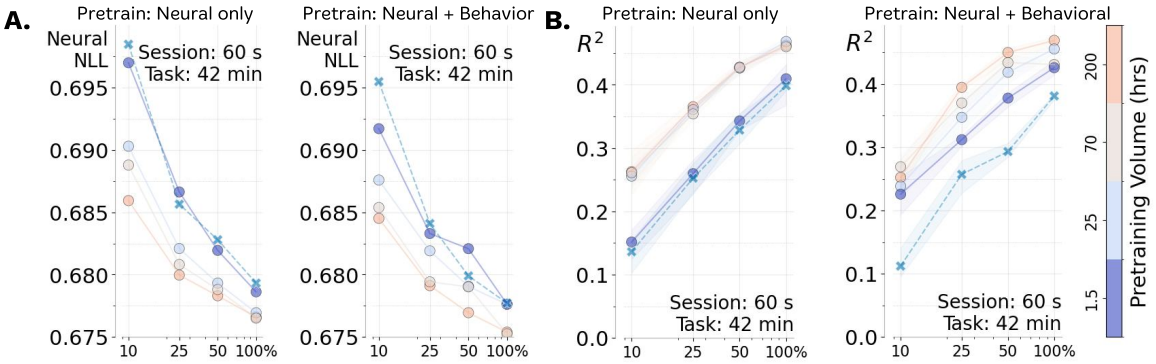

Figure C.4. Scaling of unsupervised pretraining After pretraining models up to 200 hours with only the neural reconstruction objective, we fine-tune models in a similar multiscale evaluation as in Section 5.3.1, with a newly initialized behavioral readout. We use the CST task here. **A.** left shows neural loss scales with neural-only pretraining. The right panel plots the neural loss, also present in the standard evaluation, also scales with joint pretraining. **B.** left shows that with neural pretraining, downstream decoding performance saturates at a flat performance after 25 hours. This is compared against the nonsaturated scaling from joint pretraining. Colorbar is common for all plots, and Xs are from-scratch NDT3 models.

use the standard single-stream autoregressive modeling objective as in the rest of this work in the downstream setting, we find direct linear probing of neural representations perform worse. Fig. C.4 compares the scaling on downstream neural and behavioral metrics after the standard fine-tuning procedure.

Fig. C.4A shows that downstream neural reconstruction improves with increased pretraining data either when using only the neural objective or both neural and behavioral objectives (as in the standard setting). The joint pretraining achieves advances neural metrics in all settings, illustrating that decoding behavior is a complementary objective to neural data reconstruction even for representation learning.

Fig. C.4B contrasts decoding curves in the two pretraining settings, in that neural pretraining has saturated decoding after just 25 hours of pretraining. This is consistent with the interpretation that the behavioral readout reflects only one aspect of the neural data. Together with the neural metric plots, this analysis shows scaling over solely neural data is possible, but also that decoding behavior is a complementary pretraining objective for improving neural representation learning and decoding.

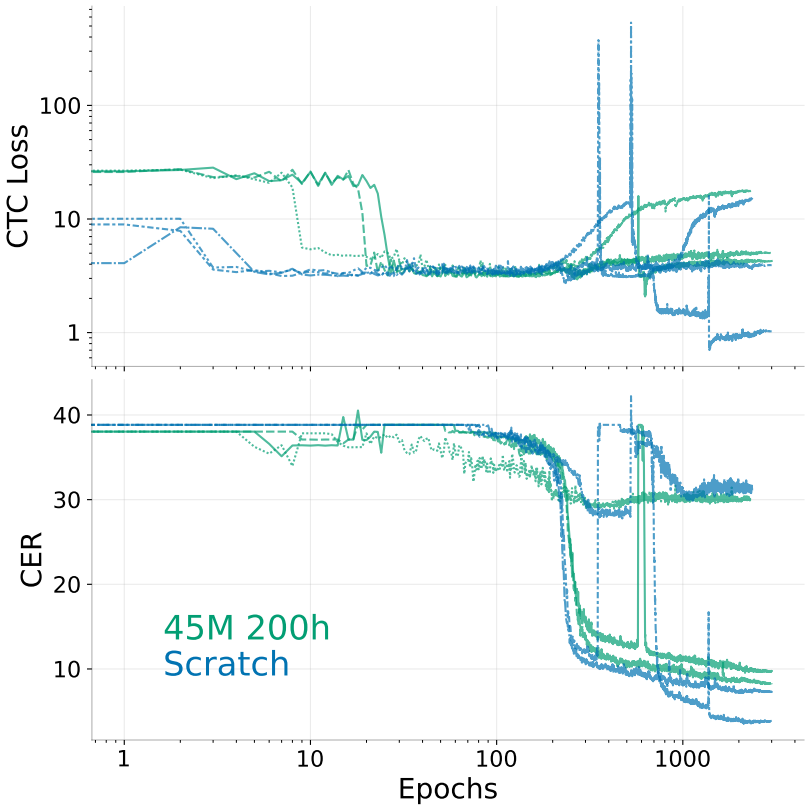

Figure C.5. Three regimes of NDT3 training for handwriting decoding. We show validation loss and character error rates for example runs of from-scratch and fine-tuned NDT3s.

Pretraining does not benefit FALCON H2 (Handwriting)

We also evaluated NDT3 for decoding of letters in a human-open loop handwriting task (FALCON H2). Although this is also a motor cortical decoding task, we excluded H2 from NDT3's aggregate evaluation since it is a sequence-to-sequence as opposed to continuous task. To apply NDT3 to this task, we pool neural tokens at each timestep and add a linear projection and optimize with a CTC loss [284]. We maintain the default neural reconstruction loss and causal attention mask, and do not apply data augmentation.

Note that RNNs are the current standard architecture for communication tasks like H2 [11, 135]. Training and tuning was less stable than for our continuous decoding tasks and required more extensive hyperparameter tuning, perhaps because the overall dataset size remains small (<1k samples), specific parameters are listed in the codebase. We observe three regimes in both training and fine-tuning. First, the model can fail to achieve an initial learning period. Second, the model can achieve reasonable nontrivial

solutions, comparable to expected performance for unaugmented RNNs (though we do not quantify this). Third, some models will exhibit learning instabilities that resolve in significantly improved performance. We illustrate these regimes in example validation curves below. Overall, the third regime is rarely achieved. More relevant to the main narrative of this work, fine-tuning appears to degrade both final solution quality and reduces the range of nontrivial hyperparameters (not shown). Investigating a sequence to sequence objective over CTC loss would be valuable future work.

Multiscale decoding on individual motor tasks

Fig. C.6A plots model performance for each of the 31 evaluation settings we study in the eight primary evaluation datasets we use. Studying any individual dataset will yield variable conclusions on whether pretraining structure is helpful, underscoring the need for proposed foundation models to be evaluated across many different datasets. We see the most clear scaling with pretraining data (color gradient with red on top) in the Critical Stability Task and Bimanual Task. FALCON tasks and Self-paced Reach appear minimally affected by scaling pretraining data, in that either pretrained models are generally slightly above a from scratch model at all data scales with no particular best pretrained model. The 2D + Click and Grasp datasets uniquely show high variability in model performance and strong degradation of the 350M 2 khr model at low data scales. Grasp instability was so high that we trained 9 seeds instead of the standard 3 to better estimate model performance. We propose this degradation is due to the instability of full fine-tuning of large models at the extremely low data scales these datasets present (e.g. 2.5 minutes at the 25% scaling). Finally, we remind that the 2D + Click, FALCON H1, and 1D Grasp Force tasks are datasets from human participants that are included in the 2 khr pretraining. Surprisingly, this leakage did not provide clear benefit to the 2 khr model.

These scaling plots also provide more precise context for baseline performance. NDT2 performs particularly poorly in the low data regime, while Wiener Filters perform poorly in the high data regimes.

In Fig. C.6B, we illustrate qualitative predictions on private datasets. These visualizations show a diversity in covariate timescales and structure. They also illustrate that the summary R^2 obscure several features of model predictions. For example, pretrained models in Cursor Y tend have false positive deflections in movement. R^2 also is not easily comparable in tasks with continuous dynamics (CST) vs. transient

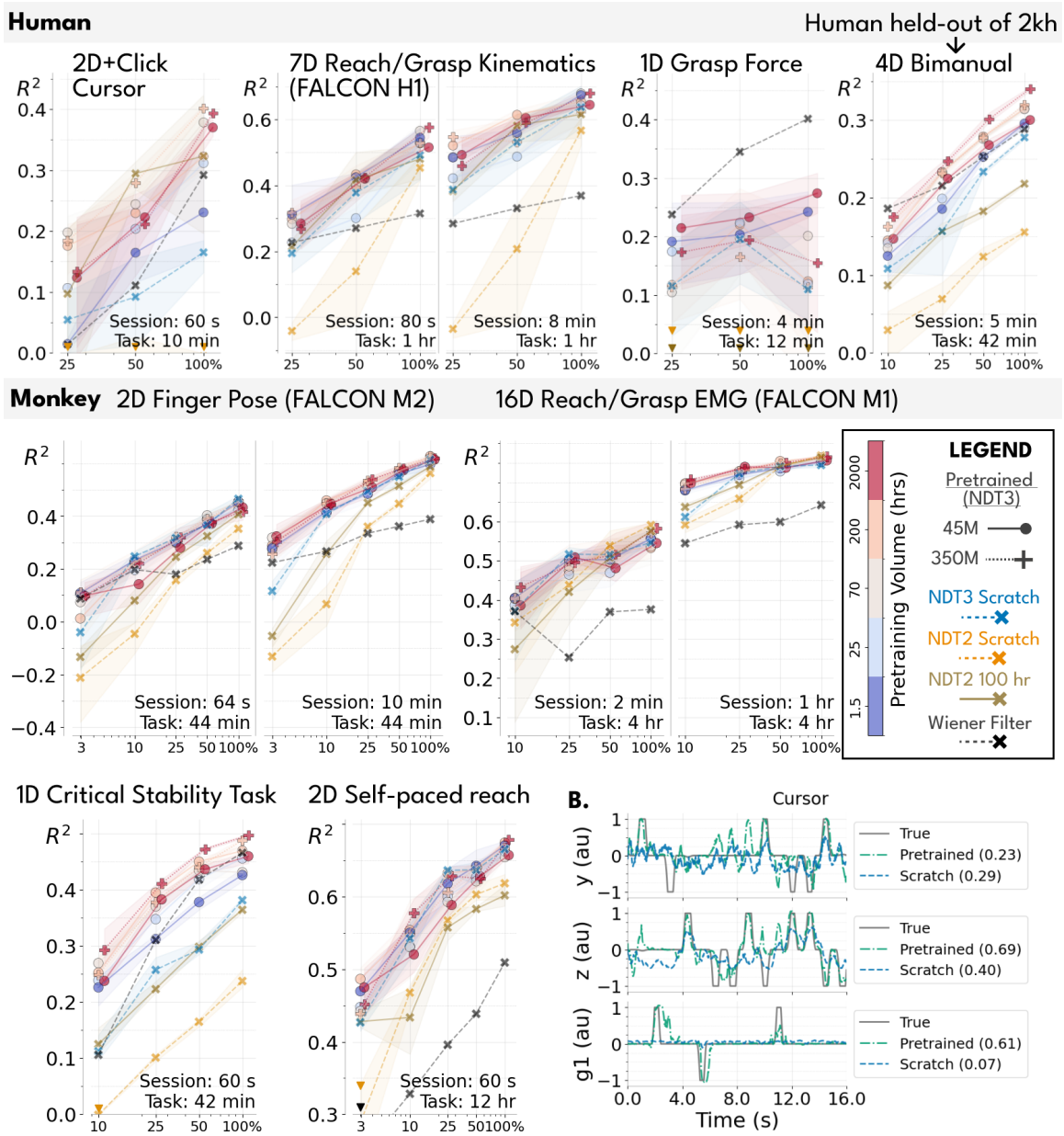

Figure C.6. A. Fine-tuning evaluations for individual datasets. Performance on both held-out (left) and held-in (right) splits are shown side by side by FALCON datasets. We shade the standard deviation of 3 model seeds in fine-tuning. Different tasks show substantial variability in benefit from pretraining. B. We show example predictions of a pretrained (45M 200h) and from-scratch NDT3 for the 2D + Click Cursor task to give a sense of what different prediction performances mean in terms of open loop data prediction. Numbers in legend are the R^2 for that model's predictions in the shown snippet.

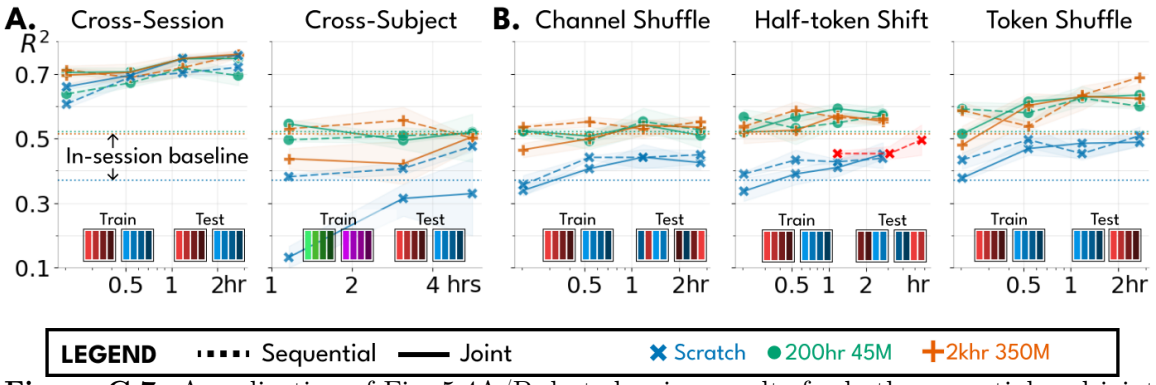

Figure C.7. A replication of Fig. 5.4A/B, but showing results for both sequential and joint fine-tuning in each setting, and on multiple fine-tuning data scales. As before, each model here tunes with some data from other settings (e.g. cross-subject data for the cross-subject panel), and a fixed amount of data on the test session. Sequential tuning first tunes on other-setting data, and then tunes on test session data. Joint tuning uses all data at once. In Fig. 5.4A/B, we only showed the better choice for each panel, i.e. we showed joint tuning for cross-session data, and sequential tuning for the other settings. In addition, we overlay how the cross-subject tuned models perform when applied to half-token shifted test data, confirming that subject and session data transfer similarly to shifted test data. Shading shows the standard deviation on 3 model seeds. In-session baseline refers to the no-intermediate tuning setting in Fig. 5.4B.

dynamics (Cursor G1).

Sequential tuning is similar to joint tuning

In Section 5.3.2, we showed that channel shuffling and half-token shifts were sufficient to reduce cross-session transfer to the extent of cross-subject transfer, in the context of a pretrained 350M 2khr NDT3 model and a from-scratch NDT3 model. Here we add 1) joint tuning results in addition to the sequential tuning in the main text, 2) extensions of this analysis to multiple intermediate tuning data scales, 3) an additional replication of this analysis for the 45M 200hr NDT3 model.

The comparison of sequential fine-tuning against with joint fine-tuning is done for completeness. Given cross-context data and a test session, we can either jointly tune on all data (as we do in Section 5.3.1), or sequentially tune first on the cross-context data and then on the test session. The latter is the protocol used in most other works and what we adopt in the main text for this analysis. We find for pretrained models that the results are similar, but that for from-scratch models, sequential tuning is essential for cross-subject transfer and slightly harmful for cross-session models

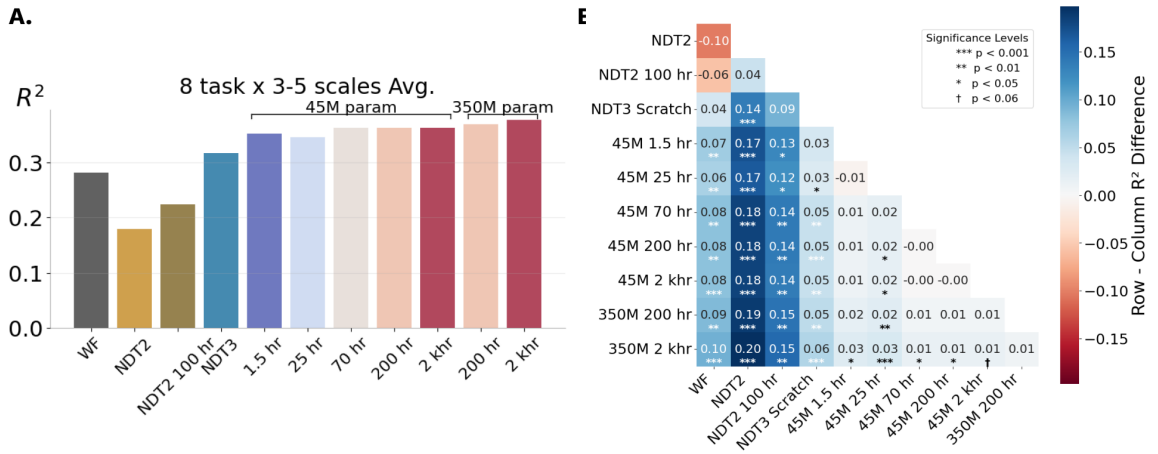

Figure C.8. A. A replication of Fig. 5.3D including an additional 25 hr and 70 hr model. The two additional models show the precise performance we measure may be noisily related to to pretraining data scale. B. Heatmap of differences between performances of pairs of models with significance tested with FDR-corrected pairwise t-tests. Note that coloring is used here to indicate differences, not significance. Positive numbers with significance indicate the row model outperforms the column model.

(Fig. C.7A). Seeing that sequential tuning is mainly advantageous for from-scratch models, we speculate that sequential tuning is particularly helpful for filtering learning signals in highly heterogeneous data, but pretrained models develop a capacity for doing such filtering during fine-tuning.

For the multiple intermediate data scale, only the largest scale was reported in the main text. Performance for most models decay towards the no intermediate tuning baseline. The exception here is cross-session tuning, where even minor amounts of data provide substantial gain, and cross-subject transfer in the joint tuning setting for from-scratch models, where, as mentioned, the model performs very poorly.

The 45M 200hr model performs differently than the 350M 2khr at individual settings but without any changes in overall trends.

Aggregate performance on all NDT3 models with significance tests

We additionally report the average performance of an NDT2 model pretrained with 100 hours of human data and two NDT3 models pretrained with 25 hours and 70 hours. These models are placed in context with the models from Fig. 5.3D, in Fig. C.8A. Note that for NDT3 models each successively larger data scale uses a strict superset of data from smaller scales. We also provide the p-values computed for the significance

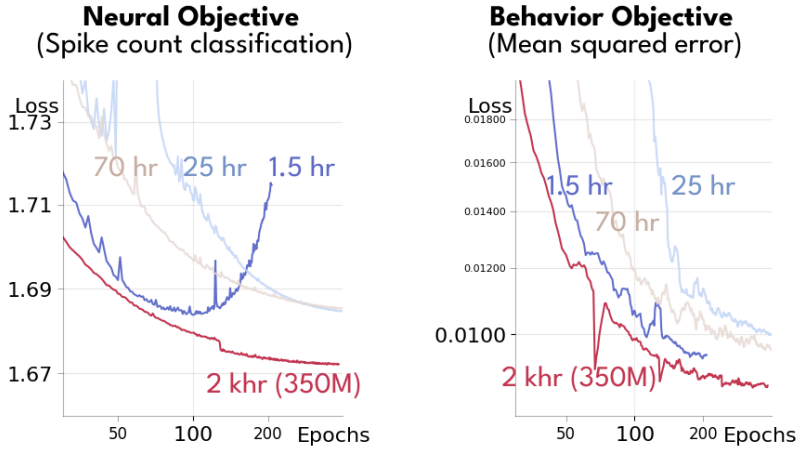

Figure C.9. Pretraining curves shown for 45M parameter NDT3s at 1.5 hour, 25 hour, and 70 hours, in addition to the curves for the 2 khr 350M parameter NDT3. We separately show metrics on neural and behavioral objectives through training. Since early stopping is used in model selection, we verify here that neither objective is overfits significantly, except for the 1.5 hour model's neural objective.

of the difference between each pair of models in Fig. C.8B. P-values are computed as FDR-corrected pairwise t-tests. The 350M 2 khr model has $p < 0.06$ improvements over all but the 350M 200 hr model. Interestingly, all other pretrained models, except the 25 hr model, appear equivalent, at least statistically. We presume this is due to the fact that our evaluation of 31 task settings may be insufficiently large.

NDT2 performs relatively poorly in our evaluation. This is true even when tuning from the public checkpoint trained on 100 hours of neural data from humans, though tuning does in general improve over the from-scratch NDT2 training. We believe pretrained NDT2's performance gap with NDT3 is partially due to NDT2's need to newly initialize decoding layers in each downstream task, which increases NDT2's dependence on thorough hyperparameter tuning. This makes NDT2 a poor candidate for a foundation model. Section C.1.3 and Section C.1.3 describe a number of methodological innovations that likely each contribute to the remaining performance differences between NDT2 and NDT3.

Neural vs Behavioral Objectives

In this work, the neural objective is present mainly as an auxiliary objective to improve downstream decoding. We see that neural and behavioral objectives are complementary in Section C.1.2.

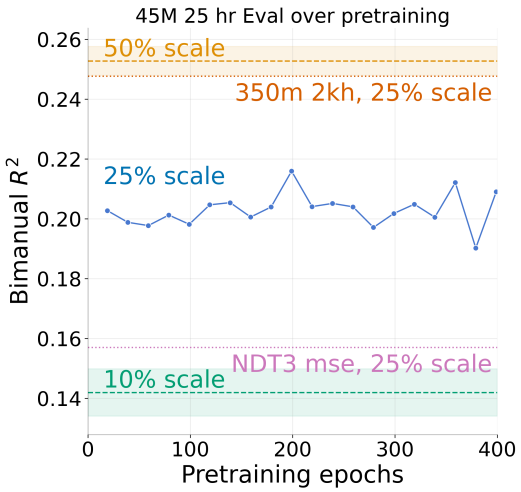

Figure C.10. Downstream evaluation conducted through pretraining of a 45M 25 hour NDT3, for the 25% scaling of the bimanual task, provided in context of the final evaluation achieved by the 10% and 50% bimanual task scale settings for the 45M 25 hour models, and the 25% scalings for the from-scratch NDT3 and best NDT3 (350M 2 khr). The benefits from pretraining are attained by epoch 20, and plateaus without overfitting for the rest of pretraining.

In Fig. C.9, we provide some additional context, showing that neural objectives also improve through pretraining, i.e. that we are not overfitting the neural objective and thus degrading decoding.

Downstream probe through pretraining

Upstream and downstream performance are generally assumed to be correlated in pretraining studies. This assumption supports the use of a single evaluation at the end of pretraining, rather than throughout pretraining, particularly if pretraining is not overfit. It is possible, however, that the narrower tasks used in evaluation may be learned or even overfit earlier than the general pretraining task. To assess this, we conduct a small downstream probe of checkpoints every 20 epochs of pretraining for a 45M 25 hour NDT3. This probe measures the performance specifically at the 25% scaling of the bimanual task. In Fig. C.10, we show this progression and compare the variability of checkpoints in pretraining with the differences across different task scales and pretraining models. The downstream probe for this task shows quick benefit from pretraining, providing the full gain over from-scratch models by the first checkpoint we evaluate at epoch 20, and then plateaus. This contrasts with the pretraining plots in Fig. C.9. The mismatch between upstream and downstream progress here

differs from correlated upstream-downstream progress in single-epoch language model studies [320], which may be due to differences in domain or pretraining scale.

Evaluation on the Neural Latents Benchmark

The Neural Latents Benchmark [127] is a benchmark for evaluating latent variable modeling of neural activity. This evaluation differs from NDT3's direct decoding evaluations in that all of NLB's metrics are derived from acausally inferred neural firing rates, akin to a pixel-level objective in computer vision. Note specifically that the decoding metric officially reported in the NLB is derived from submitted firing rates on the evaluation server; report decoding performance on NLB datasets without modeling neural activity is not an expected use of the benchmark.

Despite its distinct setting, the NLB provides two well-established datasets on which to evaluate models of motor cortical activity, providing context for NDT3's performance outside of decoding. To apply NDT3 to the NLB, we tuned NDT3 in a new downstream task where insert new a token at each timestep, from which we linearly decode firing rates of held out neurons. We performed this tuning separately for each of the maze and random target tasks (RTT), reporting the resulting co-bps scores in Table C.1. NDT3 performs poorly. From-scratch training on the single-session benchmark datasets underperform NDT1 in all tasks. Tuning from pretrained models improved performance, sample efficiency, and robustness to hyperparameters (only performance is reported here), but did not dramatically change model competitiveness. Zhang et al. [228] reported the value of more diverse neuron-level objectives for neural activity prediction, though their submission also dramatically underperforms NLB SoTA, warranting critical examination of whether pretraining benefits low-level modeling of neural activity.

Neural objectives are needed in fine-tuning

Our recommended fine-tuning for NDT3 is to match pretraining in most hyperparameters (except learning rate). This includes the use of the neural reconstruction objective, even though the downstream settings we used only evaluate behavior reconstruction. We show that including this neural objective is important in fine-tuning in Fig. C.11. Specifically, for pretrained models, ablating the neural objective can cause a substantial drop in performance. Interestingly, the drop is more significant for the larger scale

Dataset	RTT	Maze	Maze Large	Maze Medium	Maze Small
NDT1	0.1643	0.3597	0.3739	0.3081	0.2788
SoTA	0.2010	0.3650	0.3831	0.3329	0.3458
NDT3 Scratch	0.1533	0.3093	0.2892	0.1859	0.1853
NDT3 350M 2kh	0.1695	0.2775	0.2781	0.1937	0.2116

Table C.1. NDT1 vs NDT3 co-bps (higher is better) on NLB's MC Maze and MC RTT datasets in the 20ms split. The NDT3 results were produced were the standard sweeps used in this work, e.g. three random learning rates and three random seeds. SoTA results come from LangevinFlow for RTT and Maze and MINT for the Maze scaling datasets.

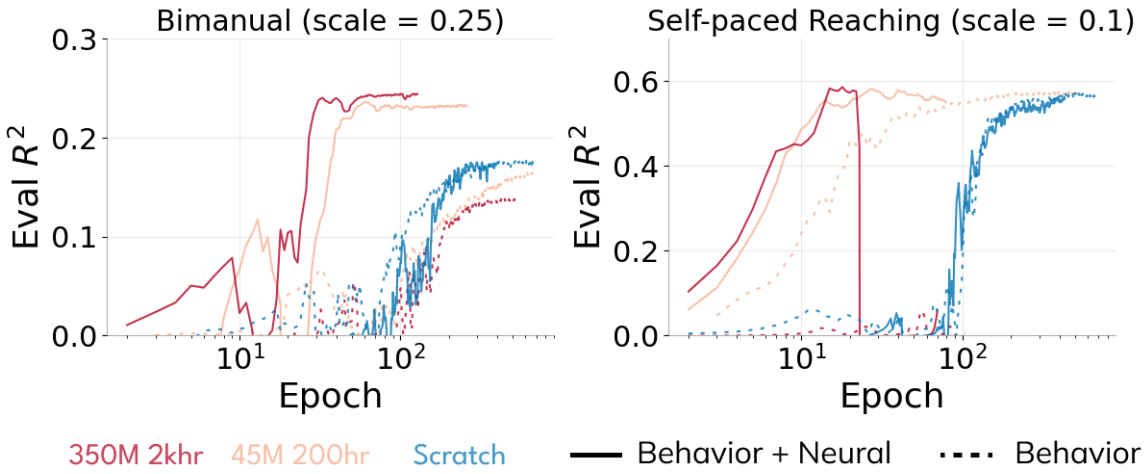

Figure C.11. We compare fine-tuning on bimanual and self-paced reach tasks (RTT), for variants with the neural objective and behavior objective to those with only the behavior objective. Curves from runs with the best validation R^2 of a standard 3 seed x 3 LR fine-tuning sweep are shown.

350M 2khr NDT3. This dependence of NDT3 on the neural objective is not desirable, but consistent with findings in other pretrained models that fine-tuning performs best when consistent with pretraining settings [321]. For from-scratch models, the neural objective does not cause significant impact.

C.1.3 Methods

Metrics and Evaluation

Throughout this work we evaluate offline decoding of continuous covariates timeseries. The metric we specifically use is the coefficient of determination, R^2, as computed by scikit-learn's *r2_score* function. R^2 is a useful metric over MSE as 1 represents

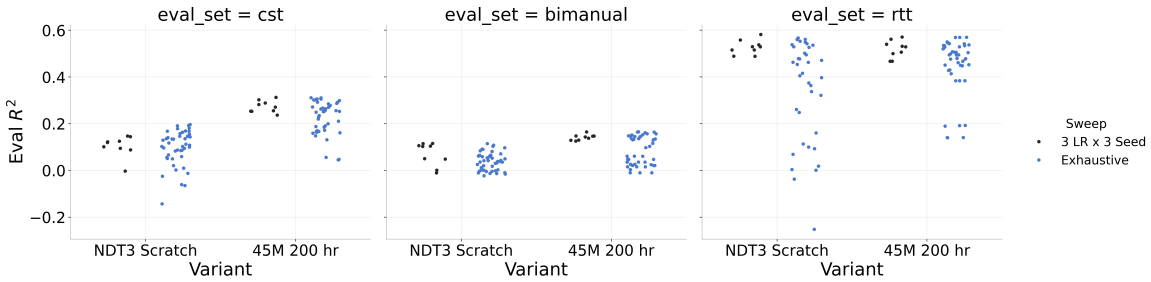

Figure C.12. For 3 monkeys datasets at 10% scale, we extend a HP sweep to 5 LRs and dropout in $[0.0, 0.1, 0.3]$ (vs default 0.1). For fine-tuning, we also sweep weight decay in $[0.0001, 0.01, 0.1]$ (vs default 0.1), while for from-scratch models we also sweep Transformer width ($[256, 512, 1024]$) vs default 512. This yields a 45-model sweep on 1 seed. We compare the range of scores achieved by this larger sweep against the standard 3 LR x 3 seed sweep.

perfect prediction and 0 is the score achieved by best-guess baseline, the mean of the data. In pretraining, R^2 is computed over the flat average of all covariate dimensions, since each datapoint has differing covariate dimensionalities. In evaluation, R^2 is computed as a variance-weighted average of R^2s in each covariate dimension. Another difference between training and evaluation metrics is that training predictions are made over batched data, while evaluation predictions are mostly computed in a *streaming* fashion. Streaming requires continuous neural data across different behavioral epochs, and so cannot be performed for the Oculomotor and CST datasets. We also omit it for the motor cortex self-paced reach dataset, which has a very large evaluation split. Streaming allows timesteps at the beginning of each sequence to leverage neural context from the preceding sequence, which raises performance slightly, as shown in the continuous vs trialized analysis (Section 5.3.3). We limit history in streaming evaluations to the max history seen in tuning (1 second).

Training

Pretraining hyperparameters were manually tuned in preliminary experiments at the 45M parameter models on small datasets. 350M models diverged at the chosen $4e-4$ peak LR, so we lowered peak LR to $1e-4$. For tuning, the explored LRs are $1e-4, 3e-4, 5e-4$ for training from scratch and $3e-5, 1e-4, 4e-4$ for fine-tuning. While this is far from an exhaustive search, we show in Fig. C.12 that other regularization hyperparameters are set to reasonable defaults such that this sweep finds near optimal results for both a from scratch model and fine-tuning the 45M model. Fine-tuning, like pretraining, is early stopped with a patience of 100

epochs. Batch size is uniformly set to 16K in pretraining, and scaled to be roughly 10-20% of dataset size in fine-tuning. NDT3 from-scratch models were trained at the 11M parameter range. Exact model configurations for different experiments are documented in the codebase.

NDT3's simple architectural design allows us to train on batches from different tasks and dimensionalities. To avoid excess padding in training, we concatenate pretraining data that is otherwise discontinuous (trialized) into 2 second data. We do not add any separator tokens, as this does not appear to have a performance impact for language models [221]. With mixed-precision training, the 350M parameter NDT3 can fit the 4-8K tokens in each input context in the memory of 40G NVIDIA A100 GPUs. Thus we can restrict NDT3's pretraining parallelism to data-parallelism.

Using Kaplan et al. [92]'s equation for FLOP computation, $C_{\text{forward}} = 2N + 2n_{\text{layer}}n_{\text{ctx}}d_{\text{attn}}$, we compute the footprint of the 350M 2kh model. We use about 0.9B FLOPs per token in the forward pass, and about 0.9T neural tokens processed over training, which yields a pretraining footprint of about 2.4e21 FLOPs.

Baselines and Other Transformers

Many models have been proposed for neural data decoding. While we believe the selected Wiener Filter and NDT2 baselines are representative of standard approaches, it may seem insufficient relative to the number of available approaches. To this end, we want emphasize that this work's primary aim is to evaluate the scaling of a base Transformer on neural data, rather than establish uniform superiority of a model over all others. Such claims can only be established in a field with many benchmarks and external evaluations, which BCI decoding currently lacks.

Wiener Filter The Wiener Filter baseline was cross-validated over regularization strength. We also swept history of neural input up to the max length provided to NDT, and reported the R^2 of the best WF according to test data in primary evaluation (slightly advantaging the WF). Generalization plots in Section 5.3.3 report the performance of WF models at these different histories. For evaluating angular generalization, WFs were only swept up to 1s history due to memory limits; performance was not varying substantially with history so we do not expect this to have impacted conclusions. The WF was for simplicity directly fit on the concatenated trial data, which may have slightly negatively impacted its performance in trialized datasets (Oculomotor, CST, Generalization analyses).

Dataset	Patience	Held-In R^2	Held-Out R^2
H1	100	$0.567_{\pm 0.034}$	$0.453_{\pm 0.030}$
H1 (reproduction)	250	$0.628_{\pm 0.011}$	$0.517_{\pm 0.016}$
H1 ([11])	250	0.62	0.52
M2	100	$0.563_{\pm 0.015}$	$0.352_{\pm 0.028}$
M2 (reproduction)	250	$0.582_{\pm 0.002}$	$0.391_{\pm 0.009}$
M2 ([11])	250	0.63	0.43

Table C.2. NDT2 H1 and M2 results when trained with 100 epochs of patience (this work) in fine-tuning vs 250 as in Karpowicz et al. [11]. We report mean and standard deviation of 3 model seeds on the FALCON evaluation (which is in turn a cross-session mean).

In the primary evaluations in Section 5.3.1, we considered WFs fit either independently per session in a dataset or jointly on all sessions, which is helpful for sessions in very low data regimes. We report the better of the 2. In generalization analyses, for simplicity, we only report joint fits, which may cause a slight downward bias in performance.

NDT2. NDT2 is a Transformer that has shown effective transfer learning across BCI datasets. It currently leads the FALCON BCI decoding benchmark and outperforms RNN decoders. NDT2 baselines were prepared with its public codebase. We trained NDT2 models both from-scratch and from the public checkpoint pretrained on 100 hours of human data. Max context length and patience were held constant across the models. This restriction to a patience of 100 epochs accounts for some difference with the reported FALCON benchmark results in Karpowicz et al. [11], as we note in Table C.2. The only departure we take from NDT2's default design is that we train NDT2 with both a neural reconstruction loss (25% masking) and a supervised decoding loss. This joint objective was introduced late in the NDT2 paper ([10] Fig 5) and performs comparably to multi-stage training while simplifying large scale evaluation. This is true for all eight evaluation tasks except CST. In the CST task, we used only the supervised decoding loss, as the token dropout used in reconstruction can dropout all neural input.

For hyperparameter tuning, we matched NDT3's tuning budget for the pretrained NDT2 checkpoint by only exploring 3 learning rates. Given mediocre NDT2 from-scratch performance, we swept NDT2 over 2 model sizes in addition to the standard 3 learning rates. We set the NDT2 from-scratch model sizes to 20M and 72M to be

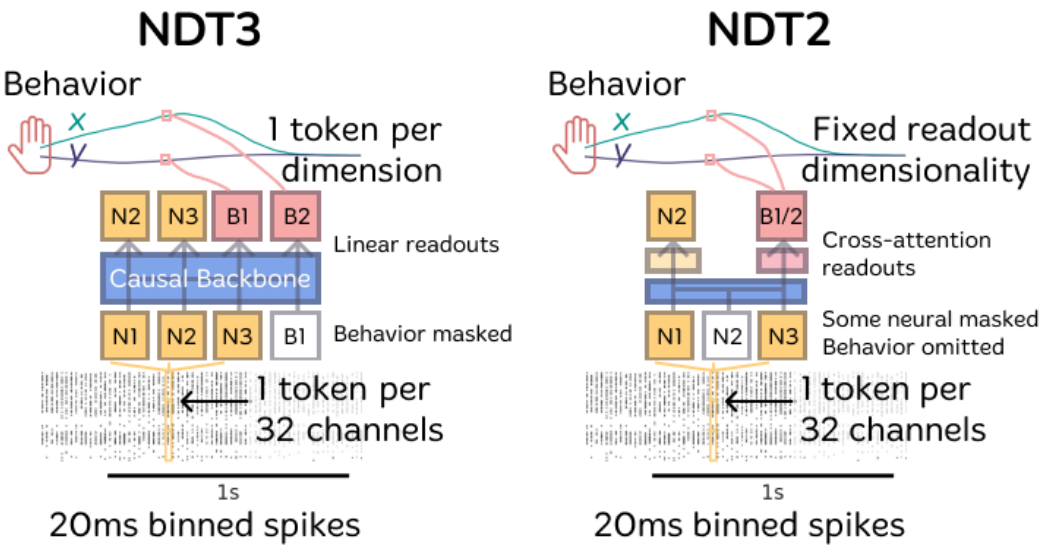

Figure C.13. NDT2 and NDT3 architectures differ mainly in the conversion of NDT2's representation learning backbone to NDT3's single multimodal stream. NDT3 directly intakes neural and behavioral tokens and predicts with a next-step objective. NDT2 employs explicit masking of input neural tokens and extracts neural and behavioral predictions at each timestep with cross-attention layers.

comparable with NDT3 45M, but note that the NDT2 pretrained checkpoint is only 6M parameters.

NDT2 vs NDT3. NDT3 builds off of NDT2 but departs in several manners to enable more streamlined scaling and analysis over heterogeneous decoding tasks, which we overview in Fig. C.13. Lower level technical changes are described in Section C.1.3. Both models relate neural data to behavior and train with both neural data and behavior prediction objectives. Both models use both these objectives in fine-tuning, but NDT2 only uses the neural objective in pretraining. NDT2's neural data objective is based on MAE [129], such that some fraction of input neural data tokens are masked and reconstructed in a decoder separate from the main backbone. However, this explicit masking was originally developed to study representation learning on images, not timeseries decoding. Causal domains like BCI control can also learn nontrivial representations simply through next-step prediction. NDT2 employed both explicit MAE masking and a causal attention mask in its backbone, which is redundant computationally and also reduces the context available to make predictions. NDT3 thus dispenses with the masking mechanism and uses next-step prediction alone. NDT2 also differs from NDT3 in its readout of behavior. Again, since NDT2 studied representation learning, it used additional cross-attention readout layers to "probe"

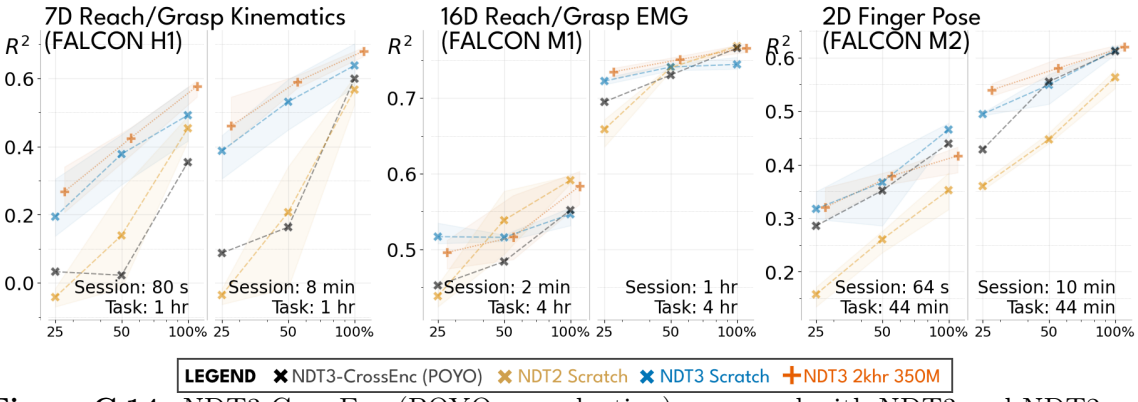

Figure C.14. NDT3-CrossEnc (POYO reproduction) compared with NDT3 and NDT2 on FALCON movement datasets. We find that the cross-attention does not provide benefits over patch-wise tokenization in NDT3. For NDT3-CrossEnc, we report the best model from a 9x larger hyperparameter sweep (over model size and dropout) than the other models.

behavior predictions at each timestep. NDT3 simplifies this two-part encoder-decoder design to a decoder-only design. In this flow, masked behavior tokens are provided at the input and filled in through the backbone, and varying the input tokens allows us to make predictions of the corresponding behavior dimensionality.

NDT-MTM vs NDT Zhang et al. [228] proposed a Multi-task Masking method (MtM) as a means of enabling improved scaling for spiking activity. MtM increases the sophistication of the masks applied in the MAE strategy used to train NDT1 and NDT2, and finds improvements on varied neural data metrics. We view this as analogous to introducing more granular objectives to improve "pixel" level metrics, and believe the results are consistent with the original image MAE [129] not achieving particularly good pixel metrics. However, decoding metrics, the focus of this work, are similar to those achieved by the baseline NDT models.

Of note, Jiang et al. [322] also shows that NDT-MTM fails to scale across subjects in the mouse neural prediction setting in which NDT-MTM was developed, replicating our core claim on the difficulty of cross-subject transfer. Given these similarities and that NDT-MTM makes no claim on advancing decoding, we did not attempt comparisons.

POYO vs NDT POYO [180] is a Transformer that can pretrain on spiking activity, proposed at a similar time as NDT2. POYO uses a cross-attention encoder, which is a qualitatively large departures from the more basic Transformer formula used in NDT. Specifically, rather than directly operating on patched neural data, as in NDT, POYO first "tokenizes" individual spikes on all channels and performs cross-attention to reduce

the large set of resulting tokens to a smaller number of latent tokens. In practice, this yields a pre-specified constant number of tokens per timestep, as opposed to NDT's choice of scaling the number of tokens depending on input data dimensionality.

The official POYO codebase was not supported as of writing this work, and neither were its pretrained weights available, so rather than attempt an exhaustive comparison, we sought a basic test of whether POYO's cross-attention encoder would enable improved performance and thus scaling in NDT3. A priori, we deem this unlikely as the Perceiver-based cross-attention encoder has not been widely adopted in other domains. Our re-implemented cross-attention encoder, holding all other elements constant within NDT3, performs worse than direct patching tokenization Fig. C.14. For clarity, we enumerate the differences between NDT3/NDT3-CrossEnc and POYO:

- NDT3 uses joint neural / behavior prediction losses. POYO is trained only with behavior prediction losses.
- NDT3 bins neural spike counts, while POYO encodes individual spike times. While the higher fidelity is likely useful, most BCI pipelines do not typically use individual spike times, and not all data releases include these more granular features.
- Dataset-specific embeddings: Both NDT2 and POYO use learned dataset embeddings to account for dataset-specific variability. NDT3 discards these learned tokens for simplicity, and accepts the small performance hit this introduces.
- Unit-specific embeddings: POYO additionally learns unit-specific embeddings for each neuron or channel, which varies across datasets. NDT's analogous spatial embedding, learned for each patch (or unit, when patch size is 1), is fixed across datasets.

While our reproduction may understate POYO's overall performance, it does suggest that the main design choice that could potentially improve scaling does not outperform simple patching. POYO does outperform NDT2 in M2, suggesting common choices with NDT3 such as rotary positional embeddings, might account for its reported benefits.

Pretraining and Evaluation datasets

Pretraining datasets were comprised of historical data from several labs, the rough composition of which is shown in Fig. 5.2B. The evaluation behavior used during

pretraining was reaching in 2 monkeys. The first monkey dataset came from a public release [323], and the second from a private dataset (Chase lab). The latter had center-out reach in standard conditions and under visual feedback perturbations. The monkey in the second dataset is also present in the 1khr monkey and 2kh and up model dataset sizes, though performing in a different set of experiments.

Inherent to the process of large-scale scraping is a loss of detail on what precise tasks were used, so we only have a qualitative description of tasks we believe are well represented. NDT3 trains on a wide variety of reaching behaviors from relatively constrained (2D center-out reaching to fixed number of targets) to relatively unconstrained (self-paced, more targets, potentially 3D) and under experimental manipulations (delayed onset, multiple targets, different error thresholds requiring more precision). These reaching behaviors are described in both endpoint kinematics and as EMG. A smaller fraction of pretraining data are isometric and force related (force exerted against manipulandums) for wrist and arm motion. Human datasets contain a variety of iBCI tasks, with closed loop datasets reflecting both high and low quality control. These tasks include reach and grasp behavior from 1-10 degrees of freedom, as well as some individuated finger tasks for clicking.

We detail evaluation datasets in Table C.3. Three datasets come from the FALCON benchmark [11], two are based on public datasets ([151, 136]), and three are private. Note we avoid the Neural Latents Benchmark [127] as it does not directly measure decoding performance. For each evaluation dataset, we specify a tuning split and an evaluation split. Only tuning split data is changed when varying data scale. Tuning and evaluation splits are block-contiguous, i.e. trials are not interleaved, for better downstream applicability.

Generalization analyses and Further evaluations

Intra-session generalization Posture, spring, and angular generalization evaluate OOD performance in the standard setup of comparing in-distribution and out-of-distribution performance directly (with changes in the underlying evaluation dataset) The intra-session temporal shift analysis is evaluated in an inverted, slightly more rigorous setting. Specifically, we trained two sets of models on the two different temporal blocks, and evaluated on an evaluation split in the later block, rather than only training on the early block and evaluating on both blocks. This way, the OOD shift is measured with respect to the same evaluation dataset.

Architectural Details

NDT3 adopts several architectural innovations used in recent Transformer models. These were compared against baselines in preliminary experiments, but formal ablations in the final experimental setting were not conducted. We defer full description of the Transformer dimensions to the public codebase.

- FlashAttention 2 [328] is used to increase training and inference speeds. On the NERSC Perlmutter cluster, with FA2, 45M NDT3 trained at about 270M neural tokens per 40G A100 hour, 350M NDT3 trained at about 70M neural tokens per A100 hour. Note, FA2 also enables use of the 350M model for real-time (<20ms) inference latency. On an NVIDIA 4090, we see mean inference times of 4ms for the 45M parameter NDT3 and 9ms for the 350M parameter NDT3.
- Positional Embeddings [227]: Rotary embeddings are applied to indicate the real-world timestep of every input token. Additionally, 48 categorical learned embeddings are reserved to distinguish token modality and position within a timestep (10 for neural, 16 for covariates, 16 for covariate constraints, 1 for reward/return, 1 for dummy tokens, remainder unused).
- QK Normalization [329, 330]: An additional layer norm is applied to the query and key embeddings, before the rotary embeddings, which helped stabilize training of the 350M parameter models.
- No context embeddings [10]: Differing from NDT2, no learned embeddings for disambiguating input datasets were prepended to each input. This was removed for simplicity. Per GATO [222] and language modeling practices, we instead leave task / dataset disambiguation to the modeling process: In pretraining, the covariate maskout strategy allows for many tasks to be specified in-context (as later behavior can be inferred on the basis of earlier neural-behavioral token relationships). In fine-tuning, the tuning dataset already uniquely specifies the function to be learned.
- Cross entropy loss for spiking data prediction: We used the standard cross entropy loss to classify spike count over the Poisson loss common in many neural data architectures. Since the overall ablation of neural objective shows no large impact in this work, it is likely that this decision should be evaluated with neural data related tasks rather than decoding.

We document the Transformer model shapes considered in our work in Table C.4. This shape is not systematically explored in our work, and is by historical artifact, slightly different than the shapes used in NLP/CV. Embedding parameters are negligible. One possible area of interest is that the feedforward expansion factor is 1 in our model, i.e. the MLP dimension is low. If MLPs do serve as memory stores in Transformers [331], increasing this shape may yield more performant model size scaling, given the heterogeneity of our datasets.

Table C.3. Evaluation datasets used for multiscale decoding and generalization analyses. The references provide extended description of the behavioral task. Dashed line separates datasets for Section 5.3.1 and for analysis. Datasets use unsorted multi-unit activity and are processed in 1s chops unless otherwise mentioned.

Dataset	Description
FALCON H1, M1, M2 [11]	3 separate single-subject multi-session datasets for different iBCI tasks. Data comes in a high data split (held-in), and a low-data split (held-out), with the intention on identifying methods that can achieve parity in the two settings. H1 is an open loop human dataset for calibrating 7D reach-and-grasp in a robot arm. M1 is a monkey reach-and-grasp task to different objects with EMG recordings. M2 is a monkey 2D finger movement task with manipulandum-measured kinematics. Scaling scores are reported on the test set.
Self-paced reach (RTT) [151]	Monkeys reach for random targets one at a time in a small planar workspace. We decode 2D arm velocity in monkey Indy. Has neural data from M1 and S1, we use M1 in Section 5.3.1 and Section 5.3.2 and S1 in Section 5.3.3.
Bimanual Cursor Control [136]	A human open loop dataset where the participant attempts movement of one or both hands to control two cursors.
2D Cursor + Click (private)	Cursor control is a classic iBCI endpoint [19, 324, 176]. Two human participants attempt movement according to visually cued cursor movement and audiovisual click cues. We also use this dataset for trial structure analysis in Section 5.3.3.
Grasp force (private)	A open-loop dataset with two human participants attempting isometric power grasps. Specifically, participants were asked to match force output according to visual cues in a Mujoco environment. Grasps cued were both static (instant onset, hold, and offset) or dynamic (gradually increasing force). This dataset is valuable for human iBCI study because force modulation is required in many motor behaviors, and grasp force has primarily only been characterized in monkeys until now [325]. Uses 2 second intervals due to long behavior timescale. We expect this dataset can be released by end of 2024.
Critical Stability Task [326] (private, trialized, sorted)	A monkey dataset collected to study continuous control relative to ballistic movement. The monkey balances a virtual cursor on a 1D workspace for up to 6 seconds.
Posture-varied Center-Out [56] (private, trialized, sorted)	A monkey center-out task, but the monkey's hand is adjusted to one of 6 different starting positions. We use the central position as center and the rest as edge.
Spring-load [54]	A monkey moves fingers, clamped together in a manipulandum for effective 1DoF, is neutral or under spring load.
Center-out, Monkey J [143] (trialized)	Used in Section 5.3.2. A monkey performs an isometric center out task. Forces are measured by the manipulandum and converted to cursor velocity signals.

Model	Layers	Width	MLP Size	Heads	Parameters (M)
NDT2 PT [10]	4	256	256	4	6
NDT3 Base	6	1024	1024	8	45
NDT3 Big	12	2048	2048	16	350

Table C.4. Transformer Model Shapes used in this work.

Appendix D

Supplement to Chapter 6 (DELETE)

D.1 Supplementary Material

D.1.1 DELETE suppresses highly entrained neural responses

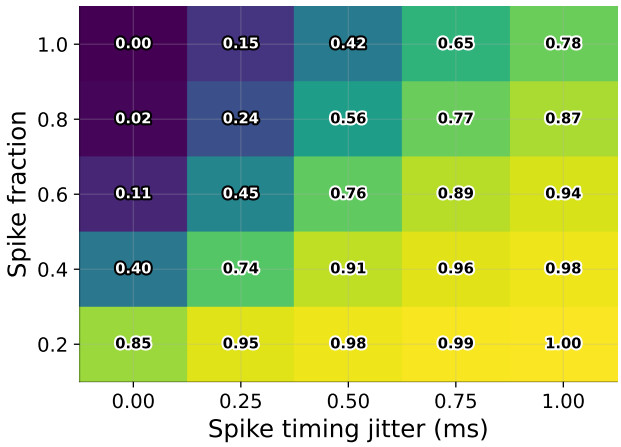

Figure D.1. A spike is identically injected after a fraction of stimulation pulses and with latency jitter to create a simulated ICMS dataset. Heatmap values show the recovery rate of these injected spikes.

Because DELETE removes predictable structure, highly synchronized activity will be attenuated. We characterize this failure mode in a controlled simulation. Following O'Shea and Shenoy [251], we sweep spike fraction (probability of an evoked spike per pulse-channel pair; 0.2–1.0) and timing jitter (full width of a uniform latency window; 0–1.0 ms) across the ICMS simulation from Fig. 6.5, training a separate DELETE model per condition. Recovery rates across this sweep are shown in Fig. D.1.

Spike recovery degrades as spikes become more reliable and more tightly locked: at fraction 1.0 and jitter 0 ms—the worst case, where every pulse produces an identically timed spike—hit rate drops to near zero. Recovery improves with increasing jitter and decreasing reliability; with jitter of at least 0.5 ms and spike fraction at most 0.6, hit rates exceed 0.75. For comparison, Fig. 10C of O'Shea and Shenoy [251] shows qualitatively similar ERAASR degradation in the same parameter regime.

D.1.2 Synthetic spike injection and normalized recall

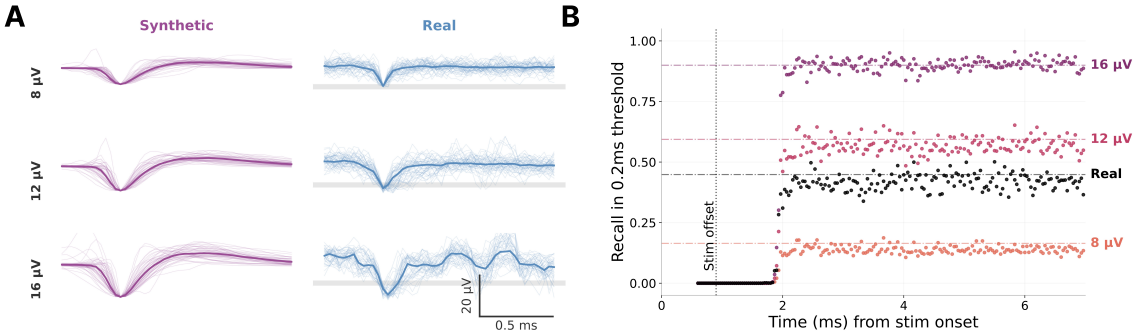

Figure D.2. Spike recovery is near complete at larger-amplitude injections. (A) Injected synthetic waveforms (Synthetic, left) compared with the recorded spikes used in the Fig. 6.8 benchmark (Real, right), grouped by peak amplitude. Thin traces are individual events, bold traces their mean. (B) Recovery rate—recall within a 0.2 ms matching tolerance—against time from stimulation onset on a Single channel Fixed dataset, for injections at 8, 12, and 16 μV alongside the recorded spikes themselves (Real). Dash-dot lines mark the corresponding recall measured in non-stimulation periods, used to normalize recall in the main text.

The Injection Recall metric (Section 6.2.4) asks what fraction of a known spike population survives denoising during stimulation. Its difficulty is that a spike's recoverability depends on its amplitude regardless of any artifact: a small spike injected into quiet background may not cross threshold at all. Raw recall therefore confounds detectability with artifact contamination, and we normalize by recall measured on the same injections in non-stimulation periods to isolate the stimulation artifact's contribution.

Fig. D.2 provides the sanity check that large-amplitude spikes, which are rare in our datasets, are recovered at near-complete rates. Fig. D.2A shows curated synthetic waveforms from Steinmetz et al. [233] against waveforms of comparable amplitude from our datasets. Synthetic waveforms are smooth by construction, while recorded

spikes carry the surrounding background, so the two are comparable in scale but not identical in shape.

Fig. D.2B shows that recall quickly recovers to ceiling rates after the stimulation transient. More importantly, actual and ceiling rates increase with amplitude, nearing 0.9 by the largest 16 μV amplitude we test.

D.1.3 Sensitivity of Activity Synchrony to the co-activation threshold

Activity Synchrony (Section 6.2.4) requires choosing what fraction of the array must be simultaneously active for a sample to count as a synchronous event. We report 25%, which is conservative enough that flagged events are unlikely to be physiological, but the choice is arbitrary. To test whether our conclusions depend on it, we sweep the threshold over 12.5–75% in Fig. D.3.

Absolute event rates fall steeply as the threshold rises, as expected: a stricter criterion requires more channels to cross together, and by 75% almost every method registers close to zero on every dataset. The overall comparison is stable. DELETE records the lowest or near-lowest synchronous event rate at every threshold and on every dataset, and the methods that leave the most residual artifact by RMS Inflation (Fig. 6.8A) are the same ones that flag the most synchronous events here.

The 12.5% threshold separates the methods most widely, but to minimize risk of reporting genuine neural synchrony, we use the more conservative 25% criterion. The higher threshold remains discriminative.

D.1.4 Interpreting waveform R^2 discrepancies

The Waveform R^2 metric compares peri-stimulation crossings with pre-stimulation waveforms from the same channel. The three selected datasets illustrate different reasons for disagreement between estimators. In the star dataset, DELETE scores below several baselines for successfully suppressing strongly stereotyped line noise-type waveforms. The square dataset replicates this metric failure mode, with only one contaminated unit visible. The triangle dataset shows a third such case, but further shows DELETE channels also contaminated with light oscillation. In general, we cannot establish that DELETE has fully suppressed the artifact in these datasets, but

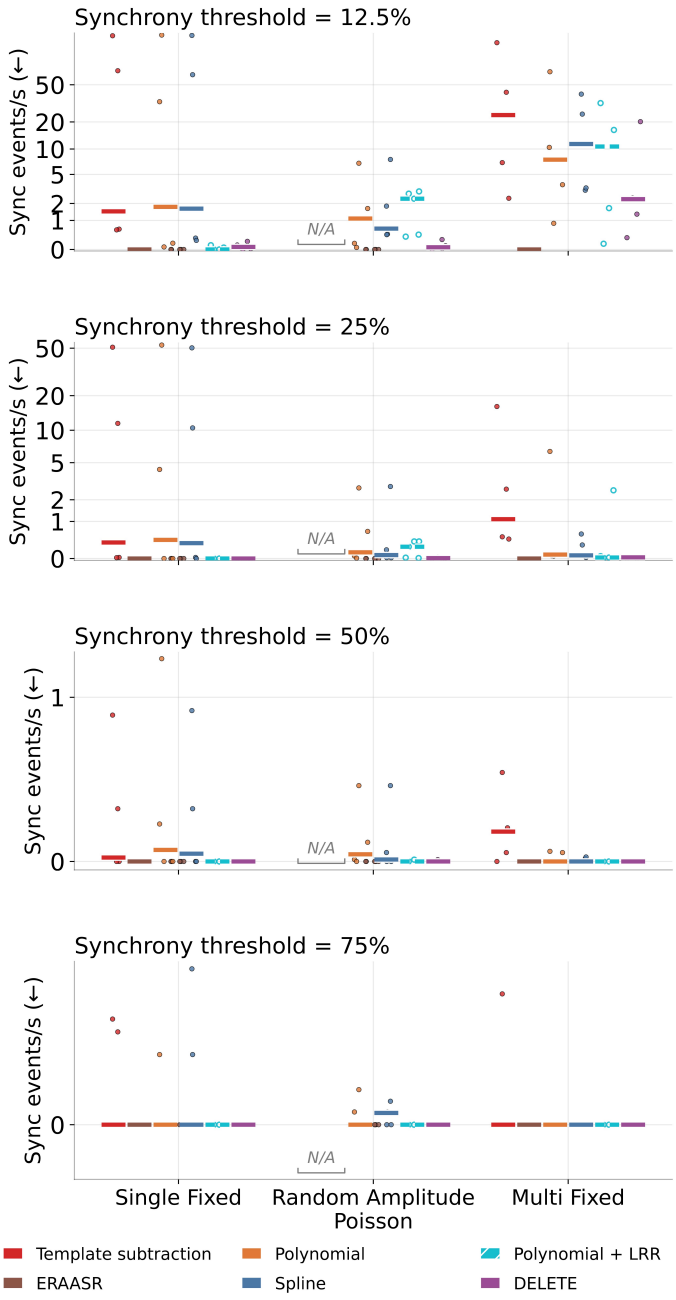

Figure D.3. Method ranking by Activity Synchrony is stable across co-activation thresholds. Synchronous events per second (lower is better) for each method and dataset, sweeping the fraction of the array that must cross threshold simultaneously for a sample to count as a synchronous event: 12.5, 25, 50, and 75%. The 25% panel is the setting reported in Fig. 6.8C. Bars mark the median across datasets and points individual datasets. Note the differing y-axis scales across panels. DELETE consistently ranks lowest or near-lowest across all thresholds and datasets.

can judge that the waveform metric can fail to reliably distinguish methods by their ability to recover physiological units.

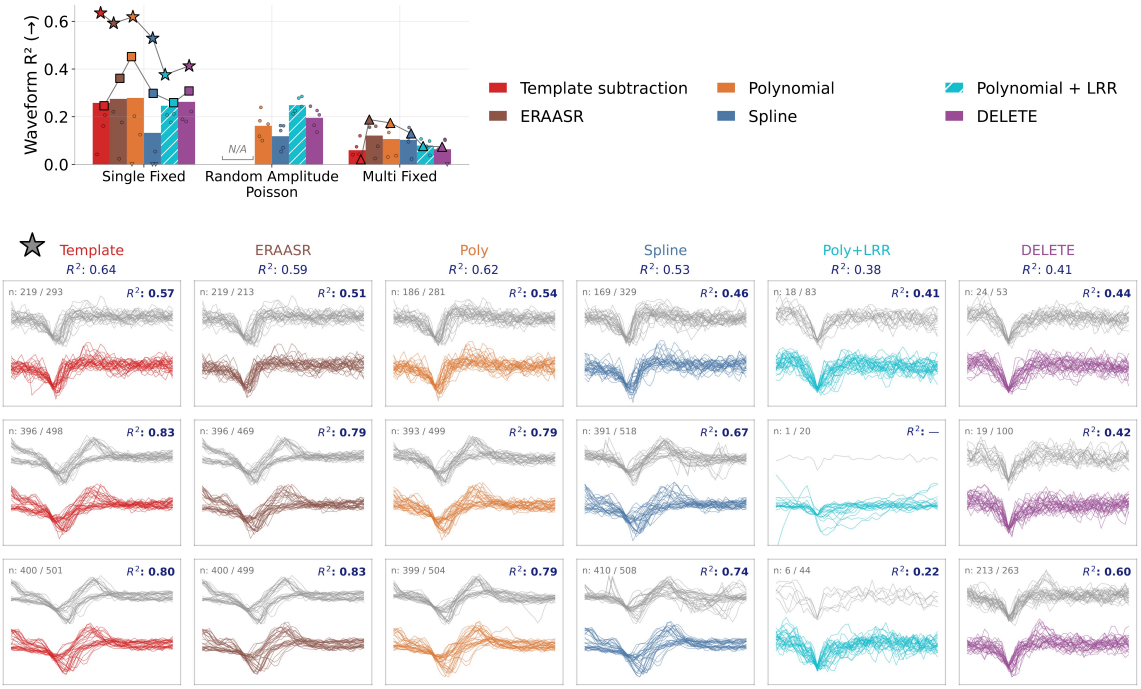

Figure D.4. Waveform R^2 dataset breakout. (Top) We reproduce the waveform R^2 benchmark evaluation from Fig. 6.8B, but mark three datasets where DELETE scores below other estimators, marked by a star, square, and triangle. (Bottom) On successive pages, we show three sampled channels from each selected dataset under all six estimators. Channel R^2 and quiet/peri-stimulation snippet counts are shown. A dash marks channels with fewer than two quiet or two peri-stimulation snippets; unscored channels contribute zero to the dataset average. Waveforms with flat segments reflect crossings near blanking intervals.

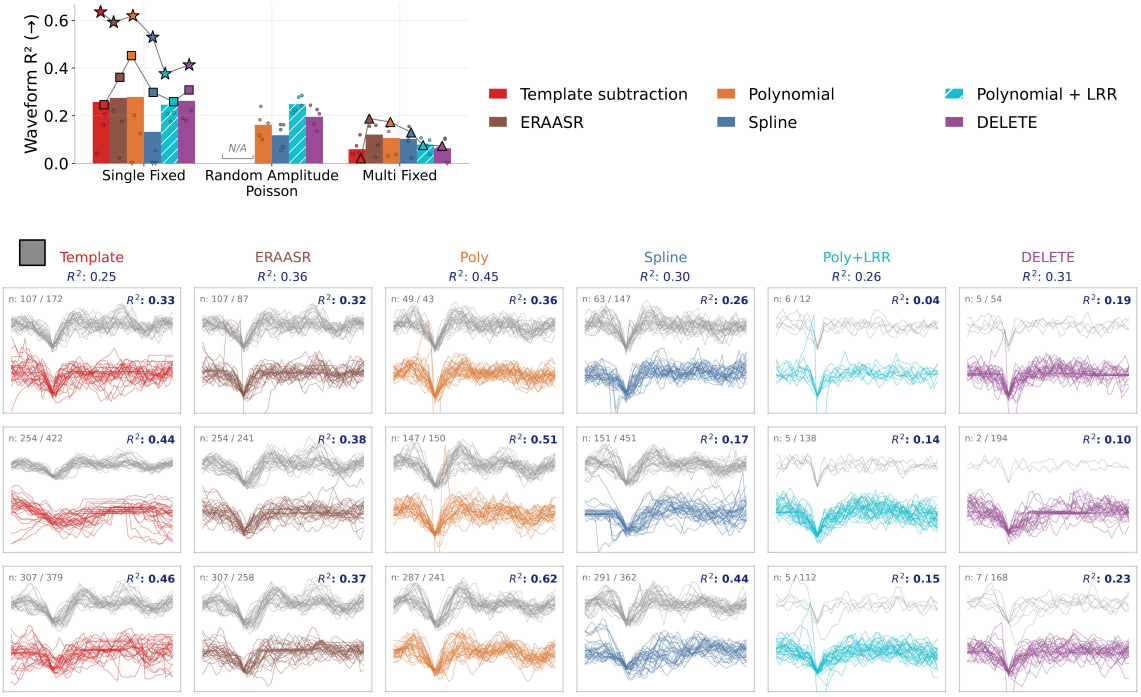

Figure D.4. Waveform R^2 dataset breakout (continued).

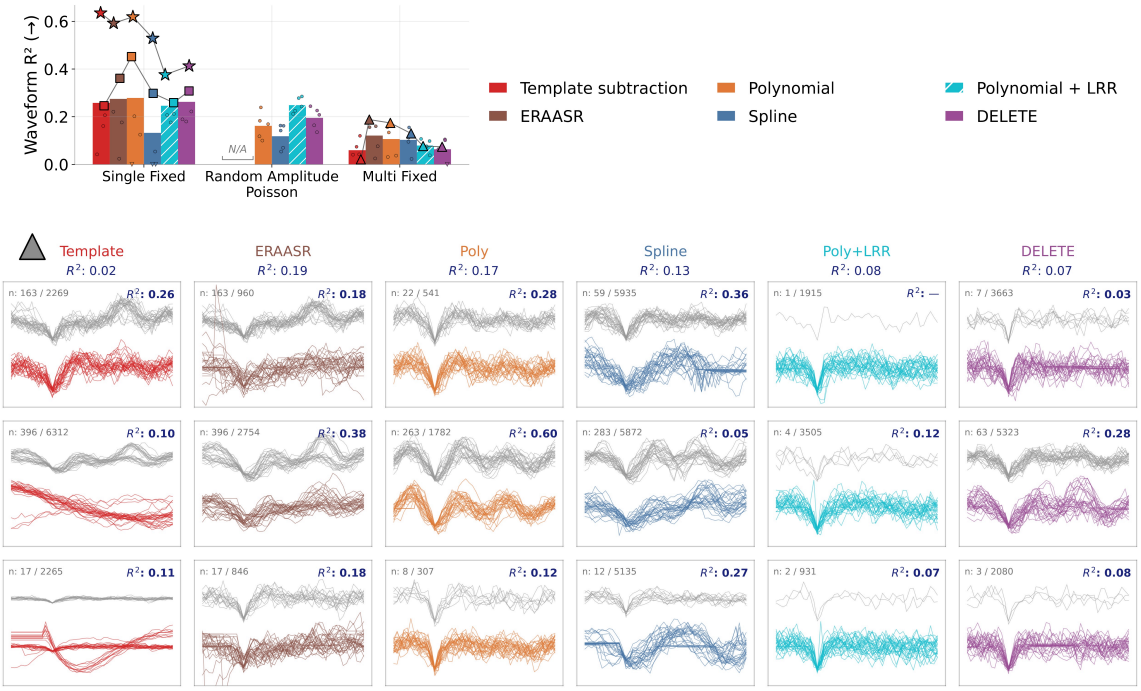

Figure D.4. Waveform R^2 dataset breakout (continued).

D.1.5 Neurometric threshold construction and per-method threshold correlations

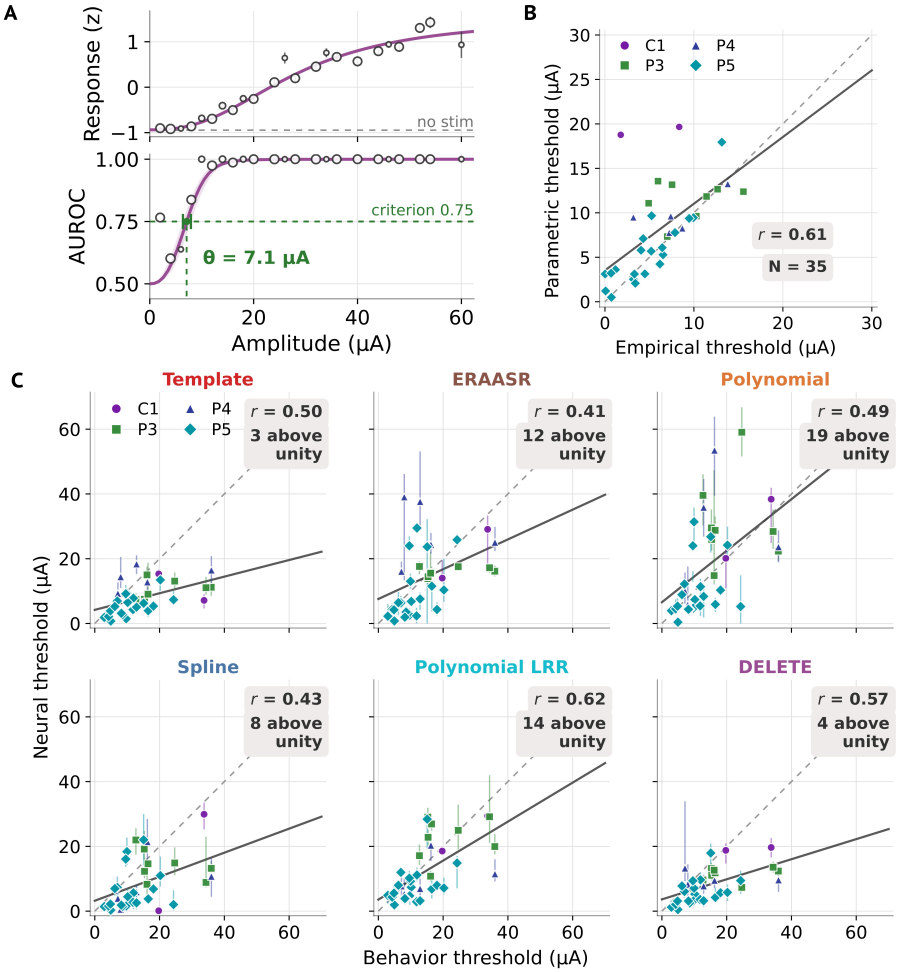

Figure D.5. ICMS neurometric methodology. (A) Construction of the neurometric threshold for one stimulating electrode (participant P5, DELETE), from 377 stimulation trials across 23 delivered amplitudes (2–60 μA) against 47 no-stimulation trials. Top: mean single-trial evidence at each delivered amplitude ($\pm$1 SE, marker area proportional to trial count) with the fitted log-sigmoid dose-response curve. The dashed rule marks the no-stimulation level, from which the response departs at about 4 μA. Bottom: empirical AUROC at each delivered amplitude (open markers) and the analytic AUROC of the same fit (log-sigmoid curve), shaded with its bootstrap interval. The fit is pinned to chance at zero current by construction. The threshold is the amplitude at which the curve reaches the 0.75 criterion, here 7.1 μA, with whiskers giving the bootstrap 95% interval on the amplitude axis. (B) Agreement between the parametric neurometric threshold used throughout and an empirical threshold fit directly to per-amplitude AUROCs. (C) Neurometric versus behavioral thresholds for each denoising method, one point per electrode-day ($n = 35$), with vertical bars giving each neurometric threshold's bootstrap interval. The dashed line is unity and the solid line the fit. These are the scatters summarized in Fig. 6.9G.

Bibliography

[1] Jorge Luis Borges. On Exactitude in Science. In *Collected Fictions*, page 325. Penguin Books, New York, 1998. Originally published as "Del rigor en la ciencia," in *Historia universal de la infamia*, 1946.

[2] Sven Dorkenwald, Arie Matsliah, Amy R Sterling, Philipp Schlegel, et al. Neuronal wiring diagram of an adult brain. *Nature*, 634:124–138, 2024. doi: 10.1038/s41586-024-07558-y.

[3] Larry F Abbott, Davi D Bock, Edward M Callaway, Winfried Denk, et al. The mind of a mouse. *Cell*, 182(6):1372–1376, 2020. doi: 10.1016/j.cell.2020.08.010.

[4] Pablo A Valdes, Chih-Chieh Jay Yu, Jenna Aronson, Debarati Ghosh, Yongxin Zhao, Bobae An, et al. Improved immunostaining of nanostructures and cells in human brain specimens through expansion-mediated protein decrowding. *Science Translational Medicine*, 16(732):eabo0049, 2024. doi: 10.1126/scitranslmed.abo0049.

[5] Ian H Stevenson and Konrad P Kording. How advances in neural recording affect data analysis. *Nature Neuroscience*, 14(2):139–142, 2011. doi: 10.1038/nn.2731.

[6] Mackenzie Weygandt Mathis and Alexander Mathis. Deep learning tools for the measurement of animal behavior in neuroscience. *Current Opinion in Neurobiology*, 60:1–11, 2020. doi: 10.1016/j.conb.2019.10.008.

[7] Eric Jonas and Konrad Paul Kording. Could a neuroscientist understand a microprocessor? *PLoS Computational Biology*, 13(1):e1005268, 2017. doi: 10.1371/journal.pcbi.1005268.

[8] John W Krakauer, Asif A Ghazanfar, Alex Gomez-Marin, Malcolm A MacIver, and David Poeppel. Neuroscience needs behavior: Correcting a reductionist bias. *Neuron*, 93(3):480–490, 2017. doi: 10.1016/j.neuron.2016.12.041.

[9] Yves Frégnac. How blue is the sky? *eNeuro*, 8(2), 2021. doi: 10.1523/ENEURO.0130-21.2021.

[10] Joel Ye, Jennifer L Collinger, Leila Wehbe, and Robert Gaunt. Neural data transformer 2: Multi-context pretraining for neural spiking activity. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=CBBtMnlTGq`.

[11] Brianna M Karpowicz, Joel Ye, Chaofei Fan, Pablo Tostado-Marcos, Fabio Rizzoglio, Clay Washington, Thiago Scodeler, Diogo de Lucena, Samuel R Nason-Tomaszewski, Matthew J Mender, Xuan Ma, Ezequiel Matias Arneodo, Leigh R Hochberg, Cynthia A Chestek, Jaimie M Henderson, Timothy Q Gentner, Vikash Gilja, Lee E Miller, Adam G Rouse, Robert A Gaunt, Jennifer L Collinger, and Chethan Pandarinath. Few-shot algorithms for consistent neural decoding (FALCON) benchmark. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://openreview.net/forum?id=FN02v4nD8y. Datasets and Benchmarks Track.

[12] Joel Ye, Fabio Rizzoglio, Xuan Ma, Adam Smoulder, Hongwei Mao, Gary Blumenthal, William Hockeimer, Nicolas Kunigk, Dalton Moore, Patrick Marino, Raeed Chowdhury, J. Patrick Mayo, Aaron Batista, Steven Chase, Michael Boninger, Charles Greenspon, Andrew B Schwartz, Nicholas Hatsopoulos, Lee E Miller, Kristofer Bouchard, Jennifer Collinger, Leila Wehbe, and Robert A Gaunt. A generalist intracortical motor decoder. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://openreview.net/forum?id=ONOe6cAE9I.

[13] Joel Ye, Taylor Hobbs, Gary Blumenthal, Leila Wehbe, and Robert Gaunt. DELETE: Denoising electrical events with a transformer encoder. In preparation, 2026.

[14] David M. Brandman, Sydney S. Cash, and Leigh R. Hochberg. Review: Human intracortical recording and neural decoding for brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(10):1687–1696, 2017. doi: 10.1109/TNSRE.2017.2677443. Epub ahead of print March 2, 2017.

[15] Chethan Pandarinath and Sliman J. Bensmaia. The science and engineering behind sensitized brain-controlled bionic hands. *Physiological Reviews*, 102 (2):551–604, 2022. doi: 10.1152/physrev.00034.2020. URL https://journals.physiology.org/doi/full/10.1152/physrev.00034.2020.

[16] Jennifer L. Collinger, Brian Wodlinger, John E. Downey, Wei Wang, Elizabeth C. Tyler-Kabara, Douglas J. Weber, Angus JC McMorland, Meel Velliste, Michael L. Boninger, and Andrew B. Schwartz. High-performance neuroprosthetic control by an individual with tetraplegia. *The Lancet*, 381 (9866):557–564, February 2013. ISSN 0140-6736, 1474-547X. doi: 10.1016/S0140-6736(12)61816-9. URL https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(12)61816-9/fulltext. Publisher: Elsevier.

[17] Francis R. Willett, Erin M. Kunz, Chaofei Fan, Donald T. Avansino, Guy H. Wilson, Eun Young Choi, Foram Kamdar, Matthew F. Glasser, Leigh R. Hochberg, Shaul Druckmann, Krishna V. Shenoy, and Jaimie M. Henderson. A high-performance speech neuroprosthesis. *Nature*, 620(7976):1031–

1036, August 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06377-x. URL https://www.nature.com/articles/s41586-023-06377-x. Publisher: Nature Publishing Group.

[18] Maitreyee Wairagkar, Nicholas S. Card, Tyler Singer,Clark, Xianda Hou, Carrina Iacobacci, Lee M. Miller, Leigh R. Hochberg, David M. Brandman, and Sergey D. Stavisky. An instantaneous voice-synthesis neuroprosthesis. *Nature*, 644:145–152, 2025. doi: 10.1038/s41586-025-09127-3.

[19] Chethan Pandarinath, Paul Nuyujukian, Christine H Blabe, Brittany L Sorice, Jad Saab, Francis R Willett, Leigh R Hochberg, Krishna V Shenoy, and Jaimie M Henderson. High performance communication by people with paralysis using an intracortical brain-computer interface. *eLife*, 6:e18554, February 2017. ISSN 2050-084X. doi: 10.7554/eLife.18554. URL https://doi.org/10.7554/eLife.18554. Publisher: eLife Sciences Publications, Ltd.

[20] Sharlene N. Flesher, Jennifer L. Collinger, Stephen T. Foldes, et al. Intracortical microstimulation of human somatosensory cortex. *Science Translational Medicine*, 8(361):361ra141, Oct 2016. doi: 10.1126/scitranslmed.aaf8083.

[21] Emily Graczyk, Brianna Hutchison, Giacomo Valle, David Bjanes, Deanna Gates, Stanisa Raspopovic, and Robert Gaunt. Clinical applications and future translation of somatosensory neuroprostheses. *Journal of Neuroscience*, 44(40), 2024.

[22] Sharlene N. Flesher, John E. Downey, Jeffrey M. Weiss, Christopher L. Hughes, Angelica J. Herrera, Elizabeth C. Tyler-Kabara, Michael L. Boninger, Jennifer L. Collinger, and Robert A. Gaunt. A brain-computer interface that evokes tactile sensations improves robotic arm control. *Science*, 372(6544):831, 2021. doi: 10.1126/science.abd0380. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8715714/.

[23] Michelle Armenta Salas, Luke Bashford, Spencer Kellis, Matiar Jafari, HyeongChan Jo, Daniel Kramer, Kathleen Shanfield, Kelsie Pejsa, Brian Lee, Charles Y. Liu, and Richard A. Andersen. Proprioceptive and cutaneous sensations in humans elicited by intracortical microstimulation. *eLife*, 7:e32904, 2018. doi: 10.7554/eLife.32904. Published April 10, 2018.

[24] Matthew S Willsey, Nishal P Shah, Donald T Avansino, Nick V Hahn, Ryan M Jamiolkowski, Foram B Kamdar, Leigh R Hochberg, Francis R Willett, and Jaimie M Henderson. A real-time, high-performance brain-computer interface for finger decoding and quadcopter control. *bioRxiv*, pages 2024–02, 2024. URL https://www.biorxiv.org/content/10.1101/2024.02.06.578107v1.

[25] Thomas Hosman, Tsam Kiu Pun, Anastasia Kapitonava, John D Simeral, and Leigh R Hochberg. Months-long high-performance fixed lstm decoder for cursor control in human intracortical brain-computer interfaces. In *2023 11th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages

1–5. IEEE, 2023.

[26] Chaofei Fan, Nick Hahn, Foram Kamdar, Donald Avansino, Guy H Wilson, Leigh Hochberg, Krishna V. Shenoy, Jaimie M. Henderson, and Francis R Willett. Plug-and-play stability for intracortical brain-computer interfaces: A one-year demonstration of seamless brain-to-text communication. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=STqaMqhtDi.

[27] Nicholas S Card, Tyler Singer-Clark, Hamza Peracha, Carrina Iacobacci, Xianda Hou, Maitreyee Wairagkar, Zachery Fogg, Elena C Offenberg, Leigh R Hochberg, Sergey D Stavisky, and David M Brandman. Long-term independent use of an intracortical brain-computer interface for speech and cursor control. *Nature Medicine*, 32(7):2504–2510, 2026. doi: 10.1038/s41591-026-04414-6.

[28] Marc W Slutzky. Brain-machine interfaces: Powerful tools for clinical treatment and neuroscientific investigations. *The Neuroscientist*, 25(2):139–154, 2019. doi: 10.1177/1073858418775355.

[29] Nikhilesh Natraj, Sarah Seko, Reza Abiri, Runfeng Miao, Hongyi Yan, Yasmin Graham, Adelyn Tu-Chan, Edward F. Chang, and Karunesh Ganguly. Sampling representational plasticity of simple imagined movements across days enables long-term neuroprosthetic control. *Cell*, 188(5):1208–1225.e32, 2025. doi: 10.1016/j.cell.2025.02.001.

[30] Daniel B. Silversmith, Reza Abiri, Nicholas F. Hardy, Nikhilesh Natraj, Adelyn Tu-Chan, Edward F. Chang, and Karunesh Ganguly. *Nature Biotechnology*, 39 (3):326–335, Mar 2021. ISSN 1546-1696. doi: 10.1038/s41587-020-0662-5. URL https://www.nature.com/articles/s41587-020-0662-5.

[31] Scott M Sternson and Bryan L Roth. Chemogenetic tools to interrogate brain functions. *Annual Review of Neuroscience*, 37:387–407, 2014. doi: 10.1146/annurev-neuro-071013-014048.

[32] Bryan L Roth. DREADDs for neuroscientists. *Neuron*, 89(4):683–694, 2016. doi: 10.1016/j.neuron.2016.01.040.

[33] Joseph Blackmore, Shamit Shrivastava, Jerome Sallet, Christopher R Butler, and Robin O Cleveland. Ultrasound neuromodulation: A review of results, mechanisms and safety. *Ultrasound in Medicine & Biology*, 45(7):1509–1536, 2019. doi: 10.1016/j.ultrasmedbio.2018.12.015.

[34] A. P. Georgopoulos, A. B. Schwartz, and R. E. Kettner. Neuronal population coding of movement direction. *Science*, 233(4771):1416–1419, 1986. doi: 10.1126/science.3749885.

[35] Meel Velliste, Sagi Perel, M Chance Spalding, Andrew S Whitford, and Andrew B Schwartz. Cortical control of a prosthetic arm for self-feeding. *Nature*, 453 (7198):1098–1101, 2008. doi: 10.1038/nature06996.

[36] B Wodlinger, J E Downey, E C Tyler-Kabara, A B Schwartz, M L Boninger, and J L Collinger. Ten-dimensional anthropomorphic arm control in a human brain-machine interface: difficulties, solutions, and limitations. *Journal of Neural Engineering*, 12(1):016011, Feb 2015. ISSN 1741-2560, 1741-2552. doi: 10.1088/1741-2560/12/1/016011. URL https://iopscience.iop.org/article/10.1088/1741-2560/12/1/016011.

[37] Vikash Gilja, Paul Nuyujukian, Cynthia A. Chestek, John P. Cunningham, Byron M. Yu, Julie M. Fan, Mark M. Churchland, Matthew T. Kaufman, Jonathan C. Kao, Stephen I. Ryu, and Krishna V. Shenoy. A high-performance neural prosthesis enabled by control algorithm design. *Nature Neuroscience*, 15 (12):1752–1757, 2012. doi: 10.1038/nn.3265.

[38] Mijail D Serruya, Nicholas G Hatsopoulos, Liam Paninski, Matthew R Fellows, and John P Donoghue. Instant neural control of a movement signal. *Nature*, 416(6877):141–142, 2002. doi: 10.1038/416141a.

[39] Dawn M Taylor, Stephen I Helms Tillery, and Andrew B Schwartz. Direct cortical control of 3D neuroprosthetic devices. *Science*, 296(5574):1829–1832, 2002. doi: 10.1126/science.1070291.

[40] Jose M Carmena, Mikhail A Lebedev, Roy E Crist, Joseph E O'Doherty, David M Santucci, Dragan F Dimitrov, Parag G Patil, Craig S Henriquez, and Miguel A. L Nicolelis. Learning to Control a Brain–Machine Interface for Reaching and Grasping by Primates. *PLOS Biology*, 1(2):null, October 2003. doi: 10.1371/journal.pbio.0000042. URL https://doi.org/10.1371/journal.pbio.0000042. Publisher: Public Library of Science.

[41] Leigh R Hochberg, Mijail D Serruya, Gerhard M Friehs, Jon A Mukand, Maryam Saleh, Abraham H Caplan, Almut Branner, David Chen, Richard D Penn, and John P Donoghue. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442(7099):164–171, 2006. doi: 10.1038/nature04970.

[42] Francis R Willett, Darrel R Deo, Donald T Avansino, Paymon Rezaii, Leigh R Hochberg, Jaimie M Henderson, and Krishna V Shenoy. Hand knob area of premotor cortex represents the whole body in a compositional way. *Cell*, 181(2): 396–409.e26, 2020. doi: 10.1016/j.cell.2020.02.043.

[43] Alexandriya M X Emonds, Elizaveta V Okorokova, Griffin H Blumenthal, Jennifer L Collinger, Sliman J Bensmaia, Lee E Miller, John E Downey, and Anton R Sobinov. Overlap in neural representations of coordinated wrist and finger movements in human motor cortex. *bioRxiv*, 2026. doi: 10.64898/2026.03.19.712976.

[44] John E Downey, Lucas Brane, Robert A Gaunt, Elizabeth C Tyler-Kabara, Michael L Boninger, and Jennifer L Collinger. Motor cortical activity changes during neuroprosthetic-controlled object interaction. *Scientific reports*, 7(1): 16947, 2017.

[45] Joseph G Makin, Joseph E O'Doherty, Mariana M B Cardoso, and Philip N

Sabes. Superior arm-movement decoding from cortex with a new, unsupervised-learning algorithm. *Journal of Neural Engineering*, 15(2):026010, Apr 2018. ISSN 1741-2560, 1741-2552. doi: 10.1088/1741-2552/aa9e95. URL https://iopscience.iop.org/article/10.1088/1741-2552/aa9e95.

[46] Brianna M Karpowicz, Yahia H Ali, Lahiru N Wimalasena, Andrew R Sedler, Mohammad Reza Keshtkaran, Kevin Bodkin, Xuan Ma, Daniel B Rubin, Ziv M Williams, Sydney S Cash, Leigh R Hochberg, Lee E Miller, and Chethan Pandarinath. Stabilizing brain-computer interfaces through alignment of latent dynamics. *Nature Communications*, 16(1):4662, 2025. doi: 10.1038/s41467-025-59652-y.

[47] James C Barrese, Naveen Rao, Kaivon Paroo, Corey Triebwasser, Carlos Vargas-Irwin, Lachlan Franquemont, and John P Donoghue. Failure mode analysis of silicon-based intracortical microelectrode arrays in non-human primates. *Journal of Neural Engineering*, 10(6):066014, 2013. doi: 10.1088/1741-2560/10/6/066014.

[48] Cynthia A Chestek, Vikash Gilja, Paul Nuyujukian, Justin D Foster, Joline M Fan, Matthew T Kaufman, Mark M Churchland, Zuley Rivera-Alvidrez, John P Cunningham, Stephen I Ryu, and Krishna V Shenoy. Long-term stability of neural prosthetic control signals from silicon cortical arrays in rhesus macaque motor cortex. *Journal of Neural Engineering*, 8(4):045005, 2011. doi: 10.1088/1741-2560/8/4/045005.

[49] Samuel C Colachis, Collin F Dunlap, Nicholas V Annetta, Sanjay M Tamrakar, Marcia A Bockbrader, and David A Friedenberg. Long-term intracortical microelectrode array performance in a human: a 5 year retrospective analysis. *Journal of Neural Engineering*, 18(4), 2021. doi: 10.1088/1741-2552/ac1add.

[50] Caleb Sponheim, Vasileios Papadourakis, Jennifer L Collinger, John Downey, Jeffrey Weiss, Lucine Pentousi, Kelly Elliott, and Nicholas G Hatsopoulos. Longevity and reliability of chronic unit recordings using the Utah, intracortical multielectrode array. *Journal of Neural Engineering*, 18(6), 2021. doi: 10.1088/1741-2552/ac3eaf.

[51] John E. Downey, Nathaniel Schwed, Steven M. Chase, Andrew B. Schwartz, and Jennifer L. Collinger. Intracortical recording stability in human brain–computer interface users. *Journal of Neural Engineering*, 15(4):046016, 2018. doi: 10.1088/1741-2552/aab7a0. URL https://doi.org/10.1088/1741-2552/aab7a0.

[52] János A. Perge, Mark L. Homer, Wasim Q. Malik, Sydney Cash, Emad Eskandar, Gerhard Friehs, John P. Donoghue, and Leigh R. Hochberg. Intra-day signal instabilities affect decoding performance in an intracortical neural interface system. *Journal of Neural Engineering*, 10(3):036004, 2013. doi: 10.1088/1741-2560/10/3/036004. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3693851/.

[53] Lahiru N. Wimalasena, Lee E. Miller, and Chethan Pandarinath. From unstable input to robust output. *Nature Biomedical Engineering*, 4(7):665–667, 2020. doi: 10.1038/s41551-020-0587-9. URL http://www.nature.com/articles/

s41551-020-0587-9.

[54] Matthew J Mender, Samuel R Nason-Tomaszewski, Hisham Temmar, Joseph T Costello, Dylan M Wallace, Matthew S Willsey, Nishant Ganesh Kumar, Theodore A Kung, Parag Patil, and Cynthia A Chestek. The impact of task context on predicting finger movements in a brain-machine interface. *eLife*, 12:e82598, jun 2023. ISSN 2050-084X. doi: 10.7554/eLife.82598. URL https://doi.org/10.7554/eLife.82598.

[55] Xuan Ma, Kevin L Bodkin, and Lee E Miller. Population activity in motor cortex is influenced by the contexts of the motor behavior. In *10th International IEEE/EMBS Conference on Neural Engineering*, 2021.

[56] Patrick J Marino, Lindsay Bahureksa, Carmen Fernández Fisac, Emily R Oby, Adam L Smoulder, Asma Motiwala, Alan D Degenhart, Erinn M Grigsby, Wilsaan M Joiner, Steven M Chase, et al. A posture subspace in primary motor cortex. *bioRxiv*, pages 2024–08, 2024.

[57] Benjamin R Cowley, Adam C Snyder, Katerina Acar, Ryan C Williamson, Byron M Yu, and Matthew A Smith. Slow drift of neural activity as a signature of impulsivity in macaque visual and prefrontal cortex. *Neuron*, 108(3):551–567, 2020.

[58] Adam L Smoulder, Patrick J Marino, Emily R Oby, Sam E Snyder, Hiroo Miyata, Nick P Pavlovsky, William E Bishop, Byron M Yu, Steven M Chase, and Aaron P Batista. A neural basis of choking under pressure. *Neuron*, 112 (20):3424–3433, 2024.

[59] Nicholas S Card, Maitreyee Wairagkar, Carrina Iacobacci, Xianda Hou, Tyler Singer-Clark, Francis R Willett, Erin M Kunz, Chaofei Fan, Maryam Vahdati Nia, Darrel R Deo, et al. An accurate and rapidly calibrating speech neuroprosthesis. *New England Journal of Medicine*, 391(7):609–618, 2024.

[60] David Sussillo, Sergey D. Stavisky, Jonathan C. Kao, Stephen I. Ryu, and Krishna V. Shenoy. Making brain–machine interfaces robust to future neural variability. *Nature Communications*, 7(1):13749, 2016. doi: 10.1038/ncomms13749. URL http://www.nature.com/articles/ncomms13749.

[61] Guy H. Wilson, Francis R. Willett, Elias A. Stein, Foram Kamdar, Donald T. Avansino, Leigh R. Hochberg, Krishna V. Shenoy, Shaul Druckmann, and Jaimie M. Henderson. Long-term unsupervised recalibration of cursor bcis. Feb 2023. doi: 10.1101/2023.02.03.527022. URL https://www.biorxiv.org/content/10.1101/2023.02.03.527022v1.

[62] Roland S Johansson and J Randall Flanagan. Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nature Reviews Neuroscience*, 10(5):345–359, 2009. doi: 10.1038/nrn2621.

[63] Sungshin Kim, Thierri Callier, Gregg A Tabot, Robert A Gaunt, Francesco V

Tenore, and Sliman J Bensmaia. Behavioral assessment of sensitivity to intracortical microstimulation of primate somatosensory cortex. *Proceedings of the National Academy of Sciences*, 112(49):15202–15207, 2015. doi: 10.1073/pnas.1509265112.

[64] Christopher L. Hughes, Sharlene N. Flesher, Jeffrey M. Weiss, Michael Boninger, Jennifer L. Collinger, and Robert A. Gaunt. Perception of microstimulation frequency in human somatosensory cortex. *eLife*, 10:e65128, 2021. doi: 10.7554/eLife.65128.

[65] Charles M. Greenspon, Giacomo Valle, Natalya D. Shelchkova, Taylor G. Hobbs, Ceci Verbaarschot, Thierri Callier, Ev I. Berger-Wolf, Elizaveta V. Okorokova, Brianna C. Hutchison, Efe Dogruoz, Anton R. Sobinov, Patrick M. Jordan, Jeffrey M. Weiss, Emily E. Fitzgerald, Dillan Prasad, Ashley Van Driesche, Qinpu He, Fang Liu, Robert F. Kirsch, Jonathan P. Miller, Ray C. Lee, David Satzer, Jorge Gonzalez-Martinez, Peter C. Warnke, Abidemi B. Ajiboye, Emily L. Graczyk, Michael L. Boninger, Jennifer L. Collinger, John E. Downey, Lee E. Miller, Nicholas G. Hatsopoulos, Robert A. Gaunt, and Sliman J. Bensmaia. Evoking stable and precise tactile sensations via multi-electrode intracortical microstimulation of the somatosensory cortex. *Nature Biomedical Engineering*, 9:935–951, 2025. doi: 10.1038/s41551-024-01299-z.

[66] Christopher L. Hughes, Sharlene N. Flesher, Jennifer L. Collinger, et al. Perception of microstimulation frequency in human somatosensory cortex. *eLife*, 10: e65128, Jul 2021. doi: 10.7554/eLife.65128.

[67] Christopher L Hughes, Sharlene N Flesher, Jeffrey M Weiss, John E Downey, Michael Boninger, Jennifer L Collinger, and Robert A Gaunt. Neural stimulation and recording performance in human sensorimotor cortex over 1500 days. *Journal of Neural Engineering*, 18(4):045012, 2021. doi: 10.1088/1741-2552/ac18ad.

[68] Charles M Greenspon, Taylor G Hobbs, Ceci Verbaarschot, Ali H Alamri, Natalya D Shelchkova, Robin Lienkämper, Joel Ye, Tyler W Simpson, Jeffrey M Weiss, David M Weir, Debbie E Harrington, Ashley Van Driesche, David Satzer, Giacomo Valle, Lee E Miller, Nicholas G Hatsopoulos, Jorge Gonzalez-Martinez, Peter C Warnke, John E Downey, Michael L Boninger, Jennifer L Collinger, and Robert A Gaunt. Long-term safety and efficacy of intracortical microstimulation in humans. *Science Translational Medicine*, 18(858):eaec3728, 2026. doi: 10.1126/scitranslmed.aec3728.

[69] Giacomo Valle, Ali H. Alamri, John E. Downey, Robin Lienkämper, Patrick M. Jordan, Anton R. Sobinov, Linnea J. Endsley, Dillan Prasad, Michael L. Boninger, Jennifer L. Collinger, Robert A. Gaunt, Nicholas G. Hatsopoulos, and Sliman J. Bensmaia. Tactile edges and motion via patterned microstimulation of the human somatosensory cortex. *Science*, 387(6731):315–322, 2025. doi: 10.1126/science.adq5978.

[70] Xing Chen, Feng Wang, Eduardo Fernandez, and Pieter R Roelfsema. Shape perception via a high-channel-count neuroprosthesis in monkey visual cortex. *Science*, 370(6521):1191–1196, 2020. doi: 10.1126/science.abd7435.

[71] Sliman J Bensmaia, Dustin J Tyler, and Silvestro Micera. Restoration of sensory information via bionic hands. *Nature Biomedical Engineering*, 7(4):443–455, 2023.

[72] Taylor G. Hobbs, Charles M. Greenspon, Ceci Verbaarschot, Giacomo Valle, Christopher Hughes, Michael L. Boninger, Sliman J. Bensmaia, and Robert A. Gaunt. Biomimetic stimulation patterns drive natural artificial touch percepts using intracortical microstimulation in humans. *Journal of Neural Engineering*, 22(3):036014, 2025. doi: 10.1088/1741-2552/adc2d4.

[73] Karthik Kumaravelu, Tucker Tomlinson, Thierri Callier, Joseph Sombeck, Sliman J. Bensmaia, Lee E. Miller, and Warren M. Grill. A comprehensive model-based framework for optimal design of biomimetic patterns of electrical stimulation for prosthetic sensation. *Journal of Neural Engineering*, 17(4): 046045, 2020. doi: 10.1088/1741-2552/abacd8.

[74] C J Brown, P J Abbas, and B J Gantz. Preliminary experience with neural response telemetry in the Nucleus CI24M cochlear implant. *The American Journal of Otology*, 19(3):320–327, 1998. URL `https://pubmed.ncbi.nlm.nih.gov/9596182/`.

[75] Alexander R Kent, Brandon D Swan, David T Brocker, Dennis A Turner, Robert E Gross, and Warren M Grill. Measurement of evoked potentials during thalamic deep brain stimulation. *Brain Stimulation*, 8(1):42–56, 2015. doi: 10.1016/j.brs.2014.09.017. URL `https://pubmed.ncbi.nlm.nih.gov/25457213/`.

[76] Nagy Mekhail, Robert M Levy, Timothy R Deer, Leonardo Kapural, Sean Li, Kasra Amirdelfan, Corey W Hunter, Steven M Rosen, Shrif J Costandi, Steven M Falowski, Abram H Burgher, Jason E Pope, Christopher A Gilmore, Farooq A Qureshi, Peter S Staats, James Scowcroft, Jonathan Carlson, Christopher K Kim, Michael I Yang, Thomas Stauss, Lawrence Poree, and Evoke Study Group. Long-term safety and efficacy of closed-loop spinal cord stimulation to treat chronic back and leg pain (Evoke): a double-blind, randomised, controlled trial. *The Lancet Neurology*, 19(2):123–134, 2020. doi: 10.1016/S1474-4422(19)30414-4. URL `https://pubmed.ncbi.nlm.nih.gov/31870766/`.

[77] Joseph T Sombeck, Juliet Heye, Karthik Kumaravelu, Stefan M Goetz, Angel V Peterchev, Warren M Grill, Sliman Bensmaia, and Lee E Miller. Characterizing the short-latency evoked response to intracortical microstimulation across a multielectrode array. *Journal of Neural Engineering*, 19(2):026044, 2022. doi: 10.1088/1741-2552/ac63e8. URL `https://pmc.ncbi.nlm.nih.gov/articles/PMC9142773/`.

[78] Yuki Minai, Joana Soldado-Magraner, Matthew A Smith, and Byron M Yu. MiSO: Optimizing brain stimulation to create neural population activity states.

In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://users.ece.cmu.edu/~byronyu/papers/MinaiNeurIPS2024.pdf.

[79] Pehuén Moure, Jacob Granley, Fabrizio Grani, Leili Soo, Antonio Lozano, Rocío López-Peco, Adrián Villamarín-Ortiz, Cristina Soto-Sánchez, Shih-Chii Liu, Michael Beyeler, and Eduardo Fernández. Deep learning-based control of electrically evoked activity in human visual cortex. *Neuron*, 2026. doi: 10.1016/j.neuron.2026.07.006.

[80] Fabrizio Grani, Cristina Soto-Sánchez, Alfonso Rodil Doblado, Rocio Lopez Peco, Pablo Gonzalez-Lopez, and Eduardo Fernandez. Neural correlates of phosphene perception in blind individuals: A step toward a bidirectional cortical visual prosthesis. *Science Advances*, 11(45):eadv8846, 2025. doi: 10.1126/sciadv.adv8846.

[81] Benyamin Haghi, Tyson Aflalo, Spencer Kellis, Charles Guan, Jorge A Gamez de Leon, Albert Yan Huang, Nader Pouratian, Richard A Andersen, and Azita Emami. Enhanced control of a brain–computer interface by tetraplegic participants via neural-network-mediated feature extraction. *Nature Biomedical Engineering*, 9(6):917–934, 2025. doi: 10.1038/s41551-024-01297-1.

[82] Patrick K Campbell, Kelly E Jones, Robert J Huber, Kenneth W Horch, and Richard A Normann. A silicon-based, three-dimensional neural interface: manufacturing processes for an intracortical electrode array. *IEEE Transactions on Biomedical Engineering*, 38(8):758–768, 1991. doi: 10.1109/10.83588.

[83] Edwin M Maynard, Craig T Nordhausen, and Richard A Normann. The Utah intracortical electrode array: a recording structure for potential brain-computer interfaces. *Electroencephalography and Clinical Neurophysiology*, 102(3):228–239, 1997. doi: 10.1016/S0013-4694(96)95176-0.

[84] Josh Merel, David Carlson, Liam Paninski, and John P Cunningham. Neuroprosthetic decoder training as imitation learning. *PLoS computational biology*, 12(5):e1004948, 2016.

[85] M. S. Willsey, S. R. Nason-Tomaszewski, S. R. Ensel, H. Temmar, M. J. Mender, J. T. Costello, et al. Real-time brain-machine interface in non-human primates achieves high-velocity prosthetic finger movements using a shallow feedforward neural network decoder. *Nature Communications*, 13:6899, 2022. doi: 10.1038/s41467-022-34452-w.

[86] Hisham Temmar, Matthew S Willsey, Joseph T Costello, Matthew J Mender, Luis H Cubillos, Jordan LW Lam, Dylan M Wallace, Madison M Kelberman, Parag G Patil, and Cynthia A Chestek. Artificial neural network for brain-machine interface consistently produces more naturalistic finger movements than linear methods. *bioRxiv*, pages 2024–03, 2024. URL https://www.biorxiv.org/content/10.1101/2024.03.01.583000v1.

[87] Chethan Pandarinath, Daniel J. O'Shea, Jasmine Collins, Rafal Jozefowicz,

Sergey D. Stavisky, Jonathan C. Kao, Eric M. Trautmann, Matthew T. Kaufman, Stephen I. Ryu, Leigh R. Hochberg, Jaimie M. Henderson, Krishna V. Shenoy, L. F. Abbott, and David Sussillo. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature Methods*, 15(10):805–815, 2018. doi: 10.1038/s41592-018-0109-9. URL http://www.nature.com/articles/s41592-018-0109-9.

[88] Joel Ye and Chethan Pandarinath. Representation learning for neural population activity with neural data transformers, 2021. URL https://doi.org/10.51628/001c.27358.

[89] Joseph T Costello, Hisham Temmar, Luis H Cubillos, Matthew J Mender, Dylan M Wallace, Matthew S Willsey, Parag G Patil, and Cynthia A Chestek. Balancing memorization and generalization in RNNs for high performance brain-machine interfaces. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://openreview.net/forum?id=FujJO3dsNj.

[90] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2022. URL https://arxiv.org/abs/2108.07258.

[91] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda

Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

[92] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

[93] Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. Scale efficiently: Insights from pre-training and fine-tuning transformers. 2022.

[94] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking ImageNet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. URL https://arxiv.org/abs/1811.08883.

[95] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc V. Le. Rethinking pre-training and self-training. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://papers.nips.cc/paper_files/paper/2020/hash/27e9661e033a73a6ad8cefcde965c54d-Abstract.html.

[96] Rahim Entezari, Mitchell Wortsman, Olga Saukh, M Moein Shariatnia, Hanie Sedghi, and Ludwig Schmidt. The role of pre-training data in transfer learning. *arXiv preprint arXiv:2302.13602*, 2023.

[97] Hubert Banville, Yohann Benchetrit, Stéphane d'Ascoli, Jérémy Rapin, and Jean-Rémi King. Scaling laws for decoding images from brain activity. *arXiv preprint arXiv:2501.15322*, 2025. doi: 10.48550/arXiv.2501.15322.

[98] CTRL-labs at Reality Labs, Patrick Kaifosh, and Thomas R Reardon. A generic non-invasive neuromotor interface for human-computer interaction. *Nature*, 645 (8081):702–711, 2025. doi: 10.1038/s41586-025-09255-w.

[99] Demetres Kostas, Stéphane Aroca-Ouellette, and Frank Rudzicz. BENDR: Using transformers and a contrastive self-supervised learning task to learn from massive amounts of EEG data. *Frontiers in Human Neuroscience*, 15, 2021. ISSN 1662-5161. doi: 10.3389/fnhum.2021.653659. URL https://www.frontiersin.org/articles/10.3389/fnhum.2021.653659.

[100] Christopher Wang, Vighnesh Subramaniam, Adam Uri Yaari, Gabriel Kreiman, Boris Katz, Ignacio Cases, and Andrei Barbu. BrainBERT: Self-supervised representation learning for intracranial recordings. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=xmcYx_reUn6.

[101] Weibang Jiang, Liming Zhao, and Bao liang Lu. Large brain model for learning generic representations with tremendous EEG data in BCI. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=QzTpTRVtrP.

[102] Zhizhang Yuan, Fanqi Shen, Meng Li, Yuguo Yu, Chenhao Tan, and Yang Yang. Brainwave: A brain signal foundation model for clinical applications, 2024. URL https://arxiv.org/abs/2402.10251.

[103] Armin W. Thomas, Christopher Ré, and Russell A. Poldrack. Self-supervised learning of brain dynamics from broad neuroimaging data, 2023.

[104] Josue Ortega Caro, Antonio Henrique de Oliveira Fonseca, Syed A Rizvi, Matteo Rosati, Christopher Averill, James L Cross, Prateek Mittal, Emanuele Zappala, Rahul Madhav Dhodapkar, Chadi Abdallah, and David van Dijk. BrainLM: A foundation model for brain activity recordings. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=RwI7ZEfR27.

[105] Geeling Chau, Christopher Wang, Sabera Talukder, Vighnesh Subramaniam, Saraswati Soedarmadji, Yisong Yue, Boris Katz, and Andrei Barbu. Population transformer: Learning population-level representations of intracranial activity, 2024. URL https://arxiv.org/abs/2406.03044.

[106] Jan-Matthis Lueckmann, Alexander Immer, Alex Bo-Yuan Chen, Peter H Li, Mariela D Petkova, Nirmala A Iyer, Luuk Willem Hesselink, Aparna Dev, Gudrun Ihrke, Woohyun Park, Alyson Petruncio, Aubrey Weigel, Wyatt Korff, Florian Engert, Jeff W Lichtman, Misha B Ahrens, Michał Januszewski, and Viren Jain. ZAPBench: A benchmark for whole-brain activity prediction in zebrafish. *arXiv preprint arXiv:2503.02618*, 2025. doi: 10.48550/arXiv.2503.02618.

[107] Ji Xia, Yizi Zhang, Shuqi Wang, Genevera I Allen, Liam Paninski, Cole Lincoln Hurwitz, and Kenneth D Miller. Inpainting the neural picture: Inferring unrecorded brain area dynamics from multi-animal datasets. *arXiv preprint arXiv:2510.11924*, 2025. doi: 10.48550/arXiv.2510.11924.

[108] Richard Antonello, Aditya Vaidya, and Alexander G. Huth. Scaling laws for language encoding models in fmri, 2024. URL https://arxiv.org/abs/2305.11863.

[109] Adrien Doerig, Rowan Sommers, Katja Seeliger, Blake Richards, Jenann Ismael, Grace Lindsay, Konrad Kording, Talia Konkle, Marcel A. J. Van Gerven, Nikolaus Kriegeskorte, and Tim C. Kietzmann. The neuroconnectionist research programme, 2022. URL https://arxiv.org/abs/2209.03718.

[110] Tyler Benster, Guy Wilson, Reshef Elisha, Francis R Willett, and Shaul Druckmann. A cross-modal approach to silent speech with llm-enhanced recognition. *arXiv preprint arXiv:2403.05583*, 2024.

[111] Weihao Xia, Raoul de Charette, Cengiz Öztireli, and Jing-Hao Xue. Umbrae:

Unified multimodal brain decoding. In *European Conference on Computer Vision (ECCV)*, 2024.

[112] Zhiwei Ding, Dat Tran, Kayla Ponder, Zhuokun Ding, Rachel Froebe, Lydia Ntanavara, Paul G Fahey, Erick Cobos, Luca Baroni, Eric Y Wang, Konstantin Willeke, Edgar Y Walker, Jacob Reimer, Fabian H Sinz, Alexander S Ecker, Katrin Franke, Xaq Pitkow, and Andreas S Tolias. Functional bipartite invariance in mouse primary visual cortex receptive fields. *Nature Neuroscience*, 29(4): 851–863, 2026. doi: 10.1038/s41593-026-02213-3.

[113] Sean L. Metzger, Kaylo T. Littlejohn, Alexander B. Silva, David A. Moses, Margaret P. Seaton, Ran Wang, Maximilian E. Dougherty, Jessie R. Liu, Peter Wu, Michael A. Berger, Inga Zhuravleva, Adelyn Tu-Chan, Karunesh Ganguly, Gopala K. Anumanchipalli, and Edward F. Chang. A high-performance neuroprosthesis for speech decoding and avatar control. *Nature*, 620(7976): 1037–1046, August 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06443-4. URL https://www.nature.com/articles/s41586-023-06443-4. Publisher: Nature Publishing Group.

[114] Yassine El Ouahidi, Jonathan Lys, Philipp Thölke, Nicolas Farrugia, Bastien Pasdeloup, Vincent Gripon, Karim Jerbi, and Giulia Lioi. REVE: A foundation model for EEG – adapting to any setup with large-scale pretraining on 25,000 subjects. *arXiv preprint arXiv:2510.21585*, 2025. doi: 10.48550/arXiv.2510.21585.

[115] Jerry Tang, Amanda LeBel, Shailee Jain, and Alexander G Huth. Semantic reconstruction of continuous language from non-invasive brain recordings. *Nature Neuroscience*, 26(5):858–866, 2023. doi: 10.1038/s41593-023-01304-9.

[116] Paul S Scotti, Atmadeep Banerjee, Jimmie Goode, Stepan Shabalin, Alex Nguyen, Ethan Cohen, Aidan J Dempster, Nathalie Verlinde, Elad Yundler, David Weisberg, Kenneth A Norman, and Tanishq Mathew Abraham. Reconstructing the mind's eye: fMRI-to-image with contrastive learning and diffusion priors. In *Advances in Neural Information Processing Systems*, 2023. doi: 10.48550/arXiv.2305.18274.

[117] Paul S Scotti, Mihir Tripathy, Cesar Kadir Torrico Villanueva, Reese Kneeland, Tong Chen, Ashutosh Narang, Charan Santhirasegaran, Jonathan Xu, Thomas Naselaris, Kenneth A Norman, and Tanishq Mathew Abraham. MindEye2: Shared-subject models enable fMRI-to-image with 1 hour of data. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. doi: 10.48550/arXiv.2403.11207.

[118] Saurabh Vyas, Matthew D Golub, David Sussillo, and Krishna V Shenoy. Computation through neural population dynamics. *Annual review of neuroscience*, 43:249–275, 2020.

[119] Ege Altan, Sara A. Solla, Lee E. Miller, and Eric J. Perreault. Estimating the

dimensionality of the manifold underlying multi-electrode neural recordings. *PLOS Computational Biology*, 17(11):1–23, 11 2021. doi: 10.1371/journal.pcbi.1008591. URL https://doi.org/10.1371/journal.pcbi.1008591.

[120] Daniel J O'Shea, Lea Duncker, Werapong Goo, Xulu Sun, Saurabh Vyas, Eric M Trautmann, Ilka Diester, Charu Ramakrishnan, Karl Deisseroth, Maneesh Sahani, et al. Direct neural perturbations reveal a dynamical mechanism for robust computation. *bioRxiv*, pages 2022–12, 2022.

[121] Patrick T Sadtler, Kristin M Quick, Matthew D Golub, Steven M Chase, Stephen I Ryu, Elizabeth C Tyler-Kabara, Byron M Yu, and Aaron P Batista. Neural constraints on learning. *Nature*, 512(7515):423–426, 2014.

[122] Saurabh Vyas, Nir Even-Chen, Sergey D Stavisky, Stephen I Ryu, Paul Nuyujukian, and Krishna V Shenoy. Neural population dynamics underlying motor learning transfer. *Neuron*, 97(5):1177–1186, 2018.

[123] Juan A. Gallego, Matthew G. Perich, Stephanie N. Naufel, Christian Ethier, Sara A. Solla, and Lee E. Miller. Cortical population activity within a preserved neural manifold underlies multiple motor behaviors. *Nature Communications*, 9 (1):4233, Oct 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-06560-z. URL https://www.nature.com/articles/s41467-018-06560-z.

[124] Juan A. Gallego, Matthew G. Perich, Raeed H. Chowdhury, Sara A. Solla, and Lee E. Miller. Long-term stability of cortical population dynamics underlying consistent behavior. *Nature Neuroscience*, 23(2):260–270, 2020. doi: 10.1038/s41593-019-0555-4. URL http://www.nature.com/articles/s41593-019-0555-4.

[125] Mostafa Safaie, Joanna C. Chang, Junchol Park, Lee E. Miller, Joshua T. Dudman, Matthew G. Perich, and Juan A. Gallego. Preserved neural dynamics across animals performing similar behaviour. *Nature*, 623(7988):765–771, November 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06714-0. URL https://www.nature.com/articles/s41586-023-06714-0. Publisher: Nature Publishing Group.

[126] Max Dabagia, Konrad P. Kording, and Eva L. Dyer. Aligning latent representations of neural activity. *Nature Biomedical Engineering*, 7(4):337–343, Apr 2023. ISSN 2157-846X. doi: 10.1038/s41551-022-00962-7. URL https://www.nature.com/articles/s41551-022-00962-7.

[127] Felix Pei, Joel Ye, David Zoltowski, Anqi Wu, Raeed H. Chowdhury, Hansem Sohn, Joseph E. O'Doherty, Krishna V. Shenoy, Matthew T. Kaufman, Mark Churchland, Mehrdad Jazayeri, Lee E. Miller, Jonathan Pillow, Il Memming Park, Eva L. Dyer, and Chethan Pandarinath. Neural latents benchmark '21: Evaluating latent variable models of neural population activity, 2022.

[128] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan,

and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[129] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. URL https://arxiv.org/abs/2111.06377.

[130] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[131] Sabera J Talukder, Jennifer J. Sun, Matthew K Leonard, Bingni W Brunton, and Yisong Yue. Deep neural imputation: A framework for recovering incomplete brain recordings. In *NeurIPS 2022 Workshop on Learning from Time Series for Health*, 2022. URL https://openreview.net/forum?id=c9qFg8UrIcn.

[132] Samuel M Peterson, Shiva H Singh, Benjamin Dichter, Kelvin Tan, Craig DiBartolomeo, Devapratim Theogarajan, Peter Fisher, and Josef Parvizi. Ajile12: Long-term naturalistic human intracranial neural recordings and pose. *Scientific Data*, 9(1):184, 2022. ISSN 2052-4463. doi: 10.1038/s41597-022-01280-y. URL https://doi.org/10.1038/s41597-022-01280-y.

[133] Iyad Obeid and Joseph Picone. The temple university hospital EEG data corpus. *Frontiers in Neuroscience*, 10:196, 2016. ISSN 1662-453X. doi: 10.3389/fnins.2016.00196. URL https://www.frontiersin.org/articles/10.3389/fnins.2016.00196.

[134] Srini Turaga, Lars Buesing, Adam M Packer, Henry Dalgleish, Noah Pettit, Michael Hausser, and Jakob H Macke. Inferring neural population dynamics from multiple partial recordings of the same neural circuit. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/01386bd6d8e091c2ab4c7c7de644d37b-Paper.pdf.

[135] Francis R. Willett, Donald T. Avansino, Leigh R. Hochberg, Jaimie M. Henderson, and Krishna V. Shenoy. High-performance brain-to-text communication via handwriting. *Nature*, 593(7858):249–254, May 2021. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-021-03506-2. URL http://www.nature.com/articles/s41586-021-03506-2.

[136] Darrel R. Deo, Francis R. Willett, Donald T. Avansino, Leigh R. Hochberg, Jaimie M. Henderson, and Krishna V. Shenoy. Translating deep learning to neuroprosthetic control. Apr 2023. doi: 10.1101/2023.04.21.537581. URL https://www.biorxiv.org/content/10.1101/2023.04.21.537581v1.

[137] Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, 617 (7960):360–368, May 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06031-6.

URL https://www.nature.com/articles/s41586-023-06031-6. Publisher: Nature Publishing Group.

[138] Nur Ahmadi, Timothy G. Constandinou, and Christos-Savvas Bouganis. Robust and accurate decoding of hand kinematics from entire spiking activity using deep learning. *Journal of Neural Engineering*, 18(2):026011, Feb 2021. ISSN 1741-2552. doi: 10.1088/1741-2552/abde8a. URL https://iopscience.iop.org/article/10.1088/1741-2552/abde8a/meta.

[139] Justin Jude, Matthew G. Perich, Lee E. Miller, and Matthias H. Hennig. Robust alignment of cross-session recordings of neural population activity by behaviour via unsupervised domain adaptation. *arXiv:2202.06159 [q-bio.NC]*, February 2022. URL http://arxiv.org/abs/2202.06159. arXiv: 2202.06159.

[140] Justin Jude, Matthew G. Perich, Lee E. Miller, and Matthias H. Hennig. Capturing cross-session neural population variability through self-supervised identification of consistent neuron ensembles. In *Proceedings of the 1st NeurIPS Workshop on Symmetry and Geometry in Neural Representations*, page 234–257. PMLR, Feb 2023. URL https://proceedings.mlr.press/v197/jude23a.html.

[141] Ran Liu, Mehdi Azabou, Max Dabagia, Jingyun Xiao, and Eva L. Dyer. Seeing the forest and the tree: Building representations of both individual and collective dynamics with transformers. Oct 2022. URL https://openreview.net/forum?id=5aZ8umizItU.

[142] Ali Farshchian, Juan A. Gallego, Joseph P. Cohen, Yoshua Bengio, Lee E. Miller, and Sara A. Solla. Adversarial domain adaptation for stable brain-machine interfaces. Jan 2019. URL https://openreview.net/forum?id=Hyx6Bi0qYm.

[143] Xuan Ma, Fabio Rizzoglio, Kevin L Bodkin, Eric Perreault, Lee E Miller, and Ann Kennedy. Using adversarial networks to extend brain computer interface decoding accuracy over time. *eLife*, 12:e84296, August 2023. ISSN 2050-084X. doi: 10.7554/eLife.84296. URL https://doi.org/10.7554/eLife.84296. Publisher: eLife Sciences Publications, Ltd.

[144] Trung Le and Eli Shlizerman. Stndt: Modeling neural population activity with a spatiotemporal transformer, 2022.

[145] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer, 2021.

[146] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[147] Maryam M. Shanechi, Amy L. Orsborn, Helene G. Moorman, Suraj Gowda,

Siddharth Dangi, and Jose M. Carmena. Rapid control and feedback rates enhance neuroprosthetic control. *Nature Communications*, 8(1):13825, Jan 2017. ISSN 2041-1723. doi: 10.1038/ncomms13825. URL https://www.nature.com/articles/ncomms13825.

[148] Eric M. Trautmann, Janis K. Hesse, Gabriel M. Stine, Ruobing Xia, Shude Zhu, Daniel J. O'Shea, Bill Karsh, Jennifer Colonell, Frank F. Lanfranchi, Saurabh Vyas, Andrew Zimnik, Natalie A. Steinmann, Daniel A. Wagenaar, Alexandru Andrei, Carolina Mora Lopez, John O'Callaghan, Jan Putzeys, Bogdan C. Raducanu, Marleen Welkenhuysen, Mark Churchland, Tirin Moore, Michael Shadlen, Krishna Shenoy, Doris Tsao, Barundeb Dutta, and Timothy Harris. Large-scale high-density brain-wide neural recording in nonhuman primates. *bioRxiv*, 2023. doi: 10.1101/2023.02.01.526664. URL https://www.biorxiv.org/content/early/2023/05/04/2023.02.01.526664.

[149] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning, 2021.

[150] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. *arXiv preprint arXiv:2004.00784*, 2020.

[151] Joseph E. O'Doherty, Mariana M. B. Cardoso, Joseph G. Makin, and Philip N. Sabes. Nonhuman primate reaching with multichannel sensorimotor cortex electrophysiology, May 2017. URL https://doi.org/10.5281/zenodo.788569.

[152] Brian M Dekleva, Jeffrey M Weiss, Michael L Boninger, and Jennifer L Collinger. Generalizable cursor click decoding using grasp-related neural transients. *Journal of Neural Engineering*, 18(4):0460e9, Aug 2021. ISSN 1741-2560, 1741-2552. doi: 10.1088/1741-2552/ac16b2. URL https://iopscience.iop.org/article/10.1088/1741-2552/ac16b2.

[153] Kundan Krishna, Saurabh Garg, Jeffrey P. Bigham, and Zachary C. Lipton. Downstream datasets make surprisingly good pretraining corpora, 2022.

[154] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning, 2020. URL https://arxiv.org/abs/1912.11370.

[155] Amir Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning, 2018.

[156] Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer, 2021.

[157] Fabio Rizzoglio, Ege Altan, Xuan Ma, Kevin L. Bodkin, Brian M. Dekleva, Sara A. Solla, Ann Kennedy, and Lee E. Miller. Monkey-to-human transfer of brain-computer interface decoders. *bioRxiv*, 2022. doi: 10.1101/2022.11.12.515040. URL https://www.biorxiv.org/content/early/2022/11/13/2022.11.12.

515040.

[158] Wei Wang, Sherwin S. Chan, Dustin A. Heldman, and Daniel W. Moran. Motor cortical representation of position and velocity during reaching. *Journal of Neurophysiology*, 97(6):4258–4270, Jun 2007. ISSN 0022-3077. doi: 10.1152/jn.01180.2006.

[159] Beata Jarosiewicz, Anish A Sarma, Jad Saab, Brian Franco, Sydney S Cash, Emad N Eskandar, and Leigh R Hochberg. Retrospectively supervised click decoder calibration for self-calibrating point-and-click brain–computer interfaces. *Journal of Physiology-Paris*, 110(4):382–391, 2016.

[160] Jensen Gao, Siddharth Reddy, Glen Berseth, Nicholas Hardy, Nikhilesh Natraj, Karunesh Ganguly, Anca D Dragan, and Sergey Levine. X2t: Training an x-to-text typing interface with online learning from user feedback. *arXiv preprint arXiv:2203.02072*, 2022.

[161] Nir Even-Chen, Sergey D Stavisky, Jonathan C Kao, Stephen I Ryu, and Krishna V Shenoy. Augmenting intracortical brain-machine interface with neurally driven error detectors. *Journal of neural engineering*, 14(6):066007, 2017.

[162] Iñaki Iturrate, Ricardo Chavarriaga, Luis Montesano, Javier Minguez, and José del R. Millán. Teaching brain-machine interfaces as an alternative paradigm to neuroprosthetics control. *Scientific Reports*, 5(1):13893, Sep 2015. ISSN 2045-2322. doi: 10.1038/srep13893. URL https://www.nature.com/articles/srep13893.

[163] Francis R. Willett, Daniel R. Young, Brian A. Murphy, William D. Memberg, Christine H. Blabe, Chethan Pandarinath, Sergey D. Stavisky, Paymon Rezaii, Jad Saab, Benjamin L. Walter, Jennifer A. Sweet, Jonathan P. Miller, Jaimie M. Henderson, Krishna V. Shenoy, John D. Simeral, Beata Jarosiewicz, Leigh R. Hochberg, Robert F. Kirsch, and A. Bolu Ajiboye. Principled bci decoder design and parameter selection using a feedback control model. *Scientific Reports*, 9 (1):8881, Jun 2019. ISSN 2045-2322. doi: 10.1038/s41598-019-44166-7. URL https://doi.org/10.1038/s41598-019-44166-7.

[164] Unknown. EvalAI leaderboard. https://eval.ai/web/challenges/challenge-page/1256/leaderboard/3184, 2022. Accessed on May 16, 2023.

[165] Paul Nuyujukian, Jose Albites Sanabria, Jad Saab, Chethan Pandarinath, Beata Jarosiewicz, Christine H. Blabe, Brian Franco, Stephen T. Mernoff, Emad N. Eskandar, John D. Simeral, Leigh R. Hochberg, Krishna V. Shenoy, and Jaimie M. Henderson. Cortical control of a tablet computer by people with paralysis. *PLOS ONE*, 13(11):e0204566, November 2018. ISSN 1932-6203. doi: 10.1371/journal.pone.0204566. URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0204566. Publisher: Public Library of Science.

[166] Tyson Aflalo, Spencer Kellis, Christian Klaes, Brian Lee, Ying Shi, Kelsie Pejsa, Kathleen Shanfield, Stephanie Hayes-Jackson, Mindy Aisen,

Christi Heck, Charles Liu, and Richard A. Andersen. Decoding motor imagery from the posterior parietal cortex of a tetraplegic human. *Science*, 348(6237):906–910, 2015. doi: 10.1126/science.aaa5417. URL https://www.science.org/doi/abs/10.1126/science.aaa5417. _eprint: https://www.science.org/doi/pdf/10.1126/science.aaa5417.

[167] Peter Brunner, Anthony L. Ritaccio, Joseph F. Emrich, Horst Bischof, and Gerwin Schalk. Rapid Communication with a "P300" Matrix Speller Using Electrocorticographic Signals (ECoG). *Frontiers in Neuroscience*, 5, 2011. ISSN 1662-453X. doi: 10.3389/fnins.2011.00005. URL https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2011.00005.

[168] Karunesh Ganguly, Dragan F Dimitrov, Jonathan D Wallis, and Jose M Carmena. Reversible large-scale modification of cortical networks during neuroprosthetic control. *Nature Neuroscience*, 14(5):662–667, May 2011. ISSN 1546-1726. doi: 10.1038/nn.2797. URL https://doi.org/10.1038/nn.2797.

[169] Gopala K. Anumanchipalli, Josh Chartier, and Edward F. Chang. Speech synthesis from neural decoding of spoken sentences. *Nature*, 568(7753):493–498, April 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-1119-1. URL https://www.nature.com/articles/s41586-019-1119-1. Publisher: Nature Publishing Group.

[170] Christian Herff, Dominic Heger, Adriana de Pesters, Dominic Telaar, Peter Brunner, Gerwin Schalk, and Tanja Schultz. Brain-to-text: decoding spoken phrases from phone representations in the brain. *Frontiers in Neuroscience*, 8, 2015. ISSN 1662-453X. doi: 10.3389/fnins.2015.00217. URL https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2015.00217.

[171] Chethan Pandarinath, K. Cora Ames, Abigail A. Russo, Ali Farshchian, Lee E. Miller, Eva L. Dyer, and Jonathan C. Kao. Latent Factors and Dynamics in Motor Cortex and Their Application to Brain–Machine Interfaces. *J Neurosci*, 38(44):9390–9401, October 2018. doi: https://doi.org/10.1523/JNEUROSCI.1669-18.2018. URL https://www.jneurosci.org/content/38/44/9390.

[172] Eva L. Dyer, Mohammad Gheshlaghi Azar, Matthew G. Perich, Hugo L. Fernandes, Stephanie Naufel, Lee E. Miller, and Konrad P. Körding. A cryptography-based approach for movement decoding. *Nat Biomed Eng*, 1(12):967–976, December 2017. ISSN 2157-846X. doi: 10.1038/s41551-017-0169-7. URL http://www.nature.com/articles/s41551-017-0169-7.

[173] Alan D. Degenhart, William E. Bishop, Emily R. Oby, Elizabeth C. Tyler-Kabara, Steven M. Chase, Aaron P. Batista, and Byron M. Yu. Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity. *Nat Biomed Eng*, 4(7):672–685, July 2020. ISSN 2157-846X. doi: 10.1038/s41551-020-0542-9. URL http://www.nature.com/articles/s41551-020-0542-9.

[174] Ali Farshchian, Juan A. Gallego, Joseph P. Cohen, Yoshua Bengio, Lee E. Miller, and Sara A. Solla. Adversarial Domain Adaptation for Stable Brain-Machine Interfaces. *ICLR 2019*, January 2019. URL http://arxiv.org/abs/1810.00045. arXiv: 1810.00045.

[175] Yule Wang, Zijing Wu, Chengrui Li, and Anqi Wu. Extraction and recovery of spatio-temporal structure in latent dynamics alignment with diffusion models. *Advances in Neural Information Processing Systems*, August 2023. URL https://neurips.cc/virtual/2023/poster/72520.

[176] Beata Jarosiewicz, Anish A. Sarma, Daniel Bacher, Nicolas Y. Masse, John D. Simeral, Brittany Sorice, Erin M. Oakley, Christine Blabe, Chethan Pandarinath, Vikash Gilja, Sydney S. Cash, Emad N. Eskandar, Gerhard Friehs, Jaimie M. Henderson, Krishna V. Shenoy, John P. Donoghue, and Leigh R. Hochberg. Virtual typing by people with tetraplegia using a self-calibrating intracortical brain-computer interface. *Science Translational Medicine*, 7(313):313ra179, 2015. doi: 10.1126/scitranslmed.aac7328. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4765319/.

[177] Guy H. Wilson, Francis R. Willett, Elias A. Stein, Foram Kamdar, Donald T. Avansino, Leigh R. Hochberg, Krishna V. Shenoy, Shaul Druckmann, and Jaimie M. Henderson. Long-term unsupervised recalibration of cursor BCIs, February 2023. URL https://www.biorxiv.org/content/10.1101/2023.02.03.527022v1. Pages: 2023.02.03.527022 Section: New Results.

[178] Chaofei Fan, Nick Hahn, Foram Kamdar, Donald Avansino, Guy Wilson, Leigh Hochberg, Krishna V. Shenoy, Jaimie Henderson, and Francis Willett. Plug-and-Play Stability for Intracortical Brain-Computer Interfaces: A One-Year Demonstration of Seamless Brain-to-Text Communication. *Advances in Neural Information Processing Systems*, 36:42258–42270, December 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/83a14a36de4502bac5b580db36e81858-Abstract-Conference.html.

[179] Joel Ye, Jennifer L. Collinger, Leila Wehbe, and Robert Gaunt. Neural Data Transformer 2: Multi-context pretraining for neural spiking activity. *Advances in Neural Information Processing Systems*, 2023. doi: 10.1101/2023.09.18.558113. URL https://www.biorxiv.org/content/early/2023/09/22/2023.09.18.558113.

[180] Mehdi Azabou, Vinam Arora, Venkataramana Ganesh, Ximeng Mao, Santosh Nachimuthu, Michael J. Mendelson, Blake Richards, Matthew G. Perich, Guillaume Lajoie, and Eva L. Dyer. A unified, scalable framework for neural population decoding. *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2310.16046.

[181] P. Sajda, A. Gerson, K.-R. Muller, B. Blankertz, and L. Parra. A data analysis competition to evaluate machine learning algorithms for use in brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*,

11(2):184–185, June 2003. ISSN 1534-4320, 1558-0210. doi: 10.1109/TNSRE.2003.814453. URL https://ieeexplore.ieee.org/document/1214716/.

[182] Ji-Hoon Jeong, Jeong-Hyun Cho, Young-Eun Lee, Seo-Hyun Lee, Gi-Hwan Shin, Young-Seok Kweon, José del R. Millán, Klaus-Robert Müller, and Seong-Whan Lee. 2020 International brain–computer interface competition: A review. *Frontiers in Human Neuroscience*, 16, July 2022. ISSN 1662-5161. doi: 10.3389/fnhum.2022.898300. URL https://www.frontiersin.org/articles/10.3389/fnhum.2022.898300.

[183] Vinay Jayaram and Alexandre Barachant. MOABB: trustworthy algorithm benchmarking for BCIs. *Journal of Neural Engineering*, 15(6):066011, 2018. URL https://iopscience.iop.org/article/10.1088/1741-2552/aadea0.

[184] NPTL. Brain-to-text benchmark '24, 2024. URL https://eval.ai/web/challenges/challenge-page/2099/leaderboard/4944. Accessed: 2024-05-21.

[185] Zhizhang Yuan, Daoze Zhang, Junru Chen, Gefei Gu, and Yang Yang. Brant-2: Foundation Model for Brain Signals, March 2024. URL http://arxiv.org/abs/2402.10251. arXiv:2402.10251 [cs, eess, q-bio] version: 4.

[186] Chaoqi Yang, M. Brandon Westover, and Jimeng Sun. BIOT: Cross-data Biosignal Learning in the Wild, May 2023. URL http://arxiv.org/abs/2305.10351. arXiv:2305.10351 [cs, eess].

[187] Geeling Chau, Yujin An, Ahamed Raffey Iqbal, Soon-Jo Chung, Yisong Yue, and Sabera Talukder. Generalizability under sensor failure: Tokenization + transformers enable more robust latent spaces, 2024. URL https://arxiv.org/abs/2402.18546.

[188] Vladislav Lomtev, Alexander Kovalev, and Alexey Timchenko. Fingerflex: Inferring finger trajectories from ECoG signals, 2023. URL https://arxiv.org/abs/2211.01960.

[189] Josue Ortega Caro, Antonio H. de O. Fonseca, Christopher Averill, Syed A. Rizvi, Matteo Rosati, James L. Cross, Prateek Mittal, Emanuele Zappala, Daniel Levine, Rahul M. Dhodapkar, Insu Han, Amin Karbasi, Chadi G. Abdallah, and David van Dijk. BrainLM: A foundation model for brain activity recordings. *bioRxiv*, 2024. doi: 10.1101/2023.09.12.557460. URL https://www.biorxiv.org/content/early/2024/01/13/2023.09.12.557460.

[190] Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J Majaj, Rishi Rajalingham, Elias B Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Franziska Geiger, et al. Brain-score: Which artificial neural network for object recognition is most brain-like? *BioRxiv*, page 407007, 2018. URL https://www.biorxiv.org/content/10.1101/407007v1.

[191] Konstantin F. Willeke, Paul G. Fahey, Mohammad Bashiri, Laura Pede, Max F. Burg, Christoph Blessing, Santiago A. Cadena, Zhiwei Ding, Konstantin-Klemens

Lurz, Kayla Ponder, Taliah Muhammad, Saumil S. Patel, Alexander S. Ecker, Andreas S. Tolias, and Fabian H. Sinz. The Sensorium competition on predicting large-scale mouse primary visual cortex activity, 2022. URL https://arxiv.org/abs/2206.08666.

[192] A. T. Gifford, B. Lahner, S. Saba-Sadiya, M. G. Vilas, A. Lascelles, A. Oliva, K. Kay, G. Roig, and R. M. Cichy. The Algonauts Project 2023 Challenge: How the Human Brain Makes Sense of Natural Scenes, 2023. URL https://arxiv.org/abs/2301.03198.

[193] Felix Pei, Joel Ye, David Zoltowski, Anqi Wu, Raeed H. Chowdhury, Hansem Sohn, Joseph E. O'Doherty, Krishna V. Shenoy, Matthew T. Kaufman, Mark Churchland, Mehrdad Jazayeri, Lee E. Miller, Jonathan Pillow, Il Memming Park, Eva L. Dyer, and Chethan Pandarinath. Neural Latents Benchmark '21: Evaluating latent variable models of neural population activity. *Advances in Neural Information Processing Systems (NeurIPS) 34, Track on Datasets and Benchmarks*, January 2022. URL http://arxiv.org/abs/2109.04463. arXiv: 2109.04463.

[194] Deshraj Yadav, Rishabh Jain, Harsh Agrawal, Prithvijit Chattopadhyay, Taranjeet Singh, Akash Jain, Shiv Baran Singh, Stefan Lee, and Dhruv Batra. EvalAI: Towards better evaluation systems for AI agents. *arXiv preprint arXiv:1902.03570*, 2019. URL https://arxiv.org/abs/1902.03570.

[195] Darrel R Deo, Francis R Willett, Donald T Avansino, Leigh R Hochberg, Jaimie M Henderson, and Krishna V Shenoy. Brain control of bimanual movement enabled by recurrent neural networks. *Scientific Reports*, 14(1):1598, 2024. URL https://www.nature.com/articles/s41598-024-51617-3.

[196] Fabio Rizzoglio, Ege Altan, Xuan Ma, Kevin L. Bodkin, Brian M. Dekleva, Sara A. Solla, Ann Kennedy, and Lee E. Miller. From monkeys to humans: observation-based EMG brain–computer interface decoders for humans with paralysis. *J. Neural Eng.*, 20(5):056040, November 2023. ISSN 1741-2552. doi: 10.1088/1741-2552/ad038e. URL https://dx.doi.org/10.1088/1741-2552/ad038e. Publisher: IOP Publishing.

[197] Breanne P. Christie, Derek M. Tat, Zachary T. Irwin, Vikash Gilja, Paul Nuyujukian, Justin D. Foster, Stephen I. Ryu, Krishna V. Shenoy, David E. Thompson, and Cynthia A. Chestek. Comparison of spike sorting and thresholding of voltage waveforms for intracortical brain-machine interface performance. *Journal of neural engineering*, 12(1):016009, February 2015. ISSN 1741-2560. doi: 10.1088/1741-2560/12/1/016009. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4332592/.

[198] Paul Nuyujukian, Joline M. Fan, Vikash Gilja, Paul S. Kalanithi, Cindy A. Chestek, and Krishna V. Shenoy. Monkey models for brain-machine interfaces: the need for maintaining diversity. *Annu Int Conf IEEE Eng Med Biol Soc*, 2011:

1301–1305, 2011. ISSN 2694-0604. doi: 10.1109/IEMBS.2011.6090306. URL https://pubmed.ncbi.nlm.nih.gov/22254555/.

[199] Adam G. Rouse and Marc H. Schieber. Spatiotemporal distribution of location and object effects in reach-to-grasp kinematics. *Journal of Neurophysiology*, 114(6):3268–3282, December 2015. ISSN 0022-3077. doi: 10.1152/jn.00686.2015. URL https://journals.physiology.org/doi/full/10.1152/jn.00686.2015. Publisher: American Physiological Society.

[200] Adam G. Rouse and Marc H. Schieber. Spatiotemporal distribution of location and object effects in the electromyographic activity of upper extremity muscles during reach-to-grasp. *Journal of Neurophysiology*, 115(6):3238–3248, June 2016. ISSN 0022-3077. doi: 10.1152/jn.00008.2016. URL https://journals.physiology.org/doi/full/10.1152/jn.00008.2016. Publisher: American Physiological Society.

[201] Adam G. Rouse and Marc H. Schieber. Spatiotemporal distribution of location and object effects in primary motor cortex neurons during reach-to-grasp. *Journal of Neuroscience*, 36(41):10640–10653, 2016. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.1716-16.2016. URL https://www.jneurosci.org/content/36/41/10640.

[202] Adam G. Rouse and Marc H. Schieber. Condition-Dependent Neural Dimensions Progressively Shift during Reach to Grasp. *Cell Reports*, 25(11):3158–3168.e3, December 2018. ISSN 2211-1247. doi: 10.1016/j.celrep.2018.11.057. URL https://www.cell.com/cell-reports/abstract/S2211-1247(18)31833-3. Publisher: Elsevier.

[203] Christian Ethier, Emily R Oby, Matthew J Bauman, and Lee E Miller. Restoration of grasp following paralysis through brain-controlled stimulation of muscles. *Nature*, 485(7398):368–371, 2012. URL https://www.nature.com/articles/nature10987.

[204] A. Bolu Ajiboye, Francis R. Willett, Daniel R. Young, William D. Memberg, Brian A. Murphy, Jonathan P. Miller, Benjamin L. Walter, Jennifer A. Sweet, Harry A. Hoyen, Michael W. Keith, P. Hunter Peckham, John D. Simeral, John P. Donoghue, Leigh R. Hochberg, and Robert F. Kirsch. Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: a proof-of-concept demonstration. *The Lancet*, 389(10081):1821–1830, May 2017. ISSN 0140-6736, 1474-547X. doi: 10.1016/S0140-6736(17)30601-3. URL https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(17)30601-3/fulltext. Publisher: Elsevier.

[205] Samuel R. Nason, Matthew J. Mender, Alex K. Vaskov, Matthew S. Willsey, Nishant Ganesh Kumar, Theodore A. Kung, Parag G. Patil, and Cynthia A. Chestek. Real-time linear prediction of simultaneous and independent movements of two finger groups using an intracortical brain-machine interface. *Neuron*, 109(19):3164–3177.e8, October 2021. ISSN 0896-6273. doi: 10.1016/j.neuron.2021.08.009.

URL https://www.cell.com/neuron/abstract/S0896-6273(21)00604-8. Publisher: Elsevier.

[206] Nishal P Shah, Donald Avansino, Foram Kamdar, Claire Nicolas, Anastasia Kapitonava, Carlos Vargas-Irwin, Leigh R Hochberg, Chethan Pandarinath, Krishna V Shenoy, Francis R Willett, et al. Pseudo-linear summation explains neural geometry of multi-finger movements in human premotor cortex. *Nature Communications*, 16(1):5008, 2025.

[207] James J. Jun, Nicholas A. Steinmetz, Joshua H. Siegle, Daniel J. Denman, Marius Bauza, Brian Barbarits, Albert K. Lee, Costas A. Anastassiou, Alexandru Andrei, Çağatay Aydın, Mladen Barbic, Timothy J. Blanche, Vincent Bonin, João Couto, Barundeb Dutta, Sergey L. Gratiy, Diego A. Gutnisky, Michael Häusser, Bill Karsh, Peter Ledochowitsch, Carolina Mora Lopez, Catalin Mitelut, Silke Musa, Michael Okun, Marius Pachitariu, Jan Putzeys, P. Dylan Rich, Cyrille Rossant, Wei-lung Sun, Karel Svoboda, Matteo Carandini, Kenneth D. Harris, Christof Koch, John O'Keefe, and Timothy D. Harris. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551(7679): 232–236, November 2017. ISSN 1476-4687. doi: 10.1038/nature24636. URL https://www.nature.com/articles/nature24636. Publisher: Nature Publishing Group.

[208] Erich D. Jarvis. Evolution of vocal learning and spoken language. *Science*, 366 (6461):50–54, 2019. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aax0287. URL https://www.science.org/doi/10.1126/science.aax0287.

[209] Pablo Tostado-Marcos, Ezequiel M. Arneodo, Lauren Ostrowski, Daril E. Brown II au2, Xavier A. Perez, Adam Kadwory, Lauren L. Stanwicks, Abdullah Alothman, Timothy Q. Gentner, and Vikash Gilja. Neural population dynamics in songbird RA and HVC during learned motor-vocal behavior. 2024. URL https://arxiv.org/abs/2407.06244.

[210] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression, 2022. URL https://arxiv.org/abs/2210.13438.

[211] Joseph Costello, Hisham Temmar, Luis Cubillos, Matthew Mender, Dylan Wallace, Matt Willsey, Parag Patil, and Cynthia Chestek. Balancing memorization and generalization in rnns for high performance brain-machine interfaces. *Advances in Neural Information Processing Systems*, 36, 2024. URL https://www.biorxiv.org/content/10.1101/2023.05.28.542435v1.

[212] Beata Jarosiewicz, Nicolas Y Masse, Daniel Bacher, Sydney S Cash, Emad Eskandar, Gerhard Friehs, John P Donoghue, and Leigh R Hochberg. Advantages of closed-loop calibration in intracortical brain–computer interfaces for people with tetraplegia. *Journal of neural engineering*, 10(4):046012, 2013. URL https://iopscience.iop.org/article/10.1088/1741-2560/10/4/046012.

[213] Steven M Chase, Andrew B Schwartz, and Robert E Kass. Bias, optimal linear

estimation, and the differences between open-loop simulation and closed-loop performance of spiking-based brain–computer interface algorithms. *Neural networks*, 22(9):1203–1213, 2009. URL https://www.sciencedirect.com/science/article/abs/pii/S0893608009000951?via%3Dihub.

[214] Shinsuke Koyama, Steven M Chase, Andrew S Whitford, Meel Velliste, Andrew B Schwartz, and Robert E Kass. Comparison of brain–computer interface decoding algorithms in open-loop and closed-loop control. *Journal of computational neuroscience*, 29:73–87, 2010. URL https://link.springer.com/article/10.1007/s10827-009-0196-9.

[215] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2021. URL https://arxiv.org/abs/2004.07219.

[216] Eran Klein, Tim Brown, Matthew Sample, Anjali R. Truitt, and Sara Goering. Engineering the Brain: Ethical Issues and the Introduction of Neural Devices. *Hastings Cent Rep*, 45(6):26–35, 2015. ISSN 0093-0334. doi: 10.1002/hast.515. URL https://onlinelibrary.wiley.com/doi/10.1002/hast.515.

[217] Anne E. Urai, Brent Doiron, Andrew M. Leifer, and Anne K. Churchland. Large-scale neural recordings call for new insights to link brain and behavior. *Nature Neuroscience*, 25:11–19, 2022. doi: 10.1038/s41593-021-00980-9.

[218] Ian H. Stevenson. Tracking advances in neural recording. Statistical Neuroscience Lab, University of Connecticut, 2023. URL https://stevenson.lab.uconn.edu/scaling/. Accessed September 6, 2024.

[219] Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.

[220] Eric Y Wang, Paul G Fahey, Kayla Ponder, Zhuokun Ding, Andersen Chang, Taliah Muhammad, Saumil Patel, Zhiwei Ding, Dat Tran, Jiakun Fu, et al. Towards a foundation model of the mouse visual cortex. *bioRxiv*, 2023.

[221] Jonas Geiping and Tom Goldstein. Cramming: Training a language model on a single gpu in one day, 2022. URL https://arxiv.org/abs/2212.14034.

[222] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent, 2022. URL https://arxiv.org/abs/2205.06175.

[223] Ingmar Schubert, Jingwei Zhang, Jake Bruce, Sarah Bechtle, Emilio Parisotto, Martin Riedmiller, Jost Tobias Springenberg, Arunkumar Byravan, Leonard

Hasenclever, and Nicolas Heess. A generalist dynamics model for control, 2023. URL https://arxiv.org/abs/2305.10912.

[224] Team Chameleon. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

[225] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling, 2021. URL https://arxiv.org/abs/2106.01345.

[226] Kuang-Huei Lee, Ofir Nachum, Mengjiao Yang, Lisa Lee, Daniel Freeman, Winnie Xu, Sergio Guadarrama, Ian Fischer, Eric Jang, Henryk Michalewski, and Igor Mordatch. Multi-game decision transformers, 2022. URL https://arxiv.org/abs/2205.15241.

[227] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

[228] Yizi Zhang, Yanchen Wang, Donato Jimenez-Beneto, Zixuan Wang, Mehdi Azabou, Blake Richards, Olivier Winter, The International Brain Laboratory, Eva Dyer, Liam Paninski, et al. Towards a" universal translator" for neural dynamics at single-cell, single-spike resolution. *arXiv preprint arXiv:2407.14668*, 2024.

[229] Armen Aghajanyan, Lili Yu, Alexis Conneau, Wei-Ning Hsu, Karen Hambardzumyan, Susan Zhang, Stephen Roller, Naman Goyal, Omer Levy, and Luke Zettlemoyer. Scaling laws for generative mixed-modal language models. In *International Conference on Machine Learning*, pages 265–279. PMLR, 2023.

[230] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer, 2024. URL https://arxiv.org/abs/2407.07726.

[231] Mark M Churchland, John P Cunningham, Matthew T Kaufman, Justin D Foster, Paul Nuyujukian, Stephen I Ryu, and Krishna V Shenoy. Neural population dynamics during reaching. *Nature*, 487(7405):51–56, 2012.

[232] Ann-Kathrin Kiessner, Robin T Schirrmeister, Joschka Boedecker, and Tonio Ball. Reaching the ceiling? empirical scaling behaviour for deep eeg pathology classification. *Computers in Biology and Medicine*, 178:108681, 2024.

[233] Nicholas A Steinmetz, Peter Zatka-Haas, Matteo Carandini, and Kenneth D Harris. Distributed coding of choice, action and engagement across the mouse brain. *Nature*, 576(7786):266–273, 2019.

[234] Eran Klein, Tim Brown, Matthew Sample, Anjali R Truitt, and Sara Goering. Engineering the brain: ethical issues and the introduction of neural devices. *Hastings Center Report*, 45(6):26–35, 2015.

[235] Samira Abnar, Mostafa Dehghani, Behnam Neyshabur, and Hanie Sedghi. Exploring the limits of large scale pre-training, 2021. URL https://arxiv.org/abs/2110.02095.

[236] Ekaterina Govorkova, Ema Puljak, Thea Aarrestad, Thomas James, Vladimir Loncar, Maurizio Pierini, Adrian Alan Pol, Nicolò Ghielmetti, Maksymilian Graczyk, Sioni Summers, Jennifer Ngadiuba, Thong Q. Nguyen, Javier Duarte, and Zhenbin Wu. Autoencoders on field-programmable gate arrays for real-time, unsupervised new physics detection at 40 MHz at the large hadron collider. *Nature Machine Intelligence*, 4(2):154–161, 2022. doi: 10.1038/s42256-022-00441-3.

[237] Jérôme Lecoq, Michael Oliver, Joshua H. Siegle, Natalia Orlova, Peter Ledochowitsch, and Christof Koch. Removing independent noise in systems neuroscience data using DeepInterpolation. *Nature Methods*, 18(11):1401–1408, 2021. doi: 10.1038/s41592-021-01285-2.

[238] Huizhuo Yuan, Yifeng Liu, Shuang Wu, Zhou Xun, and Quanquan Gu. MARS: Unleashing the power of variance reduction for training large models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 73553–73587. PMLR, 2025. URL https://proceedings.mlr.press/v267/yuan25f.html.

[239] P. G. Musial, S. N. Baker, G. L. Gerstein, E. A. King, and J. G. Keating. Signal-to-noise ratio improvement in multiple electrode recording. *Journal of Neuroscience Methods*, 115(1):29–43, 2002. ISSN 0165-0270. doi: 10.1016/S0165-0270(01)00516-7. URL https://www.sciencedirect.com/science/article/pii/S0165027001005167.

[240] Feng Wang, Xing Chen, and Pieter R. Roelfsema. Comparison of electrical microstimulation artifact removal methods for high-channel-count prostheses. *Journal of Neuroscience Methods*, 408:110169, 2024. ISSN 0165-0270. doi: 10.1016/j.jneumeth.2024.110169. URL https://www.sciencedirect.com/science/article/pii/S0165027024001146.

[241] H. Levitt. Transformed up-down methods in psychoacoustics. *The Journal of the Acoustical Society of America*, 49(2B):467–477, 1971. doi: 10.1121/1.1912375.

[242] Jeffrey M. Weiss, Sharlene N. Flesher, Robert Franklin, Jennifer L. Collinger, and Robert A. Gaunt. Artifact-free recordings in human bidirectional brain–computer interfaces. *Journal of Neural Engineering*, 16(1):016002, 2019. doi: 10.1088/1741-2552/aae748.

[243] Tianshu Zhang, Kiaran K. K. Lawson, Wee-Lih Lee, Matthew A. Petoe, Ashton Moorhead, Kristian Bulluss, Wesley Thevathasan, Hugh J. McDermott, and Thushara Perera. Stimulation artefact removal: review and evaluation of applications in evoked responses. *Journal of Neural Engineering*, 21(6):066029, 2024. doi: 10.1088/1741-2552/ad9959.

[244] G. W. Harding. A method for eliminating the stimulus artifact from digital recordings of the direct cortical response. *Computers and Biomedical Research*, 24(2):183–195, 1991. doi: 10.1016/0010-4809(91)90029-V.

[245] Daniel A. Wagenaar and Steve M. Potter. Real-time multi-channel stimulus artifact suppression by local curve fitting. *Journal of Neuroscience Methods*, 120 (2):113–120, 2002. doi: 10.1016/S0165-0270(02)00149-8.

[246] Francesco Negri, David J. Guggenmos, and Federico Barban. logLIRA: a method for reliable suppression of electrical stimulation artifacts enabling short-latency neural response recovery. *Journal of Neural Engineering*, 23(3):036008, 2026. doi: 10.1088/1741-2552/ae5f4b.

[247] Joseph T. Sombeck, Juliet Heye, Karthik Kumaravelu, Stefan M. Goetz, Angel V. Peterchev, Warren M. Grill, Sliman Bensmaia, and Lee E. Miller. Characterizing the short-latency evoked response to intracortical microstimulation across a multi-electrode array. *Journal of Neural Engineering*, 19(2):026044, 2022. doi: 10.1088/1741-2552/ac63e8.

[248] Izhar Bar-Gad, Shlomo Elias, Eilon Vaadia, and Hagai Bergman. Complex locking rather than complete cessation of neuronal activity in the globus pallidus of a 1-methyl-4-phenyl-1,2,3,6-tetrahydropyridine-treated primate in response to pallidal microstimulation. *Journal of Neuroscience*, 24(33):7410–7419, 2004. doi: 10.1523/JNEUROSCI.1691-04.2004.

[249] P. Christiaan Klink, Bruno Dagnino, Marie-Alice Gariel-Mathis, and Pieter R. Roelfsema. Distinct feedforward and feedback effects of microstimulation in visual cortex reveal neural mechanisms of texture segregation. *Neuron*, 95(1): 209–220.e3, 2017. doi: 10.1016/j.neuron.2017.05.033.

[250] Evan M. Dastin-van Rijn, Nicole R. Provenza, Jonathan S. Calvert, Ro'ee Gilron, Anusha B. Allawala, Radu Darie, Sohail Syed, Evan Matteson, Gregory S. Vogt, Michelle Avendano-Ortega, Ana C. Vasquez, Nithya Ramakrishnan, Denise N. Oswalt, Kelly R. Bijanki, et al. Uncovering biomarkers during therapeutic neuromodulation with PARRM: Period-based artifact reconstruction and removal method. *Cell Reports Methods*, 1(2):100010, 2021. doi: 10.1016/j.crmeth.2021.100010.

[251] Daniel J. O'Shea and Krishna V. Shenoy. Eraasr: an algorithm for removing electrical stimulation artifacts from multielectrode array recordings. *Journal of Neural Engineering*, 15(2):026020, 2018. doi: 10.1088/1741-2552/aaa365.

[252] Gonzalo E Mena, Lauren E Grosberg, Sasidhar Madugula, Paweł Hottowy,

Alan Litke, John Cunningham, E J Chichilnisky, and Liam Paninski. Electrical stimulus artifact cancellation and neural spike detection on large multi-electrode arrays. *PLoS Computational Biology*, 13(11):e1005842, 2017. doi: 10.1371/journal.pcbi.1005842.

[253] D. J. Caldwell, J. A. Cronin, R. P. N. Rao, K. L. Collins, K. E. Weaver, A. L. Ko, J. G. Ojemann, J. N. Kutz, and B. W. Brunton. Signal recovery from stimulation artifacts in intracranial recordings with dictionary learning. *Journal of Neural Engineering*, 17(2):026023, 2020. doi: 10.1088/1741-2552/ab7a4f.

[254] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://arxiv.org/abs/1806.07572.

[255] Pedro Domingos. Every model learned by gradient descent is approximately a kernel machine, 2020. URL https://arxiv.org/abs/2012.00152.

[256] Taylor Chomiak and Bin Hu. Axonal and somatic filtering of antidromically evoked cortical excitation by simulated deep brain stimulation in rat brain. *The Journal of Physiology*, 579(2):403–412, 2007. doi: 10.1113/jphysiol.2006.124057.

[257] Kenneth H. Britten, Michael N. Shadlen, William T. Newsome, and J. Anthony Movshon. The analysis of visual motion: a comparison of neuronal and psychophysical performance. *The Journal of Neuroscience*, 12(12):4745–4765, 1992. doi: 10.1523/JNEUROSCI.12-12-04745.1992.

[258] Mark H. Histed, Vincent Bonin, and R. Clay Reid. Direct activation of sparse, distributed populations of cortical neurons by electrical microstimulation. *Neuron*, 63(4):508–522, 2009. doi: 10.1016/j.neuron.2009.07.016.

[259] Yu-Cheng Pei, Steven S. Hsiao, James C. Craig, and Sliman J. Bensmaia. Shape invariant coding of motion direction in somatosensory cortex. *PLoS Biology*, 8 (2):e1000305, 2010. doi: 10.1371/journal.pbio.1000305.

[260] John Choi, Austin J. Brockmeier, David McNiel, Lee M. von Kraus, José C. Príncipe, and Joseph T. Francis. Eliciting naturalistic cortical responses with a sensory prosthesis via optimized microstimulation. *Journal of Neural Engineering*, 13(5):056007, 2016. doi: 10.1088/1741-2560/13/5/056007.

[261] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019. URL https://arxiv.org/abs/1804.07461.

[262] Joshua H Siegle, Xiaoxuan Jia, Séverine Durand, Sam Gale, Corbett Bennett, Nile Graddis, Greggory Heller, Tamina K Ramirez, Hannah Choi, Jennifer A Luviano, et al. Survey of spiking in the mouse visual system reveals functional hierarchy. *Nature*, 592(7852):86–92, 2021. doi: 10.1038/s41586-020-03171-x.

[263] International Brain Laboratory. A brain-wide map of neural activity dur-

ing complex behaviour. *Nature*, 645(8079):177–191, 2025. doi: 10.1038/s41586-025-09235-0.

[264] MICrONS Consortium. Functional connectomics spanning multiple areas of mouse visual cortex. *Nature*, 640(8058):435–447, 2025. doi: 10.1038/s41586-025-08790-w.

[265] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. doi: 10.48550/arXiv.1910.10683.

[266] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020. doi: 10.48550/arXiv.2101.00027.

[267] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The RefinedWeb dataset for Falcon LLM: Outperforming curated corpora with web data, and web data only. In *Advances in Neural Information Processing Systems*, 2023. doi: 10.48550/arXiv.2306.01116.

[268] Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. doi: 10.48550/arXiv.2406.17557.

[269] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. DataComp-LM: In search of the next generation of training sets for language models. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. doi: 10.48550/arXiv.2406.11794.

[270] Benjamin Lahner, Mayukh Deb, N Apurva Ratan Murty, and Aude Oliva. MOSAIC: A scalable framework for fMRI dataset aggregation and modeling of human vision. *bioRxiv*, 2025. doi: 10.64898/2025.11.28.690060.

[271] Dingkun Liu, Yuheng Chen, Zhu Chen, Zhenyao Cui, Yaozhi Wen, Jiayu An, Jingwei Luo, and Dongrui Wu. EEG-FM-Compass: Progress, benchmarking, and future directions for EEG foundation models. *arXiv preprint arXiv:2601.17883*, 2026. doi: 10.48550/arXiv.2601.17883.

[272] Francis R Willett, Jingyuan Li, Trung Le, Chaofei Fan, Mingfei Chen, Eli Shlizerman, Yue Chen, Xin Zheng, Tatsuo S Okubo, Tyler Benster, Hyun Dong Lee, Maxwell Kounga, E Kelly Buchanan, David Zoltowski, Scott W Linderman, and Jaimie M Henderson. Brain-to-text benchmark '24: Lessons learned. *arXiv*

preprint arXiv:2412.17227, 2024. doi: 10.48550/arXiv.2412.17227.

[273] Neuroprosthetics Lab, UC Davis. Brain-to-text '25 competition. Kaggle, 2025. URL https://www.kaggle.com/competitions/brain-to-text-25. Ran 2025-07-07 to 2026-01-01; 466 participants.

[274] Mehdi Azabou, Krystal Xuejing Pan, Vinam Arora, Ian Jarratt Knight, Eva L Dyer, and Blake A Richards. Multi-session, multi-task neural decoding from distinct cell-types and brain regions. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=IuU0wcO0mo.

[275] Yizi Zhang, Yanchen Wang, Mehdi Azabou, Alexandre Andre, Zixuan Wang, Hanrui Lyu, International Brain Laboratory, Eva L Dyer, Liam Paninski, and Cole L Hurwitz. Neural encoding and decoding at scale. *arXiv preprint arXiv:2504.08201*, 2025. doi: 10.48550/arXiv.2504.08201.

[276] Eric Y Wang, Paul G Fahey, Zhuokun Ding, Stelios Papadopoulos, Kayla Ponder, Marissa A Weis, Andersen Chang, Taliah Muhammad, Saumil Patel, Zhiwei Ding, Dat Tran, Jiakun Fu, Katrin Franke, Alexander S Ecker, Jacob Reimer, Xaq Pitkow, Fabian H Sinz, and Andreas S Tolias. Foundation model of neural activity predicts response to new stimulus types. *Nature*, 640(8058):470–477, 2025. doi: 10.1038/s41586-025-08829-y.

[277] Trung Le, Hao Fang, Jingyuan Li, Tung Nguyen, Lu Mi, Amy Orsborn, Uygar Sümbül, and Eli Shlizerman. SPINT: Spatial permutation-invariant neural transformer for consistent intracortical motor decoding. In *arXiv preprint arXiv:2507.08402*, 2025. doi: 10.48550/arXiv.2507.08402.

[278] Vinam Arora, Divyansha Lachi, Ian J Knight, Mehdi Azabou, Blake Richards, Cole L Hurwitz, Josh Siegle, and Eva L Dyer. Know thyself by knowing others: Learning neuron identity from population context. In *Advances in Neural Information Processing Systems*, 2025. doi: 10.48550/arXiv.2512.01199.

[279] Muquan Yu, Mu Nan, Hossein Adeli, Jacob S Prince, John A Pyles, Leila Wehbe, Margaret M Henderson, Michael J Tarr, and Andrew F Luo. Meta-learning an in-context transformer model of human higher visual cortex. In *Advances in Neural Information Processing Systems*, 2025. doi: 10.48550/arXiv.2505.15813.

[280] Mu Nan, Muquan Yu, Weijian Mai, Jacob S Prince, Hossein Adeli, Rui Zhang, Jiahang Cao, Benjamin Becker, John A Pyles, Margaret M Henderson, Chunfeng Song, Nikolaus Kriegeskorte, Michael J Tarr, Xiaoqing Hu, and Andrew F Luo. Meta-learning in-context enables training-free cross subject brain decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. doi: 10.48550/arXiv.2604.08537.

[281] Francis Willett, Erin Kunz, Chaofei Fan, Donald Avansino, Guy Wilson, Eun Young Choi, Foram Kamdar, Leigh R. Hochberg, Shaul Druckmann, Krishna V. Shenoy, and Jaimie M. Henderson. A high-performance speech

neuroprosthesis. Jan 2023. doi: 10.1101/2023.01.21.524489. URL https://www.biorxiv.org/content/10.1101/2023.01.21.524489v1.

[282] Amy L. Orsborn, Helene G. Moorman, Simon A. Overduin, Maryam M. Shanechi, Dragan F. Dimitrov, and Jose M. Carmena. Closed-loop decoder adaptation shapes neural plasticity for skillful neuroprosthetic control. *Neuron*, 82(6): 1380–1393, June 2014. ISSN 1097-4199. doi: 10.1016/j.neuron.2014.04.048.

[283] Francis R Willett, Brian A Murphy, Daniel Young, William D Memberg, Christine H Blabe, Chethan Pandarinath, Brian Franco, Jad Saab, Benjamin L Walter, Jennifer A Sweet, et al. A comparison of intention estimation methods for decoder calibration in intracortical brain–computer interfaces. *IEEE Transactions on Biomedical Engineering*, 65(9):2066–2078, 2017.

[284] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[285] Burcu Küçükoğlu, Leili Soo, David Leeftink, Fabrizio Grani, Cristina Soto Sanchez, Umut Güçlü, Marcel van Gerven, and Eduardo Fernandez. Bayesian optimization of cortical neuroprosthetic vision using perceptual feedback. *Journal of Neural Engineering*, 22(4), 2025. doi: 10.1088/1741-2552/adeae9.

[286] Leili Soo, Roberto Morolón Ruiz, Fabrizio Grani, María del Mar Ayuso Arroyave, Cristina Soto-Sánchez, and Eduardo Fernandez. Effect of mobility on phosphene detection thresholds in an intracortical visual prosthesis. *Investigative Ophthalmology & Visual Science*, 67(7):2468, 2026. ARVO Annual Meeting abstract.

[287] Aleksandr Kovalev, Antonio Lozano, Fabrizio Grani, Cristina Soto Sanchez, Leili Soo, Rocío López-Peco, Adrian Villamarin-Ortiz, Roberto Morolón Ruiz, María del Mar Ayuso Arroyave, Alfonso Rodil, and Eduardo Fernández. Masked autoencoders learn perception-relevant representations from resting state neural data. In *NeuroAI Workshop, AAAI*, 2026. doi: 10.48550/arXiv.2607.22615.

[288] Daniel L K Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624, 2014. doi: 10.1073/pnas.1403112111.

[289] Daniel L K Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, 19(3):356–365, 2016. doi: 10.1038/nn.4244.

[290] Leila Wehbe, Ashish Vaswani, Kevin Knight, and Tom Mitchell. Aligning context-based statistical models of language with brain activity during reading. In *Proceedings of the 2014 Conference on Empirical Methods in Natural*

Language Processing (EMNLP), pages 233–243, Doha, Qatar, 2014. URL https://aclanthology.org/D14-1030/.

[291] Alexander J E Kell, Daniel L K Yamins, Erica N Shook, Sam V Norman-Haignere, and Josh H McDermott. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3):630–644.e16, 2018. doi: 10.1016/j.neuron.2018.03.044.

[292] Peter Y Wang, Yi Sun, Richard Axel, L F Abbott, and Guangyu Robert Yang. Evolving the olfactory system with machine learning. *Neuron*, 109(23): 3879–3892.e5, 2021. doi: 10.1016/j.neuron.2021.09.010.

[293] Dan Schwartz, Mariya Toneva, and Leila Wehbe. Inducing brain-relevant bias in natural language processing models. In *Advances in Neural Information Processing Systems*, 2019. doi: 10.48550/arXiv.1911.03268.

[294] Omer Moussa and Mariya Toneva. Brain-tuning improves generalizability and efficiency of brain alignment in speech models. In *Advances in Neural Information Processing Systems*, 2025. doi: 10.48550/arXiv.2510.21520.

[295] Santiago A Cadena, George H Denfield, Edgar Y Walker, Leon A Gatys, Andreas S Tolias, Matthias Bethge, and Alexander S Ecker. Deep convolutional models improve predictions of macaque V1 responses to natural images. *PLoS Computational Biology*, 15(4):e1006897, 2019. doi: 10.1371/journal.pcbi.1006897.

[296] Omer Moussa and Mariya Toneva. RABBiT: Rapidly adaptive BOLD foundation model via brain-tuning for accurate zero-shot and few-shot prediction of speech-elicited responses in the brain. *arXiv preprint arXiv:2607.05171*, 2026. doi: 10.48550/arXiv.2607.05171.

[297] Trinity Chung, Yuchen Shen, Nathan C L Kong, and Aran Nayebi. Task-optimized convolutional recurrent networks align with tactile processing in the rodent brain. In *Advances in Neural Information Processing Systems*, 2025. doi: 10.48550/arXiv.2505.18361.

[298] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners, 2022.

[299] Michael C Park, Abderraouf Belhaj-Saïf, and Paul D Cheney. Chronic recording of EMG activity from large numbers of forelimb muscles in awake macaque monkeys. *Journal of Neuroscience Methods*, 96, 2000. URL https://www.sciencedirect.com/science/article/pii/S0165027000001552.

[300] Tim Sainburg, Marvin Thielk, and Timothy Q Gentner. Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires. *PLoS computational biology*, 16(10):e1008228, 2020. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1008228.

[301] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Noth-

man, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *JMLR*, 12(85):2825–2830, June 2018. URL http://arxiv.org/abs/1201.0490. arXiv: 1201.0490.

[302] John P. Cunningham, Paul Nuyujukian, Vikash Gilja, Cindy A. Chestek, Stephen I. Ryu, and Krishna V. Shenoy. A closed-loop human simulator for investigating the role of feedback control in brain-machine interfaces. *Journal of Neurophysiology*, 105(4):1932–1949, April 2011. ISSN 0022-3077. doi: 10.1152/jn.00503.2010. URL https://journals.physiology.org/doi/full/10.1152/jn.00503.2010. Publisher: American Physiological Society.

[303] Jonathan C. Kao, Paul Nuyujukian, Stephen I. Ryu, Mark M. Churchland, John P. Cunningham, and Krishna V. Shenoy. Single-trial dynamics of motor cortex and their applications to brain-machine interfaces. *Nat Commun*, 6 (1):7759, July 2015. ISSN 2041-1723. doi: 10.1038/ncomms8759. URL https://www.nature.com/articles/ncomms8759. Number: 1 Publisher: Nature Publishing Group.

[304] David Sussillo, Rafal Jozefowicz, L. F. Abbott, and Chethan Pandarinath. LFADS - Latent Factor Analysis via Dynamical Systems. *arXiv:1608.06315 [cs, q-bio, stat]*, August 2016. URL http://arxiv.org/abs/1608.06315. arXiv: 1608.06315.

[305] Mohammad Reza Keshtkaran and Chethan Pandarinath. Enabling hyperparameter optimization in sequential autoencoders for spiking neural data. *Advances in Neural Information Processing Systems*, August 2019. URL http://arxiv.org/abs/1908.07896. arXiv: 1908.07896.

[306] Mohammad Reza Keshtkaran, Andrew R. Sedler, Raeed H. Chowdhury, Raghav Tandon, Diya Basrai, Sarah L. Nguyen, Hansem Sohn, Mehrdad Jazayeri, Lee E. Miller, and Chethan Pandarinath. A large-scale neural network training framework for generalized estimation of single-trial population dynamics. *Nat Methods*, 19(12):1572–1577, December 2022. ISSN 1548-7105. doi: 10.1038/s41592-022-01675-0. URL https://www.nature.com/articles/s41592-022-01675-0. Publisher: Nature Publishing Group.

[307] Chaoqi Yang, M. Brandon Westover, and Jimeng Sun. Biot: Cross-data biosignal learning in the wild, 2023. URL https://arxiv.org/abs/2305.10351.

[308] Rahul Thapa, Bryan He, Magnus Ruud Kjaer, Hyatt Moore, Gauri Ganjoo, Emmanuel Mignot, and James Zou. Sleepfm: Multi-modal representation learning for sleep across brain activity, ecg and respiratory signals, 2024. URL https://arxiv.org/abs/2405.17766.

[309] Quilee Simeon, Leandro Venâncio, Michael A Skuhersky, Aran Nayebi, Edward S Boyden, and Guangyu Robert Yang. Scaling properties for artificial neural

network models of a small nervous system. In *SoutheastCon 2024*, pages 516–524. IEEE, 2024.

[310] Motoshige Sato, Kenichi Tomeoka, Ilya Horiguchi, Kai Arulkumaran, Ryota Kanai, and Shuntaro Sasai. Scaling law in neural data: Non-invasive speech decoding with 175 hours of eeg data, 2024. URL https://arxiv.org/abs/2407.07595.

[311] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19. ACM, January 2019. doi: 10.1145/3287560.3287596. URL http://dx.doi.org/10.1145/3287560.3287596.

[312] Edward H Adelson, James R Bergen, et al. *The plenoptic function and the elements of early vision*, volume 2. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of ..., 1991.

[313] Gregor Bachmann and Vaishnavh Nagarajan. The pitfalls of next-token prediction, 2024. URL https://arxiv.org/abs/2403.06963.

[314] Colin Lea, Rene Vidal, Austin Reiter, and Gregory D. Hager. Temporal convolutional networks: A unified approach to action segmentation, 2016. URL https://arxiv.org/abs/1608.08242.

[315] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting, 2024. URL https://arxiv.org/abs/2310.10688.

[316] Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. What language model architecture and pretraining objective work best for zero-shot generalization?, 2022. URL https://arxiv.org/abs/2204.05832.

[317] Nur Muhammad Mahi Shafiullah, Zichen Jeff Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone, 2022. URL https://arxiv.org/abs/2206.11251.

[318] Jesse Farebrother, Jordi Orbay, Quan Vuong, Adrien Ali Taïga, Yevgen Chebotar, Ted Xiao, Alex Irpan, Sergey Levine, Pablo Samuel Castro, Aleksandra Faust, Aviral Kumar, and Rishabh Agarwal. Stop regressing: Training value functions via classification for scalable deep rl, 2024. URL https://arxiv.org/abs/2403.03950.

[319] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021. URL https://arxiv.org/abs/2104.14294.

[320] Chen Yang, Junzhuo Li, Xinyao Niu, Xinrun Du, Songyang Gao, Haoran Zhang, Zhaoliang Chen, Xingwei Qu, Ruibin Yuan, Yizhi Li, Jiaheng Liu,

Stephen W. Huang, Shawn Yue, and Ge Zhang. The fine line: Navigating large language model pretraining with down-streaming capability analysis, 2024. URL https://arxiv.org/abs/2404.01204.

[321] Sachin Goyal, Ananya Kumar, Sankalp Garg, Zico Kolter, and Aditi Raghunathan. Finetune like you pretrain: Improved finetuning of zero-shot vision models, 2022. URL https://arxiv.org/abs/2212.00638.

[322] Linxing Preston Jiang, Shirui Chen, Emmanuel Tanumihardja, Xiaochuang Han, Weijia Shi, Eric Shea-Brown, and Rajesh P. N. Rao. Data heterogeneity limits the scaling effect of pretraining neural data transformers. *bioRxiv*, 2025. doi: 10.1101/2025.05.12.653551. URL https://www.biorxiv.org/content/early/2025/05/15/2025.05.12.653551.

[323] Robert D Flint, Eric W Lindberg, Luke R Jordan, Lee E Miller, and Marc W Slutzky. Accurate decoding of reaching movements from field potentials in the absence of spikes. *Journal of neural engineering*, 9(4):046006, 2012.

[324] Jonathan R Wolpaw, Niels Birbaumer, Dennis J McFarland, Gert Pfurtscheller, and Theresa M Vaughan. Brain–computer interfaces for communication and control. *Clinical neurophysiology*, 113(6):767–791, 2002.

[325] Mariana P. Branco, Lisanne M. De Boer, Nick F. Ramsey, and Mariska J. Vansteensel. Encoding of kinetic and kinematic movement parameters in the sensorimotor cortex: A brain-computer interface perspective. *European Journal of Neuroscience*, 50(5):2755–2772, September 2019. ISSN 0953-816X, 1460-9568. doi: 10.1111/ejn.14342. URL https://onlinelibrary.wiley.com/doi/10.1111/ejn.14342.

[326] Kristin M Quick, Jessica L Mischel, Patrick J Loughlin, and Aaron P Batista. The critical stability task: quantifying sensory-motor control during ongoing movement in nonhuman primates. *Journal of Neurophysiology*, 120(5):2164–2181, 2018.

[327] Kendra K. Noneman and J. Patrick Mayo. Decoding continuous tracking eye movements from cortical spiking activity. *International Journal of Neural Systems*, page S0129065724500709, October 2024. ISSN 0129-0657, 1793-6462. doi: 10.1142/S0129065724500709. URL https://www.worldscientific.com/doi/10.1142/S0129065724500709.

[328] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning, 2023. URL https://arxiv.org/abs/2307.08691.

[329] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin F. Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings,

Mark Patrick Collier, Alexey Gritsenko, Vighnesh Birodkar, Cristina Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetić, Dustin Tran, Thomas Kipf, Mario Lučić, Xiaohua Zhai, Daniel Keysers, Jeremiah Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters, 2023. URL https://arxiv.org/abs/2302.05442.

[330] Mitchell Wortsman, Peter J Liu, Lechao Xiao, Katie E Everett, Alexander A Alemi, Ben Adlam, John D Co-Reyes, Izzeddin Gur, Abhishek Kumar, Roman Novak, Jeffrey Pennington, Jascha Sohl-Dickstein, Kelvin Xu, Jaehoon Lee, Justin Gilmer, and Simon Kornblith. Small-scale proxies for large-scale transformer training instabilities. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=d8w0pmvXbZ.

[331] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories, 2021. URL https://arxiv.org/abs/2012.14913.