Learning Representations from Neural Population Dynamics: Addressing Neural Variability Across Scales

Trung Le

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2025

Reading Committee:
Eli Shlizerman, Chair
Amy Orsborn
Uygar Sümbül

Program Authorized to Offer Degree:
Electrical and Computer Engineering

University of Washington

Abstract

Learning Representations from Neural Population Dynamics:
Addressing Neural Variability Across Scales

Trung Le

Chair of the Supervisory Committee:
Eli Shlizerman
Electrical and Computer Engineering

Interactions between individual neurons, each characterized by distinct intrinsic physiological properties, collectively give rise to population responses underlying complex animal behaviors. These responses exhibit variable dynamics across trials, recording sessions, and behavioral contexts—arising from stochastic spiking at the trial level, electrode drift and neural plasticity across sessions, and task- or state-dependent modulation across behavioral contexts. This multiscale variability complicates the reliable extraction of scientific insights from population activity. Consequently, modeling and decoding from population activity necessitate methods capable of learning stable representations that capture the underlying structure of neuronal activity in the presence of neural variability caused by noise, partial observability, and domain shifts inherent in population recordings. In this dissertation, I present my studies that aim to extract useful information from population dynamics while addressing neural variability across different scales: single trials, recording sessions, and behavioral contexts. In the first study, I developed a spatiotemporal transformer to learn stable neural representations underlying stochastic firing activity of neural population on the single-trial basis. In the second study, I introduced a self-supervised framework for extracting time-invariant representations of individual neurons by modeling their dynamics across partially overlapping populations over multiple recording sessions. In the third study, I developed a lightweight adaptive framework

for online neural decoding, enabling rapid and robust generalization in unseen sessions with minimal unlabeled calibration trials and no model fine-tuning. In the fourth study, I exploited the dependence of population dynamics on behavioral contexts and presented a decoding framework leveraging context-aware representations for effective decoding of speech from population activity. Together, these studies advance a representation-centric paradigm for neural population analysis—delivering generalizable abstractions that are robust across contexts, scale to large recordings, and leverage inductive biases embedded in the population—thereby enabling effective extraction of scientific insights from population analysis and paving a way towards high-performing and robust brain–computer interfaces.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to the members of my supervisory committee for their guidance, time, and support throughout my PhD. I am especially grateful to my advisor Prof. Eli Shlizerman for taking a chance on me and for giving me the freedom to pursue my research interests. His lessons, advice and encouragement have shaped both my technical development and my growth as an independent researcher. I am deeply thankful to Prof. Amy Orsborn for being a source of inspiration since the day I took her Neural Computation and Engineering Lab class in my first year, and for her thoughtful guidance on navigating the academic trajectory. I would like to thank Prof. Uygar Sümbül for his tremendous support and for being a role model of technical expert. I will never forget our discussions that always give me in-depth perspectives on how to approach difficult problems. I am also grateful to my Graduate School Representative Prof. Vikram Iyer for taking the time to evaluate my work and for his support in ensuring that my exams proceeded smoothly.

This dissertation also would not have been possible without the collaborations and conversations with colleagues across departments and institutions. I am deeply grateful to members of the NeuroAI Lab, including Jingyuan Li, Hao Fang, and Mingfei Chen, for their close collaboration and shared commitment to rigorous and impactful research. I owe special thanks to my collaborators at the Allen Institute, especially Lu Mi, whose insights and dedication were indispensable in bringing our projects to fruition; to colleagues in the University of Washington (UW) Electrical and Computer Engineering Department, including Leo Scholl and Pavithra Rajeswaran; in the UW Computer Science and Engineering Department, including Tianxing He and Wuwei Zhang; and in the UW Applied Mathematics Department, including Ziyu Lu and Prof. Eric Shea-Brown. I am also grateful to colleagues

at the A3D3 Institute, including Rajeev Botadra, Ling-Chi Yang, Chi-Jui Chen, Prof. Scott Hauck, Prof. Shih-Chieh Hsu, and Prof. Bo-Cheng Lai. Finally, I thank Tung Nguyen (University of California, Los Angeles), Chaofei Fan (Stanford University), and Prof. Hao Wang (Rutgers University) for valuable collaborations and insights that enriched this dissertation.

I would also like to acknowledge other members of the NeuroAI Lab who contributed to an intellectually supportive and collegial environment, including Rahul Biswas, Zijun Cui, James Hazelden, Saba Heravi, Jimin Kim, Xiulong Liu, Kun Su, Ryan Vogt, Jinlin Xiang, and Yang Zheng. Beyond the science, their kindness, humor, and steady friendship carried me through the hardest stretches of graduate school. I will always cherish the camaraderie, the late-night problem-solving, the quiet acts of support that made my PhD years full of memories.

I gratefully acknowledge financial and technical support from the University of Washington, the Allen Institute, the A3D3 Institute, and the National Science Foundation (NSF), which made this research possible.

Finally, I extend my deepest thanks to my family - my parents, brothers and sister - for their unwavering love, sacrifice, and support throughout the years. A special thanks to my darling Han Pham for being by my side through all the ups and downs of this journey. Their encouragement and belief in me have been a constant source of strength, greater than I can ever fully express.

DEDICATION

To my parents, who sacrificed more than I will ever fully understand so that I could pursue opportunities far from home.

To my brothers and sister, for their unconditional support, pride, and quiet faith in me.

And to my darling Han Pham, whose love, patience, and steady presence have carried me through the hardest moments and made the joyful ones even brighter.

This work is as much theirs as it is mine.

Chapter 1
INTRODUCTION

1.1 Background

The brain is an engineering feat of nature, containing billions of neurons and trillions of synapses that collectively give rise to cognitive functions and behaviors. Understanding the principles by which these interconnected networks of neurons represent and transform information to generate behavior in response to environmental stimuli has been a central pursuit of neuroscience. Early single-unit recordings established the notions of *rate coding*, *receptive fields*, and *tuning*, suggesting that neurons encode specific stimulus or movement features via their firing patterns [Ref1, Ref2, Ref3]. As technology progressed from single microelectrodes to multi-electrode arrays and optical imaging, the scale and dimensionality of neural data expanded dramatically, enabling simultaneous observation of hundreds to tens of thousands of neurons across brain areas and over extended timescales [Ref4, Ref5, Ref6, Ref7].

This shift in recording capability catalyzed a conceptual transition: from single neurons to *neural populations*. Rather than focusing solely on analyzing the activity of individual neural units, contemporary computational neuroscience increasingly emphasizes *neural population analyses*. This idea posits that it is the populations of neurons, rather than the individual units, that are the essential units of computation in the brain [Ref8, Ref9, Ref10]. This perspective is supported by evidence from studies that found neurons across brain regions to have mixed selectivity in various neural processes such as memory and decision-making, rendering single-unit decoding of stimuli or behavior largely intractable [Ref11, Ref12, Ref13]. In addition, more mechanistic insights have been drawn from analyzing the *dynamics* of the neural population—the temporal evolution of population activity in high-dimensional state space spanned by the firing rates of individual member neurons. While the dynamics are

constituted by a large number of neurons, dimensionality reduction methods such as PCA and jPCA have revealed low-dimensional structures within the population dynamics, e.g. the rotational dynamics in motor cortex during reaches [Ref14], and latent-state models like GPFA and LDS variants have uncovered smooth latent manifolds underlying variable spiking activity [Ref15, Ref16]. Collectively, these studies demonstrated that neural populations are more than the sum of their parts: coordinated activity of many neurons implements sophisticated computations that can be probed with interpretable tools grounded in dynamical system theories.

As our ability to record from ensembles of neurons has grown exponentially, the demands for analytical tools capable of processing these ever-larger datasets have scaled accordingly. In response, much of the attention has turned to recent advances in deep learning (DL) —a field that has recently undergone its own revolution—for frameworks that are not only capable of learning sophisticated patterns hidden in the data but also capable of doing so in a scalable and label-efficient manner. Specifically, Recurrent Neural Networks (RNN) [Ref17, Ref18] and Transformers [Ref19] have become workhorses for complex sequence modeling and have contributed to a host of techniques to study and extract insights from population dynamics. RNN serves as an intuitive framework to model the time-varying activity of individual neurons as well as the neural population, since the conditional dependence of the current-time variables on the previous timesteps is baked into the sequential nature of the recurrent architecture. They have been shown to be an effective tool for modeling population dynamics, as well as for decoding limb movement and speech articulation in human and non-human primates [Ref20, Ref21, Ref22, Ref23, Ref24]. Transformers with the attention mechanism have lifted the restriction on RNN's sequential dependency, allowing flexible weighting (attention) over all observed entities (tokens – which in neural population analyses can take the form of either timesteps or neurons). Such innovation has brought about models with significantly improved expressiveness and an enhanced ability to ingest multi-modal and heterogeneous neural datasets [Ref25, Ref26, TL2, TL5, TL3]. Beyond high-performing architectures, the field of deep learning has also contributed efficient learning algorithms that enable learning

from the vast amount of noisy, unlabeled data. Self-supervised training strategies such as mask modeling, contrastive learning, and future forecasting have enabled models to find structures embedded in the data and extract robust representations that are useful for diverse downstream tasks [Ref27, Ref28, Ref29]. Taken together, deep learning offers computational frameworks that are simultaneously *expressive* and *scalable* and that, when paired with appropriate analytical tools, can yield *interpretable* insights to study neural population dynamics.

In parallel, translational research in brain–computer interfaces (BCIs) has leveraged neural population activity to decode intended actions and restore communication or motor function for individuals with paralysis. Demonstrations of such interfaces include robotic arms that allow an individual with tetraplegia to manipulate objects [Ref30]; interfaces that translate movement intents into continuous cursor control [Ref31]; and neuroprotheses that decode handwriting [Ref23] and attempted speech [Ref32, Ref24] from neural population activity. Yet decades of BCI experience have also highlighted a critical challenge in deriving insights from neural population activity: *neural variability*. On the individual neuron level, activity of individual neurons is variable, changing from trial to trial even when they are subject to identical stimuli or recorded under repeated behavioral conditions. This variability may come from the stochastic release/reception of neurotransmitters at the synapses, circuit noise from the dynamics of local excitatory and inhibitory populations, whole-brain noise arising from the interactions between brain regions, or behavioral contexts including task-irrelevant physiological states [Ref33]. On the network level, synaptic weights between neurons are weakened or reinforced over the course of learning a new task. Even after the animal has learned a task to proficiency, tuning profile of tracked neurons may drift over weeks of recordings [Ref34]. Besides these intrinsic physiological sources, variability from neural recordings may also come from the technical complications in the recording methods. Membership of neurons in the population might be inconsistent across recordings due to displacements of electrodes over time, where neurons may appear or disappear on a given channel due to their shifted distance to the electrode [Ref35]. The problem is exacerbated if

spikes are detected using threshold-crossing rather than being sorted, the former of which might result in channels containing mixed responses of multiple neurons [Ref36].

In the presence of neural variability, computational methods attempting to extract useful information from the population recordings may suffer from performance degradation if not carefully designed. A promising approach to address this problem is representation learning, in which deep learning models learn from data to construct representations that not only enhance task performance but also remain robust to the variability within neural recordings. In this dissertation, I will present four of my previous works that aim to learn robust neural representations for dynamics modeling and behavior decoding in the presence of neural variability. These works tackle the neural variability at different levels: *single trials*, *recording sessions*, and *behavioral contexts*.

1.2 Summary of Contributions

This dissertation advances computational techniques for learning representations from neural population activity that explicitly address variability within neurons across trials and within populations across recording sessions and behavioral contexts. Specifically, the contributions are as follows:

1. *SpatioTemporal Neural Data Transformer (STNDT)* [TL1], which addresses the neural variability at the *single-trial level*. STNDT is a transformer architecture which explicitly learns both the spatial coordination between neurons and the temporal progression of the population activity to uncover their underlying firing rates. The model learns robust representations of neural population activity from noisy spiking dynamics on a single-trial basis, enabling effective inference of unobserved neurons and timesteps, as well as facilitating behavior decoding from the spatiotemporal representations.

2. *Neuronal Permutation- and Time-Invariant Representations (NeuPRINT)* [TL2], which addresses the neural variability at the *session level*. NeuPRINT is a self-supervised approach to infer neuronal identity from population recordings spanning multiple

sessions. The model learns robust representations for individual neurons despite the time-varying dynamics of partially overlapping populations observed across sessions. The learned representations can then be used to decode cell transcriptomic types by a lightweight classifier.

3. *Spatial Permutation-Invariant Neural Transformer (SPINT)* [TL3], which also addresses the neural variability at the *session level*. Unlike NeuPRINT and other existing decoding approaches where the decoders need to be trained and evaluated on the same set of neurons or be fine-tuned to adapt the new set of neurons, SPINT proposed a novel adaptation method that enables generalization on novel sessions with variable size and ordering, using only a few unlabeled calibration trials and requiring no model fine-tuning. This is achieved by learning universal, behavior-relevant representations of individual neurons which facilitate generalization despite the non-stationarities in long-term recordings.

4. *Divide-and-Conquer Neural Decoder (DCoND)* [TL4], which tackles the neural variability at the level of *behavioral context*. DCoND is a brain-to-text decoding framework leveraging context-aware acoustic representations (diphones) and large language models to decode multi-session neural activity into texts. The model recognizes the context-dependence nature of neural activity during attempted speech, i.e., neural activity representing a phoneme depends not only on the phoneme being spoken but also on the context of surrounding phonemes, subsequently learns to construct fine-grained representations reflecting the context of speech (behavioral context).

The dissertation contains materials from my published works [TL1, TL2, TL3, TL4]. Together, these contributions advance a representation-learning paradigm centered on *variability across scales*. They show that population-aware, invariant, and interpretable representations are both scientifically informative and practically robust.

These studies have motivated my other concurrent works, including works where I co-

developed an interpretable transformer [TL5] to infer the nonstationary connectivity between neurons, an adaptive graph neural network [TL6] that modulates time-varying neuronal interactions to predict future neural activity, and portable, low-latency systems for BCI inference [TL7, TL8].

1.3 Dissertation Outline

The remainder of the dissertation is organized as follows:

Chapter 2 surveys the fundamental concepts covered in this dissertation, spanning neural population analyses, latent structures, and neural decoding in the presence of variability. It highlights major intracortical BCI advances and introduces modern machine learning concepts that motivate the proposed methods.

Chapter 3 introduces a spatiotemporal transformer for modeling neural population activity by learning factorized spatial and temporal structures. It demonstrates how this factorized modeling helps infer the single-trial population dynamics spanning various brain regions and behavioral tasks.

Chapter 4 presents a self-supervised framework for learning *time-invariant* neuronal identity from population activity. It demonstrates the correspondence between the learned representations and cell transcriptomic types that remains stable across recording sessions.

Chapter 5 proposes a *permutation-invariant transformer* for robust cross-session decoding. Built-in architectural invariances and training strategies emulate changes in population composition, yielding stable performance under session-to-session variability.

Chapter 6 presents a decoding framework that uses intermediate, *context-aware* acoustic units (diphones) to capture the fine-grained dynamics of speech. It demonstrates how leveraging these context-dependent representations in combination with large language models can enhance brain-to-text decoding.

Chapter 7 summarizes the key findings from this dissertation, discusses the limitations of the presented studies, and reflects on the broader implications for systems neuroscience and the development of robust brain-computer interfaces.

Chapter 2

FUNDAMENTAL CONCEPTS AND RELATED WORK

2.1 Foundations of Neural Population Analyses

Understanding how the brain encodes and processes information lies at the core of computational neuroscience. While early work on single neurons revealed fundamental physiological and computational properties of these units, contemporary approaches increasingly focus on neural ensembles to obtain a more comprehensive understanding of brain function and its relationship to behavior. Population activity is often well described as evolving on low-dimensional manifolds whose latent dynamics capture computations underlying motor control, perception, and cognition. The rich theoretical and empirical insights gained from this population-level perspective have, in turn, established neural populations as the fundamental unit of computation in the brain.

2.1.1 Neural Signals and Recording Modalities

Neural activity can be measured through various recording modalities, each having its own advantages and disadvantages in terms of invasiveness, coverage, signal-to-noise ratio (SNR), power consumption, and space/time resolution. In this dissertation, we focus on invasive recordings, in particular extracellular electrophysiology and two-photon calcium imaging, where we can obtain high SNR signals at the single-neuron level and where the datasets in this dissertation are primarily derived from.

Extracellular electrophysiology records voltage fluctuations generated by action potentials using microelectrodes, providing a direct electrical measurement of spiking with sub-millisecond temporal resolution and good SNR at the level of single units. Unlike traditional depth electrodes [Ref37] and electrode arrays (e.g., Utah arrays) [Ref38], modern high-density

probes such as Neuropixels [Ref39, Ref5] can simultaneously record from hundreds to thousands of neurons across multiple brain regions, but the spatial sampling remains relatively sparse and biased toward neurons near the electrodes.

In contrast, two-photon calcium imaging measures neural activity indirectly via fluorescence changes of synthetic dyes or genetically encoded calcium indicators that report changes in intracellular calcium concentration associated with action potentials [Ref40]. This optical readout offers single-cell spatial resolution and enables simultaneous monitoring of hundreds to thousands of neurons within a local volume, with the possibility of targeting specific cell types through genetic strategies. However, the calcium signal is a temporally filtered proxy for spiking: indicator kinetics and cellular calcium handling broaden individual events over tens to hundreds of milliseconds, limiting effective temporal resolution and complicating the recovery of precise spike timing.

As a result, electrophysiology is generally preferred when fine temporal structure and high-frequency dynamics are critical, whereas calcium imaging excels when dense spatial coverage, cell-type specificity, and the ability to track identified neurons over days to weeks are prioritized.

2.1.2 Population Dynamics and Latent Manifolds

One of the key conceptual shifts in modern systems neuroscience is the move from single-neuron tuning descriptions to population-level dynamical motifs. In sensorimotor cortex, for example, the activity of individual neurons may exhibit no readily discernible moment-by-moment correspondence to externally measured motor outputs, rendering behavior decoding and mechanistic interpretation at the single-neuron level challenging [Ref41]. This has motivated a dynamical-systems view in which neural computation is expressed in the evolving geometry of population activity rather than in isolated unit responses [Ref9]. Conceptually, this perspective draws on the theories of nonlinear dynamical systems, where trajectories in a high-dimensional state space, together with structures such as fixed points and limit cycles, provide the language for describing system behavior [Ref42, Ref43, Ref44].

Population activity at each time point can be represented as a high-dimensional vector $x_t \in \mathbb{R}^N$, where N is the number of recorded neurons. Across time, these vectors trace a trajectory through neural state space spanned by the firing activity of member neurons. Empirical evidence suggests that these trajectories often lie on a low-dimensional manifold embedded in the high-dimensional firing space [Ref41]. This low-dimensional latent structure has been reported across motor, cognitive and sensory domains, suggesting that manifold organization may be a general feature of large-scale neural activity and a convenient coordinate system for describing population computations [Ref45, Ref14, Ref10, Ref46, Ref47].

Classical methods such as Principal Component Analysis (PCA), Factor Analysis (FA), Gaussian Process Factor Analysis (GPFA) [Ref15], latent Linear Dynamical Systems (LDS) [Ref48, Ref49, Ref50, Ref51] have been used to uncover this structure. More recent approaches extended these ideas with switching linear dynamical systems [Ref16] and nonlinear deep learning models [Ref20, Ref25, Ref52]. In particular, RNN-based encoder–decoder architectures trained on reconstruction or predictive objectives provide an effective route to discovering low-dimensional latent structure [Ref20, Ref53, Ref54]. These models learn latent trajectories that accurately forecast or reconstruct single-trial neural activity while compressing its dimensionality, often revealing interpretable clusters in latent space that correspond to distinct behavioral conditions. More broadly, analyses of trained RNNs demonstrate systematic dimensionality compression in their internal representations as the network learns task-relevant dynamics, offering a mechanistic example of how low-dimensional manifolds can emerge from high-dimensional recurrent circuits [Ref55, Ref56].

Beyond identifying a manifold, an active line of work asks how neural circuits use this low-dimensional structure. Studies of motor learning with brain–computer interfaces suggest that pre-existing intrinsic manifolds can constrain the patterns of population activity that are readily learned: perturbations requiring within-manifold adjustments are typically acquired more rapidly than those demanding activity outside the intrinsic subspace [Ref57]. Complementary analyses demonstrate that even within a single task, populations can transiently reorganize activity into functionally distinct low-dimensional subspaces—such as preparatory

versus movement-related modes—highlighting how flexible computations may be implemented through structured rotations and reconfigurations of latent activity [Ref58].

The manifold framework also offers a principled way to reason about stability across time. Long-term recordings in primate motor cortex indicate that latent dynamics within a preserved manifold can remain remarkably stable over long timescales, even when the recorded single units and their tuning properties drift, suggesting that neural computations underlying consistent behavior might be preserved within a stable population-level manifold [Ref59, Ref60]. These observations are especially impactful for robust decoding and BCI design, where leveraging stable latent structure may mitigate the need for frequent recalibration under neural plasticity and recording nonstationarities.

2.2 Neural Decoding and Brain–Computer Interfaces

Neural decoding seeks to infer variables of interest—such as sensory stimuli, movement kinematics, cognitive state, or intended speech—from measured neural activity. Brain–computer interfaces (BCIs) realize this goal in a system that translates neural signals into control commands for external devices, with the overarching aim of restoring function or communication in individuals with neurological injury or disease.

Neural decoding can be formulated as a supervised task. Let $x_t \in \mathbb{R}^N$ denote the neural activity vector at time t, e.g., binned spikes, firing rates, or local field-potential features, $X_{t-T:t} = \{x_{t-T}, \ldots, x_t\} \in \mathbb{R}^{T\times N}$ denote the window of historical neural activity and let $y_t \in \mathbb{R}^D$ denote the variable labels to be decoded, e.g., movement kinematics, or discrete labels such as reaching direction, phoneme, or character. Neural decoding finds a mapping $f_\theta : \mathbb{R}^N \mapsto \mathbb{R}^D$ or $f_\theta : \mathbb{R}^{T\times N} \mapsto \mathbb{R}^D$ from neural observations to behaviorally relevant variables. For instantaneous decoding, the output depends only on the neural activity at the same instance of time:

$$\hat{y}_t = f_\theta(x_t). \tag{2.1}$$

More generally, however, the decoded output can depend on a window of neural history

$X_{t-T:t} = \{x_{t-T}, \ldots, x_t\}$:

$$\hat{y}_t = f_\theta(X_{t-T:t}). \tag{2.2}$$

Modern BCIs utilize neural modalities spanning noninvasive methods such as electroencephalography (EEG) and magnetoencephalography (MEG), as well as invasive methods including Calcium imaging, electrocorticography (ECoG) and intracortical microelectrode arrays, each involving distinct trade-offs in spatial and temporal resolution, long-term stability, and clinical risks. In particular, invasive intracortical recordings provide high-resolution access to spiking activity, enabling fine-grained decoding of motor and speech attempts and have driven some of the most compelling demonstrations of high-performance BCIs [Ref61].

2.2.1 Intracortical BCIs for Motor Restoration

The translation of intracortical BCIs to humans has produced striking advances in restoring motor function. The BrainGate studies demonstrated that individuals with tetraplegia can achieve neural control of cursors and other assistive devices from analyzing neural populations recorded from motor cortex, providing clinical proof-of-concept for iBCI feasibility [Ref62, Ref63]. Subsequent work extended these capabilities to high-dimensional control of robotic arms [Ref63]. Notably, an individual with tetraplegia achieved coordinated multi-degree-of-freedom reach-and-grasp control, illustrating that intracortical signals can support complex, functional actions in daily activities [Ref63, Ref30]. Algorithmic developments such as Kalman-filter and RNN variants have also been critical, substantially improved online performance in later studies [Ref64, Ref20].

2.2.2 Intracortical BCIs for Communication Restoration

In parallel to motor restoration, communication BCIs have also advanced rapidly. Cursor-based spelling systems demonstrated that intracortical BCIs can support reliable text decoding in people with paralysis, with improvements driven by both algorithmic and interface design [Ref31]. More recent studies have achieved substantially higher communication rates by

decoding intended handwriting, leveraging the rich temporal structure of writing-related motor execution to enable fast, accurate brain-to-text communication in real time [Ref23]. Speech neuroprostheses represent another frontier of BCI advancements. ECoG studies have shown that attempted speech can be decoded into words and sentences using RNN combined with language models [Ref32]. Systems utilizing spiking activity have also demonstrated large-vocabulary, high-throughput speech-to-text decoding, with performance approaching levels of everyday communication, though substantial work remains on reducing calibration burden and improving long-term robustness [Ref24, Ref65].

2.2.3 Challenges for Neural Decoding and Motivations for Representation Learning

Despite these advances, several challenges remain for both neural decoding and clinical translation of BCI. First, neural data are noisy, scarce, and heterogeneous as they are collected by individual labs studying different scientific questions. The lack of standardized data collection protocol, processing pipelines, and high-capability algorithms to handle these noise and heterogeneity hinders the effective analyses of large scale datasets. Second, long-term recordings suffer from neural nonstationarity that changes the composition of recorded populations over time, which can complicate the extraction of stable neural identity, degrade the performance of fixed decoders, and increase the need for frequent BCI recalibration. These challenges will require models that can efficiently learn meaningful representations of population activity while remaining robust to variability across trials, sessions, and contexts.

2.3 Neural Variability Across Scales

Neural activity is variable across multiple scales, including *trials*, *recording sessions*, and *behavioral contexts*. At the trial level, even under identical stimuli or repeated behavior, neural responses can fluctuate substantially. At longer timescales, neural representations can drift over days to weeks, changing the relationship between population activity and behavior across sessions. Another type of variability emerges across behavioral or task contexts, where changes in cognitive states, arousal, or body movements unrelated to the task of interest

can reshape the geometry of population activity. Studies have shown that a substantial fraction of what has historically been treated as "noise" may in fact reflect structured signals related to unmeasured movements, internal states, or other latent variables, complicating naive task-focused interpretations of neural activity [Ref33].

2.3.1 Sources of Neural Variability

Neural variability originates from both biophysical and technical factors. At the cellular level, intrinsic cellular and synaptic stochasticity contribute to trial-to-trial fluctuations. In particular, the probabilistic opening and closing of ion channels gives rise to channel noise, which can influence the membrane potential dynamics and spike timing variability [Ref33]. Feedforward and recurrent dynamics between networks of neurons can propagate the noise at the cellular level to a broader scale, causing variability at the network level and whole-brain level [Ref33]. Across sessions, a major technical contributor to variability in neural observations is the instability of chronic recordings. Even when the underlying behavior remains consistent, the recorded population can change because of electrode micromotion relative to the tissue, resulting in the appearance or disappearance of units and changes of their positions in the electrode space. These factors occur in parallel with biological processes such as learning and plasticity, which can further change the tuning characteristics of neurons over longer timescales.

2.3.2 Techniques to Handle Neural Nonstationarity

To tackle the issue of neural nonstationarity, alignment techniques were proposed to align the testing sessions to match the distribution of a training session, usually performed in the latent space. These techniques vary from linear methods using Canonical Correlation Analysis (CCA) [Ref60, Ref66], linear stabilizer [Ref67], linear distribution alignment [Ref68], to nonlinear using generative adversarial networks (GAN) [Ref69, Ref70], RNN [Ref71, Ref72, Ref73], and diffusion models [Ref74]. All these approaches, however, still require explicit alignment

procedures with model parameter updates and test-time labels in some cases to adapt the pretrained decoder to unseen populations.

2.4 Related Machine Learning Concepts

2.4.1 Transformers

Transformers have emerged as the *de facto* architecture for representation learning with sequences. Originally introduced for sequence-to-sequence learning in natural language processing, Transformer replaces recurrence with stacked attention and feedforward blocks, enabling highly parallelizable training and improved expressivity over long contexts [Ref19]. This architecture has since been applied to a broad range of modalities, including vision, speech, and other types of time series, serving as a promising candidate for population-level neural modeling in which interactions among many units drive the population responses over time [Ref75, Ref76, Ref77, Ref25].

Self-Attention Given an input sequence of token embeddings $H \in \mathbb{R}^{T\times d}$, self-attention computes contextualized representations by forming *queries*, *keys*, and *values*

$$Q = HW^Q, \quad K = HW^K, \quad V = HW^V, \tag{2.3}$$

where $W^Q, W^K, W^V \in \mathbb{R}^{d\times d_k}$ are learned projection matrices. The attention weights are computed by scaled dot-products,

$$\mathrm{Attn}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V. \tag{2.4}$$

This operation allows each token to aggregate information from all other tokens with input-dependent weights. In contrast to convolutional or recurrent operators with fixed local connectivity or sequential update rules, self-attention provides a flexible mechanism for learning global interaction structure, a property that is potentially important for neural data where informative dependencies may span widely separated time points or distributed subpopulations.

Multi-Head Attention To represent multiple interaction patterns in parallel, the Transformer uses multi-head attention:

$$\mathrm{MHA}(H) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O, \tag{2.5}$$

where $\mathrm{head}_i = \mathrm{Attn}(HW_i^Q, HW_i^K, HW_i^V)$ and W^O is an output projection [Ref19]. Multi-head structure thus encourages learning distinct, complementary subspaces of interaction among tokens. In population analyses, this property can be viewed as enabling simultaneous discovery of multiple latent communication or coordination patterns among neural units.

Feedforward Network and Residual Connection A standard Transformer encoder block alternates multi-head self-attention with a token-wise feedforward network:

$$H' = \mathrm{LayerNorm}(\big(H + \mathrm{MHA}(H)\big)), \tag{2.6}$$

$$H'' = \mathrm{LayerNorm}(\big(H' + \mathrm{FFN}(H')\big), \tag{2.7}$$

where $\mathrm{FFN}(\cdot)$ is typically a two-layer MLP applied identically at each token. Residual connections and normalization stabilize optimization and support deep stacking of multiple transformer blocks. While the original formulation used post-norm, pre-norm variants are widely adopted for improved gradient flow in deep models [Ref78].

Positional embeddings Since the self-attention mechanism is permutation-equivariant over tokens, Transformers require an explicit mechanism to encode order in sequences. The original architecture introduced sinusoidal positional encodings added to token embeddings [Ref19]. Learned absolute positional embeddings and relative position schemes are also common, with relative encodings often improving generalization to longer sequences by modeling distances rather than absolute indices [Ref27, Ref79, Ref80]. For neural time series, positional embeddings can represent time bins (temporal positional encoding) or indices of neural units (spatial positional encoding) depending on whether the embeddings are applied along the temporal or spatial dimension, and are essential for encoding the notion of time or neural identities when modeling population responses [TL1].

Relevance to Neural Population Analyses Transformers are conceptually well suited to several demands of modern systems neuroscience. First, self-attention provides an efficient

mechanism for capturing long-range temporal dependencies and cross-neuron interactions. This complements classical approaches based on recurrent neural networks by offering a highly expressive alternative for learning structured population representations with the large number of neurons over long time scales. Second, the wide variety of designs and training techniques accompanying the attention mechanism makes it easy to impose neuroscience domain-specific inductive biases, such as factorized attention over neurons and time, causal modeling, or context-dependent token identity. Third, the high representational capacity of Transformers and their ability to handle arbitrary number of tokens make them a plausible backbone for future large-scale pretraining on heterogeneous neural datasets, potentially enabling reusable representations for neuron cell type characterization and decoding generalization. Overall, Transformers provide a unifying architectural framework for learning rich, contextual representations from multi-channel time series data, making them the go-to architecture for neural population analyses.

2.4.2 In-Context Learning

Large Language Models (LLMs) pretrained on large corpora of texts exhibit the ability to learn new tasks in-context [Ref81]. That is, conditioning on a few demonstrations of input-target pairs, LLMs can generalize to unseen cases without updating their weights. This ICL ability has proven useful across a wide range of tasks [Ref82, Ref83]. While ICL typically underperforms a specialized LLM finetuned for a specific downstream task, it still surpasses zero-shot inference, and is particularly valuable when finetuning is not feasible due to resource constraints such as time or computational power, or the inaccessibility of proprietary LLMs [Ref84].

2.4.3 Permutation-Invariant Neural Networks for Set-Structured Inputs

While conventional neural networks are designed for fixed dimensional data instances, in many set-structured applications such as point cloud object recognition or image tagging, the inputs have no intrinsic ordering, advocating for a class of models that are permutation-

invariant by design [Ref85, Ref86, Ref87]. One such work, DeepSets, introduced a set average pooling approach serving as a universal approximator for any set function [Ref85]. Follow-up works [Ref86, Ref88] extended this pooling method to include max-pooling and attention mechanisms [Ref19].

Chapter 3

LEARNING SINGLE-TRIAL REPRESENTATIONS FROM NEURAL POPULATION DYNAMICS WITH SPATIOTEMPORAL TRANSFORMERS

This chapter contains material that was previously published in [TL1].

3.1 Background

One of the prominent questions in systems neuroscience is how neurons perform computations that give rise to behaviors. Recent evidence suggests that computation in the brain could be governed at the population level [Ref89, Ref9]. Populations of neurons are proposed to obey an internal dynamical rule that drives their activities over time [Ref8, Ref90]. Inferring these dynamics on a single trial basis is crucial for understanding the relationship between neural population responses and behavior, subsequently enabling the development of robust decoding schemes with wide applicability in brain-computer interfaces (BCI) [Ref23, Ref30, Ref91]. However, modeling population dynamics on single trials is challenging due to the stochasticity of individual neurons making their spiking activity vary from trial to trial even when they are subject to identical stimuli or recorded under repeated behavior conditions.

A direct approach to reduce the trial-to-trial variability of neural responses could be to average responses over repeated trials of the same behavior [Ref92, Ref93], to convolve the neural response with a Gaussian kernel [Ref15], or in general, to define a variety of neural activity measures [Ref94]. However, more success was found in approaches that explicitly model neural responses as a dynamical system, including methods treating the population dynamics as being linear [Ref50, Ref51], switched linear [Ref16], non-linear [Ref20, Ref25], or reduced projected nonlinear models [Ref94]. Recent approaches leveraging recurrent

neural networks (RNN) have shown promising progress in modeling distinct components of a dynamical system - neural latent states, initial conditions and external inputs - on a moment-to-moment basis [Ref20, Ref95, Ref96]. These sequential methods rely on continuous processing of neural inputs at successive timesteps, causing latency that hampers applicability in real-time decoding of neural signals. Consequently to RNN-based approaches, Neural Data Transformer (NDT) [Ref25] was proposed as a non-recurrent approach to improve inference speed by leveraging the transformers architecture which learns and predicts momentary inputs in parallel [Ref19]. While successful, NDT has only focused on modeling the relationship of neural population activity between timesteps while ignoring the rich covariation among individual neurons. Neurons in a population have been shown to have heterogeneous tuning profiles where each neuron has a different level of preference to a particular muscle movement direction [Ref97, Ref98]. Neuron pairs also exhibit certain degree of correlation in terms of trial-to-trial variability (noise correlation) that affects the ability to decode the behaviors they represent [Ref9, Ref99]. These spatial correlations characterize the amount of information that can be encoded in the neural population [Ref99], necessitating the need to model the neural population activity across both time and space dimensions.

3.2 Contributions

In this work, we propose to incorporate the information distributed along the spatial dimension to improve the learning of neural population dynamics, and introduce *SpatioTemporal* Neural Data Transformer, an architecture based on Neural Data Transformer which explicitly learns both the spatial covariation between individual neurons and the temporal progression of the entire neural population. We summarize our main contributions as follows:

- We introduce STNDT which allows the transformer to learn both the spatial coordination between neurons and the temporal progression of the population activity by letting neurons attend to each other while also attending over temporal instances.

- We propose a contrastive training scheme, complementary to the mask modeling objec-

tive, to ensure the robustness of model prediction against induced noise augmentations.

- We validate our model's performance on four neural datasets in the publicly available Neural Latents Benchmark suite [Ref100] and show that ensemble variants of our model outperforms other state-of-the-art methods, demonstrating its capability to model autonomous and non-autonomous neural dynamics in various brain regions while being agnostic to external behavior task structures.

- We show that the spatial attention, a feature unique to STNDT, identifies consistently important subsets of neurons that play an essential role in driving the response of the entire population. This exclusive attribute of STNDT provides interpretability and key insights into how the neural population distributes the computation workload among the neurons.

3.3 Methods

Problem formulation: Single-trial spiking activity of a neural population can be represented as a spatiotemporal matrix $X \in \mathbb{N}^{T \times N}$, where each column $X_i \in \mathbb{N}^T$ is the time series of one neuron, T is the number of time bins for each trial, and N is the number of neurons in the population. Each element X_{tn} in the matrix is the number of action potentials (spikes) that neuron n fires within the time bin t. Spike counts are assumed to be samples of an inhomogeneous Poisson process $P(\lambda(t, n))$ where $\lambda(t, n)$ is the underlying true firing rate of neuron n at time t. The matrix $Y \in \mathbb{R}^{T \times N}$ containing $\lambda(t, n)$ fully represents the dynamics of the neural population and explains the observable spiking data of the respective trial. We propose to learn the mapping $\phi(X; W) : X \to Y$ by the Spatiotemporal Transformer with the set of weights W.

Spatiotemporal Neural Data Transformer: At the core of the transformer architecture is the multihead attention mechanism, where feature vectors learn to calibrate the influence of other feature vectors in their transformation. Spike trains are embedded into feature matrices $\tilde{X}$ with added sinusoidal positional encoding to preserve order information as

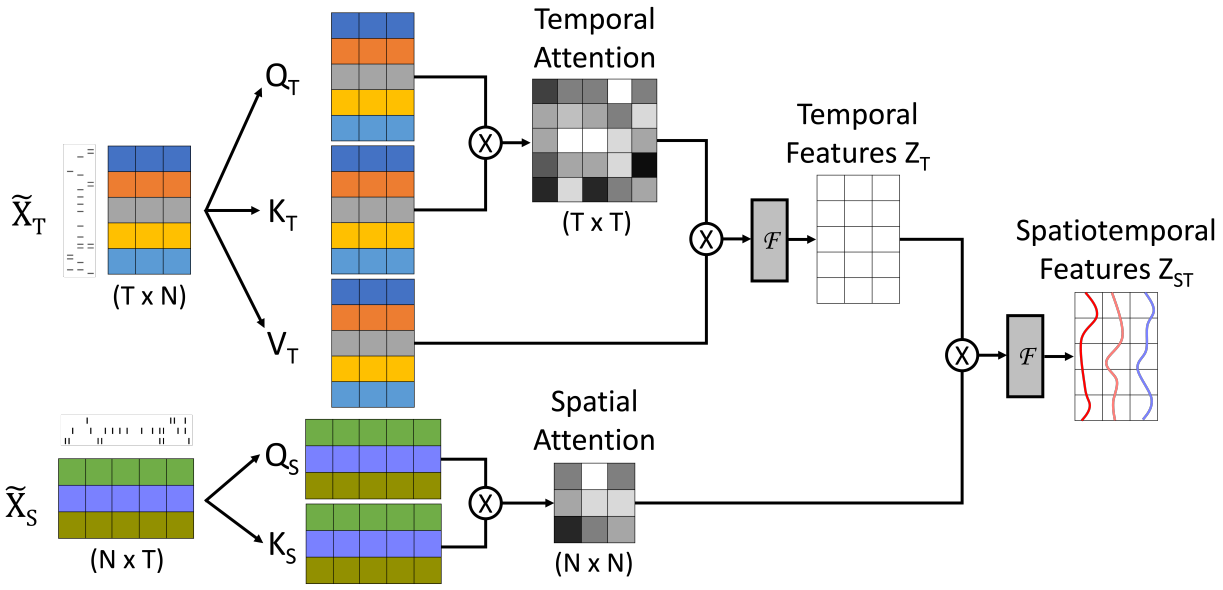

Figure 3.1: Spatiotemporal Neural Data Transformer (STNDT) architecture. Separate multihead self-attention modules are trained to learn spatial covariation and temporal progression of neural activities. Temporal attention feature matrix is treated as the matrix V upon which spatial attention is multiplied to give the final spatiotemporal features. Colors represent entities over which self attention is performed. The complete STNDT consists of multiple layers of such spatiotemporal attention modules.

initially proposed in [Ref19]. We employed separate embeddings to encode positions in each temporal and spatial dimension individually, resulting in two distinct feature embeddings $\tilde{X}_T = Emb(X) + P_T$ and $\tilde{X}_S = Emb(X^\top) + P_S$.

A set of three matrices $W_T^Q, W_T^K, W_T^V \in \mathbb{R}^{N \times N}$ are learned to transform T N-dimensional embedding $\tilde{X}_T = \{\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_T\}$ to queries $Q_T = \tilde{X}_T W_T^Q$, keys $K_T = \tilde{X}_T W_T^K$ and values $V_T = \tilde{X}_T W_T^V$, upon which latent variable Z_T is computed as:

$$Z_T = \text{Attention}(Q_T, K_T, V_T) = \mathcal{F}\left(\text{softmax}\left(\frac{Q_T K_T^\top}{\sqrt{N}}\right) V_T\right) \tag{3.1}$$

The outer product of $Q_T K_T^\top$ represents the attention each x_i pays to all other x_j and determines how much influence their values v_j have on its latent output z_i. $\mathcal{F}$ is the sequence of concatenating multiple heads and feeding through a feedforward network with ReLU activation [Ref19]. We used 2 heads for all reported models.

Implementations of transformers in popular applications such as in natural language processing literature consider each feature vector x_i as an N-dimensional token in a sequence, equivalent to a word in a sentence. Elements in the N-dimensional vector therefore serve as a convenient numerical representation and do not have inherent relationships among them. The attention mechanism thus only models the relationship between tokens in a sequence. In our application, each feature vector x_i is a collection of firing activities of N physical neurons among which there exists an interrelation as neuronal population acts as a coordinated structure with complex interdependencies rather than standalone individuals. We therefore propose to model both the temporal relationship - the evolution of neural activities - and the spatial relationship - covariability of neurons - by learning two separate multihead attention blocks (Figure 3.1). The temporal latent state $Z_\mathcal{T}$ is computed with temporal attention block as in Equation 4.8. In parallel, spatial attention block operates on the spatial embedding $\tilde{X}_S$ and learns an attention weights matrix signifying the relationship between neurons:

$$A_\mathcal{S} = \text{softmax}\left(\frac{Q_\mathcal{S} K_\mathcal{S}^\top}{\sqrt{T}}\right) \tag{3.2}$$

where $Q_\mathcal{S} = \tilde{X}_S W_\mathcal{S}^Q$ and $K_\mathcal{S} = \tilde{X}_S W_\mathcal{S}^K$.

This $A_\mathcal{S}$ matrix is then multiplied with the transpose of temporal latent state $Z_\mathcal{T}$ to incorporate the influence of spatial attention on the final spatiotemporal latent state $Z_{\mathcal{ST}}$:

$$Z_{\mathcal{ST}} = \mathcal{F}(A_\mathcal{S} Z_\mathcal{T}^\top) \tag{3.3}$$

For stable training, as in [Ref19] we used layer normalization before $\tilde{X}_T$, $\tilde{X}_S$, $A_S Z_T^\top$ and feedforward layers. Residual connections are also employed around temporal attention, feedforward layers and $A_S Z_T^\top$.

Mask modeling and contrastive losses: Similar to [Ref25], we train the spatiotemporal transformer in an unsupervised way with BERT's mask modeling objective [Ref27]. During training, a random subset of spike bins along both spatial and temporal axes of input X are masked (zero-ed out or altered) and the transformer is asked to reconstruct the log firing rate at the masked bins such that the Poisson negative log likelihood is minimized:

$$\mathcal{L}_{mask} = \sum_{i=1}^{N} \sum_{j=1}^{T} \exp(\tilde{z}_{ij}) - \tilde{x}_{ij}\tilde{z}_{ij} \tag{3.4}$$

where $\tilde{z}_{ij}$ and $\tilde{x}_{ij}$ are the log output firing rate and input spike of neuron i at timestep j if location ij is masked.

Neural dynamics are shown to be embedded in a low-dimensional space, i.e. model prediction should be fairly consistent when a smaller subset of neurons are used compared to when the entire population is taken into account. Furthermore, in stereotyped behaviors often found in neuroscience experiments, trials with the same condition should yield similar output firing rate profiles. Therefore, to enhance robustness of model prediction to neural firing variability we further constrain model firing rate outputs by a contrastive loss, such that different augmentations of the same trial input remain closer to each other and stay distant to other trial inputs. We adopt the NT-XEnt contrastive loss introduced in [Ref29]:

$$\mathcal{L}_{contrastive} = \sum_{ij} l_{ij} = \sum_{ij} -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbf{1}_{k \neq i} \exp(\text{sim}(z_i, z_k)/\tau)} \tag{3.5}$$

where $\text{sim}(u, v) = u \top v/(\|u\|\|v\|)$ is the cosine similarity between two predictions u and v on two different augmentations of input x and τ is the temperature parameter.

Transformations such as dropping out neurons and jittering samples in time have been used to create different views of neural data [Ref101]. In our work, we define the augmentation transformation as random dropout and alteration of spike counts at random elements in the original input matrix X, similar to how masking is done, i.e. zero out or change spike counts to random integers at random neurons and timesteps. See Appendix for details on probabilities used to create these augmentations.

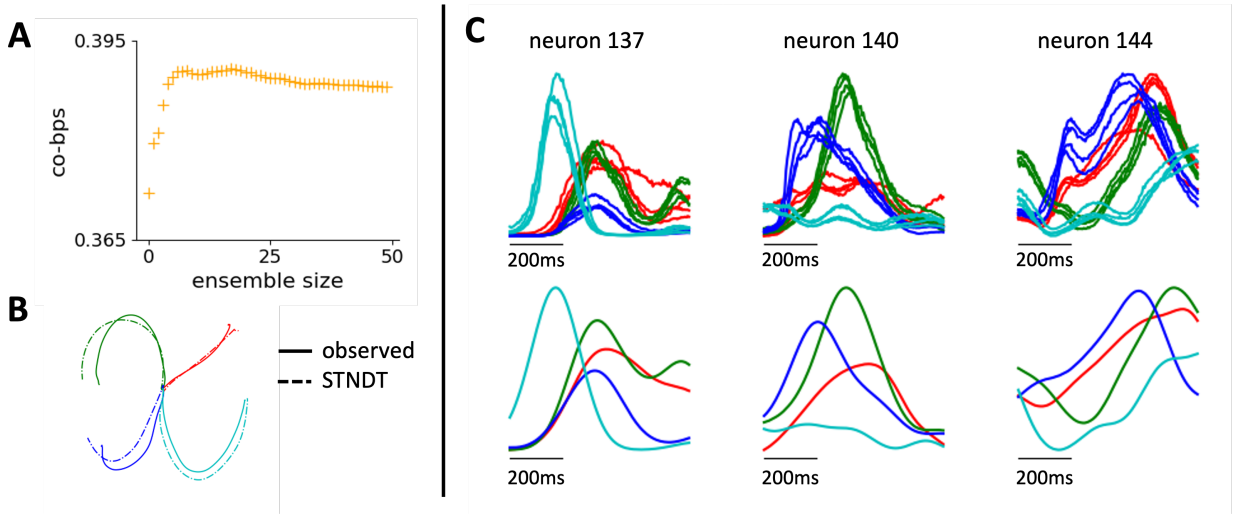

Figure 3.2: **A**: co-bps metrics improves when multiple models are ensembled together. **B**: STNDT facilitates accurate inference of behavior from spiking data. Decoded hand trajectories from 4 trials (dashed line) closely match the ground truth trajectories (solid line). **C**: STNDT uncovers the stereotyped feature of neural activity in structured behaviors. Firing rate prediction and PSTHs of three example neurons are shown. Trials belonging to the same condition are plotted with the same color (4 trials per condition shown). All results are shown for MC_Maze dataset.

Bayesian hyperparameter tuning: We follow [Ref102] to use Bayesian optimization for hyperparameters tuning. We observe that the primary metrics co-smoothing bits/spike (co-bps) are not well correlated with the mask loss (see Appendix), while co-bps, vel R^2, psth R^2 and fp-bps are more pairwise correlated. Therefore, we run Bayesian optimization to optimize co-bps for M models then select the best N models as ranked by validation co-bps, and ensemble them by taking the mean of the predicted rates of these N models.

3.4 Results

Datasets and evaluation metrics: We evaluate our model performance on four neural datasets in the publicly available Neural Latents Benchmark [Ref100]: MC_Maze, MC_RTT, Area2_Bump, and DMFC_RSG. The 4 datasets cover autonomous and non-autonomous neural population dynamics recorded on rhesus macaques in a variety of behavioral tasks (delayed reaching, self-paced reaching, reaching with perturbation, time interval reproduction) spanning multiple brain regions (primary motor cortex, dorsal premotor cortex, somatosensory cortex, dorso-medial frontal cortex). The diverse scenarios and systems offer comprehensive evaluation of a latent variable model and serve as a standardized benchmark for comparison between different modeling approaches. We use different metrics to measure performance of our model depending on the particular behavior task of each dataset, following the standard evaluation pipeline in [Ref100]. We evaluate and report our model performance on the hidden test split held by NLB to have a fair comparison with other state-of-the-art (SOTA) methods. See [Ref100] for further details of evaluation strategy and how the metrics are calculated.

- **Co-smoothing (co-bps)**: the primary metric, measuring the ability of the model to predict activity of held-out neurons it has not seen during training. Co-bps is tied to the goodness of mask loss evaluated for held-out neurons.

- **Behavior decoding (vel R^2 or tp-corr)**: measures how useful the model firing rates prediction can be used to decode behavior (the velocity of primate's hand in the cases of MC_Maze and Areas_Bump datasets, or the correlation between neural speed and time between Set cue and Go response in DMFC_RSG dataset).

- **Match to peri-stimulus time histogram (psth R^2)**: indicates how well predicted firing rates match the peri-stimuls time histogram in repeated, stereotyped task structures.

- **Forward prediction (fp-bps)**: measures model's ability to predict unseen future

Table 3.1: Performance of STNDT as compared to SOTA methods on MC_Maze and MC_RTT datasets

	MC_Maze				MC_RTT		
Methods	co-bps↑	vel R^2↑	psth R^2↑	fp-bps↑	co-bps↑	vel R^2↑	fp-bps↑
GPFA	0.1872	0.6399	0.5150	–	0.1548	0.5339	–
Smoothing	0.2109	0.6238	0.1853	–	0.1468	0.4142	–
SLDS	0.2249	0.7947	0.5330	1.1579	0.1649	0.5206	0.0620
MINT	0.3304	**0.9121**	**0.7496**	0.2076	0.1676	0.5953	0.1012
AutoLFADS	0.3364	0.9097	0.6360	0.2349	0.1868	0.6167	0.1213
iLQR-VAE	0.3559	0.8840	0.6062	0.1480	–	–	–
AESMTE1 (single)	0.3599	0.9105	0.6641	0.2470	0.1927	**0.6627**	0.1229
AESMTE3 (ensemble)	0.3676	0.9114	0.6683	0.2589	0.2053	0.6334	**0.1344**
STNDT single (ours)	0.3691	0.8985	0.6567	0.2505	0.1938	0.6143	0.0988
STNDT ensemble (ours)	**0.3862**	0.9095	0.6693	**0.2686**	**0.2095**	0.6270	0.1244

activity of the neural population. It is computed in the similar manner as co-bps but on the held-out time points of all neurons.

Baselines: We compare STNDT against the following baselines, all of which have been evaluated using the same held-out test split.

Table 3.2: Performance of STNDT as compared to SOTA methods on Area2_Bump and DMFC_RSG datasets

Methods	Area2_Bump				DMFC_RSG			
	co-bps↑	vel R^2↑	psth R^2↑	fp-bps↑	co-bps↑	tp-corr↓	psth R^2↑	fp-bps↑
GPFA	0.1680	0.5975	0.5289	–	0.1176	−0.3763	0.2142	–
Smoothing	0.1544	0.5736	0.2084	–	0.1202	−0.5139	0.2993	–
SLDS	0.1960	0.7385	0.5740	0.0242	0.1243	−0.5412	0.3372	−0.0418
MINT	0.2735	0.8877	**0.9135**	0.1483	0.1821	−0.6929	**0.7013**	0.1650
AutoLFADS	0.2569	0.8492	0.6318	0.1505	0.1829	**−0.8248**	0.6359	0.1844
iLQR-VAE	–	–	–	–	–	–	–	–
AESMTE1 (single)	0.2801	0.8675	0.6367	0.1523	0.1733	−0.6189	0.5267	0.1511
AESMTE3 (ensemble)	0.2860	**0.8999**	0.7109	**0.1603**	0.1886	−0.7601	0.6064	0.1828
STNDT single (ours)	0.2818	0.8766	0.6454	0.1357	0.1859	−0.5205	0.6051	0.1601
STNDT ensemble (ours)	**0.2898**	0.8913	0.7368	0.1476	**0.1940**	−0.4857	0.6452	**0.1910**

- **Smoothing** [Ref100]: A simple method where a Gaussian kernel is convolved with held-in spikes to produce smoothed held-in firing rates. Then a Poisson Generalized Linear Model (Poisson GLM) is fitted from the held-in smoothed rates to held-out rates.

- **GPFA** [Ref15]: extracts population latent states as a smooth and low dimensional evolution by combining smoothing and dimension reduction in a common probabilistic framework.

- **SLDS** [Ref16]: models neural dynamics as a switching linear dynamical system, which breaks down nonlinear data into sequences of simpler dynamical modes.

- **AutoLFADS** [Ref95]: models population activity as a non-linear dynamical system with bi-directional recurrent neural networks at the core and a scalable framework of hyperparameter tuning.

- **MINT** [Ref103]: an interpretable decode algorithm that exploits the sparsity and stereotypy of neural activity to interpolate neural states using a library of canonical neural trajectories.

- **iLQR-VAE** [Ref52]: improves upon LFADS with iterative linear quadratic regulator algorithm, an optimization-based recognition model to replace RNN as the inference network.

- **NDT** [Ref25]: leverages transformer architecture with some adaption to neural data to model temporal progression of neural activity across time. AESMTE1 is the best single model and AESMTE3 is the best emsemble of multiple models found as a result of Bayesian hyperparameter tuning [Ref102].

Spatiotemporal transformer achieves state-of-the-art performance in modeling autonomous dynamics We first tested STNDT on recordings of dorsal premotor (PMd) and motor cortex (M1) of a monkey performing a delayed reaching task (MC_Maze dataset) to evaluate the ability of STNDT to uncover single-trial population dynamics in a highly structured behavior. The dataset has been studied extensively in previous work [Ref20, Ref25, Ref95]. It consists of 2869 trials of monkey performing a center-out reaching task in a maze with obstructing barriers, composing 108 different conditions for straight and curved reaching trajectories. The monkey is trained to hold the cursor at the center while the target is presented and only move the cursor to reach the target after a 'Go' cue. The neural dynamics during the preparation and execution periods is well modeled as an autonomous dynamical system [Ref20].

We observed that by explicitly modeling spatial interaction, STNDT outperformed other state-of-the-art methods and improved NDT's ability to model autonomous single-trial

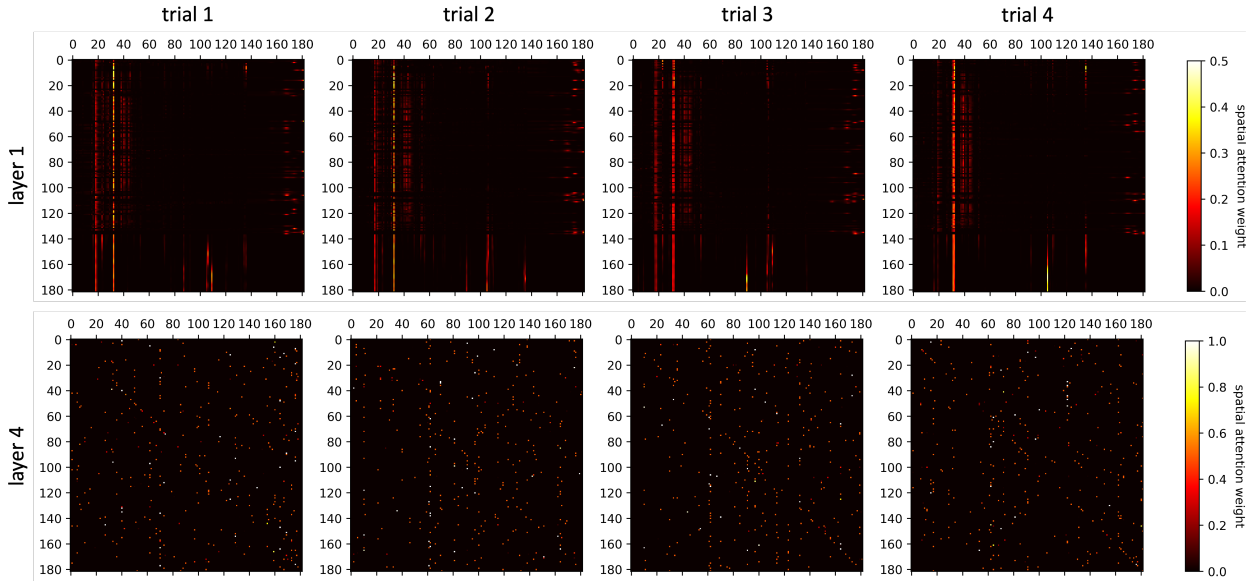

Figure 3.3: Visualization of STNDT's spatial attention weights in the first and last layers of four example trials. Attention weights in layer 1 reveal a consistent subset of neurons that are heavily attended to by all neurons in the population. The attention becomes more dispersed in deeper layers. Results are shown for 182 neurons in MC_Maze dataset.

dynamics as measured by the negative log likelihood of unobserved neural activity. The single STNDT model improved both Poisson log likelihood of heldout neurons (co-bps) and heldout timesteps (fp-bps). The performance is further increased by aggregating multiple STNDT models as shown in Table 3.1 and Figure 3.2A.

Since MC_Maze features repeated trials, the prediction of any latent variable models should uncover stereotypical patterns of neuronal responses for trials belonging to the same condition. Therefore, we computed PSTH which is the average of neural population response across trials of the same condition, and measure R^2 matching of model prediction to this PSTH. We observed that with the help of spatial modeling and contrastive loss, STNDT boosts NDT ability to recover this stereotyped firing pattern 3.1. We show in Figure 3.2C several responses of example neurons. STNDT firing rates prediction of trials under the same

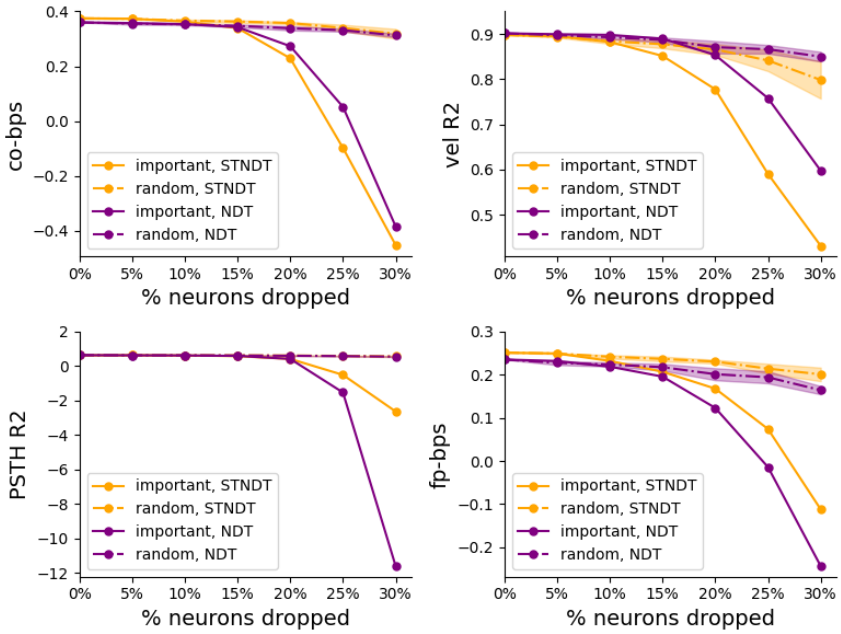

Figure 3.4: Spatial attention module, unique to STNDT, identifies important neurons that are the main driving force of population response to behavioral task. Performance of STNDT as measured by four evaluation metrics are plotted as neurons are incrementally dropped from input neural population. Performance significantly deteriorates when important neurons identified by STNDT are dropped, while only decreases slightly when random neurons are dropped. The effect of important neurons indentified by STNDT generalizes to vanilla NDT, which lacks a spatial attention structure. Shaded region represents 2 standard error of the mean. Results are shown for MC_Maze dataset.

condition exhibit a consistent, stable PSTH as desired. These predicted rates also decode behaviors accurately when mapped to hand velocity via a linear regression model (Table 3.1, Figure 3.2B).

Spatiotemporal transformer improves inference of non-autonomous neural dynamics underlying naturalistic behaviors There is much interest in systems neuro-

science to study neural dynamics in unconstrained, naturalistic behaviors as it is crucial for developing ubiquitous BCI decoders. We evaluated STNDT's applicability to this setting via recordings in primary motor cortex during self-paced reaching task (MC_RTT dataset) [Ref100, Ref104]. Unlike MC_Maze dataset, the monkey in this task continuously acquires targets which appear randomly in an 8x8 grid without preparatory periods, resulted in a wide variety of hand trajectories and trial lengths. We observe that STNDT achieves SOTA performance on the primary metric co-bps and performs on par with NDT on remaining metrics, while maintaining a more robust performance against random initializations of model weights (Table 3.1 and Appendix).

Spatiotemporal transformer better captures input-driven dynamics underlying sensory processes We next tested STNDT in a setting where unexpected input perturbations affect the neural dynamics in somatosensory cortex to probe whether STNDT can leverage spatial interaction to improve modeling of non-autonomous dynamics in this brain region. Area2_Bump dataset consists of recordings from the Area 2, which was shown in previous works to be driven by mechanical perturbation to the arm and contains information about whole-arm kinematics [Ref100, Ref105]. The task comprises of active and passive trials with a center hold period at the start. During active trials, the monkey performs a classic center-out reaching task. In passive trials, a force is applied on the monkey's hand in a random direction via a manipulandum, after which the monkey has to return to the center target and proceed with the task as in active trials. Despite the relatively small scale of the dataset, STNDT brings about further improvements to NDT performance in terms of co-bps and psth-R^2, on both single and ensemble levels.

Spatiotemporal transformer enhances prediction of neural population activity during cognitive task Dorsomedial frontal cortex (DMFC) is believed to serve as an intermediate layer between low-level sensory and motor areas, and possess distinct confluence of internal dynamics and inputs [Ref106, Ref107]. We are therefore interested to see if characterizing spatial relationship alongside temporal relationship and incorporating contrastive loss could help STNDT better model the dynamics in this brain region. We tested STNDT

Table 3.3: Pearson's correlation between spatial attention weight of a neuron versus mean and variance of its spiking activity.

	MC_Maze	MC_RTT	Area2_Bump	DMFC_RSG
ρ(spike mean, attn weight)	0.0164	0.2217	0.0327	0.0852
ρ(spike var, attn weight)	0.0124	0.2189	0.0353	0.0937

on the DMFC_RSG dataset [Ref100, Ref107] consisting of recordings from a rhesus macaque performing a time-interval reproduction task. The monkey is presented two 'Ready' and 'Set' stimuli separated by a specific time interval t_s while fixating eye and hold the joystick at the center position. It then has to execute a 'Go' response by either an eye saccade or joystick movement such that the time interval t_p between its reponse and the 'Set' cue is sufficiently close to t_s. STNDT successfully captures the dynamics in this cognitive task, outperforming NDT by a large margin across co-bps, psth-R^2 and fp-bps on both single and ensemble level (Table 3.2).

Spatial attention mechanism identifies important subsets of neurons driving the population dynamics

In Figure 3.3, we visualize spatial attention weights obtained from STNDT on the MC_Maze dataset in the first and last attention layers. Interestingly, spatial attention shows that in early layers, only a small subsets of neurons in the population are consistently attended to by all neurons. The spatial attention tends to disperse as the model goes to deeper layers. Strikingly, the subset of heavily-attended neurons stays relatively identical across different trials, hinting that these neurons might play a crucial role in driving the population response to the behavior task. We further tested this hypothesis by incrementally dropping the neurons heavily attended to (i.e. zeroing out their spiking activity input to the model) in a descending order of their attention weights identified in the first layer. We observed that dropping these important neurons identified by STNDT caused a significant decline in the model performance

(Figure 3.4). The performance decline was significantly more than the case where the same number of random neurons are dropped. To rule out the possible case that dropping neurons only has adverse effect on the spatial attention module but that effect propagates to the subsequent modules and indirectly impacts the performance of the overall STNDT pipeline, we repeated the experiment on the vanilla NDT model which, unlike STNDT, lacks a spatial attention structure. Interestingly, we observed the same performance deterioration when we dropped the spiking activity of STNDT-identified important neurons and asked a pretrained vanilla NDT to make inference on the resulting inputs. This finding suggests that the impact of the important neurons that only STNDT can identify might potentially generalize to other latent variable models that without input from these neurons, some latent variable models might not function optimally.

We further examine whether important neurons were selected by the spatial attention mechanism based on some criteria more sophisticated than simple firing statistics, as more active neurons tend to have higher signal-to-noise ratio and might encode more useful information with regard to behaviors. We find that the important neurons are not the ones with the highest spike counts or the least variability in spiking activity. In fact, attention weights of a neuron do not correlate or only correlate weakly to its firing activity statistics, as we show in Table 3.3 the Pearson's correlation of a neuron's attention weight with the mean and variance of its spiking activity. All correlation values have p-value $<$ 1e-4. These results indicate that STNDT's spatial attention has picked up on meaningful population features that are more significant than firing statistics of the neurons.

3.5 Appendix

3.5.1 Training details

We perform Bayesian hyperparameter optimization to obtain 120 candidates on each dataset for subsequent model ensembling. Training was done on RTX 2080 Ti GPUs. The sweep ranges for hyperparameters optimization are shown in Table 3.4.

Table 3.4: Training details

	MC_Maze	MC_RTT	Area2_Bump	DMFC_RSG
Dropout ratio	0 – 0.4	0 – 0.4	0 – 0.6	0 – 0.4
Temporal backward context	1 – 100	1 – 100	1 – 100	1 – 240
Temporal forward context	1 – 100	1 – 100	1 – 100	1 – 240
Initial learning rate	1e-5 – 1e-2	1e-4 – 1e-1	1e-5 – 1e-2	1e-5 – 1e-2
Learning rate warmup	0 – 7000	0 – 7000	0 – 7000	0 – 2000
Mask ratio	0 – 0.4	0 – 0.4	0 – 0.6	0 – 0.4
Zero mask ratio	0.5 – 1.0	0.5 – 1.0	0.5 – 1.0	0.5 – 1.0
Random mask ratio	0.3 – 1.0	0.6 – 1.0	0.9 – 1.0	0.9 – 1.0
Training time	∼65 hrs 6 GPUs	∼71 hrs 4 GPUs	∼19 hrs 5 GPUs	∼91 hrs 4 GPUs
Ensemble size	20	40	50	77

3.5.2 Model robustness across random initializations

To assess the robustness of STNDT against random initializations, we trained our best STNDT model and best AESMTE model with five different random seeds and report the mean as well as the standard error in Tables 3.5-3.8 below. For AESMTE, we used the same public code and the same set of hyperparameters of the best performing model they provided to ensure a fair comparison. All the results are obtained on the hidden test set held by NLB. The results indicate that STNDT maintains a gap over AESMTE and is more robust across initializations. The effect is observed on all four datasets and is most notable on the primary metric co-bps.

Table 3.5: Performance (mean±SEM) of STNDT with and without contrastive loss (CL) across 5 random seeds on MC_Maze dataset.

	MC_Maze			
Methods	co-bps↑	vel R^2↑	psth R^2↑	fp-bps↑
AESMTE1 (single)	0.3476 ± 0.0035	0.9057 ± 0.0006	0.6320 ± 0.0071	0.2365 ± 0.0031
STNDT single w/o CL	$\mathbf{0.3659} \pm 0.0003$	0.8937 ± 0.0013	$\mathbf{0.6562} \pm 0.0029$	$\mathbf{0.2446} \pm 0.0014$
STNDT single w/ CL	$\mathbf{0.3668} \pm 0.0005$	0.8932 ± 0.0012	$\mathbf{0.6534} \pm 0.0046$	$\mathbf{0.2447} \pm 0.0009$

Table 3.6: Performance (mean±SEM) of STNDT with and without contrastive loss (CL) across 5 random seeds on MC_RTT dataset.

	MC_RTT		
Methods	co-bps↑	vel R^2↑	fp-bps↑
AESMTE1 (single)	0.1729 ± 0.0090	0.5847 ± 0.0618	0.0974 ± 0.0044
STNDT single w/o CL	$\mathbf{0.1883} \pm 0.0019$	$\mathbf{0.6021} \pm 0.0051$	0.0958 ± 0.0039
STNDT single w/ CL	$\mathbf{0.1923} \pm 0.0009$	$\mathbf{0.5996} \pm 0.0060$	0.0932 ± 0.0030

3.5.3 Correlations of evaluation metrics

We show in Figure 3.5 the correlation between evaluation metrics and validation mask loss obtained at the final training epoch where the best model is checkpointed. The mask loss is still a good objective to guide the training in the early episodes. However, after reaching certain goodness of fit, it is no longer indicative of the model performance as measured by the four metrics. Therefore we chose to optimize the co-bps metric during Bayesian hyperparameter optimization.

Table 3.7: Performance (mean±SEM) of STNDT with and without contrastive loss (CL) across 5 random seeds on Area2_Bump dataset.

	Area2_Bump			
Methods	co-bps↑	vel R^2↑	psth R^2↑	fp-bps↑
AESMTE1 (single)	0.2483 ± 0.0096	0.8370 ± 0.0175	0.5628 ± 0.0423	0.1261 ± 0.0080
STNDT single w/o CL	$\mathbf{0.2717} \pm 0.0011$	$\mathbf{0.8730} \pm 0.0048$	$\mathbf{0.7145} \pm 0.0029$	$\mathbf{0.1435} \pm 0.0019$
STNDT single w/ CL	$\mathbf{0.2738} \pm 0.0009$	$\mathbf{0.8720} \pm 0.0020$	$\mathbf{0.7098} \pm 0.0038$	$\mathbf{0.1477} \pm 0.0025$

Table 3.8: Performance (mean±SEM) of STNDT with and without contrastive loss (CL) across 5 random seeds on DMFC_RSG dataset.

	DMFC_RSG			
Methods	co-bps↑	tp-corr ↓	psth R^2↑	fp-bps↑
AESMTE1 (single)	0.1795 ± 0.0008	-0.7297 ± 0.0104	0.5584 ± 0.0207	0.1597 ± 0.0041
STNDT single w/o CL	$\mathbf{0.1820} \pm 0.0011$	-0.5210 ± 0.0435	$\mathbf{0.6080} \pm 0.0015$	0.1429 ± 0.0059
STNDT single w/ CL	$\mathbf{0.1840} \pm 0.0008$	-0.5148 ± 0.0408	$\mathbf{0.6097} \pm 0.0071$	0.1444 ± 0.0095

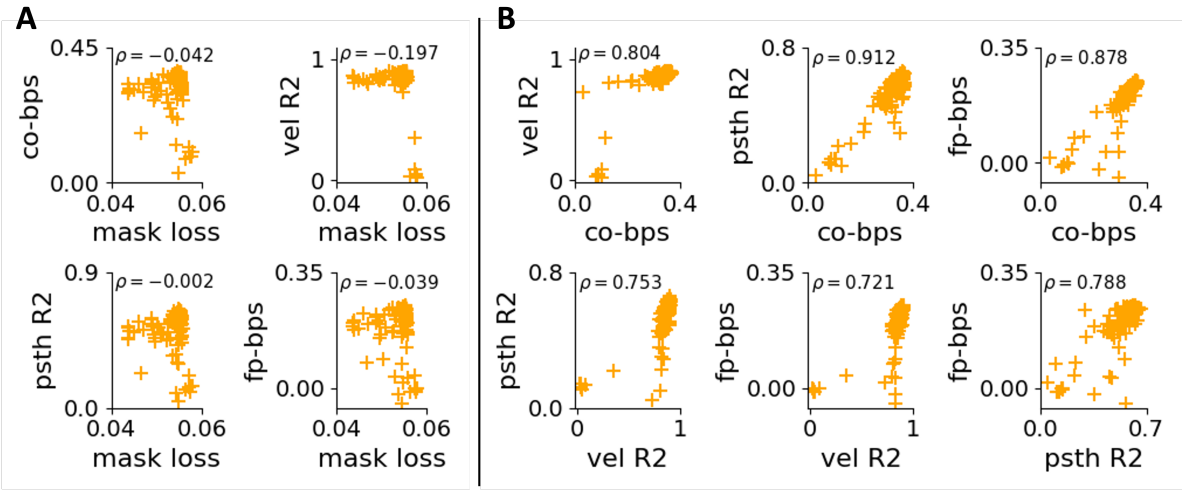

Figure 3.5: Correlations of evaluation metrics. **A**: Four evaluation metrics of 120 models obtained from Bayesian hyperparameter optimization on MC_Maze dataset are plotted against mask loss. The metrics evaluated at the end of the training do not correlate well with mask loss. **B**: The four metrics are more correlated with each other, therefore we opted for co-bps as the objective for Bayesian hyperparameter optimization.

Chapter 4

LEARNING TIME-INVARIANT REPRESENTATIONS FOR INDIVIDUAL NEURONS FROM MULTI-SESSION POPULATION DYNAMICS

This chapter contains material that was previously published in [TL2].

4.1 Background

Population recordings of neuronal activity enable relating behaviorally-relevant dynamics to the summary activity of the recorded population. While this has produced numerous insights into how the brain works [Ref9], the activity and identity of individual neurons should be analyzed to achieve a mechanistic understanding at the implementation level [Ref108], which may hold the key to new biologically-inspired algorithms [Ref109, Ref110]. Moreover, emerging experimental evidence suggests that neurons have diverse yet stable molecular identities, which can dictate their computational roles [Ref111, Ref7].

Joint (i.e., multimodal) profiling and alignment of electrophysiological features and gene expression of individual neurons suggest a good correspondence between these two modalities in slice experiments [Ref112, Ref113, Ref114, Ref115, Ref116]. Recently, population recordings of calcium activity followed by spatially registered single-cell transcriptomic recordings enabled similar joint profiling of *in-vivo* activity and molecular identity [Ref7]. Importantly, such activity depends on both the intrinsic physiological properties of neurons and the exogenous inputs (synaptic and modulatory) to those neurons, which are themselves a product of both sensory inputs to the organism and the recurrent activity in the brain.

While recording from molecularly defined neuron populations has been a popular method, these experiments do not allow for studying the concurrent responses of different neuron

types to stimuli. On the flip side, joint profiling of panneuronal population activity and transcriptomics is slow, expensive, and not available to many research labs. Learning the association between these two observation modalities can minimize the need for joint profiling and provide neurobiological insights.

The constancy of neuronal identity in adults in the face of a potentially rapidly changing environment represents a key challenge: the inferred identity should be invariant to time and the task that the organism engages with, suggesting that the inference method should ideally be invariant to those variables. In the absence of *a priori* information on identity, it is also desirable that the invariance extends to the number and the (arbitrary) ordering of experimental population. Moreover, a technical challenge common to many multimodal datasets is that only a relatively small fraction of the observations tend to be jointly characterized (or otherwise labeled), limiting the applicability of supervised approaches.

4.2 Contributions

To address these problems,

- We develop a self-supervised approach – Neuronal Time-Invariant Representations (NeuPRINT), to infer neuronal identity from population recordings by forming a model of activity dynamics that depends only on past activity of the neuron itself and statistics of past population activity that are invariant to the ordering of the individuals and asymptotically invariant to the size of the population.

- We demonstrate the utility of the inferred identities by reporting the performance of a simple classifier of transcriptomic identity on those representations and other baselines.

- We also study the impact of providing similarly invariant yet more detailed information on the population by partitioning it into center vs surround subsets, reflecting a well-known yet simple connectional and functional property of neuronal circuits [Ref117, Ref118, Ref119].

4.3 Methods

Implicit dynamics models for neuronal activity: Neuronal dynamics are significantly more complex than the HH equations because the physical distribution of the various channels on the neuronal arbor, nonlinear computing abilities of the dendrites, modulatory communication between neurons,etc. can (i) modulate the neuron-specific parameter set, (ii) add more parameters to that set, and (iii) change the functional form of HH equations. Moreover, the relationship between the membrane voltage and the observable of most neuronal population activity experiments, the calcium dynamics, is itself complex. Finally, it appears impossible to perform the detailed measurements needed to fit the parameters of the HH equations based on *in vivo* experiments with current technology. Thus, we pursue an implicit modeling approach while trying to capture the fundamental dependencies with the help of flexible deep neural network parametrization. Let $X_t^{(i)}$ denote the calcium activity of neuron i at time t and consider the following equation for dynamics:

$$\frac{dX_t^{(i)}}{dt} = f(X_t^{(i)}, \bar{P}_t^{(-i)}, \Phi^{(i)}), \tag{4.1}$$

where $\bar{P}_t^{(-i)}$ denotes the activity of all the neurons that provide (synaptic or extra-synaptic) input to neuron i at time t. $\Phi^{(i)}$ denotes a time-invariant representation for neuron i, which implicitly captures the intrinsic parameters of HH equations and other such time-invariant aspects of neuronal identity.

Permutation-invariant summary of population activity: Even if neurons were identifiable in population imaging experiments, the connectivity of neurons and our observations of it can be considered as stochastic events [Ref120]. Therefore, to enable the transfer of knowledge across sessions, experiments, and individuals, it is highly desirable to approximate the dependency of neuronal dynamics on $\bar{P}_t$ (Eq. 4.1) in a way that is invariant to permutations, number, and detailed identity of the neurons contributing to it. To achieve this, we propose to replace $\bar{P}_t$ with multiple (asymptotically) invariant statistics of the activity of a neighboring population of neurons, such as average activity [Ref121].

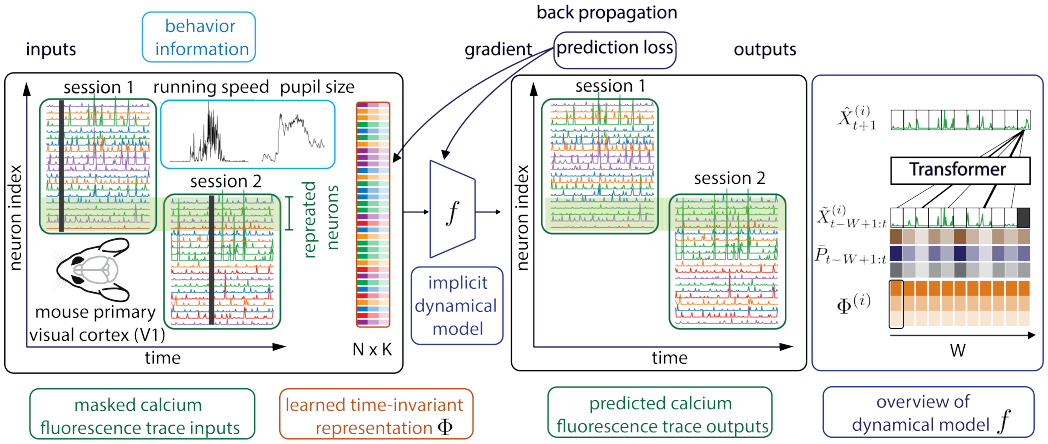

Figure 4.1: Overview of self-supervised representation learning framework NeuPRINT. Activities of N neurons (recorded by 2-photon calcium imaging of the mouse primary visual cortex) and behavior information (pupil size, running speed, etc.) across multiple sessions are used as inputs to fit an implicit dynamical model f and learn time-invariant $N \times K$ representation Φ. The learned representations are later evaluated on supervised downstream tasks to predict transcriptomic class & subclass identities. In the optimization framework, neuron-specific representation Φ_i is repeated at every time step, then concatenated with masked past neuronal activity $\tilde{X}^{(i)}_{t-W+1:t}$ and permutation-invariant population inputs $\bar{P}_{t-W+1:t}$ to form the input. The transformer model is trained to predict neural activity $\hat{X}^{(i)}_{t+1}$ at the masked step with a causal attention mask over the W-step context window.

Behavioral observations, such as pupil diameter, can serve as indirect readouts on the activity of unobserved neurons. Concatenating these observations with the aforementioned statistics will enrich the exogenous input observed by the dynamical model. We call this new concatenated variable P_t.

Center-surround partition: Synaptic connection probability between neurons depends on distance [Ref117, Ref118]. Similarly, neuronal co-variability correlates with spatial distance [Ref119]. To include this neurobiological insight while maintaining permutation-

invariance of our model, we propose a simple extension: we partition the population activity into two groups, center and surround, and compute the relevant invariant statistics for each group separately. Such partitioning is easy to obtain in calcium imaging experiments since the distances between the somata of neurons are readily available. Beyond its simplicity, this choice is also motivated by surround suppression being a connectivity motif in the brain [Ref122, Ref123].

Discrete-time dynamical model of neuronal activity: We re-write neuronal dynamics in discrete time as

$$X_{t+1}^{(i)} = f(X_{t-W+1:t}^{(i)}, (C_{t-W+1:t}^{(-i)}, S_{t-W+1:t}^{(-i)}, B_{t-W+1:t}), \Phi^{(i)}), \tag{4.2}$$

where $X \in \mathbb{R}^{N \times T}$ represents the activity data for the whole recorded population with N neurons and T time steps. C, S, and B represent D, D, and D'-dimensional permutation- and size-invariant (i.e., N) surrogates for brain activity at each time point. C and S compute identical statistics of population activity (here, mean and standard deviation) except that the statistics in C are calculated over the center partition (neurons whose distance to neuron i is at most Δ) and the statistics in S are calculated over the surround partition (neurons whose distance to neuron i is larger than Δ), see Appendix 4.5.5.1. Here, Δ is a hyperparameter. B denotes the contribution of time-resolved behavioral observations. Hence, the triplet (C_t, S_t, B_t) corresponds to $\bar{P}_t$. $\Phi^{(i)} \in \mathbb{R}^K$ denotes a K-dimensional time-invariant representation for neuron i, and W denotes the width of the available temporal context. The subscripts denote the limits of the time interval within which the corresponding variable is available to f.

Self-supervised Representation Learning Framework with Transformer: We thus propose to solve the following self-supervised optimization problem to infer both the function f and $\Phi^{(i)}$:

$$\arg \min_{f, \{\Phi^{(i)}\}_i} \sum_{i,t} \mathbb{E}_{X_{t+1}^{(i)}} ||X_{t+1}^{(i)} - f(X_{t-W+1:t}^{(i)}, (C_{t-W+1:t}^{(-i)}, S_{t-W+1:t}^{(-i)}, B_{t-W+1:t}), \Phi^{(i)})||, \tag{4.3}$$

where $||\cdot||$ denotes a norm. It is worth pointing out that f depends on the neuron of interest or other neurons in the population only through its explicit parameters. The reason for

this choice is to maintain the transferability and ubiquity of the learned model f, the first argument of the optimization, while summarizing neuronal variability with $\Phi^{(i)}$, the second argument.

The idea of inferring invariant representations for neurons directly from *in vivo* recordings by fitting a dynamical model (i.e., predicting activities in the next time step) is reminiscent of, and motivated by, the recent spectacular successes of "foundation models" in natural language modeling [Ref81], where capturing the dynamics of a complicated system produces an implicit understanding of the dynamics and identity of its components. This procedure has been shown useful for multiple downstream tasks [Ref81]. Therefore, we use a transformer model [Ref19] to parametrize the function f. Unlike previous uses of the transformers for neural data [Ref25, TL1], the input tokens in our model do not come from a countable set because raw calcium recordings are best represented by real valued signals.

To train the transformer and the time-invariant representation, we first generate masked activity inputs $\tilde{X}^{(i)}_{t+1}$, where neuronal activity at time $t+1$ is zero-out. This masked activity and the past activities $\tilde{X}^{(i)}_{t-W+1:t}$ are concatenated with the time-invariant representations $\Phi^{(i)}$ and permutation-invariant summary of population dynamics $\bar{P}_{t-W+1:t}$ to form the inputs to the transformer. We ask the transformer to predict the activity at the masked step, and compute the loss from the predicted activity $\hat{X}^{(i)}_{t+1}$ and ground truth activity $X^{(i)}_{t+1}$. The dynamical model f and time-invariant representation $\Phi^{(i)}$ are jointly learned during the optimization. To perform this task, the transformer has to learn the temporal progression of neuronal activity conditioned on the neuronal identity and the causal temporal context of individual and population statistics.

4.4 Results

4.4.1 A Multimodal Dataset

We use a recent, public multimodal dataset to train and demonstrate our model: Bugeon *et al.* [Ref7] obtained population activity recordings from the mouse primary visual cortex

Input		—	Individual without Population							Individual with Population		
Supervision		Lower Bound	Data-Limited Supervised			Unsupervised		Self-supervised	Data-Limited Supervised			Self-supervised
Task	Model	Random	LOLCAT	Trans +ISI	Trans +Raw	PCA	UMAP	NeuPRINT	LOLCAT	Trans +ISI	Trans +Raw	NeuPRINT
Subclass	KNN	0.260	—	—	—	0.263	0.281	0.415	—	—	—	**0.610**
	Linear	0.256	0.404	0.474	0.474	0.316	0.404	0.537	0.474	0.491	0.386	**0.683**
	MLP	0.302	—	0.561	0.439	0.330	0.340	0.512	—	0.526	0.386	**0.756**
Class	KNN	0.488	—	—	—	0.536	0.584	0.652	—	—	—	**0.711**
	Linear	0.526	0.600	0.664	0.669	0.544	0.576	0.697	0.608	0.680	0.608	**0.793**
	MLP	0.523	—	0.640	0.664	0.565	0.520	0.752	—	0.632	0.616	**0.807**

Table 4.1: Top-1 accuracy of transcriptomic label prediction based on representations learned by (i) our proposed self-supervised representation learning from neural dynamics framework NeuPRINT, (ii) the supervised learning method LOLCAT and its variants Transformer+ISI and Transformer+Raw, (iii) unsupervised baselines PCA and UMAP, (iv) random representations (to determine the chance-level). Note that this experiment corresponds to a data-limited regime due to limited labeled data. We performed classification using three classifiers (KNN, Linear, MLP) and two tasks: predicting the subclass from the set $\{\text{Lamp5}, \text{Pvalb}, \text{Vip}, \text{Sncg}, \text{Sst}\}$, and predicting the cell class from the set $\{\text{excitatory}, \text{inhibitory}\}$. We study the performance of different models using only individual neuronal activity vs adding population statistics as input.

(V1) via calcium imaging, followed by single-cell spatial transcriptomics of the tissue and registration of the two image sets to each other to identify the cells across the two experiments. 2-photon calcium imaging recordings were obtained with a temporal sampling frequency of 4.3Hz. And the spatial coordinates of recorded neurons are also provided. We first evaluate our approach on one animal (SB025) across 6 sessions. The recordings from this animal include 2481 neurons in total. We then extend our analysis on functional recordings from 4

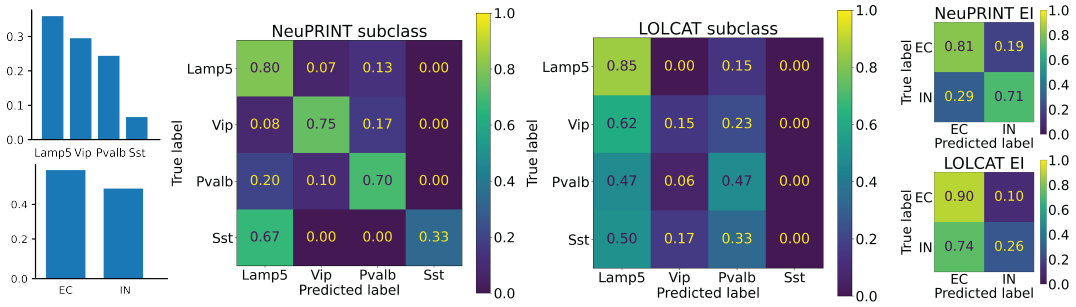

Figure 4.2: **Left:** Relative abundances of subclass and class labels. **Right:** Confusion matrices of our self-supervised representation learning framework NeuPRINT and supervised learning method LOLCAT based on predicting the cell class and subclass labels. While both of the self-supervised and supervised steps are learned with all available subclasses, we excluded the Sncg population from the confusion matrices because it represents a negligible fraction of the test set with the 80% : 10% : 10% split, so that quantification for this population would not be reliable.

mice (SB025, SB026, SB028, SB030) across 17 sessions. They contain 9728 neurons in total. Each session lasts about 20 minutes and records about 500 neurons. A small subset of neurons overlap across sessions. The subsequent transcriptomic experiment profiles mRNA expression for 72 selected genes in *ex vivo* tissue. These genes were used to identify the excitatory vs inhibitory class labels of neurons. In addition, 51% of the neurons in the inhibitory class of SB025 also have identified subclass labels (Lamp5, Pvalb, Vip, Sncg, Sst). Finally, the dataset includes behavioral information for the mice (running speed and pupil size) during the *in-vivo* recording as well as an assignment for each image frame (i.e., time point) that we call frame state from the set $\{\text{running}, \text{stationary desynchronized}, \text{stationary synchronized}\}$.

4.4.2 Benchmark Evaluation for Transcriptomic Identity Prediction

We use the aforementioned public dataset to introduce a new two-step benchmark: (i) self-supervised learning of time- and permutation-invariant representations for individual neurons

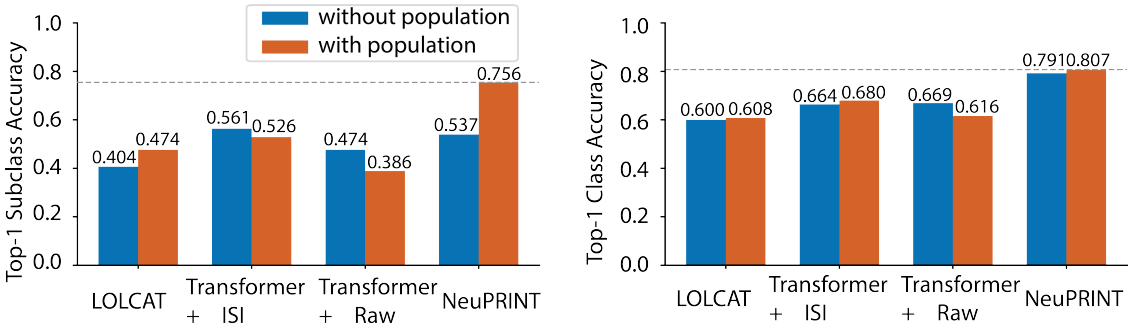

Figure 4.3: Accuracy of transcriptomic subclass and class prediction of NeuPRINT and baselines on single-mouse spontaneous activity recordings, with and without inputs from population statistics.

from population activity, (ii) prediction of labels for each individual neuron based on those representations.

We introduce a downstream classification task to predict the subclass label with supervised learning, where the neurons with subclass labels from all sessions are randomly split into train, validation and test neurons with a proportion of 80% : 10% : 10%. We further introduce another supervised downstream classification task to predict the class identity only (i.e., excitatory vs inhibitory). In this task, the validation and test neurons in the subclass prediction task are used as validation and test neurons for inhibitory neurons, and the same fraction of excitatory neurons are randomly selected as validation and test neurons. The rest of the recorded population from all sessions is used for training.

We first optimize the dynamical model (f in Eq. 4.2) and the time-invariant representation on the training set. We use past activities of the training neurons and permutation-invariant summary of population dynamics including pupil size, running speed, frame state, the mean and standard deviation of population activity, and the mean and standard deviation of center-surround activity to predict the individual neurons' activity in the next time step. After training, we fix the dynamical model f and only optimize the time-invariant representations Φ for training, validation, and test neurons under the same self-supervised learning framework.

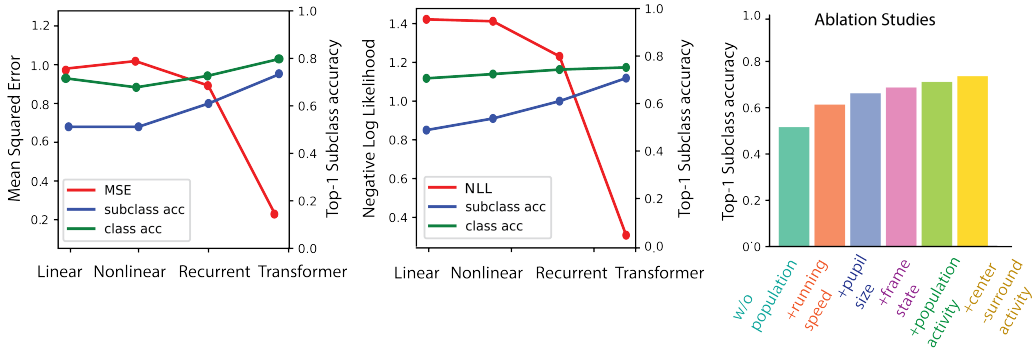

Figure 4.4: **Left:** Top-1 accuracy (or loss) of learned representations with different dynamical models (linear, nonlinear, recurrent, transformer) in the subclass prediction task. **Right:** Ablation studies to dissect the impact of the different components of the permutation-invariant summary of population dynamics including running speed, pupil size, frame state, population activity, center-surround activity in improving the accuracy, as in Table 4.5. One component is added at a time from left to right.

Following self-supervised optimization, in the second step, we evaluate the learned representation with two supervised downstream tasks (class and subclass prediction) and three simple classifiers including k-nearest neighbor (KNN), linear model, multi-layer perceptron (MLP) with one hidden layer. The training neurons' representations Φ are used to train the classifier, and validation neurons are used to tune the hyperparameters (learning rate, hidden dimensionality, number of epochs, etc.), and the top-1 accuracies of all models are reported on the test neurons (Table 4.1). See Appendix for an analysis of sensitivity.

Implementation of a spectrum of implicit dynamical models and downstream classifiers

We explore four different implicit (not mechanistic) dynamical models: linear, nonlinear, gated-recurrent network (GRU), and transformer with self-attention. We optimize the parameters of the dynamics f and the neuronal representation Φ using gradient descent for all models.

Linear model: We use a linear dynamical system where the activity at the last step is

predicted from a linear combination of the activities and statistics of the population activities, behavioral information from previous steps inside a temporal window and the time-invariant representation (which is repeated at each step). This corresponds to an autoregressive model with exogenous inputs, where statistics of the activities of other neurons and behavioral information constitute the exogenous input to the dynamics of the neuron of interest.

Nonlinear model: In addition to the linear model, a nonlinear activation was applied to the weighted activity, behavioral information, repeated neuronal representation at each step before the linear combination (i.e., nonlinear autoregressive model with exogenous inputs).

Recurrent network with gated units: The activity at the next step is predicted from the hidden state in addition to the activity at the current step and the repeated neuronal representation as the inputs.

Transformer: We implement a W-step causal attention mask such that the transformer predicts the activity of the neuron at the current time step based on the hidden states in the W previous time steps. The hidden state tasks the neuron activities, statistics of the population activities, behavioral information at that time step, and time-invariant representation repeated at each time step as inputs. We use the transformer encoder-only implementation from PyTorch [Ref124] with 2 attention heads.

Training details: For the objective function to predict the activity, we explore both mean squared error (MSE) and negative log likelihood (NLL) with a Gaussian distribution. To train the dynamical model and representation of neurons, we use a 64-dimensional embedding for the time-invariant representation. The temporal trial window size is 200 steps for the linear, nonlinear models, recurrent network and transformer. The batch size is 1024. We use the Adam optimizer [Ref125] with a learning rate of 10^{-3}.

Downstream supervised classification: For the linear and MLP classifiers, we use the cross-entropy loss to train the model. For KNN, we use the scikit-learn implementation [Ref126] with the number of nearest neighbors $k = 5$.

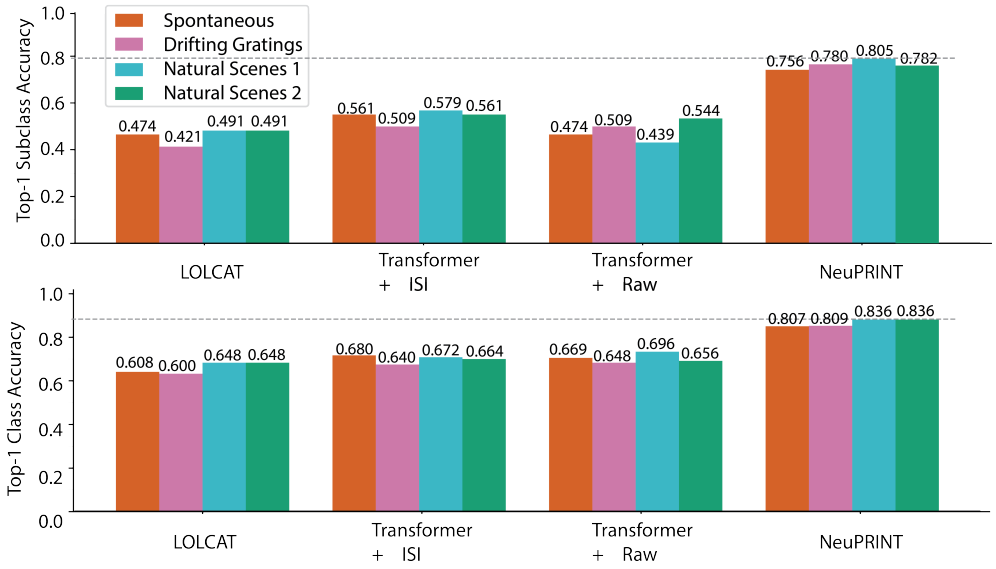

Figure 4.5: Accuracy of transcriptomic subclass and class prediction of NeuPRINT and baselines on single-mouse recordings during spontaneous activity and visual stimuli-driven (drifting gratings, natural scenes) activity.

4.4.3 Baselines

LOLCAT and its variants: LOLCAT [Ref127] is a supervised framework for predicting cell types from individual neuronal activities using a multi-head attention network. Since the attention in LOLCAT is a simple weighted sum operation, we implement two additional supervised variants Transformer+ISI and Transformer+Raw using the self-attention mechanism employed in NeuPRINT for a fair comparison. These two variants follow the attention design of [Ref19] and use a special classification token to represent the classification of the entire neuronal activity [Ref27]. Transformer+ISI operates on the inter-spike interval (ISI) distributions input summarized from non-overlapping sub-windows of continuous 2-photon calcium recordings. We use suite2p package [Ref128] to infer spikes from raw calcium traces and compute the ISI distributions. On the other hand, Transformer+Raw operates directly on the raw calcium traces. Unlike NeuPRINT, LOLCAT and the two supervised variants

train both the attention network and classifier (linear or MLP) in an end-to-end fashion, using neuronal class and subclass labels during learning (See Appendix for details). As a consequence of the supervised training scheme, LOLCAT does not extract time-invariant representations of neuronal identity and its performance is constrained by the number of labels available in the dataset.

Principal component analysis: We project the raw calcium activities to a low-dimensional representation using Principal Component Analysis (PCA) and evaluate the effectiveness of this representation for downstream tasks. The projection is performed on a randomly selected sub-window in the raw recordings of each neuron, therefore its projected representation is also not time-invariant.

Uniform manifold approximation and projection: Similar to PCA, we project the raw calcium activities to a low-dimensional representation using Uniform Manifold Approximation and Projection (UMAP) [Ref129], and evaluate the resulted time-variant representation by downstream classifiers.

Random: We further generate random representations for individual neurons, and train a supervised classifier on the random representations to measure the chance level of prediction.

4.4.4 Self-supervised Learning Demonstrates Superior Generalization Capabilities in Data-Limited Scenarios

We evaluate our proposed method NeuPRINT and other baselines under three categories as shown in Table 4.1: (i) time-invariant vs. time-variant; (ii) self-supervised representation learning vs. end-to-end supervised learning vs. unsupervised learning; (iii) activity of the neuron of interest ("Individual") as the only input to the dynamics model f vs. permutation-invariant representation of population dynamics provided as exogenous input to f. Under the data-limited scenario (i.e., a small amount of labeled samples), which describes a vast majority of neuroscience datasets, we find that our self-supervised representation learning model NeuPRINT with a supervised downstream MLP classifier outperforms the current state-of-the-art approach LOLCAT by $> 35\%$ and its variants by $> 19\%$ in the subclass

prediction task and outperforms LOLCAT by $> 20\%$ and its variants by $> 13\%$ in the class prediction task. Since LOLCAT is optimized using features extracted from non-overlapping sub-windows in an end-to-end supervised learning approach, it does not generate time-invariant representations that are critical to generalize across trials. Moreover, as shown in the confusion matrices in Figure 4.2, when the data in the subclass and class prediction tasks has an imbalanced distribution under the data-limited regime, our method can generate more balanced predictions across labels than LOLCAT. Both of these methods outperform two other standard unsupervised representation learning baselines, PCA and UMAP, which also extract time-varying representations across non-overlapping sub-windows. All of the evaluated models perform above the chance level, and we find that one-hidden layer MLP classifier improves classification accuracy over the linear classifier or KNN.

Ablation studies (Table 4.1) show that using a permutation-invariant summary of population dynamics is critical to improving the accuracy of the downstream subclass prediction task of our model. On the other hand, this effect is not significant in the class prediction task, suggesting that the intrinsic electrophysiology of neurons is significantly different between the excitatory vs inhibitory classes.

4.4.5 Learning Representations Across a Spectrum of Implicit Dynamical Models

We next investigate learning representations using our NeuPRINT framework over a spectrum of implicit dynamical models ranging from simple linear and nonlinear dynamical models to more advanced deep learning architectures such as gated recurrent networks and transformers. We also evaluate the performance of the models under two different objective functions (mean squared error vs Gaussian negative log likelihood). The results are shown in Figure 4.4. We find that the transformer, which leverages the powerful attention mechanism [Ref19] to preserve the information over a large temporal context, achieves the best performance in predicting the masked (future) neural activities, as quantified by NLL and MSE loss. This ability to capture the dynamics more faithfully explains the transformer's superior performance in inferring neuron identity since neuronal physiology correlates with molecular

expression [Ref115, Ref130]. We note that, for this relatively small dataset, the MSE loss performs better than the NLL loss based on the downstream classification accuracy.

4.4.6 Permutation-Invariant Summary of Population Dynamics Enhances the Time-invariant Representation of Individual Neurons

To further investigate the role of each component in the permutation-invariant summary of population dynamics, we perform a series of ablation studies as shown in Figure 4.4. We add each input (running speed, pupil size, frame state, population activity, and center-surround activity) to NeuPRINT one at a time. We find that all of the proposed components contribute to the success of the time-invariant representations as evaluated by the downstream transcriptomic classification task. These results support the perspective put forth in Section 4.3: how the neuron reacts to external inputs (hence the summary variables proposed here) forms a part of the neuron identity. Overall, we find using all of the available permutation-invariant components of the summary of the population recording improves the accuracy by 22% in subclass prediction.

4.4.7 Cell Type Identifiability Tends to Increase with Stimulus Relevance and Complexity

We further test our model NeuPRINT and other baselines on recordings with 3 sets of visual stimuli (drifting gratings and two different natural scene image sets [Ref7]). The results summarized in Fig. 4.5 suggest an increase in cell type identification accuracy of NeuPRINT from *in-vivo* activity as stimulus relevance and complexity increases, e.g., higher accuracy ($\sim 5\%$) in subclass prediction based on Natural Scenes 1 compared to spontaneous activity recording.

4.4.8 Extensions to Multiple Animals

We further extend our evaluations from one animal to multiple animals. We report the evaluations for all animals on the extended dataset in Table 4.2. While the performance

of LOLCAT increases on the class prediction task with this larger dataset, it still remains less accurate than our model across all tasks and classifiers. For subclass prediction, where the distribution of cells across subclasses is highly imbalanced, our method NeuPRINT outperforms LOLCAT by $\sim 23\%$, and also by $\sim 10\%$ in class prediction.

4.5 Appendix

4.5.1 Sensitivity Analysis

To quantify the sensitivity of our methods and baselines on the transcriptomic identity (neuron class and subclass) prediction benchmark, we run each model with 5 different random seeds (train/val/test random split, random initializations, etc.). We report their averaged Top-1 accuracies and the corresponding standard deviations across 5 runs for all models in Table 4.3.

Consistent with the main text, we find NeuPRINT still significantly outperforms other methods including supervised baselines LOLCAT [Ref127] and its variants Transformer+ISI and Transformer+Raw (Note the limited availability of labeled data). The standard deviations for all methods are relatively small, indicating the robustness of our evaluation framework and the significance of the performance gaps.

4.5.2 A Spectrum of Implicit Dynamical Models

We explore a spectrum of implicit dynamical models – Linear, Nonlinear, Recurrent neural network, Transformer, and two objective functions – mean squared error (MSE) when the model only predicts the mean of the output distribution; negative log likelihood (NLL) when the model predicts both mean and standard deviation of output distribution to train the models. The results are summarized in Table 4.4.

4.5.3 Roles of Permutation-Invariant Summary of Population Dynamics

To investigate the role of each component in the permutation-invariant summary of population dynamics, we perform a series of ablation studies as shown in Table 4.5. We add each input (running speed, pupil size, frame state, population activity, and center-surround activity) to NeuPRINT one at a time. We find that all of the proposed components contribute to the success of the time-invariant representations as evaluated by the downstream transcriptomic classification task.

4.5.4 Extensions to Other Visual Stimulus Conditions

We further test our model NeuPRINT and other baselines on recordings with 3 sets of visual stimuli (drifting gratings and two different natural scenes). The results in Table 4.6 suggest an increase in cell type identification accuracy from in-vivo activity as stimulus relevance and complexity increases.

4.5.5 Implementation Details

4.5.5.1 NeuPRINT

Permutation-invariant summary of population dynamics: We use C, S to represent D, D dimensional permutation- and size-invariant (i.e., N) surrogates for brain activity at each time point. C and S compute identical statistics (mean and standard deviation) of population activity except that the statistics in C are calculated over the center partition (neurons whose distance to neuron i is at most Δ) and the statistics in S are calculated over the surround partition (neurons whose distance to neuron i is larger than Δ). Here, Δ is a hyperparameter:

$$\mu_{C_t^{(-i)}} = \text{mean}(X_j(t) \,|\, j : 0 < d_{ji} < \Delta) \tag{4.4}$$

$$\sigma_{C_t^{(-i)}} = \text{std}(X_j(t) \,|\, j : 0 < d_{ji} < \Delta) \tag{4.5}$$

$$\mu_{S_t^{(-i)}} = \text{mean}(X_j(t) \,|\, j : 0 < \Delta \leq d_{ji}) \tag{4.6}$$

$$\sigma_{S_t^{(-i)}} = \text{std}(X_j(t) \,|\, j : 0 < \Delta \leq d_{ji}) \tag{4.7}$$

Batch sampling: For each batch of data we randomly sample 512 labeled and unlabeled neurons to be included in the batch. For each neuron we further randomly sample 2 sub-windows of 512 timesteps from its continuous calcium flourescene traces. Each resulting sample for the i^{th} neuron is denoted $X^{(i)}$.

Multihead attention: We use the transformer encoder architecture [Ref19] and the masking strategy as originally proposed in [Ref27]. Random timesteps in $X^{(i)}$ are masked (zero-out) with probability 0.25 and concatenated with the permutation-invariant population summary $\bar{P}$ and time-invariant representation $\phi^{(i)}$ along the feature dimension to form input $\bar{X}^{(i)}$. Note that the same learnable $\phi^{(i)}$ is repeated at every timestep, enforcing time-invariance. Similar to [Ref27], we embed input $\bar{X}^{(i)}$ and employ sinusoidal positional embedding to encode the temporal order in the input sequence, resulting in $\tilde{X}^{(i)} = \text{Emb}(\bar{X}^{(i)}) + \text{E}$.

For each input $\tilde{X}^{(i)}$, a set of weights $W^Q \in \mathbb{R}^{T \times d_q}$, $W^K \in \mathbb{R}^{T \times d_k}$, $W^V \in \mathbb{R}^{T \times d_v}$ are learned to transform input $\tilde{X}^{(i)}$ to a set of query, key, and value (Q, K, V), where $Q = \tilde{X}^{(i)} W^Q$, $K = \tilde{X}^{(i)} W^K$, $V = \tilde{X}^{(i)} W^V$. Attention between temporal tokens for one attention head is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V \tag{4.8}$$

Each head will find a different pattern in the data and produce an output of size d_v. The final attention output will be a concatenation of these single-head outputs. We use 2 heads in our model.

Feedforward layers and residual connections are subsequently applied to attention output:

$$Z^{(i)} = \tilde{X}^{(i)} + \text{MSA}(\tilde{X}^{(i)}) + \text{FF}(\tilde{X}^{(i)} + \text{MSA}(\tilde{X}^{(i)})) \tag{4.9}$$

where MSA represents the multihead attention operation, FF represents the feedforward layer with ReLU activation, and $Z^{(i)}$ represents the reconstructed calcium trace with masked

timesteps recovered.

Computational cost: NeuPRINT has 833K parameters and takes 2 hours in total for training and inference on a single NVIDIA Tesla V100 GPU. Details on the size of the datasets are mentioned in Section 4.4.1 of the main paper.

4.5.5.2 LOLCAT and Its Variants

We use the publicly available implementation of LOLCAT from [Ref127]. We implement two additional variants of LOLCAT (Transformer+ISI and Transformer+Raw), following the philosophy in [Ref127] but with the self-attention mechanism as used by NeuPRINT [Ref19]. For LOLCAT and Transformer+ISI, continuous time series of calcium traces are divided into non-overlapping sub-windows of size 64, within which the Inter-Spike Interval (ISI) distribution is computed. We use suite2p Python package [Ref128] to infer spikes and compute the ISI distribution with 16 bins, using a spike threshold of 0.2. To perform classification using the self-attention mechanism, we use a learnable special classification token [CLS] appended to the beginning of the input sequence to represent the classification output [Ref27]. No positional embedding is added to the input sequence. Therefore the transformer output at the CLS token position will represent the pooling operation as in [Ref127], and all sub-windows are treated as if they are independent trials, which enables Transformer+ISI to apply to any number of observed trials - an important design choice for LOLCAT.

We further implement Transformer+Raw, a variant of LOLCAT where the transformer operates directly on the raw calcium traces rather than the ISI distribution of calcium traces. Transformer+Raw follows the same architecture as Transformer+ISI, except that now the positional embedding is added to the input sequence to denote the temporal relationship between timesteps in the trial window.

4.5.5.3 Random / PCA / UMAP

To probe the chance-level classification performance, we train and evaluate downstream classifiers using random vectors of size 64 as representations for individual neurons, equivalent to the 64-dim $\phi^{(i)}$. To compare NeuPRINT with unsupervised methods PCA and UMAP, we first project the data to a lower dimensional space using 64 components, then train and evaluate downstream classifiers on the low-dimensional representation.

4.5.5.4 Downstream Classifiers

We use 5 nearest neighbors for KNN downstream classifier. For the downstream MLP classifier, we use a multi-layer perceptron network with a single hidden layer of size 2048 and ReLU activation.

4.5.5.5 Hyperparameters

We include all of the important hyperparameters (representation dim, window size, number of epochs, learning rate, batch size, etc.) for our NeuPRINT and other models (supervised-learning baselines LOLCAT, Transformer+ISI and Transformer+Raw, unsupervised representation learning baselines UMAP and PCA, chance-level prediction based on random features) in Table 4.7.

4.5.6 Pseudo Code

Our NeuPRINT framework includes three main components: an implicit dynamical system that uses the state-of-the-art transformer architecture to model neural dynamics; an optimization framework that fits the dynamical model and learns time-invariant representations for neurons; a supervised learning framework to train the downstream classifiers for subclass and class prediction, taking the learned time-invariant representations as inputs. The pseudo code for these three components is listed as follows:

```
Transformer(calcium_fluoroscence_trace,
            permutation_invariant_population_summary,
            time_invariant_representation):
        input_embedder = linear(input_dim, hidden_dim)
        transformer_encoder = multihead_attention(
                                        window_size,
                                        layer_dim,
                                        num_heads)
        output_decoder = linear(hidden_dim, output_dim
        masked_calcium_fluoroscence_trace =
                    mask_inputs(calcium_fluoroscence_trace)
        input = concatenate(
                masked_calcium_fluoroscence_trace,
                permutation_invariant_population_summary,
                time_invariant_representation)
        input = positional_encoding(input)
        input = input_embedder(input)
        context_mask = generate_context_mask(
                                    window_size,
                                    context_window_size)
        output = transformer_encoder(input, context_mask)
        output = output_decoder(output)
        return output

Time_Invariant_Self_Supervised_Representation_Learning(
                            calcium_fluoroscence_trace,
                            permutation_invariant_population_summary):

        time_invariant_representation = zeros(neuron_dim, embedding_dim)
```

```
        recon_model = Transformer(
                              input_dim,
                              hidden_dim,
                              window_size,
                              context_window_size,
                              layer_dim,
                              num_heads)
        optimizer = Adam(
                time_invariant_representation,
                recon_model,
                learning_rate)
        predicted_calcium_fluoroscence_trace = recon_model(
                calcium_fluoroscence_trace,
                permutation_invariant_population_summary,
                time_invariant_representation)
        recon_loss = mse(
                predicted_calcium_fluoroscence_trace[masked_steps],
                calcium_fluoroscence_trace[masked_steps])
        recon_loss.backward()
        optimizer.step()
        return time_invariant_representation


Downstream_Classifier_Supervised_Learning(
                              time_invariant_representation,
                              ground_truth_class_label):

        classifier = mlp(embedding_dim, hidden_dim, output_dim)
        optimizer = Adam(classifier, learning_rate)
```

```
predicted_class_label = classifier(time_invariant_representation)
classification_loss = cross_entropy(
                      predicted_class_label,
                      ground_truth_class_label)
classification_loss.backward()
optimizer.step()
return predicted_class_label
```

	Learned Representation	Time-Variant						Time-Invariant	
		Lower Bound	Data-Limited Supervised			Unsupervised		Self-supervised	
	Inputs	—	Individual						Individual + Population
Task	Classifier	Random	LOLCAT	Trans +ISI	Trans +Raw	PCA	UMAP	NeuPRINT w/o pop.	NeuPRINT (ours)
Subclass	KNN	0.174	—	—	—	0.333	0.348	0.419	**0.552**
	Linear	0.340	0.457	0.449	0.493	0.304	0.384	0.552	**0.590**
	MLP	0.362	—	0.442	0.423	0.384	0.406	0.552	**0.685**
Class	KNN	0.581	—	—	—	0.670	0.613	0.659	**0.700**
	Linear	0.667	0.700	0.710	0.675	0.645	0.660	**0.746**	**0.746**
	MLP	0.660	—	0.702	0.707	0.682	0.667	0.770	**0.800**

Table 4.2: **Extensions to multiple animals**: Top-1 accuracy of transcriptomic label prediction based on (i) the representations learned by our proposed self-supervised representation learning from neural dynamics framework NeuPRINT, (ii) the supervised learning method LOLCAT and its variants Transformer+ISI and Transformer+Raw, (iii) unsupervised baselines PCA and UMAP, (iv) random representations (to determine the chance-level). Note that this experiment corresponds to a data-limited regime due to limited labeled data. We performed classification using three classifiers (KNN, Linear, MLP) and two tasks: predicting the cell class from the set $\{\text{excitatory}, \text{inhibitory}\}$ and predicting the subclass from the set $\{\text{Lamp5}, \text{Pvalb}, \text{Vip}, \text{Sncg}, \text{Sst}\}$.

	Learned Representation	Time-Variant						Time-Invariant	
		Lower Bound	Data-Limited Supervised			Unsupervised		Self-supervised	
	Inputs	—	Individual						Individual + Population
Task	Classifier	Random	LOLCAT	Trans +ISI	Trans +Raw	PCA	UMAP	NeuPRINT w/o pop.	NeuPRINT (ours)
Subclass	KNN	0.196 ±0.041	—	—	—	0.345 ±0.047	0.371 ±0.070	0.424 ±0.051	**0.546 ±0.056**
	Linear	0.326 ±0.055	0.449 ±0.027	0.414 ±0.036	0.436 ±0.021	0.316 ±0.046	0.386 ±0.046	0.595 ±0.041	**0.683 ±0.042**
	MLP	0.382 ±0.062	—	0.456 ±0.068	0.422 ±0.017	0.392 ±0.044	0.496 ±0.054	0.606 ±0.054	**0.722 ±0.022**
Class	KNN	0.509 ±0.061	—	—	—	0.569 ±0.015	0.547 ±0.046	0.568 ±0.057	**0.705 ±0.017**
	Linear	0.536 ±0.035	0.616 ±0.017	0.632 ±0.026	0.652 ±0.009	0.584 ±0.053	0.583 ±0.019	0.741 ±0.035	**0.766 ±0.031**
	MLP	0.565 ±0.030	—	0.624 ±0.010	0.651 ±0.007	0.596 ±0.015	0.584 ±0.013	0.782 ±0.018	**0.803 ±0.037**

Table 4.3: **Sensitivity analysis across 5 runs with different random seeds**: Top-1 accuracy (**mean±standard deviation**) of transcriptomic label prediction based on (i) the representations learned by our proposed self-supervised representation learning from neural dynamics framework NeuPRINT, (ii) the supervised learning method LOLCAT and its variants Transformer+ISI and Transformer+Raw (abbreviated Trans+ISI and Trans+Raw, respectively) (Note that this experiment corresponds to a data-limited regime due to the size of the dataset), (iii) unsupervised baselines PCA and UMAP, (iv) random representations (to determine the chance-level). We performed classification using three different simple classifiers (KNN, Linear, MLP) and two tasks: predicting the cell class from the set $\{\text{excitatory}, \text{inhibitory}\}$ and predicting the subclass from the set $\{\text{Lamp5}, \text{Pvalb}, \text{Vip}, \text{Sncg}, \text{Sst}\}$.

Task	Linear		Nonlinear		Recurrent		Transformer	
Objective	NLL	MSE	NLL	MSE	NLL	MSE	NLL	MSE
Loss	1.422	0.984	1.412	1.022	1.231	0.887	0.308	**0.221**
Subclass	0.488	0.512	0.537	0.512	0.610	0.610	0.707	**0.756**
Class	0.706	0.716	0.724	0.679	0.743	0.743	0.752	**0.807**

Table 4.4: Comparison of reconstruction loss (first row, the smaller the better) and top-1 accuracy of class and subclass prediction using MLP downstream classifier (second and third rows, the larger the better) achieved by different dynamics models f (linear, nonlinear, RNN, and transformer) in NeuPRINT. Two objectives are used to train each model (mean squared error (MSE) or negative log likelihood (NLL)).

Subclass	w/o population inputs	+ running speed	+ pupil size	+ frame state	+ population activity	+ center-surround activity
MLP	0.512	0.609	0.658	0.683	0.732	**0.756**

Table 4.5: Ablation studies of permutation-invariant inputs representing population activity, including running speed, pupil size, frame state, permutation-invariant population representation, permutation-invariant center-surround representation. Results are reported on the subclass prediction task with an MLP downstream classifier.

	Task	Classifier	LOLCAT	Transformer+ISI	Transformer+Raw	NeuPRINT
Spontaneous Activity	Subclass	linear	0.474	0.491	0.474	**0.683**
		mlp	—	0.561	0.439	**0.756**
	Class	linear	0.608	0.680	0.669	**0.793**
		mlp	—	0.640	0.664	**0.807**
Drifting Gratings	Subclass	linear	0.421	0.509	0.404	**0.756**
		mlp	—	0.474	0.509	**0.780**
	Class	linear	0.600	0.640	0.648	**0.809**
		mlp	—	0.632	0.640	**0.809**
Natural Scenes 1	Subclass	linear	0.491	0.579	0.421	**0.780**
		mlp	—	0.544	0.439	**0.805**
	Class	linear	0.648	0.672	0.696	**0.809**
		mlp	—	0.664	0.640	**0.836**
Natural Scenes 2	Subclass	linear	0.491	0.509	0.544	**0.732**
		mlp	—	0.561	0.526	**0.782**
	Class	linear	0.648	0.664	0.656	**0.809**
		mlp	—	0.664	0.648	**0.836**

Table 4.6: Accuracy of transcriptomic subclass and class prediction of NeuPRINT and baselines on single-mouse recordings during spontaneous activity and visual stimuli-driven (drifting gratings, natural scenes) activity.

	NeuPRINT	LOLCAT	Transformer +ISI	Transformer +Raw	random/ PCA/UMAP
Representation dim	64	—	—	—	64
Encoder hidden dim	70	$[32, 16, 16]$	$[128] \times 4$	$[128] \times 4$	—
MLP classifier hidden dim	2048	—	2048	2048	2048
Number of attention heads	2	4	4	4	—
Number of attention layers	1	1	1	1	—
Window size	200	2048	512	512	2048
Context window size	2	—	8	1	—
Batch size	1024	varies	varies	varies	—
Number of epochs	400	500	500	500	—
Number of downstream epochs	5000	—	—	—	1000
Learning rate	10^{-3}	—	—	—	—
Downstream learning rate	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-3}
Dropout	0.1	—	0.1	0.1	—
KNN neighbors	5	—	5	5	5
ISI sub-window size	—	64	64	—	—
Number of ISI bins	—	16	16	—	—

Table 4.7: **Hyperparameters** of our self-supervised representation framework NeuPRINT and other baselines including end-to-end supervised learning model LOLCAT, its variants Transformer+ISI and Transformer+Raw, unsupervised representation learning models PCA and UMAP, and chance-level prediction based on random features.

Chapter 5

LEARNING SPATIAL PERMUTATION-INVARIANT REPRESENTATIONS FOR CROSS-SESSION DECODING GENERALIZATION

This chapter contains material that was previously published in [TL3].

5.1 Background

Motor behavior arises from the complex interplay between interconnected neurons, each possessing distinct functional properties [Ref3]. Deciphering the highly nonlinear mapping from the activity of these neural populations to behavior has been a major focus of intracortical Brain-Computer Interfaces (iBCI), whose applications have enabled individuals with motor impairments to control external devices [Ref30], restore communication abilities through typing [Ref31], handwriting [Ref23], and speech [Ref24].

Despite the remarkable capabilities, iBCI systems suffer from performance degradation over extended periods of time, largely attributed to the nonstationarities of the recorded populations [Ref131]. Sources of nonstationarities include shifts in electrode position, tissue impedance changes, and neural plasticity [Ref60, Ref132, Ref133]. These nonstationarities lead to changes in the number and identity of neural units picked up by recording electrodes over time. Such changes in population composition alter the learned neural activity to behavior mapping, preventing decoders trained on previous sessions to maintain robust performance on new sessions. To ensure robustness of behavior decoding over future recording sessions, one approach has focused on training deep networks using many sessions, attempting to achieve decoders that are robust to the cross-session variability [Ref134, Ref135]. This zero-shot approach requires months of labeled training data, necessitating extensive data

collection from the user. While recent methods targeting cross-subject generalization may alleviate some of this burden [Ref136, Ref66, Ref137, Ref26], degradation over long-term use still remains, advocating for the adoption of adaptive methods [Ref138, Ref68, Ref67, Ref60, Ref71]. These adaptive methods leverage the low-dimensional manifold underlying population activity that has been shown to preserve a consistent relationship with behavior over long periods of time [Ref20, Ref60]. Depending on the use of labels at test time, they can be categorized into supervised [Ref137, Ref139, Ref26], semi-supervised [Ref72], or unsupervised [Ref71, Ref68, Ref70], with varying level of success and practical utility in real-world iBCI [Ref138].

Despite the variety of technical approaches, these works share a common design philosophy: they adopt a fixed view of the neural population, assigning fixed identities and order for neural units during training. While this treatment achieves high decoding performance on held-in sessions (within-session generalization), the decoders suffer from out-of-distribution performance degradation when evaluated on held-out sessions with different sizes and unit membership (cross-session generalization). To enable transfer of the pretrained model to novel sessions, explicit alignment procedures with gradient updates to adapt model parameters are necessary, imposing disruptive and costly computation overhead for iBCI users. With these limitations of the existing approaches, we advocate for the view that an ideal, universal iBCI decoder should be invariant to the permutation of the neural population by design, and should be able to seamlessly handle inference of a variable-sized, unordered set of neural units with minimal data collected from the new setting.

In this work, we introduce SPINT - a permutation-invariant framework that can decode motor behavior from the activity of unordered sets of neural units. We contribute toward an iBCI design that can predict behavior covariates from continuous streams of neural observations and adapt gradient-free to novel sessions with few-shot unlabeled calibration data. At the core of our methods is a permutation-invariant transformer with a novel context-dependent positional embedding that allows flexible identification of neural unit identities on-the-fly. We further introduce *dynamic channel dropout*, a novel regularization method to

encourage model robustness to neural population composition. We evaluate our approach on three movement decoding datasets from the FALCON Benchmark [Ref138], demonstrating robust cross-session generalization on motor tasks in human and non-human primates. Our model outperforms zero-shot and few-shot unsupervised baselines, while not requiring any retraining or fine-tuning overhead.

5.2 Contributions

In summary, the contributions of this work include:

- We present a *transformer-based permutation-invariant framework* with a novel *context-dependent positional embedding* for few-shot unsupervised behavioral decoding. Our flexible, lightweight model enables ingestion of unordered sets of neural units during training and facilitates out-of-the-box inference on unseen neural populations.

- We introduce *dynamic channel dropout*, a novel regularization technique for iBCI applications to promote decoder robustness to the composition of input neural population.

- We evaluate our model on three motor behavioral decoding datasets in the FALCON Benchmark, showing robust gradient-free generalization to unseen sessions in the presence of cross-session nonstationarities.

5.3 Methods

5.3.1 A permutation-invariant framework for few-shot continuous behavioral decoding

We study the problem of real-time, cross-session iBCI decoding, where behavior needs to be decoded in a causal manner from a continuous stream of neural observations. Concretely, within a single session s, let $X_{i,t}$ denote the binned spiking activity of neural unit i at time t, $X_{i,:}$ denote all the activity of unit i, and $X_{:,t}$ denote the activities of all units at time t. Given a past observation window of population activity $X_{:,t-W+1:t} \in \mathbb{R}^{N_s \times W}$, where N_s is the

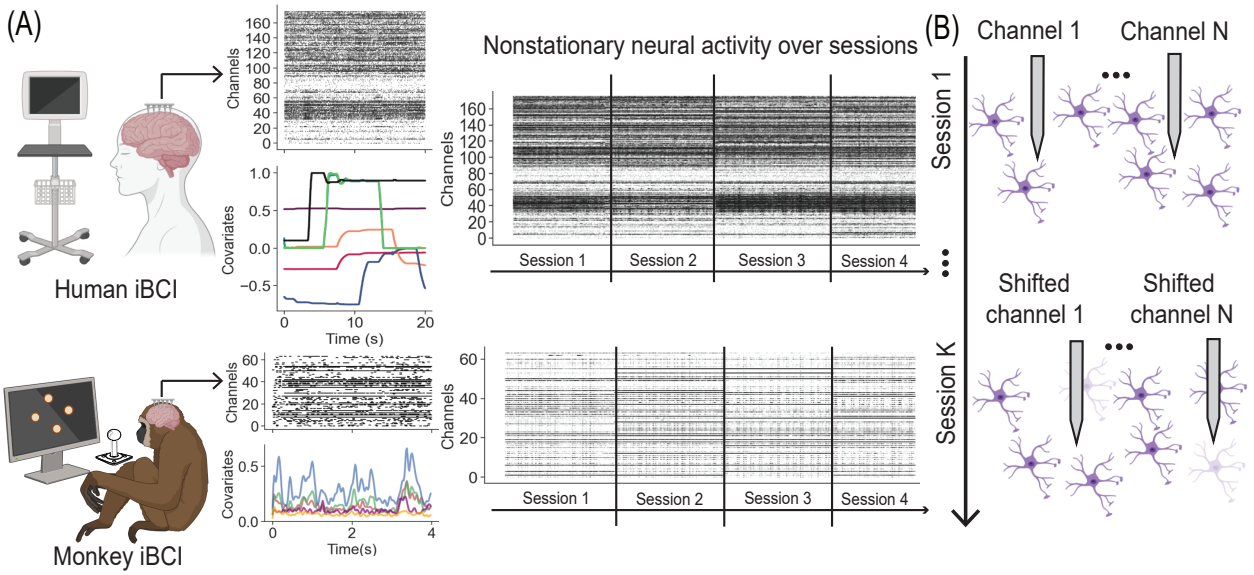

Figure 5.1: **Nonstationarities in long-term iBCI.** (A) Examples of iBCI systems in human and non-human primates. Spiking activity is recorded from multichannel electrode arrays together with behavior covariates, e.g., 7 degree-of-freedom robotic arm control or electromyography from the upper limb. Neural activity exhibits nonstationarities over recording sessions. (B) Systematic changes in neuron positions, including the introduction or loss of neurons in the vicinity of electrodes and the shifts of the entire electrode array can contribute to instability of neural recordings over time. This figure uses templates created with BioRender.com.

number of recorded neural units in session s and W is the length of the observation window, we aim to estimate the corresponding last time step of behavior output $Y_t \in \mathbb{R}^B$, where B is the dimensionality of behavior covariates. Model parameters are fitted with gradient descent using labeled data from k training (held-in) sessions and evaluated on k' testing (held-out) sessions without gradient updates or labels. At our disposal on each held-out session is a short calibration period $X_{i,[C]} \in \mathbb{R}^{T'}$ consisting of M-shot variable-length trials lasting for T' timesteps, to be used for cross-session adaptation.

Traditional approaches consider the population activity at each timestep as a "token" and

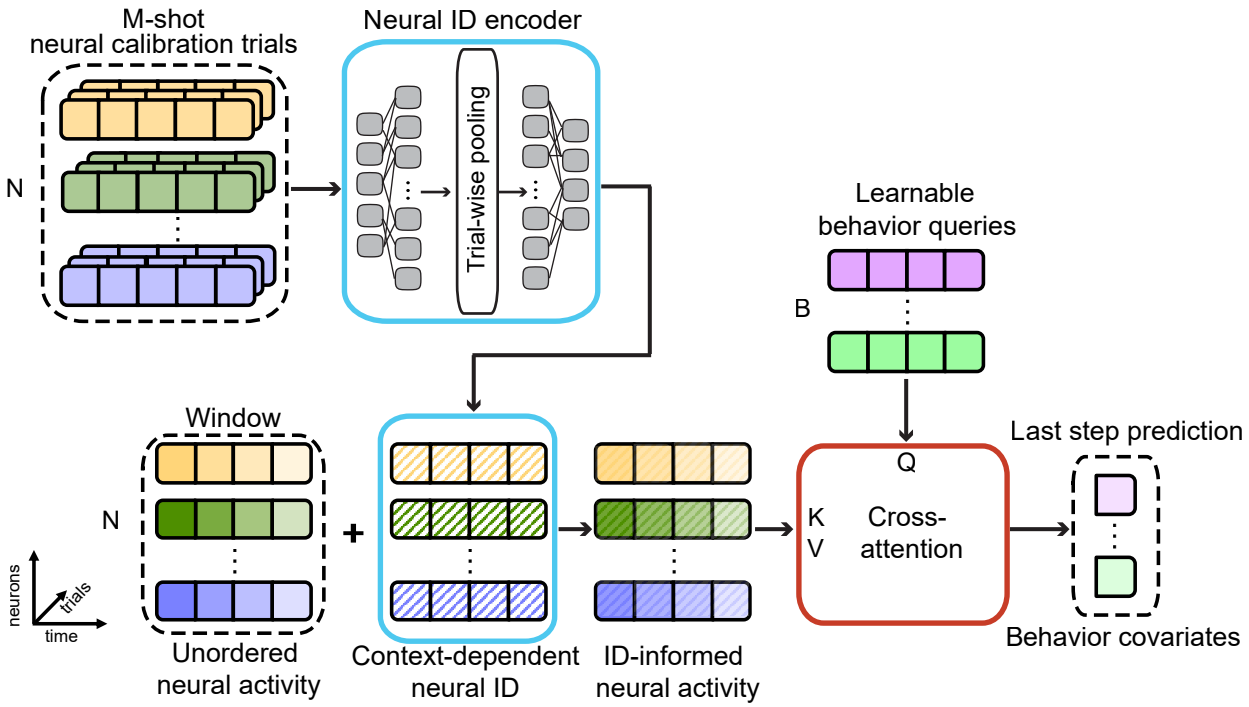

Figure 5.2: **SPINT architecture.** The model performs continuous behavioral decoding by predicting behavior covariates at the last timestep given a past window of activity from an unordered set of neural units. The universal Neural ID Encoder infers identities of the units using few-shot unlabeled calibration trials, while the cross-attention mechanism selectively aggregates information from the units to decode behavior.

decode behavior by modeling temporal dynamics of population activity [Ref15, Ref20, Ref25]. By treating temporal snapshots of population activity $X_{:,t} \in \mathbb{R}^{N_s}$ as input vectors, these approaches assume a fixed number and order of neural units, requiring explicit spatial re-alignment when applied to another session with a different size and order [Ref20, Ref71, Ref70, Ref66, Ref68]. Recent methods incorporating factorized spatial-temporal modeling [TL1, Ref140] face similar challenges, while approaches with explicit spatiotemporal tokens [Ref137, Ref139, Ref26] still require fine-tuning unit identity in novel sessions. These design choices hinder out-of-the-box generalizability of neural decoders across sessions, as a universal

decoder should ideally be invariant to the permutation and size of the input population.

To realize this goal, we treat windows of individual neural units $X_{i,t-W+1:t} \in \mathbb{R}^W$ as an *unordered set* of tokens and aggregate information from these units to decode behavior using the cross-attention mechanism [Ref19]. To compensate for the loss of consistent order that the decoder can leverage for behavior decoding, we embed a notion of neural identity to each unit based on its spiking signature during a few-shot, *unlabeled* calibration period $X_{:,[C]}$ in the same session. $X_{:,[C]}$ is either provided in limited amount in held-out sessions at test time, or is artificially sampled from held-in sessions during training. This context-dependent neural identity is inferred by a universal neural identity encoder that is shared across units and sessions, enabling *gradient-free* adaptation to novel population compositions at test time.

5.3.2 Encoding identity of neural units

Let $X_i^C \in \mathbb{R}^{M \times T}$ be the trialized version of $X_{i,[C]} \in \mathbb{R}^{T'}$. X_i^C is the collection of M calibration trials of neural unit i interpolated to a fixed length T. We infer neural identity $E_i \in \mathbb{R}^W$ of unit i by a neural network IDEncoder:

$$E_i = \text{IDEncoder}(X_i^C) = \psi(pool(\phi(X_i^C))) \tag{5.1}$$

where ψ and ϕ are multi-layer fully connected networks and *pool* is the mean pooling operation across M trials. Due to the permutation invariant nature of the mean operation and the fact that ψ and ϕ are applied trial-wise, IDEncoder is invariant to the order of M calibration trials by design [Ref85].

5.3.3 Decoding behavior via selective aggregation of information from neural population units

After inferring the identity for each unit from its calibration period, we add E_i to all X_i windows to form identity-aware representations Z_i. Z_i contains the time-varying activity of each unit while also being informed of the unit's stable identity within one session. In matrix

form:

$$Z = X + E \tag{5.2}$$

where Z_i's, X_i's, E_i's constitute rows of the Z, X, E matrices.

We leverage the cross-attention mechanism to selectively aggregate information from population units and decode behavior outputs:

$$Y = \text{CrossAttn}(Q, Z, Z) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{5.3}$$

where $K = ZW_K, V = ZW_V \in \mathrm{R}^{N_s \times W}$ are projections of the identity-informed neural activity Z, and $Q \in \mathrm{R}^{B \times W}$ is a learnable matrix to query the behavior from Z. We use the standard cross-attention module with pre-normalization and feedforward layers. Cross-attention with identity-informed neural activity (Equation 5.3) is invariant to the permutation of neural units, i.e.,

$$\text{CrossAttn}(Q, Z, Z) = \text{CrossAttn}(Q, P_R Z, P_R Z), \tag{5.4}$$

where P_R is the row permutation matrix. (See proof in Appendix). E in Equations 5.2 and 5.3 can be understood as a special kind of positional embedding for attention mechanism, where E is equivariant to the order of tokens (neural units), i.e., permuting the rows in X also permutes the rows in E accordingly. Hence, unlike traditional positional embeddings in the transformer literature where positional embeddings are fixed entities, our proposed E is *context dependent*. This context-dependent positional embedding enables cross-session generalization by design, as E is stable for all samples within the same context (session), and can readily adapt in a gradient-free manner to new populations with arbitrary size and order.

After cross-attention, we project down Y by a fully connected layer to a one-dimensional vector representing the predicted behavior covariates at the last timestep, based on which we compute the mean squared error (MSE) between the predicted and the ground truth behavior covariates. The IDEncoder and cross-attention module are trained in an end-to-end manner using this MSE objective.

5.3.4 Encouraging model robustness to inconsistent population composition

Neural population distributes its computation among many neural units, allowing us to effectively decode behavior even though we can only record neural activity with a limited number of electrodes. Leveraging this insight and in order to encourage model robustness to different compositions of neural membership across recording sessions, we employ *dynamic channel dropout*, a novel technique to avoid overfitting to the population composition seen during training. Unlike classical population dropout methods [Ref141, Ref142, Ref25] where only a fixed fraction of neurons/timesteps is zeroed-out during training, we randomly sample a dropout rate between 0 and 1 each training iteration and remove population units with the sampled dropout rate. With dynamic channel dropout, we not only encourage the model to be robust to the unit membership but also encourage it to be robust to the size of the population, leading to improved cross-session generalization (see Section 5.4.8 and Figure 5.4).

5.3.5 Gradient-free, few-shot cross-session adaptation in unseen neural populations with variable size and order

The overall framework is depicted in Figure 5.2. We use labeled data from all training sessions to train all model parameters following the above pipeline. The model naturally digests populations of arbitrary size and order in all sessions without any need of session-specific alignment layer or fixed positional embeddings for neural units, hence having the potential to scale up to a large amount of data. When testing on a held-out session, we reuse the trained IDEncoder and only need a few *unlabeled* calibration trials to infer identities of neural units in the test session, without the need of gradient descent updates to fine-tune session-specific alignment layers or unit/session embeddings. With these benefits, our proposed model removes the time and computation overhead usually required for re-calibrating neural decoders before each session, and facilitates its applicability in real-world iBCI settings where test-time labels are inherently unavailable.

5.4 Results

5.4.1 Datasets and evaluation metrics

We evaluate our approach on three continuous motor decoding tasks from the Few-shot Algorithms for Consistent Neural Decoding (FALCON) Benchmark [Ref138]. Specifically, we evaluate SPINT on the M1, M2, and H1 datasets. In M1, a monkey reached to, grasped, and manipulated an object in a variety of locations (4 possible objects, 8 locations), while neural activity was recorded from precentral gyrus and intramuscular electromyography (EMG) was recorded from 16 muscles [Ref143, Ref144, Ref145, Ref146]. In M2, a monkey made finger movements to control a virtual hand and acquired cued target positions while neural activity from the precentral gyrus and 2-D actuator velocities were captured [Ref147]. In H1, a human subject attempted to reach and grasp with their right hand according to a cued motion for a 7-degree-of-freedom robotic arm control [Ref30, Ref148, Ref149]. Each dataset comprises multiple labeled held-in sessions used to train the decoder (spanning 4, 4, and 6 days for M1, M2, and H1, respectively), and multiple held-out sessions for model evaluation (spanning 3, 4, and 7 days for M1, M2, and H1, respectively). Each held-out session provides a few public calibration trials (with optional labels) used for decoder calibration, after which the decoder is evaluated on a private test split. Cross-session performance is quantified by the mean and standard deviation of R^2 between the predicted and ground truth behavior covariates across all held-out sessions. All evaluation results were obtained on the held-out private split by submitting models to the EvalAI platform [Ref150].

5.4.2 Baselines

We compare SPINT with zero-shot (ZS) and few-shot unsupervised (FSU) baselines, since SPINT is the intersection of these two approaches. Similar to FSU approaches, SPINT makes use of a few unlabeled calibration samples in the held-out sessions; however, unlike conventional FSU approaches, SPINT does not require gradient updates for model parameters at test time, therefore bearing resemblance to ZS methods in terms of practical utility. We

call this new class of model *gradient-free few-shot unsupervised* (GF-FSU).

ZS Wiener Filter and ZS RNN: Wiener Filter is a linear model that predicts the current behavior as a weighted sum of previous timesteps [Ref151]. In addition to the Wiener Filter, we also compare with a simple RNN baseline (implemented as an LSTM [Ref17]). The WF and RNN models were fitted using a single held-in session and evaluated zero-shot on the held-out sessions.

CycleGAN [Ref70]: An FSU method where a Generative Adversarial Network (GAN) is trained using calibration data from a held-out session (day K) to transform day K's population activity to a form resembling activity from a held-in session (day 0), allowing decoders pretrained on day 0 to be reused on day K.

NoMAD [Ref71]: Another FSU method where a dynamical model and a decoder are trained on day 0 to predict behavior from the inferred dynamics. Then on day K, an alignment network is trained to match the distribution of neural latent states to that of day 0, allowing the fixed model and decoder to transfer to day K.

Wiener Filter, RNN and Transformer Oracles (OR): We include the Wiener Filter, RNN, and NDT2 - a transformer for neural data [Ref137], trained on private held-out labeled data to serve as upper bounds for model performance.

NDT2 Multi (FSS) [Ref137]: Similar to NDT2 Multi OR, but only trained on held-in and held-out few-shot calibration data with supervision.

5.4.3 SPINT outperforms zero-shot and few-shot unsupervised baselines on continuous motor decoding tasks

We show in Table 5.1 the performance of SPINT in comparison with ZS and FSU approaches. SPINT outperforms all ZS and FSU baselines across all three datasets, while requiring no retraining or fine-tuning of model parameters. Improvement is most prominent in M1, where the amount of training data is the largest ($\sim$5$\times$ more data than H1 and $\sim$6$\times$ more data than M2 in terms of recording time). Notably, SPINT surpasses Wiener Filter oracles in all datasets, which were trained with access to the private labeled data. SPINT even outperforms

	Class	M1	M2	H1
Wiener Filter (WF)	OR	0.53 ± 0.04	0.26 ± 0.03	0.21 ± 0.04
RNN	OR	0.75 ± 0.05	0.56 ± 0.04	0.44 ± 0.13
NDT2 Multi [Ref137]	OR	0.78 ± 0.04	0.58 ± 0.04	0.63 ± 0.08
NDT2 Multi [Ref137]	FSS	0.59 ± 0.07	0.43 ± 0.08	0.52 ± 0.04
WF	ZS	0.34 ± 0.06	0.06 ± 0.04	0.16 ± 0.03
RNN	ZS	-0.60 ± 0.45	-0.07 ± 0.23	0.09 ± 0.18
CycleGAN + WF [Ref70]	FSU	0.43 ± 0.04	0.22 ± 0.06	0.12 ± 0.06
NoMAD + WF [Ref71]	FSU	0.49 ± 0.03	0.20 ± 0.10	0.13 ± 0.10
SPINT (Ours)	GF-FSU	$\mathbf{0.66} \pm 0.07$	$\mathbf{0.26} \pm 0.13$	$\mathbf{0.29} \pm 0.15$

Table 5.1: Performance comparison against oracles (OR), few-shot supervised (FSS), few-shot unsupervised (FSU), and zero-shot (ZS) methods. Our SPINT approach belongs to a special class which we termed gradient-free few-shot unsupervised (GF-FSU), where models perform adaptation based on few-shot unlabeled data but without any parameter updates at test time. Results are reported as mean ± standard deviation R^2 across held-out sessions.

the FSS method NDT2 Multi on M1 dataset while unlike NDT2, it does not require access to test-time labels or model parameter updates. As we focus on cross-session transferability, all our experimental results show the cross-session performance. We include comparison on within-session performance in the Appendix.

5.4.4 SPINT requires only a minimal amount of unlabeled data for adaptation

To gauge the data efficiency of our model at test time, we trained and tested the model with varying number of calibration trials used to infer the neural unit IDs. We show in Figure 5.3(A) that SPINT could achieve reasonable cross-session generalization with a small number of few-shot trials. In M1 dataset, the model could even achieve similar performance as the

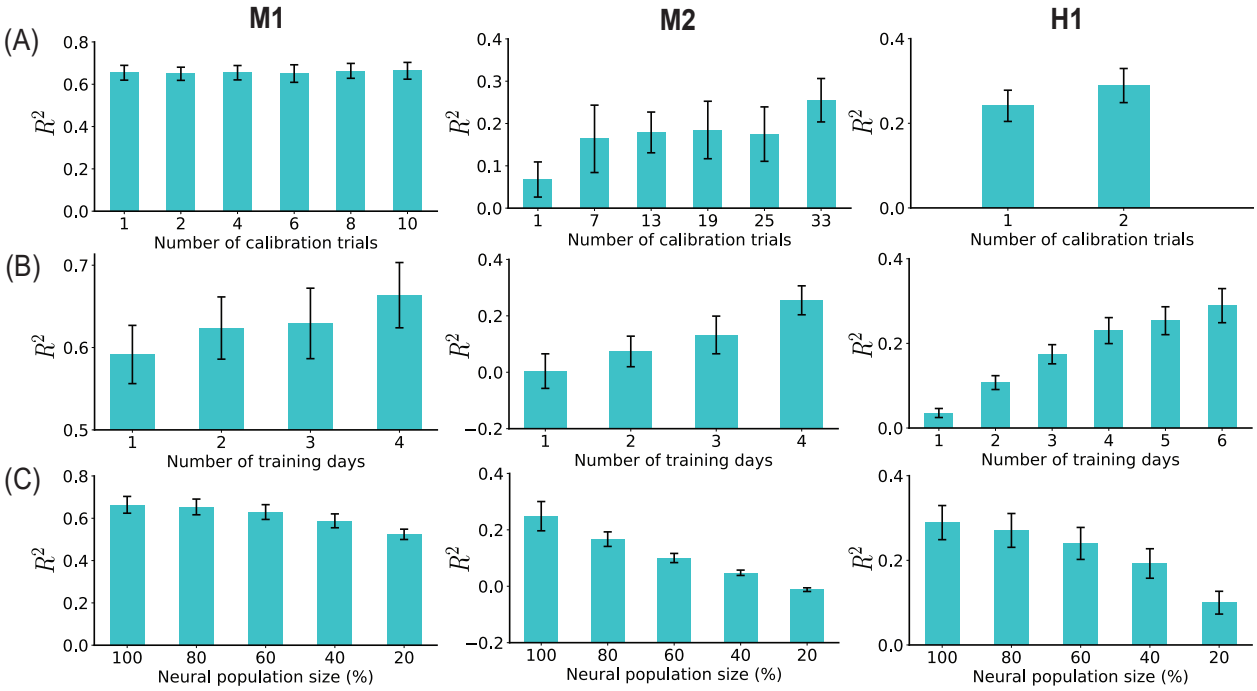

Figure 5.3: **Scaling analyses**. Cross-session performance of SPINT against number of calibration trials (A), training days (B), and population sizes (C) across M1, M2, and H1 datasets. Bars represent mean R^2 across held-out sessions, whiskers represent standard error of the mean of R^2 across held-out sessions.

best model (which uses all available calibration trials) with only one single trial. This study demonstrates the practical utility of SPINT in online iBCI, relieving the burden of data collection and label collection on users at test time.

5.4.5 SPINT performance scales well with the amount of training data

Thanks to the flexible permutation-invariant transformer network and the context-dependent positional embeddings, SPINT can ingest populations with arbitrary sizes and orders. These design choices give SPINT the ability to scale naturally with large amounts of training data. We demonstrate this scaling ability in Figure 5.3(B), where we observe a clear trend in cross-session performance when we use data from more held-in days to train the model, with

the best performance achieved when using all available training data on each dataset. This ability suggests potentials of SPINT as a large-scale pretrained model for iBCI when trained on larger datasets beyond the FALCON benchmark.

5.4.6 SPINT is robust to variable population composition

Our proposed dynamic channel dropout encourages robustness of SPINT to variable input population size and membership. We test this robustness by training SPINT on the full held-in populations with dynamic channel dropout and evaluating on variable-sized held-out populations (Figure 5.3C). At each evaluation batch, we randomly sample a subset of the original population and measure the R^2 obtained when the model makes predictions based on this limited subset. We observe robust performance in M1 with reasonable performance drop when the population gets increasingly smaller, with the model still achieving a mean held-out R^2 of 0.52 when only 20% of the original population remains, outperforming other ZS and FSU baselines with the full population.

5.4.7 SPINT maintains low-latency inference for iBCI systems

A critical consideration in iBCI system deployment is the ability of the system to perform behavior decoding in real time. We designed SPINT with this consideration in mind, using only one layer of cross-attention and two three-layer fully connected networks for IDEncoder. In Table 5.2, we report the latency achieved by SPINT as compared to other methods. Latency is defined as the amount of time a method requires to process the evaluation data divided by the duration of the evaluation data [Ref150]. The ratio less than 1 signifies the approximation to real-time iBCI inference. SPINT achieves 0.13 latency on M1 and M2, and 0.14 latency on H1, matching or outperforming transformer baselines, while being significantly below 1. In practice, SPINT could be potentially faster in terms of deployment time, as it eliminates the need for an explicit alignment step required by conventional iBCI systems.

	Class	M1	M2	H1
Wiener Filter (WF)	OR	0.06	0.08	0.14
RNN	OR	0.04	0.04	0.08
NDT2 Multi	OR	0.15	0.10	2.29
NDT2 Multi	FSS	0.13	0.10	0.30
WF	ZS	0.06	0.08	0.15
RNN	ZS	0.03	0.01	0.02
CycleGAN + WF	FSU	0.07	0.09	0.16
NoMAD + WF	FSU	0.99	0.91	1.03
SPINT (Ours)	GF-FSU	0.13	0.13	0.14

Table 5.2: Inference latency of SPINT against oracles (OR), few-shot supervised (FSS), few-shot unsupervised (FSU) and zero-shot (ZS) methods on held-out sessions (lower is better).

5.4.8 Ablation Study

We perform ablation studies to demonstrate the benefits of our context-dependent positional embeddings and dynamic channel dropout techniques. In Figure 5.4A, we compare our context-dependent positional embeddings with fixed (absolute) positional embeddings used in the vanilla transformer [Ref19], and with no positional embeddings. The conventional fixed positional embeddings break the permutation-invariance property of the cross-attention mechanism, thus are not able to generalize to populations with different compositions in held-out sessions. With no positional embeddings, the model is permutation-invariant by design; however, the loss of information about neural unit functional identities hinders the model's ability to decode the behavior these units encode. We achieve the best of both worlds by our proposed context-dependent positional embeddings, being both permutation-invariant while retaining neural identities for behavior decoding.

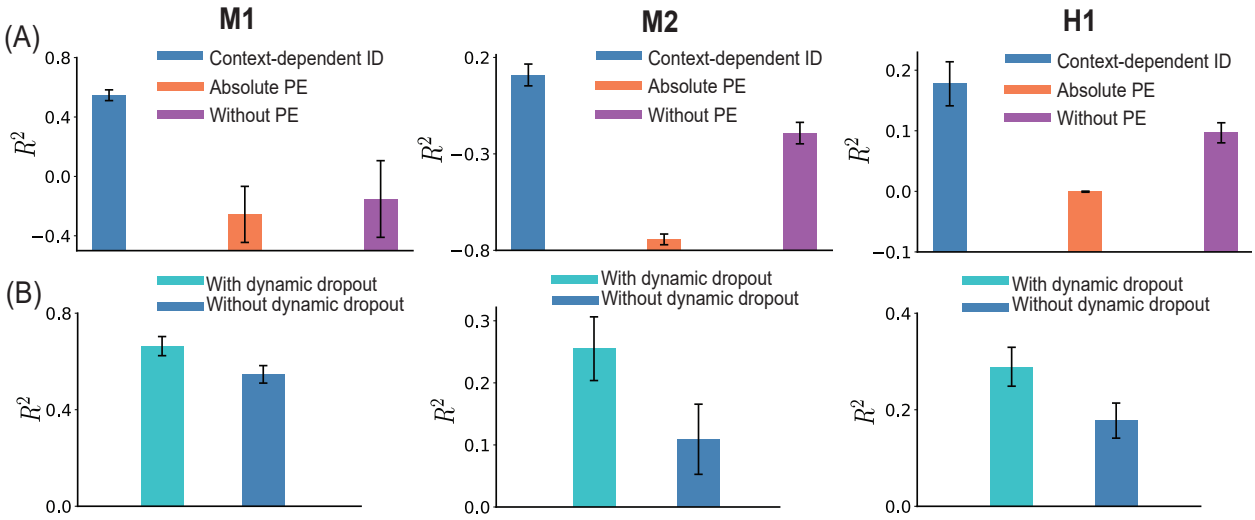

Figure 5.4: **Ablation Study**. Analyses showing the critical roles of our proposed context-dependent ID against fixed positional embeddings (PE) and no positional embeddings (A), our dynamic channel dropout against no dynamic channel dropout (B). Results are shown across M1, M2, and H1 datasets. Bars represent mean R^2 across held-out sessions, whiskers represent standard error of the mean of R^2 across held-out sessions.

To demonstrate the effectiveness of our proposed dynamic channel dropout technique, we compare the cross-session performance of SPINT with dynamic channel dropout and without dynamic channel dropout. We show in Figure 5.4B that dynamic channel dropout serves as an effective regularization technique by preventing the model from overfitting to the population composition in training sessions.

5.5 Appendix

5.5.1 Within-session performance comparison

We include the within-session performance comparison between SPINT and baselines in Table 5.3. This table is similar to Table 1 in the main paper, but with metrics obtained on EvalAI's private splits within the held-in sessions. As observed from the table, SPINT also consistently

outperforms ZS and FSU baselines on the held-in splits.

	Class	**M1**	**M2**	**H1**
Wiener Filter (WF)	OR	0.54 ± 0.01	0.27 ± 0.02	0.24 ± 0.02
RNN	OR	0.75 ± 0.03	0.59 ± 0.07	0.51 ± 0.09
NDT2 Multi [Ref137]	OR	0.77 ± 0.03	0.62 ± 0.03	0.68 ± 0.05
NDT2 Multi [Ref137]	FSS	0.77 ± 0.03	0.63 ± 0.03	0.62 ± 0.04
WF	ZS	0.46 ± 0.06	0.15 ± 0.07	0.20 ± 0.04
RNN	ZS	0.52 ± 0.15	0.20 ± 0.29	0.31 ± 0.13
CycleGAN + WF [Ref70]	FSU	0.61 ± 0.02	0.32 ± 0.03	0.15 ± 0.04
NoMAD + WF [Ref71]	FSU	0.64 ± 0.01	0.35 ± 0.05	0.21 ± 0.06
SPINT (Ours)	GF-FSU	$\mathbf{0.77} \pm 0.02$	$\mathbf{0.59} \pm 0.01$	$\mathbf{0.47} \pm 0.06$

Table 5.3: Within-session performance comparison against oracles (OR), few-shot supervised (FSS), few-shot unsupervised (FSU), and zero-shot (ZS) methods. Our SPINT approach belongs to a special class which we termed Gradient-Free Few-Shot Unsupervised (GF-FSU), where models perform adaptation based on few-shot unlabeled data but *without* any parameter updates at test time. Results are reported as mean ± standard deviation R^2 across held-in sessions, achieved on EvalAI private held-in splits.

5.5.2 Proof of SPINT's permutation-invariance

Let P_R, P_C be the row and column permutation matrices of the same permutation π ($P_C = P_R^\top = P_R^{-1}$ and $P_C P_R = I$). Also let $X' = P_R X$ and $(X^C)' = P_R X^C$ be the row-permuted neural windows and row-permuted calibration trials.

Since the ID embedding of each neural unit i is computed individually from the set of calibration trials for that unit:

$$E_i = \text{IDEncoder}(X_i^C) = \psi(\mathit{pool}(\phi(X_i^C))), \tag{5.5}$$

permuting the neural units in the original population (neural windows X or calibration trials X^C) will permute the embedding matrix E in the exact same order, i.e., $E' = P_R E$.

It follows that:

$$Z' = X' + E' = P_R X + P_R E = P_R(X + E) = P_R Z \tag{5.6}$$

In other words, Z is equivariant to the permutation of neural units.

Cross-attention performed on Z' then becomes:

$$\begin{aligned}
\text{CrossAttn}(Q, Z', Z') &= \text{CrossAttn}(Q, P_R Z, P_R Z) \\
&= \text{softmax}\left(\frac{Q W_K^\top Z^\top P_R^\top}{\sqrt{d_k}}\right) P_R Z W_V \\
&= \text{softmax}\left(\frac{Q W_K^\top Z^\top P_C}{\sqrt{d_k}}\right) P_R Z W_V \\
&= \text{softmax}\left(\frac{Q W_K^\top Z^\top}{\sqrt{d_k}}\right) P_C P_R Z W_V \\
&= \text{softmax}\left(\frac{Q W_K^\top Z^\top}{\sqrt{d_k}}\right) Z W_V \\
&= \text{CrossAttn}(Q, Z, Z)
\end{aligned} \tag{5.7}$$

where $\text{softmax}\left(\frac{Q W_K^\top Z^\top P_C}{\sqrt{d_k}}\right) = \text{softmax}\left(\frac{Q W_K^\top Z^\top}{\sqrt{d_k}}\right) P_C$ because an element is always normalized with the same group of elements in the same row regardless of whether column permutation is performed before or after softmax.

Equation 5.7 concludes Proposition 1 in the main paper.

We note that multi-layer perceptron (MLP), layer normalization, and residual connection are applied row-wise and hence do not affect the overall permutation-invariance property of our SPINT framework.

5.5.3 Correlation of attention scores and firing statistics

We ask whether the attention scores SPINT assigns for each neural unit are correlated with its firing statistics. To answer this question, in each held-out calibration window, we measure the

average attention scores over B behavior covariates, and its firing statistics (mean/standard deviation) over the held-out calibration trials, then calculate the Pearson's correlation between these two quantities using all held-out calibration windows. We show the results in Table 5.4.

We observe that the attention scores correlate moderately with the mean and the standard deviation of the neural unit's firing rates, with higher correlation for the standard deviation than the mean, suggesting that SPINT might be extracting neural units that are active (having high mean firing rates) and behaviorally relevant (having high variance throughout the calibration periods where behavior is varied) to pay attention to in behavioral decoding.

	M1	**M2**	**H1**
ρ(attention scores, mean firing rates)	0.33 ± 0.16	0.76 ± 0.03	0.51 ± 0.04
ρ(attention scores, standard deviation of firing rates)	0.45 ± 0.16	0.87 ± 0.02	0.57 ± 0.03

Table 5.4: Pearson's correlation between attention scores for each neural unit and that unit mean/standard deviation of firing rates during the held-out calibration periods. Results are reported as the mean correlation $\pm$ standard deviation across held-out sessions. All p-values are less than 0.05.

5.5.4 Implementation details

5.5.4.1 Data preprocessing

For neural activity, we use the binned spike count obtained by unit threshold crossing with the standard bin size of 20ms as set forth by the FALCON Benchmark. We follow FALCON's continuous decoding setup for all three M1, M2, and H1 datasets, where rather than decoding trialized behavior from the trialized neural activity (often performed in a non-causal manner), we decode behavior at the last step of a neural activity window, mimicking the online, causal iBCI decoding. To construct the length-W neural window at the beginning of each session,

we pre-pad the session neural time series with $(W-1)$ zeros. We discard the windows whose last time step belongs to a non-evaluated period as defined by FALCON, e.g., inter-trial periods where there is no registered kinematics.

Our IDEncoder infers neural unit identity from trialized calibration trials. As calibration trials vary in length, we interpolate all calibration trials to the same length T, where $T = 100$ for M2 and $T = 1024$ for M1 and H1. We use the Python library `scipy.interpolate.interp1d` with a cubic spline for interpolation. Note that we only perform interpolation for neural calibration trials to synchronize their trial lengths. We still use the raw spike counts for the neural windows, conforming with the continuous decoding setup.

5.5.4.2 Behavior output scaling

For M2 and H1, since values of behavior covariates are relatively small, during training we scale the network behavior predictions by a factor of 0.2 and 0.05 for M2 and H1, respectively, effectively asking the model to predict 5× and 20× the original behavior values. The MSE loss and R^2 metrics are computed between the scaled predicted outputs and the original ground truth values.

5.5.4.3 Inferring neural unit identity

We follow the permutation-invariant framework in [Ref85] for inferring identity E_i of neural unit i:

$$E_i = \text{IDEncoder}(X_i^C) = \text{MLP}_2(\frac{1}{M}\sum_{j=1}^{M}(\text{MLP}_1(X_i^{C_j}))) \tag{5.8}$$

where M is the number of calibration trials, $X_i^{C_j}$ is the neural activity of the j^{th} calibration trial of neural unit i, MLP_1 and MLP_2 are two 3-layer fully connected networks. MLP_1 projects the length-T trials to a hidden dimension H, and MLP_2 projects the length-H hidden features to length-W neural unit identity output.

Dropout	**0**	**0.2**	**0.4**	**0.6**	**0.8**	**DD [0,1]**
R^2	0.51 ± 0.13	0.62 ± 0.10	0.63 ± 0.10	0.63 ± 0.10	0.60 ± 0.09	0.64 ± 0.10

Table 5.5: SPINT's cross-session performance against dynamic dropout and different choices of fixed dropout rates. Results are reported as mean ± standard deviation across held-out calibration sessions. DD [0,1] stands for dynamic dropout with variable dropout rates between 0 and 1.

DD range	**[0, 0.1]**	**[0, 0.2]**	**[0, 0.3]**	**[0, 0.4]**	**[0, 0.5]**	**[0,1]**
R^2	0.59 ± 0.07	0.59 ± 0.07	0.61 ± 0.10	0.62 ± 0.07	0.63 ± 0.07	0.64 ± 0.10

Table 5.6: SPINT's cross-session performance across different ranges of dynamic dropout. Results are reported as mean ± standard deviation across held-out calibration sessions.

# heads	**4**	**8**	**16**	**32**	**64**
R^2	0.62 ± 0.08	0.63 ± 0.09	0.64 ± 0.10	0.65 ± 0.11	0.64 ± 0.10

Table 5.7: SPINT's cross-session performance for different cross-attention head counts. Results are reported as mean ± standard deviation across held-out calibration sessions.

# self-attention layers	**0**	**1**	**2**	**3**	**4**
R^2	0.64 ± 0.10	0.63 ± 0.13	0.57 ± 0.13	0.61 ± 0.10	0.60 ± 0.15

Table 5.8: SPINT's cross-session performance for different number of self-attention layers. Results are reported as mean ± standard deviation across held-out calibration sessions.

5.5.4.4 Behavioral decoding by cross-attention

After neural identity for all units E is inferred, we add it to the neural window input X to form the identity-aware neural activity Z, i.e., $Z = X + E$. We then use the cross-attention

# cross-attention layers	1	2	3	4	5
R^2	0.64 ± 0.10	0.65 ± 0.10	0.65 ± 0.10	0.64 ± 0.11	0.62 ± 0.13

Table 5.9: SPINT's cross-session performance for different number of cross-attention layers. Results are reported as mean ± standard deviation across held-out calibration sessions.

Window size	50	100	200	400	600
R^2	0.65 ± 0.10	0.64 ± 0.10	0.64 ± 0.10	0.60 ± 0.10	0.61 ± 0.09

Table 5.10: SPINT's cross-session performance for different context window sizes. Results are reported as mean ± standard deviation across held-out calibration sessions.

mechanism in the latent space to decode last step behavior covariates. Specifically:

$$Z_{in} = \mathrm{MLP}_{\mathrm{in}}(Z) \tag{5.9}$$

$$\tilde{Z} = Q + \mathrm{CrossAttn}(Q, \mathrm{LayerNorm}(Z_{in}), \mathrm{LayerNorm}(Z_{in})) \tag{5.10}$$

$$Z_{out} = \tilde{Z} + \mathrm{MLP}_{\mathrm{attn}}(\mathrm{LayerNorm}(\tilde{Z})) \tag{5.11}$$

$$Y = \mathrm{MLP}_{\mathrm{out}}(Z_{out}) \tag{5.12}$$

5.5.4.5 Hyperparameters

We include the notable hyperparameters used to optimize SPINT in Table 5.11. We train and evaluate models for each M1, M2, and H1 dataset separately. We train the models using all available held-in sessions and evaluate on all available held-out sessions. We use Adam optimizer [Ref125] for all training.

We include representative hyperparameter sweeps demonstrating SPINT's robustness to hyperparameter choices in Tables 5.5, 5.6, 5.7, 5.8, 5.9, 5.10. This robustness allows SPINT to effectively capture long-range context while maintaining a minimalist architecture without

	M1	**M2**	**H1**
Batch size	32	32	32
Window size	100	50	700
Max trial length	1024	100	1024
Number of IDEncoder layers	3, 3	3, 3	3, 3
Number of cross attention layers	1	1	1
Hidden dimension	1024	512	1024
Behavior scaling factor	1	0.2	0.05
Learning rate	1e−5	5e−5	1e−5

Table 5.11: Hyperparameters used to train SPINT on the M1, M2, and H1 datasets.

compromising generalizability. All sweep results were obtained on 20% of calibration trials held out from each session of the M1 dataset rather than on the EvalAI test split.

5.5.4.6 Computational resources

SPINT was trained using a single A40 GPU, consuming less than 2GB of GPU memory with batch size of 32 and taking around 12 hours, 5 hours, and 8 hours to finish 50 training epochs for M1, M2, and H1, respectively. We select checkpoints for evaluation at epoch 50 in all M1, M2, and H1 datasets.

Chapter 6

LEARNING CONTEXT-AWARE REPRESENTATIONS FOR BRAIN-TO-TEXT DECODING

This chapter contains material that was previously published in [TL4].

6.1 Background

Verbal communication is a unique feature of human social interaction. Loss of ability to articulate speech as a result of neurological pathologies such as stroke and Amyotrophic Lateral Sclerosis (ALS) can significantly reduce the quality of life for affected individuals. Recent advancements in Brain-Computer Interfaces (BCI) offer promising pathways toward restoring communication ability in these patients by translating neural activity into communicative messages. These messages can be conveyed through various modalities, including typed characters [Ref31], handwriting [Ref23], text [Ref152, Ref24, Ref153], and synthesized speech [Ref153].

Among existing speech BCI systems, the methods with highest decoding accuracy and throughput are those that translate neural signals associated with orofacial movements during attempted speech into fundamental acoustic units (phonemes), which are then decoded into words and sentences [Ref24, Ref153]. This two-staged approach typically involves (1) neural signal to phonemes: using a temporal deep network to decode a binned multi-channel neural time series into probability of phonemes being spoken at each time step, and (2) phonemes to text: employing a language model (LM) to infer the most probable sequence of words given the phoneme probabilities.

Prior work shows that decoding phonemes as an intermediate representation rather than directly decoding words, provides the system the flexibility to decode phrases from extensive

vocabularies a limited set of training examples [Ref153], since from a fixed set of 40 phonemes, one can practically construct any word of any arbitrary length. This scalability is especially advantageous given the limited availability of neural recordings in clinical settings.

While decoding single phonemes from neural activity may offer more scalability than decoding words, it remains a challenging task. Given the innate variability of neural signals, the mapping from neural activity to phonemes is many-to-one and highly nonlinear. Furthermore, evidence suggests that cortical activation patterns producing a particular phoneme is not static, but can vary depending on the context of surrounding phonemes, a phenomenon known as *coarticulation* [Ref154, Ref155]. In other words, cortical neurons at any given time during speech production are likely encoding a phoneme along with its context, rather than a phoneme in isolation. Given this observation, diphone [Ref156] - a sequence of two adjacent phonemes - is a more suitable representation for capturing this context dependency in neural signals and potentially reducing the nonlinearity in phoneme decoding. Hence we propose to decompose the phoneme classification task into subtasks of diphone classification, after which diphone probabilities are summed up to obtain the phoneme prediction, i.e. predicting phoneme distribution by marginalizing over the diphone distribution. We show that this divide-and-conquer strategy significantly enhances phoneme decoding performance.

Recently introduced approaches leverage language models, such as n-gram model, to translate phoneme probabilities into words [Ref24, Ref153, Ref157]. Notably, [Ref157] further uses GPT3.5 [Ref81] after a 5-gram model to refine the resulting word sequences into coherent sentences by ensembling multiple 5-gram transcription candidates. However, the transcription candidates generated by the n-gram model can significantly deviate from the ground truth phoneme sequence. To address this issue, we propose to augment the ensembling method in [Ref157] to include decoded phonemes alongside transcription candidates, which proves to provide extra information for GPT3.5 to infer the correct transcription. Additionally, we propose an In-Context Learning (ICL) paradigm for LLMs, enabling them to adapt quickly to newly decoded inputs in a gradient-free manner without the need for the computationally expensive finetuning process. This approach offers a more efficient alternative for improving

transcription accuracy in resource-constrained settings.

6.2 Contributions

In summary, our contributions in this work are as follows:

- We propose DCoND (**D**ivide-and-**Co**nquer **N**eural **D**ecoder), a novel framework for decoding phonemes from neural activity during attempted speech. Backed by neuroscientific insights, DCoND infers the temporal phoneme distribution by marginalizing over the diphone distribution, leveraging the context-dependent nature of phonemes in neural representation.

- We propose incorporating decoded phonemes alongside decoded words in an **L**LM-based ensembling strategy to enhance the speech decoding performance. We also propose the use of (**I**CL) paradigm (DCoND-LI) as an alternative to **F**ine**T**uning LLMs (DCoND-LIFT), offering a more efficient solution for resource-constrained brain-to-text systems.

- We demonstrate the effectiveness of our approaches on the Brain-to-Text 2024 benchmark, where our approach achieves state-of-the-art (SOTA) PER of 15.34% and WER of 5.77%, a significant improvement compared to 8.93% WER of the leading SOTA method.

6.3 Methods

Problem formulation The problem of decoding phonemes from neural activity can be formulated as follows. Let $f : X \to Z$ be the mapping from neural activity $X \in \mathbb{R}^{T \times D}$ to phoneme sequence $Z \in \mathbb{R}^{T'}$, where D is the number of neural features, T is the number of neural time bins, and T' is the number of ground truth phonemes in a sentence. We note that $T > T'$ in general, i.e. the articulation of one phoneme may span multiple timesteps. We also emphasize that there is no ground truth temporal alignment between X and Z due

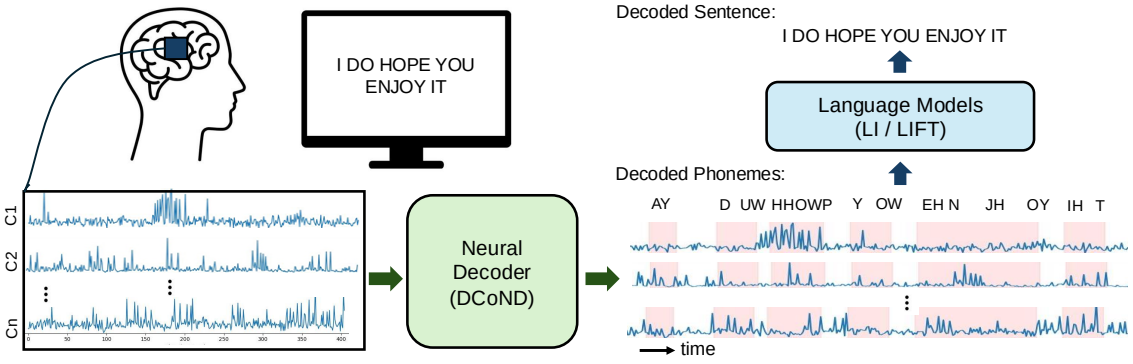

Figure 6.1: Overview of the Brain-to-Text decoding pipeline. The Neural Decoder with Divide-and-Conquer Strategy (DCoND) decodes multi-channel neural activity into phonemes. The phonemes are subsequently converted into words by LLMs using either ICL or fine-tuning techniques.

to the nature of the silent speech task. Both T and T' vary across trials depending on the length of the sentence in that trial. We aim to learn a model $f_\theta : X \to Z$ to approximate f with a set of parameters θ. We use an RNN model (GRU) for f_θ. GRU has demonstrated superior performance on this dataset, as reported in previous works [Ref24, Ref158]. A comparative study of alternative architectures, such as LSTM and transformer, is avaliable in the Appendix. Decoded phonemes Z can be subsequently translated to sentences Y with the help of a language model $h_\phi : Z \to Y$, where h_ϕ can be a pre-built statistical language model, e.g. 5-gram, or an LLM, e.g. GPT3 [Ref81]. The overall pipeline is depicted in Figure 6.1.

A Divide-and-Conquer strategy for phoneme decoding Decoding phonemes from neural activity is a nontrivial task given the highly nonlinear nature of f and the variability of the neural population dynamics. Evidence exists that the neural representations for phonemes vary depending on the surrounding contexts [Ref154, Ref155]. We illustrate this observation in Figure 6.2 where segments of phoneme-aligned neural activity form clusters in the neural space based on the context they are in. It can be seen that there is no single cluster

representing each phoneme, but rather each phoneme is represented by multiple subclusters. We further show that the subclusters are identifiable by the phoneme preceding the phoneme of interest. For instance, the phoneme AH is represented by subclusters DH $\rightarrow$ AH and SIL $\rightarrow$ AH (see further discussion in Section 6.4.4). Learning to model these context-aware sub-units of speech instead of single phonemes directly could facilitate the phoneme decoding task. Concretely,

$$f(x) := p(Z|X) = \sum_{S} p(Z, S|X) = \sum_{s \in S} g_s^Z(x) \tag{6.1}$$

where S is a random variable denoting the context surrounding the phoneme Z. To be noticed, Z takes values from phoneme classes, such that $Z \in [1, C]$. The problem of learning single phoneme classes (f) now reduces to the problem of learning the phoneme context-dependent subclasses (g_s^Z), which is more manageable and in-line with the context-dependent nature of the data. We refer to our phoneme decoder with this divide-and-conquer strategy as **DCoND**.

Diphone as a context-dependent representation of phonemes The context-dependent subclasses could be defined in multiple ways. In this work, we adopt diphone, a context-dependent representation for phoneme sequences where transitions between phonemes are the subject of interest. For example, the single phoneme representation of "hope", $H, \quad OW, \quad P$, will have a diphone representation:

$$SIL \rightarrow H, \quad H \rightarrow H, \quad H \rightarrow OW, \quad OW \rightarrow OW, \quad OW \rightarrow P, \quad P \rightarrow P, \quad P \rightarrow SIL.$$

where 'SIL' indicates the silence between the words. Diphone expands the length of phoneme sequence to $T'' = 2T'$ and increases the number of decoding classes to C^2, where $C = 40$ for the English language[1].

Formally, we reformulate the problem of decoding phoneme from neural activity as the

[1]the phonemes are defined as per CMU Pronouncing Dictionary: `http://www.speech.cs.cmu.edu/cgi-bin/cmudict/`

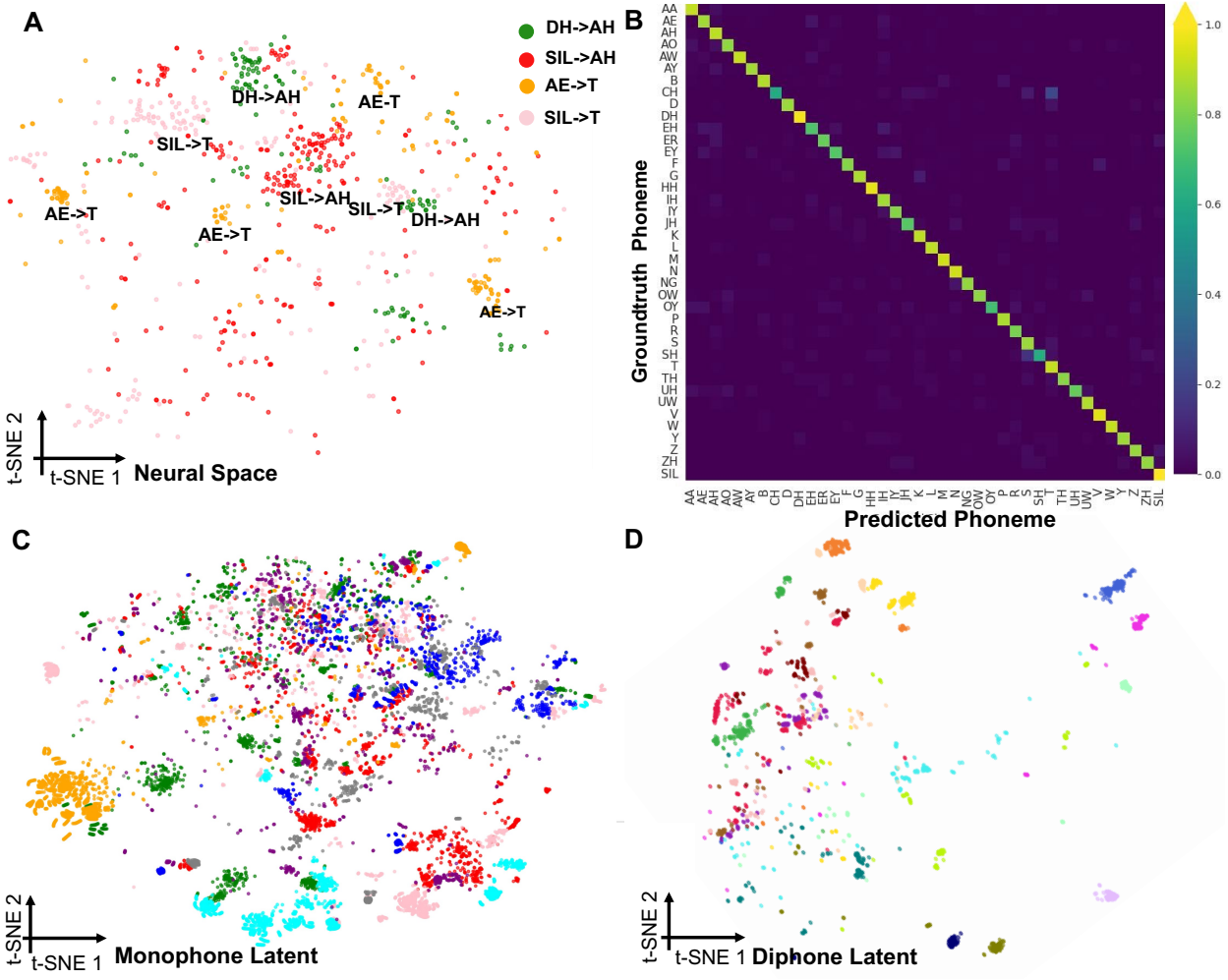

Figure 6.2: **A**: 2D t-SNE visualization of neural signal projections illustrating the context-dependent nature of phonemes in neural reprentations. Different colors indicate different diphone classes. **B**: Confusion matrix of ground truth phonemes vs. DCoND's predicted phonemes. **C**: 2D t-SNE visualization for the latent space of the neural decoder trained with single phoneme decoding objective (Monophone). Different colors indicate different phoneme classes. **D**: 2D t-SNE visualization for the latent space of the neural decoder trained with diphone decoding objective. Different colors indicate different diphone classes.

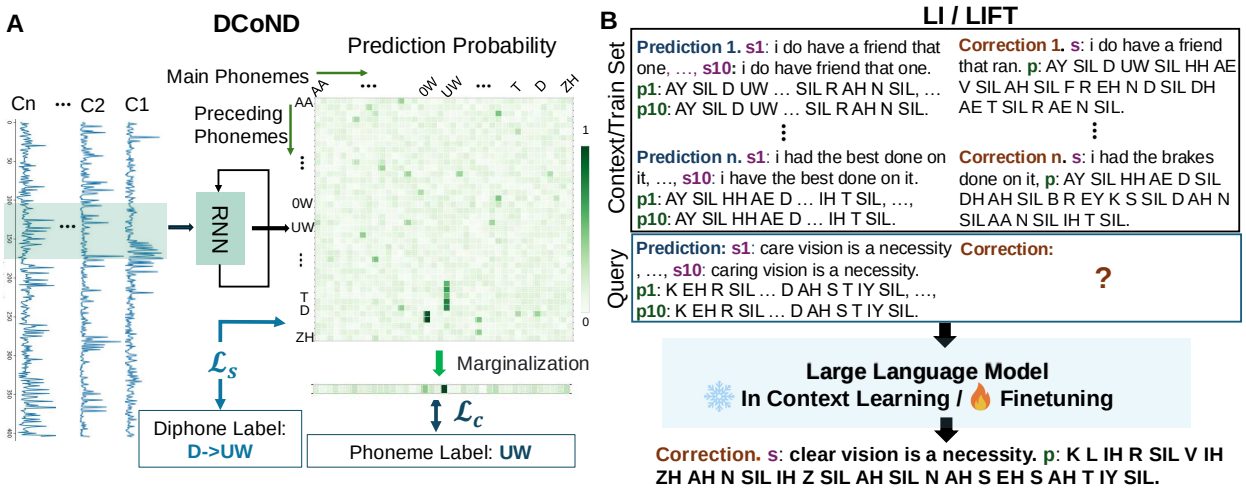

Figure 6.3: **A**: Illustration of the brain-to-phoneme decoding pipeline (DCoND). An RNN in DCoND takes multi-channel neural signals as inputs and generates diphone probabilities, which are then marginalized into single phoneme probabilities. **B**: Illustration of the ensembling method for refining transcription predictions (LI/LIFT). Given an ensemble of phoneme and transcription candidates as a query, GPT3.5 produces the most sensible transcription composed from these inputs. To do this, the LLM leverages examples of prediction-correction pairs provided either in-context at inference time (LI) or as training data during the finetuning process (LIFT).

marginalization over the distribution of diphones, conditioning on the observed neural activity

$$p(Z = c_i|X) = \sum_{c_j \in S} p(c_j, c_i|X),$$

where $p(c_j, c_i|X)$ is the probability of neural activity X encoding the diphone $c_j \rightarrow c_i$. A visualization of the marginalization process is shown in Fig. 6.3A. Neural activity is processed by an RNN to predict the probability of 40^2 diphones being spoken at each timestep. The diphone probability is depicted by a 40×40 matrix where columns correspond to the main phonemes and rows correspond to the preceding phonemes. The single phoneme probability is then obtained by summing the joint probabilities column-wise.

Parameter Optimization for Phoneme Decoding As mentioned above, we do not have the temporal alignment between T timesteps of neural activity and T' ground truth phonemes in each trial. We therefore use the Connectionist Temporal Classification (CTC) loss as proposed in [Ref159] to resolve the non-alignment issue. Specifically, we try to maximize the probability of Z given X

$$p(Z|X) = \sum_{A \in \mathcal{A}_{(X,Z)}} \prod_{t=1}^{T'} p(a_t|X), \tag{6.2}$$

where $A_{(X,Z)}$ is the set of valid alignments between X and Z.

Now that we have the diphone representation for each ground truth sentence, we consider the CTC losses over both the diphone and single phoneme representations:

$$\mathcal{L} = \alpha \mathcal{L}_c + (1 - \alpha)\mathcal{L}_s \tag{6.3}$$

where $\mathcal{L}_c = -\log(\sum_{A \in \mathcal{A}_{(X,Z)}} \prod_{t=1}^{T'} p_m(a_t|X))$ is the loss for single phoneme decoding, and $\mathcal{L}_s = -\log(\sum_{A \in \mathcal{A}_{(X,S)}} \prod_{t=1}^{T''} p(a_t|X))$ defines the loss over subclasses (diphone) decoding.

Coefficient α controls the balance of the single phoneme decoding and diphone decoding. α is designed to be small at the beginning and gradually increase over the course of training. See Appendix 6.5.7 for more implementation details.

Word Decoding with Language Models The predicted phoneme probabilities are further transformed into high-quality text through (i) generation of transcription candidates from phonemes, (ii) re-scoring of transcription candidates, and (iii) error correction using an ensemble of selected candidates.

Transcription Generation. During the phase of candidate sentence generation, we convert the predicted phoneme probabilities into words using a 5-gram model. Based on the predicted phoneme probability distribution, the 5-gram model leverages its internal word and sentence distributions to generate the most likely sentence candidates [Ref160, Ref24]. Each candidate is associated with a likelihood score provided by the 5-gram model.

Transcription Re-scoring LLMs trained on large corpora of texts, such as the Open Pre-trained Transformer (OPT) [Ref161], could provide more accurate likelihood of the

generated transcriptions. Hence, we use OPT to re-score the 5-gram likelihood outputs. The transcription candidates with the highest likelihoods are selected[Ref24].

Transcription Error Correction with Ensemble Method While the 5-gram and OPT models can correct some phoneme errors made by the phoneme decoder to produce more contextually sound sentences (transcriptions), these sentences are not always perfect. Variations of the phoneme decoding model could result in changes of generated and selected sentence candidates. Ensembles of phoneme decoding models, with each model being an expert in different situations, could mitigate the errors made by another model.

In [Ref157] GPT3.5 is finetuned to evaluate an ensemble of 10 transcription candidates and generate the most sensible sentence from the 10 candidates. However, providing GPT3.5 only the candidate transcriptions hinders the LLM's ability to understand the underlying phoneme sequences, which are the generating source of the transcriptions and might have been incorrectly converted by the 5-gram model. We therefore propose to include both the transcription candidates and the corresponding phoneme sequences as inputs to GPT3.5, tasking the model with generating both the correct transcription and phoneme sequence. An illustration of such task is shown in Fig.6.3. By finetuning the LLM in this manner, we train it to infer the relationship between predicted phonemes and the predicted transcriptions, as well as identifying common model-specific mistakes made by the phoneme decoders across their predictions. We show in Section 6.4.3 that this strategy further boosts the WER from 8.06% to 5.77%.

In addition, since finetuning LLM is a resource-intensive process, we also propose to leverage ICL as an alternative learning paradigm for refining predicted transcriptions. Instead of finetuning GPT3.5 over multiple batches of (10× predictions, 1× ground truth) pairs, we directly include N examples of these pairs as context in each prompt, along with a query input to be refined. The LLM then leverages its ICL ability to quickly refine the query transcriptions without updating its weights. The prompts used for both in-context inference and finetuning are detailed in the Appendix.

6.4 Results

6.4.1 Dataset

We demonstrate the effectiveness of DCoND-LIFT in decoding attempted speech using the Brain-to-Text Benchmark 2024 [Ref24, Ref162]. The dataset was collected from a human subject with ALS who had lost the ability to produce intelligible speech. In the experiments, the subject attempts to silently speak sentences displayed on a screen. These sentences are composed from a vocabulary set of 125,000 words. In each trial, one sentence is shown followed by an auditory 'Go' cue, after which the subject attempts to speak at their own pace. Neural activity (multiunit threshold crossings and spike band power) is recorded from the ventral premotor cortex (6V) while the subject attempted speaking. Due to the nature of the silent speech task, the correspondence between neural activity and the produced speech is unknown. The dataset is split into training, validation, and competition sets with 8800, 600, and 1200 sentences, respectively.

6.4.2 Evaluation Metrics

PER Phoneme Error Rate (PER) is calculated by comparing the decoded phoneme sequence with the ground truth phoneme sequence. After aligning the recognized phoneme sequence with the reference phoneme sequence, the number of insertions, deletions, and substitutions required to match the sequences are counted. The sum of these operations is divided by the total number of phonemes in the ground truth sequence to compute the PER. This metric reflects how accurately neural signals can be recognized into phonetic units.

WER Similar to PER, word error rate (WER) is computed by aligning the sequence of recognized words with the ground truth sentence first and then counting the number of insertions, deletions, and substitutions of words needed to reconcile any discrepancies between the two sequences. The total number of these operations is divided by the total number of words in the reference sequence to obtain WER. As neural activity is translated into phonemes before converted into words, WER reflects the performance of both neural decoder

Table 6.1: Performance comparison on Brain-to-Text 2024 Benchmark

	PER×100 ↓	WER×100 ↓	P-WER×100 ↓
NPTL [Ref24]	16.62	9.46	11.33
LISA [Ref157]	–	8.93	–
DCoND-L (Ours)	**15.34**	8.06	**8.02**
DCoND-LI (Ours)	–	7.29	–
DCoND-LIFT (Ours)	–	**5.77**	–

and the language model.

P-WER We adapt Perceptual Word Error Rate (P-WER) [Ref153] to measure the quality of phoneme decoding at the word perception level. Specifically, we use eSpeak-NG [Ref163][2] to synthesize speech from the decoded phoneme sequences. Then the synthesized speech is translated into sentences by Whisper [Ref164] from which the WER is estimated. Considering the systematic errors introduced by the eSpeak-NG synthesizer and the Whisper ASR system, we define P-WER as follows

$$\text{P-WER} = (1 - \frac{1 - \text{WER}_{Whisper\text{-}P}}{1 - \text{WER}_{Whisper\text{-}GT}}),$$

where $\text{WER}_{Whisper-GT}$ and $\text{WER}_{Whisper-P}$ are the WER measured on Whisper's decoded transcriptions when audio is synthesized with ground truth phoneme sequences (GT) and predicted phoneme sequences (P), respectively.

6.4.3 Comparison with SOTA Methods

We show DCoND-LIFT achieves state-of-the-art performance on the Brain-to-Text Benchmark 2024, where WER is the primary evaluation metric (see Table 6.1). Specifically, we compared

[2] `https://github.com/espeak-ng/espeak-ng`

DCoND-LIFT with the leading methods NPTL [Ref24] and LISA [Ref157]. NPTL uses a 5-layer RNN to decode neural activity to phonemes, followed by a combination of 5-gram and OPT language models [Ref160, Ref161] to translate decoded phonemes to texts. LISA uses the same RNN model architecture as NPTL to decode phonemes from neural activity, but leverages GPT3.5 to further improve transcriptions given by the 5-gram model.

As seen in Table 6.1, our model variants outperform the competing methods across the board. DCoND combined with 5-gram LM and OPT (DCoND-L) yields WER of 8.06%, compared to 9.46% WER of NPTL and 8.93% of LISA. Further sensitivity analysis is provided in Table 6.2 of the Appendix. Given that DCoND-L uses the same RNN backbone and LMs as NPTL, we posit that the improvements in WER come from the effectiveness of our divide-and-conquer phoneme decoding strategy. Indeed, DCoND-L achieves a better PER and P-WER (15.34% and 8.02% compared to 16.62% and 11.33% of NPTL), proving that modeling context-dependent phoneme representations facilitates the phoneme decoding task.

The WER further improves when we equip DCoND-L with the more powerful language model GPT3.5 to evaluate an ensemble of predicted transcriptions and their associated phoneme representations. When ensemble examplars are shown to GPT3.5 in-context (DCoND-LI), WER improves from 8.06% to 7.29%. This performance is achieved with 25 ICL examplars, the largest number of ICL examplars GPT3.5 can afford due to its prompt length constraint. When we finetune GPT3.5 using all available training examplars (DCoND-LIFT), WER is further boosted to 5.77%, a significant improvement over 8.93% WER of LISA. These results support our proposal of including both transcriptions and phoneme representations in the demonstrations to GPT3.5 so that it can leverage the relationship between phonemes and words to refine the transcriptions.

6.4.4 Phoneme Decoding Analyses

Neural activity represents phonemes in context-dependent clusters Previous works demonstrate that the accuracy of decoding phonemes from neural activity could degrade when phonemes are pronounced in the context of other phonemes as opposed to being pronounced

individually [Ref155]. To get a glimpse of how the brain encodes phonemes, in Fig. 6.2A we visualize phoneme-aligned segments of neural activity in the 2D t-SNE space [Ref165]. Since the dataset does not have the exact temporal correspondence between neural activity and phonemes, we leverage Dynamic Time Warping (DTW) to align the ground truth phonemes to neural activity segments according to the timestamps obtained from the decoded phonemes [Ref166]. We annotate the neural activity segments based on the resulting phoneme alignment. The visualization reveals that neural activity segments form distinct clusters in the t-SNE space. Notably, these clusters are organized based not only on single phonemes but also on the context in which they are spoken. For instance, during periods where 'T' is the main phoneme being spoken, the neural activity is organized into subclusters of AE→T (orange) and SIL→T (pink), depending on whether phoneme 'AE' or 'SIL' is spoken before 'T'. Similar observations hold for subclusters DH→AH (green) and SIL→AH (red) for phoneme 'AH'. We note that further subclusters could exist within each subcluster, suggesting a continuum of finer contexts beyond the preceding phoneme.

Decoding diphone leads to enhanced clusters in latent space We visualize in Figures 6.2C and 6.2D the latent space at the last layer of the neural decoder when trained to decode single phonemes (monophones) vs. diphones. In Figure 6.2C, each color represents a single decoded phoneme label. For clear visualization, we selected five single phoneme classes with the most samples. The clusters that correspond to single phonemes appear to spread out over the whole space, and overlap with each other. In Figure 6.2D, each color represents a decoded diphone. Since there are fewer samples for each diphone, we visualize 16 diphone classes with the highest occurrence. It can be observed that the neural decoder represents diphones in the latent space by clusters that are significantly more condensed and well-separated. Such clear structure facilitates the subsequent classification of single phonemes and demonstrates the effectiveness of our divide-and-conquer phoneme decoding method.

Phoneme Prediction Error Analysis In Figure 6.2B, we show the confusion matrix of the predicted phonemes and the ground truth phonemes. From the figure we can see that

Table 6.2: Sensitivity analysis on Brain-to-Text 2024 Benchmark. We report the mean ± standard deviation of PER, WER, and P-WER across 5 random seeds.

	PER×100 ↓	WER×100 ↓	P-WER×100 ↓
NPTL [46]	16.62	9.46	11.33
LISA [2]	–	8.93	–
DCoND-L (Ours)	**15.44 ± 0.46**	8.39 ± 0.22	**8.09 ± 1.62**
DCoND-LI (Ours)	–	7.23 ± 0.08	–
DCoND-LIFT (Ours)	–	**5.90 ± 0.08**	–

most phonemes are correctly classified with accuracy greater than 80%. The mistakes the model typically makes, if any, are on phonemes that are pronounced similarly. For example, the model usually confuses 'SH' with 'S', and 'CH' with 'TH'. Since the articulation of these phonemes is very similar, the neural activity generating them is likely to be similar. Such confusion is expected to some extent, given the ALS condition hindering the subject's ability to clearly articulate the desired words.

6.5 Appendix

6.5.1 Sensitivity Analysis

We report the mean and standard deviation of DCoND-L, DCoND-LI and DCoND-LIFT in Table 6.2. The mean and standard deviation are obtained across 5 random seeds. The proposed methods (DCoND-L, DCoND-LI and DCoND-LIFT) maintain a significant gap over the NPTL and LISA baselines [Ref24, Ref157].

Table 6.3: Diphone vs DCoND decoding performance in terms of PER and WER. Results are reported as the mean ± standard deviation across 5 random seeds.

	Diphone	DCoND
PER×100 ↓	19.14 ± 0.08	15.44 ± 0.46
WER×100 ↓	12.73 ± 0.22	11.79 ± 0.30

6.5.2 Ablation of DCoND Algorithm

We further examined the role of marginalization step in DCoND. Specifically, we ablated the $\mathcal{L}_c$ loss term, retaining only the $\mathcal{L}_s$ loss term. The predicted diphone probabilities are taken as inputs to a 3-gram language model. The resulting WER is presented in Table 6.3. For a fair comparison, we also measured the WER using a 3-gram language model with marginalized outputs from DCoND. The results in Table 6.3 demonstrate the importance of marginalizing over the predicted diphone probabilities in DCoND which achieves lower WER (11.79) compared to the case where WER marginalization procedure is ablated (WER=12.73).

6.5.3 Triphone as an alternative for context-dependent phoneme representation

Triphones expand upon diphones by incorporating a larger context. Specifically, a triphone considers one phoneme before and one phoneme after the current main phoneme. Consequently, when a neural signal segment is decoded into acoustic units based on the continuity of three phonemes, it reflects a triphone structure. For example, the single phoneme sequence

$$H, \quad OW, \quad P$$

for "hope", can be transferred to triphone

$$\text{“}SIL \rightarrow H \rightarrow OW, \quad H \rightarrow OW \rightarrow P, \quad OW \rightarrow P \rightarrow SIL\text{”}.$$

In this scenario, the time steps required for decoding single phonemes and triphones remain the same. However, triphones introduce a substantial increase in the number of classes, scaling as N^3, which can be prohibitively large (e.g., 64000 when $N = 40$). The divide and conquer idea in this case could be expressed as:

$$f(x) = p(Z = c_i|X) = \sum_{c_j \in C, c_q \in C} p(c_j, c_i, c_q|X)$$

Similar to the diphone probability matrix, these triphone classes are then mapped into a triphone matrix, where each element represents the probability of the current neural signal encoding the phoneme transition from phoneme c_j to phoneme c_i and concluding at phoneme c_q. By summing over the first and last dimensions, we obtain $p(Z = c_i|X)$. Given the potential sparsity of triphone combinations, certain triphone subclasses may not occur frequently in a given language. To mitigate this, we select the top K subclasses for each triphone sample, based on occurrence counts within the current vocabulary. Specifically, for a main phoneme c_i, we rank all possible combinations of $*- > c_i- > *$ and retain the top K as subclasses for the phoneme class c_i.

Additionally, aside from selecting the top K subclasses, an alternative approach involves grouping phones according to articulation similarity [Ref152]. This categorization leads to subclasses of the phoneme c_i as $group_j- > c_i- > group_q$. We categorize phonemes into 14 groups, encompassing Bilabial Sounds, Labiodental Sounds, Dental Sounds, Alveolar Sounds, Palatal Sounds, Velar Sounds, Glottal Sounds, Front Vowels, Central Vowels, Back Vowels, and SIL. In this context, the number of subclasses amounts to $14 * 40 * 14$, which is comparable to the number of classes when $K = 200$ (resulting in a total of 200*40 subclasses).

6.5.4 Additional Ablation Study on the Contribution of LMs

We conduct additional study to assess the role of phoneme-to-transcription generation and re-scoring methods (Figure 6.4). We show that removing the re-scoring step performed by the OPT model in DCoND-L significantly degrades WER (DCoND-3gram and DCoND-5gram), highlighting the importance of the transcription re-scoring step. In addition, the 5-gram

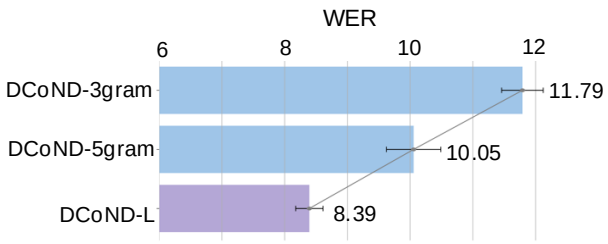

Figure 6.4: Ablation study on the contribution of re-scoring step in the phoneme-to-transcription pipeline. DCoND-3gram: DCoND decoding with 3-gram language model for transcription generation. DCoND-5gram: DCoND decoding with 5gram language model for transcription generation. DCoND-L: DCoND decoding with 5gram language model and re-scoring step. Performance is reported as the mean ± standard deviation across 5 random seeds.

model with longer phoneme dependency generates more accurate transcription candidates compared to the 3-gram model.

6.5.5 Open-Source LLMs for DCoND-LI & DCoND-LIFT

In addition to the closed-source GPT-3.5, we explore the use of the open-source Llama-3.1-70B for refining transcription predictions. We evaluated Llama-3.1-70B in both in-context learning (DCoND-LI) and fine-tuning (DCoND-LIFT) scenarios and compare it against GPT3.5 (Table 6.4). Llama-3.1-70B performs on par with GPT3.5 in ICL setting, while closely trail behind in finetuning setting, all the while outperforming NPTL and LISA baselines. These results demonstrate our method's robustness and generalizability to other LLMs besides GPT3.5, and warrant the accessibility of our methods to the broad community.

Table 6.4: We compare the effect of language model version to the DCoND-LI and DCoND-LIFT performance. The model candidates are GPT-3.5 vs Llama-3.1-70B.

	Llama-3.1-70B WER	GPT 3.5 WER
DCoND-LI	7.38	7.29
DCoND-LIFT	6.85	5.77

Table 6.5: Comparison of different model architectures on phoneme decoding performance. Phoneme Eror Rate (PER) is reported as the mean ± standard deviation across 5 random seeds.

	PER		
	Transformer	LSTM	GRU
NPTL	39.58 ± 0.15	17.49 ± 0.32	16.63 ± 0.19
DCoND	38.88 ± 0.17	16.08 ± 0.23	15.44 ± 0.46

6.5.6 Investigation on Architecture Choices for Neural Decoders

We study the effects of different model architectures on the phoneme decoding performance (PER) (Table 6.5). We observe a significant performance degradation in PER when using Transformer as the neural decoder. On the other hand, RNN counterparts (LSTM and GRU) perform decently well, with GRU being the most performant model for both single phoneme decoding (NPTL) and diphone decoding (DCoND).

6.5.7 Implementation Details

We preprocess the neural signal and construct an RNN neural encoder following the methodology outlined in [Ref24]. The raw neural signal $X \in \mathbb{R}^{T \times D}$ is initially partitioned into smaller

patches with a window size of W, resulting in a patched neural signal of shape $X \in \mathbb{R}^{T' \times (DW)}$. Overlapping between patches is permitted and determined by the stride size. $W = 14$ for diphone experiments and 32 for the triphone experiemnts. The bidirectional RNN processes these patched neural signals as inputs, which are subsequently transformed into the neural representation space $H = [h_1, h_2, \cdots, h_{T'}] \in \mathbb{R}^{T' \times d}$. A fully connected layer then maps the hidden representations to diphone or triphone subclasses, denoted as $P(S = s_i|X)$. The outputs of the fully connected layer are used to compute $\mathcal{L}_s$. The computation of single phoneme probabilities is detailed in Equation 6.3. We merge the probability computed from diphone or triphone.

During the RNN training, we utilize a batch size of 32, a learning rate of 0.02, and the Adam optimizer across various experiments the same set of parameters as used in NPTL baseline [Ref24]. To facilitate diphone and triphone learning, we initially train the subclasses for 10 epochs and then gradually increase the ratio of the single phoneme loss by 0.1 every 10 epochs until it reaches 0.6. The number of training epochs varies for single phoneme learning, diphone learning, and triphone learning. Specifically, we conduct experiments for up to 100 epochs for single phoneme learning (NPTL baseline), 120 epochs for diphone learning, and 140 epochs for triphone learning since the diphone and triphone required additional subclass training procedures. Increasing the number of training epochs can often lead the model to overfit the training data. Training was done on 2 GeForce RTX 2080 Ti with around 12GB memory. The training take around 6-8 hours.

The 5-gram model takes the predicted phoneme logits as inputs, which can be scaled by a temperature factor denoted as t using the formula $logits := logits/t$. Through experimentation, we have found that setting $t = 1.2$ generally improves the decoding performance. Therefore, we use $t = 1.2$ for our experiments, including the implementation of NPTL, which has resulted in improved baseline results. Specifically, the leaderboard score has improved from 9.76 to 9.46.

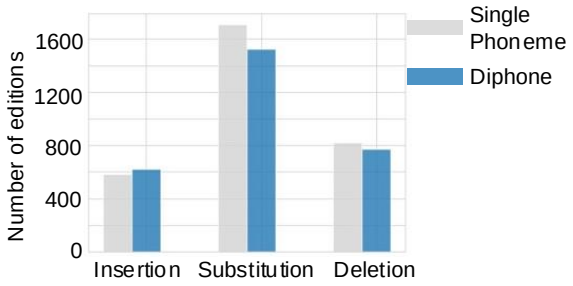

Figure 6.5: Phoneme error types analysis during single phoneme decoding and diphone.

6.5.8 Phoneme Error Analysis

We conducted a detailed analysis of the various types of errors encountered during phoneme decoding. This analysis involved assessing the operations necessary to align the decoded phoneme sequence with the ground truth phonemes, comparing scenarios where only single phoneme decoding is used versus employing diphone subclass decoding. Overall, our findings indicate that employing diphone subclass decoding leads to a reduction in the number of operations required to align the decoded sequence with the ground truth phonemes. Specifically, fewer editing operations, particularly substitutions, are needed when utilizing the diphone decoding paradigm compared to directly decoding single phonemes.

6.5.9 Prompt for GPT3.5

Prompt to GPT3.5 : Your task is to perform automatic speech recognition. Below are multiple candidate transcriptions together with their corresponding phoneme representations. The phonemes are taken from the CMU Pronouncing Dictionary. The special symbol SIL represents the start of the sentence, or the end of the sentence, or the space between two adjacent words. Based on the transcription candidates and their phoneme representations, come up with a transcription and its corresponding phoneme representation that are most accurate, ensuring the transcription is contextually and grammatically correct. Focus on key differences in the candidates that change the meaning or correctness. Avoid selections

with repetitive or nonsensical phrases. In cases of ambiguity, select the option that is most coherent and contextually sound, taking clues from the phoneme representations. The candidate phoneme representations may not always be the correct representation of the corresponding candidate transcriptions. Some phonemes in the candidate phoneme sequences might have been incorrectly added, removed, or replaced. However, the candidate phonemes contain useful information that will help you come up with the correct transcription and phoneme representation. You should translate each subgroup of phonemes that is enclosed by two SIL symbols into one single word. You should remove SIL symbols at the start or the end of the phoneme sequence. Respond with your refined transcription and its corresponding phoneme representation only, without any introductory text.

Examples of prediction and correction pairs Transcription candidate 1: but we don't know that. Transcription candidate 2: but we don't know that. Transcription candidate 3: but you don't know that. Transcription candidate 4: but you don't know that. Transcription candidate 5: but you don't know that. Transcription candidate 6: but you don't know that. Transcription candidate 7: but you don't know that. Transcription candidate 8: but you don't know that. Transcription candidate 9: but we don't know that. Transcription candidate 10: but we don't know that. Phoneme candidate 1: SIL B AH T SIL W IY SIL D OW N T SIL N OW SIL DH AE T SIL. Phoneme candidate 2: SIL B AH T SIL Y IY SIL D OW N T SIL N OW SIL DH AE T SIL. Phoneme candidate 3: SIL B AH T SIL Y UW SIL D OW N T SIL N OW SIL AE T SIL. Phoneme candidate 4: SIL B AH T SIL Y UW SIL D OW N T SIL N OW SIL DH AE T SIL. Phoneme candidate 5: SIL B AH T SIL DH UW SIL D OW N T SIL N OW SIL DH AE T SIL. Phoneme candidate 6: SIL B AH T SIL Y UW SIL D OW N T SIL N OW SIL DH AE T SIL. Phoneme candidate 7: SIL B AH T SIL Y UW SIL D OW N T SIL N OW SIL DH AE T SIL. Phoneme candidate 8: SIL B AH T SIL Y UW SIL D OW N T SIL N OW SIL DH AE T SIL. Phoneme candidate 9: SIL B AH T SIL W IY SIL D OW N T SIL N OW Z SIL DH AE T SIL. Phoneme candidate 10: SIL B AH T SIL DH IY SIL D OW N T SIL N OW SIL AE T SIL.

Table 6.6: Example of In-Context-Learning (ICL) prompts and query.

System Prompt:	Your task is to perform automatic speech recognition. You are given ten candidates of an unknown transcription. Your job is to come up with a transcription that is most accurate, relying on the context that the candidates provide. First, observe the provided examples demonstrating how the task should be done, then work on the query candidates. In each example, ten transcription candidates, their corresponding phoneme representations, and a ground truth transcription are given. The ground truth transcription is the correct transcription, while the transcription candidates and phoneme representations may or may not contain errors. Some phonemes in the phoneme sequences might have been incorrectly added, removed, or replaced. However, the phonemes contain helpful information that will help you come up with the correct transcription. You should translate each subgroup of phonemes that is enclosed by two SIL symbols into one single word. You should remove SIL symbols at the start and the end of the phoneme sequence. Make sure your transcription based on the query candidates is contextually and grammatically correct. Focus on key differences in the candidates that change the meaning or correctness. Avoid selections with repetitive or nonsensical phrases. In cases of ambiguity, select the option that is most coherent and contextually sound. Respond with your final transcription only, without any introductory text.

Context prompt:	**Example 1**: *Transcription candidate 1*: i enjoyed it very much. ⋯ *Transcription candidate 10:* i enjoyed it very much. *Phoneme candidate 1:* AY SIL EH N JH OY D SIL IH T SIL V EH R IY SIL M AH CH SIL. ⋯ *Phoneme candidate 10:* AY SIL EH N JH OY D SIL IH T SIL V EH R IY SIL M AH CH SIL. ⋯ **Ground truth phonemes:** AY SIL EH N JH OY D SIL IH T SIL V EH R IY SIL M AH CH. **Ground truth transcription**: i enjoyed it very much. ⋯ **Example N**: *Transcription candidate 1:* the ranks of asian riders are falling too. ⋯ *Transcription candidate 10:* the ranks of asian riders are willing to. *Phoneme candidate 1*: DH AH SIL R AE NG K S SIL AH V SIL EY ZH AH N SIL R AY D Z SIL AA R SIL F L D IH NG SIL T UW SIL. ⋯ *Phoneme candidate 10*: DH AH SIL R AE K S SIL AH V SIL EY ZH AH N SIL R EY D ER Z SIL AA R SIL F IY L IH NG SIL T UW SIL. **Ground truth phonemes**: DH AH SIL R AE NG K S SIL AH V SIL EY ZH AH N SIL R AY D ER Z SIL AA R SIL S W EH L IH NG SIL T UW. **Ground truth transcription**: the ranks of asian riders are swelling too
Query:	Transcription candidate 1: i'm originally from colorado. ⋯ Transcription candidate 10: i'm only from colorado. Phoneme candidate 1: SIL AY M SIL ER N AH L IY SIL F R AH M SIL K AO L ER AA D OW SIL. ⋯ Phoneme candidate 10: SIL AY M SIL AH N L IY SIL F R AH M SIL K AO L R AA D OW SIL.

Chapter 7

CONCLUSION

In this dissertation, I have developed a series of methods to learn representations from neural population activity that remain robust under variability across scales—from stochastic fluctuations on single trials, to changes in the composition of recorded populations across sessions, to context-dependent changes in neural encoding across behavioral tasks. Across four studies, I developed architectures and training strategies that incorporate inductive biases reflecting the properties of individual neurons and other objectives encoded by the populations. Together, these contributions advance a representation-learning perspective for systems neuroscience and brain–computer interfaces (BCIs), in which models are designed not only to serve the behavior decoding objective, but also to extract stable abstractions of neural processes that would be useful for later downstream tasks.

In the first line of work, I introduced STNDT—a spatiotemporal transformer architecture for modeling single-trial population dynamics. By explicitly attending over neurons and time, this model captures both the coordination between neurons and the temporal dependencies at the population level, improving upon recurrent and latent-state baselines in reconstructing the underlying firing rates, predicting held-out neurons and future activity, and supporting accurate behavior decoding. The formulation demonstrates that population-level structures can be effectively captured by attention mechanisms without the need of behavior labels.

The second line of work focused on learning time-invariant representations for individual neurons from multi-session recordings (NeuPRINT). In this study, we proposed a self-supervised framework that embeds each neuron into an identity space informed by its dynamical relationships with the surrounding population. These embeddings are designed to be stable across sessions and robust to changes in overlapping populations, enabling the

recovery of transcriptomic or functional classes with lightweight downstream classifiers. This study provides evidence that neuronal identity can be inferred from *in vivo* activity patterns and population context, reducing reliance on expensive joint molecular profiling and offering a path toward scalable, automated cell type inference.

The third line of work addressed one of the central obstacles to long-term iBCI deployment: session-to-session nonstationarity. We developed SPINT—a spatial permutation-invariant transformer that treats simultaneously recorded units as an unordered set and learns context-dependent identities from a small number of unlabeled calibration trials. This design eliminates the need for explicit channel-wise alignment and model fine-tuning at test time while maintaining strong cross-session decoding performance. The resulting framework shows that appropriately constructed permutation-invariant encoders, combined with few-shot calibration, can advance the robustness of long-term BCIs under the practical constraints of chronic implantation.

Finally, we explored neural decoding in a challenging domain: brain-to-text communication. In this setting, we introduced DCoND-LIFT—a divide-and-conquer decoding framework that maps neural activity to context-aware intermediate phonetic units and then leverages powerful language models to infer coherent texts. This approach exploits the inherent context dependence of neural representations during attempted speech and uses the acoustic context-dependent representations (diphones) as the intermediate targets for speech decoding. The framework illustrates how a carefully selected neural representation inspired by a known neuroscientific phenomenon (coarticulation) can be leveraged to achieve high-performance speech decoding when combined with large language models.

Taken together, these studies support a coherent view: robust and high-performing neural interfaces are achieved when we treat neural population analysis as a problem of representation learning under variability. By explicitly encoding desired invariances—over neurons, over time, and over experimental conditions—into model architectures and training objectives, we obtain representations that transfer across trials, sessions, and contexts more effectively than conventional methods. The techniques developed in this dissertation connect

tools from modern machine learning (self-supervision, attention, permutation invariance, sequence modeling) with concrete demands of systems neuroscience: uncovering latent dynamics, inferring neuronal identity, coping with neural nonstationarity, and decoding complex behaviors.

Despite these strengths, the studies presented in this dissertation have several overarching limitations. First, all four studies are evaluated on relatively constrained datasets, largely single-lab recordings, a limited set of brain areas, and specific motor or speech tasks. It remains unclear how far the learned representations can generalize across modalities, subjects, species, and behavioral tasks. Second, the models are trained and assessed primarily in offline, open-loop settings, optimizing reconstruction or decoding objectives on curated datasets. They have not yet captured the coupled adaptation between the brain, behavior, and algorithm that dominate real-world neural interfaces. Third, the forms of invariance we encourage across trials, sessions, and channels are largely encoded through hand-designed architectural choices and training objectives, rather than emerging from large-scale, heterogeneous pretraining.

These overarching limitations point to several promising directions for future research. A natural next step is to scale these representation-learning frameworks into *foundation models for neuroscience* trained on diverse, multi-area, multi-task, and multi-modal datasets, with objectives that encourage transfer across trials, sessions, subjects, and species. Such models could provide standardized latent spaces or neuron embeddings that serve as reusable priors for downstream analyses, including rapid decoder initialization and cell-type inference, potentially reducing the dependence on extensive task-specific supervision [Ref26, Ref167, Ref168, Ref169]. In parallel, the clinical and scientific impact of robust decoders will ultimately depend on *closed-loop evaluation* beyond the offline benchmarks emphasized in this dissertation. Future work should test whether these invariance-driven representations sustain performance under real-time feedback, user adaptation, in pathological subjects, and whether they can support stable decoding with minimal recalibration over months to years [Ref170]. Bridging large-scale pretraining on diverse and heterogeneous neural data with closed-loop studies would strengthen the adoption of these methods for real-world iBCIs under the coupled dynamics

of brain, behavior, and algorithm.

In conclusion, this dissertation demonstrates that learning structured, invariant representations from neural population dynamics offers an effective algorithmic strategy for addressing neural variability across scales. By bridging modern deep learning with domain-specific constraints of neural data, the proposed methods provide practical contributions: they illustrate how to design models that are simultaneously expressive, robust, and label-efficient, and they introduce concrete tools that advance brain–computer interfaces under realistic clinical constraints. Looking forward, these ideas open avenues for future work that scales representation learning to larger and more diverse datasets and evaluates them rigorously in closed-loop settings, with the goal of capturing multiple sources of neural variability in a data-driven manner and ultimately transforming insights gained from offline analysis into reliable neural interfaces for restoring and enhancing human function.

TRUNG LE'S PUBLICATIONS

[TL1] Trung Le and Eli Shlizerman. Stndt: Modeling neural population activity with spatiotemporal transformers. *Advances in Neural Information Processing Systems*, 35:17926–17939, 2022.

[TL2] Lu Mi*, Trung Le*, Tianxing He, Eli Shlizerman, and Uygar Sümbül. Learning time-invariant representations for individual neurons from population dynamics. *Advances in Neural Information Processing Systems*, 36:46007–46026, 2023.

[TL3] 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. *Advances in Neural Information Processing Systems*, 37, 2025.

[TL4] Jingyuan Li*, Trung Le*, Chaofei Fan, Mingfei Chen, and Eli Shlizerman. Brain-to-text decoding with context-aware neural representations and large language models. *Journal of Neural Engineering*, 22(5):056026, 2025.

[TL5] Ziyu Lu*, Wuwei Zhang*, Trung Le, Hao Wang, Uygar Sümbül, Eric Todd SheaBrown, and Lu Mi. Netformer: An interpretable model for recovering dynamical connectivity in neuronal population dynamics. In *The Thirteenth International Conference on Learning Representations*, 2025.

[TL6] Jingyuan Li, Leo Scholl, Trung Le, Pavithra Rajeswaran, Amy Orsborn, and Eli Shlizerman. Amag: Additive, multiplicative and adaptive graph neural network for forecasting neuron activity. *Advances in Neural Information Processing Systems*, 36, 2024.

[TL7] Ling-Chi Yang, Chi-Jui Chen, Trung Le, Scott Hauck, Shih-Chieh Hsu, and Bo-Cheng Lai. Bram-aware quantization for efficient transformer inference via a tile-based architecture on a fpga. *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 33, 2024.

[TL8] Hao Fang, Trung Le, Chijui Chen, Jingyuan Li, Eli Shlizerman, Bo-Cheng Lai, and Amy L Orsborn. Toward lightweight and fast inference neural decoder design using quantization-aware training: A simulation study. *Authorea Preprints*.

[TL9] Jingyuan Li, Trung Le, and Eli Shlizerman. Al-sar: Active learning for skeleton-based action recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[TL10] Francis R Willett, Jingyuan Li, Trung Le, Chaofei Fan, Mingfei Chen, Eli Shlizerman, Yue Chen, Xin Zheng, Tatsuo S Okubo, Tyler Benster, et al. Brain-to-text benchmark'24: Lessons learned. *arXiv preprint arXiv:2412.17227*, 2024.

[TL11] Yizi Zhang, Linyang He, Chaofei Fan, Tingkai Liu, Han Yu, Trung Le, Jingyuan Li, Scott Linderman, Lea Duncker, Francis R Willett, et al. Decoding inner speech with an end-to-end brain-to-text neural interface. *arXiv preprint arXiv:2511.21740*, 2025.

BIBLIOGRAPHY

[Ref1] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500, 1952.

[Ref2] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

[Ref3] Apostolos P Georgopoulos, Andrew B Schwartz, and Ronald E Kettner. Neuronal population coding of movement direction. *Science*, 233(4771):1416–1419, 1986.

[Ref4] Francesco Randi, Anuj K Sharma, Sophie Dvali, and Andrew M Leifer. Neural signal propagation atlas of caenorhabditis elegans. *Nature*, 623(7986):406–414, 2023.

[Ref5] 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.

[Ref6] Saskia EJ de Vries, Jerome A Lecoq, Michael A Buice, Peter A Groblewski, Gabriel K Ocker, Michael Oliver, David Feng, Nicholas Cain, Peter Ledochowitsch, Daniel Millman, et al. A large-scale standardized physiological survey reveals functional organization of the mouse visual cortex. *Nature neuroscience*, 23(1):138–151, 2020.

[Ref7] Stephane Bugeon, Joshua Duffield, Mario Dipoppa, Anne Ritoux, Isabelle Prankerd, Dimitris Nicoloutsopoulos, David Orme, Maxwell Shinn, Han Peng, Hamish Forrest,

et al. A transcriptomic axis predicts state modulation of cortical interneurons. *Nature*, 607(7918):330–338, 2022.

[Ref8] Krishna V Shenoy, Maneesh Sahani, and Mark M Churchland. Cortical control of arm movements: a dynamical systems perspective. *Annual review of neuroscience*, 36(1):337–359, 2013.

[Ref9] Shreya Saxena and John P Cunningham. Towards the neural population doctrine. *Current opinion in neurobiology*, 55:103–111, 2019.

[Ref10] Saurabh Vyas, Matthew D Golub, David Sussillo, and Krishna V Shenoy. Computation through neural population dynamics. *Annual review of neuroscience*, 43(1):249–275, 2020.

[Ref11] David Raposo, Matthew T Kaufman, and Anne K Churchland. A category-free neural population supports evolving demands during decision-making. *Nature neuroscience*, 17(12):1784–1792, 2014.

[Ref12] Sam McKenzie, Christopher S Keene, Anja Farovik, John Bladon, Ryan Place, Robert Komorowski, and Howard Eichenbaum. Representation of memories in the cortical–hippocampal system: Results from the application of population similarity analyses. *Neurobiology of Learning and Memory*, 134:178–191, 2016.

[Ref13] Howard Eichenbaum. Barlow versus hebb: When is it time to abandon the notion of feature detectors and adopt the cell assembly as the unit of cognition? *Neuroscience letters*, 680:88–93, 2018.

[Ref14] 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.

[Ref15] Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Advances in neural information processing systems*, 21, 2008.

[Ref16] Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. Bayesian learning and inference in recurrent switching linear dynamical systems. In *Artificial Intelligence and Statistics*, pages 914–922. PMLR, 2017.

[Ref17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Ref18] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[Ref19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Ref20] 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, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.

[Ref21] Wei-Hsien Lee, Brianna M Karpowicz, Chethan Pandarinath, and Adam G Rouse. Identifying distinct neural features between the initial and corrective phases of precise reaching using autolfads. *Journal of Neuroscience*, 44(20), 2024.

[Ref22] Shreya Saxena, Abigail A Russo, John Cunningham, and Mark M Churchland. Motor cortex activity across movement speeds is predicted by network-level strategies for generating muscle activity. *Elife*, 11:e67620, 2022.

[Ref23] 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, 2021.

[Ref24] 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, et al. A high-performance speech neuroprosthesis. *Nature*, 620(7976):1031–1036, 2023.

[Ref25] Joel Ye and Chethan Pandarinath. Representation learning for neural population activity with neural data transformers. *arXiv preprint arXiv:2108.01210*, 2021.

[Ref26] Mehdi Azabou, Vinam Arora, Venkataramana Ganesh, Ximeng Mao, Santosh Nachimuthu, Michael Mendelson, Blake Richards, Matthew Perich, Guillaume Lajoie, and Eva Dyer. A unified, scalable framework for neural population decoding. *Advances in Neural Information Processing Systems*, 36:44937–44956, 2023.

[Ref27] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Ref28] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[Ref29] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[Ref30] 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, 2013.

[Ref31] 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, 2017.

[Ref32] David A Moses, Sean L Metzger, Jessie R Liu, Gopala K Anumanchipalli, Joseph G Makin, Pengfei F Sun, Josh Chartier, Maximilian E Dougherty, Patricia M Liu, Gary M Abrams, et al. Neuroprosthesis for decoding speech in a paralyzed person with anarthria. *New England Journal of Medicine*, 385(3):217–227, 2021.

[Ref33] 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(1):11–19, 2022.

[Ref34] Laura N Driscoll, Noah L Pettit, Matthias Minderer, Selmaan N Chettih, and Christopher D Harvey. Dynamic reorganization of neuronal activity patterns in parietal cortex. *Cell*, 170(5):986–999, 2017.

[Ref35] 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 *NeurIPS Workshop on Symmetry and Geometry in Neural Representations*, pages 234–257. PMLR, 2023.

[Ref36] 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, 2014.

[Ref37] Robert M Dowben and Jerzy E Rose. A metal-filled microelectrode. *Science*, 118(3053):22–24, 1953.

[Ref38] 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.

[Ref39] 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, et al. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551(7679):232–236, 2017.

[Ref40] Karel Svoboda, Winfried Denk, David Kleinfeld, and David W Tank. In vivo dendritic calcium dynamics in neocortical pyramidal neurons. *Nature*, 385(6612):161–165, 1997.

[Ref41] John P Cunningham and Byron M Yu. Dimensionality reduction for large-scale neural recordings. *Nature neuroscience*, 17(11):1500–1509, 2014.

[Ref42] Morris W Hirsch, Stephen Smale, and Robert L Devaney. *Differential equations, dynamical systems, and an introduction to chaos*. Academic press, 2013.

[Ref43] John Guckenheimer and Philip Holmes. *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, volume 42. Springer Science & Business Media, 2013.

[Ref44] Stephen Wiggins. *Introduction to applied nonlinear dynamical systems and chaos*. Springer, 2003.

[Ref45] Afsheen Afshar, Gopal Santhanam, M Yu Byron, Stephen I Ryu, Maneesh Sahani, and Krishna V Shenoy. Single-trial neural correlates of arm movement preparation. *Neuron*, 71(3):555–564, 2011.

[Ref46] Christopher D Harvey, Philip Coen, and David W Tank. Choice-specific sequences in parietal cortex during a virtual-navigation decision task. *Nature*, 484(7392):62–68, 2012.

[Ref47] Federico Carnevale, Victor de Lafuente, Ranulfo Romo, Omri Barak, and Néstor Parga. Dynamic control of response criterion in premotor cortex during perceptual detection under temporal uncertainty. *Neuron*, 86(4):1067–1077, 2015.

[Ref48] Liam Paninski, Yashar Ahmadian, Daniel Gil Ferreira, Shinsuke Koyama, Kamiar Rahnama Rad, Michael Vidne, Joshua Vogelstein, and Wei Wu. A new look at state-space models for neural data. *Journal of computational neuroscience*, 29(1):107–126, 2010.

[Ref49] Jakob H Macke, Lars Buesing, John P Cunningham, Byron M Yu, Krishna V Shenoy, and Maneesh Sahani. Empirical models of spiking in neural populations. *Advances in neural information processing systems*, 24, 2011.

[Ref50] 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. *Nature communications*, 6(1):1–12, 2015.

[Ref51] Yuanjun Gao, Evan W Archer, Liam Paninski, and John P Cunningham. Linear dynamical neural population models through nonlinear embeddings. *Advances in neural information processing systems*, 29, 2016.

[Ref52] Marine Schimel, Ta-Chu Kao, Kristopher T Jensen, and Guillaume Hennequin. ilqr-vae: control-based learning of input-driven dynamics with applications to neural data. *bioRxiv*, 2021.

[Ref53] Michael Nolan, Bijan Pesaran, Eli Shlizerman, and Amy Orsborn. Multi-block rnn autoencoders enable broadband ecog signal reconstruction. *bioRxiv*, pages 2022–09, 2022.

[Ref54] Kun Su, Xiulong Liu, and Eli Shlizerman. Predict & cluster: Unsupervised skeleton based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9631–9640, 2020.

[Ref55] Stefano Recanatesi, Matthew Farrell, Madhu Advani, Timothy Moore, Guillaume Lajoie, and Eric Shea-Brown. Dimensionality compression and expansion in deep neural networks. *arXiv preprint arXiv:1906.00443*, 2019.

[Ref56] S Recanatesi, M Farrell, G Lajoie, S Deneve, M Rigotti, and E Shea-Brown. Predictive learning as a network mechanism for extracting low-dimensional latent space representations. nat commun 12: 1417, 2021.

[Ref57] 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.

[Ref58] Gamaleldin F Elsayed, Antonio H Lara, Matthew T Kaufman, Mark M Churchland, and John P Cunningham. Reorganization between preparatory and movement population responses in motor cortex. *Nature communications*, 7(1):13239, 2016.

[Ref59] Juan A Gallego, Matthew G Perich, Lee E Miller, and Sara A Solla. Neural manifolds for the control of movement. *Neuron*, 94(5):978–984, 2017.

[Ref60] 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.

[Ref61] David M Brandman, Sydney S Cash, and Leigh R Hochberg. human intracortical recording and neural decoding for brain–computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(10):1687–1696, 2017.

[Ref62] 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.

[Ref63] Leigh R Hochberg, Daniel Bacher, Beata Jarosiewicz, Nicolas Y Masse, John D Simeral, Joern Vogel, Sami Haddadin, Jie Liu, Sydney S Cash, Patrick Van Der Smagt, et al. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485(7398):372–375, 2012.

[Ref64] Vikash Gilja, Paul Nuyujukian, Cindy A Chestek, John P Cunningham, Byron M Yu, Joline M Fan, Mark M Churchland, Matthew T Kaufman, Jonathan C Kao, Stephen I Ryu, et al. A high-performance neural prosthesis enabled by control algorithm design. *Nature neuroscience*, 15(12):1752–1757, 2012.

[Ref65] 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.

[Ref66] 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, 2023.

[Ref67] 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. *Nature biomedical engineering*, 4(7):672–685, 2020.

[Ref68] 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. *Nature biomedical engineering*, 1(12):967–976, 2017.

[Ref69] 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. *arXiv preprint arXiv:1810.00045*, 2018.

[Ref70] 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, 2023.

[Ref71] Brianna M Karpowicz, Yahia H Ali, Lahiru N Wimalasena, Andrew R Sedler, Mohammad Reza Keshtkaran, Kevin Bodkin, Xuan Ma, Lee E Miller, and Chethan Pandarinath. Stabilizing brain-computer interfaces through alignment of latent dynamics. *BioRxiv*, pages 2022–04, 2022.

[Ref72] 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, 2023.

[Ref73] Ayesha Vermani, Il Memming Park, and Josue Nassar. Leveraging generative models for unsupervised alignment of neural time series data. In *The Twelfth International Conference on Learning Representations*, 2023.

[Ref74] 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*, 36:38988–39005, 2023.

[Ref75] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[Ref76] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

[Ref77] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

[Ref78] Toan Q Nguyen and Julian Salazar. Transformers without tears: Improving the normalization of self-attention. *arXiv preprint arXiv:1910.05895*, 2019.

[Ref79] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

[Ref80] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[Ref81] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[Ref82] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[Ref83] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[Ref84] Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. *arXiv preprint arXiv:2305.16938*, 2023.

[Ref85] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

[Ref86] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[Ref87] Hengyuan Xu, Liyao Xiang, Hangyu Ye, Dixi Yao, Pengzhi Chu, and Baochun Li. Permutation equivariance of transformers and its applications. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5987–5996, 2024.

[Ref88] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pages 3744–3753. PMLR, 2019.

[Ref89] Rafael Yuste. From the neuron doctrine to neural networks. *Nature reviews neuroscience*, 16(8):487–497, 2015.

[Ref90] Krishna V Shenoy and Jonathan C Kao. Measurement, manipulation and modeling of brain-wide neural population dynamics. *Nature Communications*, 12(1):1–5, 2021.

[Ref91] Beata Jarosiewicz, Anish A Sarma, Daniel Bacher, Nicolas Y Masse, John D Simeral, Brittany Sorice, Erin M Oakley, Christine Blabe, Chethan Pandarinath, Vikash Gilja, et al. Virtual typing by people with tetraplegia using a self-calibrating intracortical brain-computer interface. *Science translational medicine*, 7(313):313ra179–313ra179, 2015.

[Ref92] Rafael Levi, Pablo Varona, Yuri I Arshavsky, Mikhail I Rabinovich, and Allen I Selverston. The role of sensory network dynamics in generating a motor program. *Journal of Neuroscience*, 25(42):9807–9815, 2005.

[Ref93] Miguel AL Nicolelis, Luiz A Baccala, Rick CS Lin, and John K Chapin. Sensorimotor encoding by synchronous neural ensemble activity at multiple levels of the somatosensory system. *Science*, 268(5215):1353–1358, 1995.

[Ref94] Eli Shlizerman, Konrad Schroder, and J Nathan Kutz. Neural activity measures and their dynamics. *SIAM Journal on Applied Mathematics*, 72(4):1260–1291, 2012.

[Ref95] 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. *bioRxiv*, 2021.

[Ref96] Feng Zhu, Andrew Sedler, Harrison A Grier, Nauman Ahad, Mark Davenport, Matthew Kaufman, Andrea Giovannucci, and Chethan Pandarinath. Deep inference of latent dynamics with spatio-temporal super-resolution using selective backpropagation through time. *Advances in Neural Information Processing Systems*, 34, 2021.

[Ref97] Margaret Yvonne Mahan and Apostolos P Georgopoulos. Motor directional tuning across brain areas: directional resonance and the role of inhibition for directional accuracy. *Frontiers in neural circuits*, 7:92, 2013.

[Ref98] Adam Kohn, Ruben Coen-Cagli, Ingmar Kanitscheider, and Alexandre Pouget. Correlations and neuronal population information. *Annual review of neuroscience*, 39:237–256, 2016.

[Ref99] Bruno B Averbeck, Peter E Latham, and Alexandre Pouget. Neural correlations, population coding and computation. *Nature reviews neuroscience*, 7(5):358–366, 2006.

[Ref100] 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, et al. Neural latents benchmark'21: Evaluating latent variable models of neural population activity. *arXiv preprint arXiv:2109.04463*, 2021.

[Ref101] Ran Liu, Mehdi Azabou, Max Dabagia, Chi-Heng Lin, Mohammad Gheshlaghi Azar, Keith Hengen, Michal Valko, and Eva Dyer. Drop, swap, and generate: A self-

supervised approach for generating neural activity. *Advances in Neural Information Processing Systems*, 34:10587–10599, 2021.

[Ref102] Darin Sleiter, Joshua Schoenfield, and Mike Vaiana. ae-nlb-2021. `https://github.com/agencyenterprise/ae-nlb-2021.git`, 2021.

[Ref103] Sean Perkins. Mint: Mesh of idealized neural trajectories. `https://github.com/neurallatents/nlb_workshop/blob/main/MINT.pdf`, 2022.

[Ref104] Joseph G Makin, Joseph E O'Doherty, Mariana MB Cardoso, and Philip N Sabes. Superior arm-movement decoding from cortex with a new, unsupervised-learning algorithm. *Journal of neural engineering*, 15(2):026010, 2018.

[Ref105] Raeed H Chowdhury, Joshua I Glaser, and Lee E Miller. Area 2 of primary somatosensory cortex encodes kinematics of the whole arm. *Elife*, 9, 2020.

[Ref106] Mattia Rigotti, Omri Barak, Melissa R Warden, Xiao-Jing Wang, Nathaniel D Daw, Earl K Miller, and Stefano Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497(7451):585–590, 2013.

[Ref107] Hansem Sohn, Devika Narain, Nicolas Meirhaeghe, and Mehrdad Jazayeri. Bayesian computation through cortical latent dynamics. *Neuron*, 103(5):934–947, 2019.

[Ref108] David Marr. *Vision: A computational investigation into the human representation and processing of visual information*. MIT press, 2010.

[Ref109] Hannah Bos, Anne-Marie Oswald, and Brent Doiron. Untangling stability and gain modulation in cortical circuits with multiple interneuron classes. *bioRxiv*, pages 2020–06, 2020.

[Ref110] Yuhan Helena Liu, Stephen Smith, Stefan Mihalas, Eric Shea-Brown, and Uygar Sümbül. Cell-type–specific neuromodulation guides synaptic credit assignment in a spiking neural network. *Proceedings of the National Academy of Sciences*, 118(51):e2111821118, 2021.

[Ref111] Anirban Paul, Megan Crow, Ricardo Raudales, Miao He, Jesse Gillis, and Z Josh Huang. Transcriptional architecture of synaptic communication delineates gabaergic neuron identity. *Cell*, 171(3):522–539, 2017.

[Ref112] Nathan W Gouwens, Staci A Sorensen, Fahimeh Baftizadeh, Agata Budzillo, Brian R Lee, Tim Jarsky, Lauren Alfiler, Katherine Baker, Eliza Barkan, Kyla Berry, et al. Integrated morphoelectric and transcriptomic classification of cortical gabaergic cells. *Cell*, 183(4):935–953, 2020.

[Ref113] Federico Scala, Dmitry Kobak, Matteo Bernabucci, Yves Bernaerts, Cathryn René Cadwell, Jesus Ramon Castro, Leonard Hartmanis, Xiaolong Jiang, Sophie Laturnus, Elanine Miranda, et al. Phenotypic variation of transcriptomic cell types in mouse motor cortex. *Nature*, 598(7879):144–150, 2021.

[Ref114] Rohan Gala, Nathan Gouwens, Zizhen Yao, Agata Budzillo, Osnat Penn, Bosiljka Tasic, Gabe Murphy, Hongkui Zeng, and Uygar Sümbül. A coupled autoencoder approach for multi-modal analysis of cell types. *Advances in Neural Information Processing Systems*, 32, 2019.

[Ref115] Dmitry Kobak, Yves Bernaerts, Marissa A Weis, Federico Scala, Andreas S Tolias, and Philipp Berens. Sparse reduced-rank regression for exploratory visualisation of paired multivariate data. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 70(4):980–1000, 2021.

[Ref116] Yeganeh Marghi, Rohan Gala, Fahimeh Baftizadeh, and Uygar Sumbul. Joint inference of discrete cell types and continuous type-specific variability in single-cell datasets with mmidas. *bioRxiv*, pages 2023–10, 2023.

[Ref117] Jennifer S Lund, Alessandra Angelucci, and Paul C Bressloff. Anatomical substrates for functional columns in macaque monkey primary visual cortex. *Cerebral cortex*, 13(1):15–24, 2003.

[Ref118] Robert B Levy and Alex D Reyes. Spatial profile of excitatory and inhibitory synaptic connectivity in mouse primary auditory cortex. *Journal of Neuroscience*, 32(16):5609–5619, 2012.

[Ref119] Robert Rosenbaum, Matthew A Smith, Adam Kohn, Jonathan E Rubin, and Brent Doiron. The spatial structure of correlated neuronal variability. *Nature neuroscience*, 20(1):107–114, 2017.

[Ref120] Eric Jonas and Konrad Kording. Automatic discovery of cell types and microcircuitry from neural connectomics. *Elife*, 4:e04250, 2015.

[Ref121] Jakub M Tomczak, Maximilian Ilse, and Max Welling. Deep learning with permutation-invariant operator for multi-instance histopathology classification. *arXiv preprint arXiv:1712.00310*, 2017.

[Ref122] Stephen W Kuffler. Discharge patterns and functional organization of mammalian retina. *Journal of neurophysiology*, 16(1):37–68, 1953.

[Ref123] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.

[Ref124] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[Ref125] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Ref126] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[Ref127] Aidan Schneider, Mehdi Azabou, Louis McDougall-Vigier, David F Parks, Sahara Ensley, Kiran Bhaskaran-Nair, Tomasz Nowakowski, Eva L Dyer, and Keith B Hengen. Transcriptomic cell type structures in vivo neuronal activity across multiple timescales. *Cell Reports*, 42(4), 2023.

[Ref128] Marius Pachitariu, Carsen Stringer, Sylvia Schröder, Mario Dipoppa, L Federico Rossi, Matteo Carandini, and Kenneth D Harris. Suite2p: beyond 10,000 neurons with standard two-photon microscopy. *BioRxiv*, page 061507, 2016.

[Ref129] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018.

[Ref130] Rohan Gala, Agata Budzillo, Fahimeh Baftizadeh, Jeremy Miller, Nathan Gouwens, Anton Arkhipov, Gabe Murphy, Bosiljka Tasic, Hongkui Zeng, Michael Hawrylycz, et al. Consistent cross-modal identification of cortical neurons with coupled autoencoders. *Nature computational science*, 1(2):120–127, 2021.

[Ref131] 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, et al. 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.

[Ref132] 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.

[Ref133] 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.

[Ref134] 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.

[Ref135] 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.

[Ref136] 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. *Journal of Neural Engineering*, 20(5):056040, 2023.

[Ref137] Joel Ye, Jennifer Collinger, Leila Wehbe, and Robert Gaunt. Neural data transformer 2: multi-context pretraining for neural spiking activity. *Advances in Neural Information Processing Systems*, 36:80352–80374, 2023.

[Ref138] 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, et al. Few-shot algorithms for consistent neural decoding (falcon) benchmark. *bioRxiv*, pages 2024–09, 2024.

[Ref139] Joel Ye, Fabio Rizzoglio, Adam Smoulder, Hongwei Mao, Xuan Ma, Patrick Marino, Raeed Chowdhury, Dalton Moore, Gary Blumenthal, William Hockeimer, et al. A generalist intracortical motor decoder. *bioRxiv*, pages 2025–02, 2025.

[Ref140] Ran Liu, Mehdi Azabou, Max Dabagia, Jingyun Xiao, and Eva Dyer. Seeing the forest and the tree: Building representations of both individual and collective dynamics with transformers. *Advances in Neural Information Processing Systems*, 35:2377–2391, 2022.

[Ref141] Mohammad Reza Keshtkaran and Chethan Pandarinath. Enabling hyperparameter optimization in sequential autoencoders for spiking neural data. *Advances in neural information processing systems*, 32, 2019.

[Ref142] 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. *Nature Methods*, pages 1–6, 2022.

[Ref143] 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, 2015.

[Ref144] 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, 2016.

[Ref145] 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.

[Ref146] Adam G Rouse and Marc H Schieber. Condition-dependent neural dimensions progressively shift during reach to grasp. *Cell reports*, 25(11):3158–3168, 2018.

[Ref147] 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, 2021.

[Ref148] B Wodlinger, JE Downey, EC Tyler-Kabara, AB Schwartz, ML Boninger, and JL Collinger. Ten-dimensional anthropomorphic arm control in a human brain- machine interface: difficulties, solutions, and limitations. *Journal of neural engineering*, 12(1):016011, 2014.

[Ref149] 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):831–836, 2021.

[Ref150] EvalAI. Falcon benchmark challenge. `https://eval.ai/web/challenges/challenge-page/2319/evaluation`, 2025. Accessed: 2025-05-12.

[Ref151] Norbert Wiener. *Extrapolation, interpolation, and smoothing of stationary time series*. The MIT press, 1964.

[Ref152] 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*, 9:217, 2015.

[Ref153] 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, et al. A high-performance neuroprosthesis for speech decoding and avatar control. *Nature*, pages 1–10, 2023.

[Ref154] Kristofer E Bouchard and Edward F Chang. Neural decoding of spoken vowels from human sensory-motor cortex with high-density electrocorticography. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6782–6785. IEEE, 2014.

[Ref155] Emily M Mugler, James L Patton, Robert D Flint, Zachary A Wright, Stephan U Schuele, Joshua Rosenow, Jerry J Shih, Dean J Krusienski, and Marc W Slutzky. Direct classification of all american english phonemes using signals from functional speech motor cortex. *Journal of neural engineering*, 11(3):035015, 2014.

[Ref156] Jon P Nedel, Rita Singh, and Richard M Stern. Phone transition acoustic modeling: application to speaker independent and spontaneous speech systems. In *INTERSPEECH*, pages 572–575, 2000.

[Ref157] 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.

[Ref158] Yohann Benchetrit, Hubert Banville, and Jean-Rémi King. Brain decoding: toward real-time reconstruction of visual perception. *arXiv preprint arXiv:2310.19812*, 2023.

[Ref159] 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.

[Ref160] Yajie Miao, Mohammad Gowayyed, and Florian Metze. Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding. In *2015 IEEE workshop on automatic speech recognition and understanding (ASRU)*, pages 167–174. IEEE, 2015.

[Ref161] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[Ref162] 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, et al. Data for: A high-performance speech neuroprosthesis [dataset]. *Dryad*, pages 1–6, 2023.

[Ref163] Alexander Epaneshnikov Reece H. Dunn, Valdis Vitolins. espeak ng text-to-speech. *GitHub*.

[Ref164] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. arxiv (2022). *arXiv preprint arXiv:2212.04356*, 2022.

[Ref165] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[Ref166] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[Ref167] Mehdi Azabou, Krystal Xuejing Pan, Vinam Arora, Ian Jarratt Knight, Eva L Dyer, and Blake Aaron Richards. Multi-session, multi-task neural decoding from distinct cell-types and brain regions. In *The Thirteenth International Conference on Learning Representations*.

[Ref168] Mehdi Azabou, Vinam Arora, Venkataramana Ganesh, Ximeng Mao, Santosh Nachimuthu, Michael Mendelson, Blake Richards, Matthew Perich, Guillaume Lajoie, and Eva Dyer. A unified, scalable framework for neural population decoding. *Advances in Neural Information Processing Systems*, 36, 2024.

[Ref169] Yizi Zhang, Yanchen Wang, Mehdi Azabou, Alexandre Andre, Zixuan Wang, Hanrui Lyu, The International Brain Laboratory, Eva Dyer, Liam Paninski, and Cole Hurwitz. Neural encoding and decoding at scale. *arXiv preprint arXiv:2504.08201*, 2025.

[Ref170] 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, 2014.

[Ref171] Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

[Ref172] Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*, 2021.

[Ref173] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

[Ref174] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

[Ref175] Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and stability in in-context learning. In *International Conference on Machine Learning*, pages 19565–19594. PMLR, 2023.

[Ref176] Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR, 2023.

[Ref177] Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

[Ref178] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

[Ref179] Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *arXiv preprint arXiv:2306.09927*, 2023.

[Ref180] Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in neural information processing systems*, 36, 2024.

[Ref181] Licong Lin, Yu Bai, and Song Mei. Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining. *arXiv preprint arXiv:2310.08566*, 2023.

[Ref182] Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *Advances in Neural Information Processing Systems*, 36:45614–45650, 2023.

[Ref183] Allan Raventós, Mansheej Paul, Feng Chen, and Surya Ganguli. Pretraining task diversity and the emergence of non-bayesian in-context learning for regression. *Advances in Neural Information Processing Systems*, 36, 2024.

[Ref184] Ryuichiro Hataya, Kota Matsui, and Masaaki Imaizumi. Automatic domain adaptation by transformers in in-context learning. *arXiv preprint arXiv:2405.16819*, 2024.

[Ref185] Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers. *arXiv preprint arXiv:2212.10559*, 2022.

[Ref186] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

[Ref187] Aaditya Singh, Stephanie Chan, Ted Moskovitz, Erin Grant, Andrew Saxe, and Felix Hill. The transient nature of emergent in-context learning in transformers. *Advances in Neural Information Processing Systems*, 36, 2024.

[Ref188] Ekin Akyürek, Bailin Wang, Yoon Kim, and Jacob Andreas. In-context language learning: Arhitectures and algorithms. *arXiv preprint arXiv:2401.12973*, 2024.

[Ref189] Apostolos P Georgopoulos, John F Kalaska, Roberto Caminiti, and Joe T Massey. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *Journal of Neuroscience*, 2(11):1527–1537, 1982.

[Ref190] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[Ref191] James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. Explainable prediction of medical codes from clinical text. *arXiv preprint arXiv:1802.05695*, 2018.

[Ref192] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*, 2019.

[Ref193] Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the dark secrets of bert. *arXiv preprint arXiv:1908.08593*, 2019.

[Ref194] Yongjie Lin, Yi Chern Tan, and Robert Frank. Open sesame: getting inside bert's linguistic knowledge. *arXiv preprint arXiv:1906.01698*, 2019.

[Ref195] Hamidreza Ghader and Christof Monz. What does attention in neural machine translation pay attention to? *arXiv preprint arXiv:1710.03348*, 2017.

[Ref196] Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. Visualizing and measuring the geometry of bert. *Advances in Neural Information Processing Systems*, 32, 2019.

[Ref197] Elad Schneidman, Michael J Berry, Ronen Segev, and William Bialek. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 440(7087):1007–1012, 2006.

[Ref198] Dmitry R Lyamzin, Jakob H Macke, and Nicholas A Lesica. Modeling population spike trains with specified time-varying spike rates, trial-to-trial variability, and pairwise signal and noise correlations. *Frontiers in computational neuroscience*, 4:144, 2010.

[Ref199] Manuel Molano-Mazon, Arno Onken, Eugenio Piasini, and Stefano Panzeri. Synthesizing realistic neural population activity patterns using generative adversarial networks. *arXiv preprint arXiv:1803.00338*, 2018.

[Ref200] Poornima Ramesh, Mohamad Atayi, and Jakob H Macke. Adversarial training of neural encoding models on population spike trains. 2019.

[Ref201] Mohammad Bashiri, Edgar Walker, Konstantin-Klemens Lurz, Akshay Jagadish, Taliah Muhammad, Zhiwei Ding, Zhuokun Ding, Andreas Tolias, and Fabian Sinz. A flow-based latent state generative model of neural population responses to natural images. *Advances in Neural Information Processing Systems*, 34:15801–15815, 2021.

[Ref202] Stephen Keeley, Mikio Aoi, Yiyi Yu, Spencer Smith, and Jonathan W Pillow. Identifying signal and noise structure in neural population activity with gaussian process factor models. *Advances in Neural Information Processing Systems*, 33:13795–13805, 2020.

[Ref203] Kai Chen, Guang Chen, Dan Xu, Lijun Zhang, Yuyao Huang, and Alois Knoll. Nast: non-autoregressive spatial-temporal transformer for time series forecasting. *arXiv preprint arXiv:2102.05624*, 2021.

[Ref204] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *European Conference on Computer Vision*, pages 507–523. Springer, 2020.

[Ref205] Yuren Cong, Wentong Liao, Hanno Ackermann, Bodo Rosenhahn, and Michael Ying Yang. Spatial-temporal transformer for dynamic scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16372–16382, 2021.

[Ref206] Emre Aksan, Manuel Kaufmann, Peng Cao, and Otmar Hilliges. A spatio-temporal transformer for 3d human motion prediction. In *2021 International Conference on 3D Vision (3DV)*, pages 565–574. IEEE, 2021.

[Ref207] Weihuang Chen, Fangfang Wang, and Hongbin Sun. S2tnet: Spatio-temporal transformer networks for trajectory prediction in autonomous driving. In *Asian Conference on Machine Learning*, pages 454–469. PMLR, 2021.

[Ref208] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[Ref209] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[Ref210] Niru Maheswaranathan, Lane T McIntosh, David B Kastner, Josh Melander, Luke Brezovec, Aran Nayebi, Julia Wang, Surya Ganguli, Stephen A Baccus, et al. Deep learning models reveal internal structure and diverse computations in the retina under natural scenes. biorxiv. *URL: https://www. biorxiv. org/content/early/2018/06/14/340943. http://dx. doi. org/10.1101/340943. arXiv: https://www. biorxiv. org/content/early/2018/06/14/340943. full. pdf*, 2018.

[Ref211] Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[Ref212] Tze Hui Koh, William E Bishop, Takashi Kawashima, Brian B Jeon, Ranjani Srinivasan, Yu Mu, Ziqiang Wei, Sandra J Kuhlman, Misha B Ahrens, Steven M Chase, et al. Dimensionality reduction of calcium-imaged neuronal population activity. *Nature Computational Science*, 3(1):71–85, 2023.

[Ref213] Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. Learnable latent embeddings for joint behavioral and neural analysis. *arXiv preprint arXiv:2204.00673*, 2022.

[Ref214] Yuan Zhao and Il Memming Park. Variational latent gaussian process for recovering single-trial dynamics from population spike trains. *Neural computation*, 29(5):1293–1316, 2017.

[Ref215] Yves Bernaerts, Michael Deistler, Pedro J Goncalves, Jonas Beck, Marcel Stimberg, Federico Scala, Andreas S Tolias, Jakob H Macke, Dmitry Kobak, and Philipp Berens. Combined statistical-mechanistic modeling links ion channel genes to physiology of cortical neuron types. *bioRxiv*, pages 2023–03, 2023.

[Ref216] Tsungnan Lin, Bill G Horne, Peter Tino, and C Lee Giles. Learning long-term dependencies in narx recurrent neural networks. *IEEE Transactions on Neural Networks*, 7(6):1329–1338, 1996.

[Ref217] Daniel Durstewitz. A state space approach for piecewise-linear recurrent neural networks for identifying computational dynamics from neural measurements. *PLoS computational biology*, 13(6):e1005542, 2017.

[Ref218] Christine Grienberger and Arthur Konnerth. Imaging calcium in neurons. *Neuron*, 73(5):862–885, 2012.

[Ref219] Jonathan W Pillow, Jonathon Shlens, Liam Paninski, Alexander Sher, Alan M Litke, EJ Chichilnisky, and Eero P Simoncelli. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–999, 2008.

[Ref220] Wilson Truccolo, Uri T Eden, Matthew R Fellows, John P Donoghue, and Emery N Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of neurophysiology*, 93(2):1074–1089, 2005.

[Ref221] Yoshua Bengio and Yann LeCun. Scaling learning algorithms towards AI. In *Large Scale Kernel Machines*. MIT Press, 2007.

[Ref222] Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

[Ref223] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

[Ref224] Yu-Ting Lan, Kan Ren, Yansen Wang, Wei-Long Zheng, Dongsheng Li, Bao-Liang Lu, and Lili Qiu. Seeing through the brain: image reconstruction of visual perception from human brain signals. *arXiv preprint arXiv:2308.02510*, 2023.

[Ref225] Weihao Xia, Raoul de Charette, Cengiz Oztireli, and Jing-Hao Xue. Dream: Visual decoding from reversing human visual system. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8226–8235, 2024.

[Ref226] Jingyuan Sun, Mingxiao Li, Zijiao Chen, and Marie-Francine Moens. Neurocine: Decoding vivid video sequences from human brain activties. *arXiv preprint arXiv:2402.01590*, 2024.

[Ref227] Alexandre Défossez, Charlotte Caucheteux, Jérémy Rapin, Ori Kabeli, and Jean-Rémi King. Decoding speech perception from non-invasive brain recordings. *Nature Machine Intelligence*, 5(10):1097–1107, 2023.

[Ref228] Milán András Fodor, Tamás Gábor Csapó, and Frigyes Viktor Arthur. Towards decoding brain activity during passive listening of speech. *arXiv preprint arXiv:2402.16996*, 2024.

[Ref229] David Sussillo, Rafal Jozefowicz, LF Abbott, and Chethan Pandarinath. Lfads-latent factor analysis via dynamical systems. *arXiv preprint arXiv:1608.06315*, 2016.

[Ref230] 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, 2024.

[Ref231] Sean L Metzger, Jessie R Liu, David A Moses, Maximilian E Dougherty, Margaret P Seaton, Kaylo T Littlejohn, Josh Chartier, Gopala K Anumanchipalli, Adelyn Tu-

Chan, Karunesh Ganguly, et al. Generalizable spelling using a speech neuroprosthesis in an individual with severe limb and vocal paralysis. *Nature communications*, 13(1):6510, 2022.

[Ref232] Spencer Kellis, Kai Miller, Kyle Thomson, Richard Brown, Paul House, and Bradley Greger. Decoding spoken words using local field potentials recorded from the cortical surface. *Journal of neural engineering*, 7(5):056007, 2010.

[Ref233] Xiaomei Pei, Dennis L Barbour, Eric C Leuthardt, and Gerwin Schalk. Decoding vowels and consonants in spoken and imagined words using electrocorticographic signals in humans. *Journal of neural engineering*, 8(4):046028, 2011.

[Ref234] Christian Herff, Lorenz Diener, Miguel Angrick, Emily Mugler, Matthew C Tate, Matthew A Goldrick, Dean J Krusienski, Marc W Slutzky, and Tanja Schultz. Generating natural, intelligible speech from brain activity in motor, premotor, and inferior frontal cortices. *Frontiers in neuroscience*, 13:1267, 2019.

[Ref235] Gopala K Anumanchipalli, Josh Chartier, and Edward F Chang. Speech synthesis from neural decoding of spoken sentences. *Nature*, 568(7753):493–498, 2019.

[Ref236] Guy H Wilson, Sergey D Stavisky, Francis R Willett, Donald T Avansino, Jessica N Kelemen, Leigh R Hochberg, Jaimie M Henderson, Shaul Druckmann, and Krishna V Shenoy. Decoding spoken english from intracortical electrode arrays in dorsal precentral gyrus. *Journal of neural engineering*, 17(6):066007, 2020.

[Ref237] Peter Bell and Steve Renals. Regularization of context-dependent deep neural networks with context-independent multi-task training. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4290–4294. IEEE, 2015.

[Ref238] Erfan Loweimi, Andrea Carmantini, Peter Bell, Steve Renals, and Zoran Cvetkovic. Phonetic error analysis beyond phone error rate. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[Ref239] David Gaddy and Dan Klein. Digital voicing of silent speech. *arXiv preprint arXiv:2010.02960*, 2020.

[Ref240] David Gaddy and Dan Klein. An improved model for voicing silent speech. *arXiv preprint arXiv:2106.01933*, 2021.

[Ref241] David Marshall Gaddy. *Voicing Silent Speech*. University of California, Berkeley, 2022.

[Ref242] Geoffrey S Meltzner, James T Heaton, Yunbin Deng, Gianluca De Luca, Serge H Roy, and Joshua C Kline. Development of semg sensors and algorithms for silent speech recognition. *Journal of neural engineering*, 15(4):046031, 2018.

[Ref243] Szu-Chen Jou, Tanja Schultz, Matthias Walliczek, Florian Kraft, and Alex Waibel. Towards continuous speech recognition using surface electromyography. In *Ninth International Conference on Spoken Language Processing*, 2006.

[Ref244] Tanja Schultz and Michael Wand. Modeling coarticulation in emg-based continuous speech recognition. *Speech Communication*, 52(4):341–353, 2010.

[Ref245] Arnav Kapur, Shreyas Kapur, and Pattie Maes. Alterego: A personalized wearable silent speech interface. In *23rd International conference on intelligent user interfaces*, pages 43–53, 2018.

[Ref246] Lorenz Diener, Gerrit Felsch, Miguel Angrick, and Tanja Schultz. Session-independent array-based emg-to-speech conversion using convolutional neural networks. In *Speech Communication; 13th ITG-Symposium*, pages 1–5. VDE, 2018.

[Ref247] Matthias Janke and Lorenz Diener. Emg-to-speech: Direct generation of speech from facial electromyographic signals. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(12):2375–2385, 2017.

[Ref248] Xupeng Chen, Ran Wang, Amirhossein Khalilian-Gourtani, Leyao Yu, Patricia Dugan, Daniel Friedman, Werner Doyle, Orrin Devinsky, Yao Wang, and Adeen Flinker. A neural speech decoding framework leveraging deep learning and speech synthesis. *Nature Machine Intelligence*, pages 1–14, 2024.

[Ref249] Yiqian Yang, Yiqun Duan, Qiang Zhang, Renjing Xu, and Hui Xiong. Decode neural signal as speech. *arXiv preprint arXiv:2403.01748*, 2024.

[Ref250] Rajesh Kumar Aggarwal and Mayank Dave. Acoustic modeling problem for automatic speech recognition system: conventional methods (part i). *International Journal of Speech Technology*, 14:297–308, 2011.

[Ref251] Xuedong Huang, James Baker, and Raj Reddy. A historical perspective of speech recognition. *Communications of the ACM*, 57(1):94–103, 2014.

[Ref252] Rohit Prabhavalkar, Takaaki Hori, Tara N Sainath, Ralf Schlüter, and Shinji Watanabe. End-to-end speech recognition: A survey. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[Ref253] Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012.

[Ref254] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

[Ref255] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

[Ref256] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*, 2019.

[Ref257] Matteo Frigo, Charles E Leiserson, Harald Prokop, and Sridhar Ramachandran. Cache-oblivious algorithms. In *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)*, pages 285–297. IEEE, 1999.

[Ref258] Daniel Milstein, Jason Pacheco, Leigh Hochberg, John D Simeral, Beata Jarosiewicz, and Erik Sudderth. Multiscale semi-markov dynamics for intracortical brain-computer interfaces. *Advances in Neural Information Processing Systems*, 30, 2017.

[Ref259] David Poeppel, William J Idsardi, and Virginie Van Wassenhove. Speech perception at the interface of neurobiology and linguistics. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 363(1493):1071–1086, 2008.

[Ref260] Kai J Miller, Dora Hermes, and Nathan P Staff. The current state of electrocorticography-based brain–computer interfaces. *Neurosurgical focus*, 49(1):E2, 2020.

[Ref261] Mariska J Vansteensel, Elmar GM Pels, Martin G Bleichner, Mariana P Branco, Timothy Denison, Zachary V Freudenburg, Peter Gosselaar, Sacha Leinders, Thomas H Ottens, Max A Van Den Boom, et al. Fully implanted brain–computer interface in a locked-in patient with als. *New England Journal of Medicine*, 375(21):2060–2066, 2016.

[Ref262] Katharina Linse, Elisa Aust, Markus Joos, and Andreas Hermann. Communication matters—pitfalls and promise of hightech communication devices in palliative care of severely physically disabled patients with amyotrophic lateral sclerosis. *Frontiers in neurology*, 9:379945, 2018.

[Ref263] Sakhia Darjaa, Miloš Cerňak, Štefan Beňuš, Milan Rusko, Róbert Sabo, and Marián Trnka. Rule-based triphone mapping for acoustic modeling in automatic speech recognition. In *Text, Speech and Dialogue: 14th International Conference, TSD 2011, Pilsen, Czech Republic, September 1-5, 2011. Proceedings 14*, pages 268–275. Springer, 2011.

[Ref264] Fréjus AA LAleye, Laurent Besacier, Eugène C Ezin, and Cina Motamed. First automatic fongbe continuous speech recognition system: Development of acoustic models and language models. In *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 477–482. IEEE, 2016.

[Ref265] Partha Pratim Ray. Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, 2023.

[Ref266] 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, Aparna Srinivasan, Eun Young Choi, Matthew F. Glasser, Leigh R. Hochberg, Jaimie M. Henderson, Kiarash Shahlaie, Sergey D. Stavisky, and David M. Brandman. An Accurate and Rapidly Calibrating Speech Neuroprosthesis. *New England Journal of Medicine*, 391(7):609–618, 2024.

[Ref267] Yizi Zhang, Yanchen Wang, Donato Jiménez-Benetó, Zixuan Wang, Mehdi Azabou, Blake Richards, Renee Tung, Olivier Winter, Eva Dyer, Liam Paninski, et al. Towards

a" universal translator" for neural dynamics at single-cell, single-spike resolution. *Advances in Neural Information Processing Systems*, 37:80495–80521, 2024.

[Ref268] Nauman Ahad, Mark A Davenport, and Eva L Dyer. Time series domain adaptation via channel-selective representation alignment. *Transactions on Machine Learning Research*.

[Ref269] Jingyun Xiao, Ran Liu, and Eva L Dyer. Gaformer: Enhancing timeseries transformers through group-aware embeddings. In *The Twelfth International Conference on Learning Representations*, 2024.

[Ref270] Liam Paninski. Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Computation in Neural Systems*, 15(4):243, 2004.

[Ref271] Wei Wu, Yun Gao, Elie Bienenstock, John P Donoghue, and Michael J Black. Bayesian population decoding of motor cortical activity using a kalman filter. *Neural computation*, 18(1):80–118, 2006.

[Ref272] Tobias Pistohl, Tonio Ball, Andreas Schulze-Bonhage, Ad Aertsen, and Carsten Mehring. Prediction of arm movement trajectories from ecog-recordings in humans. *Journal of neuroscience methods*, 167(1):105–114, 2008.

[Ref273] Wei Wu, Jayant E Kulkarni, Nicholas G Hatsopoulos, and Liam Paninski. Neural decoding of hand motion using a linear state-space model with hidden states. *IEEE Transactions on neural systems and rehabilitation engineering*, 17(4):370–378, 2009.

[Ref274] Byron M Yu, Afsheen Afshar, Gopal Santhanam, Stephen Ryu, Krishna V Shenoy, and Maneesh Sahani. Extracting dynamical structure embedded in neural activity. *Advances in neural information processing systems*, 18, 2005.

[Ref275] Vernon Lawhern, Wei Wu, Nicholas Hatsopoulos, and Liam Paninski. Population decoding of motor cortical activity using a generalized linear model with hidden states. *Journal of neuroscience methods*, 189(2):267–280, 2010.

[Ref276] Anqi Wu, Nicholas A Roy, Stephen Keeley, and Jonathan W Pillow. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. *Advances in neural information processing systems*, 30, 2017.

[Ref277] Timothy D Kim, Thomas Z Luo, Jonathan W Pillow, and Carlos D Brody. Inferring latent dynamics underlying neural population activity via neural differential equations. In *International Conference on Machine Learning*, pages 5551–5561. PMLR, 2021.

[Ref278] Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, 617(7960):360–368, 2023.

[Ref279] 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.

[Ref280] Avery Hee-Woon Ryoo, Nanda H Krishna, Ximeng Mao, Mehdi Azabou, Eva L Dyer, Matthew G Perich, and Guillaume Lajoie. Generalizable, real-time neural decoding with hybrid state-space models. *arXiv preprint arXiv:2506.05320*, 2025.

[Ref281] Jingyuan Li, Leo Scholl, Trung Le, Pavithra Rajeswaran, Amy Orsborn, and Eli Shlizerman. Amag: Additive, multiplicative and adaptive graph neural network for forecasting neuron activity. *Advances in Neural Information Processing Systems*, 36:8988–9014, 2023.

[Ref282] Mehdi Azabou, Mohammad Gheshlaghi Azar, Ran Liu, Chi-Heng Lin, Erik C Johnson, Kiran Bhaskaran-Nair, WashU-St Louis, Max Dabagia, Bernardo Avila-

Pires, Lindsey Kitchell, et al. Mine your own view: A self-supervised approach for learning representations of neural activity.

[Ref283] Nicholas A Steinmetz, Cagatay Aydin, Anna Lebedeva, Michael Okun, Marius Pachitariu, Marius Bauza, Maxime Beau, Jai Bhagat, Claudia Böhm, Martijn Broux, et al. Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science*, 372(6539):eabf4588, 2021.