Inferring Spatiotemporal Structure through Self-Supervised Learning: From Predictive Coding to Neuro Foundation Models

Linxing Jiang

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

Doctor of Philosophy

University of Washington

2026

Reading Committee:

Rajesh P. N. Rao, Chair

Matthew Golub

Bing W. Brunton

Program Authorized to Offer Degree:
Computer Science & Engineering

University of Washington

Abstract

Inferring Spatiotemporal Structure through Self-Supervised Learning: From Predictive Coding to Neuro Foundation Models

Linxing Jiang

Chair of the Supervisory Committee:
Rajesh P. N. Rao
Paul G. Allen School of Computer Science & Engineering

A hallmark of intelligence is the ability to infer spatiotemporal structure from streams of external input. In this thesis, I investigate prediction-based self-supervised learning as a principle for learning such structure in both biological and artificial systems. I first study sparse coding variational autoencoders as a bridge between classical predictive coding models and modern self-supervised generative models, showing that weight normalization constraints substantially improve decoder filter quality while preserving sparse, V1-like representations. I then introduce dynamic predictive coding (DPC), a hierarchical model of spatiotemporal prediction and sequence learning in the neocortex. The model assumes that higher cortical levels modulate the temporal dynamics of lower levels, correcting their predictions of dynamics using prediction errors. When trained by minimizing spatiotemporal prediction errors, DPC accounts for a wide range of neuronal and cognitive phenomena, from V1-like space-time receptive fields and cortical-like temporal response hierarchies to postdictive motion perception, cue-triggered sequence recall, and higher-order temporal abstractions. Finally, inspired by the recent success of self-supervised pretraining in large language models, I investigate whether similar approaches can learn generalizable structure from large-scale neural recordings. Using neural data transformers trained on multi-session mouse electrophysiology datasets, I show that pretraining improves neural activity prediction but scales modestly, with brain-region mismatch and session-to-session variability strongly limiting

transfer. Together, these results suggest that prediction-based self-supervised learning can be a powerful principle for discovering spatiotemporal structure in data. At the same time, scaling this principle to large-scale neural recordings will require methods that account for the heterogeneity of neural dynamics across brain regions, recording sessions, and individuals.

DEDICATION

For my family, born and chosen.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

While writing the rest of this thesis, I imagined many possible openings for this acknowledgments section. I could begin, earnestly and directly, by thanking the people who made this work possible, though that felt almost too conventional. I could begin with the long arc of my years in Seattle, now finally approaching its close, though I worried that it might try the reader's patience. I could begin with a joke or a quote, but none of the candidates seemed quite right.

Now that I have cheated my way into beginning my acknowledgments section, I would like to sincerely thank my advisor, Professor Rajesh Rao. It rarely happens that the first professor one has in an introductory course eventually becomes one's PhD advisor, but that is exactly what happened to me after I took Raj and Adrienne's computational neuroscience Coursera course as a sophomore. Raj is the best advisor *anyone could ever ask for*. Professionally, Raj has treated me as a colleague and a peer rather than just a student, and has given me the freedom to pursue the research directions that interest me, even when they were not necessarily the ones he found most interesting. In everyday life, Raj is an extremely kind, caring, and fun person. He has offered to deliver me COVID test kits when I was sick, given me rides to faraway meetings, and shown me great compassion when I talked about personal issues in our meetings. One highlight I will always remember is when Raj sent me a Slack message asking whether I knew of Xuanzang, one of the most famous Chinese Buddhist monks, who traveled to India in the 7th century to study Buddhism. At the end of the exchange, Raj said, "Well, you both made the great choice of picking Indian advisors!" I am forever grateful for his keen insight, warm support, and wonderfully playful personality, which accompanied my PhD journey, and am so lucky to continue to have Raj involved in my next chapter after graduation.

I would also like to thank my committee members, Professors Bing Brunton, Matt

Golub, and Nick Steinmetz, for their invaluable feedback and support throughout my PhD. I owe tremendous gratitude to Professor Andrea Stocco and Dr. Darby Losey, my very first academic advisors and mentors, who guided me through my first research project as an undergraduate and encouraged me to apply to PhD programs. I am also grateful to Professors Kathryn Dickerson and Alison Adcock, who hosted me in their lab for my first post-undergraduate research job. My time at Duke University was a transformative and decisive experience that convinced me to pursue research in computational neuroscience.

I am grateful to all of my collaborators from CSE and the Computational Neuroscience Center for enlightening discussions from which I have learned so much: Professors Eric Shea-Brown and Adrienne Fairhall, Shirui Chen, Yusi Chen, Stefano Recanatesi, Iman Tanumihardja, Xufeng Dai, Saghar Mirbagheri, Matt Bryan, Dimi Gklezakos, Vishwas Sathish, Prashant Rangarajan, Ares Fisher, and Luciano de la Iglesia. I also owe special thanks to Professor Ruth Anderson, who gave me my first TA job and was so supportive when I was trying to sort out visa issues during my fifth-year master's program, even promising me a TA position for the full year. Thank you to my internship mentors at Meta, Zahra Shakeri and Maryam Daneshi, for giving me the opportunity to experience industry research and for inspiring my future research directions.

We all know that research can sometimes be a lonely and frustrating process, but with a group of loving, funny, and supportive friends, it becomes much less unbearable. Thank you to my lifelong friend from high school, Xiaorang Guo, who is pursuing his PhD in Germany and has flown to the US many times to visit and support me. He is the type of friend I would trust with my family. In a sense, earning this PhD has finally given me a not-too-awkward way to tell him how much I appreciate his friendship, without having to say it to his face. So I am thankful for that as well. Thank you to Evan Blajev, who has been there for me since my undergraduate days and helped carry me through the difficult early years of graduate school. Thank you to the friends I have made here in Seattle: Weijia Shi, Xiaochuang Han, Sirui Lu, Tuochao Chen, Yuxue Yang, Zhenda Lu, Samantha Sun, Zoe Steine-Hanson, Timmy Pham, Yuren Pang, Wanyin Lin, Jingwei Ma, Xia Su, Daogao Liu,

and many, many others. I am deeply grateful to Patrick Wang, whose quiet care, patience, and belief in me have made even the hardest parts of this journey feel less lonely. You are my chosen family.

Finally, I would like to thank my parents and family in Harbin for their unconditional love and support. I could not have done this without your sacrifices.

Chapter 1

INTRODUCTION

As humans, perceiving the world through our senses may seem to be one of the most passive activities we could engage in. After all, perceiving the color of the sky on a typical Seattle winter morning, or the sound of one's roommate playing video games while we are trying to sleep, is usually not optional. However, beneath the seemingly simple act of opening our eyes to see the world, the brain is believed to perform complicated computations often described as *perception as inference*.

A normative theory for understanding perception is that the brain uses an *internal model* of the external world to infer the hidden causes of its sensory inputs and maintain beliefs about those causes. In the early work of Gregory et al. (1980), perception was defined as hypothesis testing, emphasizing the process of inferring explanations for sensory inputs. This view of perception as inference, rather than as a purely bottom-up feature-extraction process, is well illustrated by binocular rivalry. When conflicting monocular images are presented separately to the two eyes (Fig 1.1A), subjects do not perceive a stable mixture or superposition of the two stimuli. Instead, perception alternates between the two competing interpretations every few seconds (Tong et al., 2006). Such rivalry is difficult to explain if perception is only a passive readout of sensory input. It is more naturally understood as the brain entertaining and revising competing hypotheses about the hidden causes of ambiguous sensory evidence.

Internal models also help disambiguate sensory inputs that admit multiple interpretations. The footprint illusion in Fig 1.1B provides a simple example: the same image can be perceived as either convex or concave depending on its orientation. This perceptual difference arises because the visual system appears to use a "light-from-above" prior assumption (Sun and Perona, 1998). Such assumptions are necessary because visual perception is an ill-posed problem: many configurations of the three-dimensional world can produce the

Figure 1.1: **Bayesian perception**. **(A)** Example stimuli (dichoptic orthogonal gratings) presented to the left and right eye simultaneously that induce binocular rivalry. Subjects perceive exclusively one of the two grating orientations, with perception switching between the two orientations every few seconds. Adapted from Tong et al. (2006). **(B)** An illustration of the effects of the light-from-above prior assumption that the brain appears to use in 3D visual perception of a 2D image. The image of the two footprints on the right is a 180 degree rotation of the image on the left, but the perception of the shape of the footprints is markedly different. Adapted from Ernst and Di Luca (2011).

same two-dimensional retinal image. The observer is typically unaware of these prior assumptions; they are incorporated by neural circuits subconsciously to compute beliefs over hidden causes, implementing perception as a form of unconscious inference.

What are the learning mechanisms underlying the brain's internal model? One critical aspect to consider is the spatiotemporal *regularity* of the world. If the world consisted only of random noise, there would be no statistically meaningful or useful structure to learn, and the brain would have no reason to build an internal model. Instead, the world is highly structured, and the learning rules of internal models must discover which regularities in external stimuli are behaviorally relevant. In the "light-from-above" example, the meaningful structure refers to the direction of illumination rather than to other factors (e.g., the material properties of the object). When we enter a new environment, such as a house we have never been to before, we know not to look for a cooking knife in the bathroom, since we have learned the spatiotemporal structure of a typical house environment and the relationship between its compartments.

In this thesis, I investigate prediction-based self-supervised learning as a principle by which internal models can discover spatiotemporal regularities in both biological and artificial systems. On the biological side, predictive coding (Rao and Ballard, 1999) is a prominent theory of how cortical areas build an internal model of the world through feedforward and feedback connections. Extending this framework, I propose dynamic predictive coding (DPC), a hierarchical model of spatiotemporal prediction and sequence learning in the neocortex. DPC assumes that higher cortical levels modulate the temporal dynamics of lower levels, correcting their predictions of dynamics using prediction errors. When trained by minimizing spatiotemporal prediction errors, DPC accounts for a wide range of neuronal and cognitive phenomena, from V1-like space-time receptive fields and cortical-like temporal response hierarchies to postdictive motion perception, cue-triggered sequence recall, and higher-order temporal abstractions.

Large language models (LLMs) have recently revolutionized the field of artificial intelligence, demonstrating that self-supervised pretraining on large-scale data can lead to powerful generalization and transfer. Interestingly, the way such systems are trained closely resembles the principle of predictive coding: the models are trained to predict the next

token in a sequence and learn by minimizing prediction errors. Through this simple objective, LLMs learn to discover rich spatiotemporal structure in language and generalize to artificially constructed structures in a zero-shot manner. The latter part of this thesis investigates whether this large-scale pretraining approach can be applied to neural recordings to learn generalizable structure in neural activity. Using neural data transformers trained on multi-session mouse electrophysiology datasets, I examine whether self-supervised pretraining improves neural activity prediction as the amount and diversity of data increase. The results show that pretraining can improve downstream prediction, but the improvement scales modestly and is strongly limited by heterogeneity across brain regions and recording sessions.

The remainder of the thesis is organized as follows:

- Chapter 2 reviews predictive coding theories of cortical function. It introduces the Rao-Ballard predictive coding model and related formulations, connects predictive coding to the free energy principle and active inference, and summarizes experimental evidence for prediction and prediction-error signals in the sensory cortex.

- Chapter 3 studies the sparse coding variational autoencoder (SVAE), a model that combines classical sparse coding with variational inference in deep neural networks. I show that weight normalization improves decoder filter optimization while preserving sparse, V1-like representations.

- Chapter 4 introduces dynamic predictive coding (DPC), a hierarchical model of spatiotemporal prediction and sequence learning. I show that DPC can learn space-time receptive fields, temporal response hierarchies, postdictive motion effects, cue-triggered sequence recall, and higher-order temporal abstractions through prediction error minimization.

- Chapter 5 investigates the scaling properties of neural data transformers trained on large-scale mouse electrophysiology datasets. I show that data heterogeneity, including

brain-region mismatch and session-to-session variability, limits the scaling benefits of self-supervised pretraining on neural data.

- Chapter 6 concludes this thesis and discusses open questions.

1.1 Prior Publications

Portions of this thesis are based on the following prior publications:

- Linxing Preston Jiang and Rajesh P. N. Rao. Predictive Coding Theories of Cortical Function. *Oxford Research Encyclopedia of Neuroscience*, November 2022. doi: 10.1093/acrefore/9780190264086.013.328. URL `https://oxfordre.com/neuroscience/view/10.1093/acrefore/9780190264086.001.0001/acrefore-9780190264086-e-328`.

- Linxing Preston Jiang and Luciano de la Iglesia. Improved Training of Sparse Coding Variational Autoencoder via Weight Normalization. *arXiv*, 2021. doi: 10.48550/arXiv.2101.09453. URL `https://arxiv.org/abs/2101.09453`.

- Linxing Preston Jiang and Rajesh P. N. Rao. Dynamic Predictive Coding Explains Both Prediction and Postdiction in Visual Motion Perception. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 45(45), 2023. URL `https://escholarship.org/uc/item/6r74j9k1`.

- Linxing Preston Jiang and Rajesh P. N. Rao. Dynamic predictive coding: A model of hierarchical sequence learning and prediction in the neocortex. *PLOS Computational Biology*, 20(2):e1011801, 2024. doi: 10.1371/journal.pcbi.1011801. URL `https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1011801`.

- Linxing Preston Jiang, Shirui Chen, Emmanuel Tanumihardja, Xiaochuang Han, Weijia Shi, Eric Shea-Brown, and Rajesh P. N. Rao. Data Heterogeneity Limits the Scaling Effect of Pretraining in Neural Data Transformers. In *COLM 2025 Workshop on LLM*

for Scientific Discovery, July 2025a. URL https://openreview.net/forum?id=WiECjcxNw4.

Chapter 2

PREDICTIVE CODING THEORIES OF CORTICAL FUNCTION

2.1 Introduction

How can neural circuits in the cortex learn internal models of the world, and how can such circuits combine prior beliefs with sensory evidence for Bayesian inference? Predictive coding offers a possible neural implementation. The predictive coding model of Rao and Ballard (1999) assumes that the areas comprising the cortical hierarchy (Hubel and Wiesel, 1959; Felleman and Van Essen, 1991) implement a hierarchical generative model of the sensory world. The neural activities at each level of the hierarchy represent the brain's internal belief of the hidden causes of the stimuli at a particular abstraction level (e.g., edges, object parts, objects). Furthermore, the model assumes that the top-down feedback connections from higher to lower order cortical areas convey predictions of lower-level activities. The bottom-up feedforward connections in turn convey prediction errors, calculated as the difference between the top-down predictions and actual activities. The neural activities at each level representing the beliefs about the hidden causes are jointly influenced by both the top-down predictions and the bottom-up error signals. Overall, the model assumes the goal of the cortex is to minimize prediction errors across all levels. Importantly, the above neural operations can be interpreted within a Bayesian framework: the top-down predictions convey prior beliefs based on learned expectations while the bottom-up prediction errors carry evidence from the current input. Predictive coding combines these two sources of information, weighted according to their reliability (inverse variances or "precisions"), to compute the posterior beliefs over hidden causes at each level. The objective of minimizing prediction errors across all levels can thus be shown to be equivalent to finding the *maximum a posteriori* (MAP) estimates of the hidden causes.

The phrase "predictive coding" was originally used to capture a form of efficient coding. The center-surround receptive fields and biphasic temporal antagonism in responses of

cells in the retina and lateral geniculate nucleus (LGN) can be interpreted as performing decorrelation through a simple form of predictive coding: rather than conveying the local intensity directly, retinal and LGN cells can be interpreted as sending the differences (errors) between the local intensity and a prediction of that intensity computed as a linear weighted sum of nearby values in space and preceding input values in time (Srinivasan et al., 1982; Dong and Atick, 1995a; Huang and Rao, 2011). In auditory information processing, Smith and Lewicki (2006) used the same efficient coding principle to derive a model which yields kernels (filter weights) that closely match auditory filters.

More broadly, predictive coding can be viewed as Bayesian inference in the context of Rao and Ballard's hierarchical predictive coding model (Rao and Ballard, 1997, 1999). This model was originally proposed to explain extra-classical receptive field effects and contextual modulation. More recent models inspired by predictive coding have demonstrated that a network trained to predict future inputs can explain a number of other cortical properties (Singer et al., 2018; Lotter et al., 2020). Beyond the cortex, the idea of computing errors between top-down predictions and lower-level inputs is consistent with theories of the cerebellum (Bell et al., 1997; Wolpert et al., 1998) and models of dopamine responses as reward prediction errors (Schultz et al., 1997). These examples suggest that the general principle of predictive coding could be a widely applicable and flexible algorithmic strategy implemented by the brain across different regions to support perception, motor control, and reward-based learning.

Empirical evidence for prediction and prediction error signals in the cortex has been growing at a fast pace. Neural responses corresponding to prediction errors induced by visual mismatches during self-generated locomotion have been discovered in layer 2/3 of the primary visual cortex (V1) in rodents (Keller et al., 2012; Fiser et al., 2016). Predictive signals have been found in V1 when an animal is adapted to visual-locomotion coupling in a virtual environment (Fiser et al., 2016). The cortex also learns to predict novel auditory stimuli coupled to an animal's locomotion and once learned, suppresses the responses to the learned stimuli in primary auditory cortex (Schneider et al., 2018), consistent with prediction error minimization. More recent studies (Jordan and Keller, 2020) have found some support for the distinct computational roles of the laminar structure of cortical columns proposed

by predictive coding theories. Recent research has also found that unexpected stimuli which induce large prediction error signals can drive synaptic learning in neural circuits (Gillon et al., 2021), as expected in a predictive coding circuit that uses prediction errors to learn a generative model of the world.

2.2 Predictive Coding Models: An Overview

The predictive coding model of Rao & Ballard begins with the assumption that sensory inputs are being generated by hidden states or "causes" in the external world via an unknown generative model. The goal of the brain then is to learn this generative model over many inputs. Perception, for any given sensory input $\mathbf{I}$, involves inverting this generative model, that is, estimating the hidden states or causes of input $\mathbf{I}$ given a learned generative model. Neural activities in the predictive coding model are assumed to represent estimates of the hidden state (also known as the latent variable) vector as estimated by the predictive coding neural network, given the observed sensory input vector $\mathbf{I}$. The prior distribution of hidden states is assumed to be $p(\mathbf{r})$, which imposes a constraint on neural activities such as sparse activation. The observation model $p(\mathbf{I}|\mathbf{r})$ is the likelihood that input $\mathbf{I}$ is generated given the cause or hidden state $\mathbf{r}$. The predictive coding model assumes that $p(\mathbf{I}|\mathbf{r})$ is parameterized by a matrix $\mathbf{U}$, which is assumed to be learned and encoded in the "top-down" synaptic weights of the network. *Inference and learning* correspond respectively to estimating $\mathbf{r}$ (equivalent to perception) and learning an estimate $\mathbf{U}$ (corresponding to synaptic learning), both with the goal of maximizing the joint probability $p(\mathbf{I}, \mathbf{r})$. Since $p(\mathbf{I})$ is constant, this is equivalent to maximizing the posterior probability $p(\mathbf{r}|\mathbf{I})$, also known as *maximum a posteriori* (MAP) inference.

In the predictive coding model, the likelihood $p(\mathbf{I}|\mathbf{r})$ is governed by the following equation, which relates the hidden state $\mathbf{r}$ to the input $\mathbf{I}$ via a function f and a matrix $\mathbf{U}$:

$$\mathbf{I} = f(\mathbf{U}\mathbf{r}) + \mathbf{n}. \tag{2.1}$$

Here, $\mathbf{n}$ is assumed to be zero mean Gaussian noise with covariance $\sigma^2\mathbb{I}$ ($\mathbb{I}$ is the identity matrix). This equation states that the input is assumed to be generated as a linear combination of the columns of matrix $\mathbf{U}$ weighted by the elements of $\mathbf{r}$, followed by a function f

and additive noise. The function f is a linear or nonlinear function (e.g., identity function, rectification function, or a sigmoidal function). The columns of $\mathbf{U}$ can be regarded as the "basis" vectors (e.g., edges or "parts" of an image or scene) that can be used to compose an input according to the values in the hidden "causes" vector $\mathbf{r}$. Given Eq 2.1 and the fact that $\mathbf{n}$ is zero mean Gaussian, the negative logarithm of the likelihood $p(\mathbf{I}|\mathbf{r})$ can be shown to be proportional to:

$$H_1 = \frac{1}{\sigma^2}\|\mathbf{I} - f(\mathbf{U}\mathbf{r})\|_2^2 = \frac{1}{\sigma^2}(\mathbf{I} - f(\mathbf{U}\mathbf{r}))^\top(\mathbf{I} - f(\mathbf{U}\mathbf{r})), \tag{2.2}$$

where $\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$ denotes the Euclidean or L_2 norm of vector $\mathbf{x}$. H_1 is the sum of squared errors between the image $\mathbf{I}$ and its reconstruction (or "prediction") $f(\mathbf{U}\mathbf{r})$ across all pixels, weighted by the inverse noise variance (or precision) $\frac{1}{\sigma^2}$. The predictive coding model also allows prior probability distributions $p(\mathbf{r})$ and $p(\mathbf{U})$ for the parameters $\mathbf{r}$ and $\mathbf{U}$, respectively. Taking these priors into account, we obtain the overall optimization function:

$$H = H_1 + H_2 \tag{2.3}$$

with

$$H_2 = g(\mathbf{r}) + h(\mathbf{U}), \tag{2.4}$$

where $g(\mathbf{r})$ and $h(\mathbf{U})$ are proportional to the negative logarithms of $p(\mathbf{r})$ and $p(\mathbf{U})$, respectively. If one assumes that both prior distributions are zero mean Gaussians with inverse variances α and λ, respectively, one obtains:

$$H_2 = g(\mathbf{r}) + h(\mathbf{U}) = \alpha \sum_j r_j^2 + \lambda \sum_{i,j} U_{ij}^2. \tag{2.5}$$

Minimizing the overall optimization function H is thus equivalent to MAP estimation. Predictive coding minimizes this objective function using both inference (of $\mathbf{r}$) and learning (of $\mathbf{U}$). Inference of $\mathbf{r}$ is implemented by a recurrent neural network that performs gradient descent on H with respect to $\mathbf{r}$ for each input. Remarkably, rather than being chosen a priori, the architecture of the predictive coding neural network is predicted from first principles by the gradient descent equations for optimizing H with respect to $\mathbf{r}$ (see "Network Dynamics and Synaptic Learning" section for details). The matrix $\mathbf{U}$ is represented by the synaptic

weights of the same network and learned through gradient descent on H with respect to $\mathbf{U}$ across many inputs.

2.2.1 Sparse Coding as a Special Case of Predictive Coding

The sparse coding model of Olshausen and Field (1996) for learning simple cell-like receptive fields can be regarded as a special case of the predictive coding model described above. In their model, the choice of the likelihood $p(\mathbf{I}|\mathbf{r})$ remains the same as above, but the prior $p(\mathbf{r})$ for the hidden state (causes) $\mathbf{r}$ is assumed to be a heavy-tailed distribution such as a Laplace distribution. Such a prior encourages sparsity in $\mathbf{r}$ (majority of the elements of $\mathbf{r}$ are zero or close to zero). Their model does not explicitly assume any specific prior for the synaptic weights $\mathbf{U}$. The inference and learning processes are almost identical to those for a single-level predictive coding model (see "Network Dynamics and Synaptic Learning" section). When applied to natural image patches, their model produces localized, orientation-selective receptive fields (columns of $\mathbf{U}$) similar to those of V1 simple cells, compared to using a Gaussian prior, which produces more global receptive fields. Such a sparseness prior promotes statistical independence in the output and encourages efficiency by selecting only a small subset of features to encode information (Olshausen and Field, 1996, 1997; Barlow, 1961). The underlying assumption here is that objects in the natural world are composed of a wide variety of features (or parts) but any given object is composed of only a small subset of them. This is consistent with the view that the brain evolved to adopt ecologically useful priors for learning its neural representations in its quest to learn an internal model of the world appropriate for the organism's ecological niche.

2.2.2 Hierarchical Predictive Coding

The above-described generative model can be extended to multiple hierarchical levels by assuming that the hidden state can be generated by a higher-level representation $\mathbf{r}^h$, corresponding to more abstract image properties than the lower-level representation:

$$\mathbf{r} = \mathbf{r}^{td} + \mathbf{n}^{td} = f(\mathbf{U}^h\mathbf{r}^h) + \mathbf{n}^{td}, \tag{2.6}$$

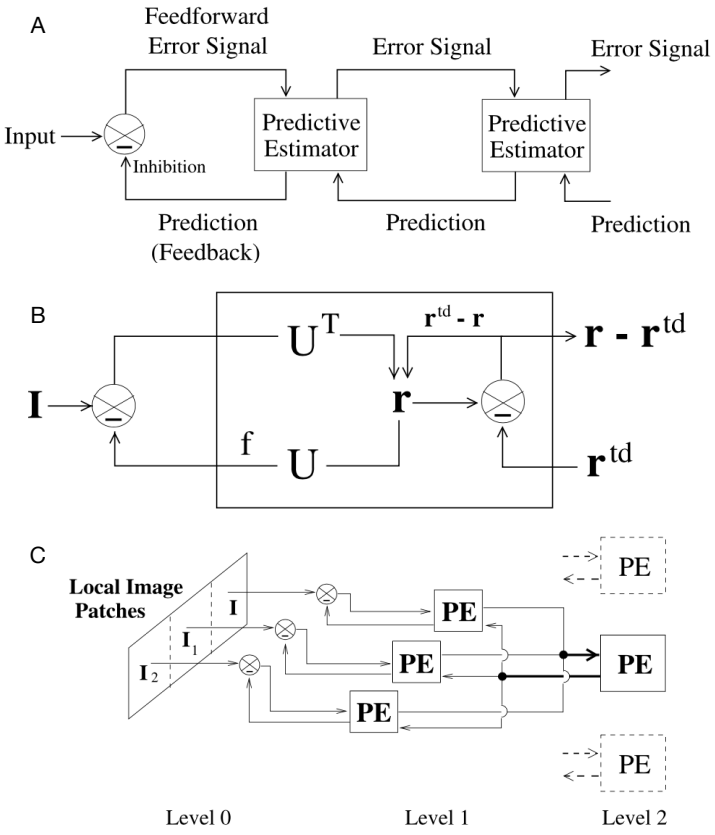


Figure 2.1: Caption continued at next page

where $\mathbf{r}^{td} = f(\mathbf{U}^h\mathbf{r}^h)$ is the top-down prediction of $\mathbf{r}$ and $\mathbf{n}^{td}$ is zero mean Gaussian noise with variance σ_{td}^2. The lower-level neurons have smaller receptive fields and represent a local image region by estimating the hidden state $\mathbf{r}$. The higher-level neurons estimate their state $\mathbf{r}^h$ based on several lower-level hidden states $\mathbf{r}$ associated with local image patches. This arrangement results in a progressive convergence of inputs from lower to higher levels and an increase in receptive field size as one ascends the hierarchical network, until the receptive fields of the highest-level neurons span the entire input image.

The overall optimization function for the hierarchical predictive coding model is:

$$H = \frac{1}{\sigma^2}(\mathbf{I} - f(\mathbf{U}\mathbf{r}))^\top(\mathbf{I} - f(\mathbf{U}\mathbf{r})) + \frac{1}{\sigma_{td}^2}(\mathbf{r} - \mathbf{r}^{td})^\top(\mathbf{r} - \mathbf{r}^{td}) + g(\mathbf{r}) + g(\mathbf{r}^h) + h(\mathbf{U}) + h(\mathbf{U}^h), \tag{2.7}$$

Figure 2.1: (Previous page.) **Hierarchical Predictive Coding Model. (A)** The general architecture of the hierarchical predictive coding model. Each predictive estimator (PE) module maintains an internal representation, generates top-down prediction (lower arrows, feedback), and receives bottom-up prediction errors (upper arrows, feedforward). **(B)** Components of a PE module. Feedforward connections carry bottom-up prediction errors from the lower level. Feedback connections deliver top-down predictions to the lower level. The internal representation neurons correct their current estimate $\mathbf{r}$ using both the bottom-up prediction error and the top-down prediction error. A separate class of error-detecting neurons compute the discrepancy between the current estimate and its top-down prediction, and send the error to the higher level. **(C)** An example two-level hierarchical network. Three image patches at Level 0 are processed separately by three Level 1 PE modules. These three Level 1 modules converge to provide input (prediction errors) to a single Level 2 module, which attempts to predict the states $\mathbf{r}$ in all three of these Level 1 PE modules. This convergence effectively increases the receptive field size of neurons as one ascends the hierarchy. (figure from Jiang and Rao, 2022)

where $g(\mathbf{r}^h)$ and $h(\mathbf{U}^h)$ are terms proportional to the negative logarithm of the priors for $\mathbf{r}^h$ and $\mathbf{U}^h$ respectively. Minimizing H is again equivalent to maximizing the posterior $p(\mathbf{r}, \mathbf{r}^h, \mathbf{U}, \mathbf{U}^h|\mathbf{I})$. Perceptual inference involves minimizing H with respect to $\mathbf{r}$ and $\mathbf{r}^h$ jointly, and learning involves minimizing H with respect to $\mathbf{U}$ and $\mathbf{U}^h$. Note that the first level state $\mathbf{r}$ is now conditioned on the second level state $\mathbf{r}^h$ and synaptic weights $\mathbf{U}^h$, but an additional prior constraint such as sparseness may be placed on $\mathbf{r}$ as well (the $g(\mathbf{r})$ term).

2.2.3 Network Dynamics and Synaptic Learning

Given the hierarchical generative model above, a MAP estimate of $\mathbf{r}$ can be obtained using gradient descent on H with respect to $\mathbf{r}$:

$$\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = -\frac{k_1}{2}\frac{\partial H}{\partial \mathbf{r}} = \frac{k_1}{\sigma^2}\mathbf{U}^\top \frac{\partial f^\top}{\partial \mathbf{x}}(\mathbf{I} - f(\mathbf{U}\mathbf{r})) + \frac{k_1}{\sigma_{td}^2}\left(\mathbf{r}^{td} - \mathbf{r}\right) - \frac{k_1}{2}g'(\mathbf{r}), \tag{2.8}$$

where k_1 is a positive constant governing the rate of descent toward a minimum for H, $\mathbf{x} = \mathbf{U}\mathbf{r}$, and g' is the derivative of g with respect to $\mathbf{r}$. A discrete time implementation of the above-mentioned dynamics leads to the following update equation for $\mathbf{r}$ at each time step (represented by neural activities or firing rates):

$$\hat{\mathbf{r}}_t = \hat{\mathbf{r}}_{t-1} + \frac{k_1}{\sigma^2}\mathbf{U}^\top \frac{\partial f^\top}{\partial \mathbf{x}_{t-1}} \left(\mathbf{I} - f\left(\mathbf{U}\hat{\mathbf{r}}_{t-1}\right)\right) + \frac{k_1}{\sigma_{td}^2}\left(\mathbf{r}_{t-1}^{td} - \hat{\mathbf{r}}_{t-1}\right) - \frac{k_1}{2} g'\left(\hat{\mathbf{r}}_{t-1}\right). \tag{2.9}$$

This equation, derived from first principles, specifies recurrent network dynamics for hierarchical predictive coding in terms of how the firing rate (or neural response) vector at a given level should be updated over time. At each time step, the neural activity vector $\mathbf{r}$ is multiplied by the feedback matrix $\mathbf{U}$ and a new prediction is generated for the lower level (Fig 2.1A and Fig 2.1B). This prediction is then subtracted from the lower-level representation $\mathbf{I}$ to generate the bottom-up error $(\mathbf{I} - f(\mathbf{U}\mathbf{r}))$, which is filtered by the feedforward weights $\mathbf{U}^\top$ and the gradient of the function f. Note that the bottom-up synaptic weights are the transpose of the top-down synaptic weights in this model, although this assumption can be relaxed using an approach similar to the one used in variational autoencoders (VAEs) (see "Predictive Coding and the Free Energy Principle" section). The neural response vector $\mathbf{r}$ is updated based on a weighted combination of the bottom-up prediction error $(\mathbf{I} - f(\mathbf{U}\mathbf{r}))$ and the top-down prediction error $\mathbf{r}^{td} - \mathbf{r}$ (Fig 2.1B). Each error is weighted by the inverse of the corresponding noise variance: The larger the noise variance, the smaller the weight given to that error term, consistent with the concept of Kalman filtering (see section "Prediction in Time: Spatiotemporal Predictive Coding and Kalman Filtering").

The learning rule for the feedback synaptic weights $\mathbf{U}$ (and feedforward weights $\mathbf{U}^\top$) is obtained by using gradient descent on H with respect to $\mathbf{U}$:

$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} = -\frac{k_2}{2}\frac{\partial H}{\partial \mathbf{U}} = \frac{k_2}{\sigma^2}\frac{\partial f^\top}{\partial \mathbf{x}}(\mathbf{I} - f(\mathbf{U}\mathbf{r}))\mathbf{r}^\top - k_2\lambda\mathbf{U}, \tag{2.10}$$

where k_2 is a positive parameter determining the learning rate of the network and $\mathbf{x} = \mathbf{U}\mathbf{r}$. Note that this learning rule is a form of Hebbian plasticity: for the feedforward weights $\mathbf{U}^\top$, the input presynaptic activity is the residual error $(\mathbf{I} - f(\mathbf{U}\mathbf{r}))$ (weighted by $\frac{\mathrm{d}f^\top}{\mathrm{d}\mathbf{x}}$) and the output postsynaptic activity is $\mathbf{r}$. More importantly, unlike backpropagation, the learning rule above is local since the feedforward connection explicitly conveys the prediction error

at each level. To ensure stability, learning of synaptic weights operates on a slower time scale than the dynamics of $\mathbf{r}$: The learning rate k_2 is a much smaller value than the rate k_1 governing the dynamics of the network. For static inputs, this implies that the network responses $\mathbf{r}$ converge to an estimate for the current input before the synaptic weights $\mathbf{U}$ are updated based on this converged estimate. An example two-level hierarchical network is depicted in Fig 2.1C.

2.2.4 Feedforward Perception as the Initial Inference Step in Predictive Coding

How does the traditional feedforward "bucket brigade" model of perception, where inputs are processed sequentially in one area and passed on to the next (e.g., LGN →V1 →V2 ...), align with the hierarchical predictive coding view of cortical processing? The answer to this question is easy to obtain from Eq 2.9 by considering what happens in the very first time step $t = 1$ when $\hat{\mathbf{r}}_0 = \mathbf{0}$ and the two top-down prediction terms $f(\mathbf{U}\hat{\mathbf{r}}_0)$ and $\mathbf{r}^{td}$ are also both $\mathbf{0}$. In this case, if $g'(\mathbf{0})$ is also $\mathbf{0}$, Eq 2.9 reduces to:

$$\hat{\mathbf{r}}_t = \frac{k_1}{\sigma^2}\mathbf{U}^\top \frac{\partial f^\top}{\partial \mathbf{x}_{t-1}}\mathbf{I}. \tag{2.11}$$

Thus, the first feedforward pass through the network multiplies the input $\mathbf{I}$ with the feedforward weights $\mathbf{U}^\top$ (besides the other multiplicative factors). Assuming this happens at all patches of an image, this equation describes exactly the type of operation implemented by a standard feedforward layer where the filters are given by the rows of $\mathbf{U}^\top$. In other words, for a static input, if the top-down predictions are assumed to be zero, a hierarchical predictive coding network (e.g., Fig 2.1C) initializes its estimates at all levels in the same manner as a deep neural network via a feedforward pass through all layers, before proceeding to further minimize prediction errors by generating top-down predictions from these initial estimates and refining them based on prediction errors.

2.2.5 Predictive Coding and the Free Energy Principle

Predictive coding and the principle of prediction error minimization are closely related to variational inference and learning, which form the basis for VAEs in machine learning research (Dayan et al., 1995; Kingma and Welling, 2014) as well as the free energy principle

in neuroscience as proposed by Friston and colleagues (Friston, 2005, 2010; Friston and Kiebel, 2009). This relationship is briefly summarized below.

MAP inference, as employed in the predictive coding model above, finds an estimate that maximizes the posterior distribution $p(\mathbf{r}|\mathbf{I})$. Variational inference aims to find the full posterior distribution instead of a point estimate. Applying Bayes' rule:

$$p(\mathbf{r}|\mathbf{I}) = \frac{p(\mathbf{I}|\mathbf{r})p(\mathbf{r})}{p(\mathbf{I})} = \frac{p(\mathbf{I}|\mathbf{r})p(\mathbf{r})}{\int \mathrm{d}\mathbf{r} p(\mathbf{I}|\mathbf{r})p(\mathbf{r})}. \tag{2.12}$$

The normalizing factor (denominator) contains multidimensional integrals that are usually intractable to compute (e.g., if $p(\mathbf{r})$ is a sparsity-inducing Laplace distribution in sparse coding). Due to this intractability, variational inference approximates the posterior as follows: the true posterior probability distribution p_θ parameterized by parameters θ is approximated with a more tractable distribution q_φ parameterized by parameters φ. The "error" between the two distributions is quantified using the Kullback-Leibler (KL) divergence between the posterior probabilities of the latent variable $\mathbf{r}$ given the input data $\mathbf{I}$:

$$\begin{aligned} KL\left(q_\varphi(\mathbf{r}|\mathbf{I}) \| p_\theta(\mathbf{r}|\mathbf{I})\right) &= \int q_\varphi(\mathbf{r}|\mathbf{I}) \log \frac{q_\varphi(\mathbf{r}|\mathbf{I})}{p_\theta(\mathbf{r}|\mathbf{I})} \mathrm{d}\mathbf{r} \\ &= \int q_\varphi(\mathbf{r}|\mathbf{I}) \log \frac{q_\varphi(\mathbf{r}|\mathbf{I})}{p_\theta(\mathbf{r},\mathbf{I})} \mathrm{d}\mathbf{r} + \int q_\varphi(\mathbf{r}|\mathbf{I}) \log p_\theta(\mathbf{I}) \mathrm{d}\mathbf{r} \\ &= \int q_\varphi(\mathbf{r}|\mathbf{I}) \log \frac{q_\varphi(\mathbf{r}|\mathbf{I})}{p_\theta(\mathbf{r},\mathbf{I})} \mathrm{d}\mathbf{r} + \log p_\theta(\mathbf{I}) \\ &= \mathcal{F} + \log p_\theta(\mathbf{I}), \end{aligned} \tag{2.13}$$

where $\mathcal{F}$ is called the "variational free energy" and $\log p_\theta(\mathbf{I})$ is called the data log likelihood (given model parameters θ) or model evidence. Note that variational free energy $\mathcal{F}$ should not be confused with the physical notion of free energy (e.g., in thermodynamics), although there is a similarity in their definitions.

Rewriting Eq 2.13, we have:

$$\begin{aligned} \log p_\theta(\mathbf{I}) &= KL\left(q_\varphi(\mathbf{r}|\mathbf{I}) \| p_\theta(\mathbf{r}|\mathbf{I})\right) - \mathcal{F} \\ &= KL\left(q_\varphi(\mathbf{r}|\mathbf{I}) \| p_\theta(\mathbf{r}|\mathbf{I})\right) + \mathcal{L}, \end{aligned} \tag{2.14}$$

where $\mathcal{L} = -\mathcal{F}$ is called the evidence lower bound (or ELBO) in the variational learning and VAE literature since $\log p_\theta(\mathbf{I}) \geq \mathcal{L}$ (the KL divergence is nonnegative). It can be seen

that an organism or artificial agent can increase model evidence (data log likelihood) by maximizing the ELBO $\mathcal{L}$ or equivalently, minimizing variational free energy $\mathcal{F}$ with respect to the latent state and parameters. Note that since $\mathcal{F} = KL\left(q_\varphi(\mathbf{r}|\mathbf{I})\|p_\theta(\mathbf{r}|\mathbf{I})\right) - \log p_\theta(\mathbf{I})$ and $\log p_\theta(\mathbf{I})$ does not depend on $\mathbf{r}$ or φ, maximizing the ELBO (minimizing $\mathcal{F}$) with respect to $\mathbf{r}$ and φ is equivalent to minimizing the KL divergence between the approximating tractable distribution q and the true distribution p.

To make the connection to predictive coding, the definition of variational free energy $\mathcal{F}$ used in Eq 2.13 can be rewritten as follows:

$$\begin{aligned}
\mathcal{F} &= \int q_\varphi(\mathbf{r}|\mathbf{I}) \log \frac{q_\varphi(\mathbf{r}|\mathbf{I})}{p_\theta(\mathbf{I}|\mathbf{r})p_\theta(\mathbf{r})} \mathrm{d}\mathbf{r} \\
&= \int q_\varphi(\mathbf{r}|\mathbf{I}) \log \frac{q_\varphi(\mathbf{r}|\mathbf{I})}{p_\theta(\mathbf{r})} \mathrm{d}\mathbf{r} + \int q_\varphi(\mathbf{r}|\mathbf{I}) \log \frac{1}{p_\theta(\mathbf{I}|\mathbf{r})} \mathrm{d}\mathbf{r} \\
&= KL(q_\varphi(\mathbf{r}|\mathbf{I})||p_\theta(\mathbf{r})) - \mathbb{E}_q\left[\log p_\theta(\mathbf{I}|\mathbf{r})\right]. \qquad (2.15)
\end{aligned}$$

Using the relationship in Eq 2.2 for the negative logarithm of $p_\theta(\mathbf{I}|\mathbf{r})$ and using α as the constant of proportionality for Eq 2.2, the free energy for the predictive coding model is given by:

$$\begin{aligned}
\mathcal{F} &= KL(q_\varphi(\mathbf{r}|\mathbf{I})||p_\theta(\mathbf{r})) - \mathbb{E}_q\left[-\alpha H_1\right] \\
&= KL(q_\varphi(\mathbf{r}|\mathbf{I})||p_\theta(\mathbf{r})) + \alpha \mathbb{E}_q\left[\frac{1}{\sigma^2}\|\mathbf{I} - f(\mathbf{U}\mathbf{r})\|_2^2\right] \\
&= \text{KL divergence between posterior and prior for } \mathbf{r} \\
&\quad + \alpha \times \text{mean squared prediction error.} \qquad (2.16)
\end{aligned}$$

Thus, within the predictive coding framework, minimizing the variational free energy $\mathcal{F}$, as advocated by the free energy principle of brain function (Bogacz, 2017; Friston, 2010), is equivalent to finding an approximating posterior distribution q_φ that both minimizes prediction errors while also attempting to be close to the prior for $\mathbf{r}$. This can be regarded as a full-distribution version of the predictive coding model described above, which uses MAP inference to find an optimal point estimate that minimizes prediction errors while also being constrained by the negative logarithm of the prior (Eq 2.3).

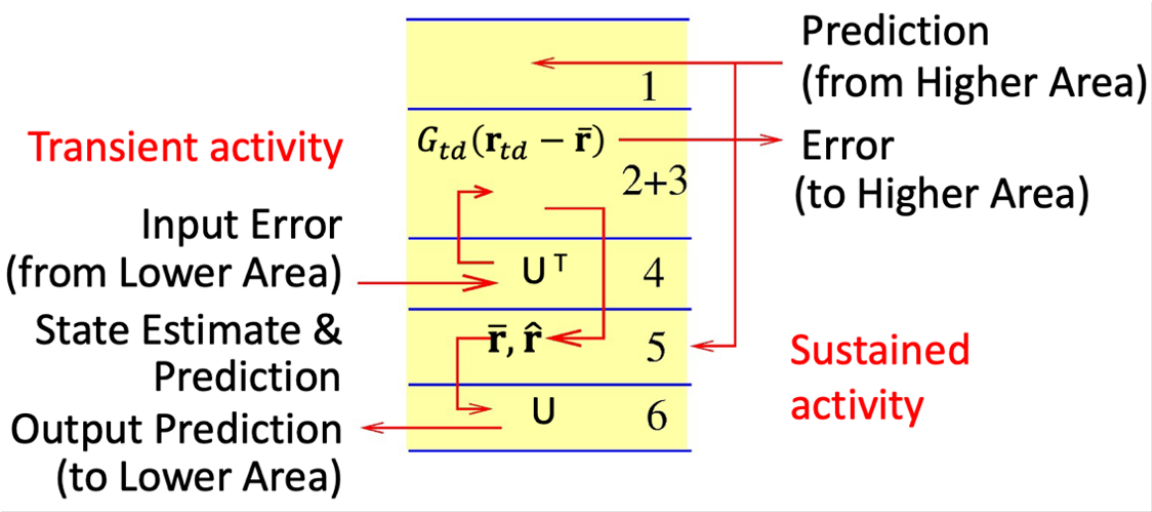


Figure 2.2: **Putative laminar implementation of the Rao-Ballard predictive coding model.** A mapping of the hierarchical predictive coding model components (see Fig 2.1) to cortical layers. (figure from Jiang and Rao, 2022)

2.2.6 Neuroanatomical Implementation of Predictive Coding

Rao and Ballard (1999) postulated two groups of neurons at each hierarchical level with distinct computational goals (Fig 2.2). One group of neurons maintains an internal representation (state estimate) for generating top-down predictions of lower-level activities. These neurons are hypothesized to be in the deep layers 5/6 of cortical columns and are predicted by the model to exhibit sustained activity to maintain predictions to lower levels. A different group of neurons at the same level calculates prediction errors to be conveyed to the next higher level. These were suggested to be layer 2/3 neurons which send connections to "higher" order cortical areas and which are expected to exhibit transient activity. Since prediction errors can be positive or negative, Rao and Ballard (1999) proposed two subclasses of error-detecting neurons, one subclass representing positive errors and another representing negative errors, similar to on-center off-surround and off-center on-surround neurons in the retina and LGN.

In general, as seen above in endstopping and other contextual effects, the model predicts

that layer 2/3 neurons are suppressed when the stimuli are predictable (i.e., consistent with natural image statistics) while deeper layer neurons remain active. Stimuli that deviate from natural image statistics ("novel" stimuli) on the other hand elicit large responses in layer 2/3 neurons. The model also predicts that prediction error signals are used for unsupervised learning of the synaptic connections in the predictive coding network, driving the synaptic weights to better reflect the structure of the input stimuli.

2.2.7 A Common Misconception About the Predictive Coding Model

One of the most common misconceptions about the predictive coding model is that the model predicts suppression of all neural activity when stimuli become predictable. This has led some authors to state that experimental evidence showing neurons not being suppressed or maintaining persistent firing for predictable inputs contradicts the predictive coding model. On the contrary, the predictive coding model requires a group of neurons to maintain the internal representation (state estimate $\hat{\mathbf{r}}$) at each hierarchical level for generating predictions for the lower level (see "Hierarchical Predictive Coding" and Fig 2.2). Thus, in the predictive coding model, the neurons that are suppressed when stimuli become predictable are error-detecting neurons that are distinct from the neurons maintaining the network's internal representation of the external world. Similar to the efficient coding models of the retina and LGN (Srinivasan et al., 1982; Dong and Atick, 1995a), redundancy reduction occurs primarily in the feedforward pathways of the Rao-Ballard predictive coding model, with the feedback pathways remaining active to convey predictions.

2.3 Empirical Evidence for Predictive Coding

Experimental evidence has been mounting for predictive processing in the cortex thanks to advances in neuronal recording and stimulation techniques such as optical imaging and optogenetics. Particularly relevant to the hierarchical predictive coding model proposed by Rao and Ballard (1999) are findings of top-down predictive "internal representation" neurons and bottom-up error-detecting neurons in a cortical column. These findings appear to suggest that the cortex may indeed be implementing a hierarchical generative model of the natural world. We briefly review the experimental evidence below.

2.3.1 Internal Representation Neurons and Prediction Error Neurons in the Cortex

The hierarchical predictive coding model predicts the existence of at least two functionally distinct classes of neurons in the cortex: internal state representation neurons $\mathbf{r}$, which maintain the current estimate of state at a given hierarchical level and are postulated to reside in the deeper layers 5/6 of the cortex, and error-detecting neurons $\mathbf{r} - \mathbf{r}^{td}$ in layers 2/3, which compute the difference between the current state estimate and its top-down prediction from a higher level. Recent studies have provided evidence for both types of neurons in the cortex.

Keller et al. (2012) recorded neural activities from layer 2/3 cells in the monocular visual cortex of behaving mice that were head-fixed and running on a spherical treadmill. The mice were exposed to 10–30 minutes of visual feedback as they ran on the treadmill. In normal "feedback" trials, the visual flow stimuli provided to the mouse were full-field vertical gratings coupled to the mouse's locomotion on the treadmill. In "mismatch" trials, visual-locomotion mismatches were delivered randomly as brief visual flow halts (1 second). As a control, the mice also went through "playback trials" in which visual flow was passively viewed without locomotion.

The authors found that 13.0% of the visual cortical neurons recorded responded predominantly to feedback mismatches. Fig 2.3A shows a sample neuron (cell number 677) that responded mainly to mismatch trials (orange shading). Also, 23.6% of the neurons responded mainly to feedback trials in which visual flow feedback was predictable (cell number 452 in Fig 2.3A). The mismatch responses were also significant in the population average (Fig 2.3B) and the activity onset in mismatch trials was much stronger than that in the other trials. Furthermore, the mismatch signals encoded the degree of mismatch – a visual flow halt during faster locomotion resulted in a stronger response than during slow locomotion (Fig 2.3C, darker lines denote faster speed at the time of visual flow halt).

V1 neurons have also been found to be predictive of spatial locations after adapting to a new environment. In an experiment by Fiser et al. (2016), mice went through a virtual tunnel with blocks of two different grating patterns (A or B) separated by distinct landmarks. The five trial conditions only differed in the fifth block, where the grating

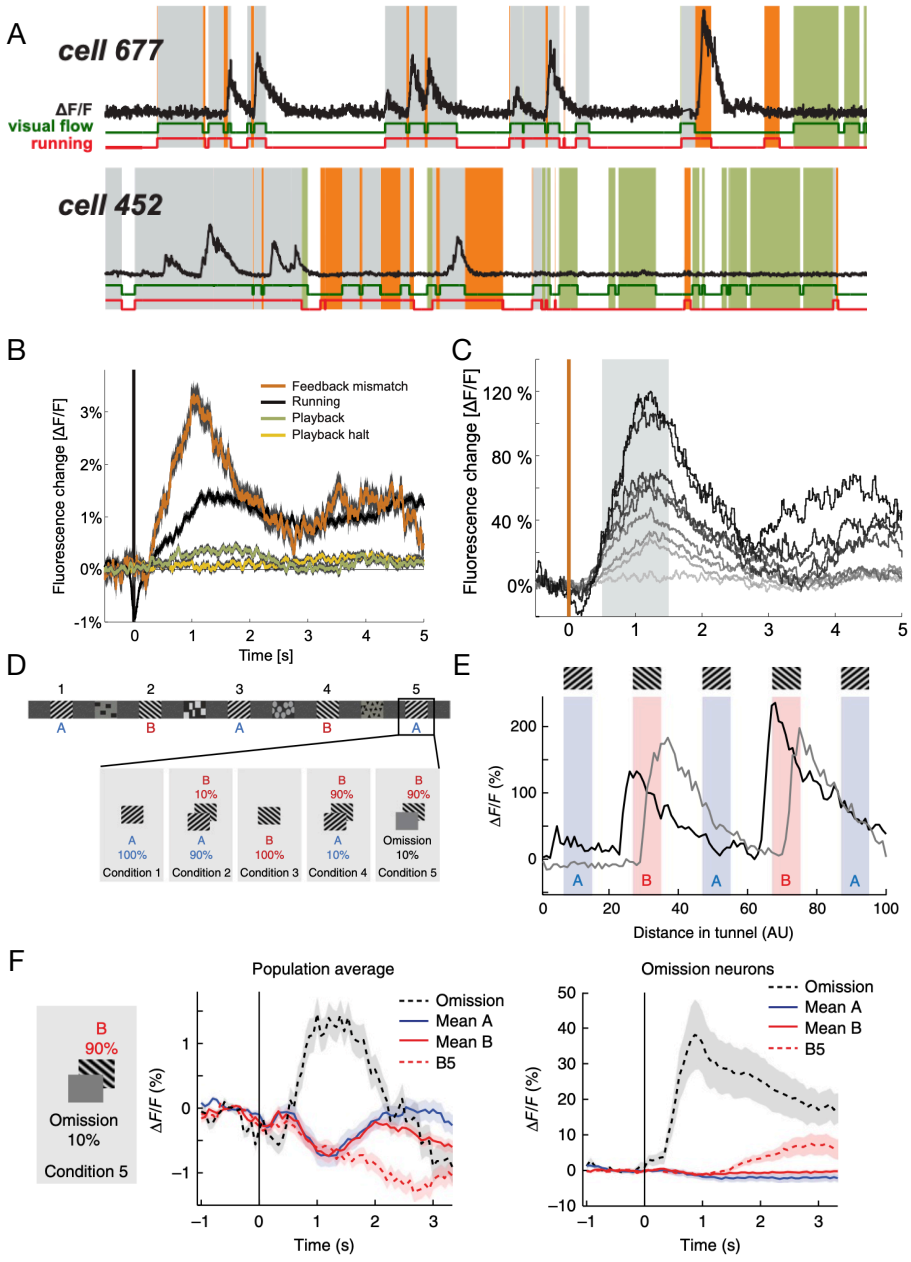


Figure 2.3: Caption continued at next page

patterns A and B as well as omission with no visual stimuli had different probabilities of occurring (see Fig 2.3D). After adaptation, some neurons developed predictive responses to specific visual stimuli based on spatial information. As shown in Fig 2.3E, an example

Figure 2.3: (Previous page.) **Evidence for predictive neurons and error-detecting neurons in two visual-locomotion tasks. (A)** Sample fluorescence change ($\Delta F/F$) traces from two cells (black). Green trace: binary indicator of visual flow. Red trace: binary indicator of locomotion. Gray shading: visual feedback with locomotion. Orange shading: feedback mismatch (no visual flow with motion). Green shading: playback (visual flow with no motion). White: baseline. Cell 677 responded predominantly to feedback mismatch while cell 452 responded mainly to visual feedback with locomotion. **(B)** Average population response to onsets (time=0) of different trials: feedback mismatch (orange), running (black), playback (green), and passive viewing of playback halts (yellow). **(C)** Mismatch responses during various speeds of locomotion. Darker traces represent faster locomotion speed. **(D)** Schematic representation of the texture lining both walls of the virtual tunnel that mice ran through. The last block indicates the percentage of appearance of each texture (including omission trials) under different conditions. **(E)** Two example neurons' response traces when the mouse ran through the tunnel. Blue shading represents the period when the mouse was in Block A, red shading represents Block B. Both neurons are spatially selective only to Block B, not Block A. The neuron depicted by the black trace showed predictive responses (activation prior to entering Block B), while the neuron depicted by the gray trace was only responsive after the mouse entered Block B. **(F)** left: average population response to omission trials (black dashed) compared to Block A trials (blue), Block B trials (red), and the 90% Block B5 trials (red dashed). Panel F, right: average responses of the omission-selective neural population. Adapted from Keller et al. (2012) (Panels A–C) and Fiser et al. (2016) (Panels D–F). (figure from Jiang and Rao, 2022)

neuron (black trace) showed strong activation before the mouse perceived Block B (but not Block A). In contrast, another sample neuron (gray trace) showed activation after entering Block B (but not Block A). The authors also discovered prediction error responses similar to those reported by Keller et al. (2012). The population average of neural activities during omission trials was much greater than during A and B trials (Fig 2.3F, left). Moreover, a

subset of neurons (2.3%) developed omission-selectivity – they showed large responses only to the omission trials (Fig 2.3F, right).

Other studies have also documented neural responses carrying predictive information. Xu et al. (2012) found that after rats adapted to a visual moving dot trajectory, a brief flash at the starting point of the same trajectory triggered the same sequential firing pattern in the rat's V1 as evoked by the full-sequence stimulus. Similarly, Gavornik and Bear (2014) discovered that after an animal is exposed to a sequence of stimuli during training, V1 regenerates the sequential response even when certain elements of the sequence are omitted.

Prediction and prediction error-like signals have also been found in cortical areas in the human visual cortex (e.g., Murray et al. (2002)) and the hierarchical face processing region of the monkey inferior temporal cortex (IT) (Tsao et al., 2006; Freiwald and Tsao, 2010). Schwiedrzik and Freiwald (2017) exposed macaque monkeys to fixed pairs of face images with different head orientations and identities such that the successor face image can be predicted from the preceding face image. Neurons in the lower-level face area ML (middle lateral section of the superior temporal sulcus) displayed large responses when the pair association was violated (either in identity, or head orientation, or both). Furthermore, prediction errors resulting from view violation (head orientation) diminished and eventually vanished during the late phase of responses while those resulting from identity violation remained significant. This is consistent with the interpretation that the top-down predictive signals from the view-invariant neurons in higher-level anterior lateral and anterior medial areas suppress the view mismatch responses (encoded locally in the lower-level ML area), while identity-related mismatch signals are propagated through feedforward circuits for further processing. In another study, Issa et al. (2018) used different face-part configuration stimuli (typical versus atypical) and found that the lower-level areas of the hierarchy (posterior IT and central IT) signal deviations of their preferred features from the expected configurations, whereas the top level (anterior IT) maintained a preference for natural, frontal face-part configuration. The authors further discovered that the early responses in central IT and anterior IT are correlated with late responses in posterior IT: Images that produced large responses in higher-level areas early are followed by reduced activities in lower-level areas, consistent with top-down prediction signals subduing lower-level responses.

In another experiment, Choi et al. (2018) showed that a hierarchical inference model could explain the effect of feedback signals from the prefrontal cortex to intermediate visual cortex V4 as top-down predictions of partially occluded shapes.

Schneider et al. (2018) explored the effects of learning on prediction error-like activity in the primary auditory cortex. Rats were given artificial auditory feedback coupled to their locomotion: The pitch of the sound was proportional to the rat's running speed. They found that a group of neurons in the rat's primary auditory cortex initially responded strongly to the artificial auditory feedback ("reafferent sound") but over the course of several days, the neuronal circuits learned to suppress this activity. The suppression occurred whenever the reafferent sound was coupled to the rat's locomotion and did not occur when a nonreafferent sound was played or when the reafferent sound was played during resting. The gradual suppression of responses is consistent with how the predictive coding model learns an internal model of the environment: as the network learns to predict the artificial sound coupled to the rat's locomotion, the predictions get better, resulting in decreasing prediction errors which manifest as suppression of the auditory neurons' activities.

The results discussed above provide evidence for predictive neural activity and prediction error-like responses in the cortex. The Rao and Ballard model additionally postulates that layer 2/3 neurons compute and convey the prediction errors while neurons in the deeper layers 5/6 maintain the state estimate. Recent experiments have attempted to test these predictions. While it is hard to distinguish the state estimating "internal representation" neurons from those driven by bottom-up sensory stimuli (see review by Keller and Mrsic-Flogel (2018) for further discussions), there is a growing body of evidence suggesting that layer 2/3 neurons may indeed play a role in comparing bottom-up information and top-down predictions.

2.3.2 Layer 2/3 Neurons as Top-Down Bottom-Up Signal Comparators

For biological networks to use prediction errors to correct their estimate, both positive and negative errors need to be represented. At any input location, a positive prediction error ($(\mathbf{I} - \mathbf{Ur} > 0)$) occurs when the input is not predicted (or incorrectly predicted) while a

negative prediction error ($(\mathbf{I} - \mathbf{Ur} < 0)$) occurs when a predicted input is omitted. Rao and Ballard (1999) postulated that layer 2/3 in the cortex may employ two different groups of neurons, one to convey positive errors and another for negative errors, similar to on-center, off-surround and off-center, on-surround ganglion cells in the retina (Srinivasan et al., 1982).

To test this theory, Jordan and Keller (2020) used an experimental setup similar to the one used in Keller et al. (2012): mice ran on a treadmill with locomotion-coupled visual flow feedback. Whole-cell recordings were obtained from both layer 2/3 and layer 5/6 neurons in V1. Visual feedback could be interrupted with a brief flow halt (1 second) at random times to generate visual-locomotion mismatch events. Out of 32 neurons recorded in layer 2/3, 17 neurons showed depolarizing activities (Fig 2.4A, left, depolarizing mismatch (dMM) neurons) and 6 neurons showed hyperpolarizing activities (Fig 2.4A, right, hyperpolarizing mismatch (hMM) neurons) during mismatch trials. These results suggest that the dMM and hMM neurons in layer 2/3 may subserve the function of encoding positive and negative prediction errors.

In addition, 30% of the neurons exhibited significant correlations between the mismatch responses and the speed of locomotion (visual halts that occurred during faster locomotion generated "stronger" mismatch signals). The sign of the correlation was also different between dMM and hMM neurons, with dMM neurons showing a positive correlation (Fig 2.4B, left) and hMM neurons showing a negative correlation (Fig 2.4B, right). These results are consistent with Keller and colleagues' calcium imaging study previously discussed Keller et al. (2012) (Fig 2.3C), showing that the responses of layer 2/3 neurons could potentially signal the quantitative level of prediction errors.

Jordan and Keller (2020) also investigated the differences between the responses of layer 2/3 neurons and deeper layer 5/6 neurons during normal visual feedback trials and mismatch trials. A much lower ratio of neurons in layers 5/6 (5 out of 14) responded predominantly to mismatch trials. Additionally, larger activity during mismatch trials was rare (1 neuron), with 7 neurons exhibiting reduced activities. The difference in responses between superficial and deep layer neurons was significant in mismatch trials (Fig 2.4C, right) but not in normal visual flow trials (Fig 2.4C, left). To further characterize the influence of visual flow and locomotion on layer 2/3 neurons versus layer 5/6 neurons, correlations between the activities

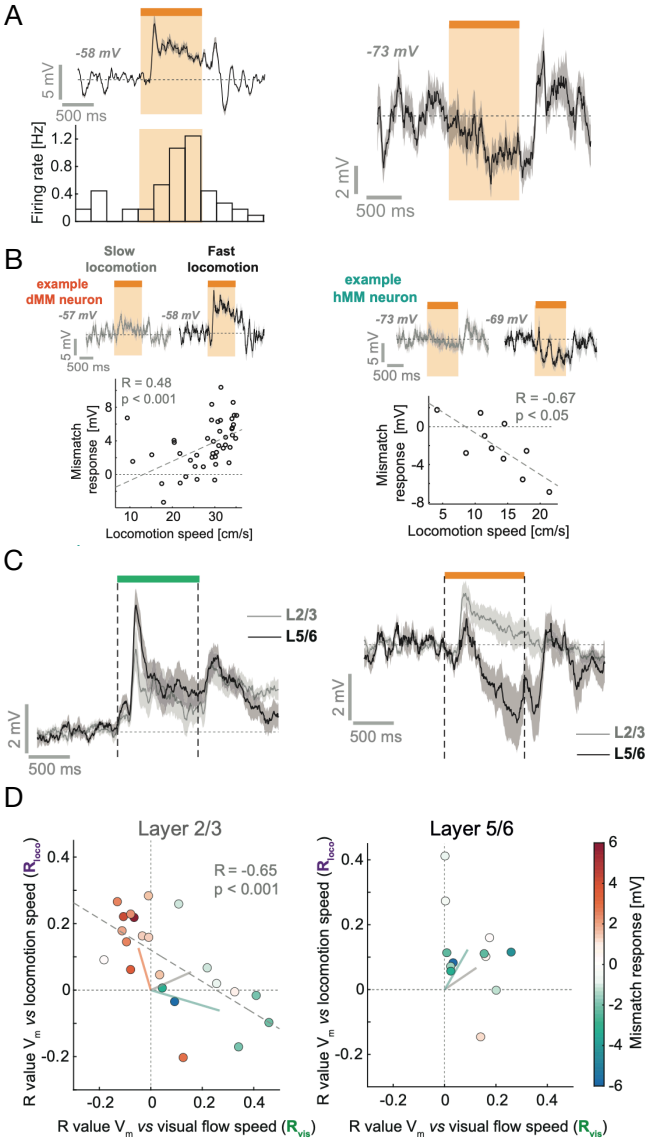


Figure 2.4: Caption continued at next page

of these neurons and locomotion speed or visual flow speed were calculated. As seen in Fig 2.4D (left plot), the distribution of correlations in layer 2/3 was bimodal: Activities of most dMM neurons were positively correlated with locomotion speed and negatively correlated with visual flow speed (and vice versa for hMM neurons). On the other hand, activities of layer 5/6 neurons were mostly positively correlated with both locomotion speed

Figure 2.4: (Previous page.) **Comparison of layer 2/3 versus layer 5/6 neurons in the mouse visual cortex. (A)** left: Average membrane potential (Vm) response (top) and firing rate histogram (bottom) during mismatch trials of a layer 2/3 neuron (dMM), showing depolarizing Vm. Orange shading represents the mismatch trial period. The mean prestimulus voltage is –58mV. Panel A, right: Average Vm of another layer 2/3 neuron (hMM) showing hyperpolarizing Vm during mismatch trials. **(B)**, left: An example dMM neuron showing a positive correlation between response and locomotion speed. Panel B, right: An example hMM neuron showing a negative correlation between response and locomotion speed. **(C)** Comparison of average responses between layer 2/3 neurons (light gray) and layer 5/6 neurons (dark gray) during visual feedback trials (left) and during mismatch trials (right). **(D)** Scatter plot of correlation coefficients between the membrane potential of the neurons and visual flow speed (x-axis) or locomotion speed (y-axis). The dot color denotes the membrane potential change in response in mismatch trials. The gray dashed line is the fitted linear regression line to the data. Solid lines represent the average responses of dMM neurons (red), hMM neurons (turquoise), and unclassified neurons (gray). Panel D shows layer 2/3 neurons (left) and layer 5/6 neurons (right). Adapted from Jordan and Keller (2020). (figure from Jiang and Rao, 2022)

and visual flow speed (Fig 2.4D, right). These results suggest that layer 2/3 neurons are well-suited to computing the error between the locomotion-generated predictions of visual inputs and the actual visual input, whereas the deeper layer 5/6 neurons may integrate top-down predictions (here, from motor areas) and bottom-up input to compute an estimate of the state at the current hierarchical level.

The difference in neural responses to expected versus unexpected visual flows in layer 2/3 versus layer 5 was also confirmed in a recent study by Gillon et al. (2021). The authors used an open-loop experiment (no sensorimotor coupling to locomotion) with stimuli consisting of moving squares. Expectation violations were created in some trials by making 25% of the visual squares move in the opposite direction compared to the other 75%. The

authors found that somatic and distal apical dendritic populations in layer 5 did not exhibit significantly different responses to expected versus unexpected visual flow, whereas both layer 2/3 somatic and distal apical dendritic populations showed a significant difference in responses. Additionally, this difference increased over days of exposure. Gillon et al. also found learning effects when mice were exposed to Gabor sequence stimuli for several consecutive days. The responses to unexpected stimuli (in this case, novel Gabor stimuli replacing an expected stimulus in a sequence) were predictive of how these responses evolve in subsequent sessions on a cell-by-cell basis. Besides implicating layer 2/3 neurons in prediction error computation, these results further confirm that the neural responses to unexpected stimuli (i.e., prediction errors) can drive learning in neural circuits, an important computational prediction of the predictive coding model (Rao and Ballard, 1999) (see Eq 2.10).

The larger distribution of error detecting neurons in superficial layers than deep layers was also confirmed by Hamm et al. (2021) in awake mice with visual oddball paradigms. The authors additionally showed that optogenetic suppression of prefrontal inputs to V1 reduced the contextual selectivity of the error detecting neurons, consistent with the effect of top-down signals in the predictive coding model. Finally, through laminar local field potential recordings in monkeys, Bastos et al. (2020) showed that predictability of visual stimuli affects neural activities in the superficial and deep layers differently—during predictable trials, there was an enhancement of alpha and beta power in the deep layers of the cortex whereas during unpredictable trials, an increase in spiking and gamma power was observed in the superficial layers.

2.4 Discussion

By casting Bayesian inference and learning in terms of minimizing prediction errors based on an internal model of the world, predictive coding provides a unifying view of perception and learning. Perception is equated with Bayesian inference of hidden states of the world and proceeds by forming predictive hypotheses about inputs that are corrected based on prediction errors. Learning corresponds to using the inferred states to build an internal model of the world that minimizes prediction errors through synaptic plasticity. Actions

can further minimize prediction errors with respect to future goals via active inference.

The hierarchical predictive coding model (Rao and Ballard, 1999) assumes that the hierarchical structure of the cortex forms predictive hypotheses at multiple levels of abstractions to explain input data. The model postulates that feedback connections between cortical areas convey predictions of expected neural activity from higher to lower levels, while the feedforward connections convey the prediction errors back to the higher level to correct the neural activity at that level.

Several recent experimental studies have discovered neurons in the visual and auditory cortex that encode predictions or prediction errors in a variety of sensory-motor tasks (Keller et al., 2012; Fiser et al., 2016; Schneider et al., 2018). Some studies have tested more detailed neuroanatomical predictions such as the role of cortical layer 2/3 neurons in error computation (Jordan and Keller, 2020). Others have shown that these error-related neural activities can drive learning in synaptic connections (Gillon et al., 2021). Although further tests are required, the experimental results reviewed above support the hypothesis that the cortex implements a predictive model of the world, uses this model to generate predictions, and utilizes prediction errors to both correct its moment-to-moment estimates and to learn a better model of the world.

There remain many aspects of predictive coding that require further exploration and experimental corroboration. For example, are layer 5/6 neurons computing and maintaining the hidden state as specified by Eq 2.8? Are the inverse variances in Eq 2.8 ("precisions" terms in the free energy principle; see Friston (2010)) computed in the cortex? If so, how are they used to weigh the bottom-up and top-down terms in the predictive coding network dynamics (Eq 2.8)?

In the next two chapters, we present two parallel directions from the predictive coding framework reviewed here. Chapter 3 introduces techniques that improve sparse coding variational autoencoder training so that their latent spaces preserve sparse, V1-like representations as generative filters. Chapter 4 offers a new hierarchical spatiotemporal formulation of predictive coding, providing normative explanations for a variety of neurophysiological and cognitive phenomena in terms of the brain learning a hierarchical generative model of the world.

Chapter 3

IMPROVED TRAINING OF SPARSE CODING VARIATIONAL AUTOENCODER VIA WEIGHT NORMALIZATION

In this chapter, we study the sparse coding variational autoencoder (SVAE), a model that combines sparse coding with variational inference and learning. We show that the original SVAE training procedure leaves many decoder filters under-optimized, and then introduce a weight normalization scheme that improves decoder filter quality while preserving sparse, V1-like representations.

3.1 Introduction

One key challenge in theoretical neuroscience is to understand the computation carried through the visual pathways. Hubel and Wiesel first showed that neurons in mammalian primary visual cortex (V1) have spatially localized and orientation-selective receptive fields (Hubel and Wiesel, 1959) that interestingly resemble edge detectors or "parts" of objects. This connection between visual neurons' neurophysiological properties and the statistics of the environment was successfully explored by Olshausen & Field (1996). In their model, they proposed that the primary visual cortex is learning a generative model of the visual world with sparsely activated neural activities. Remarkably, such a model produces edge detector style Gabor-like spatial filters that are similar to V1 receptive fields. This is known as the "sparse coding" model of V1.

Sparse coding (or sparse dictionary learning) has been extensively studied in the machine learning community as an unsupervised generative model of images (Lee et al., 2006; Gregor and LeCun, 2010; Chen et al., 2018). It has also been shown to be robust to adversarial attacks (Paiton et al., 2020; Sulam et al., 2020). More recently, Barello et al. (2024) proposed a sparse coding variational autoencoder (SVAE) model that combines a variational autoencoder (VAE) (Kingma and Welling, 2014) with a sparse coding decoder for learning sparse structure in data. Compared to the original sparse coding model, SVAE shows bet-

ter reconstruction performances and allows stochastic latent representations, which is more neurally plausible than a deterministic maximum a posteriori (MAP) estimate in traditional sparse coding.

The work presented in this chapter focuses on improving the quality of the learned decoder in SVAE. We show empirically that the formulation and training of SVAE lead to a large portion of unoptimized filters in the decoder, commonly known as the "over-pruning" problem in VAEs (Bowman et al., 2016; Kingma et al., 2016; Yeung et al., 2017). We propose three heuristics to improve the training of SVAE: First, we weighed the Kullback–Leibler (KL) divergence term in the loss function by β, a technique proposed by the β-VAE model (Higgins et al., 2017). Using a β term less than 1 smooths the effect of the sparsity prior, causing large gradients on only a few filters. Second, we used a more expressive encoder architecture with ResNet blocks (He et al., 2016) to replace the linear filters in the original SVAE for better posterior approximation. Most importantly, we applied the same projected gradient descent step in the original sparse coding to constrain the decoder filters to have unit length. This constraint drastically improves the number of filters that are optimized and have similar quality receptive fields to the sparse coding model. We validate our claims by comparing the performance of the original SVAE and our training procedure on natural image patches. SVAE trained with our approach shows similar reconstruction error to the original training and produces qualitatively better filters that resemble parts of images. We further show that the unit-length constraint of the filters (which can be viewed as lateral inhibition in the cortex) is critical for the formation of Gabor-like filters on both natural images and the MNIST dataset.

We first introduce the formulation of sparse coding and sparse coding variational autoencoders (SVAEs) in this section. The sparse coding model minimizes the following energy function

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{z}} E = \|\mathbf{x} - \mathbf{U}\mathbf{z}\|_2^2 + \lambda\|\mathbf{z}\|_1 \\ &\quad \text{s.t.} \|\mathbf{U}_i\|_2 \leq 1 \qquad\qquad \forall i = 1, 2, \ldots, N \end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^D$ denotes the input, $\mathbf{U} \in \mathbb{R}^{D \times N}$ represents the receptive fields (RFs) or filters of the model, and $\mathbf{z} \in \mathbb{R}^N$ represents the neural activation (latent variables). The L_1 penalty

on $\mathbf{z}$ is a relaxation of the L_0 penalty that promotes sparsity in $\mathbf{z}$ (only a small subset of the components is nonzero). λ is a scalar that controls the degree of the sparsity penalty. In addition, each filter (column of $\mathbf{U}$) is constrained to have unit L_2 norm to prevent a few filters with large weights from dominating image reconstruction. We will show in later sections and results that this is a key constraint to promote filter quality and increase the number of active latent variables.

Inference A common practice is to use proximal gradient descent rather than the vanilla gradient descent for faster convergence of the latent code. We use the iterative shrinkage threshold algorithm (ISTA) (Boyd and Vandenberghe, 2004) which takes a shrinkage step after a gradient update. The gradient update is defined as:

$$\frac{\partial E}{\partial \mathbf{z}} = -2\mathbf{U}^\intercal(\mathbf{x} - \mathbf{U}\mathbf{z})$$

and the shrinkage update is defined as

$$\begin{aligned}\mathbf{z}' &= \text{Shrinkage}_\lambda(\mathbf{z}) \\ &= \text{sign}(\mathbf{z}) \max(|\mathbf{z}| - \lambda, 0)\end{aligned}$$

We consider $\mathbf{z}$ as converged if the change of its L_2 norm before and after one update is less than 1%.

Learning After $\mathbf{z}$ converges for the current input $\mathbf{x}$, we update $\mathbf{U}$ using projected gradient descent. The update rule is defined as

$$\frac{\partial E}{\partial \mathbf{U}} = -2(\mathbf{x} - \mathbf{U}\mathbf{z})\mathbf{z}^\intercal$$

where η is the learning rate. After a gradient update, we project each column of $\mathbf{U}$ to unit norm following the constraint.

The inference step of sparse coding can be seen as performing a MAP estimate through gradient updates. Variational autoencoders (VAEs), on the other hand, perform inference using a feedforward mapping from the observation to the latent posterior using shared parameters across observations. Such mapping is then learned through minimizing the

Kullback–Leibler (KL) divergence between the approximating distribution and the true posterior

$$D_{KL}(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}|\mathbf{x}))$$

Since we cannot tractably compute the true posterior, VAEs choose to optimize the evidence lower bound (ELBO) $\mathcal{L}$ defined as

$$D_{KL}(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}|\mathbf{x})) = \log p_\theta(\mathbf{x}) - \mathcal{L}$$

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{\mathbf{z}\sim q(\cdot|\mathbf{x})}\left[\log p_\theta(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z}|\mathbf{x})\right] \\ &= \mathbb{E}_{\mathbf{z}\sim q(\cdot|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - D_{KL}\left(q(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})\right)\end{aligned}$$

In the original VAE formulation (Kingma and Welling, 2014), the proposal distribution $q(\mathbf{z}|\mathbf{x})$, the latent prior $p_\theta(\mathbf{z})$, and the likelihood term $p_\theta(\mathbf{x}|\mathbf{z})$ are all chosen to be Gaussian. The dimension of the latent code $\mathbf{z}$ is typically much smaller than the data dimension, forcing the model to learn low-dimensional structures that generate the true data distribution.

SVAE In the work of Barello et al. (2024), the authors proposed three modifications to the original VAE: (1) Make the dimension of the latent variables overcomplete (larger than the input dimension); (2) Use a sparsity-inducing prior (e.g. $p_\theta(\mathbf{z}) \sim \text{Laplace}(0,1)$) instead of the Gaussian prior; (3) Parameterize the decoder with a single linear layer rather than a deep neural network. In SVAE, the encoder replaces the iterative inference (ISTA), generating a full posterior $q(\mathbf{z}|\mathbf{x})$ instead of a single MAP estimation. The decoder behaves like the sparse coding filters $\mathbf{U}$ by taking a sampled $\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})$ and reconstructing the input $\hat{\mathbf{x}} = \mathbf{U}\mathbf{z}$. The proposal distribution and the likelihood term remain Gaussian, and the encoder is parameterized with two linear layers followed by a ReLU nonlinearity, and two separate linear layers that generate the mean and the log variance of the proposal distribution, respectively.

However, in practice, we found that this SVAE formulation leads to a large number of noise filters learned in the decoder. Figure 3.2 shows the decoder filters learned by sparse coding (left) and SVAE (middle). Only a small subset of filters in SVAE decoder resemble

the oriented bandpass Gabor filters like the RFs in V1, while most of the filters are not optimized to represent sparse structure in natural image data. In the next section, we present a few heuristics to improve the number of active filters learned in SVAE (Fig 3.2, right).

3.2 Methods

To tackle the issue of under-optimized filters in the decoder of SVAE, we propose the following heuristics to improve the training of SVAE.

Balancing between reconstruction and KL divergence Following the idea proposed in β-VAE (Higgins et al., 2017), we weigh the KL divergence term in ELBO to be less than 1 in order to smooth the effect of the sparse prior placing heavy gradients on only a few filters. The new ELBO term then becomes

$$\mathcal{L} = \mathbb{E}_{\mathbf{z}\sim q(\cdot|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \beta D_{KL}\left(q(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})\right)$$

More expressive encoder To improve the quality of the approximating posterior, we replace the linear layers in the original SVAE with ResNet blocks (He et al., 2016).

SVAE decoder with unit norm constraint Most importantly, we noticed that in the end-to-end training of SVAE, the filters of the decoder no longer have the constraint of having unit L_2 norm. We suspect that the over-pruning issue of VAE training (Yeung et al., 2017) and the sparsity-inducing prior together exacerbate the imbalance of training, which causes only a few filters to have dominating gradients, leaving most of the filters not fully optimized. Therefore, we propose to use the same projected gradient descent step in the original sparse coding model on SVAE to ensure each decoder filter is constrained to unit L_2 norm. We show that this drastically improves the number of optimized filters (see Results).

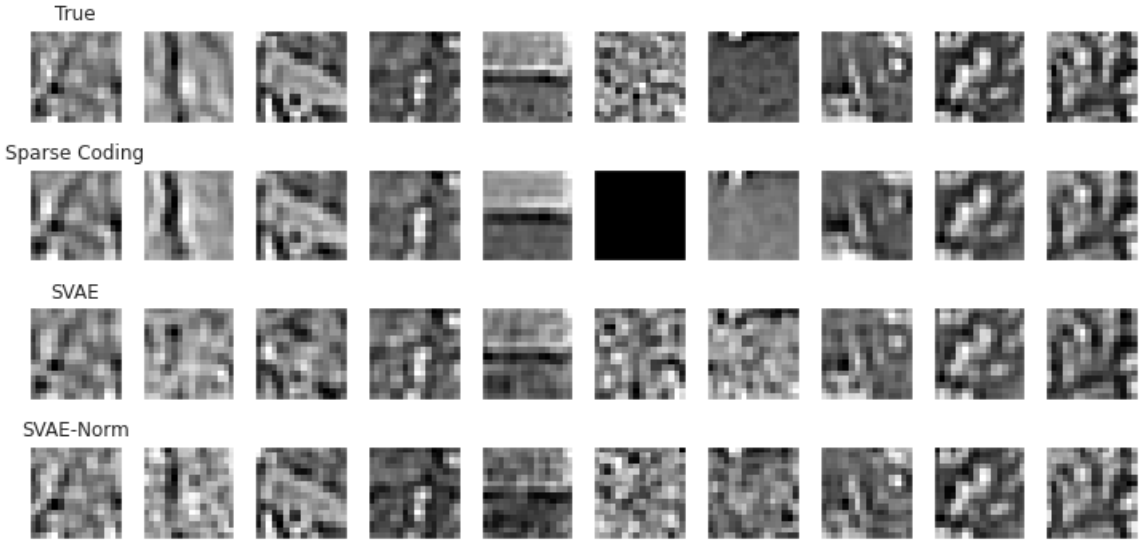


Figure 3.1: Reconstructions of 16x16 natural image patches in the held-out test set using the different models.

3.3 Results

Here, we present the performance comparison among the traditional sparse coding model, SVAE, and SVAE with our training heuristics. To further investigate the effect of decoder weight normalization, we also trained two SVAEs both with the first two proposed heuristics (β weighting, ResNet block encoders), but one with weight normalization ("SVAE-Norm") and the other one without. We experimented with natural image patches and MNIST handwritten digits. The natural image patches[1] are spatially whitened with a low pass filter $R(f) = fe^{-(f/f_0)^4}$, $f_0 = 200$ cycles/image. We used $z_i \sim \text{Laplace}(0, 0.1)\ \forall i = 1 \dots N$ as the factorized sparsity-inducing prior for all experiments.

3.3.1 Reconstruction

We first examined the performance of the models on reconstruction. Figure 3.1 shows 10 reconstructed patches, while Table 3.1 shows the pixelwise mean squared error (MSE) of

[1] `http://www.rctn.org/bruno/sparsenet/`

each model's reconstructions on the entire test set. Our normalized SVAE model (SVAE-Norm) achieves the lowest MSE, followed by SVAE. The traditional sparse coding model performs worse than both.

Model	MSE	STD (Monte Carlo Samples)
Sparse Coding	0.0150	N/A
SVAE	0.00875	1.631e-5
SVAE-Norm	0.00769	2.329e-5

Table 3.1: The mean squared error and standard deviation for reconstructions of natural images in the held-out test set over 50 trials.

3.3.2 Neural representation and activity

Next, we examined the receptive fields of the neurons in the model's decoder, shown in Fig 3.2. In the SVAE formulation, these filters should behave similarly to the traditional sparse coding filters. SVAE-Norm exhibits the clearest Gabors, followed by the sparse coding model. SVAE has some Gabor-like structures, but most of its neurons have either PCA-like receptive fields (Olshausen and Field, 1996) (e.g. Fig 3.2 middle, first row, middle column) or are unoptimized (gray filters).

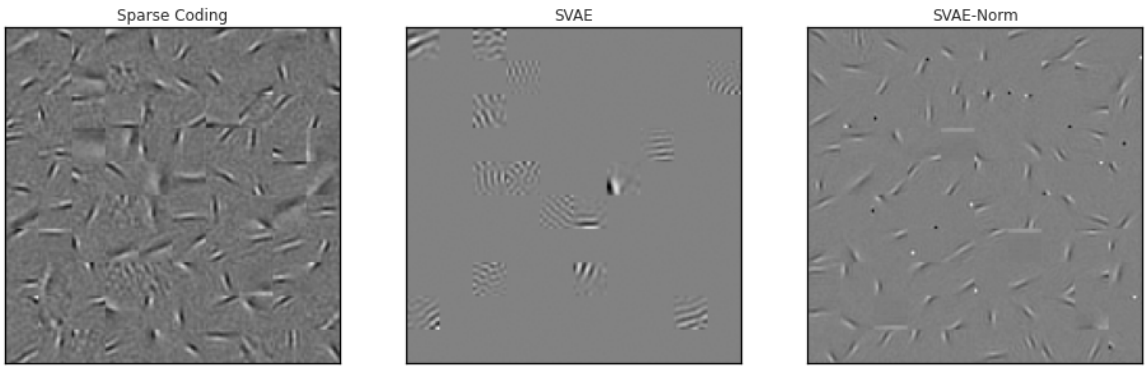


Figure 3.2: Receptive fields for 100 neurons in the representation layer of each of the models.

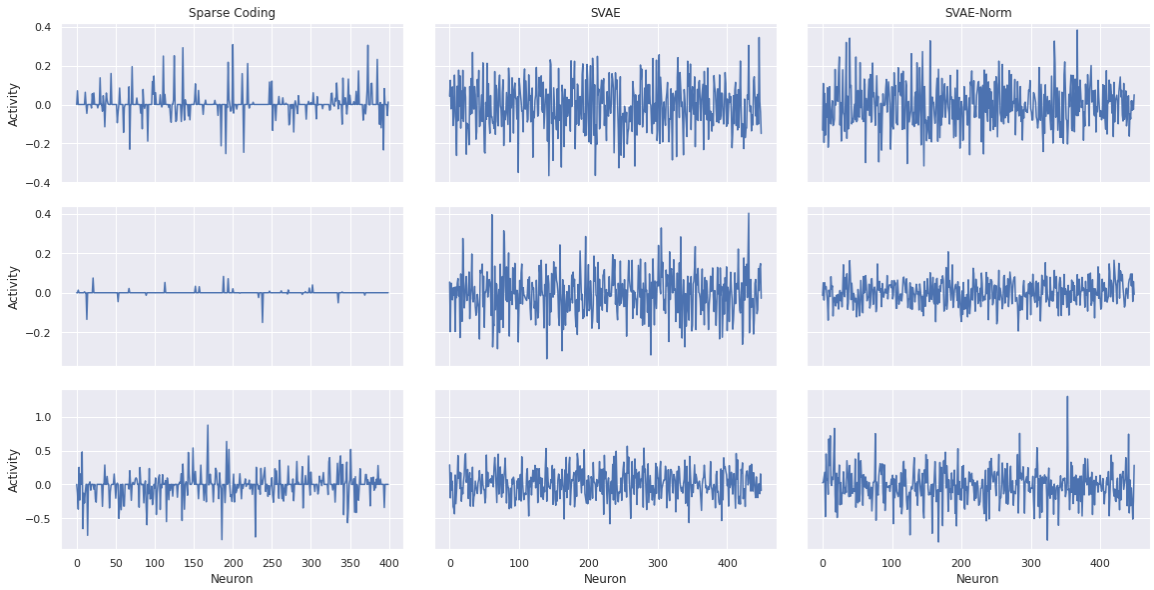


Figure 3.3: Activity of all the neurons in the representation layer of our networks for 3 different natural image patch inputs shown in each row.

When presented with a natural image patch, only a few neurons have a large activation (Fig 3.3, each row shows a different input patch). This is consistent with biological and computational findings in the sparse-coding domain. However, the sparse coding model has a clearer sparse activation across the 3 images (left column), while the VAEs visually appear to be noisier (middle and right column). This is expected due to the different inference procedures (ISTA vs. amortized inference + sampling), although it is interesting to see that strictly sparse code (true zero activations) is not required to produce "parts of images" structures in the filters (in SVAEs).

To evaluate the conditional distribution $p_\theta(\mathbf{x}|\mathbf{z})$ learned by the decoder, we generated images by sampling from the prior Laplace distribution (Fig 3.4). Although these images are noisy, they look similar to the natural image patches of Fig 3.1, suggesting the SVAEs learned a meaningful hidden representation that generates input data.

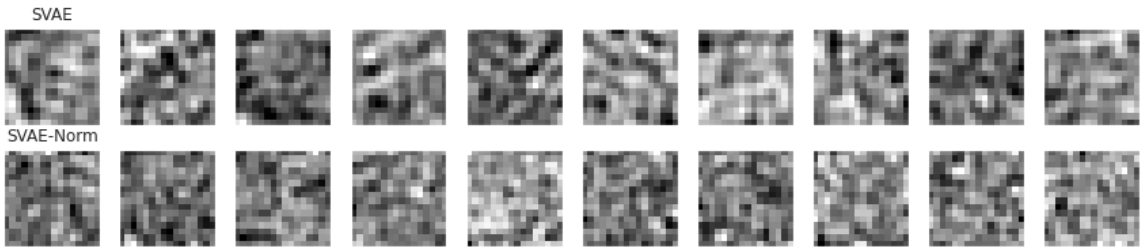
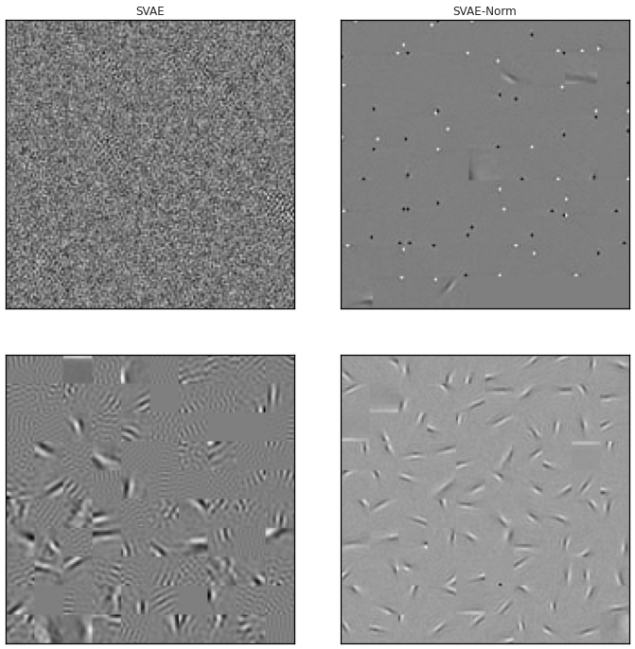


Figure 3.4: Images generated by random sampling from the prior distribution of our VAE models.

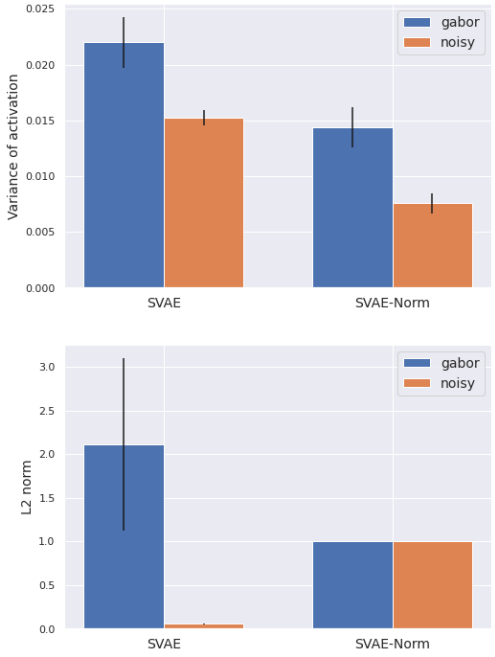
3.3.3 Noise filters

We observed that many neurons are under-optimized and learned white-noise-like filters rather than Gabor-like filters in the SVAE model (Fig 3.5(a), top left). This suggests that a majority of the filters were not learning meaningful latent structure of the data. We were able to isolate these neurons in the SVAE model by thresholding based on the standard deviation of the latent code activation on the test set. We used 0.5 as the threshold for both SVAE and SVAE-Norm models, and found that 296 filters out of 450 in SVAE are below the threshold. We visualized the variance distribution of these filters in Fig 3.5(b) top panel. In Fig 3.5(b), we show that these "noise" filters have extremely small L_2 norms, compared to the filters that show clear Gabor-like structure (Fig 3.5(a), bottom left). However, with weight normalization applied to SVAE during training, only 76 filters out of 450 are below the threshold, and these filters are mainly highly sparse filters that encode single pixels of the images (Fig 3.5(a), top right). The majority of the filters in the SVAE-Norm model now show Gabor-like structure similar to the traditional sparse coding models and V1 RFs (Fig 3.5(a), bottom right).

To validate the effect of weight normalization in learning quality filters, we ran the same two SVAE models on the MNIST dataset, with or without weight normalization. Figure 3.6 shows 100 randomly sampled decoder filters from the two models. The original SVAE model only shows 7 filters with large norms but noisy structures, while all filters in the SVAE-Norm model show stroke-like structures that resemble parts of MNIST digits. For



(a) The receptive fields of neurons in the SVAE-Norm model (right) and the original SVAE (left) are shown. The top shows those which did not achieve activity greater than 0.5 for any image in the test set, and the bottom shows the rest. This separates the filters into noise (top) and Gabor (bottom) groups, with up to 100 randomly selected neurons shown. The SVAE has 154 Gabor and 296 noise neurons, while the SVAE-Norm has 374 Gabor and 76 noise neurons.



(b) The top plot shows variance of activation across the test set for the 2 groups of neurons in both models. The bottom plot shows the L_2 norm for both groups and models. Note in the SVAE-Norm model the vectors were normalized to 1.

Figure 3.5: Effect of weight normalization on SVAE decoder. (a) Example of Gabor-like vs. noisy filters; (b) Noisy filters show smaller variances on test set with extremely small norm length compared to the Gabor-like filters.

a visualization of all of the filters learned by the two models on the two datasets, see the supporting information figures below.

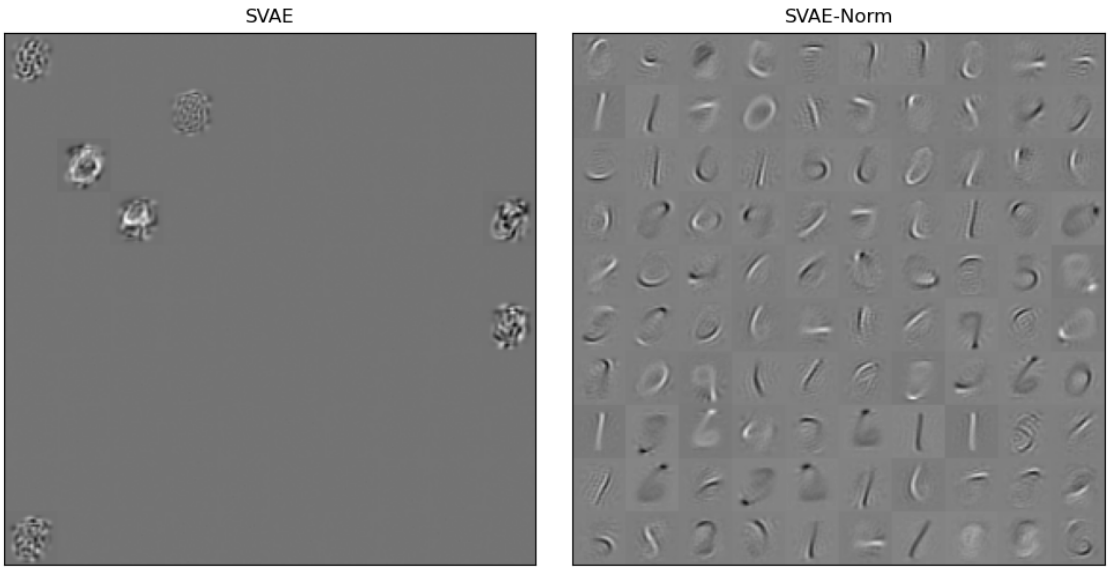


Figure 3.6: 100 randomly sampled receptive fields trained from MNIST data. Left: SVAE; Right: SVAE with weight normalization

3.4 Discussion

In this chapter, we show that a weight normalization scheme for training sparse coding variational autoencoders is critical for decoder optimization. To gain some insights into the efficacy of the approach, the projected gradient descent on the decoder can be thought of as a special case of the weight normalization proposed by Salimans et al. (2016). The weight normalization trick reparameterizes neural network weights $\mathbf{w}$ as

$$\mathbf{w} = \frac{g}{\|\mathbf{v}\|_2}\mathbf{v}$$

where g defines the length of the unit vector $\frac{\mathbf{v}}{\|\mathbf{v}\|_2}$. The SVAE decoder can be thought of as reparameterized weights with $g = 1$. Weight normalization has been shown to accelerate model training and encourage disentangled representation learning. We expect having a unit norm constraint on the SVAE decoder to have similar effects. Future work could focus on verifying the effect of normalization on hierarchical latent variable models, e.g. extending hierarchical sparse coding (Zeiler et al., 2010, 2011) in a VAE setting.

3.5 Supporting information

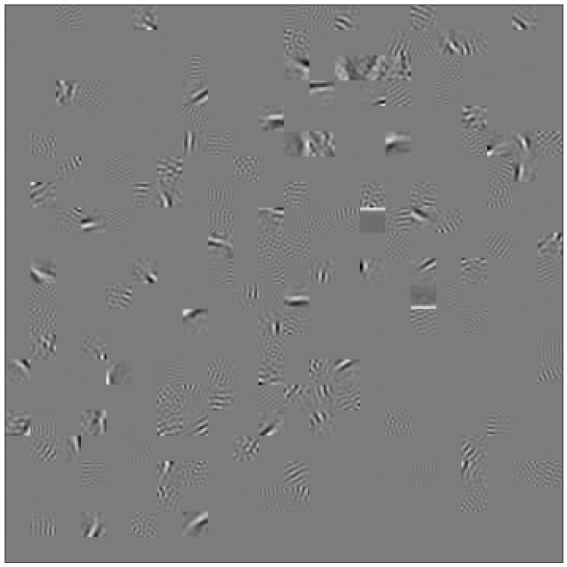


Figure 3.7: All SVAE decoder filters learned on natural image patches

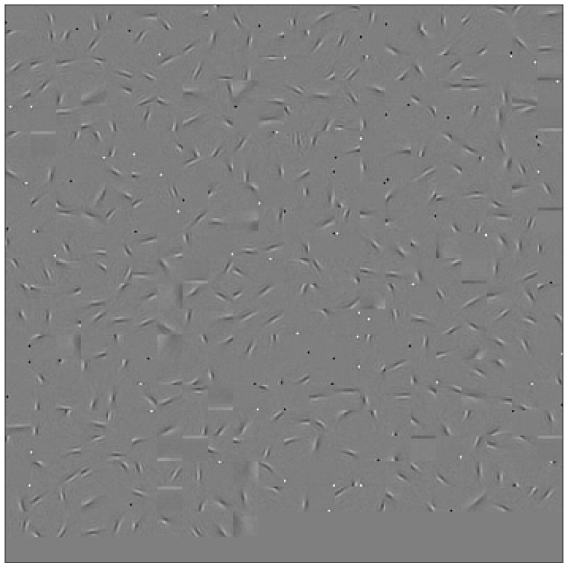


Figure 3.8: All SVAE decoder filters with weight normalization learned on natural image patches

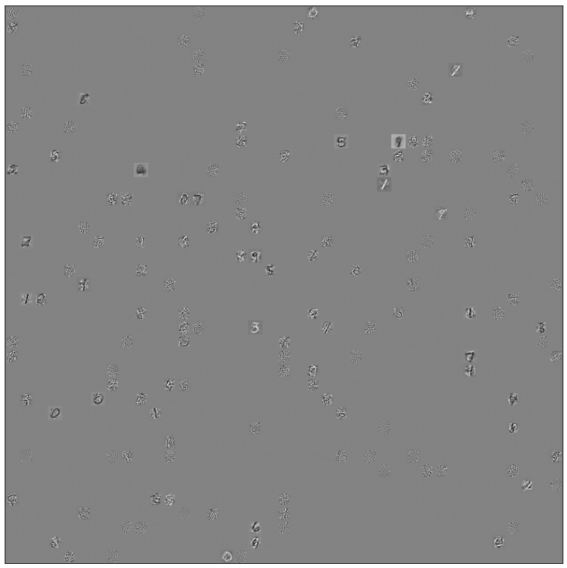


Figure 3.9: All SVAE decoder filters learned on MNIST

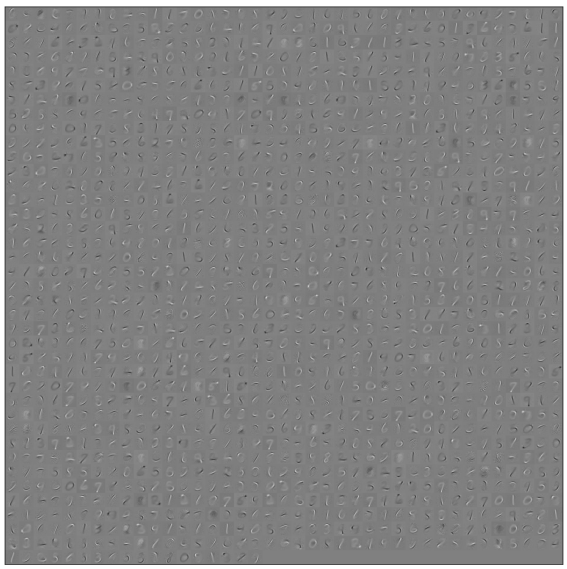


Figure 3.10: All SVAE decoder filters with weight normalization learned on MNIST

Chapter 4

DYNAMIC PREDICTIVE CODING: A MODEL OF HIERARCHICAL SEQUENCE LEARNING AND PREDICTION IN THE NEOCORTEX

In this chapter, we present dynamic predictive coding (DPC), a hierarchical generative model for learning and predicting temporally structured sensory sequences. We show that DPC learns space-time receptive fields, hierarchical temporal representations, predictive and postdictive motion effects, cue-triggered sequence recall, and higher-order temporal abstractions through prediction error minimization.

4.1 Introduction

The ability to predict future stimuli and event outcomes is critical for perceiving and interacting with a highly dynamic world. At the neural circuit level, predictions could compensate for neural transmission delays and engage with the world in real-time. At the cognitive level, planning a sequence of actions to achieve a desired goal relies on predictions of the sensory consequences of motor commands. These abilities are predicated on two requirements: (a) the brain must infer the dynamics of sensory stimuli to make spatiotemporal predictions based on an internal model of the world, and (b) the brain's temporal representations must span different timescales to support predictions over both short and long horizons.

Many experimental studies have provided evidence for such computations. Predictive representations of upcoming stimuli have been found in various open and closed-loop paradigms where animals developed experience-dependent visual and auditory expectations (Xu et al., 2012; Keller et al., 2012; Gavornik and Bear, 2014; Fiser et al., 2016; Schneider et al., 2018). Other empirical evidence suggests that cortical representations exhibit a hierarchy of timescales and an increase in stability from lower-order to higher-order areas across both sensory and cognitive regions (Murray et al., 2014; Runyan et al., 2017; Siegle et al., 2021; Brunec and Momennejad, 2022). We asked the question: could such phenomena be explained by the neocortex learning a spatiotemporal generative model based on a temporal

hierarchy of representations?

Predictive coding provides a unifying framework for understanding perception and prediction in terms of learning hierarchical generative models of the environment (Rao and Ballard, 1999; Friston, 2010; Huang and Rao, 2011; Keller and Mrsic-Flogel, 2018; Jiang and Rao, 2022). Here, we present dynamic predictive coding (DPC), a new predictive coding model for learning hierarchical temporal representations. The central idea of our proposal is that our perceptual system learns temporally abstracted representations that encode entire sequences rather than single points at any given time. Specifically, DPC assumes that higher-level model neurons modulate the *transition dynamics* of lower-level networks, building on the computational concept of *hypernetworks* (Ha et al., 2017). Hypernetworks are neural networks that generate the parameters (synaptic weights) for another neural network. However, generating an entire set of high-dimensional synaptic weights is not neurally plausible. Instead, DPC models the transition dynamics at a lower level of a hierarchy using a small set of modulation weights for a group of learned transition matrices. These weights implement "top-down" gain modulation of the lower-level synapses (Ferguson and Cardin, 2020; Shine et al., 2021) and are predicted by the higher level through a feedback network (a hypernetwork) connecting the higher to the lower level. Compared to previous normative models of video processing that either do not learn the temporal dynamics between images (van Hateren and Ruderman, 1998; Kayser et al., 2001; Olshausen, 2002; Wiskott and Sejnowski, 2002; Berkes and Wiskott, 2005) or presume a fixed temporal hierarchy (Singer et al., 2018, 2023) (see Discussion), the DPC model offers a neural implementation of spatiotemporal prediction that learns the transition dynamics of the input and adapts its hierarchical temporal representation to the intrinsic timescales of the data.

We tested the DPC model using a two-level neural network trained on natural and artificial image sequences to minimize spatiotemporal prediction errors. After training, the lower-level neurons developed space-time receptive fields similar to those found in simple cells in the primary visual cortex (V1) (DeAngelis et al., 1995). Neurons in the second level learned to capture input dynamics on a longer timescale and their responses exhibited greater stability compared to responses in the first level, similar to the temporal response hierarchies observed in the cortex (Murray et al., 2014; Runyan et al., 2017; Siegle et al., 2021;

Brunec and Momennejad, 2022). We further show that the learned sequence representations in the network can explain both predictive and postdictive effects seen in visual processing (Eagleman and Sejnowski, 2000; Hogendoorn et al., 2008; Shimojo, 2014; Hogendoorn, 2022), reproducing several aspects of the flash-lag illusion (Nijhawan, 1994; Eagleman and Sejnowski, 2000; Nijhawan, 2008). When linked to an associative memory mimicking the role of the hippocampus, the network allowed storage of episodic memories and exhibited cue-triggered activity recall after repeated exposure to a fixed input sequence, an effect previously reported in rodents (Xu et al., 2012), human V1 (Ekman et al., 2017; Bang et al., 2018; Lu et al., 2021) and monkey V4 (Eagleman and Dragoi, 2012). Lastly, when extended to three levels, the top-level neurons learned to encode the transition dynamics of the second-level states, which in turn encoded the transition dynamics of the first-level states, thereby yielding a hierarchical temporal representation of input image sequences. Together, these results support the hypothesis that the neocortex uses dynamic predictive coding based on a hierarchical spatiotemporal generative model to learn and interpret input sequences at multiple levels of temporal abstractions. Some of the results presented herein appeared previously in a conference proceedings (Jiang and Rao, 2023).

4.2 Results

4.2.1 Dynamic predictive coding

The DPC model assumes that spatiotemporal inputs are generated by a hierarchical generative model (Fig 4.1a) (see also (George and Hawkins, 2009)). We describe here a two-level hierarchical model (see Discussion for the possibility of extending the model to more levels). The lower level of the model follows the traditional predictive coding model in generating images using a set of spatial filters $\mathbf{U}$ and a latent state vector $\mathbf{r}_t$, which is sparse (Olshausen and Field, 1996), for each time step t: $\mathbf{I}_t = \mathbf{U}\mathbf{r}_t + \mathbf{n}$ where $\mathbf{n}$ is zero mean Gaussian white noise. The temporal dynamics of the state $\mathbf{r}_t$ is modeled using K learnable transition matrices $\{\mathbf{V}_k\}_{k=1}^{K}$ which can be linearly combined using a set of "modulation" weights given by a K-dimensional vector $\mathbf{w}$. This vector of weights is generated by the higher-level state vector $\mathbf{r}^h$ using a function $\mathcal{H}$ (Fig 4.1b), implemented as a neural network (a "hy-

pernetwork" (Ha et al., 2017) – see "Hypernetworks and neural gain modulation" in the supporting information for this chapter):

$$\mathbf{w} = \mathcal{H}(\mathbf{r}^h) \tag{4.1}$$

$$\mathbf{V} = \sum_{k=1}^{K} w_k \mathbf{V}_k. \tag{4.2}$$

Here, w_k is the kth component of the vector $\mathbf{w}$. The lower-level state vector at time $t+1$ is generated as $\mathbf{r}_{t+1} = \text{ReLU}(\mathbf{V}\mathbf{r}_t) + \mathbf{m}$ where $\mathbf{m}$ is zero mean Gaussian white noise. Note that this is one particular parameterization for top-down modulation of the lower-level transition dynamics, with the hypernetwork formulation allowing other types of parameterizations (see "Hypernetworks and neural gain modulation" in the supporting information for this chapter). The generative model in Fig 4.1b can be implemented in a hierarchical neural network: the higher-level state $\mathbf{r}^h$, represented by higher-level neurons, generates a top-down modulation $\mathbf{w}$ via a top-down feedback neural network $\mathcal{H}$, and this top-down input $\mathbf{w}$ influences the groups of lower-level neurons representing $\mathbf{V}_i$ through gain modulation (Ferguson and Cardin, 2020; Shine et al., 2021) (see "Hypernetworks and neural gain modulation" in the supporting information for this chapter for details). We propose that such a computation could be implemented by cortical pyramidal neurons receiving top-down modulation via their apical dendrites (through gain control (Larkum et al., 2004; Shine et al., 2021)) and the recurrent state $\mathbf{r}_t$ (and input prediction errors) via their basal dendrites, and integrating these to predict the next state (Fig 4.1c).

When an input sequence is presented, the model employs a Bayesian filtering approach to perform online inference on the latent vectors (Rao, 1999) by minimizing a loss function that includes prediction errors and penalties from prior distributions over the latent variables (see Methods). Given the model's estimates $\hat{\mathbf{r}}_t$ and $\hat{\mathbf{r}}^h$ at time t, the estimate $\hat{\mathbf{r}}_{t+1}$ of $\mathbf{r}$ at time $t+1$ is computed by gradient descent to minimize the sum of the input prediction error $\|\mathbf{I}_{t+1} - \mathbf{U}\hat{\mathbf{r}}_t\|_2^2$ and the temporal state prediction error $\|\mathbf{r}_{t+1} - \text{ReLU}(\mathbf{V}\hat{\mathbf{r}}_t)\|_2^2$ plus a sparseness penalty. Similarly, the second-level estimate $\hat{\mathbf{r}}^h$ is updated using the temporal prediction error plus a prior-related penalty. The model's parameters are learned by minimizing the same prediction errors across all time steps and input sequences, further reducing the errors

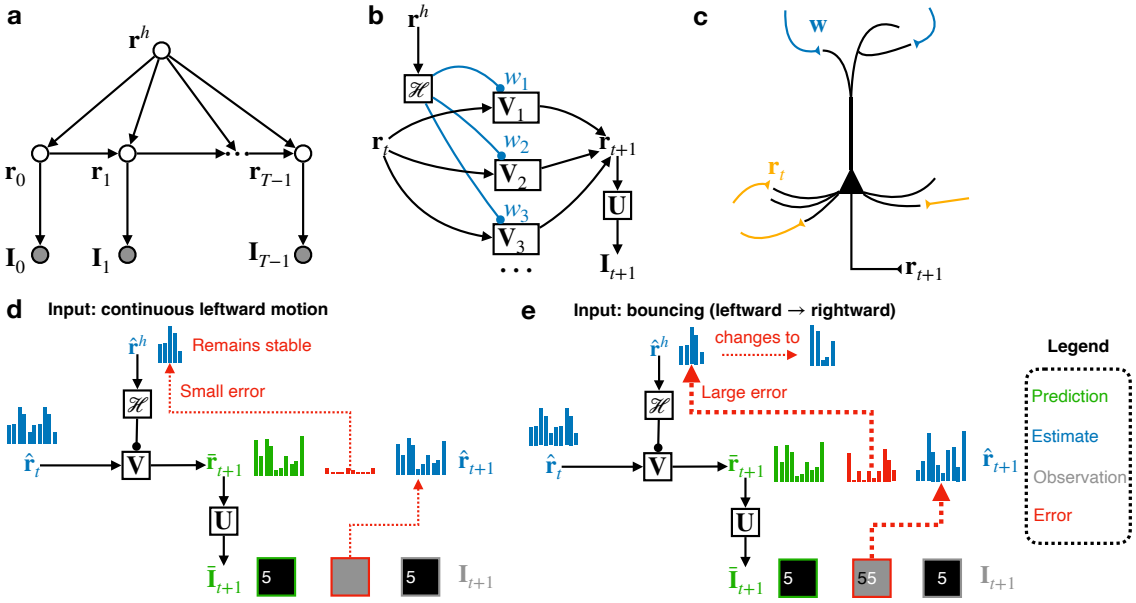


Figure 4.1: **Dynamic predictive coding.** **(a)** Generative model for dynamic predictive coding. **(b)** Parameterization of the model. The higher-level state modulates the lower-level transition matrices through a top-down network ("hypernetwork") $\mathcal{H}$. **(c)** A possible neural implementation of the generative model using cortical pyramidal neurons. Pyramidal neurons receive the top-down embedding vector input via synapses at apical dendrites and the current recurrent state vector via basal dendrites, and produce as their output the next state vector. **(d)** Schematic depiction of an inference step when the dynamics at the lower level is stable. The higher-level state remains stable due to minimal prediction errors. **(e)** Depiction of an inference step when the lower-level dynamics changes. The resulting large prediction errors drive updates to the higher-level state to account for the new lower-level dynamics. (figure from Jiang and Rao, 2024)

not accounted for by the inference process above for latent vectors (see Methods).

4.2.2 Hierarchical predictive coding of natural videos

We implemented the DPC model described above using a two-level neural network where neural responses represent estimates of the latent state vectors and whose synaptic weights

represent the spatial filters and transition parameters. We used $K = 5$ transition matrices for the first level (more matrices did not significantly improve performance – see Improvement on test set loss saturates as the number of transition matrices increases in the supporting information for this chapter). Perception in the DPC network corresponds to estimating the latent vectors by updating neural responses (through network dynamics) to minimize prediction errors via gradient descent (see Methods). Updating network parameters to further reduce prediction errors corresponds to learning (slow changes in synaptic weights through synaptic plasticity).

Fig 4.1d and Fig 4.1e illustrate the inference process for both levels of the network. The network generates top-down and lateral predictions (green) using the current two-level state estimates (blue). If the input sequence is predicted well by the top-down-modulated transition matrix $\mathbf{V}$, the higher-level response $\mathbf{r}^h$ remains stable due to small prediction errors (Fig 4.1d). When a non-smooth transition occurs in the input sequence, the resulting large prediction errors are sent to the higher level via feedforward connections (red arrows, Fig 4.1e), driving changes in $\mathbf{r}^h$ to predict new dynamics for the lower level.

We trained the network on thousands of natural image sequences extracted from a video recorded by a person walking on a forest trail (frame size: 16×16 pixels, sequence length: 10 frames ($\sim$0.35 seconds)). The frames were spatially and temporally whitened to simulate retinal and lateral geniculate nucleus (LGN) processing (Olshausen and Field, 1996; Dong and Atick, 1995a). The image sequences reserved for testing did not overlap in space or time with the training sequences. Fig 4.2a illustrates the inference process on an example natural image sequence by the network. The first row displays the ground truth input $\mathbf{I}_t$ for 10 time steps: each frame was shown sequentially to the model. The next row shows the model's predictions $\mathbf{U}\bar{\mathbf{r}}_t$ for each time step t, where $\bar{\mathbf{r}}_t$ was predicted by the previous state estimate $\hat{\mathbf{r}}_{t-1}$: $\bar{\mathbf{r}}_t = \text{ReLU}(\mathbf{V}\hat{\mathbf{r}}_{t-1})$. The prediction errors $\mathbf{I}_t - \mathbf{U}\bar{\mathbf{r}}_t$ are shown in the third row. The prediction errors were the largest in the first two steps as the model inferred the spatial features and the transition dynamics from the initial inputs. The subsequent predictions were more accurate, resulting in minimized prediction errors. Finally, the last row shows the corrected estimates $\mathbf{U}\hat{\mathbf{r}}_t$ after $\bar{\mathbf{r}}_t$ has been updated to $\hat{\mathbf{r}}_t$ through prediction error minimization. Fig 4.2b shows the lower- (top) and higher-level

(middle) neural responses to the natural video sequence in Fig 4.2a. The bottom panel of Fig 4.2b shows the top-down dynamics modulation generated by the higher level.

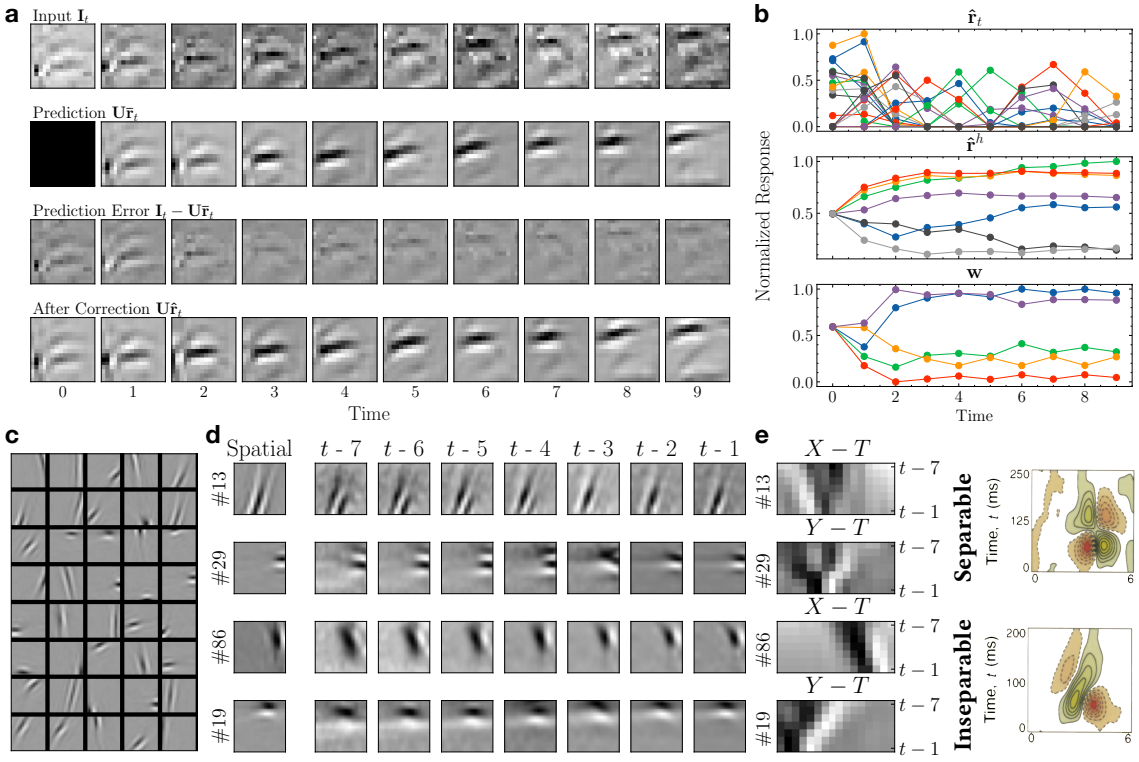


Figure 4.2: Caption continued at next page

We examined the learned spatial receptive fields (RFs) of the model neurons at the first level and qualitatively compared them with the spatial RFs of simple cells in the primary visual cortex (V1). A subset of the spatial filters (columns of $\mathbf{U}$) learned by the model from our natural videos dataset are shown in Fig 4.2c. These filters resemble oriented Gabor-like edge or bar detectors, similar to the localized, orientation-selective spatial RFs found in V1 simple cells (Hubel and Wiesel, 1959; Olshausen and Field, 1996). To measure the spatiotemporal receptive fields of the lower-level neurons, we ran a reverse correlation experiment (Ringach and Shapley, 2004; Talebi and Baker, 2012) with a continuous natural video clip ($\approx$ 47 minutes) extracted from the same forest trail natural video used for training. This video was not shown to the model during either training or testing (see

Figure 4.2: (Previous page.) **Predictive coding of natural videos and learned space-time receptive fields**. **(a)** Inference on an example input image sequence of 10 frames. Top to bottom: Input sequence; model's prediction of the current input from the previous step (the first step prediction being zero); prediction error (predicted input subtracted from the actual input); model's final estimate of the current input after prediction error minimization. **(b)** The trained DPC network's response to the natural image sequence in (a). Each plotted line represents the responses of a model neuron over 10 time steps. Top: responses of the 20 most active lower-level neurons (some colors are repeated); middle: responses of seven randomly chosen higher-level neurons; bottom: predicted transition dynamics (each line is the modulation weight for a basis transition matrix at the lower level). **(c)** 40 example spatial receptive fields (RFs) learned from natural videos. Each square tile is a column of $\mathbf{U}$ reshaped to a 16×16 image. **(d)** Space-Time RFs (STRFs) of four example lower-level neurons. First column: the spatial RFs of the example neurons. Next seven columns: the STRFs of the example neurons revealed by reverse correlation mapping. **(e)** Left panel: space-time plots of the example neurons in (d). Right panel: space-time plots of the RFs of two simple cells in the primary visual cortex of a cat (adapted from DeAngelis et al. (1995)). (figure from Jiang and Rao, 2024)

Methods). Fig 4.2d shows the spatiotemporal receptive fields for four example lower-level model neurons, computed by weighting input frames from the seven previous time steps $\mathbf{I}_{t-7}, \mathbf{I}_{t-6}, \ldots, \mathbf{I}_{t-1}$ by the response $\hat{\mathbf{r}}_t$ they caused at the current time step t (see Methods). The resulting average spatiotemporal receptive fields are shown as seven-image sequences labeled $t-7, t-6, \ldots, t-1$ (lasting ≈ 250 milliseconds in total). The first column labeled "Spatial" shows the spatial RFs of the example neurons.

To compute the space-time receptive fields (STRFs), we took the spatiotemporal $X-Y-T$ receptive field cubes and collapsed either the X or Y dimension, depending on which axis had time-invariant responses. Fig 4.2e left panel shows the $X/Y-T$ receptive fields of these example neurons. For comparison, Fig 4.2e right panel shows the STRFs of simple

cells in the primary visual cortex (V1) of a cat (adapted from DeAngelis et al. (1995)).

DeAngelis et al. (1995) categorized the receptive fields of simple cells in V1 to be space-time separable (Fig 4.2e top row) and inseparable (Fig 4.2e bottom row). Space-time separable receptive fields maintain the spatial form of bright/dark-excitatory regions over time but switch their polarization: the space-time receptive field can thus be obtained by multiplying separate spatial and temporal receptive fields. Space-time inseparable receptive fields on the other hand exhibit bright/dark-excitatory regions that shift gradually over time, showing an orientation in the space-time domain. Neurons with space-time inseparable receptive fields are direction-selective, responding to motion in only one direction. As seen in Fig 4.2e left panel, the neurons in the lower level of our network learned V1-like separable and inseparable STRFs, based on the principle of spatiotemporal prediction error minimization. To our knowledge, these results represent one of the first demonstrations of the emergence of both separable and inseparable STRFs in a recurrent network model by predictive coding of natural videos presented frame-by-frame. Previous demonstrations (e.g., (van Hateren and Ruderman, 1998; Olshausen, 2002; Singer et al., 2018)) have typically required chunks or all the frames of a video to be provided as a single input to a network, which is hard to justify biologically (see Discussion).

4.2.3 Temporal hierarchy through prediction of dynamics

Next, we show that the two-level DPC network learned a hierarchical temporal representation of input videos. In our formulation of the model, a higher-level state vector predicts the *dynamics* of the lower-level states. This implies that the higher-level network neurons will have stable activation for input sequences with consistent dynamics (Fig 4.1d). When a change occurs in input dynamics, we expect the higher-level responses to switch to a different activation profile to minimize prediction errors (Fig 4.1e). We hypothesize that the different timescales of neural responses observed in the cortex (Murray et al., 2014; Runyan et al., 2017; Siegle et al., 2021; Brunec and Momennejad, 2022) could be an emergent property of the cortex learning a similar hierarchical generative model.

We tested this hypothesis in our DPC network trained on natural videos. As seen in

the inference example in Fig 4.2b, the lower-level responses change rapidly as the stimulus moves (top panel). The higher-level responses (middle panel) and the predicted transition dynamics (right panel) were more stable after the initial adaptation to the motion. Since the stimulus continued to follow roughly the same dynamics (leftward motion) after the first two steps, the transition matrix predicted by the higher-level neurons continued to be accurate for the rest of the steps, leading to small prediction errors and few changes in the responses. Note that we did not enforce a longer time constant or smoothness constraint for $\mathbf{r}^h$ during inference – the longer timescale and more stable responses are entirely a result of the higher-level neurons learning to predict the lower-level dynamics under the proposed generative model.

To quantify this learned hierarchical temporal representation, we computed the autocorrelation of the lower- and higher-level responses to unseen natural videos and fitted an exponential decay function (see Methods). As Fig 4.3a shows, the autocorrelation of the higher-level responses $\mathbf{r}^h$ is greater than that of the lower-level response $\mathbf{r}$ and has a slower decay rate (exponential time constant for $\mathbf{r}^h$: 5.49 steps; for $\mathbf{r}$: 2.18 steps). To factor out the effect of the natural video statistics, we computed the same autocorrelation using Gaussian white noise sequences. We found that the hierarchy of timescales still exists ($\mathbf{r}^h$: 3.11 steps; $\mathbf{r}$: 0.18 steps). Fig 4.3b shows the autocorrelation of nonhuman primate neural responses from the medial-temporal (MT) area in the visual cortex (assumed to be lower in the processing stream) and lateral prefrontal cortex (LPFC) (assumed to be higher) for time periods preceding a motion task (Murray et al., 2014). These neural responses show a difference in timescales qualitatively similar to the different response timescales in our hierarchical DPC network. To further understand the model's ability to learn hierarchical temporal representations, we trained a DPC network on the Moving MNIST dataset (Srivastava et al., 2015). Each image sequence in this dataset contains ten 18×18 pixel frames showing a single example of a handwritten digit (chosen from the original MNIST dataset) moving in a particular direction. The digit's motion is limited to up, down, left, or right directions with a fixed speed. Fig 4.3c illustrates the trained network's inference process on an example image sequence. Similar to the responses to the natural video sequence, the lower-level responses displayed fast changes while the higher-level responses spanned a

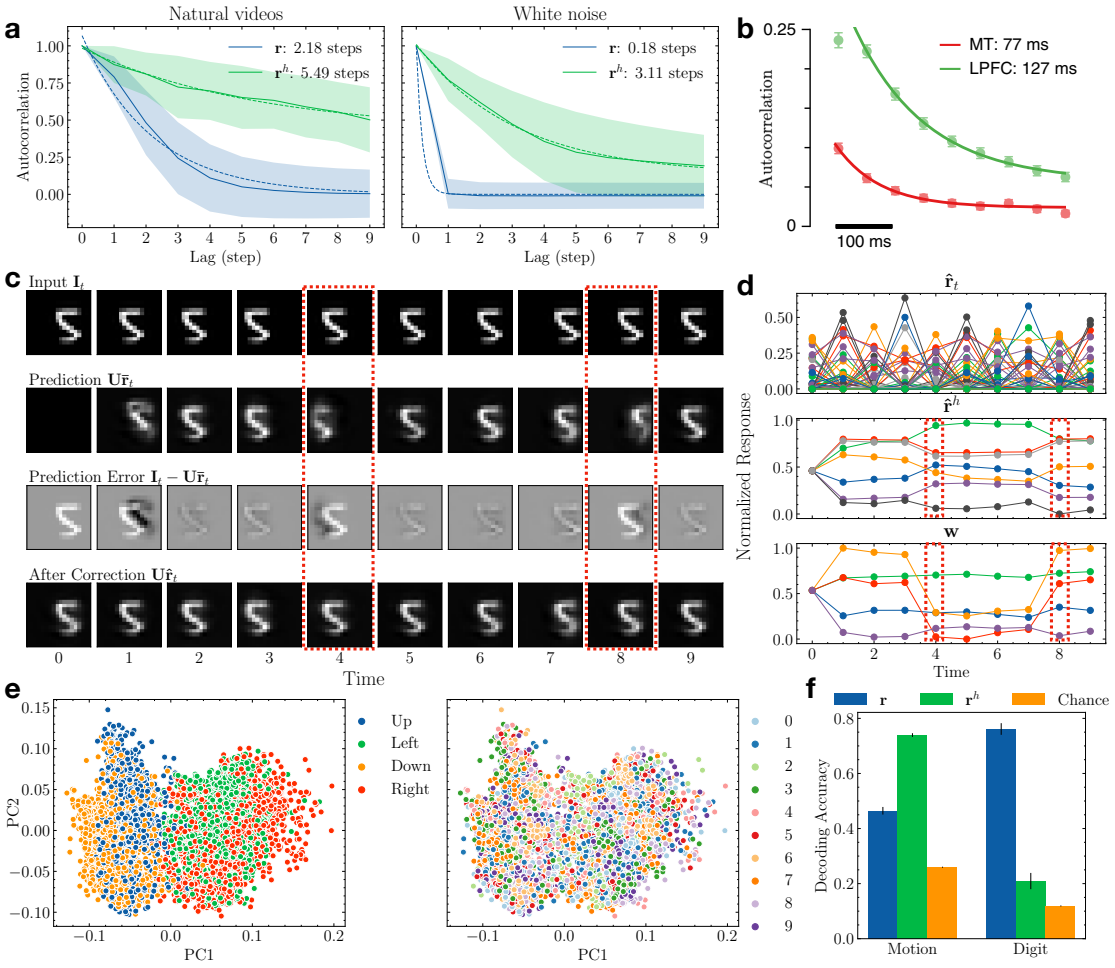


Figure 4.3: Caption continued at next page

longer timescale and showed greater stability (Fig 4.3d). Note that at time $t = 4$ and $t = 8$, the input dynamics changed as the digit "bounced" against the boundaries and started to move in the opposite direction (Fig 4.3c red dashed box). The higher-level neurons' predictions resulted in large prediction errors at those times (Fig 4.3c third row). The prediction errors caused notable changes in the higher-level responses $\mathbf{r}^h$ (Fig 4.3d red dashed boxes). For the rest of the steps, $\mathbf{r}^h$ remained stable and generated accurate predictions of the stable dynamics.

Lastly, we confirmed that lower-level transition dynamics are indeed encoded in the

Figure 4.3: (Previous page.) **Hierarchical temporal representation with different timescales. (a)** Autocorrelation of the lower- and higher-level responses in the trained network with natural videos. Shaded area denotes ±1 standard deviation. Dotted lines show fitted exponential decay functions. Left: response recorded during natural video stimuli; right: white noise stimuli. **(b)** Autocorrelation of the neural responses recorded from MT and LPFC of monkeys. Adapted from Murray et al. (2014) **(c)** Inference for an example Moving MNIST sequence in a trained network. The red dashed boxes mark the time steps when the dynamics of the input changed. **(d)** The network's responses to the input Moving MNIST sequence in (c). Note the changes in the higher-level responses after the input dynamics changed (red dashed boxes); this gradient-based change helps to minimize prediction errors. **(e)** Higher-level responses to the Moving MNIST sequences visualized in the 2D space of the first two principal components. Left: responses colored according to motion direction; right: responses colored according to digit identities. **(f)** Comparison of decoding performance for motion direction versus digit identity using lower- and higher-level neural responses. Error bars: ±1 standard deviation from 10-fold cross validation. Orange: chance accuracies. (figure from Jiang and Rao, 2024)

higher-level responses. We performed principal component analysis (PCA) on the higher-level responses $\mathbf{r}^h$ for the Moving MNIST sequences in the test set. Fig 4.3e visualizes these responses in the space of their first two principal components (PCs), colored by either the motion direction (left) or digit identities (right). The responses clearly formed clusters according to input motion direction but not digit identities. We then trained a support vector machine with radial basis function (RBF) kernel (Murphy, 2012) to map $\mathbf{r}$ and $\mathbf{r}^h$ to motion directions and digit identities (Fig 4.3f). Using the higher-level responses, the classifier yielded 73.9% 10-fold cross-validated classification accuracy on the four motion directions (chance accuracy: 26.0%, computed as the number of majority labels in the test set divided by the total number of labels). Using the lower-level responses resulted in significantly less classification accuracy for motion direction (46.5%, $p \ll 0.001$, t-test). In

contrast, decoding accuracy for digit identity was significantly higher using the lower-level responses (76.1%) compared to using the higher-level responses (20.9%, $p \ll 0.001$, t-test). These results show that due to the structure of its generative model, the DPC network learned to disentangle to a significant extent the motion information in an input video from image content (here, digit identity), yielding a factored representation of input image sequences.

4.2.4 Predictive and postdictive effects in visual motion processing

The ability of the DPC model to encode entire sequences at the higher level (cf. the "timeline" model of perception (Hogendoorn, 2022)) leads to new normative and computational interpretations of visual motion phenomena such as the flash-lag illusion (Nijhawan, 1994; Eagleman and Sejnowski, 2000; Nijhawan, 2008), explaining both predictive and postdictive effects (Hogendoorn et al., 2008; Hogendoorn, 2022). The flash-lag illusion refers to the phenomenon that a flashed, intermittent object is perceived to be "lagged" behind the percept of a continuously moving object even though the physical locations of the two objects are aligned or the same (Nijhawan, 1994, 2008). Though this illusion is commonly attributed to the predictive nature of the perceptual system (Nijhawan, 1994), Eagleman and Sejnowski (2000) proposed a postdictive mechanism based on psychophysical results that the motion of the moving object *after* the flash can change the percept of events at the time of the flash. The potential interplay between prediction and postdiction in shaping perception was also studied by Hogendoorn et al. (2008; 2022). The authors designed an interference paradigm with different reaction-speed tasks and showed that when the trajectory of the object unexpectedly reverses, predictive effects (extrapolation) are observed at short latencies but postdictive effects (interpolation) are observed at longer latencies (Fig 4.4i and Fig 4.4j). We propose that prediction error minimization with a hierarchical temporal representation, as in the DPC model, provides a natural explanation for these predictive and postdictive effects. In a DPC network, the higher-level state $\mathbf{r}^h$ predicts entire sequences of lower-level states following the same dynamics (Fig 4.3). When the dynamics of observations change (e.g., motion reversal), the higher-level state is updated to minimize prediction errors, re-

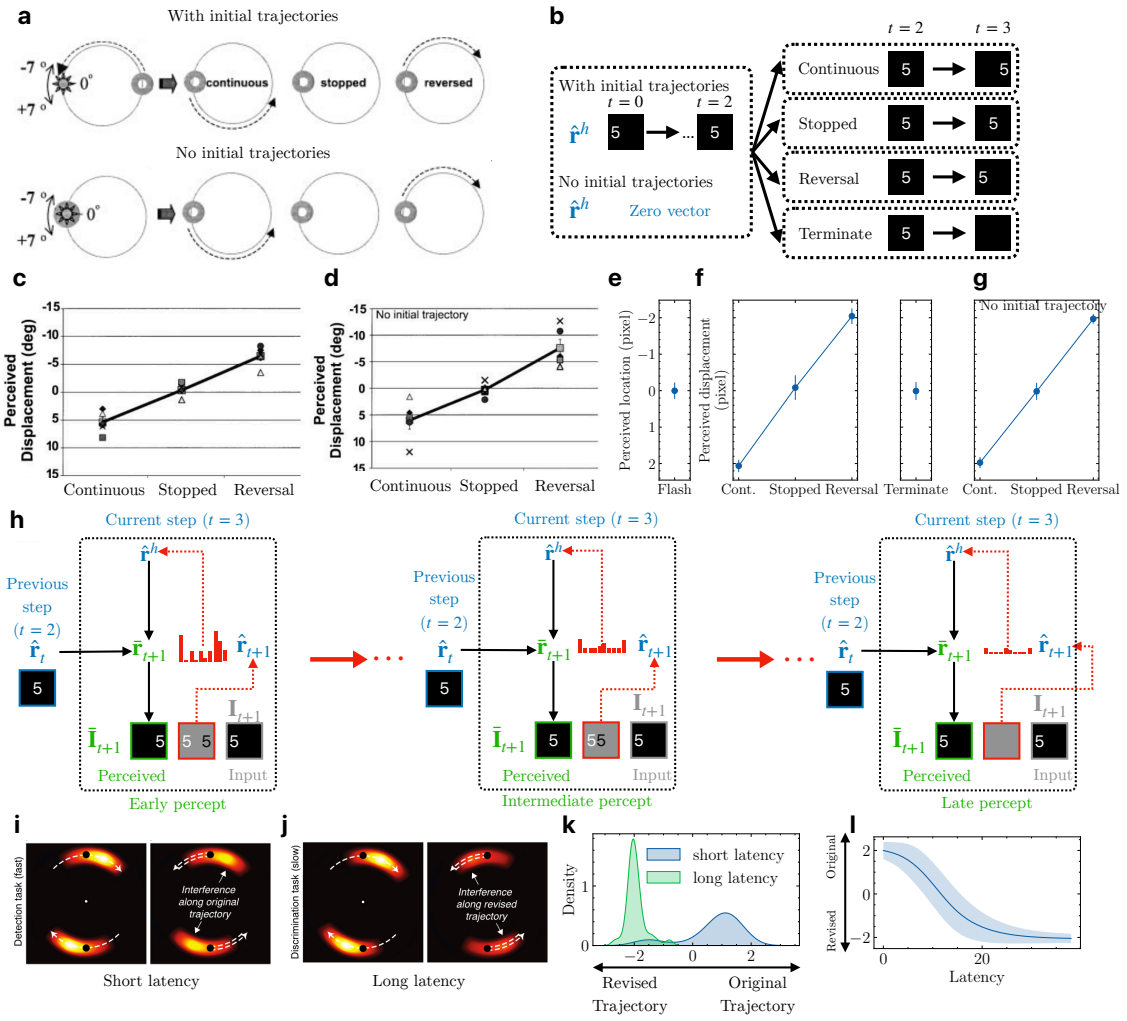


Figure 4.4: Caption continued at next page

sulting in a revised state that represents the motion-reversed sequence spanning both past and future inputs. This process corresponds to postdiction in visual processing (Shimojo, 2014). For the flash-lag experiment, we predict that the higher-level neurons of a trained DPC network will form a static sequence percept when presented with a flashed object and a directional sequence percept for a moving object, causing perceived lags between the two objects as observed in the flash-lag illusion (Nijhawan, 1994).

We first test these predictions of the DPC model on the experimental conditions used

Figure 4.4: (Previous page.) **Flash-lag illusion and object representations in apparent motion. (a)** The flash-lag test conditions used by Eagleman and Sejnowski (2000). The moving ring could have an initial trajectory (top) or no trajectory (bottom). At the time of the flash (bright disk), the ring could move along the initial trajectory, stop, or reverse its trajectory. Adapted from Eagleman and Sejnowski (2000). **(b)** Two test conditions (left) regarding initial trajectories of the moving object (a digit) in the flash-lag experiment with the model, and four test conditions (right) for the moving object. The flashed object was shown at time t and turned off at time $t+1$ (same as the "Terminate" condition). **(c & d)** Psychophysical estimates for human subjects reported by Eagleman and Sejnowski (2000) when the moving object had initial trajectories (c) or no initial trajectory (d). **(e)** Perceived location of the flashed object in the DPC model at time $t+1$. The error bar indicates ± 1 standard deviation (measured across presentations of different digits). **(f)** Perceived displacement between the moving object (with initial trajectories) and the flashed object in the DPC model for the four test conditions. **(g)** Same as (f) but with no initial trajectory for the moving object. **(h)** Illustration of the prediction-error-driven dynamics of the perception of the moving object in the model when the trajectory reversed at time $t+1$. Red ellipsis between panels denotes the prediction error minimization process. **(i)** Interference pattern during human apparent motion perception with continuous motion (left) and reversed motion (right) at short latency (fast detection task). Brighter color denotes more interference. Dashed arrows represent object motion direction. Adapted from Hogendoorn (2022). **(j)** Same as (i) but at long latency (slow discrimination task) (Hogendoorn, 2022). **(k)** Perceived location of the moving object in the DPC model at time $t+1$ probed at short versus long latency during prediction error minimization. Positive values denote distance along the original trajectory. Negative values denote distance along the reversed trajectory. Short and long latency correspond to "Early percept" and "Late percept" respectively in part (h). **(l)** Perceived location of the digit at all latencies during the prediction error minimization process in part (h). (figure from Jiang and Rao, 2024)

by Eagleman and Sejnowski (2000). In their experiment, the stimuli consisted of a flashed disk and a ring moving in a circle. Before the flash, the ring could have an initial trajectory (Fig 4.4a top) or no initial trajectory (Fig 4.4a bottom). After the flash, the ring could continue moving on its initial trajectory ("continuous"), stop moving ("stopped"), or move on the reversed trajectory ("reversed"). A flash appeared in a seven-degree range that ex-

tended above and below the ring on its trajectory. The participants then indicated whether a flashed white disk occurred above or below the center of the moving ring. Positive displacements denoted lags along the initial trajectory of the ring, while negative displacements denoted the reversed direction.

To simulate these testing conditions, we used the Moving MNIST test set and extracted 134 test sequences with consistent leftward or rightward motion. For each of these 134 sequences, we simulated the two test conditions used by Eagleman and Sejnowski (2000) (with or without initial trajectory): the higher-level state $\hat{\mathbf{r}}^h$ was either inferred from the first three steps ($t = 0, 1, 2$) of the input sequence, or initialized to the zero vector (Fig 4.4b left). For each of these two test conditions, we simulated the four test cases used in Eagleman and Sejnowski (2000) regarding the motion of the moving object at the time of the flash (Fig 4.4b right). Note that flashed stimuli correspond to the "no initial trajectory, terminate" condition. We computed the location of a digit as the center of mass of pixel values in the 2D image; the perceived location at time t was defined similarly based on the predicted image at time t. As Fig 4.4e shows, the perceived location of a flashed object at $t = 3$ strongly overlapped with the physical flashed location at $t = 2$, showing that the prediction errors drove the higher-level state estimates to predict no change in object location for the flashed object. Fig 4.4f shows the perceived displacement between the moving object (with initial trajectories) and the flashed object, computed as the difference in perceived locations at $t = 3$ between the moving object and the flashed object. Positive displacements followed the original trajectory direction and negative displacements followed the reversed direction. The perceived displacements in the model were significantly different in the three test conditions (Fig 4.4f left panel, $p \ll 0.001$, one-way ANOVA test) and were similar to the psychophysical results reported by Eagleman & Sejnowski (Fig 4.4c). Fig 4.4g confirms that the initial trajectories of the moving object had no effects on the model's flash-lag illusion, consistent with the reported results (Fig 4.4d) (Eagleman and Sejnowski, 2000). These results validate the explanation provided by the DPC model on the flash-lag effect: for a hierarchical generative model with representations of sequences, a flashed or stopped/terminated moving object leads to inference of a static object sequence (Fig 4.4e), while continuous or reversed motion leads to inference of a moving object se-

quence, resulting in the perceived lags along the corresponding directions (Fig 4.4f).

One aspect of motion perception the previous results do not illustrate is the interplay between postdiction and prediction. Hogendoorn et al. investigated this effect in an experiment on apparent motion perception (Hogendoorn et al., 2008). Participants were instructed to report the detection of a visual cue (short latency task) or differentiate between two visual cues (long latency task) during apparent motion. These visual cues could either be along the apparent motion trajectory or the reversed trajectory. The authors found that upon reversing the apparent motion trajectory, predictive effects dominated perception at short latency (detection task, Fig 4.4i), with the most interference (measured in terms of the participants' reaction times) along the original motion trajectory. At longer latency (differentiation task, Fig 4.4j), most interference was along the reversed trajectories, indicating that postdictive effects dominated perception.

We hypothesize that the prediction error minimization process of DPC could explain this interplay between prediction and postdiction, as illustrated by Fig 4.4h which depicts the gradient-descent-based optimization process of Fig 4.1e (and Eq 4.18). Early percepts of the model are dominated by the spatiotemporal prediction using the optimal estimates from the previous step (Fig 4.4h left). When a motion reversal occurs, feedforward prediction errors gradually correct the second-level state (Fig 4.4h middle) until convergence (Fig 4.4h right). Therefore, late percepts in the model correspond to error-corrected spatiotemporal predictions. Note that due to the discrete temporal nature of the DPC model (unit time steps), this process is considered to happen "at" one particular time step (*e.g.*, early versus late percept "at" $t = 3$ in Fig 4.4h).

To test this hypothesis, we used the same trained DPC network and probed its percept of the moving object at the time of reversal under the "with initial trajectory, reversal" condition (Fig 4.4b). At short latency (10% of steps into prediction error correction, Fig 4.4h early percept), the perceived locations for the moving object in most test sequences were along the original trajectory, as denoted by positive displacements compared to the final step before reversal ($t = 2$) (Fig 4.4k blue). At longer latency (90%, Fig 4.4h late percept), the moving object's perceived locations were flipped and along the reversed trajectory (negative displacements; Fig 4.4k green, $p \ll 0.001$, t-test). This is consistent with psychophysical

findings (Hogendoorn et al., 2008; Hogendoorn, 2022) that when the motion of the object unexpectedly reversed, prediction effects were observed at short latency ($\approx$ 350 ms, Fig 4.4i right panel, bright color denotes locations of interference due to prediction) while postdictive effects were observed at longer latency ($\approx$ 620 ms, Fig 4.4j right panel, bright color denotes locations of interference due to postdiction). Fig 4.4l plots the moving object's perceived location in our model throughout the error correction process: the perceived location varies smoothly from being along the original direction initially to along the reversed direction at greater latencies. These results make a testable prediction: if probed at an intermediate level of latency (between 350 ms and 620 ms), the maximal interference should overlap with the object's location at the time of reversal (*i.e.*, at the black dots in Fig 4.4i and Fig 4.4j), as suggested by Fig 4.4l.

4.2.5 Cue-triggered recall and episodic memory

A number of experiments in rodents have shown that the primary visual cortex (V1) encodes predictive representations of upcoming stimuli (Xu et al., 2012; Keller et al., 2012; Gavornik and Bear, 2014; Fiser et al., 2016; Jordan and Keller, 2020). In one of the first such studies, Xu et al. (2012) demonstrated that after exposing rats repeatedly to the same moving dot visual sequence (Fig 4.5a), displaying only the starting dot stimulus triggered sequential firing in V1 neurons in the same order as when displaying the complete sequence (Fig 4.5a). Similar effects have been reported in monkey (Eagleman and Dragoi, 2012) and human (Ekman et al., 2017; Bang et al., 2018; Lu et al., 2021; Ekman et al., 2022) visual cortical areas as well. The generative model of DPC provides a highly efficient computational basis for episodic memories and sequence prediction. DPC assumes sequences are generated by a factorized representation: a single (lower-level) representation of the content ("what") provided at the first step and a single (higher-level) representation of dynamics (motion or "where"). These two representations are inferred during sequence perception as the explanations (or causes) of a given input sequence.

It is known that factored information from the visual cortex makes its way, via the medial and lateral entorhinal cortices, to the hippocampus (Manns and Eichenbaum, 2006).

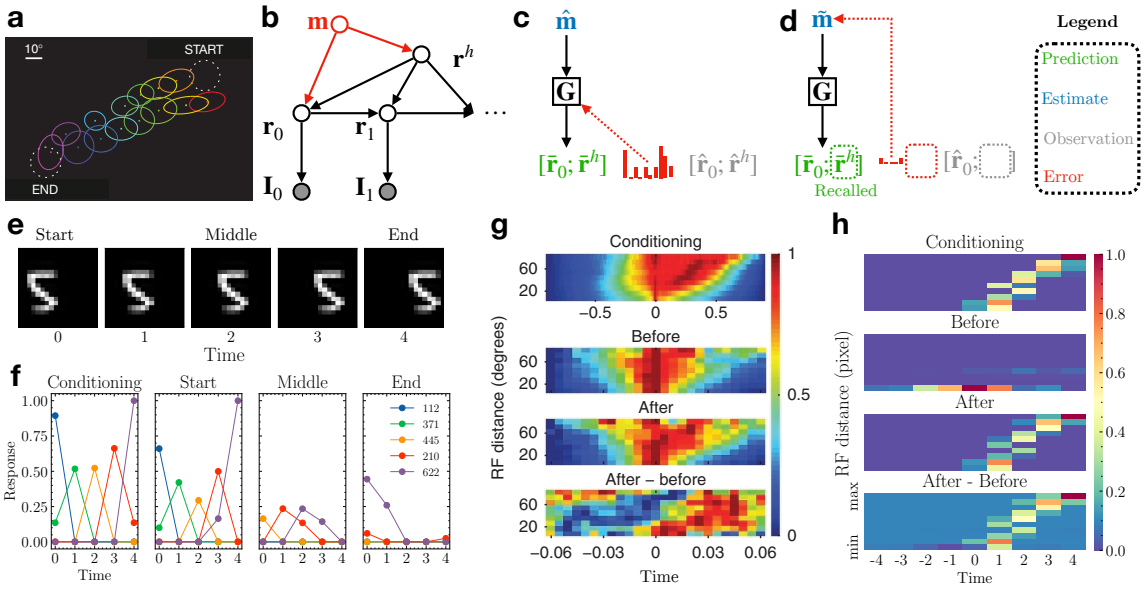


Figure 4.5: Caption continued at next page

The hippocampus has been implicated both in the formation of episodic memories (Burgess et al., 2002; Tulving, 2002; Sugar and Moser, 2019) and in mediating activity recall in the neocortex (Gelbard-Sagiv et al., 2008; Bosch et al., 2014; Hindy et al., 2016; Barron et al., 2020) through its outputs to the entorhinal cortex, which in turn conveys this information to downstream areas via feedback connections. Because the DPC model encodes an entire sequence in terms of a single dynamics vector $\mathbf{r}^h$ (along with the content $\mathbf{r}_0$ at the first step), it suggests a simple mechanism for storing sequential experiences as episodic memories, namely, storing the vector $\mathbf{r}^h$ (along with $\mathbf{r}_0$) in an associative memory, mimicking the role of the hippocampus.

To test this hypothesis, we augmented the DPC model with an associative memory that uses a vector $\mathbf{m}$ to bind the content vector $\mathbf{r}_0$ with the dynamics of the sequence $\mathbf{r}^h$, thereby encoding an episodic memory: the new generative model is shown in Fig 4.5b. Given the initial cue $\mathbf{r}_0$ (inferred from the first image frame in the sequence), the associative memory (emulating the hippocampus) recalls the episodic sequence dynamics $\mathbf{r}^h$ which modulates the transition dynamics in the DPC network (representing the visual cortex) to

Figure 4.5: (Previous page.) **Cue-triggered activity recall in the DPC model. (a)** The experimental setup of Xu et al. (adapted from Xu et al. (2012)). A bright dot stimulus moved from START to END repeatedly during conditioning. Activities of neurons whose receptive fields (colored ellipses) were along the dot's trajectory were recorded. **(b)** Generative model combining an associative memory and DPC. The red part denotes the augmented memory component that binds the initial content vector $\mathbf{r}_0$ and the dynamics vector $\mathbf{r}^h$ to encode an episodic memory. **(c)** Depiction of the memory encoding process. The presynaptic memory activity and postsynaptic prediction error jointly shape the memory weights $\mathbf{G}$. **(d)** Depiction of the recall process. Prediction error on the partial observation $\hat{\mathbf{r}}_0$ drives the convergence of the memory estimates $\tilde{\mathbf{m}}$ and recalls the higher-level dynamics vector $\bar{\mathbf{r}}^h$ as a top-down prediction. The red dotted box depicts the prediction error between the missing observations for $\mathbf{r}^h$ and the prediction $\bar{\mathbf{r}}^h$; this error is ignored during recall, implementing a form of robust predictive coding (Rao, 1998). **(e)** The image sequence used to simulate conditioning and testing for our memory-augmented DPC network. **(f)** Responses of the lower-level neurons of the network. Colored lines represent the five most active lower-level neurons at each step. Left to right: neural responses during conditioning, testing the network with a single start frame, middle frame, and end frame. **(g, h)** Normalized pairwise cross correlation of **(g)** primary visual cortex neurons (adapted from Xu et al. (2012)) and **(h)** the lower-level model neurons. Top: during conditioning; middle two: testing with the starting stimulus, before and after conditioning; bottom: the differences between cross correlations, "After" minus "Before" conditioning. (figure from Jiang and Rao, 2024)

complete the sequence recall. Specifically, we added to the trained DPC network another higher level of predictive coding to implement associative memory (Rao and Ballard, 1999; Salvatori et al., 2021): the memory vector estimate $\hat{\mathbf{m}}$ predicts both the content vector $\hat{\mathbf{r}}_0$ and motion dynamics vector $\hat{\mathbf{r}}^h$ and uses the prediction error to correct itself (Fig 4.5c). Upon convergence of $\hat{\mathbf{m}}$, the associative memory network stores this vector by updating its weights $\mathbf{G}$ using Hebbian plasticity based on presynaptic activity $\hat{\mathbf{m}}$ and postsynaptic

prediction error (Rao and Ballard, 1999) (Fig 4.5c, see Methods). During recall, the memory estimate $\tilde{\mathbf{m}}$ is driven by the $\hat{\mathbf{r}}_0$ inferred from the cue and the prediction error (Fig 4.5d, dashed boxes denote the missing input). The dynamics vector $\mathbf{r}^h$ is then recalled as the top-down prediction after $\tilde{\mathbf{m}}$ has converged (Fig 4.5d, green dashed box). Note that during conditioning, no learning occurs in the DPC network – only the weights of the memory network $\mathbf{G}$ are optimized to store the episodic memory $\hat{\mathbf{m}}$.

We simulated the experiment of Xu et al. (2012) using a moving MNIST sequence from the test set shown in Fig 4.5e. After conditioning (5 repetitions of the sequence), the network was tested with the starting frame only, the middle frame only, and the end frame only. The lower-level responses $\hat{\mathbf{r}}_0$ of the DPC network were used to recall the dynamics component $\bar{\mathbf{r}}^h$ from the memory. The recalled dynamics were then used to predict a sequence of lower-level responses in the DPC network. We found that the lower-level model neurons exhibited cue-triggered activity recall given only the start frame of the sequence (Fig 4.5f Start). Cueing the network with the middle frame triggered weak recall, consistent with findings by Xu et al. (see Fig 3c in Ref Xu et al., 2012). The end frame did not trigger recall (Xu et al., 2012). We found that the sequence recall is cue-specific – when trained with sequences that have distinct digits and dynamics, the DPC network successfully recalled the correct sequence when cued with different starting digits (Cue-triggered recall is cue-specific in the supporting information for this chapter).

Lastly, following the analysis done by Xu et al. (2012), we plotted the pairwise cross-correlation of the lower-level model neurons as a function of their spatial RF distances when tested with the starting frame of the sequence (see Methods). As Fig 4.5h shows, the peaks of the correlation showed a clear rightward slant after conditioning, consistent with the experimental results (Fig 4.5g). This indicates a strong sequential firing order in the lower-level model neurons elicited by the starting cue, where neurons farther apart have longer lags in response cross-correlations, a phenomenon that was nonexistent before conditioning (Fig 4.5g). These simulation results support our hypothesis that cue-triggered recall could be the result of the hippocampus, acting as an associative memory, binding factorized sequence representations of content and dynamics from the neocortex and recalling the corresponding dynamics component given the content cue.

4.2.6 *Estimating higher-order transition dynamics with a three-level model*

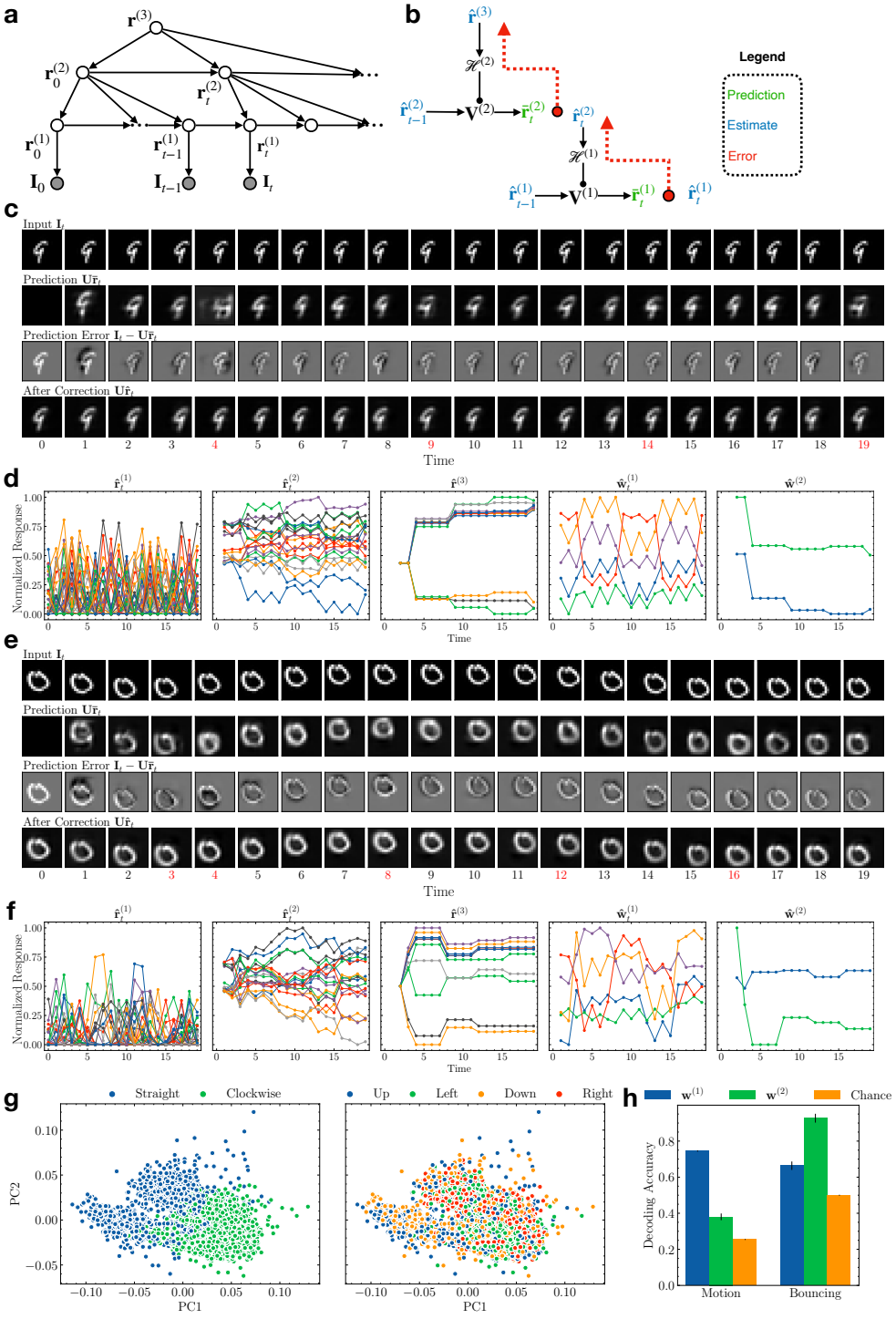


Figure 4.6: Caption continued at next page

Figure 4.6: (Previous page.) **Three-level DPC model learns progressively more abstract temporal representations.** **(a)** Generative model for three-level DPC. **(b)** Schematic depiction of an inference process. Observation nodes are omitted for clarity. **(c)** Inference for an example Moving MNIST sequence with "straight bouncing" dynamics. Red time steps mark the moments when the first-level prediction error exceeded the threshold, causing the network to transition to a new second-level state (see Methods). For these time steps, the predictions (second row) are by the second-level neurons, while the rest are by the first-level neurons as in Fig 4.3. **(d)** The network's responses to the Moving MNIST sequence in (c). Left to right: first-level responses, second-level responses, third-level responses, first-level modulation weights, second-level modulation weights. **(e)** Same as (d) but with "clockwise bouncing" dynamics. **(f)** Same as (d) but for the sequence in (e). **(g)** Third-level responses to the Moving MNIST sequences visualized in the 2D space of the first two principal components. Left: responses colored according to bouncing type; right: responses colored according to motion direction. **(h)** Comparison of decoding performance for bouncing type versus motion direction using the modulation weights generated by the second and third level. Error bars: ± 1 standard deviation from 10-fold cross validation. Orange: chance accuracies. (figure from Jiang and Rao, 2024)

Our results thus far involved a two-level DPC model whose second-level states predicted the first-level state transitions. Since the second-level state is assumed to characterize the *entire* sequence (Fig 4.1), it cannot predict higher-order transitions such as digits bouncing at the boundaries in the Moving MNIST dataset, which requires different second-level state representations (Fig 4.3). This section shows that adding a third level allows the DPC model to learn and infer the *transition dynamics of second-level states*, thus capturing a temporally more abstract representation of sequences at the third level.

Fig 4.6a shows the generative model for the three-level DPC model. Just as the second-level states modulate the transition function of the first-level states in the two-level model (Fig 4.1), the third-level states modulate the transitions of the second-level states. During

inference (Fig 4.6b), when the first-level prediction error is larger than a threshold (see Methods), the second-level state transitions to the next state, following the transition function predicted by the current third-level state. The second-level prediction error is conveyed to the third level to correct its state estimate, in the same way as the first-level prediction error corrects the second-level state.

The Moving MNIST dataset we used for Fig 4.3 exhibited only one type of transition dynamics of the second-level states, namely, transitioning from moving left to moving right, or moving up to moving down, and vice versa (henceforth referred to as "straight bouncing" dynamics; Fig 4.6c). To demonstrate that the third-level states can learn different second-level dynamics, we added digit sequences with "clockwise bouncing" dynamics to the dataset (for example, a digit moving to the left and hitting the boundary will move upward instead of rightward, and so on; Fig 4.6e). This makes the second-level state transition function ambiguous until the first bouncing event. If the third-level representations learned by DPC capture the second-level transition dynamics, we expect the first large prediction error at the second level (occurring at a boundary) to update the third-level state estimate to represent either straight bouncing dynamics or clockwise bouncing dynamics. Thereafter, the third-level state estimate should remain stable as long as the bouncing type remains the same.

In the following, we use the superscript (i) to denote the level i. We trained a three-level neural network on the augmented Moving MNIST dataset (with the two types of bouncing dynamics discussed above). The network uses two second-level transition matrices $\{\mathbf{V}_1^{(2)}, \mathbf{V}_2^{(2)}\}$ and a top-down network $\mathcal{H}^{(2)}$ (from the third to the second level), in addition to all the parameters in the two-level model. The first and second-level transition matrices were pretrained (see Methods). Fig 4.6c and Fig 4.6d show an inference example of the three-layer network on an input sequence with straight bouncing dynamics. Red time steps denote the moments when the first-level prediction errors were larger than the set threshold, causing the second-level neurons to change their activities to transition to the next state (this can be seen as a neural implementation of terminal states in a hierarchical HMM (Fine et al., 1998) (see Methods)). As seen in the second row in Fig 4.6c, at the first bouncing event ($t = 4$), the second-level prediction was not accurate; the third-level neural responses were updated to minimize this prediction error (see Fig 4.6d). For the rest of the sequence,

the predictions are accurate at the bouncing events ($t = 9, 14, 19$) and the third-level neural responses remained stable. The panels in Fig 4.6d show an increase in response stability and timescale from the first to the third-level neural responses (first three panels), as well as in the modulation weights that define the first and second-level transition dynamics (last two panels). Fig 4.6e and Fig 4.6f show a different example with clockwise bouncing dynamics. Similar to the example above, the third-level responses showed notable changes at times $t = 3$ and 4 but remained stable for the rest of the sequence. Comparing the second-level modulation weights in Fig 4.6d and Fig 4.6f, it is clear that the third-level DPC neurons estimated different bouncing types and generated opposite modulation strengths for the two types of sequences.

We performed PCA on the third-level responses $\mathbf{r}^{(3)}$ obtained at the end of the Moving MNIST sequences ($t = 19$) in the test set. Fig 4.6g visualizes these responses in the space of their first two principal components (PCs), colored either by bouncing dynamics type (left) or moving direction (right). The third-level responses form clusters according to the bouncing dynamics type but not motion direction. We then used SVMs to decode the bouncing dynamics type or motion direction from $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$, the weights predicted by $\mathbf{r}^{(2)}$ and $\mathbf{r}^{(3)}$ respectively. As shown in Fig 4.6h, using $\mathbf{w}^{(2)}$, the classifier yielded 92.7% 10-fold cross-validated classification accuracy on the two bouncing types (chance accuracy: 50.0%). Using $\mathbf{w}^{(1)}$ resulted in significantly less classification accuracy for bouncing type (66.4%, $p \ll 0.001$, t-test). In contrast, decoding accuracy for the four motion directions (chance accuracy: 25.4%) was significantly higher using $\mathbf{w}^{(1)}$ (74.5%) compared to using $\mathbf{w}^{(2)}$ (38.0%, $p \ll 0.001$, t-test). These results show that the three-level DPC model succeeded in learning a temporal hierarchy, with the third-level states encoding the longest timescale feature, *i.e.* the type of bouncing dynamics, by modulating the transition function of the second-level states, which in turn encoded intermediate timescale features (motion direction).

4.3 Methods

4.3.1 Hierarchical generative model

We assume that the observation $\mathbf{I}_t \in \mathbb{R}^M$ at time t is generated by a lower-level latent variable $\mathbf{r}_t \in \mathbb{R}^N$. The latent variable $\mathbf{r}_t$ is generated by the previous step latent variable $\mathbf{r}_{t-1}$ and the higher-level latent variable, $\mathbf{r}^h \in \mathbb{R}^{N_h}$. Together, the generative model factorizes as follows:

$$p(\mathbf{I}_{0:T-1}, \mathbf{r}_{0:T-1}, \mathbf{r}^h) = p(\mathbf{r}^h)p(\mathbf{r}_0) \prod_{t=0}^{T-1} p(\mathbf{I}_t \mid \mathbf{r}_t) \prod_{t=1}^{T-1} p(\mathbf{r}_t \mid \mathbf{r}_{t-1}, \mathbf{r}^h). \tag{4.3}$$

Each component of the factorization is parameterized as follows:

$$\mathbf{r}^h \sim \mathcal{N}(0, 1) \tag{4.4}$$

$$\mathbf{r}_t \mid (\mathbf{r}_{t-1}, \mathbf{r}^h) \sim \mathcal{N}(\mathbf{f}(\mathbf{r}^h, \mathbf{r}_{t-1}), \sigma_r^2 I) \tag{4.5}$$

$$\mathbf{I}_t \mid \mathbf{r}_t \sim \mathcal{N}(\mathbf{U}\mathbf{r}_t, \sigma^2 I), \tag{4.6}$$

where $\mathcal{N}$ denotes the normal distribution and I denotes the identity matrix. The mean $\mathbf{r}_t^\mu = \mathbf{f}(\mathbf{r}^h, \mathbf{r}_{t-1})$ is given by:

$$\mathbf{w} = \mathcal{H}_\theta(\mathbf{r}^h) \tag{4.7}$$

$$\mathbf{V} = \sum_{k=1}^{K} w_k \mathbf{V}_k \tag{4.8}$$

$$\mathbf{r}_t^\mu = \mathrm{ReLU}(\mathbf{V}\mathbf{r}_{t-1}). \tag{4.9}$$

Here, $\mathcal{H}_\theta$ is a function (neural network) parameterized by θ.

To sum up, the trainable parameters of the model include spatial filters $\mathbf{U}$, K transition matrices $\mathbf{V}_1, \ldots, \mathbf{V}_K$, and the neural network parameters θ. The latent variables are $\mathbf{r}_{0:T-1}$ and $\mathbf{r}^h$. See "Summary of the DPC generative model" in the supporting information for this chapter for a more detailed description of the model architecture.

4.3.2 Prediction error minimization

Here, we derive the loss function used for inference and learning under the assumed generative model. We focus on finding the *maximum a posteriori* (MAP) estimates of the latent

variables using a Bayesian filtering approach. At time t, the posterior of $\mathbf{r}_t$ conditioned on the input observations up to time t, $\mathbf{I}_{0:t}$, and the higher-level variable $\mathbf{r}^h$ can be written as follows using Bayes' theorem:

$$p(\mathbf{r}_t \mid \mathbf{I}_{0:t}, \mathbf{r}^h) \propto p(\mathbf{I}_t \mid \mathbf{r}_t) p(\mathbf{r}_t \mid \mathbf{I}_{0:t-1}, \mathbf{r}^h), \tag{4.10}$$

where the first term on the right-hand side is the likelihood function defined by Eq 4.6. The second term is the posterior of $\mathbf{r}_t$ given input up to the previous step and the higher-level state:

$$p(\mathbf{r}_t \mid \mathbf{I}_{0:t-1}, \mathbf{r}^h) = \int p(\mathbf{r}_t \mid \mathbf{r}_{t-1}, \mathbf{r}^h) p(\mathbf{r}_{t-1} \mid \mathbf{I}_{0:t-1}, \mathbf{r}^h) \mathrm{d}\mathbf{r}_{t-1}, \tag{4.11}$$

where the first term inside the integral is the lower-level transition dynamics defined by Eq 4.5. Note that the parameterization of the transition distribution is generated by the higher-level latent variable as specified by Eqs 4.7 to 4.9.

Putting Eq 4.10 and Eq 4.11 together, we get

$$p(\mathbf{r}_t \mid \mathbf{I}_{0:t}, \mathbf{r}^h) \propto p(\mathbf{I}_t \mid \mathbf{r}_t) \int p(\mathbf{r}_t \mid \mathbf{r}_{t-1}, \mathbf{r}^h) p(\mathbf{r}_{t-1} \mid \mathbf{I}_{0:t-1}, \mathbf{r}^h) \mathrm{d}\mathbf{r}_{t-1}, \tag{4.12}$$

which defines a recursive way to infer the posterior of $\mathbf{r}_t$ at time t. In this model, we only maintain a single point (MAP) estimate of the posterior at each time step, so we simplify the posterior distribution $p(\mathbf{r}_{t-1} \mid \mathbf{I}_{0:t-1}, \mathbf{r}^h)$ as a Dirac delta function:

$$p(\mathbf{r}_{t-1} \mid \mathbf{I}_{0:t-1}, \mathbf{r}^h) \approx \delta(\mathbf{r}_{t-1} - \hat{\mathbf{r}}_{t-1}), \tag{4.13}$$

where $\hat{\mathbf{r}}_{t-1}$ is the MAP estimate from the previous step. Now we can further simplify Eq 4.12 as

$$p(\mathbf{r}_t \mid \mathbf{I}_{0:t}, \mathbf{r}^h) \propto p(\mathbf{I}_t \mid \mathbf{r}_t) p(\mathbf{r}_t \mid \hat{\mathbf{r}}_{t-1}, \mathbf{r}^h). \tag{4.14}$$

This gives the posterior of all the latent variables at time t as

$$p(\mathbf{r}_t, \mathbf{r}^h \mid \mathbf{I}_{0:t}) \propto p(\mathbf{I}_t \mid \mathbf{r}_t) p(\mathbf{r}_t \mid \hat{\mathbf{r}}_{t-1}, \mathbf{r}^h) p(\mathbf{r}^h \mid \mathbf{I}_{0:t}). \tag{4.15}$$

We can find the MAP estimates of the latent variables by minimizing the negative log of Eq 4.15. Substituting the generative assumptions (Eqs 4.4 to 4.6), we get:

$$\bar{\mathbf{r}}_t = \mathbf{f}(\mathbf{r}^h, \hat{\mathbf{r}}_{t-1}) \tag{4.16}$$

$$\mathcal{L}_t = \frac{1}{2\sigma^2} \|\mathbf{I}_t - \mathbf{U}\mathbf{r}_t\|_2^2 + \frac{1}{2\sigma_r^2} \|\mathbf{r}_t - \bar{\mathbf{r}}_t\|_2^2 + \lambda \|\mathbf{r}_t\|_1 + \lambda_h \|\mathbf{r}^h\|_2^2, \tag{4.17}$$

where λ and λ_h are the sparsity penalty for $\mathbf{r}_t$ and the Gaussian prior penalty for $\mathbf{r}^h$, respectively. Note that we approximate $p(\mathbf{r}^h \mid \mathbf{I}_{0:t})$ with the unconditional prior $p(\mathbf{r}^h)$ so that at each step the dynamics are estimated using only the local pairwise transition and the prior. Using Eq 4.17, we compute the MAP estimate of $\mathbf{r}_t$ at time t as

$$\hat{\mathbf{r}}_t = \arg\min_{\mathbf{r}_t} \mathcal{L}_t. \tag{4.18}$$

At each time step, we update the current estimate of $\mathbf{r}^h$ to minimize $\mathcal{L}_t$ as well:

$$\hat{\mathbf{r}}^h = \arg\min_{\mathbf{r}^h} \mathcal{L}_t. \tag{4.19}$$

To begin the recursive estimation (without the temporal prediction from the previous step), we compute the MAP estimate of the first step latent variable $\mathbf{r}_0$ using the following reduced loss

$$\mathcal{L}_0 = \frac{1}{2\sigma^2}\|\mathbf{I}_0 - \mathbf{U}\mathbf{r}_0\|_2^2 + \lambda\|\mathbf{r}_0\|_1 \tag{4.20}$$

$$\hat{\mathbf{r}}_0 = \arg\min_{\mathbf{r}_0} \mathcal{L}_0. \tag{4.21}$$

The parameters of the model can be optimized by minimizing the same prediction errors summed across time and averaged across different sequences, using the MAP estimates of the latent variables. See Inference & learning process in the supporting information for this chapter for detailed pseudocode describing the inference and learning procedure.

4.3.3 Data and preprocessing

For the natural video dataset, we extracted 65520 image sequences from a YouTube video (link here) recorded by a person walking on a forest trail (image size: 16×16 pixels, sequence length: 10 frames (≈ 0.35 seconds, uniformly sampled in time). The image sequences do not overlap with each other spatially or temporally. Each sequence was spatially and temporally whitened to simulate retinal and LGN processing following the methods in Olshausen & Field (1996) and Dong & Atick (1995b). 58,968 sequences were used to train the model and the remaining 6,552 were reserved for testing.

For the Moving MNIST dataset (Srivastava et al., 2015), we used 10000 image sequences (image size: 18×18 pixels, sequence length: 10 frames), each sequence containing a fixed

digit moving in a particular direction. The motion of the digits was restricted to upward, downward, leftward, or rightward directions. When a digit hit the boundary, its motion direction was inverted (leftward to rightward, upward to downward, and vice versa). No whitening procedures were performed on the MNIST sequences. 9,000 sequences were used to train the model and the remaining 1,000 were reserved for testing.

4.3.4 Reverse correlation for computing space-time receptive fields

The reverse correlation stimuli for deriving the space-time receptive fields (Fig 4.2) were extracted from the same natural video data but without any spatial or temporal overlap with the training and test set. We used 50 continuous image sequences with 80,000 steps ($\approx$ 47 minutes, spanning the same time range but with no spatial overlap) and computed the space-time receptive fields (STRFs) as the firing-rate-weighted average of input frame sequences of length seven ($\approx$ 250 ms), across time and sequences.

Formally, let $\left\{\mathbf{I}^{(j)}_{0:T-1}\right\}_{j=1}^{J}$ be the J stimulus sequences of length T and let τ be the length of the STRFs (here, $J = 50, T = 80,000, \tau = 7$). For each neuron i, its space-time receptive field STRF_i has dimensions $M \times \tau$, where M is the dimensionality of a single image frame vector (here, $M = 100$ after vectorizing the 10×10 image frame). We compute the STRF of neuron i as follows

$$\text{STRF}_i = \frac{1}{J(T-\tau)} \sum_{j=1}^{J} \sum_{t=\tau}^{T-1} \hat{r}^{(j)}_{t,i} \mathbf{I}^{(j)}_{t-\tau:t-1}, \tag{4.22}$$

where $\hat{r}^{(j)}_{t,i}$ is the predicted firing rate of neuron i at time t in sequence j and $\mathbf{I}^{(j)}_{t-\tau:t-1}$ is the image sequence from time $t-\tau$ to $t-1$ in sequence j. This procedure is analogous to the calculation of the spike-triggered average widely used in neurophysiology (Ringach and Shapley, 2004); in this case, we computed the average of input sequences weighted by the activity $\hat{r}$ caused by the sequence.

4.3.5 Autocorrelation for quantifying timescales

Since our model responds deterministically to the same sequence (MAP estimation), we cannot follow the exact approach of Murray et al. (2014) that relies on across-trial variability

to the same stimulus. We computed the autocorrelation of single neuron responses to natural videos and Gaussian white noise sequences. We averaged the single-neuron autocorrelations across the lower or higher level, and trials. Formally, let $r_{t,i,j}$ be the response of neuron i at time t during trial j. We computed the autocorrelation with lag k as follows:

$$\mu_{i,j} = \frac{1}{T}\sum_{t=0}^{T-1} r_{t,i,j} \tag{4.23}$$

$$\sigma_{i,j} = \sqrt{\frac{1}{T}\sum_{t=0}^{T-1}(r_{t,i,j} - \mu_{i,j})^2} \tag{4.24}$$

$$\rho_{i,j}(k) = \frac{\sum_{t=0}^{T-k-1}(r_{t,i,j} - \mu_{i,j})(r_{t+k,i,j} - \mu_{i,j})}{(T-k)\sigma_{i,j}\sigma_{i,j}} \quad \forall i = 1 \dots N. \tag{4.25}$$

To compute the autocorrelation for an entire population at lag k, we took the average of the autocorrelations across all N neurons and J trials:

$$\rho(k) = \frac{1}{NJ}\sum_{i=1}^{N}\sum_{j=1}^{J}\rho_{i,j}(k). \tag{4.26}$$

For the results shown in Fig 4.3b and Fig 4.3c, we chose $J = 500$, $T = 50$ and computed the autocorrelation with lag $k = 0 \dots 9$ for the lower- and higher-level neurons. White noise pixels were i.i.d. samples from $\mathcal{N}(0, 0.0075)$. We also computed the autocorrelation for both levels with natural videos selected from the same stimuli used for the reverse correlation analysis (note though with natural videos the stationary mean and variance assumption is less valid). To quantify the timescale of the autocorrelation decay, we fitted an exponential decay function $\rho_{\text{fit}}(k) = a\exp(-k/\tau) + b$ to the autocorrelation data on each level (through Scipy `optimize.curve_fit` function), where a, b, and τ are fitted parameters of the function and τ represents the response timescale following the definition by Murray et al. (2014).

4.3.6 Flash-lag and postdiction simulation

We extracted five-step sequences that have a consistent leftward or rightward motion from the Moving MNIST test set sequences (134 sequences in total, see Fig 4.5e for an example). To simulate the test conditions used by Eagleman & Sejnowski (2000), we either used the first three steps of the sequences to infer a motion (dynamics) estimate $\hat{\mathbf{r}}^h$ (conditions with initial trajectories), or initialized $\hat{\mathbf{r}}^h$ to zero vectors (conditions without initial trajectories).

Depending on the test condition, the moving object stimulus at $t = 3$ could move following the original trajectory ("Continuous"), remain at the same location ("Stopped"), move in the reversed trajectory ("Reversed"), or disappear, shown as an empty frame ("Terminated"), as shown in Fig 4.4a. The stimuli for simulating flashes correspond to the "no initial trajectory, terminate case".

The model's percept of either the moving object or the flashed object at $t = 3$ was computed as the top-down spatiotemporal prediction of the image after correcting the prediction error at $t = 3$:

$$\bar{\mathbf{w}} = \mathcal{H}_\theta(\hat{\mathbf{r}}^h) \tag{4.27}$$

$$\bar{\mathbf{V}} = \sum_{k=1}^{K} \bar{w}_k \mathbf{V}_k \tag{4.28}$$

$$\bar{\mathbf{I}}_3 = \mathbf{U}\left(\mathrm{ReLU}\left(\bar{\mathbf{V}}\hat{\mathbf{r}}_2\right)\right). \tag{4.29}$$

Here, $\mathcal{H}_\theta$ and $\mathbf{V}_1, \ldots, \mathbf{V}_K$ are the parameters defined in Eqs 4.7 and 4.8, $\hat{\mathbf{r}}^h$ is the optimal higher-level estimate at $t = 3$ (Eq 4.19), and $\hat{\mathbf{r}}_2$ is the optimal lower-level estimate at $t = 2$. We computed the location of the percept as the center of mass of the percept image $\bar{\mathbf{I}}_3$. The displacement in percept between the moving object and the flashed object was calculated as

$$\text{Displacement} = \begin{cases} \text{CoM-x}\left(\bar{\mathbf{I}}_3^{\text{moving}}\right) - \text{CoM-x}\left(\bar{\mathbf{I}}_3^{\text{flash}}\right), & \text{rightward motion,} \\ \text{CoM-x}\left(\bar{\mathbf{I}}_3^{\text{flash}}\right) - \text{CoM-x}\left(\bar{\mathbf{I}}_3^{\text{moving}}\right), & \text{leftward motion} \end{cases} \tag{4.30}$$

where CoM-x($\mathbf{I}$) returns the horizontal location (x dimension) of the center of mass of $\mathbf{I}$. Therefore, a positive displacement is along the original trajectory of the moving object, while a negative displacement is along the reversed trajectory.

To compute the plots shown in Fig 4.4g and Fig 4.4h, we used the "with initial trajectory, reversal" test condition. The displacement was computed as in Eq 4.30 but $\bar{\mathbf{I}}_3^{\text{flash}}$ was replaced by the input image $\mathbf{I}_2$ at $t = 2$, at every value of $\mathbf{r}^h$ through the error correction process at $t = 3$ (Eqs 4.27 to 4.30, see Fig 4.4b Perceived versus Input).

4.3.7 Sequence learning and recall simulation

To simulate the sequence learning experiment of Xu et al. (2012), we used a five-step sequence extracted from a Moving MNIST test set sequence (Fig 4.5e). We augmented the hierarchical generative model of DPC with an associative memory layer $\mathbf{m}$, which implements predictive coding of the joint higher-level state $\mathbf{r}^h$ and the lower-level state $\mathbf{r}_0$ through synapses $\mathbf{G}$ (Rao and Ballard, 1999; Salvatori et al., 2021) (see "Summary of the memory model" in the supporting information for this chapter for model details):

$$(\mathbf{r}_0, \mathbf{r}^h) \mid \mathbf{m} \sim \mathcal{N}(\mathbf{G}\mathbf{m}, \sigma_m^2 I). \tag{4.31}$$

The memory layer was trained separately (the DPC network weights were fixed during conditioning and recall) by minimizing the prediction error:

$$\mathcal{L}_{\text{memory}}(\mathbf{G}, \mathbf{m}) = \|\mathbf{s} - \mathbf{G}\mathbf{m}\|_2^2 + \lambda_m \|\mathbf{m}\|_2^2, \tag{4.32}$$

where $\mathbf{s} = [\hat{\mathbf{r}}_0; \hat{\mathbf{r}}^h]$ is the concatenated lower- and higher-level state estimates from DPC and λ_m is the regularization penalty on $\mathbf{m}$.

During memory encoding, $\hat{\mathbf{r}}_0$ and $\hat{\mathbf{r}}^h$ were estimated by the two-level DPC network from the sequence shown in Fig 4.5e. Then the memory vector $\mathbf{m}$ was estimated by gradient descent on Eq 4.32, yielding the optimal estimate $\hat{\mathbf{m}}$:

$$\hat{\mathbf{m}} = \arg\min_{\mathbf{m}} \mathcal{L}_{\text{memory}}(\mathbf{G}, \mathbf{m}). \tag{4.33}$$

The remaining prediction error drives rapid synaptic plasticity in $\mathbf{G}$ through gradient descent on the same equation (Fig 4.5c):

$$\mathbf{G}' \leftarrow \mathbf{G} - \eta_G \frac{\partial \mathcal{L}_{\text{memory}}(\mathbf{G}, \hat{\mathbf{m}})}{\partial \mathbf{G}}, \tag{4.34}$$

where η_G is the learning rate for $\mathbf{G}$. During conditioning, we updated the synaptic weights five times using Eq 4.34. During recall, given the cue for the beginning of the sequence, the full memory vector $\mathbf{m}$ is retrieved by minimizing the prediction error with respect to the initial lower-level state $\mathbf{r}_0$ portion of the stored memory only (Salvatori et al., 2021) (Fig 4.5d):

$$\tilde{\mathbf{m}} = \arg\min_{\mathbf{m}} \|\tilde{\mathbf{s}} - \mathbf{b} \odot (\mathbf{G}\mathbf{m})\|_2^2 + \lambda_m \|\mathbf{m}\|_2^2, \tag{4.35}$$

where $\tilde{\mathbf{s}} = [\hat{\mathbf{r}}_0; \mathbf{0}^{N_h}]$ denotes the partial input (visual cue representing the first element of the sequence), $\odot$ denotes element-wise multiplication, and $\mathbf{b} \in \{0,1\}^{N+N_h}$ is a binary mask:

$$b_i = \begin{cases} 1 & \text{if } i \leq N \\ 0 & \text{otherwise} \end{cases}. \tag{4.36}$$

The rest of the sequence was obtained by retrieving the stored higher-level state $\bar{\mathbf{r}}^h$ (the dynamics of the sequence) as the top-down prediction through $\tilde{\mathbf{m}}$ (Fig 4.5d). Once the dynamics were retrieved, we tested sequential recall in the network by predicting an entire five-step sequence using the lower-level vector $\hat{\mathbf{r}}_0$ from the visual cue and the retrieved dynamics vector $\bar{\mathbf{r}}^h$ from the memory (Eqs 4.7 to 4.9). We tested recall in the network using three different visual cues: the starting frame ($t = 0$), the middle frame ($t = 2$), or the end frame ($t = 4$) (see Fig 4.5e and Fig 4.5f). The cross correlation plot was computed following the same procedure as the one described in Xu et al. (2012) (Fig 4.5g and Fig 4.5h).

4.3.8 Three-level DPC model

To make the level notation clear, we denote the first and second-level states $\mathbf{r}$ and $\mathbf{r}^h$ from the two-level model as $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$ in the three-level model, and denote the highest (third-level) state as $\mathbf{r}^{(3)}$. We use superscripts to denote level and subscripts to denote time, unless noted otherwise. The trainable parameters for the three-level model include those for the two-level model as well as two second-level transition matrices $\{\mathbf{V}_1^{(2)}, \mathbf{V}_2^{(2)}\}$ and the third-level top-down network $\mathcal{H}_\theta^{(2)}$ that generates the second-level modulation weights.

Pretraining the second-level transition matrices

Before training the three-level network, we first pretrained two separate two-level DPC networks, each with a single transition matrix $\mathbf{V}^{(2)}$ on Moving MNIST sequences with either straight bouncing type or clockwise bouncing type. We performed inference on the second-level states with the following loss function:

$$\mathcal{L}_t = \frac{1}{2\sigma^2}\|\mathbf{I}_t - \mathbf{U}\mathbf{r}_t^{(1)}\|_2^2 + \frac{1}{2\sigma_r^2}\|\mathbf{r}_t^{(1)} - \bar{\mathbf{r}}_t^{(1)}\|_2^2 + b_t\left(\frac{1}{2\sigma_{r^{(2)}}^2}\|\mathbf{r}_t^{(2)} - \bar{\mathbf{r}}_t^{(2)}\|_2^2\right) + \lambda\|\mathbf{r}_t\|_1, \tag{4.37}$$

where $\sigma^2_{\mathbf{r}^{(2)}}$ is the variance of second-level prediction errors, $b_t \in \{0, 1\}$ is a binary mask that equals 1 if the first-level prediction error is larger than a threshold estimated from the training set and 0 otherwise (Prediction error threshold robustly finds changes of dynamics in the supporting information for this chapter), and $\bar{\mathbf{r}}_t^{(2)} = \mathbf{V}^{(2)}\mathbf{r}_{t-1}^{(2)}$ is the predicted second-level state. We learned $\mathbf{V}^{(2)}$ by gradient descent on the same loss as in Eq 4.37, summed across time and averaged across sequences, using the MAP estimates of the latent variables.

Three-level model training

We inferred the second- and third-level states using the same loss function (Eq 4.37), with the predicted second-level state $\bar{\mathbf{r}}_t^{(2)}$ defined as

$$\mathbf{w}^{(2)} = \mathcal{H}_\theta^{(2)}(\mathbf{r}^{(3)}) \tag{4.38}$$

$$\mathbf{V}^{(2)} = \sum_{k=1}^{K^{(2)}} w_k^{(2)} \mathbf{V}_k^{(2)} \tag{4.39}$$

$$\bar{\mathbf{r}}_t^{(2)} = \mathbf{V}^{(2)}\mathbf{r}_{t-1}^{(2)}, \tag{4.40}$$

where $K^{(2)} = 2$ is the number of second-level transition matrices. Comparing this definition with Eqs 4.7 to 4.9, it is easy to see that the third-level top-down prediction is recursively defined in the same way as the top-down prediction in the two-level model. After obtaining the MAP estimates of the second- and third-level states, we learned the third-level top-down network $\mathcal{H}_\theta^{(2)}$ by gradient descent on the same loss. See Inference & learning process for the three-level DPC model in the supporting information for this chapter for detailed pseudocode describing the inference and learning procedure for the three-level model.

4.4 Discussion

Our results show that dynamic predictive coding (DPC) can learn hierarchical temporal representations of sequences through top-down modulation of lower-level dynamics. Specifically, we showed that by minimizing prediction errors on image sequences, a two-level DPC neural network develops V1-like separable and inseparable space-time receptive fields at the lower level (DeAngelis et al., 1995), and representations encoding sequences at a longer timescale at the higher level (Murray et al., 2014; Siegle et al., 2021). The trained

DPC network provides a normative explanation for the flash-lag effect (Eagleman and Sejnowski, 2000) and accounts for both prediction and postdiction in visual motion processing (Hogendoorn et al., 2008; Shimojo, 2014; Hogendoorn, 2020). The temporal abstraction of sequences in a DPC network suggests a new mechanism for storing and retrieving episodic memories by linking the DPC network to an associative memory, emulating the interaction between the neocortex and the hippocampus. We show that such a memory-augmented DPC model explains cue-triggered activity recall in the visual cortex (Xu et al., 2012). Finally, we show that the top level of a three-level DPC network captures the higher-order temporal statistics encoding the transition dynamics of the second-level states, which in turn capture the temporal statistics of the first-level states. Taken together, the hierarchical temporal representations learned by DPC, ranging from the lowest-level space-time representations similar to those observed in visual cortical simple cells (Fig 4.2), through the intermediate-level representations of steady motion (Fig 4.3), to the highest-level representation of how such motion changes over a longer timescale (Fig 4.6), emulate the spatiotemporal representations observed in visual cortical hierarchies, particularly along the dorsal visual pathway (Mishkin et al., 1983).

4.5 Supporting information

Hypernetworks and neural gain modulation

A hypernetwork $\mathcal{H}_\phi$ is a neural network parameterized by synaptic weights and bias parameters ϕ that generates the parameters (weights and bias) of another "primary" neural network $\mathbf{f}$ (Ha et al., 2017):

$$\theta = \mathcal{H}_\phi(\mathbf{z}) \tag{4.41}$$

$$\mathbf{y} = \mathbf{f}_\theta(\mathbf{x}), \tag{4.42}$$

where $\mathbf{z}$ and $\mathbf{x}$ are the inputs to the hypernetwork and the primary network, respectively, and $\mathbf{y}$ is the output of the primary network. Therefore, the responses of the primary network to the same input $\mathbf{x}$ can be different, depending on the parameters θ generated by the hypernet. The input $\mathbf{z}$ to the hypernet can be a top-down input, a task input or context, an external

input, or even past inputs $\mathbf{x}$. The hypernet model thus provides a general framework for one neural network (hypernet) modulating the function being computed by another neural network (primary net).

A special case of the hypernet model corresponds to neural gain modulation – a phenomenon where external or internal drives modulate the input-output (I/O) relationship of a neuron (Ferguson and Cardin, 2020; Shine et al., 2021). Typically, gain modulation is implemented through multiplicative (altering the slope of the I/O relationship) or additive (shifting the I/O relationship) mechanisms on either the input or the response (Larkum et al., 2004; Silver, 2010; Ferguson and Cardin, 2020). These multiplicative and additive components can be seen as the output of the hypernetwork fed to a primary network with fixed parameters θ. For example, the recurrent network model proposed by Stroud et al. (Eq 1 in Stroud et al., 2018) can be written as (following our notation)

$$\tau \frac{\mathrm{d}\mathbf{r}_t}{\mathrm{d}t} = -\mathbf{r}_t + \mathbf{V}\mathbf{f}(\mathbf{r}_t; \mathbf{e}), \tag{4.43}$$

where τ is the time constant, $\mathbf{r}_t$ is the neural activity vector at time t, $\mathbf{V}$ is the recurrent weight matrix, and $\mathbf{e}$ is the gain modulation that works as a pointwise multiplicative factor on the input ($f_i(r_i; e_i) \approx \tanh(e_i r_i)$, see Equation 2 in Ref. Stroud et al., 2018). Here, $\mathbf{e}$ can be seen as the output of a hypernetwork $\mathcal{H}_\phi$ that modulates the recurrent network with weights $\theta = \mathbf{V}$.

Similarly, the DPC generative model for hierarchical temporal prediction through top-down modulation of recurrent connections (Eqs 4.7 to 4.9) can be implemented using the recurrent network model:

$$\mathbf{w} = \mathcal{H}_\phi(\mathbf{r}^h) \tag{4.44}$$

$$\tau \frac{\mathrm{d}\mathbf{r}_t}{\mathrm{d}t} = -\mathbf{r}_t + \sum_k w_k \mathbf{V}_k \,\mathrm{ReLU}(\mathbf{r}_t) \tag{4.45}$$

$$= -\mathbf{r}_t + \sum_k \mathbf{V}_k \mathbf{f}_k(\mathbf{r}_t; w_k). \tag{4.46}$$

where $\mathbf{f}_k(\mathbf{r}_t; w_k) = w_k \,\mathrm{ReLU}(\mathbf{r}_t)$ is the input modulation function which modulates the inputs $\mathrm{ReLU}(\mathbf{r}_t)$ to group k of neurons multiplicatively with the weight w_k. Therefore, comparing Eq 4.43 with Eq 4.46, the gain modulation model of Stroud et al. (2018) uses

per-neuron multiplicative gain modulation while DPC generates a *more diffuse* top-down multiplicative gain modulation targeting *groups of lower-level neurons*, with the same top-down modulation strength w_k for group k. Such a model for top-down modulation is consistent with previous work suggesting that cortical feedback connections are more diffuse than feedforward connections (Perkel et al., 1986; Salin and Bullier, 1995; Rockland and Ojima, 2003) (but see also (Markov et al., 2014)).

Summary of the DPC generative model

Table 4.1 summarizes the parameters and values used in the DPC generative model. The top-down dynamics prediction network $\mathcal{H}$ (a hypernetwork) is a multi-layer perceptron (MLP) as follows. The first layer of the MLP contains 10 hidden neurons, followed by a LayerNorm layer (Ba et al., 2016) and an ELU activation function (Clevert et al., 2016). The last two layers of the MLP are composed of (1) a linear layer with 10 hidden neurons, and (2) an output layer that maps this hidden activity to $\mathbf{w} \in \mathbb{R}^K$.

Parameter	Description	Value (Natural/MNIST)
$\mathbf{r}_t \in \mathbb{R}^N \quad \forall t = 0 \dots T-1$	Lower-level latent variable from time 0 to $T-1$	$N = 512/648$
$\mathbf{r}^h \in \mathbb{R}^{N_h}$	Higher-level latent variable	$N_h = 20/20$
$\mathbf{I}_t \in \mathbb{R}^M \quad \forall t = 0 \dots T-1$	Input observations from time 0 to $T-1$	$M = 256/324$
$\mathbf{U} \in \mathbb{R}^{M \times N}$	Learnable spatial filters	Learned
$\sigma^2 \in \mathbb{R}$	Spatial variance	1 / 1
$\sigma_r^2 \in \mathbb{R}$	Temporal variance	0.4 / 0.4
K	Number of transition matrices	$K = 5/5$
$\mathbf{V}_k \in \mathbb{R}^{N \times N} \quad \forall k = 1 \dots K$	Transition matrices	Learned
$\mathcal{H}_\theta$	Top-down dynamics network	Learned
λ	Sparsity penalty for $\mathbf{r}_t$	$5 \times 10^{-4}/1 \times 10^{-3}$
λ_h	Prior penalty for $\mathbf{r}^h$	$5 \times 10^{-6}/5 \times 10^{-6}$

Table 4.1: **DPC generative model parameters and values.**

Inference and learning

Table 4.2 summarizes the optimizers and learning rates used for inference and learning in DPC. Algorithm 1 describes the inference (finding the MAP estimates of the latent variables) and learning (parameter estimation) process.

Optimizee / Other Hyperparameters	Optimizer (learning rate: Natural/MNIST) / Value
Inference	
$\mathbf{r}_t \quad \forall t = 0 \dots T-1$	SGD (3/2.5)
$\mathbf{r}^h$	ADAM ($1 \times 10^{-3}/1 \times 10^{-3}$)
Convergence criteria	$\|\Delta\|_2 \leq 0.01$
Learning	
$\mathbf{U}$	SGD (2.5/0.5)
$\mathbf{V}_k \in \mathbb{R}^{N \times N} \quad \forall k = 1 \dots K$	ADAM ($5 \times 10^{-4}/1 \times 10^{-3}$)
$\mathcal{H}_\theta$	ADAM ($5 \times 10^{-4}/1 \times 10^{-3}$)
Batch size	512/512
Epochs	100/120
Learning rate exponential decay	0.98 / 0.985

Table 4.2: **Optimizers and learning rates used for inference and learning in the DPC experiments**. Here Δ denotes the difference in $\mathbf{r}_t$ or $\mathbf{r}^h$ from before and after the current iteration of gradient descent.

Training results

Fig 4.7a visualizes the evolution of test set loss as a function of training epochs. Besides results using five transition matrices ($K = 5$), we additionally trained models with $K = 1, \dots, 6$, each with eight random initializations of weights. DPC with $K = 1$ is equivalent to a one-level Kalman filter model by Rao (1999) with an additional sparseness constraint

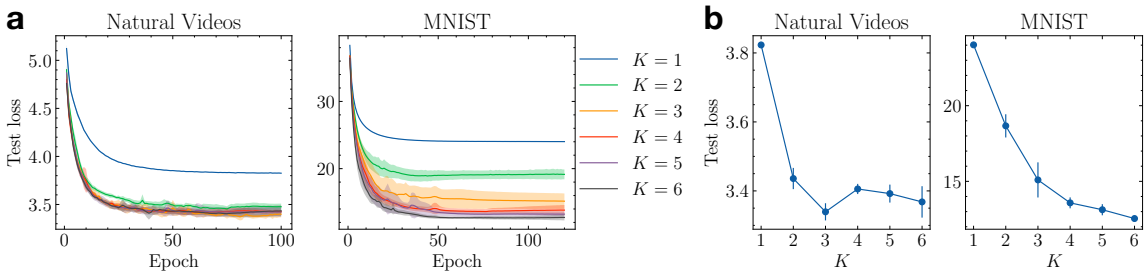


Figure 4.7: **Improvement on test set loss saturates as the number of transition matrices increases.** **(a)** Test set loss as training proceeded. Shaded area denotes ±1 standard deviation computed over eight runs with random initialization for each K. $K = 1$ shows the performance of the single-layer model. **(b)** Best test loss as K increases. Error bars denote ±1 standard deviation. (figure from Jiang and Rao, 2024)

on neural activities (Olshausen and Field, 1996). As shown in Fig 4.7, all trained models converged after around 50 training epochs. DPC models with two-level architectures ($K > 1$) performed significantly better on the test sets. Such improvement in test set performance saturated as K increased. Fig 4.7b shows test set loss as a function of K for natural videos and the moving MNIST dataset.

Summary of the memory model

Table 4.3 summarizes the parameters and values used in the associative memory model. Table 4.4 summarizes the optimizers and learning rates used for inference and learning in the memory layer.

Fig 4.8 shows examples of cue-specific sequence recall by the associative memory model after training on different sequences. The upper left quadrant shows the image sequence used in our experiment in Fig 4.5, and the image sequence recalled by the trained memory-augmented DPC network. To test that the recall is cue-specific, we additionally conditioned the network with three sequences with different digits and dynamics. As the rest of Fig 4.8 shows, the memory-augmented DPC network successfully recalled the correct dynamics associated with the different digit cues, demonstrating the cue-specificity of the network.

Algorithm 1 Inference & learning process

Input: Image sequences dataset $\mathcal{D}$

Parameters: $\mathbf{U}, \{\mathbf{V}_k\}_{k=1}^{K}, \theta$

while the parameters have not converged **do**
  Sample minibatch $\{\mathbf{I}_{0:T-1}^{(m)}\}_{m=1}^{M}$ from dataset $\mathcal{D}$
  Initialize total loss $\mathcal{L} = 0$
  `// Inference`
  for each sequence $\mathbf{I}_{0:T-1}$ in the minibatch **do**
    Initialize $\mathbf{r}_0, \mathbf{r}^h$ as zero vectors
    $\hat{\mathbf{r}}_0 = \arg\min_{\mathbf{r}_0} \mathcal{L}_0(\mathbf{r}_0, \mathbf{U}, \mathbf{I}_0)$
    Update total loss $\mathcal{L} = \mathcal{L} + \mathcal{L}_0(\hat{\mathbf{r}}_0, \mathbf{U}, \mathbf{I}_0)$
    for $t = 1 \dots T-1$ **do**
      Initialize $\mathbf{r}_t$ as a zero vector
      $\hat{\mathbf{r}}_t = \arg\min_{\mathbf{r}_t} \mathcal{L}_t(\mathbf{r}_t, \hat{\mathbf{r}}_{t-1}, \hat{\mathbf{r}}^h, \mathbf{U}, \{\mathbf{V}_k\}_{k=1}^{K}, \theta, \mathbf{I}_t)$
      $\hat{\mathbf{r}}^h = \arg\min_{\mathbf{r}^h} \mathcal{L}_t(\hat{\mathbf{r}}_t, \hat{\mathbf{r}}_{t-1}, \mathbf{r}^h, \mathbf{U}, \{\mathbf{V}_k\}_{k=1}^{K}, \theta, \mathbf{I}_t)$
      $\mathcal{L} = \mathcal{L} + \mathcal{L}_t(\hat{\mathbf{r}}_t, \hat{\mathbf{r}}_{t-1}, \hat{\mathbf{r}}^h, \mathbf{U}, \{\mathbf{V}_k\}_{k=1}^{K}, \theta, \mathbf{I}_t)$
    end for
  end for
  `// Learning`
  Update parameters using gradients calculated from $\frac{1}{M}\mathcal{L}$ and their corresponding optimizers from Table 4.2
end while

Return $\mathbf{U}, \{\mathbf{V}^{(k)}\}_{k=1}^{K}, \theta$

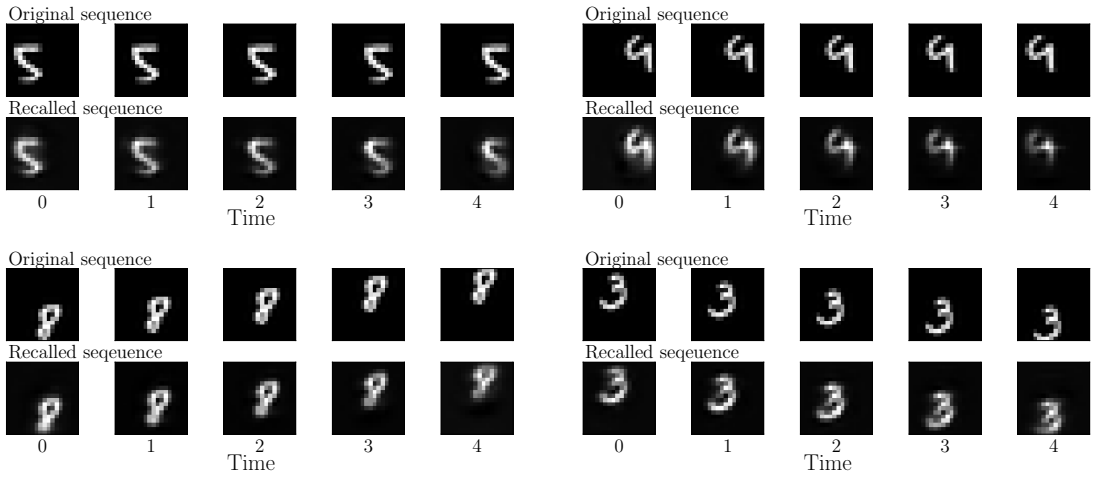


Figure 4.8: **Cue-triggered recall is cue-specific.** Four examples of cue-specific sequence recall by the associative memory model after training on different sequences, when given the first frame as the cue. In each quadrant: top: the original image sequence; bottom: cue-triggered recall of the stored sequence. (figure from Jiang and Rao, 2024)

Parameter	Description	Value
$\mathbf{m} \in \mathbb{R}^{N_m}$	memory latent variable from time 1 to T	$N_m = 256$
$\mathbf{G} \in \mathbb{R}^{(N+N_h)\times N_m}$	Memory weights	Learned
λ_m	Prior penalty for $\mathbf{m}$	0

Table 4.3: **Memory model parameters and values.**

Three-level DPC model

We first describe the procedure for computing the threshold for detecting large first-level prediction errors. Using the pretrained two-level DPC model, we collect the distribution of all first-level prediction errors $\|\hat{\mathbf{r}}_t^{(1)} - \bar{\mathbf{r}}_t^{(1)}\|_2^2$ as defined in Eq 4.37. Fig 4.9a shows the distribution of prediction errors on the training set. We set the threshold $\rho = 0.73$ where the cumulative density of the error distribution (orange curve) reaches 0.75. To verify the

Optimizee / Other Hyperparameters	Optimizer (learning rate) / Value
Inference	
m	ADAM (0.005)
Convergence criteria	$\|\Delta\|_2 \leq 0.01$
Learning	
G	ADAM (0.001)
Epochs	5
Learning rate exponential decay	0.99

Table 4.4: **Optimizers and learning rates used for inference and learning in the memory model.** Here Δ denotes the difference in **m** from before and after the current iteration of gradient descent.

efficacy of this threshold, in Fig 4.9b we show example input sequences from the test set, where the red arrows mark the time steps when the first-level prediction errors exceeded the threshold ρ. As shown, using this threshold robustly finds the moments when the input dynamics changed, in both straight bouncing sequences and clockwise bouncing sequences.

Table 4.5 describes the additional parameters and values in the three-level DPC model, in addition to those shown in Table 4.1 for the two-level model. The third-level top-down dynamics prediction network $\mathcal{H}_\theta^{(2)}$ is a multi-layer perceptron that is identical to the previously described second-level network $\mathcal{H}_\theta^{(1)}$, except the last output layer that has output dimension $K^{(2)} = 2$. Table 4.6 summarizes the optimizers and learning rates used for inference and learning in the three-level DPC experiments. Lastly, we describe the inference and learning process of the three-level model in Algorithm 2.

Algorithm 2 Inference & learning process for the three-level DPC model

Input: Image sequences dataset $\mathcal{D}$

Parameters: $\mathbf{U}, \{\mathbf{V}_k^{(1)}\}_{k=1}^{K^{(1)}}, \{\mathbf{V}_k^{(2)}\}_{k=1}^{K^{(2)}}, \theta$, ρ (the first-level error threshold)

while the parameters have not converged **do**
  Sample minibatch $\{\mathbf{I}_{0:T-1}^{(m)}\}_{m=1}^{M}$ from dataset $\mathcal{D}$
  Initialize total loss $\mathcal{L} = 0$
  `// Inference`
  for each sequence $\mathbf{I}_{0:T-1}$ in the minibatch **do**
    Initialize $\mathbf{r}_0^{(1)}, \mathbf{r}_0^{(2)}, \mathbf{r}^{(3)}$ as zero vectors
    $\hat{\mathbf{r}}_0^{(1)} = \arg\min_{\mathbf{r}_0^{(1)}} \mathcal{L}_0(\mathbf{r}_0, \mathbf{U}, \mathbf{I}_0)$ ▷ Eq 20
    Update total loss $\mathcal{L} = \mathcal{L} + \mathcal{L}_0(\hat{\mathbf{r}}_0^{(1)}, \mathbf{U}, \mathbf{I}_0)$
    $\hat{\mathbf{r}}_1^{(1)}, \hat{\mathbf{r}}_1^{(2)} = \arg\min_{\mathbf{r}_1^{(1)}, \mathbf{r}_1^{(2)}} \mathcal{L}_1(\mathbf{r}_1^{(1)}, \mathbf{r}_1^{(2)}, \hat{\mathbf{r}}_0^{(1)}, \mathbf{U},$ ▷ Eq 17
    $\{\mathbf{V}_k^{(1)}\}_{k=1}^{K^{(1)}}, \theta, \mathbf{I}_1)$
    $\mathcal{L} = \mathcal{L} + \mathcal{L}_1(\hat{\mathbf{r}}_1^{(1)}, \hat{\mathbf{r}}_1^{(2)}, \hat{\mathbf{r}}_0^{(1)}, \mathbf{U},$
    $\{\mathbf{V}_k^{(1)}\}_{k=1}^{K^{(1)}}, \theta, \mathbf{I}_1)$
    for $t = 2 \dots T-1$ **do**
      `// Same two-level inference as before`
      $\hat{\mathbf{r}}_t^{(1)}, \hat{\mathbf{r}}_t^{(2)} = \arg\min_{\mathbf{r}_t^{(1)}, \mathbf{r}_t^{(2)}} \mathcal{L}_t(\mathbf{r}_t^{(1)}, \mathbf{r}_t^{(2)}, \hat{\mathbf{r}}_{t-1}^{(1)}, \mathbf{U},$ ▷ Eq 17
      $\{\mathbf{V}_k^{(1)}\}_{k=1}^{K^{(1)}}, \theta, \mathbf{I}_t)$
      `// Compute binary mask`
      $b_t = \left(\|\hat{\mathbf{r}}_t^{(1)} - \bar{\mathbf{r}}_t^{(1)}\|_2^2 > \rho\right)$
      `// Eq 37`
      $\hat{\mathbf{r}}_t^{(2)}, \hat{\mathbf{r}}_t^{(3)} = \arg\min_{\mathbf{r}_t^{(2)}, \mathbf{r}^{(3)}} \mathcal{L}_t(\mathbf{r}_t^{(2)}, \mathbf{r}^{(3)}, \hat{\mathbf{r}}_{t-1}^{(1)}, \hat{\mathbf{r}}_{t-1}^{(2)}, b_t, \mathbf{U},$
      $\{\mathbf{V}_k^{(1)}\}_{k=1}^{K^{(1)}}, \{\mathbf{V}_k^{(2)}\}_{k=1}^{K^{(2)}}, \theta, \mathbf{I}_t)$
      $\mathcal{L} = \mathcal{L} + \mathcal{L}_t(\hat{\mathbf{r}}_t^{(2)}, \hat{\mathbf{r}}_t^{(3)}, \hat{\mathbf{r}}_{t-1}^{(1)}, \hat{\mathbf{r}}_{t-1}^{(2)}, b_t, \mathbf{U},$
      $\{\mathbf{V}_k^{(1)}\}_{k=1}^{K^{(1)}}, \{\mathbf{V}_k^{(2)}\}_{k=1}^{K^{(2)}}, \theta, \mathbf{I}_t)$
    end for
  end for
  `// Learning`
  Update parameters using gradients calculated from $\frac{1}{M}\mathcal{L}$ and their corresponding optimizers from Table 4.6
end while

Return $\{\mathbf{V}_k^{(2)}\}_{k=1}^{K^{(2)}}, \theta$

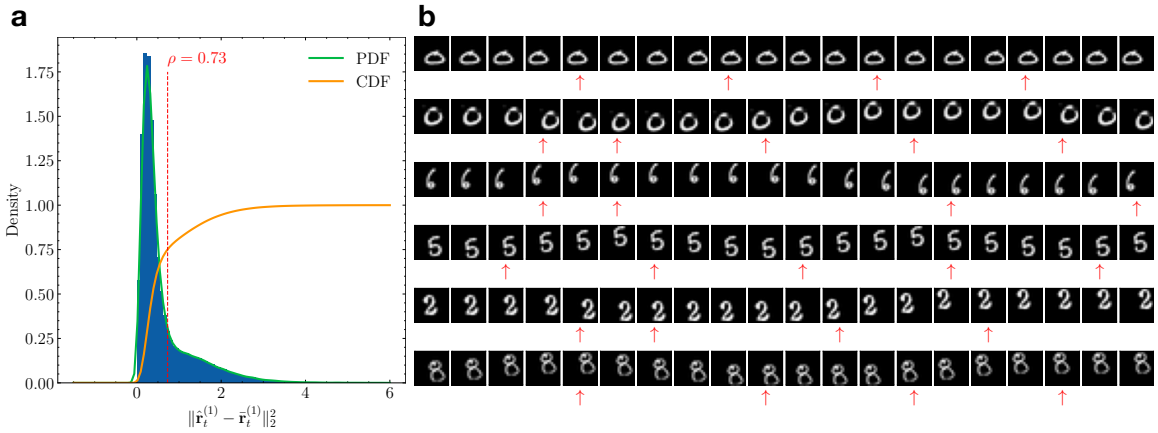


Figure 4.9: **Prediction error threshold robustly finds changes of dynamics. (a)** The distribution of first-level prediction errors in the two-level DPC model on the Moving MNIST training set. The red dashed line denotes the threshold $\rho = 0.73$, where the cumulative density reaches 0.75. **(b)** Examples of input sequences in the test set. The red arrows mark time steps when the first-level prediction errors exceeded ρ, corresponding to changes in input dynamics. (figure from Jiang and Rao, 2024)

Parameter	Description	Value
$\mathbf{r}^{(3)} \in \mathbb{R}^{N^{(3)}}$	Third-level latent variable	$N^{(3)} = 10$
$\sigma^2_{r(2)} \in \mathbb{R}$	Variance of second-level prediction error	1
$K^{(2)}$	Number of second-level transition matrices	2
$\mathbf{V}_k^{(2)} \in \mathbb{R}^{N^{(2)} \times N^{(2)}} \quad \forall k = 1 \dots K^{(2)}$	Learnable second-level transition matrices	Learned
$\mathcal{H}_\theta^{(2)}$	Third-level top-down dynamics network	Learned

Table 4.5: **Additional parameters and values for the three-level DPC model.**

Optimizee / Criteria		Optimizer / Value
Inference		
$\mathbf{r}_t^{(2)}$	$\forall t = 1 \dots T-1$	ADAM (1×10^{-3})
$\mathbf{r}^{(3)}$		ADAM (1×10^{-3})
Convergence criterion		$\|\Delta\|_2 \leq 0.01$
Learning		
$\mathbf{V}_k \in \mathbb{R}^{N^{(2)} \times N^{(2)}}$	$\forall k = 1 \dots K^{(2)}$	ADAM (1×10^{-3})
$\mathcal{H}_\theta^{(2)}$		ADAM (1×10^{-3})

Table 4.6: **Additional optimizers and learning rates used for inference and learning in the three-level DPC experiments.**

Chapter 5

DATA HETEROGENEITY LIMITS THE SCALING EFFECT OF PRETRAINING IN NEURAL DATA TRANSFORMERS

In this chapter, we turn our focus from predictive coding, a normative theory of cortical function, to predictive models trained on neural recordings. Inspired by recent advances in large-scale machine learning systems such as large language models, we ask whether modern foundation-model methods can learn generalizable representations from heterogeneous neural recordings.

5.1 Introduction

Recent advances in foundation models have revolutionized the modern machine learning paradigm. Across domains such as language and vision, it has been shown that "pretraining" a generic model on large-scale data before "finetuning" it to the actual tasks achieves much better performance than task-specific models (Devlin et al., 2019; Brown et al., 2020; Chung et al., 2024). This success has inspired similar efforts to develop **neuro foundation models**[1], where the goal is to develop foundation models trained on large, multi-session, multi-animal recordings of neural activity. This direction is particularly compelling as modern recording technologies are producing neural datasets that are rapidly increasing in scale, across animals and modalities (Jun et al., 2017; Steinmetz et al., 2021). These advancements have also spurred public interest and investment in brain-computer interfaces (BCIs), with companies such as Neuralink and Synchron accelerating invasive BCI clinical trials in humans (Oxley et al., 2016). A single large-scale foundation model that can be finetuned for multiple decoding tasks for BCIs could substantially broaden the applicability of the technology to a wider range of clinical and commercial users.

A major driver of recent machine learning advances is **scaling**. Scaling refers to the

[1]We use the term "neuro" rather than "neural" because "neural" is often used as a general descriptor for algorithms based on neural networks.

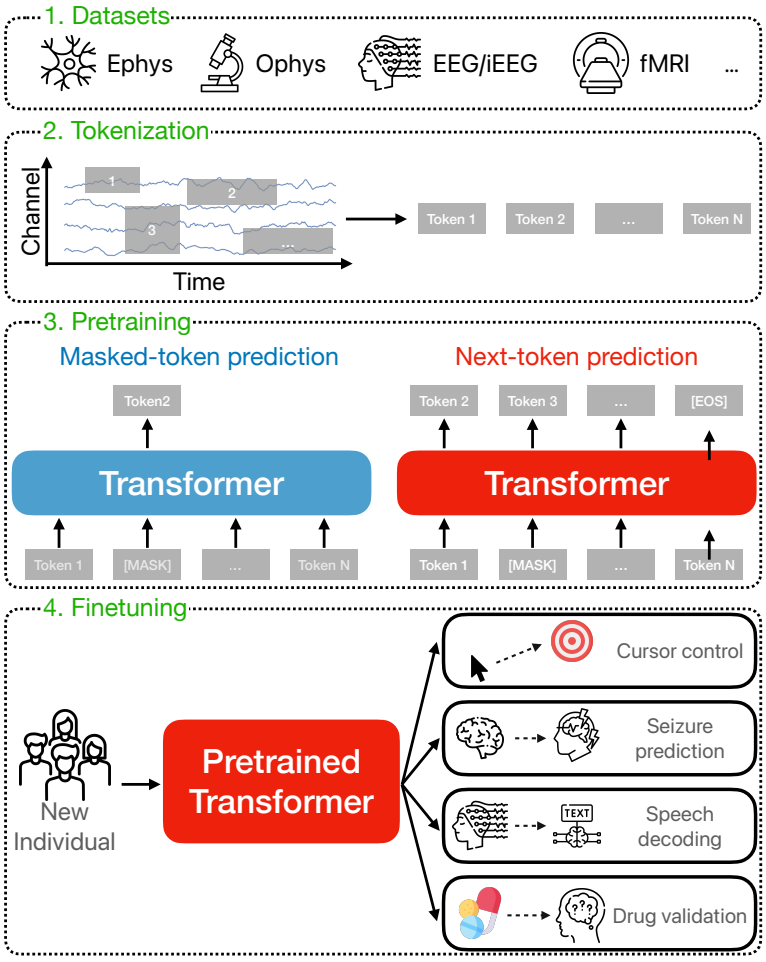


Figure 5.1: **Overview of the key components for building neuro foundation models.** We discuss various tokenization approaches for neural data in Section 5.1.1.

empirical observation that model performance often improves when the amount of training data and model capacity increase (Kaplan et al., 2020). Developments of large language models (LLMs) over the past few years have heavily relied on the expectation of their scaling behaviors (Kaplan et al., 2020; Hoffmann et al., 2022; Muennighoff et al., 2023). Besides language models, similar scaling trends have also been observed in models applied to other domains such as audio and video. A common belief is that similar scaling trends will hold for neural data. However, neural recordings pose unique challenges: data collected across brain

regions, sessions, and individuals often exhibit substantial variability (Laboratory et al., 2021, 2025; Waschke et al., 2021). Even within the same recording session, stochasticity of neuronal firing and uncontrolled behavior can lead to significant trial-to-trial variability (Harris and Thiele, 2011; Stringer et al., 2019; Peterson et al., 2021). Furthermore, neural data can be non-stationary due to synaptic plasticity that induces gradual changes in population dynamics across days (Rule et al., 2019; Driscoll et al., 2022). These challenges raise a key question: Can neuro foundation models overcome these sources of variability and learn more generalizable representations with more pretraining data?

Figure 5.1 presents an overview of the key components for building neuro foundation models. Here, we introduce (1) different tokenization approaches for neural data, arguably the most distinctive aspect of neuro foundation models compared to large-scale pretrained models in other domains, and (2) scaling law research in LLMs and the key factors used to characterize model scaling behaviors. We then present the work in this chapter on investigating how substantial variability in neural data, i.e. data heterogeneity, affects the scaling behaviors of neuro foundation models trained on mouse electrophysiology data.

5.1.1 Tokenization of Neural Data

In NLP and LLM research, tokenization is the process of converting raw text into a sequence of discrete units, or *tokens*, that define the vocabulary a language model operates over (Sennrich et al., 2016; Kudo and Richardson, 2018). This vocabulary serves two roles. First, it provides a lookup table that maps each token to an input embedding vector fed into the model. Second, its size determines the dimensionality of the model's output space for next-token prediction. The token embeddings themselves are learned parameters, optimized jointly with the rest of the model. Although text tokenization remains an active area of research (Pagnoni et al., 2025; Liu et al., 2025; Hwang et al., 2026), frequency-based algorithms like byte-pair encoding (Sennrich et al., 2016) are widely used in practice, offering good compression of raw texts on a sub-word level.

How to feed raw neural data into models is a much more challenging question, largely due to the constraints of recording technologies. Unlike text symbols whose meanings are

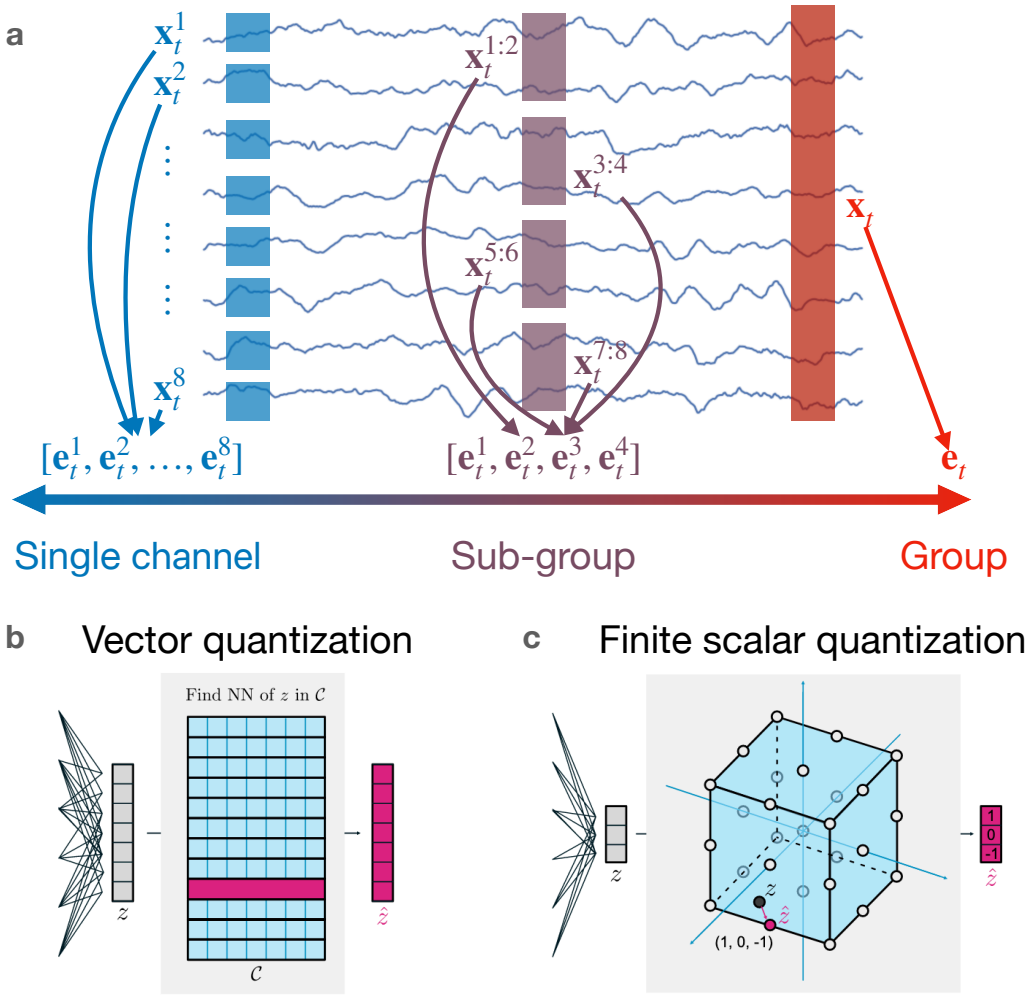


Figure 5.2: **Tokenization of neural data.** **(a)** Direct tokenization. A key difference in direct tokenization approaches is how individual channels are represented. Here, the schematic shows the full spectrum of group-level, sub-group-level, and single-channel tokenization choices. **(b) & (c)** schematic for vector tokenization (VQ) and finite scalar quantization (FSQ), respectively. Adapted from (Mentzer et al., 2024).

consistent across contexts, neural data are extremely heterogeneous: they are often recorded from different brain regions across many individuals, under a wide range of environmental and behavioral conditions. Take single-neuron electrophysiology data as an example. A group of N neurons recorded in one session from an animal could have few overlaps with a

group of M neurons recorded in a different session due to differences in electrode placement, spike sorting constraints, or neural plasticity over time (Harris and Thiele, 2011; Stringer et al., 2019; Peterson et al., 2021; Rule et al., 2019; Driscoll et al., 2022). An analogy to video is that each neural recording contains a shuffled subset of randomly selected pixels from a video, with unknown spatial correspondence and no guarantee that the same "pixels" reappear across datasets.

Therefore, to overcome these challenges, it is critical that the tokenization process constructs a shared latent space in which the model can learn representations that align and generalize across heterogeneous data sources. Because we consider tokenizers across multiple neural data modalities, we use the term ***channel*** to denote the smallest spatial unit of the modality. For instance, a channel may refer to a single neuron in spiking electrophysiology or calcium imaging, an electrode in an electroencephalography (EEG) grid, or a voxel in functional magnetic resonance imaging (fMRI). We organize tokenization methods along two axes: (1) the degree of channel grouping (group-level, sub-group-level, and single-channel tokens), and (2) whether tokenization is treated as a separate training stage (van den Oord et al., 2017; Rombach et al., 2022).

Direct tokenization

We start with approaches that do not treat tokenization as a separate training stage, which is similar to mainstream LLMs whose token embeddings are learned end-to-end together with the main network. A key design choice among direct tokenization approaches is how individual channels are represented (Fig 5.2(a)).

Group tokenization. One commonly used approach (also widely applied in neural population analyses (Vyas et al., 2020; Stringer and Pachitariu, 2024)) is to map the entire population of N channels at a given time step t, $\mathbf{x}_t \in \mathbb{R}^N$, to an embedding $\mathbf{e}_t \in \mathbb{R}^d$:

$$\mathbf{e}_t \coloneqq \mathbf{f}_{\text{enc}}(\mathbf{x}_t), \tag{5.1}$$

where $\mathbf{f}_{\text{enc}} : \mathbb{R}^N \to \mathbb{R}^d$ is commonly implemented as a linear layer or a simple multi-layer perceptron (MLP) (Fig 5.2(a) right). A corresponding decoder can map embeddings or hidden representations of the main network back to the raw data space via $\mathbf{f}_{\text{dec}} : \mathbb{R}^d \to \mathbb{R}^N$.

For multi-session datasets with varying numbers of channels N_i, one can introduce session-specific encoding layers $\mathbf{f}_{\text{enc}}^{(i)} : \mathbb{R}^{N_i} \to \mathbb{R}^d$. Such an approach for "stitching" raw data from multiple sessions to a shared feature space was popularized by the LFADS model (Pandarinath et al., 2018), which learns linear mappings per session that transform raw data to input features of a nonlinear recurrent network. The original Neural Data Transformer (NDT) model (Ye and Pandarinath, 2021) does not consider multi-session training, so the raw spike counts were fed directly into the model (i.e., $\mathbf{f}_{\text{enc}}$ is the identity mapping). The multitask-masking model (MtM) (Zhang et al., 2024) similarly uses grouped-channel tokenizers for multi-session training, while incorporating multiple brain-region-based masking schemes of the input for pretraining.

Sub-group tokenization. Rather than grouping all channels into a single token, a common variant is to group K channels ($K < N$) into multiple spatial patches, which are then (typically linearly) mapped to tokens (Fig 5.2(a) middle). This sub-group tokenization design is analogous to the patch embeddings used in the Vision Transformer (ViT) (Dosovitskiy et al., 2021). By generating multiple spatial tokens from the channels, the main transformer can learn spatial relationships among channels through attention, relieving this burden away from relatively simple encoder networks of the tokenizer. The NDT2 and NDT3 models use this tokenizer design to group multiple neurons together and produce multiple tokens per time step (Ye et al., 2023, 2025). Notably, the same linear projection layer is used for all patches in these models. Additionally, spatial embeddings were added to different patches to distinguish their identities. Though less common, models of EEG or intracranial EEG (iEEG) data have also used sub-group tokens based on brain regions of electrode placements (Zheng et al., 2024).

Single-channel tokenization. Naturally, we can achieve the highest level of spatial resolution by setting $K = 1$, yielding *channel*-level tokens (Fig 5.2(a) left). Channel tokenization is commonly used in EEG and iEEG foundation models, where a chunk of neural data $\mathbf{x}_{t:t+L}^{(c)}$ of length L from channel c is mapped to a token through linear or 1D convolution layers (Yang et al., 2023; Zhang et al., 2023; Oganesian et al., 2025). For datasets of single neuron activities, the POYO-series of work (Azabou et al., 2023, 2025) uses per-neuron, per-spike level tokenization, where a token representing a spiking event of

neuron i at time t is formulated as

$$\mathbf{e}_t^{(i)} \coloneqq [\text{Embed}(\text{neuron } i); t]^\top \in \mathbb{R}^d. \tag{5.2}$$

Here, Embed(neuron i) $\in \mathbb{R}^{d-1}$ is the learned embedding of neuron i, and $t \in \mathbb{R}$ is the timestamp of the spike time. Thus, POYO's tokenizer converts population activity into a token sequence whose length equals the total number of spikes across all neurons.

As K decreases, the spatial resolution of tokens increases because each token represents fewer channels. The main advantage of a smaller K is that more spatial information about inter-channel relationships is preserved in the token sequences, which can facilitate better alignment across different channels from multiple sessions and subjects through transformer attention. An analogy can be made here to word-level (higher K) versus subword-level (lower K) tokenization in LLMs. With subword tokenization, words like "eating" and "drinking" may be broken into "eat", "drink", and "ing" tokens, which helps the model better understand the meaning of the "ing" suffix. Such a level of generalization is not possible with word-level tokenization.

However, the obvious disadvantage of spatially finer-grained tokens is computational cost. The computational cost of the attention mechanism grows quadratically with the token sequence lengths (Vaswani et al., 2017). For neural data with N channels and T time steps, the group tokenization produces sequences of length $\mathcal{O}(T)$, whereas channel-level tokenizers like POYO yield sequences of length $\mathcal{O}(NT)$. For EEG or iEEG data, the number of channels is typically small (around 100) and the lengths of tokens are typically heavily reduced from raw data by convolution. Therefore, channel-level tokenization is more tractable in these data modalities. In contrast, neuron-level datasets usually contain a much higher number of channels. Directly applying self-attention on channel-level tokens is most certainly intractable. We introduce different types of architectural designs (such as POYO's PerceiverIO architecture (Jaegle et al., 2022)) that reduce computation cost on channel-level embeddings. Sub-group tokenization offers a compromise between computational efficiency and preserved spatial resolution. However, patching channels into tokens can introduce additional inductive biases, which may reduce transferability across sessions or subjects (Ye et al., 2025).

In summary, direct tokenization approaches face a fundamental tradeoff: preserving spatial information across channels while keeping computational cost tractable. It remains an active area of research to investigate how different tokenization choices can optimally balance pretraining efficiency, downstream task performance, and cross-subject generalization across neural data modalities.

Tokenization as a separate stage

In contrast to direct tokenization approaches, one can instead train a tokenizer on neural data as a separate stage prior to training the main model. A trained tokenizer converts raw neural data to feature embeddings in a latent space, which become the input to the main network. The purpose of a separate tokenizer is to learn a compressed representation that aligns data across sessions and subjects while reducing dimensionality and noise. The main network can then learn spatiotemporal features in this lower-dimensional latent space, an approach that has proven essential for visual generation tasks (Rombach et al., 2022; Peebles and Xie, 2023). Here, we describe in detail the most commonly used class of tokenizers in neuro foundation models based on vector-quantized variational autoencoders (VQ-VAEs) (van den Oord et al., 2017).

We consider the common case where raw data from each channel is encoded to latent vectors independently ($K = 1$). For a vanilla autoencoder, the setup is

$$\mathbf{z} = \mathbf{f}_{\text{enc}}(\mathbf{x}) \tag{5.3}$$

$$\hat{\mathbf{x}} = \mathbf{f}_{\text{dec}}(\mathbf{z}), \tag{5.4}$$

where $\mathbf{x} \in \mathbb{R}^T$ is the one-dimensional signal of a single channel with length T, and $\mathbf{z} \in \mathbb{R}^d$ is the latent representation of the raw data. Weights of the encoder and decoder are optimized to minimize the reconstruction loss

$$\mathcal{L}(\mathbf{x}) = \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2. \tag{5.5}$$

The extra step of processing by a VQ-VAE is to replace the continuous latent vector $\mathbf{z}$ with its nearest neighbor in a learned codebook $\mathcal{E} = \{\mathbf{e}^{(i)}\}_{i=1}^{M}$ consisting of M vector embeddings

(Fig 5.2(b)):

$$i = \underset{m=1,\ldots,M}{\arg\min} \|\mathbf{e}^{(m)} - \mathbf{z}\|_2^2 \tag{5.6}$$

$$\tilde{\mathbf{z}} = \mathbf{e}^{(i)}. \tag{5.7}$$

Instead of $\mathbf{z}$, $\tilde{\mathbf{z}}$ is used to produce the reconstruction $\hat{\mathbf{x}} = \mathbf{f}_{\text{dec}}(\tilde{\mathbf{z}})$. The objective function is then

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{codebook}} + \beta\mathcal{L}_{\text{commitment}} \\ &= \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \|\mathbf{e}^{(i)} - \texttt{sg}(\mathbf{z})\|_2^2 + \beta\|\mathbf{z} - \texttt{sg}(\mathbf{e}^{(i)})\|_2^2,\end{aligned} \tag{5.8}$$

where $\texttt{sg}$ is the stop-grad operator. The first term in Eq 5.8 is the same reconstruction loss as those in vanilla autoencoders. Since the arg min operation is non-differentiable, we use straight-through gradient estimation (STE) to copy the gradient from the decoder back to the encoder. The second term (codebook loss) updates the codebook embeddings to be close to the encoder output, since it receives no gradients from the reconstruction loss due to STE. Lastly, the commitment loss (third term in Eq 5.8) encourages the encoder to generate outputs close to the embeddings in the codebook.

In practice, a length T signal $\mathbf{x}$ is mapped to multiple codebook embeddings to ensure a reasonable compression rate. The resulting latents are often referred to as discrete latents, since each latent corresponds to one of the M embeddings in the VQ codebook $\mathcal{E}$. Once the tokenizer is trained, raw inputs are first mapped to the codebook embeddings, which then serve as input to the main transformer network. For the self-supervised objectives of the main network, the main network is trained to predict the index of the codebook entry, rather than the raw input or continuous embedding directly. This reframes the typical regression problem of predicting raw input signals as the same multi-class classification problem in LLMs.

Many state-of-the-art EEG and iEEG foundation models adopt a two-stage design that pairs a VQ-VAE tokenizer with a main transformer model. Du-IN (Zheng et al., 2024) trains its tokenizer on sub-group-level patches of iEEG data drawn from similar brain regions. LaBram (Jiang et al., 2024) is an EEG foundation model that trains a VQ-VAE tokenizer

with a transformer backbone. Notably, LaBram's tokenizer reconstructs both the amplitude and the phase of the frequencies from the original input signal. The follow-up work of LaBram, NeuroLM, trains the tokenizer to reconstruct both the raw input and the amplitude of frequencies (without the phase) (Jiang et al., 2025b). CodeBrain (Ma et al., 2026) uses two separate codebooks in its tokenizer, one trained to reconstruct the frequency information like LaBram, and the other one trained through contrastive learning on raw signals. The authors show that combining these codebooks improves decoding accuracies compared to using either signal codebook. The two codebooks provide complementary views of the data: the same latent code in the time domain can be disambiguated by additional latent codes capturing frequency content to distinguish events such as sleep stages.

Variants of vector quantization Several variant approaches have been proposed to improve the vector quantization procedure described above. Residual Vector Quantization (RVQ) extends standard VQ by converting the raw data to multiple embeddings from different codebooks sequentially (Zeghidour et al., 2022). RVQ first quantizes the input $\mathbf{z}$ with one embedding $\mathbf{e}^{(1)}$ (same as VQ), then quantizes the residual error ($\mathbf{z} - \mathbf{e}^{(1)}$) with another embedding $\mathbf{e}^{(2)}$ from the second codebook, and so on. The final representation is the sum of several codebook embeddings, one from each stage. This multi-codebook design lets RVQ achieve much higher reconstruction quality than the standard single-stage VQ. RVQ has been shown to improve generation performance in audio and image domains, and has recently been explored as an extension of LaBram's tokenizer in NeuroRVQ (Barmpas et al., 2025).

Finite Scalar Quantization (FSQ) is a simpler alternative to VQ that achieves much higher codebook usage rate (Mentzer et al., 2024) (Fig 5.2(c)). Instead of quantizing the input to the nearest neighbor embedding in the codebook, FSQ proposes to directly quantize each dimension of the latent vector to a finite set of values (e.g., $\{-1, 0, 1\}$ or M equally spaced bins). Such an approach is codebook-free: instead of learning a table of embeddings, discrete tokens are represented by unique combinations of per-dimension scalar quantization levels. Moreover, the loss function now only requires the first term in Eq 5.8 since there are no codebook parameters to optimize. The authors have shown that FSQ achieves

comparable or better reconstruction and generation performances than VQ with a much higher codebook usage rate. These properties make FSQ a promising, yet underexplored, direction for neural data tokenization.

5.1.2 Scaling Behaviors in Neuro Foundation Models

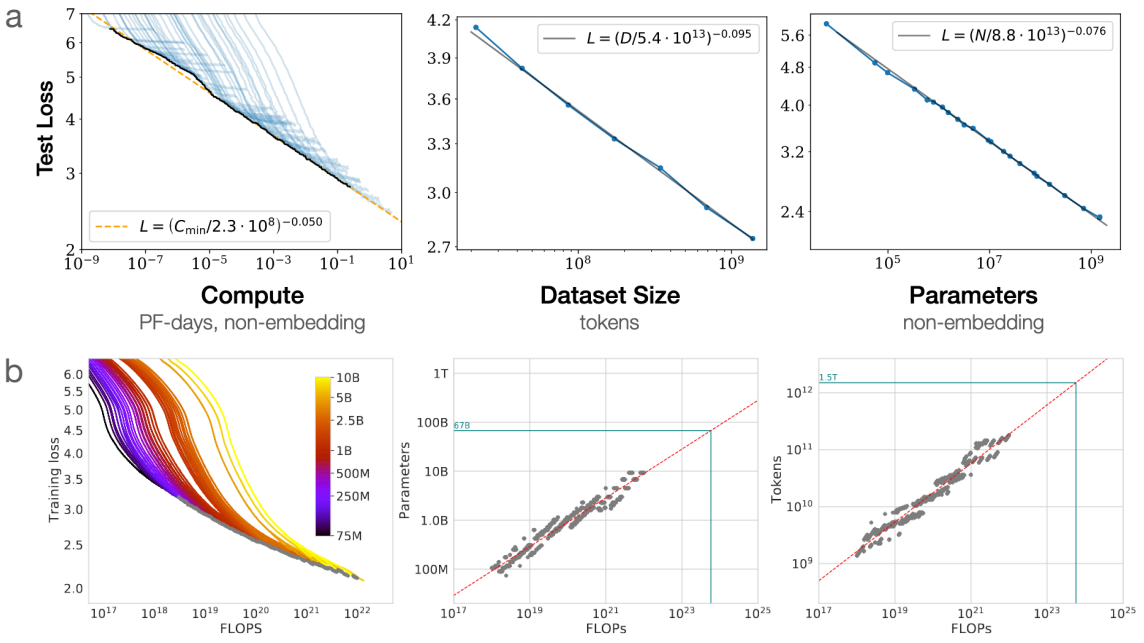


Figure 5.3: **Scaling laws in LLMs. (a)** Test loss in LLMs decreases predictably with three key factors: compute (left), dataset size (middle), and model size (right). The empirical losses of the model decrease as a power law with respect to each of the three factors. Adapted from Kaplan et al. (2020). **(b)** The scaling law curves fitted by Hoffmann et al. (2022). The left panels plot the best test loss of all runs with the x-axis being the compute budget. The middle and right panels show the optimal model size and dataset size under a compute budget, respectively. Their 70B model, Chinchilla, requires 1.5 trillion training tokens to achieve optimal performance.

In the previous sections, we introduced key components for building a neuro foundation model, including tokenization design, transformer attention mechanisms, and training ob-

jectives for pretraining and finetuning. In this section, we turn to a major driver of recent machine learning advances: ***scaling***.

Scaling refers to the empirical observation that model performance often improves when training data size and model capacity increase. Importantly, Kaplan et al. (2020) have shown that a model's test loss can decrease *predictably* as model size and training data grow. Specifically, model performance strongly depends on the scale of the following three factors: the number of model parameters N, the size of the dataset D, and the amount of compute C used for training (Fig 5.3(a)). Compute C is typically measured in the number of floating-point operations (FLOPs) during training. For a standard dense transformer with forward and backward passes, a rough equation for estimating C is

$$C \approx 6ND. \tag{5.9}$$

When fitting scaling law curves, one measures the trend of a model's test loss by varying one of the three factors while keeping the other two constant. In the literature, the basic empirical observation is that test loss decreases as a power law with the scale factor. A commonly used fitting form is (using N as an example):

$$L(N) \approx \left(\frac{N_c}{N}\right)^{\alpha_N}, \tag{5.10}$$

where N_c is a scale constant and α_N is the power law coefficient fitted from loss points. Fig 5.3(a) shows the fitted curves of test loss with respect to each of the three factors.

An important implication of scaling laws is captured by Eq 5.9: for a fixed compute budget C, there exists an explicit tradeoff between model size and training data size. A key follow-up work by Hoffmann et al. (2022) studied how to choose a compute-optimal combination of model size and dataset size (Fig 5.3(b)). Their empirical results suggest that large-scale models are often *under-trained*, and that compute-optimal training requires scaling model size and training tokens roughly in equal proportion. Their 70B-parameter model, Chinchilla, was trained on 1.4T tokens and achieved better performance than 180B- and 280B-parameter models trained under the same compute budget.

To date, there are no studies on the scaling behaviors of neuro foundation models at a comparable scale to LLMs or vision models. While several recent studies have demonstrated

performance gains from multi-session pretraining on a wide range of encoding and decoding tasks, they typically lack fine-grained scaling analyses on the benefits of gradually increasing pretraining data (Azabou et al., 2023, 2025; Zhang et al., 2024, 2025). Most comparisons are limited to models trained on single sessions versus entire datasets with few increments in the middle, making it unclear how different sources of heterogeneity impact performance scaling. Moreover, it remains unknown whether all pretraining sessions contribute equally to downstream performance improvements. As pretraining scales to thousands of sessions and hours of data (Ye et al., 2025; Azabou et al., 2025), understanding the scaling behaviors of the model becomes increasingly critical.

In this chapter, we systematically investigate how data heterogeneity affects the scaling behavior of neural data transformers (NDTs) (Ye and Pandarinath, 2021; Zhang et al., 2024; Ye et al., 2025) using two large-scale datasets released from the International Brain Laboratory (Laboratory et al., 2025; International Brain Laboratory et al., 2025). These datasets differ in the consistency of recorded brain regions across sessions, offering us an opportunity to study how different levels of brain region heterogeneity in pretraining affect scaling. We further examine the effects of implicit heterogeneity such as session-to-session variability. Through a proposed session-selection procedure, we identify the impact of each pretraining session on downstream performance improvements. Our main findings include:

- We found that greater region-wise heterogeneity in pretraining data led to reduced improvements in neuron- and region-level activity prediction performance.

- To identify implicit heterogeneity, a session-selection procedure based on single-session finetuning performance can effectively identify the most beneficial single sessions for pretraining.

- Models trained with as few as five selected sessions outperformed those with randomly chosen sessions even when the full dataset was used, demonstrating the impact of session-to-session variability in performance scaling.

Together, these findings suggest that previous claims regarding the scaling benefits of pretraining without detailed incremental experiments may be premature, pointing to the need

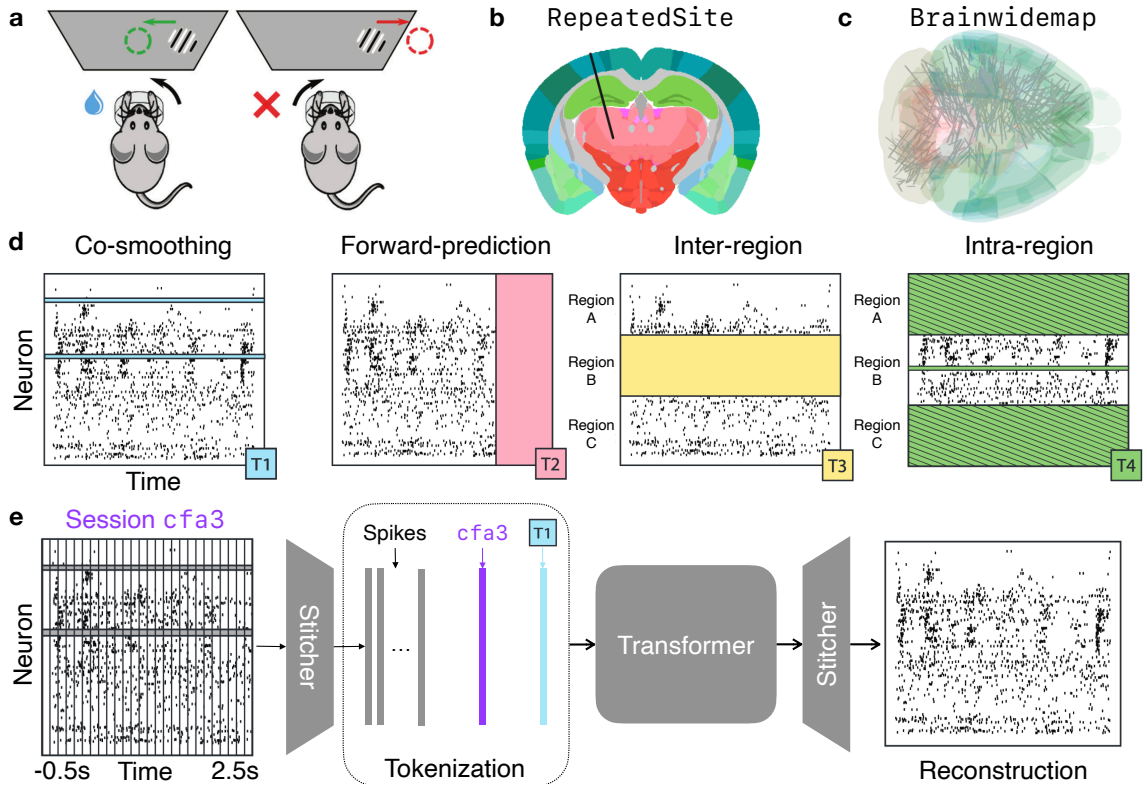


Figure 5.4: **Experimental Setup.** **(a)** Schematic of the visual decision-making task performed by mice. **(b)** Planned probe insertion location (black line) for all sessions in the `RepeatedSite` dataset. **(c)** Different probe insertion locations (gray lines) for different sessions in the `Brainwidemap` dataset. **(d)** Four different masking schemes of raw spike counts for model training. **(e)** The model architecture. Sub-figures adapted from Laboratory et al. (2021, 2025); International Brain Laboratory et al. (2025); Zhang et al. (2024). See text for details.

for rigorous scaling analyses in future work on neural foundation models to accurately assess the promise of large-scale pretraining.

5.2 Experimental Setup

Figure 5.4 summarizes the experimental setup used throughout this study, which mostly follows Zhang et al. (2024) whose experiments were conducted on a subset of the same

`RepeatedSite` dataset used here. This section discusses the datasets, training pipeline, and evaluation metrics in detail below. More details are included in Section 5.5.1.

5.2.1 Datasets

We used two multi-brain-region, multi-animal/session datasets from the International Brain Lab (IBL) collected from mice. Animals performed a visual decision-making task where they detected the presence of a visual grating (of varying contrast) to their left or right and rotated a wheel to bring the stimulus to the center (Fig 5.4(a)). The main difference between the two IBL datasets lies in how often brain regions were repeatedly recorded across sessions. In the `RepeatedSite` dataset (henceforth `RS`), each session attempted to record from the *same* brain regions (Fig 5.4(b), black line shows planned electrode insertion position). In contrast, the `Brainwidemap` dataset (henceforth `BWM`) aimed to cover as many *different* brain regions as possible (Fig 5.4(c), gray lines show planned insertion positions), leading to little repetition of regions across sessions.

We used 89 out of 91 sessions in `RS`, excluding two sessions with fewer than one hundred trials. Five out of 89 sessions were held out for finetuning and evaluation. We randomly selected 200 sessions out of 460 sessions in `BWM` for pretraining and 10 sessions for finetuning and evaluation. Trials within each session were randomly split into training, validation, and test sets using an 8:1:1 ratio. Each trial included three seconds of neural activity, spanning from 0.5 seconds before to 2.5 seconds after stimulus onset with 20 ms bins for spike counts. The data from each session is thus a three-dimensional (trials $\times$ timesteps $\times$ neurons) tensor of integer spike counts.

5.2.2 Training pipeline

Multi-masking scheme During training, input spike count vectors were masked in one of four ways, as illustrated in Figure 5.4(d): (1) Co-smoothing: selected neurons' activities are masked; (2) Forward-prediction: all neurons' selected timesteps are masked; (3) Inter-region: all neurons' activities in a selected brain region are masked; (4) Intra-region: selected neurons in a selected brain region, along with all other neurons in other brain regions, are

masked. Models were trained to reconstruct masked input from the unmasked input (Devlin et al., 2019) by maximizing the log likelihood of the Poisson distribution, with the model outputs as the predicted firing rates. We also applied causal attention in NDT if inputs were masked with forward-prediction to ensure no future neural activities were used to predict the present. Zhang et al. (2024) showed that this multi-masking scheme significantly outperforms a scheme using forward-prediction masks alone in spike prediction tasks.

Model architecture We used the neural data transformer (NDT) architecture by Ye and Pandarinath (2021) that has been widely applied to neural encoding and decoding tasks (Le and Shlizerman, 2022; Ye et al., 2025; Zhang et al., 2025). NDT also achieves state-of-the-art performance on the IBL dataset we used (Zhang et al., 2024, 2025). Since different sessions have different numbers of neurons recorded, a session-specific linear layer (encoding "stitcher") maps raw spike counts to spike embeddings (Fig 5.4(e) left) whose dimensions are shared across sessions (Pandarinath et al., 2018). A session embedding and a masking scheme embedding are also appended to input sequences. Lastly, another session-specific linear layer (decoding stitcher) maps the output of the transformer back to reconstructed spike rates (Fig 5.4(e) right).

5.2.3 Evaluation

Baseline To show the effect of scaling up pretraining data, we directly trained single-session models on the training set of each heldout session as the baseline models.

Metrics The bits-per-spike metric (BPS) is widely used to evaluate neural activity prediction performance (Rieke et al., 1999; Pei et al., 2021; Zhang et al., 2024, 2025):

$$\text{bits-per-spike}\left(\hat{\lambda}, \mathbf{X}\right) = \frac{1}{n_{sp} \log 2}\left(\mathcal{L}\left(\mathbf{X}; \hat{\lambda}\right) - \mathcal{L}\left(\mathbf{X}; \bar{\lambda}\right)\right), \tag{5.11}$$

where $\hat{\lambda}$ is the predicted spike rates by the model, $\mathbf{X}$ is the true spike counts, n_{sp} is the total spike count of $\mathbf{X}$, $\mathcal{L}$ is the log likelihood function of Poisson, and $\bar{\lambda}$ is the mean firing rate of $\mathbf{X}$. The BPS metric essentially evaluates the goodness-of-fit statistics of a model over the null model, normalized by the spike counts. Changes in BPS directly reflect changes in

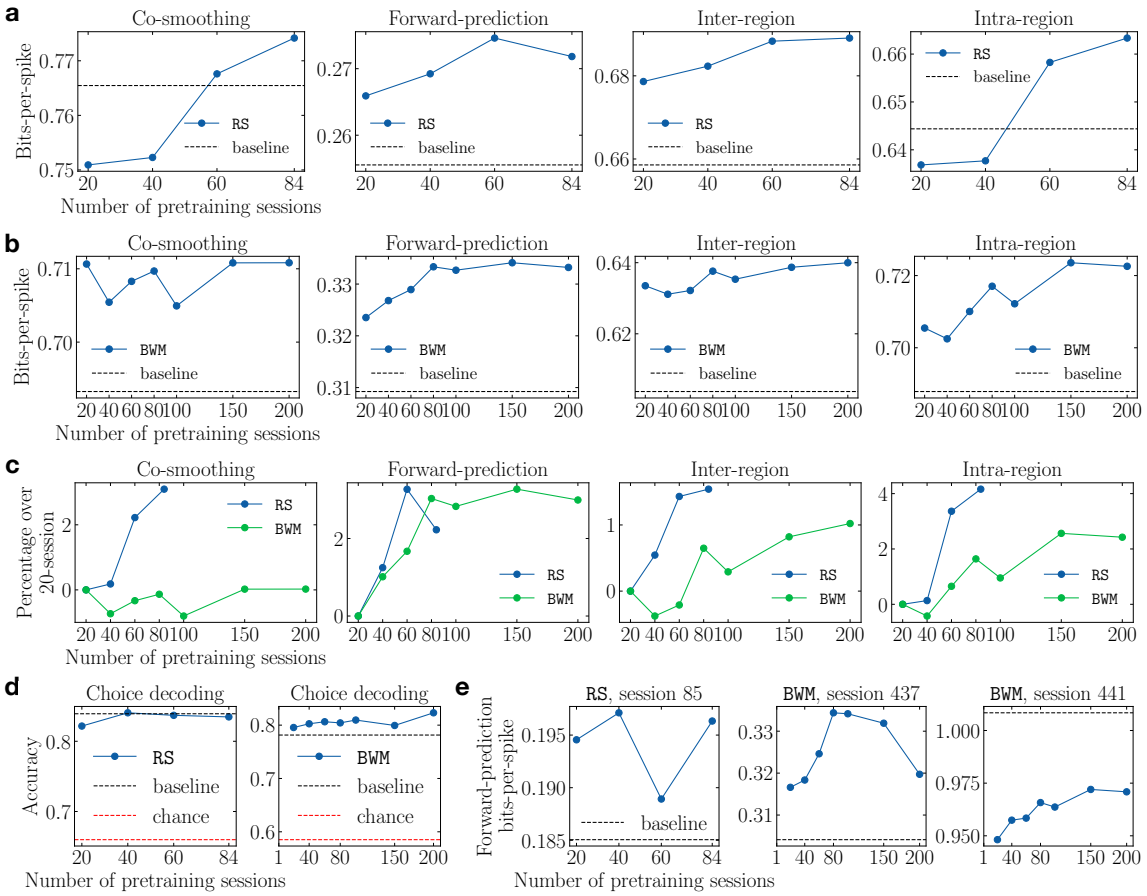


Figure 5.5: **Scaling Behavior of NDT Models.** Plots show pretrained NDT models' finetuning performances as the number of pretraining sessions increases. **(a)** Performances of each neural activity prediction task on `RS` data. Black dashed lines show the baseline models' performances. **(b)** Same as (a) but on `BWM` data. **(c)** Percentage improvements of models pretrained with more sessions over the 20-session model. **(d)** Choice decoding performances on `RS` (left) and `BWM` (right). Red dashed line shows the chance prediction accuracy. **(e)** Forward-prediction performance examples on individual heldout sessions from `RS` (first panel) and `BWM` (second & third panels).

model log likelihood $\mathcal{L}\left(\mathbf{X}; \hat{\lambda}\right)$ when evaluated on the same dataset, as other terms remain constant. For all experiments, we report the metrics on the test sets of the heldout sessions after finetuning models to their training sets.

Tasks Each masking scheme used for training corresponds to a leave-one-out evaluation task of activity prediction, namely: (1) Co-smoothing: Activities of each neuron were predicted from all other neurons; (2) Forward-prediction: The model predicted neural activities in continuous, non-overlapping windows of 200 ms at a time (10 timesteps) given previous ground truth activities, starting from stimulus onset to 2.2 seconds after stimulus onset (110 timesteps in total); (3) Inter-region: Activities of all neurons in each region were predicted from all other regions; (4) Intra-region: Activities of each neuron in a region were predicted from all other neurons in that region. This was repeated for each region.

5.3 Results

5.3.1 Data heterogeneity limits NDT models' scaling behavior

As mentioned, the difference in brain region overlaps between `RS` and `BWM` provides a natural setting to study how pretraining data heterogeneity affects model scaling behavior. `BWM` sessions contain neural activity from largely non-overlapping brain regions, resulting in greater single-neuron and brain-region heterogeneity compared to `RS`. We hypothesize that increased heterogeneity in `BWM` will reduce the scaling benefits from pretraining on more sessions.

We conducted our scaling analysis as follows. Using the `RS` dataset, we pretrained NDT models on 20, 40, 60, and 84 sessions, then finetuned them to each of five heldout sessions. For `BWM`, we pretrained models on 20, 40, 60, 80, 100, 150 and 200 sessions (out of 460 total), then finetuned them on ten heldout sessions. During finetuning, a new session embedding and two session-specific stitchers were learned from scratch while the mask embedding and the core NDT parameters were initialized from pretraining.

Performance scaling is weaker in `BWM` than `RS`

Before analyzing whether pretrained models' performances scale with more pretraining data, we first confirm they indeed outperformed baseline models (Section 5.5.2). Next, we investigate whether the task performances scale with increased pretraining data. Figure 5.5(a) and (b) show the evaluation results of models trained on `RS` and `BWM` data, respectively.

Although performance generally improved with more pretraining data, the scaling effects were relatively modest. Figure 5.5(c) illustrates the percentage performance gains relative to the 20-session model, with the largest improvement of 4.2% observed on the intra-region task on `RS` using more than four times the amount of data. Notably, performance gains were reduced across the co-smoothing, inter-region, and intra-region tasks on `BWM` compared to `RS`. For co-smoothing in particular, scaling benefits were negligible in `BWM`. In contrast, the forward-prediction performance scaled consistently on the two datasets. However, the forward-prediction performance also plateaued around 80 pretraining sessions, suggesting a potential upper limit on the performance improvements achievable through pretraining given the current scope of data. These results support our hypothesis that greater brain region heterogeneity in `BWM` limits the effectiveness of pretraining, particularly on single-neuron and region-level tasks.

To probe the quality of the NDT's internal representations, we trained a logistic regression classifier on the output of the third intermediate transformer block (out of five, see Section 5.5.1) to predict the animals' decisions. Previous work typically performed such decoding analyses using reconstructed spikes rather than models' internal representations (Pei et al., 2021; Zhang et al., 2024). Under the latter setup, improvements in choice decoding performance could reflect better spike reconstruction rather than better representations of choice-related latent states underlying the data. With our setup, decoding accuracies are higher in `RS` than `BWM` (Section 5.5.2). Figure 5.5(d) shows the changes in classification accuracy with increasing amounts of pretraining data. There was no clear scaling of decoding performance on either dataset. Taken together, these results highlight the importance of investigating the scaling behaviors of pretrained models with more fine-grained data increments.

We also observed large cross-session variability in the finetuning performances. Panel (e) of Fig 5.5 shows the forward-prediction performance of three heldout sessions (see Section 5.5.3 for all sessions). In addition to the substantial differences in absolute bits-per-spike values, their scaling trends deviate from the session averages (second panel in Fig 5.5(a) & (b)). Such variability indicates a more implicit form of data heterogeneity that comes from individual differences among animals. Given this observation and our previous find-

ings on the pretrained models' limited scaling behavior, a natural question arises: **Can we identify more beneficial sessions than others in the pretraining dataset for improving scaling performance?** We answer this question in the next subsection.

5.3.2 Identifying more beneficial single sessions for performance scaling

We hypothesize that each pretraining session exhibits varying degrees of distribution shift relative to a heldout session. We call this implicit data heterogeneity, which arises from subtle individual differences among animals and sessions that are harder to identify than explicit sources of heterogeneity such as task design and brain regions. We expect that models pretrained on sessions "closer" to the heldout sessions will achieve higher performances more data-efficiently than models pretrained with randomly selected sessions.

To test this, we conducted our experiments on the `RS` dataset, which allows us to control the brain region heterogeneity as discussed in the previous subsection. We trained NDT models with only forward-prediction masking to focus on this evaluation task, which exhibits the most consistent scaling behavior on the two datasets (Fig 5.5(c)). For more fine-grained scaling analysis, we pretrained models on 1, 2, 3, 4, 5, 10, 20, 30, 40, and all 84 sessions.

Ranking pretraining sessions by single-session finetuning performances

First, we propose using single-session finetuning performances as an estimate of the "closeness" between the data distributions of a pretraining session and a heldout session. Figure 5.6 illustrates this process: during the pretraining stage, we trained 84 single-session models, one for each pretraining session (Fig 5.6(a)). During the finetuning stage, for a particular heldout session, we trained two new stitchers (for encoding and decoding) for each of the pretrained transformers while keeping the transformers' weights frozen (Fig 5.6(b)). This ensures the finetuning performance maximally depends on the features learned from the pretraining session, as the only adjustable weights were the input/output linear layers that map the raw spike counts to the frozen feature space and back. Lastly, we report each model's forward-prediction performance on the heldout session's validation set, yielding 84 metric values – one per pretraining session. The pretraining stage (Fig 5.6(a)) was per-

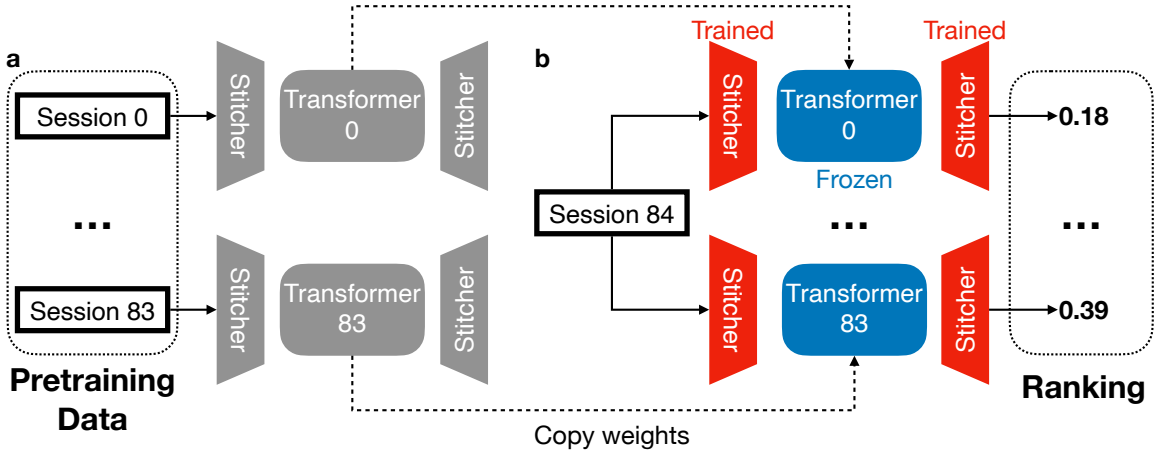


Figure 5.6: **Schematic of the ranking process. (a)** Pretraining stage: We trained 84 single-session models, each consisting of a transformer and two session-specific stitchers. **(b)** Finetuning stage: For each pretrained model, we trained two new stitchers on the heldout session's training set, keeping the transformer weights frozen. Models were ranked by their bits-per-spike metric on the heldout session's validation set.

formed once, while the finetuning and ranking stage (Fig 5.6(b)) were repeated for each heldout session. See Section 5.5.4 for the ranked single-session finetuning performances.

We conducted our data scaling experiments by incrementally selecting more pretraining sessions in three session-selection orders: random, ranked (based on the procedure above), and reverse-ranked. To reduce variance, we used three random seeds for both ranking sessions and training models in the scaling analysis, including the baseline models that were directly trained on the heldout sessions. For the random order, we used three different shuffles of the session list. The transformers' weights were frozen for all finetuning experiments across data orders to be consistent with the ranking procedure. We limited experiments to a maximum of 40 pretraining sessions (except for the full 84-session case) since more selected sessions overlap as we exhaust the pretraining data.

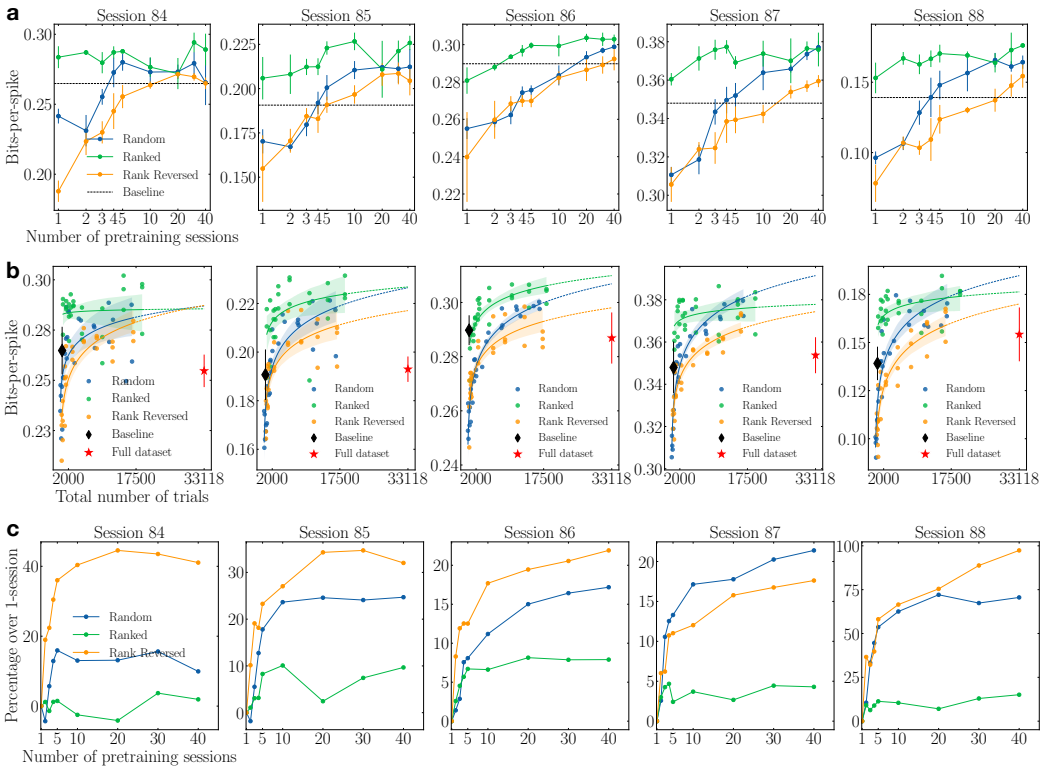


Figure 5.7: **Scaling performances under different session orders.** **(a)** Forward-prediction performances of each heldout session as we increased pretraining sessions according to random (blue), ranked (green), or reverse-ranked (orange) order. The error bars show the standard deviation over three seeds (ranked/reverse-ranked) or three shuffled orders (random). Black dashed lines show the baseline models' performance (averaged over three seeds). **(b)** Same as (a) but with the total number of trials as the x-axis. Linear regressions were fitted with logarithmic x values and dashed lines show extrapolated predictions. Shading shows the standard deviations. Red stars show the performances of the models pretrained with all pretraining sessions. **(c)** Percentage improvements of models pretrained with more sessions over the 1-session model.

Pretraining on five top-ranked sessions outperforms all random sessions

Figure 5.7(a) shows the performances of our scaling analysis on each heldout session's test set with different session orders (see Section 5.5.6 for qualitative examples). The results clearly

show that in all heldout sessions, models pretrained with ranked session order outperform those trained with randomly chosen sessions. Importantly, the models pretrained with reverse-ranked sessions achieved worse performances than random-order models, proving the validity of our ranking procedure based on single-session finetuning. Notably, the performance differences in ranked, random, and reverse-ranked settings are more pronounced in low-data regimes (fewer than ten sessions). Since the number of trials was different among sessions, we also plotted the models' performance in Figure 5.7(a) against the total number of trials from the pretraining sessions for fairer comparisons. As shown in Figure 5.7(b), the same performance differences hold among the different session-selection orders given the same number of trials. We fit the models' performances using linear regression (with logarithmic input). In contrast to the success of "scaling laws" in machine learning (Kaplan et al., 2020; Hoffmann et al., 2022; Muennighoff et al., 2023), the actual pretraining performance using the entire 84 `RS` sessions (Fig 5.7(b) red stars) is consistently lower than the extrapolated performance (Fig 5.7(b) dashed lines), indicating limited scaling effects for the neural IBL data with the NDT model. This further supports our hypothesis that differences in pretraining and finetuning data distributions greatly affect the promise of neural data scaling.

Table 5.1: Percentage improvements over baseline with different session selection procedures.

	Session selection order (best of all experiments)			
Heldout session	Random	Ranked	Reverse-ranked	Ranked (top 5 sessions)
84	5.74%	**11.08%**	2.54%	8.69%
85	11.33%	**18.87%**	9.39%	16.92%
86	3.13%	**4.78%**	0.89%	3.39%
87	8.37%	**8.43%**	3.30%	8.43%
88	19.12%	**26.60%**	10.95%	22.44%
Average	9.54%	**13.95%**	5.42%	11.97%

Table 5.1 summarizes the best percentage improvements over the baseline models for each session selection order, along with the performance of models pretrained on five top-ranked sessions. On average, models using rank-ordered session data achieved a 4% greater improvement over the baseline than models using random-ordered session data. Remarkably, models trained on just five ranked sessions outperformed the best models trained on randomly selected sessions, indicating an over 8× gain in data efficiency compared to 40 random session models, which outperformed the models trained on all sessions (Fig 5.7(b)). However, this also implies a reduced scaling effect compared to randomly selected sessions. Figure 5.7(c) demonstrates the percentage performance gains from using more pretraining data relative to using one pretraining session under each session selection order. The scaling effects when using the ranked sessions were clearly weaker than when using random or reverse-ranked sessions. Indeed, models with five ranked sessions already achieved 86% of the best model performance with all 40 ranked sessions (Table 5.1), suggesting that most of the pretraining benefit is concentrated in the top few sessions. The top sessions in different rankings were also sufficiently different based on the finetuning session used (Section 5.5.5). Taken together, our analysis suggests that apparent scaling benefits in multi-session datasets can be highly sensitive to the specific sessions selected, due to substantial individual differences across sessions. Thus, it is extremely important for studies that claim scaling benefits to show detailed experimental results with fine-grained data increments.

5.4 Discussion

Our results show that data heterogeneity in multi-session electrophysiology datasets fundamentally limits performance improvements expected from increasing pretraining data. Future work can focus on (1) scaling experiments with other architectures beyond NDT and different learning objectives, including supervised approaches; (2) different modalities of neural data such as calcium imaging and local field potentials[2], and (3) more computationally efficient session selection strategies. In conclusion, our results show that pretraining

[2]Concurrent work on motor decoding by Ye et al. (2025) reports that the benefits from pretraining the model on 2000 hours of data are virtually nonexistent when finetuning datasets exceed 100 minutes, supporting our hypothesis that data heterogeneity issues extend beyond our dataset.

neural encoding models with more sessions does not naturally lead to improved downstream performance. We strongly advocate for rigorous scaling analyses in future work on neuroscience foundation models to account for data heterogeneity effects.

5.5 Supporting information

5.5.1 Experimental setup details

In this section, we detail our experimental setup introduced in Section 5.2.

Model architecture

Our model follows the architecture of Zhang et al. (2024). At each time step t, a raw spike count vector $\mathbf{x}_t \in \mathbb{R}^{N_i}$ from session i (with N_i neurons recorded) is projected via a session-specific linear layer to a spike token with dimension d. Another linear layer with Softsign activation maps the spike tokens to embeddings. A session embedding and a masking scheme embedding were appended to the spike embedding sequence. Learned position embeddings are added to the input embeddings, making up the final input to the transformer block. Lastly, another session-specific linear layer maps the transformer output back to the spike count vectors of dimension N_i for session i.

Hyperparameter selection

We tuned learning rates, dropout rates, weight decay rates, and batch sizes using small-scale experiments on single- and ten-session models. Optimal values were chosen based on test losses from pretraining sessions in `RS`. Other hyperparameter values were inherited from the implementation of Zhang et al. (2024). Table 5.2 summarizes the hyperparameters we used for all experiments. We increased the spike token dimension to 1024 for `BWM` experiments to account for higher numbers of neurons. We used seed 42 for all experiments in Section 5.3.1, and seeds 10, 20, and 42 for experiments in Section 5.3.2 that required three seeds.

The two session-specific linear layers contain approximately 1.2 million parameters on average per session. The number of parameters in NDT (shared across sessions) is roughly 12 million with the values in Table 5.2, which achieved better performance than smaller 8

Table 5.2: Hyperparameter values across experiments

Hyperparameter	Value
	Model
Spike token dim	668
Embedding dim	512
Feedforward dim	1024
# attention heads	8
# transformer blocks	5
Dropout rate	0.2
	Training
Optimizer	AdamW (Loshchilov and Hutter, 2018)
Learning rate	1e-4
Learning rate scheduler	OneCycle (Smith and Topin, 2018)
Weight decay	0.01
Batch size	16
Gradient clipping	1
# Epochs	1000
Masking ratio	0.3

million models and larger 24/38 million models on `RS`. We did not run larger models on `BWM` due to computing resource constraints.

Compute

All models were trained on a single Nvidia A40 or L40 GPU. Single-session training and finetuning take about one hour to train on average. The full 84-session model on `RS` takes about 3.5 days, while the full 200-session model on `BWM` takes about a week to train. Finetuning jobs in Section 5.3.2 are significantly faster since we only train the two linear stitchers, with

each taking about 20 minutes to finish on average.

5.5.2 Pretraining offers performance improvements across tasks over baseline models

Table 5.3 presents the evaluation metrics achieved by the baseline models and the best pretrained-then-finetuned models on `RS` and `BWM`, averaged over all heldout sessions. Pretraining improves performance on both datasets, though gains are smaller for co-smoothing and choice decoding tasks than others. Notably, co-smoothing metrics are higher than intra-region in RS but not in BWM, suggesting that models trained on RS data benefited more from cross-region information. This is consistent with the distinct recording strategies of the two datasets.

Table 5.3: Evaluation metrics of baseline and best pretrained models on `RS` and `BWM`. Higher values indicate better performance.

	`RS`		`BWM`	
Task	Baseline	Pretrained	Baseline	Pretrained
Co-smoothing (BPS)	0.765	**0.774**	0.693	**0.711**
Forward-prediction (BPS)	0.256	**0.275**	0.309	**0.334**
Inter-region (BPS)	0.659	**0.689**	0.604	**0.640**
Intra-region (BPS)	0.644	**0.663**	0.688	**0.724**
Choice decoding (accuracy)	0.839	**0.841**	0.781	**0.823**

5.5.3 Cross-session variability in scaling behaviors

Here, we show detailed experimental results of the scaling analysis in Section 5.3.1. Figure 5.8 and Figure 5.9 show the finetuning performances on each heldout session in `RS` and `BWM`, respectively. As discussed in Section 5.3.1, there exists large cross-session variability in the finetuning performances among heldout sessions. These results suggest differential contributions of different sessions in pretraining, which we investigated in Section 5.3.2.

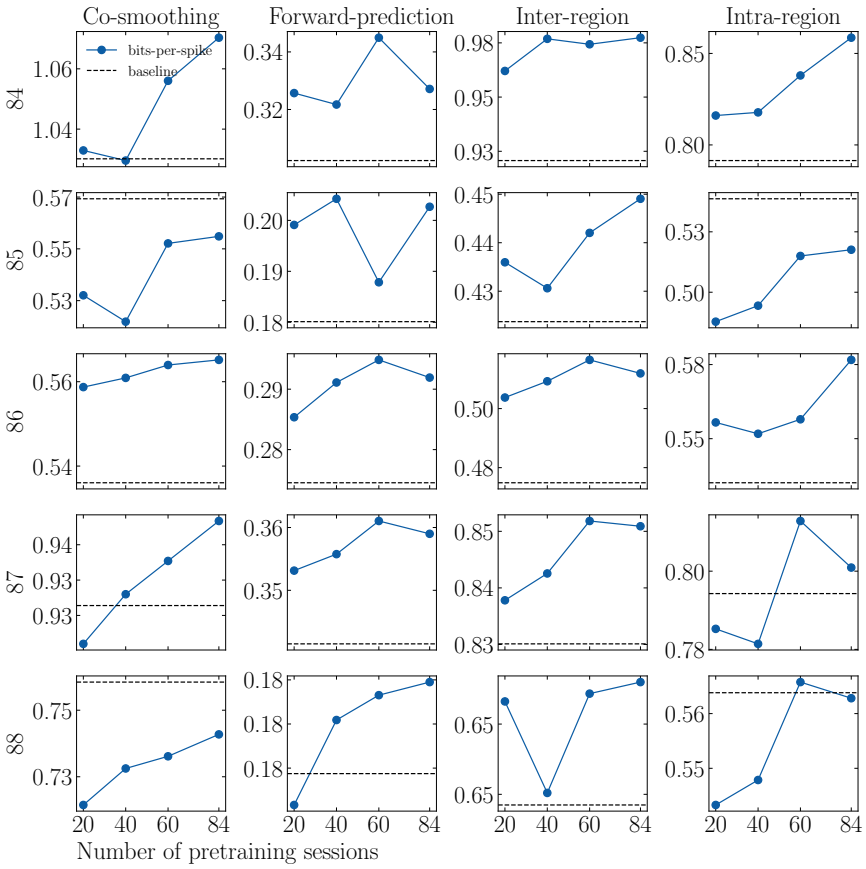


Figure 5.8: **Pretrained models' finetuning performances on each heldout session in** `RS`. Black dashed lines show the baseline models' performances.

5.5.4 Finetuning results from the session-selection procedure

Here, we show the results of the session-selection procedure described in Section 5.3.2. Figure 5.10 shows the single-session finetuning performances on the validation set of each heldout session, sorted from high to low. As the figure shows, there exist large differences between the best and worst single-session finetuning performances, which are more noticeable in Sessions 84, 85, and 88 than in others.

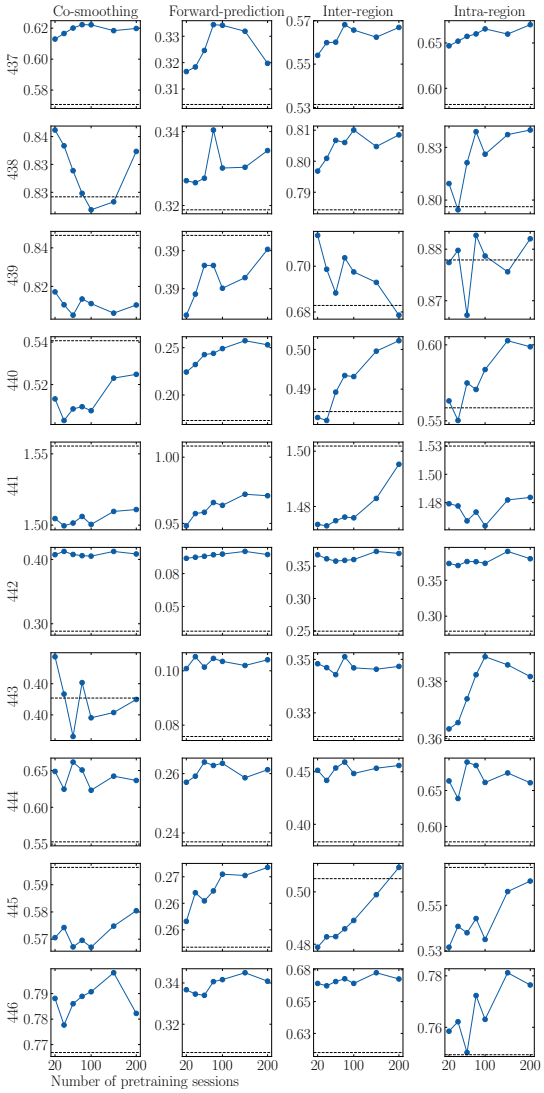


Figure 5.9: **Pretrained models' finetuning performances on each heldout session in** `BWM`. Black dashed lines show the baseline models' performances.

5.5.5 Session-specificity of the rankings

Here, we examine how session-specific the rankings were. Figure 5.11 shows the number of sessions shared in all five top-k ranking of the heldout sessions. The top 20 sessions were highly session-specific: no session appeared in the top five for all held-out sessions, and only three were shared in the top 23. In contrast, rankings became increasingly similar beyond

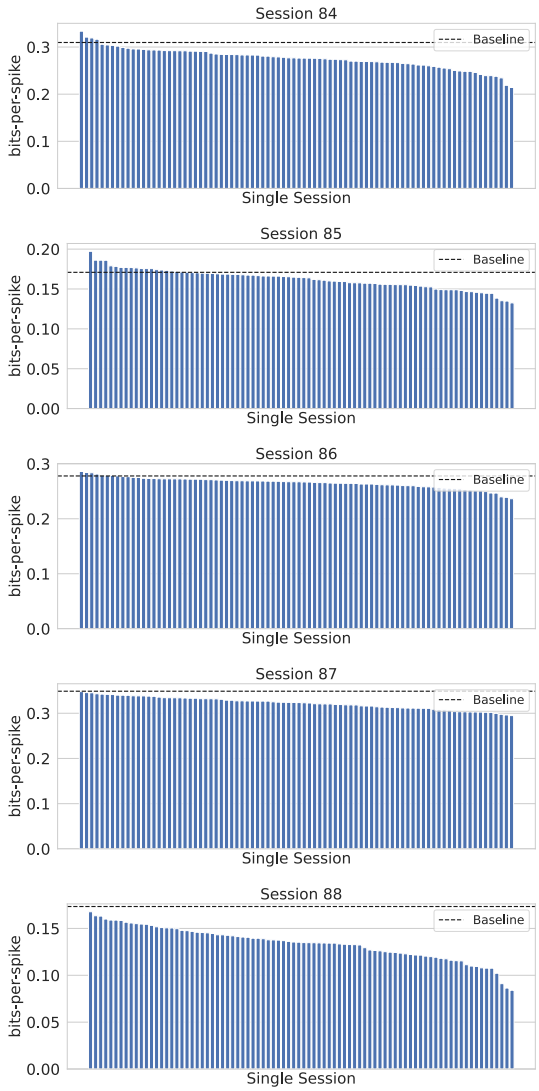


Figure 5.10: **Single-session finetuning performances on the validation set of the heldout sessions from the session-selection procedure.** Black dashed lines show the baseline models' performances.

23 sessions, with 43 sessions shared in all five top 53 rankings. These results suggest that only a small number of top-ranked sessions have a strong impact on performance, while the remaining pretraining sessions are more consistent across held-out sessions and affect model performance less significantly.

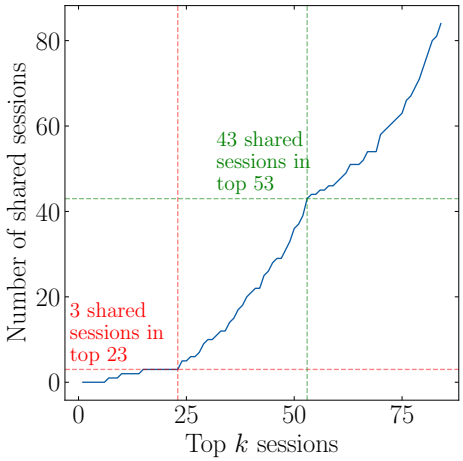


Figure 5.11: Number of shared sessions in the top-k ranking of all five heldout sessions.

5.5.6 Qualitative comparison on forward-prediction performances

Here, we qualitatively compare the forward-prediction results on some example neurons from Session 84. As shown in Figure 5.12, predictions made by the models trained on five top-ranked sessions (green) match the ground truth activity dynamics (blue) much better than those trained by five reverse-ranked sessions (orange). Neurons were randomly selected from the 100 most active ones.

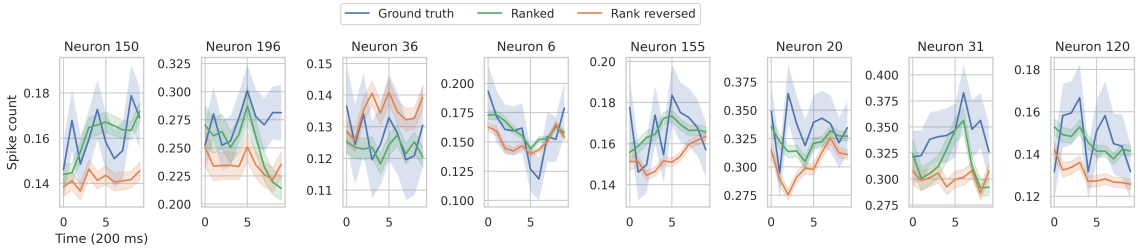


Figure 5.12: **Qualitative comparison of forward-prediction performances between ranked and reverse-ranked models.** Ground truth activities and predictions are averaged over trials. Shading shows standard error of the mean.

Chapter 6

CONCLUSION

In this thesis, I investigated prediction-based self-supervised learning as a principle for learning structured representations in biological and artificial systems. Chapter 3 focused on improving the learned dictionary of sparse coding variational autoencoders (SVAEs), demonstrating that weight normalization improves filter quality to preserve the sparse, bandpass, V1-like representations. Chapter 4 introduced dynamic predictive coding (DPC), a hierarchical model of spatiotemporal prediction in the neocortex that learns space-time receptive fields, temporal response hierarchies, predictive and postdictive motion effects, cue-triggered sequence recall, and higher-order temporal abstractions through spatiotemporal prediction error minimization. Chapter 5 examined neural data transformers trained on multi-session mouse electrophysiology datasets, showing that self-supervised pretraining can improve neural activity prediction, but these gains are limited by data heterogeneity across brain regions and recording sessions.

I conclude by discussing the implications and open questions for the DPC framework and neuro foundation models, the two main focuses of this thesis.

6.1 Temporal Hierarchies, Episodic Memory, and Action-Based Prediction under the DPC Framework

The key to the DPC model's ability to capture lower-level dynamics with relatively stable higher-level response vectors $\mathbf{r}^h$ is the top-down modulation of transition dynamics of entire lower-level state sequences, using the weights $\mathbf{w}$ generated by the higher level. There has been increasing interest in neuroscience in the role of modulatory inputs (*e.g.*, encoding top-down contextual information) in shaping the dynamics of recurrent neural networks in the brain (Mante et al., 2013; Stroud et al., 2018; Masse et al., 2018). The DPC model ascribes an important role to these modulatory inputs in enabling cortical circuits to learn temporal hierarchies. The neural implementation used in this chapter can be seen as top-

down feedback ($\mathbf{w}$) targeting the distal apical dendrites of lower-level pyramidal neurons, thereby changing their gain (Fig 4.1c). Such a mechanism, which has been shown to be possible experimentally (Larkum et al., 2004; Shai et al., 2015; Ferguson and Cardin, 2020; Shine et al., 2021), can also modulate perceptual detection (Takahashi et al., 2016). Note that although we chose to model the top-down influence as multiplicative gain modulation, it would be theoretically equivalent to model it as an additive component or concatenate it as an extra input for prediction (*e.g.*, predicting first-level transitions as $\bar{\mathbf{r}}_t = \mathbf{f}(\mathbf{r}_{t-1}, \mathbf{r}^h)$, where $\mathbf{f}$ is a multi-layer perceptron). However, such an implementation may be less efficient (in terms of the number of parameters required to reach the same level of performance) under certain conditions, compared to a hypernetwork-based implementation (Galanti and Wolf, 2020) such as our implementation based on multiplicative gain modulation.

Recent advances in deep learning (LeCun et al., 2015) have spurred several efforts to learn spatiotemporal hierarchies from sensory data. Lotter et al. developed a deep learning model called "PredNet" for learning a hierarchical predictive coding-inspired model for natural videos (Lotter et al., 2017, 2020). After training, the model was shown to produce a wide range of visual cortical properties and motion illusions. However, in PredNet, higher-level neurons predict lower-level prediction *errors* rather than neural activities or dynamics, making it unclear what the underlying generative model is. It is also unclear if PredNet learns a temporal response hierarchy as found in the cortex. A different model, proposed by Singer et al. (2018) and later extended to hierarchies (Singer et al., 2023), is trained by making higher layers predict lower layer activities: after training, model neurons in different layers displayed different levels of tuning properties and direction selectivity similar to neurons in the dorsal visual pathway. However, similar to the sparse coding and ICA models discussed above for spatiotemporal sequences, the Singer et al. model also requires a sequence of images to be presented as a single input to the network, and the hierarchy of timescales is hard-coded (higher-level neurons predict future lower-level neural activities by receiving a fixed-length chunk of neural activities as input). The above models also do not provide explanations for postdiction or episodic memory and recall.

Many experimental studies have shown an increase in temporal representation stability and response timescales as one goes from lower-order to higher-order areas in the visual

and other parts of the cortex (Murray et al., 2014; Runyan et al., 2017; Siegle et al., 2021; Brunec and Momennejad, 2022; Piasini et al., 2021; Henin et al., 2021). Most computational models have studied this phenomenon through mechanistic rate-based models with parameters based on connectivity data (Chaudhuri et al., 2015; Joglekar et al., 2018) or spiking network models (van Meegen and van Albada, 2021). Kiebel et al. (2008) proposed a model where second-level states generate a single parameter for the first-level Lorenz attractor as the slower "sensory cause" parameter. DPC generalizes this model by assuming higher-level states fully determine the lower-level transition function by predicting the transition dynamics of lower-level states. Under this formulation, temporal hierarchies emerge naturally as a consequence of the neocortex learning from temporally structured data (*e.g.*, stable dynamics in short time windows). This view is consistent with findings that response timescales are functionally dynamic and could expand for cognitive tasks such as working memory (Gao et al., 2020).

Previous normative models of postdiction in visual processing often relate the effect to the concept of Bayesian smoothing (or backward message passing) (Eagleman and Sejnowski, 2000; Rao et al., 2001). We have shown that a trained two-level DPC network with higher-level sequence representations also exhibits postdictive effects without the need for smoothing. In the event of a temporal irregularity (*e.g.*, an unexpected motion reversal), the higher-level state in the DPC network is updated to reflect a new revised input sequence, naturally implementing postdiction through online hierarchical Bayesian filtering (Figs 4.1 and 4.3). Our flash-lag simulation results are consistent with the Bayesian filtering model from Khoei et al. (2017) showing that the flash-lag effect can be produced through an internal model that explicitly represents object velocity. The higher-level sequence representation in the DPC model supports an implicit (and more generalized) representation of velocity and reproduces the same internal dynamics of the "speed" estimate at motion reversal (compare Fig 4.4h with Fig 6 in Khoei et al. (2017)). It is worth noting that the trained DPC network learned to predict no motion (static sequence) for the flashed object even though it was never trained on static object sequences and did not assume a prior of zero speed (Khoei et al., 2017). This emergent property was also seen in PredNet, which learned to predict relatively little motion for a flashed bar stimulus (Lotter et al., 2020).

The higher-level sequence representations of DPC, when combined with an associative memory, support the formation of episodic memories and cue-triggered activity recall (Xu et al., 2012; Gavornik and Bear, 2014; Eagleman and Dragoi, 2012; Ekman et al., 2017; Bang et al., 2018; Lu et al., 2021). The associative memory in our model forms an episodic memory by binding the inferred content representation and dynamics representation from the DPC network during conditioning. When an initial portion of the sequence is presented during testing, the stored episodic memory is retrieved, generating the dynamics component which modulates the lower-level network to enable full recall of the sequence. Though previously considered to only require V1 plasticity (Xu et al., 2012; Gavornik and Bear, 2014), sequence learning is severely impaired in mice with hippocampal damage (Finnie et al., 2021). Coordinated activity between V1 and the hippocampus has also been found in human V1 during recall (Ekman et al., 2022). These experimental results support the involvement of the hippocampus in sequence learning, consistent with our model. Overall, the memory-augmented DPC model offers a highly efficient computational basis for forming and recalling episodic memories (Foster, 2017; Barron et al., 2020), where a single representation of content and transition dynamics from all sensory areas of the neocortex can be bound together as a memory and later retrieved upon receiving partial input.

In our three-level DPC model, the second-level state (under the influence of the current third-level state) predicts the next second-level state only when the first-level prediction errors are larger than an estimated threshold. This can be seen as a neural (and continuous-valued) implementation of "terminal states" in hierarchical hidden Markov models (HHMMs) (Fine et al., 1998). In an HHMM, when a terminal state is reached at a lower level, the corresponding sub-HMM is deemed to be completed and the higher-level state then transitions to the next higher-level state (which activates the next sub-HMM at the lower level). In our hierarchical DPC model, small first-level prediction errors are resolved locally between the first and second level, indicating a continuing sub-sequence. When the error exceeds the threshold, the sub-sequence ends and the second-level transitions are activated. Any second-level prediction errors are resolved between the second and third level through third-level state inference. We used post-hoc estimation of the error threshold after training but future work could attempt to estimate the threshold online in terms of the in-

verse variance or "precision" of prediction errors (Friston, 2005). Additionally, second-level transitions could also correlate with the spatial information in the videos (*e.g.* bouncing only happens when the digit is near the boundary). Models whose second-level states depend on both the previous first- and previous second-level states could learn this type of transition (Linderman et al., 2017).

The DPC model can be extended to action-conditioned prediction and hierarchical planning (see, e.g., (Rao et al., 2024) for initial steps in this direction). There is a growing body of evidence that neural activity in the sensory cortex is predictive of the sensory consequences of an animal's own actions (Keller et al., 2012; Keller and Mrsic-Flogel, 2018; Jordan and Keller, 2020; Fiser et al., 2016; Zmarz and Keller, 2016). These results can be understood in the context of a DPC model in which the transition function at each level is a function of both a state and an action at that level, thereby allowing the hierarchical network to predict the consequences of actions at multiple levels of abstraction (Rao et al., 2024). Such a model allows probabilistic inference to be used not only for perception but also for hierarchical planning, where actions are selected to minimize the sensory prediction errors with respect to preferred goal states. Such a model is consistent with theories of active inference (Friston, 2010) and planning by inference (Attias, 2003; Verma and Rao, 2005, 2006; Botvinick and Toussaint, 2012; Levine, 2018), and opens the door to understanding the neural basis of navigation and planning (Momennejad, 2020; Brunec and Momennejad, 2022; Stachenfeld et al., 2017) as an emergent property of prediction error minimization.

Lastly, it would be an interesting direction to explore the cortical circuits that enable action-based prediction under the DPC framework, in particular the thalamocortical loop (see Shine (2021); Miller-Hansen and Sherman (2022) for the anatomy of the hierarchical structure of the thalamocortical loop). In a recent theory proposed by Rao (2024), the Layer 5–thalamus–Layer 2/3 loop is hypothesized to implement action-based prediction and "action prediction error" computation. A strong piece of supporting evidence was recently discovered by Furutachi et al. (2024): unexpected sensory inputs drive both pulvinar input to V1 and local VIP interneurons in Layer 2/3 of V1. A prediction error response thus requires a cooperative interaction between the thalamus and local cortical circuits. Future theoretical and experimental work could explore how thalamocortical loops implement

action-based modulation in predictive circuits.

6.2 What Will Enable True Scaling Behavior in Neuro Foundation Models?

The scaling results in Chapter 5 leave several open questions for how neuro foundation models should be evaluated and improved. Here, I discuss several challenges that need to be addressed in future work to truly enable scaling behavior in neuro foundation models.

Scaling curves in LLM pretraining often measure the perplexity metric on the test set. For neuro foundation models, a unique challenge is that, under most tokenizer designs, models cannot perform true zero-shot evaluation on data from an unseen individual or session. As discussed in Section 5.1.1, group-level tokenizers require new lightweight encoder—decoder mapping layers, while channel-level tokenizers require initializing new embeddings for channels from unseen datasets. Either design requires gradients from the objective to adapt the new weights or embeddings.

A naive way to solve this issue is to always construct training datasets with samples drawn from all sessions. However, this may quickly become intractable as the dataset sources grow or models need to be tested on new recordings. A pragmatic approach used in our Chapter 5 study is to always finetune the pretrained model on unseen data and report scaling results on the finetuning dataset, using the same metrics as during pretraining. A recent study on contrastive learning of single-neuron representations can perform zero-shot evaluation through a mixture of spatial and temporal attention modules (Arora et al., 2025). Developing foundation models that enable true zero-shot evaluation is an important direction for future research, and would make it possible to better characterize data scaling properties.

Though some downstream tasks concern neural activity predictions, as in Chapter 5, a wide range of other applications requires finetuning the foundation model to objectives such as cursor control, seizure detection, speech decoding, etc. These objectives are sufficiently different from the self-supervised pretraining objective. For foundation models on neural data, evidence for a reliable correlation between pretraining metrics and downstream task metrics is currently limited.

This two-stage setup is similar to foundation models for images finetuned for image

understanding tasks. Models trained with contrastive methods such as CLIP and SigLIP (Radford et al., 2021; Zhai et al., 2023) show considerable performance improvements as the model size increases. Image-only models like the DINO-series (Caron et al., 2021; Siméoni et al., 2025) also show stronger compute scaling than image-text contrastive models on certain tasks (e.g. Figure 2 in (Siméoni et al., 2025)). MAE (He et al., 2022) and AIE (El-Nouby et al., 2024) are trained with generative objectives (pixel prediction) and finetuned to discriminative tasks, resembling a typical neuro foundation model setup. AIE in particular has shown robust dataset and parameter scaling behaviors where the downstream task metric is the top-1 image recognition accuracy (Figure 4 in El-Nouby et al. (2024)). For neuro foundation models, investigating whether improved scaling in generative pretraining leads to downstream task improvements is of utmost importance.

As discussed above, the heterogeneity of neural data presents a unique challenge to neuro foundation models, as it is unknown how well the foundation model can generalize across subjects and sessions. Our results in Chapter 5 show that session-to-session variability significantly impacts the pretraining benefit. However, this study only tests a specific combination of data modality, tokenizer, and architecture design — namely, NDT with a group-level tokenizer trained on spiking data with an autoregressive loss. Future work should systematically investigate the optimal combination of these design choices for scaling.

Additionally, different modalities of neural data recorded under different tasks and species may also affect model scaling behaviors. Concurrent work on motor decoding by Ye et al. (2025) reports that the benefits from pretraining the model on 2000 hours of data are virtually nonexistent when finetuning datasets exceed 100 minutes. However, better scaling behaviors have been observed in image decoding tasks using fMRI data in the work of Banville et al. (2025). Determining the best spatiotemporal resolution for a given recording modality and behavioral context remains an interesting direction.

Model interpretability remains an important open question for neuro foundation models. Besides the applications on decoding-oriented downstream tasks, one promise such models hold is to advance our understanding of large-scale neuroscience data as a data-analysis tool (Stringer and Pachitariu, 2024). Current efforts on interpreting models mainly rely on post-hoc dimensionality reduction on learned embeddings to correlate them with known labels

such as brain regions, cell types, cortical hierarchies, etc. Recent work took a different approach and treated neuro foundation models as in-silico hypothesis testing machines: Wang et al. (2025) trained a foundation model on large-scale mouse visual cortex recordings that generalizes to new mice and predicts responses to out-of-distribution stimulus domains. Unlike most of the models discussed, their model takes the video stimuli the mice saw as input rather than neural recordings, and directly predicts neural activity. This model can generalize to new mice with minimal training and successfully predicted responses across various new stimulus domains, such as coherent motion and noise patterns. Exploring how foundation models could not only be validated by existing neurophysiological data, but also generate testable predictions at the circuit and neural-dynamics level, is an important direction for using these tools to aid scientific discovery.

BIBLIOGRAPHY

Vinam Arora, Divyansha Lachi, Ian Jarratt Knight, Mehdi Azabou, Blake Aaron Richards, Cole Lincoln Hurwitz, Josh Siegle, and Eva L. Dyer. Know thyself by knowing others: Learning neuron identity from population context. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://openreview.net/forum?id=zt3RKc6VBp.

Hagai Attias. Planning by Probabilistic Inference. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, volume R4 of *Proceedings of Machine Learning Research*, pages 9–16. PMLR, January 2003. URL http://proceedings.mlr.press/r4/attias03a.html.

Mehdi Azabou, Vinam Arora, Venkataramana Ganesh, Ximeng Mao, Santosh B. Nachimuthu, Michael Jacob Mendelson, Blake Aaron Richards, Matthew G. Perich, Guillaume Lajoie, and Eva L. Dyer. A Unified, Scalable Framework for Neural Population Decoding. In *Advances in Neural Information Processing Systems*, November 2023. doi: 10.52202/075280-1948. URL https://openreview.net/forum?id=sw2Y0sirtM.

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 *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/953390c834451505703c9da45de634d8-Abstract-Conference.html.

J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer Normalization. *arXiv*, 2016. doi: 10.48550/arXiv.1607.06450. URL http://arxiv.org/abs/1607.06450.

Ji Won Bang, Yuka Sasaki, Takeo Watanabe, and Dobromir Rahnev. Feature-Specific Awake Reactivation in Human V1 after Visual Training. *Journal of Neuroscience*, 38 (45):9648–9657, November 2018. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.0884-18.2018. URL https://www.jneurosci.org/content/38/45/9648.

Hubert Banville, Yohann Benchetrit, Stéphane d'Ascoli, Jérémy Rapin, and Jean-Rémi King. Scaling laws for decoding images from brain activity. *arXiv*, 2025. doi: 10.48550/arXiv.2501.15322. URL http://arxiv.org/abs/2501.15322.

Horace B. Barlow. Possible principles underlying the transformations of sensory messages. In Walter A. Rosenblith, editor, *Sensory Communication*, pages 217–234. MIT

Press, Cambridge, MA, 1961. URL https://academic.oup.com/mit-press-scholarship-online/book/20714/chapter/180090664.

Konstantinos Barmpas, Na Lee, Dimitrios Chalatsis, William Raftery, Yannis Panagakis, Dimitrios A. Adamos, Nikolaos Laskaris, Alexandros Koliousis, Dario Farina, and Stefanos Zafeiriou. NeuroRVQ: Multi-scale biosignal tokenization for generative foundation models. *arXiv*, 2025. doi: 10.48550/arXiv.2510.13068. URL https://arxiv.org/abs/2510.13068.

Helen C. Barron, Ryszard Auksztulewicz, and Karl Friston. Prediction and memory: A predictive coding account. *Progress in Neurobiology*, 192:101821, September 2020. ISSN 0301-0082. doi: 10.1016/j.pneurobio.2020.101821. URL https://www.sciencedirect.com/science/article/pii/S0301008220300769.

André M. Bastos, Mikael Lundqvist, Ayan S. Waite, Nancy Kopell, and Earl K. Miller. Layer and rhythm specificity for predictive routing. *Proceedings of the National Academy of Sciences*, November 2020. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.2014868117. URL https://www.pnas.org/content/early/2020/11/20/2014868117. Publisher: National Academy of Sciences Section: Biological Sciences.

Curtis C. Bell, Victor Z. Han, Yoshiko Sugawara, and Kirsty Grant. Synaptic plasticity in a cerebellum-like structure depends on temporal order. *Nature*, 387(6630):278–281, May 1997. ISSN 1476-4687. doi: 10.1038/387278a0. URL https://www.nature.com/articles/387278a0. Number: 6630 Publisher: Nature Publishing Group.

Pietro Berkes and Laurenz Wiskott. Slow feature analysis yields a rich repertoire of complex cell properties. *Journal of Vision*, 5(6):9, July 2005. ISSN 1534-7362. doi: 10.1167/5.6.9. URL https://doi.org/10.1167/5.6.9.

Rafal Bogacz. A tutorial on the free-energy framework for modelling perception and learning. *Journal of Mathematical Psychology*, 76:198–211, February 2017. ISSN 00222496. doi: 10.1016/j.jmp.2015.11.003. URL https://linkinghub.elsevier.com/retrieve/pii/S0022249615000759.

Sander E. Bosch, Janneke F. M. Jehee, Guillén Fernández, and Christian F. Doeller. Reinstatement of Associative Memories in Early Visual Cortex Is Signaled by the Hippocampus. *Journal of Neuroscience*, 34(22):7493–7500, May 2014. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.0805-14.2014. URL https://www.jneurosci.org/content/34/22/7493.

Matthew Botvinick and Marc Toussaint. Planning as inference. *Trends in Cognitive Sciences*, 16(10):485–488, October 2012. ISSN 1364-6613. doi: 10.1016/j.tics.2012.08.006. URL https://www.sciencedirect.com/science/article/pii/S1364661312001957.

Samuel R. Bowman, L. Vilnis, Oriol Vinyals, Andrew M. Dai, R. Józefowicz, and S. Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 2016. doi: 10.18653/v1/K16-1002. URL https://aclanthology.org/K16-1002/.

Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. URL https://web.stanford.edu/~boyd/cvxbook/.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Iva K. Brunec and Ida Momennejad. Predictive Representations in Hippocampal and Prefrontal Hierarchies. *Journal of Neuroscience*, 42(2):299–312, January 2022. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.1327-21.2021. URL https://www.jneurosci.org/content/42/2/299.

Neil Burgess, Eleanor A Maguire, and John O'Keefe. The Human Hippocampus and Spatial and Episodic Memory. *Neuron*, 35(4):625–641, August 2002. ISSN 0896-6273. doi: 10.1016/S0896-6273(02)00830-9. URL https://www.sciencedirect.com/science/article/pii/S0896627302008309.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, 2021. doi: 10.1109/ICCV48922.2021.00951. URL https://ieeexplore.ieee.org/document/9709990. ISSN: 2380-7504.

Rishidev Chaudhuri, Kenneth Knoblauch, Marie-Alice Gariel, Henry Kennedy, and Xiao-Jing Wang. A Large-Scale Circuit Mechanism for Hierarchical Dynamical Processing in the Primate Cortex. *Neuron*, 88(2):419–431, October 2015. ISSN 0896-6273. doi: 10.1016/j.neuron.2015.09.008. URL https://www.sciencedirect.com/science/article/pii/S0896627315007655.

Y. Chen, Dylan M. Paiton, and B. Olshausen. The sparse manifold transform. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://papers.nips.cc/paper/8251-the-sparse-manifold-transform.

Hannah Choi, Anitha Pasupathy, and Eric Shea-Brown. Predictive Coding in Area V4: Dynamic Shape Discrimination under Partial Occlusion. *Neural Computation*, 30(5):1209–1257, May 2018. ISSN 0899-7667. doi: 10.1162/neco_a_01072. URL https://doi.org/10.1162/neco_a_01072.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling Instruction-Finetuned Language Models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. ISSN 1533-7928. URL http://jmlr.org/papers/v25/23-0870.html.

D. A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). In *International Conference on Learning Representations*, 2016. URL http://arxiv.org/abs/1511.07289.

Peter Dayan, Geoffrey E. Hinton, Radford M. Neal, and Richard S. Zemel. The Helmholtz Machine. *Neural Computation*, 7(5):889–904, September 1995. ISSN 0899-7667. doi: 10.1162/neco.1995.7.5.889. URL https://doi.org/10.1162/neco.1995.7.5.889.

Gregory C. DeAngelis, Izumi Ohzawa, and Ralph D. Freeman. Receptive-field dynamics in the central visual pathways. *Trends in Neurosciences*, 18(10):451–458, October 1995. ISSN 01662236. doi: 10.1016/0166-2236(95)94496-R. URL https://linkinghub.elsevier.com/retrieve/pii/016622369594496R.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Dawei W Dong and Joseph J Atick. Statistics of natural time-varying images. *Network: Computation in Neural Systems*, 6(3):345–358, January 1995a. ISSN 0954-898X. doi: 10.1088/0954-898X_6_3_003. URL https://doi.org/10.1088/0954-898X_6_3_003.

Dawei W Dong and Joseph J Atick. Temporal decorrelation: a theory of lagged and nonlagged responses in the lateral geniculate nucleus. *Network: Computation in Neural Systems*, 6(2):159–178, 1995b. ISSN 0954-898X. doi: 10.1088/0954-898X_6_2_003.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, October 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Laura N. Driscoll, Lea Duncker, and Christopher D. Harvey. Representational drift: Emerging theories for continual learning and experimental future directions. *Current Opinion in Neurobiology*, 76:102609, October 2022. ISSN 0959-4388. doi: 10.1016/j.conb.2022.102609. URL https://www.sciencedirect.com/science/article/pii/S0959438822001039.

David M. Eagleman and Terrence J. Sejnowski. Motion Integration and Postdiction in Visual Awareness. *Science*, 287(5460):2036–2038, March 2000. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.287.5460.2036. URL https://www.science.org/doi/10.1126/science.287.5460.2036.

Sarah L. Eagleman and Valentin Dragoi. Image sequence reactivation in awake V4 networks. *Proceedings of the National Academy of Sciences*, 109(47):19450–19455, November 2012. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1212059109. URL https://www.pnas.org/content/109/47/19450.

Matthias Ekman, Peter Kok, and Floris P. de Lange. Time-compressed preplay of anticipated events in human primary visual cortex. *Nature Communications*, 8(1):15276, May 2017. ISSN 2041-1723. doi: 10.1038/ncomms15276. URL https://www.nature.com/articles/ncomms15276.

Matthias Ekman, Giulia Gennari, and Floris P. de Lange. Probabilistic forward replay of anticipated stimulus sequences in human primary visual cortex and hippocampus. *bioRxiv*, January 2022. doi: 10.1101/2022.01.26.477907. URL https://www.biorxiv.org/content/10.1101/2022.01.26.477907v1.

Alaaeldin El-Nouby, Michal Klein, Shuangfei Zhai, Miguel Ángel Bautista, Vaishaal Shankar, Alexander T. Toshev, Joshua M. Susskind, and Armand Joulin. Scalable pre-training of large autoregressive image models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 12371–12384. PMLR, July 2024. URL https://proceedings.mlr.press/v235/el-nouby24a.html.

Marc O. Ernst and Massimiliano Di Luca. Multisensory Perception: From Integration to Remapping. In *Sensory Cue Integration*. Oxford University Press, 2011. ISBN 978-0-19-538724-7. doi: 10.1093/acprof:oso/9780195387247.003.0012. URL https://oxford.universitypressscholarship.com/10.1093/acprof:oso/9780195387247.001.0001/acprof-9780195387247-chapter-12.

D. J. Felleman and D. C. Van Essen. Distributed Hierarchical Processing in the Primate Cerebral Cortex. *Cerebral Cortex*, 1(1):1–47, January 1991. ISSN 1047-3211, 1460-2199. doi: 10.1093/cercor/1.1.1. URL https://academic.oup.com/cercor/article-lookup/doi/10.1093/cercor/1.1.1.

Katie A. Ferguson and Jessica A. Cardin. Mechanisms underlying gain modulation in the cortex. *Nature Reviews Neuroscience*, 21(2):80–92, February 2020. ISSN 1471-0048. doi: 10.1038/s41583-019-0253-y. URL https://www.nature.com/articles/s41583-019-0253-y.

Shai Fine, Yoram Singer, and Naftali Tishby. The Hierarchical Hidden Markov Model: Analysis and Applications. *Machine Learning*, 32(1):41–62, July 1998. ISSN 1573-0565. doi: 10.1023/A:1007469218079. URL https://doi.org/10.1023/A:1007469218079.

Peter S. B. Finnie, Robert W. Komorowski, and Mark F. Bear. The spatiotemporal organization of experience dictates hippocampal involvement in primary visual cortical plasticity. *Current Biology*, 31(18):3996–4008.e6, September 2021. ISSN 0960-9822. doi: 10.1016/j.cub.2021.06.079. URL https://www.cell.com/current-biology/abstract/S0960-9822(21)00908-8.

Aris Fiser, David Mahringer, Hassana K. Oyibo, Anders V. Petersen, Marcus Leinweber, and Georg B. Keller. Experience-dependent spatial expectations in mouse visual cortex. *Nature Neuroscience*, 19(12):1658–1664, December 2016. ISSN 1546-1726. doi: 10.1038/nn.4385. URL https://www.nature.com/articles/nn.4385.

David J. Foster. Replay Comes of Age. *Annual Review of Neuroscience*, 40(1):581–602, 2017. doi: 10.1146/annurev-neuro-072116-031538. URL https://doi.org/10.1146/annurev-neuro-072116-031538.

Winrich A. Freiwald and Doris Y. Tsao. Functional Compartmentalization and Viewpoint Generalization Within the Macaque Face-Processing System. *Science*, 330 (6005):845–851, November 2010. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1194908. URL https://science.sciencemag.org/content/330/6005/845. Publisher: American Association for the Advancement of Science Section: Report.

Karl Friston. A theory of cortical responses. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 360(1456):815–836, April 2005. doi: 10.1098/rstb.2005.1622. URL https://royalsocietypublishing.org/doi/full/10.1098/rstb.2005.1622. Publisher: Royal Society.

Karl Friston. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2):127–138, February 2010. ISSN 1471-003X, 1471-0048. doi: 10.1038/nrn2787. URL http://www.nature.com/articles/nrn2787.

Karl Friston and Stefan Kiebel. Predictive coding under the free-energy principle. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1521): 1211–1221, May 2009. ISSN 0962-8436. doi: 10.1098/rstb.2008.0300. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2666703/.

Shohei Furutachi, Alexis D. Franklin, Andreea M. Aldea, Thomas D. Mrsic-Flogel, and Sonja B. Hofer. Cooperative thalamocortical circuit mechanism for sensory prediction errors. *Nature*, 633(8029):398–406, September 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07851-w. URL https://www.nature.com/articles/s41586-024-07851-w.

Tomer Galanti and Lior Wolf. On the Modularity of Hypernetworks. In *Advances in Neural Information Processing Systems*, volume 33, pages 10409–10419. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/75c58d36157505a600e0695ed0b3a22d-Abstract.html.

Richard Gao, Ruud L van den Brink, Thomas Pfeffer, and Bradley Voytek. Neuronal timescales are functionally dynamic and shaped by cortical microarchitecture. *eLife*, 9:e61277, November 2020. ISSN 2050-084X. doi: 10.7554/eLife.61277. URL https://doi.org/10.7554/eLife.61277.

Jeffrey P. Gavornik and Mark F. Bear. Learned spatiotemporal sequence recognition and prediction in primary visual cortex. *Nature Neuroscience*, 17(5):732–737, May 2014. ISSN 1546-1726. doi: 10.1038/nn.3683. URL https://www.nature.com/articles/nn.3683.

Victor Geadah, Gabriel Barello, Daniel Greenidge, Adam S. Charles, and Jonathan W. Pillow. Sparse-Coding Variational Autoencoders. *Neural Computation*, 36(12):2571–2601, 2024. doi: 10.1162/neco_a_01715. URL https://direct.mit.edu/neco/article/36/12/2571/124821/Sparse-Coding-Variational-Autoencoders.

Hagar Gelbard-Sagiv, Roy Mukamel, Michal Harel, Rafael Malach, and Itzhak Fried. Internally Generated Reactivation of Single Neurons in Human Hippocampus During Free Recall. *Science*, 322(5898):96–101, October 2008. doi: 10.1126/science.1164685. URL https://www.science.org/doi/10.1126/science.1164685.

Dileep George and Jeff Hawkins. Towards a Mathematical Theory of Cortical Micro-circuits. *PLOS Computational Biology*, 5(10):e1000532, October 2009. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1000532. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000532.

Colleen J. Gillon, Jason E. Pina, Jérôme A. Lecoq, Ruweida Ahmed, Yazan Billeh, Shiella Caldejon, Peter Groblewski, Tim M. Henley, India Kato, Eric Lee, Jennifer Luviano, Kyla Mace, Chelsea Nayan, Thuyanh Nguyen, Kat North, Jed Perkins, Sam

Seid, Matthew Valley, Ali Williford, Yoshua Bengio, Timothy P. Lillicrap, Blake A. Richards, and Joel Zylberberg. Learning from unexpected events in the neocortical microcircuit. *bioRxiv*, January 2021. doi: 10.1101/2021.01.15.426915. URL https://www.biorxiv.org/content/10.1101/2021.01.15.426915v1.

K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning*, pages 399–406. Omnipress, 2010. URL https://icml.cc/Conferences/2010/papers/449.pdf.

Richard Langton Gregory, Hugh Christopher Longuet-Higgins, and N. S. Sutherland. Perceptions as hypotheses. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 290(1038):181–197, July 1980. doi: 10.1098/rstb.1980.0090. URL https://royalsocietypublishing.org/doi/10.1098/rstb.1980.0090. Publisher: Royal Society.

David Ha, Andrew M. Dai, and Quoc V. Le. HyperNetworks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkpACe1lx.

Jordan P. Hamm, Yuriy Shymkiv, Shuting Han, Weijian Yang, and Rafael Yuste. Cortical ensembles selective for context. *Proceedings of the National Academy of Sciences*, 118(14), April 2021. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.2026179118. URL https://www.pnas.org/content/118/14/e2026179118. Publisher: National Academy of Sciences Section: Biological Sciences.

Kenneth D. Harris and Alexander Thiele. Cortical state and attention. *Nature Reviews Neuroscience*, 12(9):509–523, September 2011. ISSN 1471-0048. doi: 10.1038/nrn3084. URL https://www.nature.com/articles/nrn3084.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988, 2022. doi: 10.1109/CVPR52688.2022.01553. URL https://ieeexplore.ieee.org/document/9879206. ISSN: 2575-7075.

Simon Henin, Nicholas B. Turk-Browne, Daniel Friedman, Anli Liu, Patricia Dugan, Adeen Flinker, Werner Doyle, Orrin Devinsky, and Lucia Melloni. Learning hierarchical sequence representations across human cortex and hippocampus. *Science Advances*, 7, February 2021. doi: 10.1126/sciadv.abc4530. URL https://www.science.org/doi/abs/10.1126/sciadv.abc4530.

I. Higgins, Loïc Matthey, A. Pal, C. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Nicholas C. Hindy, Felicia Y. Ng, and Nicholas B. Turk-Browne. Linking pattern completion in the hippocampus to predictive coding in visual cortex. *Nature Neuroscience*, 19(5):665–667, May 2016. ISSN 1546-1726. doi: 10.1038/nn.4284. URL https://www.nature.com/articles/nn.4284. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 5 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Learning and memory;Psychology;Visual system Subject_term_id: learning-and-memory;psychology;visual-system.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, volume 35, pages 30016–30030, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/c1e2faff6f588870935f114ebe04a3e5-Abstract-Conference.html.

Hinze Hogendoorn. Motion Extrapolation in Visual Processing: Lessons from 25 Years of Flash-Lag Debate. *Journal of Neuroscience*, 40(30):5698–5705, July 2020. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.0275-20.2020. URL https://www.jneurosci.org/content/40/30/5698.

Hinze Hogendoorn. Perception in real-time: predicting the present, reconstructing the past. *Trends in Cognitive Sciences*, 26(2):128–141, February 2022. ISSN 1364-6613. doi: 10.1016/j.tics.2021.11.003. URL https://www.sciencedirect.com/science/article/pii/S1364661321002886.

Hinze Hogendoorn, Thomas A. Carlson, and Frans A. J. Verstraten. Interpolation and extrapolation on the path of apparent motion. *Vision Research*, 48(7):872–881, March 2008. ISSN 0042-6989. doi: 10.1016/j.visres.2007.12.019. URL https://www.sciencedirect.com/science/article/pii/S0042698907005779.

Yanping Huang and Rajesh P. N. Rao. Predictive coding. *WIREs Cognitive Science*, 2(5):580–593, March 2011. ISSN 1939-5086. doi: 10.1002/wcs.142. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/wcs.142.

D. H. Hubel and T. N. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, 148(3):574–591, October 1959. ISSN 0022-3751.

doi: 10.1113/jphysiol.1959.sp006308. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1363130/.

Sukjun Hwang, Brandon Wang, and Albert Gu. Dynamic chunking for end-to-end hierarchical sequence modeling. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ZbfLR9NbNF.

International Brain Laboratory, Dora Angelaki, Brandon Benson, et al. A brain-wide map of neural activity during complex behaviour. *Nature*, 645(8079):177–191, September 2025. doi: 10.1038/s41586-025-09235-0. URL https://www.nature.com/articles/s41586-025-09235-0.

Elias B Issa, Charles F Cadieu, and James J DiCarlo. Neural dynamics at successive stages of the ventral visual stream are consistent with hierarchical error signals. *eLife*, 7:e42870, November 2018. ISSN 2050-084X. doi: 10.7554/eLife.42870. URL https://doi.org/10.7554/eLife.42870. Publisher: eLife Sciences Publications, Ltd.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J. Henaff, Matthew Botvinick, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver IO: A General Architecture for Structured Inputs & Outputs. In *International Conference on Learning Representations*, October 2022. URL https://openreview.net/forum?id=fILj7WpI-g.

Linxing Preston Jiang and Luciano de la Iglesia. Improved Training of Sparse Coding Variational Autoencoder via Weight Normalization. *arXiv*, 2021. doi: 10.48550/arXiv.2101.09453. URL https://arxiv.org/abs/2101.09453.

Linxing Preston Jiang and Rajesh P. N. Rao. Predictive Coding Theories of Cortical Function. *Oxford Research Encyclopedia of Neuroscience*, November 2022. doi: 10.1093/acrefore/9780190264086.013.328. URL https://oxfordre.com/neuroscience/view/10.1093/acrefore/9780190264086.001.0001/acrefore-9780190264086-e-328.

Linxing Preston Jiang and Rajesh P. N. Rao. Dynamic Predictive Coding Explains Both Prediction and Postdiction in Visual Motion Perception. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 45(45), 2023. URL https://escholarship.org/uc/item/6r74j9k1.

Linxing Preston Jiang and Rajesh P. N. Rao. Dynamic predictive coding: A model of hierarchical sequence learning and prediction in the neocortex. *PLOS Computational Biology*, 20(2):e1011801, 2024. doi: 10.1371/journal.pcbi.1011801. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1011801.

Linxing Preston Jiang, Shirui Chen, Emmanuel Tanumihardja, Xiaochuang Han, Weijia Shi, Eric Shea-Brown, and Rajesh P. N. Rao. Data Heterogeneity Limits the Scaling Effect of Pretraining in Neural Data Transformers. In *COLM 2025 Workshop on LLM for Scientific Discovery*, July 2025a. URL https://openreview.net/forum?id=WiECjcxNw4.

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

Weibang Jiang, Yansen Wang, Bao-liang Lu, and Dongsheng Li. NeuroLM: A universal multi-task foundation model for bridging the gap between language and EEG signals. In *International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=Io9yFt7XH7.

Madhura R. Joglekar, Jorge F. Mejias, Guangyu Robert Yang, and Xiao-Jing Wang. Inter-areal Balanced Amplification Enhances Signal Propagation in a Large-Scale Circuit Model of the Primate Cortex. *Neuron*, 98(1):222–234.e8, April 2018. ISSN 0896-6273. doi: 10.1016/j.neuron.2018.02.031. URL https://www.sciencedirect.com/science/article/pii/S0896627318301521.

Rebecca Jordan and Georg B. Keller. Opposing Influence of Top-down and Bottom-up Input on Excitatory Layer 2/3 Neurons in Mouse Primary Visual Cortex. *Neuron*, 108 (6):1194–1206.e5, December 2020. ISSN 08966273. doi: 10.1016/j.neuron.2020.09.024. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627320307480.

James J. Jun, Nicholas A. Steinmetz, Joshua H. Siegle, Daniel J. Denman, Marius Bauza, Brian Barbarits, Albert K. Lee, Costas A. Anastassiou, Alexandru Andrei, Çağatay Aydın, Mladen Barbic, Timothy J. Blanche, Vincent Bonin, João Couto, Barundeb Dutta, Sergey L. Gratiy, Diego A. Gutnisky, Michael Häusser, Bill Karsh, Peter Ledochowitsch, Carolina Mora Lopez, Catalin Mitelut, Silke Musa, Michael Okun, Marius Pachitariu, Jan Putzeys, P. Dylan Rich, Cyrille Rossant, Wei-lung Sun, Karel Svoboda, Matteo Carandini, Kenneth D. Harris, Christof Koch, John O'Keefe, and Timothy D. Harris. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551(7679):232–236, 2017. ISSN 1476-4687. doi: 10.1038/nature24636. URL https://www.nature.com/articles/nature24636.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling Laws for Neural Language Models. *arXiv*, January 2020. doi: 10.48550/arXiv.2001.08361. URL http://arxiv.org/abs/2001.08361.

Christoph Kayser, Wolfgang Einhäuser, Olaf Dümmer, Peter König, and Konrad Körding. Extracting Slow Subspaces from Natural Videos Leads to Complex Cells. In *International Conference on Artificial Neural Networks*, Lecture Notes in Computer Science, pages 1075–1080, 2001. doi: 10.1007/3-540-44668-0_149. URL https://doi.org/10.1007/3-540-44668-0_149.

Georg B. Keller and Thomas D. Mrsic-Flogel. Predictive Processing: A Canonical Cortical Computation. *Neuron*, 100(2):424–435, October 2018. ISSN 08966273. doi: 10.1016/j.neuron.2018.10.003. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627318308572.

Georg B. Keller, Tobias Bonhoeffer, and Mark Hübener. Sensorimotor Mismatch Signals in Primary Visual Cortex of the Behaving Mouse. *Neuron*, 74(5):809–815, June 2012. ISSN 08966273. doi: 10.1016/j.neuron.2012.03.040. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627312003844.

Mina A. Khoei, Guillaume S. Masson, and Laurent U. Perrinet. The Flash-Lag Effect as a Motion-Based Predictive Shift. *PLOS Computational Biology*, 13(1):e1005068, January 2017. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1005068. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005068.

Stefan J. Kiebel, Jean Daunizeau, and Karl J. Friston. A Hierarchy of Time-Scales and the Brain. *PLOS Computational Biology*, 4(11):e1000209, November 2008. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1000209. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000209. Publisher: Public Library of Science.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations*, 2014. URL https://iclr.cc/archive/2014/conference-proceedings/.

Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://papers.nips.cc/paper/2016/hash/ddeebdeefdb7e7e7a697e1c3e3d8ef54-Abstract.html.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, volume System Demonstrations, pages 66–71. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012/.

International Brain Laboratory, Valeria Aguillon-Rodriguez, Dora Angelaki, Hannah Bayer, Niccolo Bonacchi, Matteo Carandini, Fanny Cazettes, Gaelle Chapuis, Anne K Churchland, Yang Dan, Eric Dewitt, Mayo Faulkner, Hamish Forrest, Laura Haetzel, Michael Häusser, Sonja B Hofer, Fei Hu, Anup Khanal, Christopher Krasniak, Ines Laranjeira, Zachary F Mainen, Guido Meijer, Nathaniel J Miska, Thomas D Mrsic-Flogel, Masayoshi Murakami, Jean-Paul Noel, Alejandro Pan-Vazquez, Cyrille Rossant, Joshua Sanders, Karolina Socha, Rebecca Terry, Anne E Urai, Hernando Vergara, Miles Wells, Christian J Wilson, Ilana B Witten, Lauren E Wool, and Anthony M Zador. Standardized and reproducible measurement of decision-making in mice. *eLife*, 10:e63711, May 2021. ISSN 2050-084X. doi: 10.7554/eLife.63711. URL https://doi.org/10.7554/eLife.63711.

International Brain Laboratory, Kush Banga, Julius Benson, Jai Bhagat, Dan Biderman, Daniel Birman, Niccolò Bonacchi, Sebastian A. Bruijns, Kelly Buchanan, Robert AA Campbell, Matteo Carandini, Gaëlle A. Chapuis, Anne K. Churchland, M. Felicia Davatolhagh, Hyun Dong Lee, Mayo Faulkner, Berk Gerçek, Fei Hu, Julia Huntenburg, Cole Hurwitz, Anup Khanal, Christopher Krasniak, Christopher Langfield, Petrina Lau, Nancy Mackenzie, Guido T. Meijer, Nathaniel J. Miska, Zeinab Mohammadi, Jean-Paul Noel, Liam Paninski, Alejandro Pan-Vazquez, Cyrille Rossant, Noam Roth, Michael Schartner, Karolina Socha, Nicholas A. Steinmetz, Karel Svoboda, Marsa Taheri, Anne E. Urai, Shuqi Wang, Miles Wells, Steven J. West, Matthew R. Whiteway, Olivier Winter, Ilana B. Witten, and Yizi Zhang. Reproducibility of in vivo electrophysiological measurements in mice. *eLife*, 13, May 2025. doi: 10.7554/eLife.100840.3. URL https://elifesciences.org/articles/100840.

Matthew E. Larkum, Walter Senn, and Hans-R. Lüscher. Top-down Dendritic Input Increases the Gain of Layer 5 Pyramidal Neurons. *Cerebral Cortex*, 14(10):1059–1070, October 2004. ISSN 1047-3211. doi: 10.1093/cercor/bhh065. URL https://doi.org/10.1093/cercor/bhh065.

Trung Le and Eli Shlizerman. STNDT: Modeling Neural Population Activity with Spatiotemporal Transformers. In *Advances in Neural Information Processing Systems*, volume 35, pages 17926–17939, December 2022. doi: 10.52202/068431-1303. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/72163d1c3c1726f1c29157d06e9e93c1-Abstract-Conference.html.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521 (7553):436–444, May 2015. ISSN 1476-4687. doi: 10.1038/nature14539. URL https://www.nature.com/articles/nature14539.

H. Lee, A. Battle, R. Raina, and A. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*, volume 19, pages 801–808, 2006. URL https://papers.nips.cc/paper/2979-efficient-sparse-coding-algorithms.

Sergey Levine. Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review. *arXiv*, May 2018. doi: 10.48550/arXiv.1805.00909. URL http://arxiv.org/abs/1805.00909.

Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 914–922. PMLR, April 2017. URL https://proceedings.mlr.press/v54/linderman17a.html.

Alisa Liu, Jonathan Hayase, Valentin Hofmann, Sewoong Oh, Noah A. Smith, and Yejin Choi. SuperBPE: Space travel for language models. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=lcDRvffeNP#discussion.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, September 2018. URL https://openreview.net/forum?id=Bkg6RiCqY7.

William Lotter, Gabriel Kreiman, and David D. Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ewdt9xe.

William Lotter, Gabriel Kreiman, and David Cox. A neural network trained for prediction mimics diverse features of biological neurons and perception. *Nature Machine Intelligence*, 2(4):210–219, April 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-0170-9. URL http://www.nature.com/articles/s42256-020-0170-9.

Junshi Lu, Lu Luo, Qian Wang, Fang Fang, and Nihong Chen. Cue-triggered activity replay in human early visual cortex. *Science China Life Sciences*, 64(1):144–151, January 2021. ISSN 1869-1889. doi: 10.1007/s11427-020-1726-5. URL https://doi.org/10.1007/s11427-020-1726-5.

Jingying Ma, Feng Wu, Qika Lin, Yucheng Xing, Chenyu Liu, Ziyu Jia, and Mengling Feng. CodeBrain: Bridging decoupled tokenizer and multi-scale architecture for EEG foundation model. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=msJgEkjwh5.

Joseph R. Manns and Howard Eichenbaum. Evolution of declarative memory. *Hippocampus*, 16(9):795–808, 2006. ISSN 1098-1063. doi: 10.1002/hipo.20205. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/hipo.20205.

Valerio Mante, David Sussillo, Krishna V. Shenoy, and William T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):

78–84, November 2013. ISSN 1476-4687. doi: 10.1038/nature12742. URL https://www.nature.com/articles/nature12742.

N. T. Markov, J. Vezoli, P. Chameau, A. Falchier, R. Quilodran, C. Huissoud, C. Lamy, P. Misery, P. Giroud, S. Ullman, P. Barone, C. Dehay, K. Knoblauch, and H. Kennedy. Anatomy of hierarchy: Feedforward and feedback pathways in macaque visual cortex. *Journal of Comparative Neurology*, 522(1):225–259, 2014. doi: 10.1002/cne.23458.

Nicolas Y. Masse, Gregory D. Grant, and David J. Freedman. Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization. *Proceedings of the National Academy of Sciences*, 115(44):E10467–E10475, October 2018. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1803839115. URL http://www.pnas.org/lookup/doi/10.1073/pnas.1803839115.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=8ishA3LxN8.

Andrew J. Miller-Hansen and S. Murray Sherman. Conserved patterns of functional organization between cortex and thalamus in mice. *Proceedings of the National Academy of Sciences*, 119(21):e2201481119, May 2022. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.2201481119. URL https://www.pnas.org/doi/10.1073/pnas.2201481119.

Mortimer Mishkin, Leslie G. Ungerleider, and Kathleen A. Macko. Object vision and spatial vision: two cortical pathways. *Trends in Neurosciences*, 6:414–417, January 1983. ISSN 0166-2236, 1878-108X. doi: 10.1016/0166-2236(83)90190-X. URL https://www.cell.com/trends/neurosciences/abstract/0166-2236(83)90190-X. Publisher: Elsevier.

Ida Momennejad. Learning Structures: Predictive Representations, Replay, and Generalization. *Current Opinion in Behavioral Sciences*, 32:155–166, April 2020. ISSN 2352-1546. doi: 10.1016/j.cobeha.2020.02.017. URL https://www.sciencedirect.com/science/article/pii/S2352154620300371.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A. Raffel. Scaling Data-Constrained Language Models. In *Advances in Neural Information Processing Systems*, volume 36, pages 50358–50376, December 2023. doi: 10.52202/075280-2191. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/9d89448b63ce1e2e8dc7af72c984c196-Abstract-Conference.html.

Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, Cambridge, Massachusetts, 2012. ISBN 978-0-262-01802-9. URL https://mitpressbookstore.mit.edu/book/9780262018029.

John D. Murray, Alberto Bernacchia, David J. Freedman, Ranulfo Romo, Jonathan D. Wallis, Xinying Cai, Camillo Padoa-Schioppa, Tatiana Pasternak, Hyojung Seo, Daeyeol Lee, and Xiao-Jing Wang. A hierarchy of intrinsic timescales across primate cortex. *Nature Neuroscience*, 17(12):1661–1663, December 2014. ISSN 1546-1726. doi: 10.1038/nn.3862. URL https://www.nature.com/articles/nn.3862.

Scott O. Murray, Daniel Kersten, Bruno A. Olshausen, Paul Schrater, and David L. Woods. Shape perception reduces activity in human primary visual cortex. *Proceedings of the National Academy of Sciences*, 99(23):15164–15169, November 2002. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.192579399. URL https://www.pnas.org/content/99/23/15164. Publisher: National Academy of Sciences Section: Biological Sciences.

Romi Nijhawan. Motion extrapolation in catching. *Nature*, 370(6487):256–257, July 1994. ISSN 1476-4687. doi: 10.1038/370256b0. URL https://www.nature.com/articles/370256b0.

Romi Nijhawan. Visual prediction: Psychophysics and neurophysiology of compensation for time delays. *Behavioral and Brain Sciences*, 31(2):179–198, April 2008. ISSN 1469-1825, 0140-525X. doi: 10.1017/S0140525X08003804. URL https://www.cambridge.org/core/journals/behavioral-and-brain-sciences/article/visual-prediction-psychophysics-and-neurophysiology-of-compensation-for-time-delays/F0AE60F24239AD6435157A2C401C8BC6.

Lucine L. Oganesian, Saba Hashemi, and Maryam M. Shanechi. BaRISTA: Brain scale informed spatiotemporal representation of human intracranial neural activity. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=LDjBDk3Czb.

Bruno A. Olshausen. Sparse coding of time-varying natural images. *Journal of Vision*, 2(7):130–130, November 2002. ISSN 1534-7362. doi: 10.1167/2.7.130. URL http://jov.arvojournals.org/article.aspx?articleid=2120200.

Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, June 1996. ISSN 1476-4687. doi: 10.1038/381607a0. URL https://www.nature.com/articles/381607a0.

Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, December 1997.

ISSN 00426989. doi: 10.1016/S0042-6989(97)00169-7. URL https://linkinghub.elsevier.com/retrieve/pii/S0042698997001697.

Thomas J. Oxley, Nicholas L. Opie, Sam E. John, Gil S. Rind, Stephen M. Ronayne, Tracey L. Wheeler, Jack W. Judy, Alan J. McDonald, Anthony Dornom, Timothy J. H. Lovell, Christopher Steward, David J. Garrett, Bradford A. Moffat, Elaine H. Lui, Nawaf Yassi, Bruce C. V. Campbell, Yan T. Wong, Kate E. Fox, Ewan S. Nurse, Iwan E. Bennett, Sébastien H. Bauquier, Kishan A. Liyanage, Nicole R. van der Nagel, Piero Perucca, Arman Ahnood, Katherine P. Gill, Bernard Yan, Leonid Churilov, Christopher R. French, Patricia M. Desmond, Malcolm K. Horne, Lynette Kiers, Steven Prawer, Stephen M. Davis, Anthony N. Burkitt, Peter J. Mitchell, David B. Grayden, Clive N. May, and Terence J. O'Brien. Minimally invasive endovascular stent-electrode array for high-fidelity, chronic recordings of cortical neural activity. *Nature Biotechnology*, 34(3):320–327, 2016. ISSN 1546-1696. doi: 10.1038/nbt.3428. URL https://www.nature.com/articles/nbt.3428.

Artidoro Pagnoni, Ramakanth Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason E Weston, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Ari Holtzman, and Srini Iyer. Byte latent transformer: Patches scale better than tokens. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9238–9258, 2025. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.453. URL https://aclanthology.org/2025.acl-long.453/.

Dylan M. Paiton, Charles G. Frye, Sheng Y. Lundquist, Joel D Bowen, R. Zarcone, and B. Olshausen. Selectivity and robustness of sparse coding networks. *Journal of Vision*, 20, 2020. doi: 10.1167/jov.20.12.10.

Chethan Pandarinath, Daniel J. O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D. Stavisky, Jonathan C. Kao, Eric M. Trautmann, Matthew T. Kaufman, Stephen I. Ryu, Leigh R. Hochberg, Jaimie M. Henderson, Krishna V. Shenoy, L. F. Abbott, and David Sussillo. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature Methods*, 15(10):805–815, October 2018. ISSN 1548-7105. doi: 10.1038/s41592-018-0109-9. URL https://www.nature.com/articles/s41592-018-0109-9. Number: 10 Publisher: Nature Publishing Group.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. doi: 10.1109/ICCV51070.2023.00387. URL https://openaccess.thecvf.com/content/ICCV2023/html/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.html.

Felix C. Pei, Joel Ye, David M. Zoltowski, Anqi Wu, Raeed Hasan Chowdhury, Hansem Sohn, Joseph E. O'Doherty, Krishna V. Shenoy, Matthew Kaufman, Mark M.

Churchland, Mehrdad Jazayeri, Lee E. Miller, Jonathan W. Pillow, Il Memming Park, Eva L. Dyer, and Chethan Pandarinath. Neural Latents Benchmark '21: Evaluating latent variable models of neural population activity. In *Advances in Neural Information Processing Systems*, August 2021. URL https://openreview.net/forum?id=KVMS3fl4Rsv. Datasets and Benchmarks Track (Round 2).

D. J. Perkel, J. Bullier, and H. Kennedy. Topography of the afferent connectivity of area 17 in the macaque monkey: A double-labelling study. *Journal of Comparative Neurology*, 253(3):374–402, 1986. doi: 10.1002/cne.902530307.

Steven M. Peterson, Satpreet H. Singh, Nancy X. R. Wang, Rajesh P. N. Rao, and Bingni W. Brunton. Behavioral and Neural Variability of Naturalistic Arm Movements. *eNeuro*, 8(3), May 2021. ISSN 2373-2822. doi: 10.1523/ENEURO.0007-21.2021. URL https://www.eneuro.org/content/8/3/ENEURO.0007-21.2021.

Eugenio Piasini, Liviu Soltuzu, Paolo Muratore, Riccardo Caramellino, Kasper Vinken, Hans Op de Beeck, Vijay Balasubramanian, and Davide Zoccolan. Temporal stability of stimulus representation increases along rodent visual cortical hierarchies. *Nature Communications*, 12(1):4448, July 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-24456-3. URL https://www.nature.com/articles/s41467-021-24456-3.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, July 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Rajesh P. N. Rao. Correlates of Attention in a Model of Dynamic Visual Recognition. In *Advances in Neural Information Processing Systems*, volume 10, pages 879–885, 1998. URL http://papers.nips.cc/paper/1416-correlates-of-attention-in-a-model-of-dynamic-visual-recognition.pdf.

Rajesh P. N. Rao. An optimal estimation approach to visual perception and learning. *Vision Research*, 39(11):1963–1989, June 1999. ISSN 00426989. doi: 10.1016/S0042-6989(98)00279-X. URL https://linkinghub.elsevier.com/retrieve/pii/S0042698998002 79X.

Rajesh P. N. Rao. A sensory–motor theory of the neocortex. *Nature Neuroscience*, 27(7):1221–1235, July 2024. ISSN 1546-1726. doi: 10.1038/s41593-024-01673-9. URL https://www.nature.com/articles/s41593-024-01673-9.

Rajesh P. N. Rao and Dana H. Ballard. Dynamic Model of Visual Recognition Predicts Neural Response Properties in the Visual Cortex. *Neural Computation*, 9(4):721–763, May 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.4.721. URL https://doi.org/10.1162/neco.1997.9.4.721.

Rajesh P. N. Rao and Dana H. Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87, January 1999. ISSN 1097-6256, 1546-1726. doi: 10.1038/4580. URL http://www.nature.com/articles/nn0199_79.

Rajesh P. N. Rao, David M. Eagleman, and Terrence J. Sejnowski. Optimal Smoothing in Visual Motion Perception. *Neural Computation*, 13(6):1243–1253, June 2001. ISSN 0899-7667. doi: 10.1162/08997660152002843. URL https://doi.org/10.1162/08997660152002843.

Rajesh P. N. Rao, Dimitrios C. Gklezakos, and Vishwas Sathish. Active Predictive Coding: A Unifying Neural Model for Active Perception, Compositional Learning, and Hierarchical Planning. *Neural Computation*, 36(1):1–32, January 2024. ISSN 0899-7667. doi: 10.1162/neco_a_01627. URL https://doi.org/10.1162/neco_a_01627.

Fred Rieke, David Warland, Rob de Ruyter van Steveninck, and William Bialek. *Spikes: Exploring the Neural Code*. MIT Press, Cambridge, MA, USA, 1999. ISBN 978-0-262-68108-7. URL https://mitpress.mit.edu/9780262681087/spikes/.

Dario Ringach and Robert Shapley. Reverse correlation in neurophysiology. *Cognitive Science*, 28(2):147–166, March 2004. ISSN 03640213. doi: 10.1207/s15516709cog2802_2. URL http://doi.wiley.com/10.1207/s15516709cog2802_2.

K. S. Rockland and H. Ojima. Multisensory convergence in calcarine visual areas in macaque monkey. *International Journal of Psychophysiology*, 50(1):19–26, 2003. doi: 10.1016/S0167-8760(03)00121-1.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. doi: 10.1109/CVPR52688.2022.01042. URL https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

Michael E Rule, Timothy O'Leary, and Christopher D Harvey. Causes and consequences of representational drift. *Current Opinion in Neurobiology*, 58:141–147, October 2019. ISSN 0959-4388. doi: 10.1016/j.conb.2019.08.005. URL https://www.sciencedirect.com/science/article/pii/S0959438819300303.

Caroline A. Runyan, Eugenio Piasini, Stefano Panzeri, and Christopher D. Harvey. Distinct timescales of population coding across cortex. *Nature*, 548(7665): 92–96, August 2017. ISSN 1476-4687. doi: 10.1038/nature23020. URL https://www.nature.com/articles/nature23020.

Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://papers.nips.cc/paper/6114-weight-normalization-a-simple-reparameterization-to-accelerate-training-of-deep-neural-networks.

P. A. Salin and J. Bullier. Corticocortical connections in the visual system: structure and function. *Physiological Reviews*, 75(1):107–154, 1995. doi: 10.1152/physrev.1995.75.1.107.

Tommaso Salvatori, Yuhang Song, Yujian Hong, Lei Sha, Simon Frieder, Zhenghua Xu, Rafal Bogacz, and Thomas Lukasiewicz. Associative Memories via Predictive Coding. In *Advances in Neural Information Processing Systems*, volume 34, pages 3874–3886, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/1fb36c4ccf88f7e67ead155496f02338-Abstract.html.

David M. Schneider, Janani Sundararajan, and Richard Mooney. A cortical filter that learns to suppress the acoustic consequences of movement. *Nature*, 561(7723):391–395, September 2018. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-018-0520-5. URL http://www.nature.com/articles/s41586-018-0520-5.

Wolfram Schultz, Peter Dayan, and P. Read Montague. A Neural Substrate of Prediction and Reward. *Science*, 275(5306):1593–1599, March 1997. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.275.5306.1593. URL https://science.sciencemag.org/content/275/5306/1593. Publisher: American Association for the Advancement of Science Section: Articles.

Caspar M. Schwiedrzik and Winrich A. Freiwald. High-Level Prediction Signals in a Low-Level Area of the Macaque Face-Processing Hierarchy. *Neuron*, 96(1):89–97.e4, September 2017. ISSN 0896-6273. doi: 10.1016/j.neuron.2017.09.007. URL https://www.cell.com/neuron/abstract/S0896-6273(17)30842-5. Publisher: Elsevier.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162/.

Adam S. Shai, Costas A. Anastassiou, Matthew E. Larkum, and Christof Koch. Physiology of Layer 5 Pyramidal Neurons in Mouse Primary Visual Cortex: Coincidence Detection through Bursting. *PLOS Computational Biology*, 11(3):e1004090, March 2015. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1004090. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004090.

Shinsuke Shimojo. Postdiction: its implications on visual awareness, hindsight, and sense of agency. *Frontiers in Psychology*, 5, 2014. ISSN 1664-1078. doi: 10.3389/fpsyg.2014.00196. URL https://www.frontiersin.org/articles/10.3389/fpsyg.2014.00196.

James M. Shine. The thalamus integrates the macrosystems of the brain to facilitate complex, adaptive brain network dynamics. *Progress in Neurobiology*, 199:101951, April 2021. ISSN 0301-0082. doi: 10.1016/j.pneurobio.2020.101951. URL https://www.sciencedirect.com/science/article/pii/S0301008220302069.

James M. Shine, Eli J. Müller, Brandon Munn, Joana Cabral, Rosalyn J. Moran, and Michael Breakspear. Computational models link cellular mechanisms of neuromodulation to large-scale neural dynamics. *Nature Neuroscience*, 24(6):765–776, June 2021. ISSN 1546-1726. doi: 10.1038/s41593-021-00824-6. URL https://www.nature.com/articles/s41593-021-00824-6.

Joshua H. Siegle, Xiaoxuan Jia, Séverine Durand, Sam Gale, Corbett Bennett, Nile Graddis, Greggory Heller, Tamina K. Ramirez, Hannah Choi, Jennifer A. Luviano, Peter A. Groblewski, Ruweida Ahmed, Anton Arkhipov, Amy Bernard, Yazan N. Billeh, Dillan Brown, Michael A. Buice, Nicolas Cain, Shiella Caldejon, Linzy Casal, Andrew Cho, Maggie Chvilicek, Timothy C. Cox, Kael Dai, Daniel J. Denman, Saskia E. J. de Vries, Roald Dietzman, Luke Esposito, Colin Farrell, David Feng, John Galbraith, Marina Garrett, Emily C. Gelfand, Nicole Hancock, Julie A. Harris, Robert Howard, Brian Hu, Ross Hytnen, Ramakrishnan Iyer, Erika Jessett, Katelyn Johnson, India Kato, Justin Kiggins, Sophie Lambert, Jerome Lecoq, Peter Ledochowitsch, Jung Hoon Lee, Arielle Leon, Yang Li, Elizabeth Liang, Fuhui Long, Kyla Mace, Jose Melchior, Daniel Millman, Tyler Mollenkopf, Chelsea Nayan, Lydia Ng, Kiet Ngo, Thuyahn Nguyen, Philip R. Nicovich, Kat North, Gabriel Koch Ocker, Doug Ollerenshaw, Michael Oliver, Marius Pachitariu, Jed Perkins, Melissa Reding, David Reid, Miranda Robertson, Kara Ronellenfitch, Sam Seid, Cliff Slaughterbeck, Michelle Stoecklin, David Sullivan, Ben Sutton, Jackie Swapp, Carol Thompson, Kristen Turner, Wayne Wakeman, Jennifer D. Whitesell, Derric Williams, Ali Williford, Rob Young, Hongkui Zeng, Sarah Naylor, John W. Phillips, R. Clay Reid, Stefan Mihalas, Shawn R. Olsen, and Christof Koch. Survey of spiking in the mouse visual system reveals functional hierarchy. *Nature*, 592(7852):86–92, April 2021. ISSN 1476-4687. doi: 10.1038/s41586-020-03171-x. URL https://www.nature.com/articles/s41586-020-03171-x.

R. A. Silver. Neuronal arithmetic. *Nature Reviews Neuroscience*, 11(7):474–489, 2010. doi: 10.1038/nrn2864.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv*, 2025. doi: 10.48550/arXiv.2508.10104. URL `http://arxiv.org/abs/2508.10104`.

Yosef Singer, Yayoi Teramoto, Ben DB Willmore, Jan WH Schnupp, Andrew J King, and Nicol S Harper. Sensory cortex is optimized for prediction of future input. *eLife*, 7:e31557, June 2018. ISSN 2050-084X. doi: 10.7554/eLife.31557. URL `https://doi.org/10.7554/eLife.31557`.

Yosef Singer, Luke Taylor, Ben D. B. Willmore, Andrew J. King, and Nicol S. Harper. Hierarchical temporal prediction captures motion processing along the visual pathway. *eLife*, 12:e52599, October 2023. doi: 10.7554/eLife.52599. URL `https://doi.org/10.7554/eLife.52599`.

Evan C. Smith and Michael S. Lewicki. Efficient auditory coding. *Nature*, 439(7079): 978–982, February 2006. ISSN 1476-4687. doi: 10.1038/nature04485. URL `https://www.nature.com/articles/nature04485`. Number: 7079 Publisher: Nature Publishing Group.

Leslie N. Smith and Nicholay Topin. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. *arXiv*, May 2018. doi: 10.48550/arXiv.1708.07120. URL `http://arxiv.org/abs/1708.07120`.

Mandyam Veerambudi Srinivasan, Simon Barry Laughlin, and Andreas Dubs. Predictive coding: a fresh view of inhibition in the retina. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 216(1205):427–459, November 1982. ISSN 0080-4649. doi: 10.1098/rspb.1982.0085.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised Learning of Video Representations using LSTMs. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 843–852, Lille, France, July 2015. PMLR. URL `https://proceedings.mlr.press/v37/srivastava15.html`.

Kimberly L. Stachenfeld, Matthew M. Botvinick, and Samuel J. Gershman. The hippocampus as a predictive map. *Nature Neuroscience*, 20(11):1643–1653, November 2017. ISSN 1546-1726. doi: 10.1038/nn.4650. URL `https://www.nature.com/articles/nn.4650`.

Nicholas A. Steinmetz, Cagatay Aydin, Anna Lebedeva, Michael Okun, Marius Pachitariu, Marius Bauza, Maxime Beau, Jai Bhagat, Claudia Böhm, Martijn Broux, Susu Chen, Jennifer Colonell, Richard J. Gardner, Bill Karsh, Fabian Kloosterman, Dimitar Kostadinov, Carolina Mora-Lopez, John O'Callaghan, Junchol Park, Jan Putzeys, Britton Sauerbrei, Rik J. J. van Daal, Abraham Z. Vollan, Shiwei Wang, Marleen Welkenhuysen, Zhiwen Ye, Joshua T. Dudman, Barundeb Dutta, Adam W. Hantman, Kenneth D. Harris, Albert K. Lee, Edvard I. Moser, John O'Keefe, Alfonso Renart, Karel Svoboda, Michael Häusser, Sebastian Haesler, Matteo Carandini, and Timothy D. Harris. Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science*, 372(6539):eabf4588, 2021. doi: 10.1126/science.abf4588. URL https://www.science.org/doi/10.1126/science.abf4588.

Carsen Stringer and Marius Pachitariu. Analysis methods for large-scale neuronal recordings. *Science*, 386(6722):eadp7429, 2024. doi: 10.1126/science.adp7429. URL https://www.science.org/doi/10.1126/science.adp7429.

Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Charu Bai Reddy, Matteo Carandini, and Kenneth D. Harris. Spontaneous behaviors drive multidimensional, brainwide activity. *Science*, 364(6437):eaav7893, April 2019. doi: 10.1126/science.aav7893. URL https://www.science.org/doi/full/10.1126/science.aav7893. Publisher: American Association for the Advancement of Science.

Jake P. Stroud, Mason A. Porter, Guillaume Hennequin, and Tim P. Vogels. Motor primitives in space and time via targeted gain modulation in cortical networks. *Nature Neuroscience*, 21(12):1774–1783, December 2018. ISSN 1546-1726. doi: 10.1038/s41593-018-0276-0. URL https://www.nature.com/articles/s41593-018-0276-0.

Jørgen Sugar and May-Britt Moser. Episodic memory: Neuronal codes for what, where, and when. *Hippocampus*, 29(12):1190–1205, 2019. ISSN 1098-1063. doi: 10.1002/hipo.23132. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/hipo.23132.

Jeremias Sulam, Ramchandran Muthumukar, and R. Arora. Adversarial robustness of supervised sparse coding. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/170f6aa36530c364b77ddf83a84e7351-Abstract.html.

Jennifer Sun and Pietro Perona. Where is the sun? *Nature Neuroscience*, 1(3):183–184, July 1998. ISSN 1097-6256, 1546-1726. doi: 10.1038/630. URL http://www.nature.com/articles/nn0798_183.

Naoya Takahashi, Thomas G. Oertner, Peter Hegemann, and Matthew E. Larkum. Active cortical dendrites modulate perception. *Science*, 354(6319):1587–1590, December 2016. doi: 10.1126/science.aah6066. URL https://www.science.org/doi/10.1126/science.aah6066.

V. Talebi and C. L. Baker. Natural versus Synthetic Stimuli for Estimating Receptive Field Models: A Comparison of Predictive Robustness. *Journal of Neuroscience*, 32 (5):1560–1576, February 2012. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.4661-12.2012. URL https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.4661-12.2012.

Frank Tong, Ming Meng, and Randolph Blake. Neural bases of binocular rivalry. *Trends in Cognitive Sciences*, 10(11):502–511, November 2006. ISSN 13646613. doi: 10.1016/j.tics.2006.09.003. URL https://linkinghub.elsevier.com/retrieve/pii/S1364661306002403.

Doris Y. Tsao, Winrich A. Freiwald, Roger B. H. Tootell, and Margaret S. Livingstone. A Cortical Region Consisting Entirely of Face-Selective Cells. *Science*, 311(5761):670–674, February 2006. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1119983. URL https://science.sciencemag.org/content/311/5761/670. Publisher: American Association for the Advancement of Science Section: Report.

Endel Tulving. Episodic memory: From mind to brain. *Annual Review of Psychology*, 53(1):1–25, 2002. doi: 10.1146/annurev.psych.53.100901.135114. URL https://doi.org/10.1146/annurev.psych.53.100901.135114.

Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html.

J H van Hateren and D L Ruderman. Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 265: 2315–2320, December 1998. doi: 10.1098/rspb.1998.0577.

Alexander van Meegen and Sacha J. van Albada. Microscopic theory of intrinsic timescales in spiking neural networks. *Physical Review Research*, 3(4):043077, October 2021. doi: 10.1103/PhysRevResearch.3.043077. URL https://link.aps.org/doi/10.1103/PhysRevResearch.3.043077. Publisher: American Physical Society.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Deepak Verma and Rajesh P. Rao. Goal-Based Imitation as Probabilistic Inference over Graphical Models. In *Advances in Neural Information Processing Systems*, volume 18, pages 1393–1400, 2005. URL https://proceedings.neurips.cc/paper/2005/hash/db5cea26ca37aa09e5365f3e7f5dd9eb-Abstract.html.

Deepak Verma and Rajesh P. N. Rao. Planning and Acting in Uncertain Environments using Probabilistic Inference. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2382–2387, October 2006. doi: 10.1109/IROS.2006.281675. ISSN: 2153-0866.

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. doi: 10.1146/annurev-neuro-092619-094115. URL https://doi.org/10.1146/annurev-neuro-092619-094115. _eprint: https://doi.org/10.1146/annurev-neuro-092619-094115.

Eric Y. Wang, Paul G. Fahey, Zhuokun Ding, Stelios Papadopoulos, Kayla Ponder, Marissa A. Weis, Andersen Chang, Taliah Muhammad, Saumil Patel, Zhiwei Ding, Dat Tran, Jiakun Fu, Casey M. Schneider-Mizell, Nuno Maçarico da Costa, R. Clay Reid, Forrest Collman, Nuno Maçarico da Costa, Katrin Franke, Alexander S. Ecker, Jacob Reimer, Xaq Pitkow, Fabian H. Sinz, and Andreas S. Tolias. Foundation model of neural activity predicts response to new stimulus types. *Nature*, 640(8058):470–477, April 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-08829-y. URL https://www.nature.com/articles/s41586-025-08829-y.

Leonhard Waschke, Niels A. Kloosterman, Jonas Obleser, and Douglas D. Garrett. Behavior needs neural variability. *Neuron*, 109(5):751–766, March 2021. ISSN 0896-6273. doi: 10.1016/j.neuron.2021.01.023. URL https://www.sciencedirect.com/science/article/pii/S0896627321000453.

Laurenz Wiskott and Terrence J. Sejnowski. Slow Feature Analysis: Unsupervised Learning of Invariances. *Neural Computation*, 14(4):715–770, April 2002. ISSN 0899-7667. doi: 10.1162/089976602317318938.

Daniel M Wolpert, R. Chris Miall, and Mitsuo Kawato. Internal models in the cerebellum. *Trends in Cognitive Sciences*, 2(9):338–347, September 1998. ISSN 1364-6613. doi: 10.1016/S1364-6613(98)01221-2. URL https://www.sciencedirect.com/science/article/pii/S1364661398012212.

Shengjin Xu, Wanchen Jiang, Mu-ming Poo, and Yang Dan. Activity recall in a visual cortical ensemble. *Nature Neuroscience*, 15(3):449–455, March 2012. ISSN 1546-1726. doi: 10.1038/nn.3036. URL https://www.nature.com/articles/nn.3036.

Chaoqi Yang, M. Westover, and Jimeng Sun. BIOT: Biosignal transformer for cross-data learning in the wild. In *Advances in Neural Information Processing Systems*, volume 36, pages 78240–78260, 2023. doi: 10.52202/075280-3420. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/f6b30f3e2dd9cb53bbf2024402d02295-Abstract-Conference.html.

Joel Ye and Chethan Pandarinath. Representation learning for neural population activity with Neural Data Transformers. *Neurons, Behavior, Data analysis, and Theory*, 5(3):1–18, August 2021. doi: 10.51628/001c.27358. URL https://nbdt.scholasticahq.com/article/27358-representation-learning-for-neural-population-activity-with-neural-data-transformers.

Joel Ye, Jennifer L. Collinger, Leila Wehbe, and Robert Gaunt. Neural Data Transformer 2: Multi-context Pretraining for Neural Spiking Activity. In *Advances in Neural Information Processing Systems*, November 2023. URL https://nips.cc/virtual/2023/poster/72443.

Joel Ye, Fabio Rizzoglio, Adam Smoulder, Hongwei Mao, Xuan Ma, Patrick Marino, Raeed Chowdhury, Dalton Moore, Gary Blumenthal, William Hockeimer, Nicolas G. Kunigk, J. Patrick Mayo, Aaron Batista, Steven Chase, Adam Rouse, Michael L. Boninger, Charles Greenspon, Andrew B. Schwartz, Nicholas G. Hatsopoulos, Lee E. Miller, Kristofer E. Bouchard, Jennifer L. Collinger, Leila Wehbe, and Robert Gaunt. A Generalist Intracortical Motor Decoder. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/a00000e6a2208172700510bcd69d48e9-Abstract-Conference.html.

Serena Yeung, A. Kannan, Yann Dauphin, and Li Fei-Fei. Tackling over-pruning in variational autoencoders. *arXiv*, 2017. doi: 10.48550/arXiv.1706.03643. URL https://arxiv.org/abs/1706.03643.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. SoundStream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2022. ISSN 2329-9290. doi: 10.1109/TASLP.2021.3129994. URL https://dl.acm.org/doi/10.1109/TASLP.2021.3129994.

Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, and R. Fergus. Deconvolutional networks. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2528–2535, 2010. doi: 10.1109/CVPR.2010.5539957.

Matthew D. Zeiler, Graham W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *2011 International Conference on Computer Vision*, pages 2018–2025, 2011. doi: 10.1109/ICCV.2011.6126474.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11941–11952, October 2023. doi: 10.1109/ICCV51070.2023.01100. URL https://ieeexplore.ieee.org/document/10378327.

Daoze Zhang, Zhizhang Yuan, Yang Yang, Junru Chen, Jingjing Wang, and Yafeng Li. Brant: Foundation model for intracranial neural signal. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-1144. URL https://openreview.net/forum?id=DDkl9vaJyE.

Yizi Zhang, Yanchen Wang, Donato M. Jiménez-Benetó, Zixuan Wang, Mehdi Azabou, Blake Richards, Renee Tung, Olivier Winter, The International B. Laboratory, Eva Dyer, Liam Paninski, and Cole Hurwitz. Towards a "Universal Translator" for Neural Dynamics at Single-Cell, Single-Spike Resolution. In *Advances in Neural Information Processing Systems*, volume 37, pages 80495–80521, December 2024. doi: 10.52202/079017-2559. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/934eb45b99eff8f16b5cb8e4d3cb5641-Abstract-Conference.html.

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. In *International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=vOdz3zhSCj.

Hui Zheng, Hai-Teng Wang, Wei-Bang Jiang, Zhong-Tao Chen, Li He, Pei-Yang Lin, Peng-Hu Wei, Guo-Guang Zhao, and Yun-Zhe Liu. Du-IN: Discrete units-guided mask modeling for decoding speech from intracranial neural signals. In *Advances in Neural Information Processing Systems*, pages 79996–80033, 2024. doi: 10.52202/079017-2542. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/92559987ee79e42a2b01d534a54682ee-Abstract-Conference.html.

Pawel Zmarz and Georg B. Keller. Mismatch Receptive Fields in Mouse Visual Cortex. *Neuron*, 92(4):766–772, November 2016. ISSN 0896-6273. doi: 10.1016/j.neuron.2016.09.057. URL https://www.cell.com/neuron/abstract/S0896-6273(16)30699-7.