UC Berkeley
UC Berkeley Electronic Theses and Dissertations

Title
The impact of correlated variability on models of neural coding

Permalink
https://escholarship.org/uc/item/2s93m6wd

Author
Sachdeva, Pratik Singh

Publication Date
2021

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

The impact of correlated variability on models of neural coding

by

Pratik Singh Sachdeva

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Physics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Michael R. DeWeese, Chair, Co-chair
Adjunct Assistant Professor Kristofer E. Bouchard, Co-chair
Professor Na Ji
Professor Bruno A. Olshausen

Summer 2021

The impact of correlated variability on models of neural coding

Copyright 2021
by
Pratik Singh Sachdeva

Abstract

The impact of correlated variability on models of neural coding

by

Pratik Singh Sachdeva

Doctor of Philosophy in Physics

University of California, Berkeley

Professor Michael R. DeWeese, Chair, Co-chair

Adjunct Assistant Professor Kristofer E. Bouchard, Co-chair

Variability is a prominent feature of neural systems: neural responses to repeated presentations of the same external stimulus will typically vary from trial to trial. Furthermore, neural variability exhibits pairwise correlations, commonly referred to as *correlated variability*. Correlated variability is a pervasive neural phenomenon that arises due to a variety of sources including shared input, biological noise, global fluctuations, and neural activity unobserved by experimental apparatuses. It is of theoretical interest because of its importance for models of neural coding: the existence of correlated variability can improve or harm neural coding depending on its structure. In this work, we examine how correlated variability impacts neural coding for both analyses on decoding efficacy and parametric models of neural activity. First, we demonstrate that correlated variability induced by noise sources common to a neural population can be manipulated by heterogeneous synaptic weighting to improve neural coding, even at the cost of amplifying the noise. Second, we demonstrate that correlated variability in neural data exhibits worse than chance decoding fidelity, and identify biological constraints in achieving optimal neural representations. Third, we examine how an improved inference algorithm for common parametric models can shape the scientific interpretation of common systems neuroscience models, despite the presence of correlated variability in the data. Lastly, we identify how omitting correlated variability arising from unobserved activity in parametric models of tuning and functional coupling can bias parametric estimates, and propose a new model and inference procedure to mitigate these biases. Together, our results highlight the importance of correlated variability on a wide range neural coding models.

I am fortunate that this narrative, despite the sliver of human knowledge and experience it encompasses, will be preserved for future scientists to read. Many in history were not afforded such a privilege.

This work is dedicated to those whose stories and legacies were lost to social injustice, colonialism, and hegemony.

Contents

List of Figures

List of Tables

Acknowledgments

I cannot hope to enumerate all the ways that my family helped me reach this point. My parents were able to provide me an emotional, economic, and educational support system which ensured that I was always in a position to succeed, from elementary to graduate school. My mother nurtured empathy, compassion, and resilience within me. My father is ultimately the reason I got into science and began my PhD, and I look up to his integrity and thoroughness in the pursuit of good science every day. Above all, my parents instilled within me a sense of integrity, work ethic, and aspiration for justice that I try to carry forward to this day. My brother was my constant companion growing up. I am so happy that you have become one of my closest friends, and that we are able to live so close together. I look forward to our future nature adventures, overcoming injuries, and eating copious amounts of (hopefully vegan, to your displeasure) food.

I was lucky to have two advisors, Mike DeWeese and Kris Bouchard, who fostered and steered my growth as a researcher. Mike's unbridled optimism and energy helped ease my transition from physics to neuroscience. Our initial discussions had a significant impact on the research topics I pursued within neuroscience. Kris has been an invaluable mentor, training me to become a better thinker, writer, presenter, and scientist. I will always appreciate his dedication to mentorship, in both science and life. Participating in both labs exposed me to a broad range of subjects, including physics, theoretical and experimental neuroscience, machine learning, statistics, information theory, and others. This exposure helped me develop both the flexibility and confidence to approach new fields and literature which in turn supported my transition into new research during my PhD. I'm grateful to both Mike and Kris in providing the support, advice, and mentorship to successfully navigate both transitions.

I would not be the researcher I am today without Jesse Livezey's mentorship. Jesse has supported me through every one of my thesis projects (he is a co-author on all of them) by freely offering his expertise and knowledge, whether it be on coding practices, research techniques, narrative building, or writing. We've had a fruitful collaboration for the past four years. At the same time, both his and his partner's, Sarah Maslov's commitment to social justice has challenged me to push forward my own activism. Most of all, Jesse and Sarah have been great friends. Lastly, thank you to Luca for your screeches and farts over Zoom, which provided the source of inspiration I needed to finish this thesis.

My research was conducted at the Redwood Center for Theoretical Neuroscience, a large collaboration between four PIs and their groups. Working in the Redwood Center has been a joy, and I continue to be humbled and inspired by the hard-working and brilliant students and postdocs who have walked its halls. My officemates in 567 – Chris (C-Dub) Warner, Paxon Frady, Vasha Dutell, and Yubei Chen – have never failed to bring a smile to my face through our wide range of shenanigans (stretch breaks, gymnastics, orange tossing, etc.). Our office, due to its large size and available couch, had frequent visitors who always brought fruitful and entertaining discussions. Thanks to Charles Frye, David Clark, Dylan Paiton, Eric Dodds, Eric Weiss, Mayur Mudigonda, Max Dougherty, Neha Wadia, Ryan

Zarcone, and the many others of the Redwood Center who have helped me become a better researcher. Mike Fang (Mike Fang).

It's rare for one to stay in touch with childhood friends, and even rarer to end up living with them as adults. I'm lucky to have been able to live with three of my oldest friends throughout graduate school. The founding members of the North Berkeley Youth Center – Patrick, Dhyan, and Eli – have been a constant source of laughs, misadventures, and growth. In particular, Patrick has been my roommate for the last 6 years, and my oldest friend – it is still surreal that we are going on dumb food adventures and making equally dumb videos to this day. Thank you, as well, to the affiliate members of the NBYC: Adam, Amaia, Gautier, Jannes, Joyce, and Kristina, for brightening and enlivening our home.

The Bay Area has become my home away from home. I have come to love the Bay Area (and in particular, Oakland) with all my heart, and a significant reason is the support system and family that I'm lucky to have here. Thank you to my family: Timmy Bhaiya, Manpreet Bhabi, Wade Masi, Kirpa, Himmet, Ganeev, Aman Bhaiya, Harleen Bhabi, Bayant, Harnoor, Bira Masi, and Kim Didi, for your support, taking me on hikes, feeding me great food, letting me do my laundry at your homes, and never letting me feel too far from home.

Two of my core friend groups from undergrad at Washington University in St. Louis supported me throughout graduate school, either via in-person trips or video chats over Zoom (particularly during the COVID-19 pandemic). Thank you to the Wolfpack – Caleb, Cecilia, and Will – for the laughs, conversations on life, and the clowning (particularly on Cecilia). We've grown so much together, and I can't wait to see where life will continue to take us. Thank you to the Scrubs – Ethan, JPei, Noodle, Ryan, Sara – for board games (in-person and over Zoom), general silliness, and always making life seem a little lighter. In particular, thanks for Ryan and Masha (and Winslow!) for coming to your senses and joining me in the Bay Area.

Navigating the logistics of graduate school is not easy. The Physics student services and support staff have worked tirelessly to ensure that I and other Physics graduate students were able to find our way through classes, forms, funding, and more. In particular, thank you to Anne Takizawa, Brian Cunningham, Claudia Trujillo, Donna Sakima, Joelle Miles, and Kathy Lee, for the support you've provided me in my capacity as a Physics graduate student, GSI, researcher, and activist. Thank you in particular to Brian and Joelle for their unwavering support in the student effort to shape and build positive norms in the Physics community.

My graduate years have involved two pivots: the first from physics to neuroscience, and the second from neuroscience to the emerging "data-driven social justice" (the title still needs some work-shopping). The second pivot was challenging, but two experiences have been formative in providing me the requisite training and experience needed to venture further into the field. First, with DataKind, I completed a project on predictive modeling for community health environments. I'm grateful to Michael, Tali, Erika, and my teammates for providing the support to be able to see that project to completion. Second, the Data Science for Social Good Fellowship at the University of Washington provided me instrumental training, networking, and experience. Thank you to Sarah, Anissa, Loren, Matt, Scott,

and Spencer for your mentorship. And of course, thank you my squad – Ari, Hikari, and Juandalyn – for your laughs, lessons, and friendship. Somehow, we made it all work over Zoom.

This work was completed in Berkeley and Oakland, California. This land belonged to a variety of groups who have come to collectively be referred to as the Ohlone people. They were violently stripped of their land and possessions by Spanish colonizers in the early 1800s.

Chapter 1

Introduction

1.1 Overview

A central goal of theoretical and computational systems neuroscience is to characterize neural coding, or how external stimuli are represented in the activity of neural populations [57]. This task is particularly difficult in systems neuroscience, given the size and complexity of neural systems [156]. We typically achieve this by specifying a *model* that relates the structure of spiking activity in an ensemble of neurons to the input stimuli. The model, when fit to neural data, or optimized to satisfy a desired property, can then be interpreted to gain insight into the underlying neural system.

The structure of a model – i.e., its mathematical or computational formulation – reflects our assumptions about the properties of the neural system and its constituent units. These assumptions effectively act as constraints, which serve to simplify the model and abstract away details that may not be relevant for the question at hand. For example, we may choose to model a neural system with functional units that output a continuous firing rate. This choice reflects several assumptions: that the neurons are the only relevant units in the system (thereby omitting other neural bodies); that firing of action potentials is the relevant means by which neurons communicate (thereby neglecting other neural signals); and that firing rate is the meaningful quantity in the action potential (thereby neglecting any spike timing). Thus, there is a natural tension between a model's expressiveness (less constraints) and simplicity (more constraints). The more expressive a model, the more powerfully it can capture the details of the underlying system. These benefits come at the cost of the model's interpretability.

At the same time, a model should possess sufficient *degrees of freedom*, which reflect the properties of the system that could conceivably be manipulated. In almost all cases, the degrees of freedom correspond to parameters that we are free to choose. To interpret the model, then, we choose some parameter configuration within the degrees of freedom that satisfies a criterion. This criterion could reflect a desirable biological property or a suitable fit to existing neural data. After a model is optimized, we can examine the fit parameters

– or some output of the model that depends on them – to aid in answering questions about the neural system in question. In the example described above, this could entail fitting parameters that detail how the neuron's firing depends on an input stimulus. After performing the fit, we can analyze the parameters to determine what stimulus conditions evoke the most activity from the neurons.

The two criterion in choosing the model parameters reflect two different approaches in computational neuroscience. In the former, we approach the problem from first principles by asking, for example, how a neural system might perform a specific computation given biological constraints. This approach often invokes the efficient coding hypothesis, which posits that neural populations construct "efficient" representations of input stimuli [22]. From this perspective, any consistent structure observed in the activities of neural ensembles can be framed in terms of how it might "improve" the neural coding via some metric of efficiency. A second approach involves constructing phenomenological models that we fit to experimentally recorded neural data. These models can be sufficiently abstract such that they facilitate interpretation, but robust enough to capture the structure in the neural data. Furthermore, they can serve a predictive purpose (if they are capable of generating data) or can be interpreted by examining the fitted parameters directly. Both approaches aid our understanding of the neural system by allowing us to characterize its structure and behavior.

When we use a model in a systems neuroscience setting, we must choose some neural phenomena by which to constrain the model. One such phenomenon is that of neural variability: neurons respond variably under presentations of the same stimulus. Thus, to account for such variability, we commonly utilize *probabilistic models*. These models constrain the average neural response, with the trial-to-trial variability accounted for by some inherent stochasticity in the model. Thus, the model can be interpreted in how it relates the stimulus to the neural response. In a standard example, simple cells in visual cortex can be modeled with a linear model to relate the stimulus – a drifting grating – to their firing rate. The model parameters demonstrate that, on average, neurons are most active in response to a preferred stimulus value [189]. The remaining trial-to-trial variability is simply modeled as noise, whether it be Gaussian or Poisson.

However, neural variability contains additional structure that can influence a model. Specifically, variability in neural responses is correlated across pairs of neurons. This *correlated variability* has been observed consistently throughout cortex, under a variety of experimental conditions, and is of paramount importance for neural coding [51, 11, 106]. Its presence stems from a variety of sources. Generally, correlated variability arises from inputs shared across neurons in a neural circuit. Any variability in the inputs that cannot be explained by the stimulus – whether it be true biological noise, unobserved neurons, global fluctuations, attentional state, etc. – will result in correlated variability across the observed neural population. Thus, the phenomenon of correlated variability reflects both fundamental biological noise as well as our limitations in probing the entire neural system.

In this thesis, we examine the role of correlated variability in models of neural coding. We approach this general question from two perspectives: the usage of theoretical models optimized for decoding accuracy and phenomenological models fit to experimentally recorded

data. Thus, this thesis can be divided roughly into two parts, corresponding to each of these perspectives. In the first half, we examine correlated variability from an efficient coding perspective, and ask whether its presence is beneficial for stimulus decoding. Broadly, we answer the following questions:

(1) Neural circuits receive thousands of common inputs, some of which may be extraneous to the stimulus at hand. How do these common noise sources interact with synaptic weighting to produce correlated variability, and thus impact decoding performance? More simply, can neural circuits overcome common noise sources via their synaptic weighting? (Chapter 2)

(2) To what degree is the correlated variability observed in neural systems structured in a manner to optimize decoding? More simply, is correlated variability *efficient* from a decoding perspective? (Chapter 3)

In the second half, we turn to how correlated variability as a structure of neural activity impacts the fitting of phenomenological models of neural activity:

(3) Correlated structure generally impedes the fitting of phenomenological models, because it introduces correlations among predictive features. Can we develop improved inference techniques that are stable to such structure in common systems neuroscience models? How does this influence their interpretation relative to traditional approaches? (Chapter 4)

(4) How can we simultaneously model a neuron's dependence on external input (e.g., stimuli) and internal input (e.g., other neurons) given that we only record from a subset of the complete neural population? The unobserved neural activity – a source of correlated variability – will bias parameter estimates in simple phenomenological models. How do we account for correlated variability in such systems neuroscience models? (Chapter 5)

Lastly, a chapter in this thesis is dedicated to the development of software engineering tools for science (Chapter 6). Each of the projects in this thesis comes with a relatively polished software package capable of reproducing their analyses and figures. Thus, this final chapter serves as an outline to interested researchers in developing such packages for their scientific work.

The remaining two sections in this chapter are structured as follows. In Section 1.2, we present a gentle overview of correlated variability and the requisite background literature for later chapters. We introduce correlated variability, its importance for decoding, how to measure its decoding strength, and how it impacts the scaling of information in neural populations. In Section 1.3, we provide research summaries, while additionally framing each project thematically within the broader scope of the thesis. Each research summary is accompanied with a "research narrative". As many graduate students know, it is difficult to predict the research trajectory of a project at its onset. Experiments fail, results may be

surprising (or negative), new literature arises, and things never go according to plan. Thus, publications rarely present an accurate picture of the research trajectory. Instead, they frame the results as an expected output of a "well-designed" or "well-motivated" research process. The narratives presented here, therefore, serve to shed light on how these projects started, how they changed, and how they were finished.

1.2 Correlated Variability and its Importance for Neural Coding

Neural activity throughout cortex has long understood to be "variable:" spiking responses can differ considerably between multiple presentations of the same stimulus [197, 58]. For example, in one of the earliest studies on this subject, simple cells in cat striate cortex were presented with drifting gratings at different contrasts and spatial frequencies [58]. Importantly, each stimulus was presented multiple times. The author found that the number of spike responses varied from trial to trial, and furthermore, the variance of the spike counts scaled with the average firing rate. These observations, and others, formed the basis of theoretical studies modeling spiking responses as Poisson processes. Additionally, they laid the groundwork for studies examining the higher-order statistics of neural variability.

From a neural coding perspective, such variability is undesirable if the animal must use the neural responses to decode the stimulus. The solution to decoding in the presence of trial-to-trial variability is *redundancy*. If there are enough neurons in the population, with different stimulus preferences, their responses can be aggregated to compute a reasonable estimate of the stimulus [11]. Thus, neural variability can be "averaged away" provided that a circuit contains a suitable number of neurons. Given the extraordinary number of neurons in even a simple microcircuit, "averaging away" population coding is achievable. At the same time, neural decoding is likely only a small part of the overall neural computation, and "averaging away" only requires linear decoding. A more advanced decoder, operating on neurons that are coding for multiple stimuli, could likely achieve good decoding performance.

The "averaging away" view becomes more complicated if the variability carries higher-order structure *across* the neural population. A landmark study by Zohary, Shadlen, and Newsome [226] first characterized the second-order structure in neural variability at the pairwise level. Specifically, they recorded from a population of neurons in the middle temporal visual area (MT) of rhesus monkey, presented with a motion detection task of random dot images. They observed that pairs of neurons in the population typically exhibited a weak correlation in the variability of their responses, with an average of 0.12 across the population. Importantly, in a simple model of population coding, they demonstrated that neurons exhibiting positive correlations in their variability will saturate the population's signal-to-noise ratio as a function of circuit size. Furthermore, this saturation would occur no matter the strength of the observed correlation. Their result was an indictment against the prevailing "averaging away" view in population coding. Their observation implies that there is a

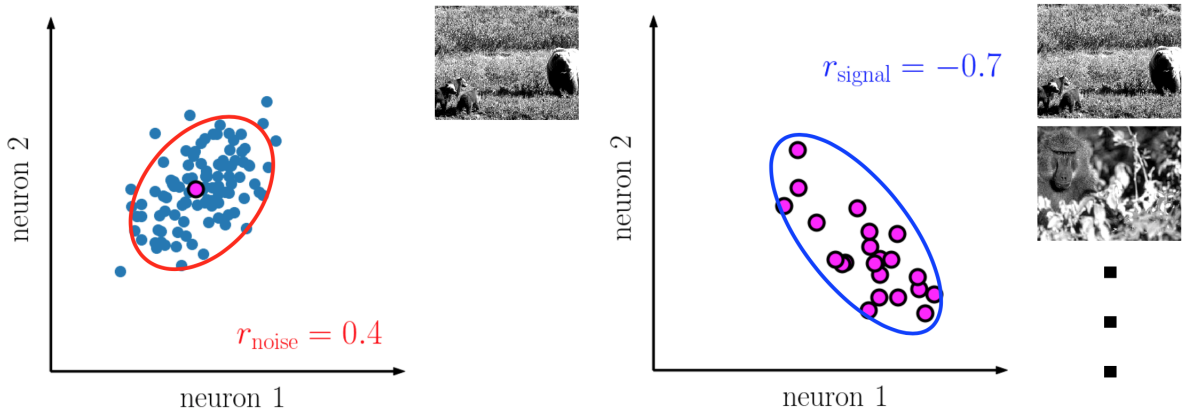


Figure 1.1: **Noise and signal correlations.** Each plot depicts the neural space, with each axis referring to the activity of a neuron in response to a stimulus. **Left:** Noise correlations. Each blue point refers to the neural response on a distinct trial for a single stimulus (depicted to the right as an image). Pink point denotes the mean response across trials. The blue points across trials exhibit a correlation in the neural space, with magnitude given by 0.4 (covariance ellipse depicted in red). This correlation is the noise correlation. **Right:** Signal correlations. Each pink point denotes the mean response to a distinct stimulus (depicted to the right as images). The mean responses exhibit a correlation in the neural space, with magnitude given by -0.7 (covariance ellipse depicted in blue). This is the signal correlation.

practical limit on the number of neurons in a cortical microcircuit coding for a particular stimulus, and sets an upper bound for coding, psychophysical, and behavioral capacity.

A toy example serves to concretely demonstrate the phenomenon of correlated variability. Consider the simple case of two neurons. We examine their responses in the *neural space*, where each axis denotes one neuron's response (Fig. 1.1). On repeated presentations of the same stimulus, the neural responses between the pair of neurons are correlated. The exact value of this correlation is referred to as the *noise correlation* (Fig. 1.1: red covariance ellipse). At the same time, we can define a *signal correlation*, or the correlation amongst the average neural responses to a variety of stimuli (Fig. 1.1: right). The relationship between the signal and noise correlation is important from a theoretical perspective, and will be discussed in the following section.

Thus far, we have used various terminology to refer to second-order structure in the variability of neural activity, including "noise correlations," "shared variability," and "correlated variability." In this thesis, we will largely refer to it as correlated variability. Correlated variability is largely a descriptive term, whereas "noise correlations" presumes that the underlying variability is "noise," while "shared variability" can be used more generally than second-order structure. We only use the term "noise correlations" when referring to the *quantity* describing the correlation in the variability (i.e., as a specific number, as done in Fig. 1.1). Furthermore, we use "shared variability" when explicitly defined and appropriate.

A landmark theoretical study followed the work of Zohary et al., examining the role of correlated variability in a population code [2]. Abbott and Dayan grounded their analysis in the Fisher information, an information theoretic quantity that measures the ability of a representation $\mathbf{r} = f(x)$ to reconstruct some observable quantity x [53]. Using the Fisher information, they assessed the strength of a population code, analytically, under various configurations of correlated variability. Abbott & Dayan were able to reproduce the saturating behavior observed by Zohary et al. in the Fisher information for positive correlated variability. They additionally scenarios in which the presence of correlated variability did *not* saturate the Fisher information. Even more strikingly, Abbott & Dayan provided cases where correlated variability *improved* decoding relative to the case of independent noise.

The work by Abbott & Dayan spurred a long line of theoretical work examining the implications of correlated variability for neural coding. This work has largely been concerned with elucidating whether correlated variability improves or harms coding fidelity, both in the finite-neuron and infinite-neuron case. Additionally, these theoretical analyses have been accompanied by a robust line of experimental work examining the properties of noise correlations in various brain regions, behavioral settings, and population sizes [51].

Correlated variability: harmful or beneficial?

We turn to a canonical toy model to better understand why correlated variability can either improve or harm neural coding. Consider, once again, the neural space of a two neuron population. This system is presented with two stimuli, s_1 and s_2, which evoke mean responses as depicted in Figure 1.2. Given these mean responses, an optimal decoding plane can be drawn to efficiently reconstruct the stimulus despite the presence of neural variability. In the case of uncorrelated variability (Fig. 1.2, left) the neural responses overlap across the optimal decoder, which reduces the accuracy of the decoder. If the variability is positively correlated, however (i.e., the system exhibits positive noise correlations), then the variability is reshaped in such a way that it overlaps even more, resulting in reduced decoding performance (Fig. 1.2, middle). However, if the variability is *negatively* correlated, i.e., the system exhibits negative noise correlations, then the variability lies parallel to the optimal decoder (Fig. 1.2, right). In this case, the coding performance improves relative to the uncorrelated case. Framed in another way, the relationship between the signal correlations (in this case, it is positive) and noise correlations informs whether decoding improves due to the correlated variability.

In practice, the situation is more complicated. The correlated variability structure exists across the entire neural population, not just two neurons. Furthermore, we are not simply concerned with discriminating between two stimuli, but often many pairwise stimuli existing on a spectrum. Earlier work, building on that of Dayan & Abbott, found that noise correlations will harm neural coding [181, 175]. More recent work has found that the picture is complicated, particularly when the population of neurons exhibits diverse tuning [62, 136]. Furthermore, correlated variability exhibits stimulus-dependence [92], which may improve neural coding in early sensory areas such as retina [227, 66]. Overall, the general conclusion is that correlated variability may have varying impacts on neural coding depending on its

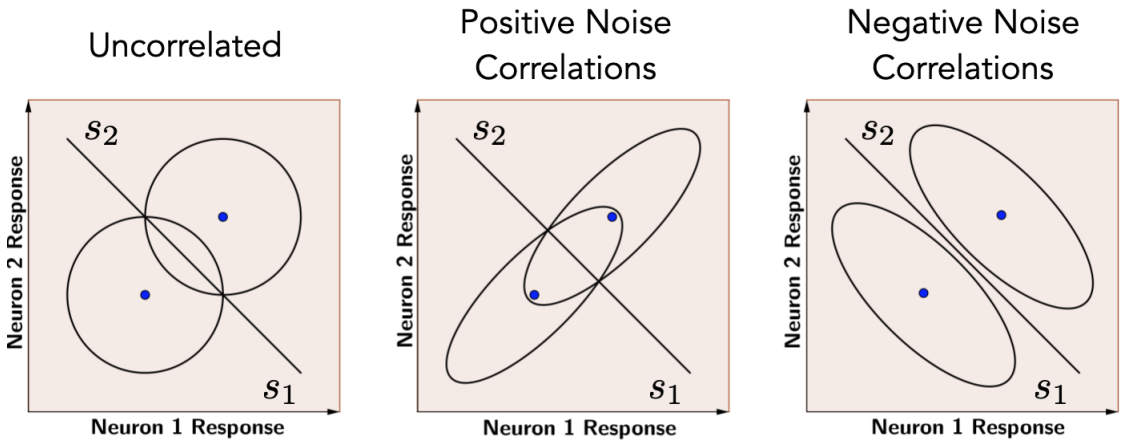


Figure 1.2: **Correlated variability can improve or harm decoding depending on its structure.** Each axis denotes the response of a neuron in the two-neuron population. Blue points denote mean responses to stimuli s_1 and s_2. Diagonal line depicts the optimal decoder given the mean firing rates. **Left:** Uncorrelated variability results in neural responses that may overlap trial-to-trial. **Middle:** Positive noise correlations reshape the variability to harm decoding relative to the uncorrelated case, by increasing the overlap. **Right:** Negative noise correlations improve decoding by reshaping variability to lie parallel to the decoder.

structure, relationship with the tuning properties of the neural population, and brain region in which it occurs.

An important point here is that "harmful" or "beneficial" must be interpreted in relative terms. That is, when we ask whether correlated variability is harmful or beneficial for neural coding, there is an implicit alternative variability structure to which we are comparing. This can be thought of as the null model. If we are comparing to the null model of *no* variability, then obviously correlated variability will always be harmful – we would rather have no variability then any at all, whether it is correlated or not. In Figure 1.2, we are implicitly comparing to the null model of uncorrelated noise, where the variances of the neural activity are kept the same, but the off-diagonal covariance structure is destroyed. The question of whether this is an appropriate null model to compare to is of particular interest in this thesis, and will be explored more deeply in Chapter 3.

Measuring the strength of a neural code

Thus far, we have not precisely defined "the strength of a neural code" beyond stating quantities that past authors have examined. Since the neural code provides some representation of the external stimulus, information theory is well-suited to assessing the fidelity of a neural code. It provides a rich set of tools by which the amount of "information" about the stimulus is captured in a population, and thus has served as the foundation for much of the theoretical

analyses on correlated variability and neural coding.

The predominant measure used to evaluate a neural code in the correlated variability literature is the Fisher information. The Fisher information is framed in terms of decoding: it sets a limit by which the readout of a population code can determine the value of the stimulus. Formally, it sets a lower bound to the variance of an unbiased estimator for the stimulus. Framed in terms of neural code, the Fisher information of a representation $\mathbf{r} = f(s)$ quantifies how well the stimulus s can be decoded given the representation. Specifically, for any estimator $\hat{s}$ that depends on the representation $\mathbf{r}$, the variance of that estimator is lower bounded by the Fisher information:

$$\text{var}(\hat{s}) \geq \frac{1}{I_F(s)}. \tag{1.1}$$

Equation (1.1) is known as the *Cramer-Rao bound*. While the Fisher information provides a lower bound to the variance of *any* unbiased estimator, there may not exist an estimator that saturates the bound.

The Fisher information is given by the variance of the *score*, which is the derivative of the log-representation with respect to the stimulus [53]. Define the probability density of the representation, dependent on the stimulus s, as $p(\mathbf{r}; s)$. Then, the Fisher information is

$$I_F(s) = \mathbb{E}\left[\left(\frac{\partial}{\partial s}\log p(\mathbf{r}; s)\right)^2 \middle| s\right] \tag{1.2}$$

$$= \int \left(\frac{\partial}{\partial s}\log p(\mathbf{r}; s)\right)^2 p(\mathbf{r}; s) ds. \tag{1.3}$$

The Fisher information can alternatively be written as a second derivative of the score:

$$I_F(s) = -\mathbb{E}\left[\frac{\partial^2}{\partial s^2}\log p(\mathbf{r}; s)\middle| s\right]. \tag{1.4}$$

In practice, the Fisher information is often analytically intractable. In the case where the representation takes on a Gaussian noise model, where $p(\mathbf{r}; s)$ can be described as a Gaussian with mean $\mathbf{f}(s)$ and covariance $\mathbf{\Sigma}(s)$, the Fisher information takes on the form

$$I_F(s) = \frac{\partial \mathbf{f}(s)}{\partial s}^T \mathbf{\Sigma}^{-1}(s)\frac{\partial \mathbf{f}(s)}{\partial s} + \text{Tr}\left[\mathbf{\Sigma}^{-1}(s)\mathbf{\Sigma}'(s)\mathbf{\Sigma}^{-1}(s)\mathbf{\Sigma}'(s)\right]. \tag{1.5}$$

When the covariance is stimulus-independent, the second term vanishes, and the expression reduces to the *linear Fisher information* (LFI):

$$I_{LFI}(s) = \frac{\partial \mathbf{f}(s)}{\partial s}^T \mathbf{\Sigma}^{-1}(s)\frac{\partial \mathbf{f}(s)}{\partial s} \tag{1.6}$$

The LFI serves as a lower bound for the Fisher information and thus is a useful proxy when the Fisher information is challenging to calculate analytically. Furthermore, the LFI

comes with its own decoder – the locally optimal *linear* estimator [106] – which serves as its namesake.

The linear Fisher information is the predominant measure used in correlated variability analyses. It is favored for several reasons. First, it describes the strength of a neural code from the perspective of decoding, which is desirable in correlated variability settings. Furthermore, it is easy to calculate, is often a good lower bound for the Fisher information, and comes with a decoder of the representation [216]. Thus, we know that a neural code with corresponding Fisher information could actually achieve the prescribed decoding variance with a simple linear estimator. Lastly, the linear Fisher information can be analogized to a signal-to-noise ratio, where the signal, or discriminability (i.e., the derivative of the tuning curve) is scaled according to the inverse covariance matrix. Thus, its form is easily interpretable.

Another measure of interest is the Shannon mutual information. The mutual information quantifies the reduction in uncertainty of one random variable given knowledge of another. In the context are neural coding, we are interested in quantifying how much knowledge of the neural representation $\mathbf{r}$ reduces uncertainty about the stimulus s. The mutual information, then, is defined as

$$I[s, \mathbf{r}] = \int ds d\mathbf{r}\ p(s, \mathbf{r}) \log \left(\frac{p(s, \mathbf{r})}{p(s)p(\mathbf{r})} \right). \tag{1.7}$$

The mutual information is one of the foundational results of information theory, and likely the quantity of highest interest due to its desirable properties. However, it is notoriously difficult to calculate, either analytically or numerically. This holds especially true in the high-dimensional neural context. Thus, mutual information is not often explored in the context of correlated variability, since estimating it from data would require an extraordinarily large number of samples given the size of the population. However, it has been used in some theoretical analyses, and we return to it in Chapter 2.

Differential correlations limit information

A substantial portion of the theoretical work examining the impact of correlated variability on a neural code has been concerned with the scaling properties of information as a function of population size. This interest is rooted in the efficiency of information processing throughout cortex. Animal behavior on tasks is not perfect, which implies that the total amount of information about the stimulus is finite (Fig. 1.3). If early sensory areas, such as V1, can continually add information as a function of population size, this implies that downstream processing loses much information through suboptimal computation (Fig. 1.3: red line). If, instead, the information saturates, but at a value much higher than the information available in the behavior, then downstream processing is inefficient, but less so than the unsaturated case (Fig. 1.3: purple line). At the other extreme, if information saturates at a low value, close to that of behavior, this implies that neural systems are limited at the early sensory input, and downstream processing is efficient (Fig. 1.3: blue line).

The question, then, becomes: does correlated variability limit the growth of information in a population, as Zohary et al. found? If so, what is the structure of such information-limiting variability? This question was answered by a landmark paper by Moreno-Bote et al. [128]. They identified a particular structure of correlations – *differential correlations* – whose presence in the covariance structure of the neural population will cause the information to saturate as a function of the number of neurons. Differential correlations take on the form $\mathbf{f}'\mathbf{f}'^T$, where $\mathbf{f} = \mathbf{f}(s)$ is the mean response of the neural population as a function of the stimulus (i.e., the tuning curves). Specifically, the covariance matrix $\mathbf{\Sigma}(s)$ must take on the form

$$\mathbf{\Sigma}(s) = \mathbf{\Sigma}_0(s) + \mathbf{f}'(s)\mathbf{f}'(s)^T \tag{1.8}$$

where $\mathbf{\Sigma}_0(s)$ is a positive semi-definite matrix. Moreno-Bote et al. demonstrate that the differential correlations are the only correlations that can saturate the linear Fisher information. However, this holds only for rank-1 correlations (i.e., outer products) and only for the linear Fisher information as the measure of interest.

The form of differential correlations clearly motivates their name, due to the derivative in the outer product. A geometric viewpoint, however, provides a deeper understanding to why they take on their particular form. Differential correlations contain a direction in the covariance eigenspectrum that aligns with the derivative of the tuning curve. Why should such a direction saturate information? This is variability that lies along the stimulus manifold – i.e., variability that causes the stimulus to take on a different value. There is nothing a population can do to rid of variability that mimics a change in the stimulus. Adding neurons to the population will lead to decreasing gains in information, causing saturation. Thus, "shared input noise", or an noise that is carried by the stimulus into the system, is a source of differential correlations. Moreno-Bote et al. also specify suboptimal computations as a source of differential correlations.

Detection of differential correlations is difficult because assessing such scaling properties requires on the order of thousands of simultaneously recorded neurons. Recent advances in recording technologies have allowed experimentalists to record at the scales necessary to detect differential correlations. Multiple papers have come out recently confirming that differential correlations are prevalent in visual cortex [160, 94] and prefrontal cortex [23], establishing differential correlations as the dominant source of information saturation in the brain. Thus, these results imply that the blue line in Figure 1.3 is the most likely scenario for early sensory areas.

Sources of correlated variability

Thus far, we have motivated our discussions of correlated variability based off its detection in experiment and implication for decoding in theoretical analyses. However, we have not discussed why such structure arises in the first place, nor its biophysical sources. The magnitude of correlated variability can be large, depending on the brain region (correlations up

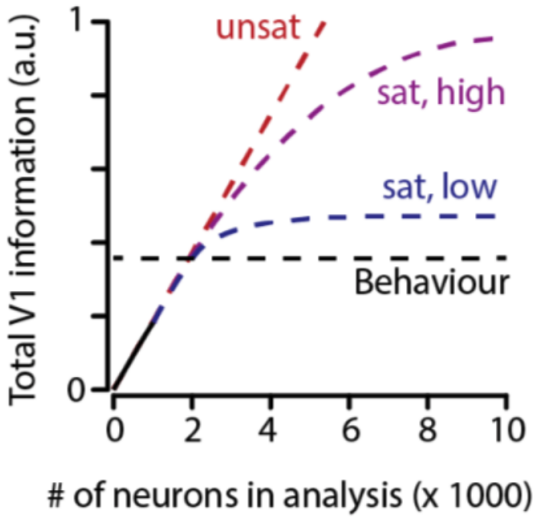


Figure 1.3: **Scaling of information in early sensory areas relative to behavioral performance.** Image taken from [127].

to 0.5 in some recordings) [51], implying that such structure could not arise simply due to chance. However, there is no single source of correlated variability that can easily explain such observations. Broadly, however, correlated variability can be thought of as arising from true variability sources (i.e., biological noise) [64] and recurrent dynamics in cortex that are modulated by state dependence [60].

There are variety of "true" noise sources in the brain, including shot noise in the retina, channel noise, stochastic synaptic vesicle release, thermal fluctuations, and others [64]. Many of these noise sources can be thought of as "private variability," or inherent to single neurons. However, to downstream neurons that often receive overlapping input from upstream neurons, this private variability becomes their "shared variability" [174]. This, in combination with common signal arising from the stimulus, or other computations, is a source of correlated variability in neural circuits. However, the stimulus and biological noise are not the only contributions to neural activity. Indeed, neural activity can be modulated by global fluctuations, such as arousal, attentional state, learning, task engagement, and others [60]. Thus, particularly in downstream processing, correlated variability is significantly impacted by recurrent dynamics.

As stated in the prior section, differential correlations are caused by shared input noise and suboptimal computations. This is well understood in the case of shared input noise: any noise carried by the stimulus cannot be averaged away, and thus will lead to information saturation [96]. Suboptimal computations can be understood broadly: if the algorithm by which a neural circuit performs a certain computation is incorrect, then information cannot increase without bound [28]. However, the manner in which suboptimal computations can arise on a mechanistic level via extraneous input or other upstream noise sources has not been explored fully. For example, *common noise*, or noise that is a shared input, but can be

manipulated by features of the neural circuit such as synaptic weighting, is also a prominent feature of neural systems, and can induce correlated variability. We explore how common noise impacts correlated variability in Chapter 2.

Correlated variability in phenomenological models of neural activity

Since correlated variability is a pervasive phenomenon in neural activity, its role is important to assess in the fitting of phenomenological models of neural activity. Generally, our goal is to attempt to relate important features of neural activity – such as external stimuli and functional coupling – to the generation of neural activity. We typically do this via a probabilistic model, which can capture some degrees of freedom in the neural activity, leaving the remaining variability to the stochastic component of the model. Then, when the models are fit to the data, their parameters can be interpreted to gain insight into the underlying biological processes that generated the data. The degree to which the assumptions about the controlled degrees of freedom align with reality affects how accurate the model is, and therefore the degree to which it provides accurate scientific conclusions. If correlated variability is not modeled explicitly, or the inference procedure is not robust to its presence, we may obtain biased parameter estimates, making the model unreliable.

Correlated variability is often neglected in such phenomenological models. Typically, it is not a phenomenon of interest, either because it is not relevant for the task at hand, or because introducing it into the model would impede parameter inference. Thus, building models and inference procedures that are either robust to correlated variability, or capture it explicitly, is necessary to ensure that we extract correct scientific conclusions from our models. For example, we consider inference procedures that explicitly consider stability as a criteria of importance (i.e., they are stable to the variability inherent in the data) in Chapter 4. Such inference procedures are more likely to give accurate parameter estimates, even when correlated variability exists in the data. On the other hand, we also consider modeling the correlated variability explicitly by treating it as an unobserved source of variability in Chapter 5. By modeling the correlated variability explicitly, we obtained different systems neuroscience conclusions about the datasets we examined.

1.3 Summary of Results

Heterogeneous synaptic weighting improves neural coding in the presence of common noise

The study of information-limiting correlations by [128] was landmark in its identification of the exact structure of correlated variability that produces information saturation in neural populations. The authors called this correlated variability *differential correlations*, gave its analytic form, and the conditions under which it may arise: shared input noise and

suboptimal computations. Differential correlations almost surely arise in neural populations due to the unavoidable fact that the stimulus will carry its own noise into the system, which qualifies as a form of "shared input noise." However, the authors did not discuss in further detail other conditions in which differential correlations may arise.

Downstream excitatory neurons can receive up to thousands of inputs, some of which may serve as a representation of the stimulus, while others may be extraneous input not relevant for decoding. It is not practical for the neuron to zero out the extraneous input, because such input may be relevant on another trial or serve other downstream computation. Thus, this extraneous input, for purposes of decoding the stimulus, is a "noise source." However, it is not "shared input noise" as Moreno-Bote et al. envision it, because it can be manipulated by the neuron via its synaptic weighting. We call such noise sources "common noise." How do common noise sources generate correlated variability, impact decoding, and fit into the theory of differential correlations? This project sought to answer these questions.

We examined common noise in a simple linear-nonlinear model and assessed the networks decoding ability under various synaptic weight configurations. We found that diverse synaptic weighting improves neural decoding in the presence of a common noise source, even if the weighting *amplifies* the common noise. On the other hand, homogeneous synaptic weighting that does not amplify the common noise will induce differential correlations. We also found that, in the nonlinear regime, such improvements only hold up to a certain level of heterogeneity, beyond which it produces worse coding performance. Lastly, we characterized how the relationship between private variability and correlated variability impacts the optimal amount of synaptic heterogeneity. Together, our results shed light on how correlated variability can be induced via common noise sources, characterizes the structure of this variability, and assesses its impact on neural coding.

Research Narrative

This project began largely as exploratory work inspired by Moreno-Bote et al.'s differential correlations paper [128]. We began by attempting to induce differential correlations in a very simple linear-nonlinear network. Since Moreno-Bote et al. claimed that shared input noise would induce differential correlations, we added a noise term as an input to the network. Crucially (and somewhat naively), the noise term was allowed to be weighted by the network. This can be seen as a somewhat odd choice, as noise terms are not thought to be manipulated by a system. But when we found that the Fisher information *improved* when the noise was weighted more heavily, we were surprised enough to keep investigating.

We interpreted the initial results as synaptic weighting *preventing* differential correlations. We presented these results, along with this understanding of it, at Cosyne. However, we received criticism of this interpretation, particularly by the authors of the differential correlation paper. They argued that "shared input noise" cannot be manipulated by the system, and thus our network had not "prevented" differential correlations. Ultimately, our results were interesting, but required reframing in order for the community to find them interesting.

The fundamental result of the paper – that heterogeneous synaptic weighting improves neural coding in the presence of common noise – was obtained within a few weeks of the initial exploratory research. It took two more years of work to build a coherent narrative around this result, in particular identifying "common noise" as the biophysical property of interest, as well as framing the results within the interplay of shared and private variability. This narrative building shaped additional analyses to perform, which led to the remaining results of the paper. Lastly, we bolstered our analysis by including additional numerical experiments examining the mutual information.

Optimal correlated variability is biologically implausible

Studies that examine the benefit or harm of correlated variability in population coding require a benchmark against which to compare the observed correlated variability structure. The standard benchmark is the null model with *no* correlated variability, while retaining the per-neuron variability. In a covariance matrix, this can be achieved by setting all off-diagonal components equal to zero. In neural data, this is typically done by *shuffling* neural responses across trials. Shuffling maintains the first-order structure (the average responses) while destroying any second-order structure (pairwise correlations).

Posing the question of whether correlated variability *benefits* neural coding implies that observed correlated variability might be an intentional structure of the neural system. That is, biological systems, by virtue of optimizing their neural activity for decoding, have purposely shaped the observed correlated variability structure. Ultimately, when framed in the context of efficient coding, this becomes a question of optimality. While the shuffle null model tests against the benchmark of *no* correlated variability, it does not speak to whether correlated variability is optimal, or approaches optimality. Thus, it is possible that correlated variability improves coding relative to no correlations, but is still suboptimal relative to what could be achievable. Such an observation would weaken the hypothesis that correlated variability is purposely structured to improve coding fidelity.

Thus, we sought to assess whether correlated variability in neural systems is optimal. Since the shuffle null model is not sufficient for answering this question, we proposed two new null models: the rotation null model and the factor analysis null model. Rather than considering no correlational structure, the rotation null model rotates the covariance matrix in the neural space. Thus, it preserves the eigenspectrum of the observed neural activity, but allows it to take any orientation in the neural space. The factor analysis model similarly rotates components of the observed covariance matrix. However, it relaxes the assumption that the neural system can rotate the entire covariance matrix, and instead assumes it can only shape a *shared* component, while assuming there is a private variability component inherent to the activity of each neuron.

We found that, across diverse datasets, the rotation and factor analysis null models suggest that neural activity is highly suboptimal. Furthermore, the sub-optimality worsens with increasing circuit size. This is in contrast to the shuffle null model, which suggests that neural circuits are highly optimal, or close to optimal, at lower population sizes. To

better understand the surprising result that neural populations are highly suboptimal, we compared the structure of observed correlated variability to that of the optimal correlated variability structure under each null model.

We assessed the observed correlated variability structure using two measures of biological plausibility: the marginal distribution of the neural activity and the Fano factor. In the former, we assessed whether the optimal marginal distributions of the neural activity was similar to that of the observed marginal distributions. In the latter, we compared the optimal Fano factors to the observed Fano factors. We found that, in both cases, the observed neural activity was closest to optimal when the optimal Fano factor and marginal distributions were achievable. When they were not – e.g., it would result in negative firing rates, or Fano factors that were biologically implausible – then the neural activity was highly suboptimal.

Together, our results demonstrate that neural circuits may be biophysically limited in achieving optimal correlated variability arrangements. Thus, while correlated variability may be structured in a way to improve neural coding relative to independent variability, it is still limited to being structured in a highly suboptimal fashion.

Research Narrative

This project began when Jesse Livezey made the observation that shuffling neural responses may be a poor null model for answering questions about optimality. Instead, rotating the covariance would be a more suitable null model. He also suggested that alternative measures other than the linear Fisher information – such as the symmetric KL-divergence – could be better suited for decoding scenarios in which the underlying stimulus was categorical, rather than continuous. Thus, the initial version of this project considered both new null models and new metrics for examining correlated variability.

The initial results demonstrating that the rotation null model indicated the sub-optimality of the neural code were obtained by Jesse relatively early in the project. They, along with some theoretical predictions about the optimal orientation of the symmetric KL-divergence, were presented at Cosyne in 2019.

Crafting a narrative for this observation was difficult, because the observed neural data performed *so* poorly according to the null model. It took a couple more years to flesh out the results on additional datasets (with the same findings) in addition to performing large scale analyses that demonstrated that our initial observations were sound. Furthermore, we realized that in the cases where the sub-optimality was particularly bad, the optimal results provided by the rotation null model didn't really make sense biologically. So, the narrative became apparent: optimal noise correlations are biologically unattainable. Thus far, the set of results was becoming rather large, and we decided to remove the exploration of categorical stimuli as a set of main results, since they no longer fit thematically.

At the same time, Jesse suggested an additional null model – the factor analysis null model – which was more believable null model than the rotation null model, and provided similar results. This fleshed out both the narrative and results of the project, and we were quickly able to finish the results and figures. At the time of this writing, Jesse has suggested

an additional null model – one that provides equal density for any off-diagonal *correlation structure* – and we are currently attempting to fit this null model in our pipeline.

Improved inference in coupling, encoding, and decoding models and its consequence for neuroscientific interpretation

A central goal of systems neuroscience is to understand the relationships amongst constituent units in neural populations, and their modulation by external factors, using high-dimensional and stochastic neural recordings. Parametric statistical models (e.g., coupling, encoding, and decoding models), play an instrumental role in accomplishing this goal. However, extracting conclusions from a parametric model requires that it is fit using an inference algorithm capable of selecting the correct parameters and properly estimating their values. This is particularly difficult in neural data, which possesses an abundance of structure that may not be captured in a phenomenological model. Thus, it is crucial that an inference procedure is robust to additional structure it may not explicitly model in order for a model to be scientifically useful.

Correlated variability is pervasive in neural datasets. Standard systems neuroscience models, including coupling, tuning, and decoding models, do not generally capture the structure of correlated variability present in the data. In most cases, modeling it explicitly complicates model fitting. Thus, it is generally more worthwhile to fit a simpler model, but use an inference procedure that is stable to the data generating process.

Traditional approaches to parameter inference have been shown to suffer from failures in both selection and estimation. The recent development of algorithms that ameliorate these deficiencies raises the question of whether past work relying on such inference procedures have produced inaccurate systems neuroscience models, thereby impairing their interpretation.

We used algorithms based on Union of Intersections, a statistical inference framework based on stability principles, capable of improved selection and estimation. We fit functional coupling, encoding, and decoding models across a battery of neural datasets using both UoI and baseline inference procedures, and compared the structure of their fitted parameters. Across recording modality, brain region, and task, we found that UoI inferred models with increased sparsity, improved stability, and qualitatively different parameter distributions, while maintaining predictive performance. We obtained highly sparse functional coupling networks with substantially different community structure, more parsimonious encoding models, and decoding models that relied on fewer single-units. Together, these results demonstrate that improved parameter inference, achieved via UoI, reshapes interpretation in diverse neuroscience contexts.

Research Narrative

The Union of Intersections (UoI) framework had already been developed prior to this project's onset. We needed to use UoI early on in the triangular model project (see below) to perform

selection in some early synthetic experiments. A gap in the development of UoI was a paper that explicitly explored its application to diverse neuroscience datasets. Since we were exploring tuning and coupling models in the triangular model project, this was a natural project to explore in parallel. Thus, this project largely consisted of gathering a wide array of datasets, parsing them, fitting models with UoI, and comparing to baseline procedures.

At the same time, this project pushed forward the development of the UoI script into a full software package. Thus, our paper on PyUoI, was finished in parallel to this project. We developed additional fitting procedures, including $\text{UoI}_{\text{Logistic}}$ and $\text{UoI}_{\text{Poisson}}$. This allowed us to proceed with fitting additional neuroscience models, such as spiking coupling models and decoding models. Once we obtained fitted models, we had to explore how to interpret the models, which included a range of secondary analyses.

The narrative of this project waffled between a strict methods paper on UoI applied to neural data and a more general paper focusing on improving inference in parametric neuroscience models. The idea of the second narrative was that an inference procedure that exhibits improved inference will ultimately change the neuroscience interpretation, and assessing these changes in interpretation is important. It took a couple submissions to different journals to eventually pin down the final narrative, which was a largely methods based paper (it was accepted to Journal of Neuroscience Methods), but with a focus on the interpretation of the fitted models.

Overall, this project (other than PyUoI) was the most straightforward in its research narrative. We set out with a clear research goal, and the end product looked similar to what we expected it might be. As is often the case, many of the major results were obtained relatively quickly. It took longer to sort out the details of the narrative (i.e., focusing on interpretation), which became more clear as the results and figures solidified.

Identifying and mitigating statistical biases in neural models of tuning and functional coupling

Phenomenological models of neural activity allow us to assess how important neurobiological factors relate to the generation of neural activity. In systems neuroscience, two fundamental factors of interest include tuning, or how neurons respond to external stimuli, and functional coupling, or how neurons respond to the activities of neighboring neurons. These two factors can be thought of as external, or exogenous to the neural system, and internal, or endogenous to the neural system. Statistical models, such as generalized linear models, have been used to describe neural responses with tuning and functional coupling, achieving high predictive accuracy while using fitted parameters to provide insight into which factors are important and their relative importance. Furthermore, past studies have demonstrated that the inclusion of coupling decreases the magnitudes of the tuning parameters. This effect has been interpreted as an "explaining away" of tuning, i.e., a decrease in the relative importance of external factors for the generation of neural activity.

However, extracting conclusions about neural activity from model parameters requires

that their estimates are unbiased. For example, models that incorporate both tuning and coupling fail to account for unobserved activity, which is a source of correlated variability in neural activity. Thus, not only do these models not reproduce the phenomenon of correlated variability, but their parameter estimates are likely biased due to model incompleteness. Such parameter biases may jeopardize past conclusions about neural activity.

In this project, we proposed the triangular model, a latent-variable model of neural activity that is more complete than the tuning and coupling model. In particular, it models unobserved activity using a low-dimensional latent state, which reproduces the phenomenon of noise correlations. Additionally, it directly models the data generation process of the coupling neurons, allowing tuning to influence the "target" neuron directly and indirectly via the coupled neurons.

We demonstrated that parameter estimates obtained by fitting the tuning and coupling model to data generated from the triangular model are biased due to the fact that the unobserved activity is not modeled. We characterized this bias as the *simultaneous equations bias*, or an *omitted variables bias*, previously studied in the econometrics literature. We further demonstrated that the triangular model is *structurally non-identifiable*, where infinite parameter configurations exist for each value of the likelihood. Both of these issues impede interpretability: the simultaneous equations bias prevents us from obtaining accurate parameter estimates, while structurally non-identifiability prevents us from having a unique parameter set for an optimized model.

We proposed an inference procedure that solves both issues in the triangular model. First, we show that inducing sufficient sparsity – where some of the parameters are exactly zero – mitigates the identifiability bias. Second, using the expectation-maximization algorithm, we develop an inference procedure that fits the triangular model to the data, thereby sidestepping the simultaneous equations bias. This inference procedure can either induce sparsity on its own by imposing ℓ_1 penalties on the relevant parameters, or utilize a selection profile obtained by an alternative method. We demonstrate that our inference procedure is capable of unbiased estimation in synthetic data. Furthermore, we characterize the scenarios in which inference breaks down, shedding light on parameter inference in the triangular model at large.

Lastly, we applied our inference procedure to multiple neural datasets, finding that it resulted in noticeable changes relative to a tuning and coupling model. Most strikingly, it elevated the tuning modulation relative to the coupling model, implying that some of the previously observed "explaining away" may have due to the simultaneous equations bias. At the same time, it did not elevate tuning modulations to that of the tuning model alone, implying that coupling, and unobserved activity, does explain away tuning to some degree. Together, our results shed light on the simultaneous equations bias and structural non-identifiability in the parametric models for systems neuroscience.

Research Narrative

This project was motivated by a paper examining models of tuning and functional coupling [188]. In this paper, Stevenson et al. find that the inclusion of functional coupling in a tuning model "explains away" tuning. That is, by combining both tuning and coupling features into a single model, the relative importance of tuning for predicting neural activity is downplayed, since the neighboring neurons shared some of that explanatory power. The "relative importance" was assessed by the magnitudes of the fitted parameter values.

Kris Bouchard had two concerns with this paper. First, the "tuning and coupling" model used by the authors neglected to consider that tuning influences both the "target" neuron and "non-target" neuron jointly (i.e., their graphical model was incomplete). Second, they enforced sparsity by applying an ℓ_1 penalty to the parameters of the problem, which is known to lead to shrinkage. Thus, it is unclear whether some of the "explaining away" may have been artificial due to shrinkage. These critiques require a new graphical model as well as a new inference procedure, which was the starting point for the project.

We began by generating synthetic data from the triangular model (though we did not call it that at the time) in various parameter regimes to assess the parameter fits. At some point, Kris had the idea of inducing correlated variability in the triangular model, by having the error terms be positively correlated with each other. This led to noticeable biases in the tuning parameters, particularly in the downward direction. This led to the key, motivating observation: the decrease in tuning parameters observed in the previous paper could simply be a byproduct of the bias that we were observing.

We found that this bias was studied by the econometrics community. Initially, inspired by some approaches in the econometrics literature, we developed an ad-hoc procedure called Iterated Two-Stage Factor Analysis (ITSFA), that seemed to perform well in correcting for the bias in synthetic data. We also applied ITSFA to neural data, finding elevation of tuning modulations in some cases, and not in others. We hypothesized that the heterogeneity in tuning modulation changes corresponded to the distribution of noise correlations, with some initial evidence.

While our work was being reviewed at NeurIPS (and eventually rejected), we began developing an expectation-maximization approach to perform parameter inference in the problem. This required recasting the triangular model as a latent variable problem (it initially was not quite so), and performing a large amount of algebra to develop the rules. We found that, when implemented, EM didn't quite work, which puzzled us. This led to our discovery that the triangular model was not identifiable – that is, we analytically derived a transformation that maintained the log-likelihood for any parameter configuration. The structural non-identifiability was challenging for inference, because during optimization, we could end up anywhere on an identifiability family.

Our initial approach to handle this was to develop a "constraint" that we could apply to the parameter fits to obtain a desirable solution. The intention was that we would perform parameter optimization, apply the constraint as a post-hoc procedure, and end up with a final parameter solution. The constraint would satisfy a desirable property about the neural

system. We developed a whole suite of constraints – some that worked well, and some that didn't. At the end of the day, though, there was no constraint that worked exceptionally well, nor were they desirable or flexible enough to apply outside of the triangular model context.

At some point, we realized that the non-zero parameters were *not* preserved under an identifiability transform. Thus, having a sparse set of parameters could serve as the desirable constraint. We were able to perform some experiments initially that supported this hypothesis, and it was a desirable, general constraint because of our work described in the previous section. It wasn't until we were able to prove the conditions under which sparsity could alleviate the identifiability issue that we were confident that we could proceed forward.

To utilize the sparsity constraint, we needed an inference procedure capable of setting parameters exactly equal to zero during optimization. Our initial approach was to simply include an ℓ_1 penalty on the EM optimizer. The tricky aspect was that we needed two penalties: one for the tuning parameters, and one for the coupling parameters. Building, scaling, and testing this optimizer took a large amount of time. Furthermore, running it on a large scale synthetic experiment produced middling results, and it broke down easily in more difficult parameter inference regimes.

It was at this point that it became clear that the selection profile was very important in the problem, with dramatic impacts on the values of the estimated parameters. This motivated a change of how we formulated the inference procedure: as in the UoI case, we needed to separate selection and estimation. We had previously developed several ad-hoc selection procedures. Thus, on the synthetic side, we could examine how a wide range of selection profiles perform in terms of triangular model inference, relative to the tuning and coupling model. This project is currently still in progress at the time of this writing.

Chapter 2

Heterogeneous synaptic weighting improves neural coding in the presence of common noise

Chapter Co-authors
Jesse A. Livezey
Michael R. DeWeese

Neural circuits receive thousands of common inputs, some of which may be extraneous to the stimulus at hand. These noise sources, which we call common noise, can induce correlated variability, particularly through how they are shaped by features of neural computation such as synaptic weighting. Their impact on correlated variability and the coding fidelity of a neural population is not well understood. How do these common noise sources interact with synaptic weighting to produce correlated variability, and thus impact decoding performance? More simply, can neural circuits overcome common noise sources via their synaptic weighting? This chapter seeks to answer these questions.

2.1 Introduction

Variability is a prominent feature of many neural systems – neural responses to repeated presentations of the same external stimulus will typically vary from trial to trial [174]. Furthermore, neural variability often exhibits pairwise correlations, so that pairs of neurons are more (or less) likely to be co-active than they would be by chance if their fluctuations in activity to a repeated stimulus were independent. These so-called "noise correlations" (which we also refer to as "shared variability") have been observed throughout the cortex [12, 51], and their presence has important implications for neural coding [226, 2].

If the activities of individual neurons are driven by a stimulus shared by all neurons but corrupted by noise that is independent for each neuron (so-called "private variability"),

then the signal can be recovered by simply averaging the activity across the population [2, 119]. If instead some variability is shared across neurons (*i.e.*, there are noise correlations), naively averaging the activity across the population will not necessarily recover the signal, no matter how large the population [226]. An abundance of theoretical work has explored how shared variability can be either beneficial or detrimental to the fidelity of a population code (relative to the null model of only private variability amongst the neurons), depending on its structure and relationship with the tuning properties of the neural population [226, 2, 217, 181, 13, 52, 45, 62, 128, 136].

One general conclusion of this work highlights the importance of the geometric relationship between noise correlations and a neural population's signal correlations [12, 84]. To illustrate this, the mean responses of a neural population across a variety of stimuli (*i.e.*, those responses represented by receptive fields or tuning curves) can be examined in the neural space (Fig. 2.1a, black curves). The correlations amongst the mean responses for different stimuli specify the signal correlations for a neural population [12]. Private variability exhibits no correlational structure, and thus its relationship with the signal correlations is determined by the mean neural activity and the individual variances (Fig. 2.1a, left). Shared variability, however, may reshape neural activity to lie, for example, orthogonal to the mean response curve (Fig. 2.1a, middle). In the case of Figure 2.1a, middle, neural coding is improved (relative to private variability), because the variability occupies regions of the neural space that are not traversed by the mean response curve [126]. Shared variability can also harm performance, however. Recent work has identified *differential correlations* – those that are proportional to the products of the derivatives of tuning functions (Fig. 2.1a, right) – as particularly harmful to the performance of a population code [128]. While differential correlations are consequential, they may serve as a small contribution to a population's total shared variability, leaving "non-differential correlations" as the dominant component of shared variability [106, 127, 93].

The sources of neural variability – and their respective contributions to the private and shared components – will have a significant impact on shaping the geometry of the population's correlational structure, and therefore its coding ability [37]. For example, private sources of variability such as channel noise or stochastic synaptic vesicle release could be averaged out by a downstream neuron receiving input from the population [64]. However, sources of variability shared across neurons – such as the variability of pre-synaptic spike trains from neurons that synapse onto multiple neurons – would introduce shared variability and place different constraints on a neural code [174, 96]. In particular, differential correlations are typically induced by shared input noise (*i.e.*, noise carried by a stimulus) or suboptimal computations [29, 96].

Past work has examined the contributions of private and shared sources to variability in cortex [8, 59]. Specifically, by partitioning sub-threshold variability of a neural population into private components (synaptic, thermal, channel noise in the dendrites, and other local sources of variability) and shared components (variability induced by afferent connections), it was found that the private component of the total variability was quite small, while the shared component can be much larger (Fig. 2.1b and c). Thus, neural populations must

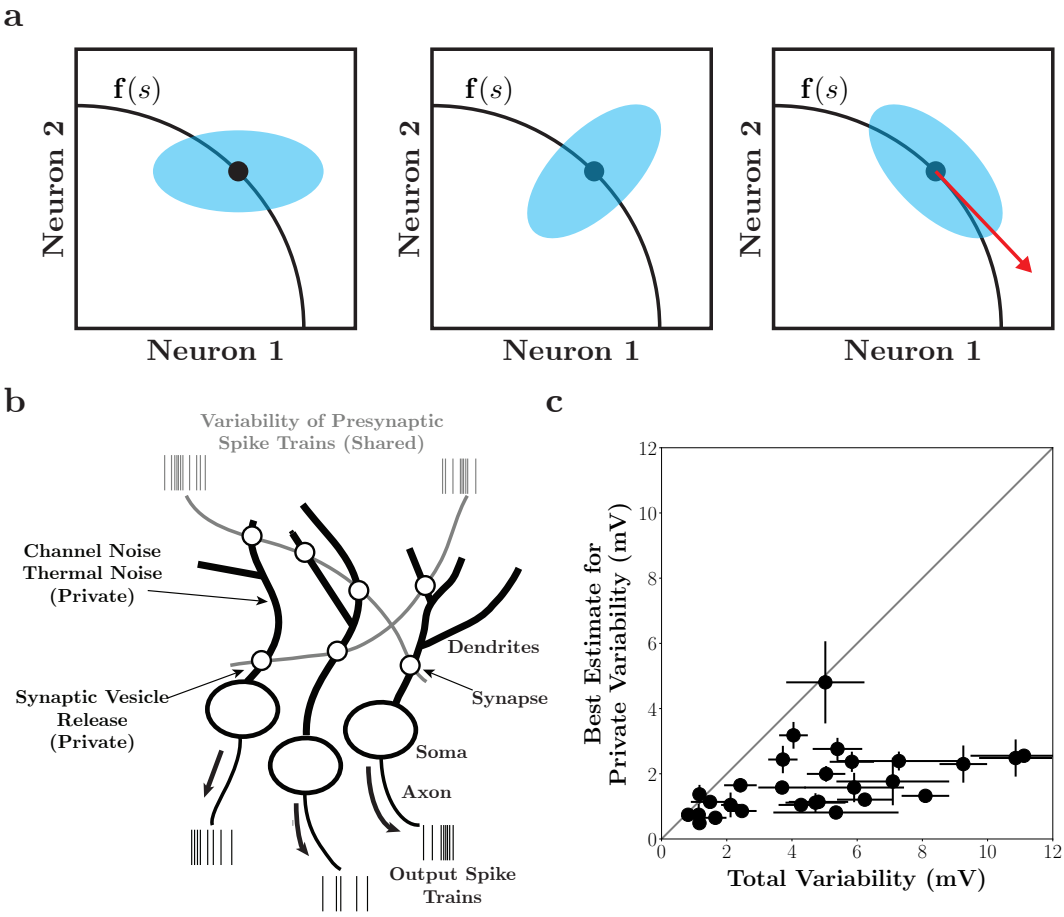


Figure 2.1: **(a)** The geometric relationship between neural activity and shared variability. Black curves denote mean responses to different stimuli. Variability for a specific stimulus (black dot) may be private (left), shared (middle), or take on the structure of differential correlations (right). The red arrow represents the tangent direction of the mean stimulus response. **(b)** Schematic of the types of variability that a neural population can encounter. The variability of a neural population contains both private components (*e.g.*, synaptic vesicle release, channel noise, thermal noise, etc.) and shared components (*e.g.*, variability of pre-synaptic spike trains, shared input noise). Shared variability can be induced by the variability of afferent connections (which is shared across a postsynaptic population) or inherited from the stimulus itself. Furthermore, shared variability is shaped by synaptic weighting. **(c)** Estimates of the private variability contributions to the total variability of neurons ($N = 28$) recorded from auditory cortex of anesthetized rats. Diagonal line indicates the identity. Figure reproduced from [59].

contend with the large shared component of a neuron's variability. The incoming structure of shared variability and its subsequent shaping by the computation of a neural population

is an important consideration for evaluating the strength of a neural code [228].

Moreno-Bote et al. demonstrated that shared input noise is detrimental to the fidelity of a population code [128]. Here, we instead examine sources of shared variability which do not necessarily result in differential correlations (*i.e.*, they do not appear as shared input noise) and thus can be manipulated by features of neural computation such as synaptic weighting. We refer to these noise sources as "common noise" to distinguish them from the aforementioned special case of "shared input noise" [203, 110]. For example, a common noise source could include an upstream neuron whose action potentials are "noisy" in the sense that they are unimportant for the computation of the current stimulus. Common noise, because it is manipulated by synaptic weighting, can serve as a source of nondifferential correlations (*e.g.*, Fig. 2.1a, middle), thereby having either a beneficial or harmful impact on the strength of the population code. We aim to better elucidate the nature of this impact.

We consider a linear-nonlinear architecture [144, 97, 151] and explore how its neural representation is impacted by both a common source of variability and private noise sources affecting individual neurons independently. This simple architecture allowed us to analytically assess coding ability using both Fisher information [2, 217, 213, 214], and Shannon mutual information. We evaluated the coding fidelity of both the linear representation and the nonlinear representation after a quadratic nonlinearity as a function of the distribution of synaptic weights that shape the shared variability within the representations [3, 63, 165, 142]. We find that the linear stage representation's coding fidelity improves with diverse synaptic weighting, even if the weighting amplifies the common noise in the neural circuit. Meanwhile, the nonlinear stage representation also benefits from diverse synaptic weighting in a regime where common noise may be amplified, but not too strongly. Moreover, we found that the distribution of synaptic weights that optimized the network's performance depended strongly on the relative amount of private and shared variability. In particular, the neural circuit's coding fidelity benefits from diverse synaptic weighting when shared variability is the dominant contribution to the variability. Together, our results highlight the importance of diverse synaptic weighting when a neural circuit must contend with sources of common noise.

2.2 Methods

The code used to conduct the analyses described in this paper is publicly available on Github [161].

Network Architecture

We consider the linear-nonlinear architecture depicted in Figure 2.2. The inputs to the network consist of a stimulus s along with common (Gaussian) noise ξ_C. The N neurons in the network take a linear combination of the inputs and are further corrupted by i.i.d.

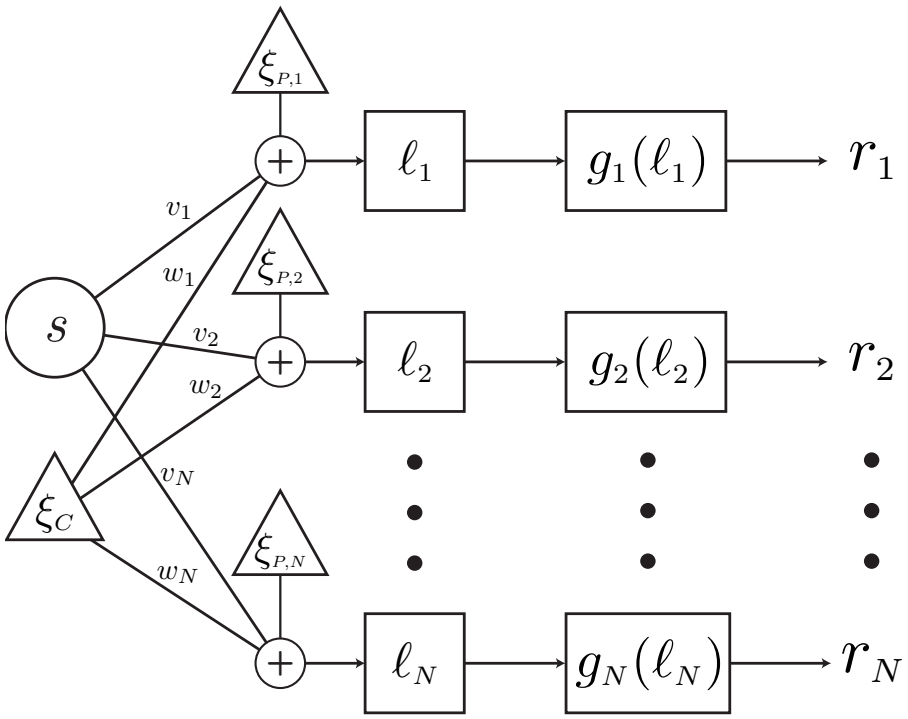


Figure 2.2: Linear-nonlinear Network Architecture. The network takes as its inputs a stimulus s and common noise ξ_C. A linear combination of these quantities is corrupted by individual private noises $\xi_{P,i}$. The output of this linear stage is then passed through a nonlinearity $g_i(\ell)$ to produce a "firing rate" r_i. The weights for the linear stage of the network, v_i and w_i, can be thought of as synaptic weighting. Importantly, the common noise is distinct from shared input noise because it is manipulated by the synaptic weighting.

private Gaussian noise. Thus, the output of the linear stage for the ith neuron is

$$\ell_i = v_i s + w_i \sigma_C \xi_C + \sigma_P \xi_{P,i}, \tag{2.1}$$

where $\xi_{P,i}$ is the private noise, v_i and w_i are the weights, and the common and private noise terms are scaled by positive constants σ_C and σ_P. The noisy linear combination is passed through a nonlinearity $g_i(\ell_i)$ whose output r_i can be thought of as a firing rate.

Thus, the network-wide computation is given by

$$\mathbf{r} = \mathbf{g}(\mathbf{v}s + \mathbf{w}\sigma_C \xi_C + \sigma_P \boldsymbol{\xi}_P) \tag{2.2}$$

where $\mathbf{g}(\boldsymbol{\ell})$ is an element-wise application of the network nonlinearity.

Measures of Coding Strength

In order to assess the fidelity of the population code represented by $\boldsymbol{\ell}$ or $\mathbf{r}$, we turn to the Fisher information and the Shannon mutual information [53]. The former has largely been

utilized in the context of sensory decoding and correlated variability [2, 12, 106] while the latter has been well studied in the context of efficient coding [10, 22, 30, 156].

The Fisher information sets a limit by which the readout of a population code can determine the value of the stimulus. Formally, it sets a lower bound to the variance of an unbiased estimator for the stimulus. In terms of the network architecture, the Fisher information of the representation $\mathbf{r}$ (or $\boldsymbol{\ell}$) quantifies how well s can be decoded given the representation. For Gaussian noise models with stimulus-independent covariance, the Fisher information is equal to the linear Fisher information (LFI):

$$I_{LFI}(s) = \frac{\partial \mathbf{f}(s)}{\partial s}^T \boldsymbol{\Sigma}^{-1}(s) \frac{\partial \mathbf{f}(s)}{\partial s} \tag{2.3}$$

where $\mathbf{f}(s)$ and $\boldsymbol{\Sigma}(s)$ are the mean and covariance of the response (here $\mathbf{r}$ or $\boldsymbol{\ell}$) to the stimulus s. In other cases, the LFI serves as a lower bound for the Fisher information and thus is a useful proxy when the Fisher information is challenging to calculate analytically. The estimator for I_{LFI} is the locally optimal linear estimator [106].

The Shannon mutual information quantifies the reduction in uncertainty of one random variable given knowledge of another

$$I[s, \mathbf{f}] = \int ds d\mathbf{f}\ p(s, \mathbf{f}) \log \left(\frac{p(s, \mathbf{f})}{p(s)p(\mathbf{f})} \right). \tag{2.4}$$

Earlier work demonstrated that the Fisher information provides a lower bound for the Shannon mutual information in the case of Gaussian noise [39]. However, more recent work has revealed that the relationship between the two is more nuanced, particularly in the cases where the noise model is non-Gaussian [211]. Thus, we supplement our assessment of the network's coding ability by measuring the mutual information, $I[s, \mathbf{r}]$, between the neural representation $\mathbf{r}$ and the stimulus s. As with the Fisher information, the mutual information is often intractable, but fortunately can be estimated from data. Specifically, we will employ the estimator developed by Kraskov and colleagues, which utilizes entropy estimates from k-nearest neighbor distances [108].

Structured Weights

The measures of coding strength are a function of the weights that shape the interaction of the stimulus and noise in the network. Thus, the choice of the synaptic weight distribution impacts the calculation of these quantities. We first consider the case of "structured weights" in order to obtain analytical expressions for measures of coding strength. Structured weights take on the form

$$\mathbf{w} = \left(\underbrace{1 \cdots 1}_{N/k \text{ times}} \quad \underbrace{2 \cdots 2}_{N/k \text{ times}} \quad \cdots \quad \underbrace{k \cdots k}_{N/k \text{ times}} \right)^T. \tag{2.5}$$

Specifically, the structured weight vectors are parameterized by an integer k which divides the N weights into k homogeneous groups. The weights across the groups span the positive integers up to k. Importantly, larger k will only increase the weights in the vector. Thus, in the above scheme, increased "diversity" can only be achieved by increasing k, which will invariably result in an amplification of the signal to which the weight vector is applied. In the case that k does not evenly divide N, each group is repeated $\lceil N/k \rceil$ times, except the last group, which is only repeated $N-(N-1)\cdot\lceil N/k \rceil$ times (*i.e.*, the last group is truncated to ensure the weight vector is of size N).

Additionally, we consider cases in which k is of order N, *e.g.*, $k = N/2$. Allowing k to grow with N ensures that typical values for the weights grow with the population size. This contrasts with the case in which k is a constant, such as $k = 4$, which sets a maximum weight value independent of the population size.

Unstructured Weights

While the structured weights allow for analytical results, they possess an unrealistic distribution of synaptic weighting. Thus, we also consider the case of "unstructured weights," in which the synaptic weights are drawn from some parameterized probability distribution:

$$\mathbf{v} \sim p(\mathbf{v}; \theta_{\mathbf{v}}); \;\; \mathbf{w} \sim p(\mathbf{w}; \theta_{\mathbf{w}}). \tag{2.6}$$

We calculate both information theoretic quantities over many random draws from these distributions, and observe how these quantities behave as some subset of the parameters θ are varied. In particular, we focus on the lognormal distribution [90], which has been found to describe the distribution of synaptic weights well in slice electrophysiology [183, 166]. Specifically, the weights take on the form

$$\mathbf{w} \sim \Delta + \text{Lognormal}(\mu, \sigma), \tag{2.7}$$

where $\Delta > 0$. For a lognormal distribution, an increase in μ will increase the distribution's mean, median, and mode (Fig. 2.3e, inset). Thus, μ as a parameter acts similarly to k for the structured weights in that increased weight diversity must be accompanied by an increase in their magnitude.

2.3 Results

We consider the network's coding ability after both the linear stage ($\boldsymbol{\ell}$) and the nonlinear stage ($\mathbf{r}$). In other words, the linear stage can be considered the output of the network assuming each of the functions $g_i(\ell_i)$ is the identity. Furthermore, due to the data processing inequality, the qualitative conclusions we obtain from the linear stage should apply for any one-to-one nonlinearity.

Linear Stage

The Fisher information about the stimulus in the linear representation can be shown to be (see Appendix 2.5 for the derivation)

$$I_F(s) = \frac{1}{\sigma_P^2} \frac{(\sigma_P^2/\sigma_C^2)\,|\mathbf{v}|^2 + (|\mathbf{v}|^2|\mathbf{w}|^2 - (\mathbf{v}\cdot\mathbf{w})^2)}{(\sigma_P^2/\sigma_C^2) + |\mathbf{w}|^2} \tag{2.8}$$

$$= \frac{|\mathbf{v}|^2}{\sigma_P^2} \frac{(\sigma_P^2/\sigma_C^2) + |\mathbf{w}|^2 \sin^2\theta}{(\sigma_P^2/\sigma_C^2) + |\mathbf{w}|^2} \tag{2.9}$$

which is equivalent to the linear Fisher information in this case. In equation 2.9, θ refers to the angle between $\mathbf{v}$ and $\mathbf{w}$. The mutual information can be expressed as (see Appendix 2.5 for the derivation)

$$I[s, \boldsymbol{\ell}] = \frac{1}{2} \log\left[1 + \sigma_S^2 I_F(s)\right]. \tag{2.10}$$

For the case the mutual information, we have assumed the prior distribution for the stimulus is Gaussian with zero mean and variance σ_S^2.

Examining equation (2.9) reveals that increasing the norm of $\mathbf{v}$ without changing its direction (i.e., changing θ) will increase the Fisher information, while increasing the norm of $\mathbf{w}$ without changing its direction will either decrease or maintain information (since $0 \leq \sin^2\theta \leq 1$). Additionally, if $\mathbf{v}$ and $\mathbf{w}$ become more aligned while leaving their norms unchanged, the Fisher information will decrease (since $\sin^2\theta$ will decrease). This decrease in Fisher information is consistent with the observation that alignment of $\mathbf{v}$ and $\mathbf{w}$ will produce differential correlations. If $\mathbf{v}$ and $\mathbf{w}$ are changed in a way that modulates both their norm and direction, the impact on Fisher information is less transparent.

To better understand the Fisher information, we impose a parameterized structure on the weights that allows us to increase weight diversity without decreasing the magnitude of any of the weights. This weight parameterization, which we call the structured weights, is detailed in Section 2.2. We chose this parameterization for two reasons. First, we desired a scheme in which an increase in diversity must be accompanied by an amplification of common noise. We chose this behavior so that any improvement in coding ability can only be explained by the increase in diversity, rather than a potential decrease in common noise. Secondly, we desired analytic expressions for the Fisher information as a function of population size, which is possible with this form of structured weights.

Under the structured weight parameterization, equations (2.8) and (2.10) can be explored by varying the choice of k for both $\mathbf{v}$ and $\mathbf{w}$ (we will refer to them as $k_\mathbf{v}$ and $k_\mathbf{w}$, respectively). It is simplest and most informative to examine these quantities by setting $k_\mathbf{v} = 1$ while allowing $k_\mathbf{w}$ to vary, as amplifying and diversifying $\mathbf{v}$ will only increase coding ability for predictable reasons (this is indeed the case for our network) [175, 62]. While increasing $k_\mathbf{w}$ will boost the overall amount of noise added to the neural population, it also changes the direction of the noise in the higher-dimensional neural space. Thus, while we might expect

that adding more noise in the system would hinder coding, the relationship between the directions of the noise and stimulus vectors in the neural space also plays a role.

We first consider how the Fisher information and mutual information are impacted by the choice of $k_{\mathbf{w}}$. In the structured regime, we have

$$|\mathbf{v}|^2 = N \tag{2.11}$$

$$\mathbf{v} \cdot \mathbf{w} = \frac{N}{k} \sum_{i=1}^{k} i = \frac{N(k+1)}{2} \tag{2.12}$$

$$|\mathbf{w}|^2 = \frac{N}{k} \sum_{i=1}^{k} i^2 = \frac{N(k+1)(2k+1)}{6}, \tag{2.13}$$

which allows us to rewrite equation (2.8) as

$$I_F(s) = I_F = \frac{N}{2\sigma_P^2} \frac{12(\sigma_P^2/\sigma_C^2) + N(k^2-1)}{6(\sigma_P^2/\sigma_C^2) + N(2k^2+3k+1)}. \tag{2.14}$$

The form of the mutual information follows directly from plugging equation (2.14) into equation (2.10).

The analytical expressions for the structured regime reveal the asymptotic behavior of the information quantities. Neither quantity saturates as a function of the number of neurons, N, except in the case of $k_{\mathbf{w}} = 1$ (Fig. 2.3a, b). In this regime, increasing the population size of the system also enhances coding fidelity. Furthermore, both quantities are monotonically increasing functions of the common noise synaptic heterogeneity, $k_{\mathbf{w}}$ (Fig. 2.3c, d), implying that decoding is enhanced despite the fact that the amplitude of the common noise is magnified for larger $k_{\mathbf{w}}$. Our analytical results show linear and logarithmic growth for the Fisher and mutual information, respectively, as one might expect in the case of Gaussian noise [39]. These qualitative results hold for essentially any choice of $(\sigma_S, \sigma_P, \sigma_C)$.

In the case of $k_{\mathbf{w}} = 1$, the signal and common noise are aligned perfectly in the neural representation. Thus, the common noise becomes equivalent in form to shared input noise. As a consequence, we observe the saturation of both Fisher information and mutual information as a function of the neural population. This saturation implies the existence of differential correlations, consistent with the observation that information-limiting correlations occur under the presence of shared input noise [96].

The structured weight distribution described above allows us to derive analytical results, but the limitation to only a fixed number of discrete synaptic weight values is not realistic for biological networks. Thus, we utilize unstructured weights, described in Section 2.2, in which the synaptic weights are drawn from a lognormal distribution. In this case, we estimate the linear Fisher information and the mutual information over many random draws according to $w_i \sim \Delta + \text{Lognormal}(\mu, \sigma^2)$. We are primarily concerned with varying μ, as an increase in this quantity uniformly increases the mean, median, and mode of the lognormal distribution (Fig. 2.3e, inset), akin to increasing $k_{\mathbf{w}}$ for the structured weights.

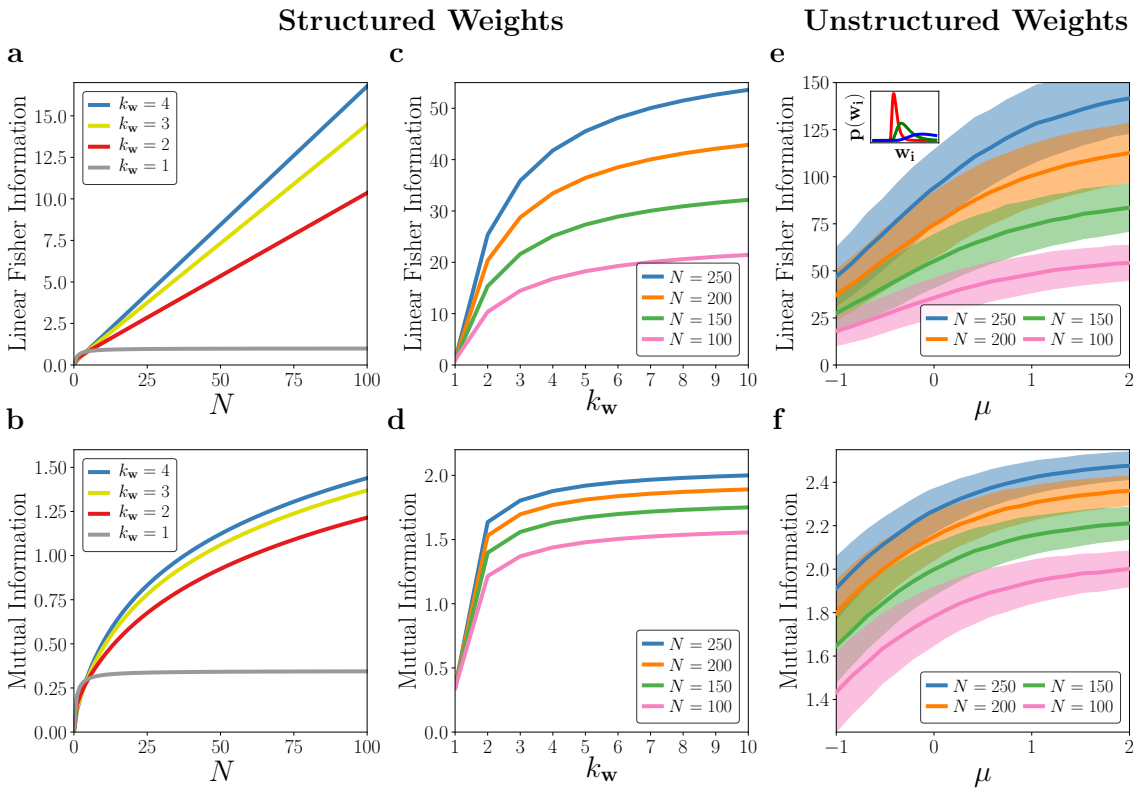


Figure 2.3: Network coding performance of the linear stage representation. Here, the noise variances are $\sigma_P^2 = \sigma_C^2 = 1$. Fisher information is shown on the top row while mutual information is shown on the bottom row. **(a)**, **(b)** Structured weights. Linear Fisher Information and Mutual Information are shown as a function of the population size, N, across different levels of weight heterogeneity, $k_{\mathbf{w}}$ (indicated by color). **(c)**, **(d)** Linear Fisher Information and Mutual Information are shown as a function of weight heterogeneity, $k_{\mathbf{w}}$, for various population sizes, N. **(e)**, **(f)** Unstructured weights. Linear Fisher Information and Mutual Information are shown as a function of the mean of the lognormal distribution used to draw common noise synaptic weights. Information quantities are calculated across 1000 random drawings of weights: solid lines depict the means while the shaded region indicates one standard deviation. Inset: the distribution of weights for various choices of μ. Increasing μ shifts the distribution to the right, increasing heterogeneity.

Our numerical analysis demonstrates that increasing μ increases the average Fisher information and average mutual information across population sizes (Fig. 2.3e, f: bold lines). In addition, the benefits of larger weight diversity are felt more strongly by larger populations (Fig. 2.3e, f: different colors).

In the structured weight regime, our analytical results show that weight heterogeneity can ameliorate the harmful effects of *additional* information-limiting correlations induced by common noise mimicking shared input noise. They do not imply that weight heterogene-

ity prevents differential correlations, as the common noise in this model is manipulated by synaptic weighting, in contrast with true shared input noise. For unstructured weights, we once again observe that larger heterogeneity affords the network improved coding performance, despite the increased noise in the system. Together, these results show that linear networks could manipulate common noise to prevent it from causing induced differential correlations. However, neural circuits, which must perform other computations that may dictate the structure of the weights on the common noise inputs, can still achieve good decoding performance provided that the circuits' synaptic weights are heterogeneous.

Quadratic Nonlinearity

We next consider the performance of the network after a quadratic nonlinearity $g_i(x) = x^2$ for all neurons i. This nonlinearity has been used in a neural network model to perform quadratic discriminant analysis [142] and as a transfer function in complex cell models [3, 63, 165]. Furthermore, we chose this nonlinearity because we were able to calculate the linear Fisher information analytically (as an approximation to the Fisher information). See Appendix 5.3 for a numerical analysis with an exponential nonlinearity. However, the mutual information is apparently not analytically tractable; we performed a numerical approximation using simulated data.

Linear Fisher Information

An analytic expression of the linear Fisher information is calculated in Appendix 2.5. Its analytic form is too complicated to be restated here, but we will examine it numerically for both the structured and unstructured weights. The qualitative behavior of the Fisher information depends on the magnitude of the common variability (σ_C) and private variability (σ_P) in a more complicated fashion than the linear stage, which depends on these variables primarily through their ratio σ_C/σ_P. Thus, we separately consider how common and private variability impact coding efficacy under various synaptic weight structures.

As before, we first consider the structured weights with $k_{\mathbf{v}}$ set to 1 while only varying $k_{\mathbf{w}}$. We start with the special case where $\sigma_P = \sigma_C = 1$ (*i.e.*, equal private and common noise variance). Here, the Fisher information saturates for both $k_{\mathbf{w}} = 1$ and $k_{\mathbf{w}} = 2$, but increases without bound for larger $k_{\mathbf{w}}$ (Fig. 2.4a). We can also consider the case where the structured weight heterogeneity grows in magnitude with the population size (*i.e.*, $k_{\mathbf{w}}$ is a function of N). In this scenario, the Fisher information is much smaller and saturates (Fig. 2.4a, dashed lines).

The information saturation (or growth) for various k_w can be understood in terms of the geometry of the covariance describing the neural population's variability. Information saturation occurs if the principal eigenvector(s) of the covariance align closely (but not necessarily exactly) with the differential correlation direction, $\mathbf{f}'$, while the remaining eigenvectors quickly become orthogonal to $\mathbf{f}'$ as population size increases [128] (see Appendix 2.5 for more details). When $k_w = 1$, the common noise aligns perfectly with the stimulus and so

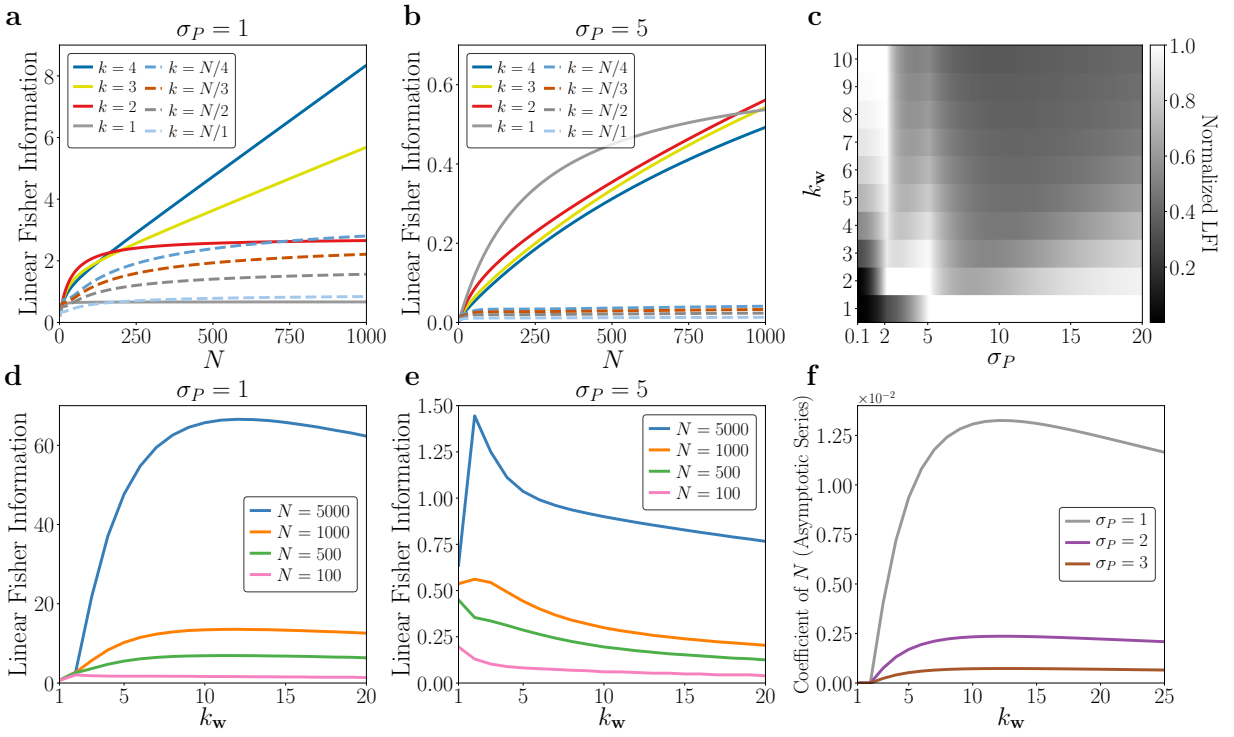


Figure 2.4: Linear Fisher information after quadratic nonlinearity in a network with structured weights. **(a)** Fisher information as a function of population size when $\sigma_P = \sigma_C = 1$, *i.e.*, private and common noise have equal variances. Solid lines denote constant k while dashed lines denote k scaling with population size. **(b)** Same as (a), but for a network where private variance dominates ($\sigma_P = 5, \sigma_C = 1$). **(c)** Normalized fisher information: for a choice of σ_P, the Fisher information is calculated for a variety of $k_{\mathbf{w}}$ (y-axis) and divided by the maximum Fisher information (across the $k_{\mathbf{w}}$, for the choice of σ_P). For a given σ_P, the normalized Fisher information is equal to one at the value of $k_{\mathbf{w}}$ which maximizes decoding performance. **(d)** Behavior of the Fisher information as a function of synaptic weight heterogeneity for various population sizes ($\sigma_P = \sigma_C = 1$). **(e)** Same as (d), but for networks where private variance dominates ($\sigma_P = 5, \sigma_C = 1$). **(f)** The coefficient of the linear term in the asymptotic series of the Fisher information at different levels of private variability. At $k_{\mathbf{w}} = 1, 2$, the coefficient of N is exactly zero.

the principal eigenvector of the covariance aligns exactly with $\mathbf{f}'$ (as in Fig. 2.1a, right). When $k_w > 1$, the principal eigenvector aligns closely, but not exactly, with the differential correlation direction. However, when $k_w = 2$, the remaining eigenvectors become orthogonal quickly enough for information to saturate. This does not occur when $k_w > 2$. The case of $k_w \sim O(N)$, meanwhile, is slightly different. Here, the variances of the covariance matrix scale with population size, so that the neurons simply exhibit too much variance for any meaningful decoding to occur. However, we believe that it is unreasonable to expect that

the synaptic weights of a neural circuit scale with the population size, making this scenario biologically implausible.

When private variability dominates, we observe qualitatively different finite network behavior ($\sigma_P = 5$, Fig. 2.4b). For $N = 1000$, both $k_{\mathbf{w}} = 1$ and $k_{\mathbf{w}} = 2$ exhibit better performance relative to larger values of $k_{\mathbf{w}}$ (by contrast, the case with $k_{\mathbf{w}} \sim O(N)$ quickly saturates). We note that, unsurprisingly, the increase in private variability has decreased the Fisher information for all cases we considered compared to $\sigma_P = 1$ (compare the scales of Fig. 2.4a and Fig. 2.4b). Our main interest, however, is identifying effective synaptic weighting strategies *given* some amount of private and common variability.

The introduction of the squared nonlinearity produces qualitatively different behavior at the finite network level: in contrast with Figure 2.3, increased heterogeneity does not automatically imply improved decoding. In fact, there is a regime in which increased heterogeneity improves Fisher information, beyond which we see a reduction in decoding performance (Fig. 2.4d). If the private variability is increased, this regime shrinks or becomes nonexistent, depending on the population size (Fig. 2.4e). Furthermore, entering this regime for higher private variability requires smaller $k_{\mathbf{w}}$ (*i.e.*, less weight heterogeneity).

The results shown in Figure 2.4d and Figure 2.4e imply that there exists an interesting relationship between the network's decoding ability, its private variability, and its synaptic weight heterogeneity $k_{\mathbf{w}}$. To explore this further, we examine the behavior of the Fisher information at a fixed population size ($N = 1000$) as a function of both σ_P and $k_{\mathbf{w}}$ (Fig. 2.4c). To account for the fact that an increase in private variability will always decrease the Fisher information, we calculate the *normalized* Fisher information: for a given choice of σ_P, each Fisher information is divided by the maximum across a range of $k_{\mathbf{w}}$ values. Thus, a normalized Fisher information allows us to determine what level of synaptic weight heterogeneity maximizes coding fidelity, given a particular level of private variability σ_P.

Figure 2.4c highlights three interesting regimes. When the private variability is small, the network benefits from larger weight heterogeneity on the common noise. But as the neurons become more noisy, the "Goldilocks zone" in which the network can leverage larger noise weights becomes constrained. When the private variability is large, the network achieves superior coding fidelity by having less heterogeneous weights, despite the threat of induced differential correlations from the common noise. Between these regimes, there are transitions for which many choices of $k_{\mathbf{w}}$ result in equally good decoding performance.

It is important to point out that Figures 2.4a-e only captures finite network behavior. Therefore, we extended our analysis by validating the asymptotic behavior of the Fisher information as a function of the private noise by examining its asymptotic series at infinity (Fig. 2.4f). For $k_{\mathbf{v}} = 1, 2$, the coefficient of the linear term is zero for any choice of σ_P, implying that the Fisher information always saturates. In addition, when the common noise weights increase with population size (*i.e.*, $k_{\mathbf{w}} \sim O(N)$), the asymptotic series is always sublinear (not shown in Fig. 2.4f). Thus, there are multiple cases in which the structure of synaptic weighting can induce differential correlations in the presence of common noise. Increased heterogeneity allows the network to escape these induced differential correlations and achieve linear asymptotic growth. If $k_{\mathbf{w}}$ becomes too large, however, the linear asymptotic

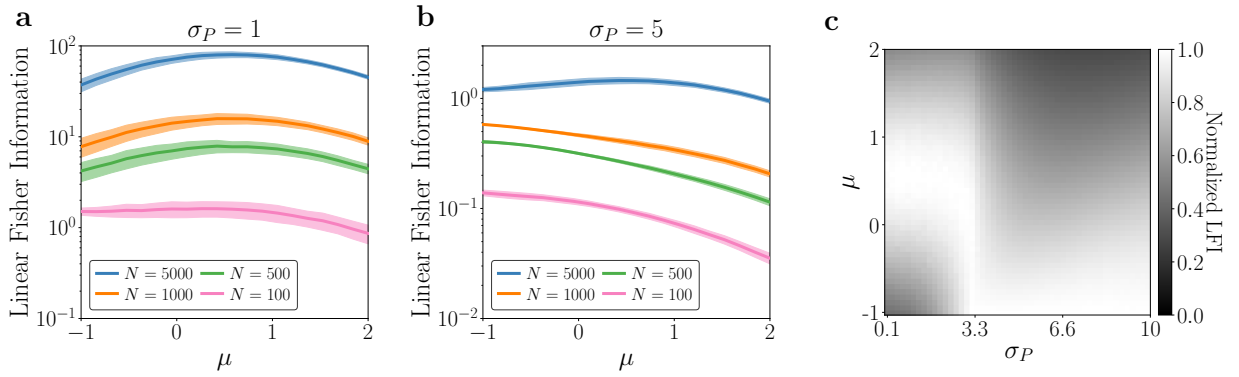


Figure 2.5: Linear Fisher information after quadratic nonlinearity, unstructured weights. In contrast to Figure 4, subplots (a) and (b) are plotted on a log-scale. **(a)** Linear Fisher information as a function of the mean, μ, of the lognormal distribution used to draw the common noise synaptic weights. Solid lines denote means while shaded regions denote one standard deviation across the 1000 drawings of weights from the lognormal distribution. **(b)** Same as (a), but for networks in which private variability dominates ($\sigma_P = 5$, $\sigma_C = 1$). **(c)** Normalized Linear Fisher information. Same plot as Figure 2.4c, but the average Fisher information across the 1000 samples is normalized across μ (akin to normalizing across $k_{\mathbf{w}}$).

growth begins to decrease. Once $k_{\mathbf{w}}$ scales as the population size, differential correlations are once again significant.

Next, we reproduce the above analysis with unstructured weights. As before, we draw 1000 samples of common noise weights from a shifted lognormal distribution with varying μ. The behavior of the average (linear) Fisher information is qualitatively similar to that of the structured weights (Fig. 2.5). There exists a regime for which larger weight heterogeneity improves the decoding performance, beyond which coding fidelity decreases (Fig. 2.5a). If the private noise variance dominates, this regime begins to disappear for smaller networks (Fig. 2.5b). Thus, with very noisy neurons, the coding fidelity of the network is improved when the synaptic weights are less heterogeneous (and therefore, smaller).

To summarize these results, we once again plot the normalized Fisher information (this time, normalized across choices of μ and averaged over 1000 samples from the lognormal distribution) for a range of private variabilities (Fig. 2.5c). The heat map exhibits a similar transition at a specific level of private variability. At this transition, a wide range of μ's provide the network with similar decoding ability. For smaller σ_P, we see behavior comparable to Figure 2.5a, where there exists a regime of improved Fisher information. Beyond the transition, the network performs better with less diverse synaptic weighting, though it becomes less stringent as σ_P increases. The behavior exhibited by this heat map is similar to Figure 2.4c, but contains fewer uniquely identifiable regions. This may imply that the additional regions in Figure 2.4c are an artifact of the structured weights.

The amount of the common noise will also impact how the network behaves and what levels of synaptic weight heterogeneity are optimal. For example, consider a network with

private noise variability set to $\sigma_P = 1$. When common noise is small, the Fisher information is comparable among various choices of synaptic weight diversity (Fig. 2.6a). When the common noise dominates, however, the network benefits strongly from diverse weighting (Fig. 2.4b), though it is punished less severely for having $k_{\mathbf{w}}$ scale with N (Fig. 2.6b, dashed lines; compare to Fig. 2.4b). These observations are true at finite population size. As before, the Fisher information saturates for $k_{\mathbf{w}} = 1, 2$ and $k_{\mathbf{w}} \sim O(N)$, no matter the choice of common noise variance.

We calculated the normalized Fisher information across a range of common noise strengths to determine the optimal synaptic weight distribution. The results for structured weights and unstructured weights are shown in Figures 2.6c and 2.6d, respectively. While they strongly resemble Figure 2.4c and Figure 2.5c, they exhibit opposite qualitative behavior. As before, there are three identifiable regions in Figure 2.6c, each divided by abrupt transitions where many choices of $k_{\mathbf{w}}$ are equally good for decoding. For small common noise, the coding fidelity is improved with less heterogeneous weights, but as the common noise increases, the network enters the "Goldilocks regions". After another abrupt transition near $\sigma_C \approx 0.34$, the network performance is greatly improved by heterogeneous weights.

Thus, common noise and private noise seem to have opposite impacts on the optimal choice of synaptic weight heterogeneity. When private noise dominates, the Fisher information is maximized under a set of homogeneous weights, since coding ability is harmed by amplification of common noise. When common noise dominates, the network coding is improved under diverse weighting: this prevents additional differential correlations and furthermore helps the network cope with the punishing effects on coding due to the amplified noise.

How should we choose the synaptic weight distribution within the extremes of private or common noise dominating? We assess the behavior of the Fisher information as both σ_P and σ_C are varied over a wide range. For the structured weights, we calculate the choice of $k_{\mathbf{w}}$ that maximized the network's Fisher information (within the range $k_{\mathbf{w}} \in [1, 10]$) (Fig. 2.6e). For the unstructured weights, we calculate the choice of μ that maximizes the network's average Fisher information over 1000 drawings of $\mathbf{w}$ from the lognormal distribution specified by μ (Fig. 2.6f).

Figures 2.6e and 2.6f reveal that the network is highly sensitive to the values of σ_P and σ_C. Figure 2.6e exhibits a band like structure and abrupt transitions in the value of $k_{\mathbf{w}}$ which maximizes Fisher information. This band-like structure would most likely continue to form for smaller σ_P if we allowed $k_{\mathbf{w}} > 10$. One might expect that the band-like structure is due to the artificial structure in the weights; however, we see that Figure 2.6f also exhibits these types of bands. Note that the regime of interest for us is when private variability is a smaller contribution to the total variability than the common variability. When this is the case, Figures 2.6e and 2.6f imply that a population of neurons will be best served by having a diverse set of synaptic weights, even if the weights amplify irrelevant signals.

Together, these results highlight how the introduction of the nonlinearity in the network reveal an intricate relationship between the amount of shared variability, private variability, and the optimal synaptic weight heterogeneity. Our observations that the network benefits

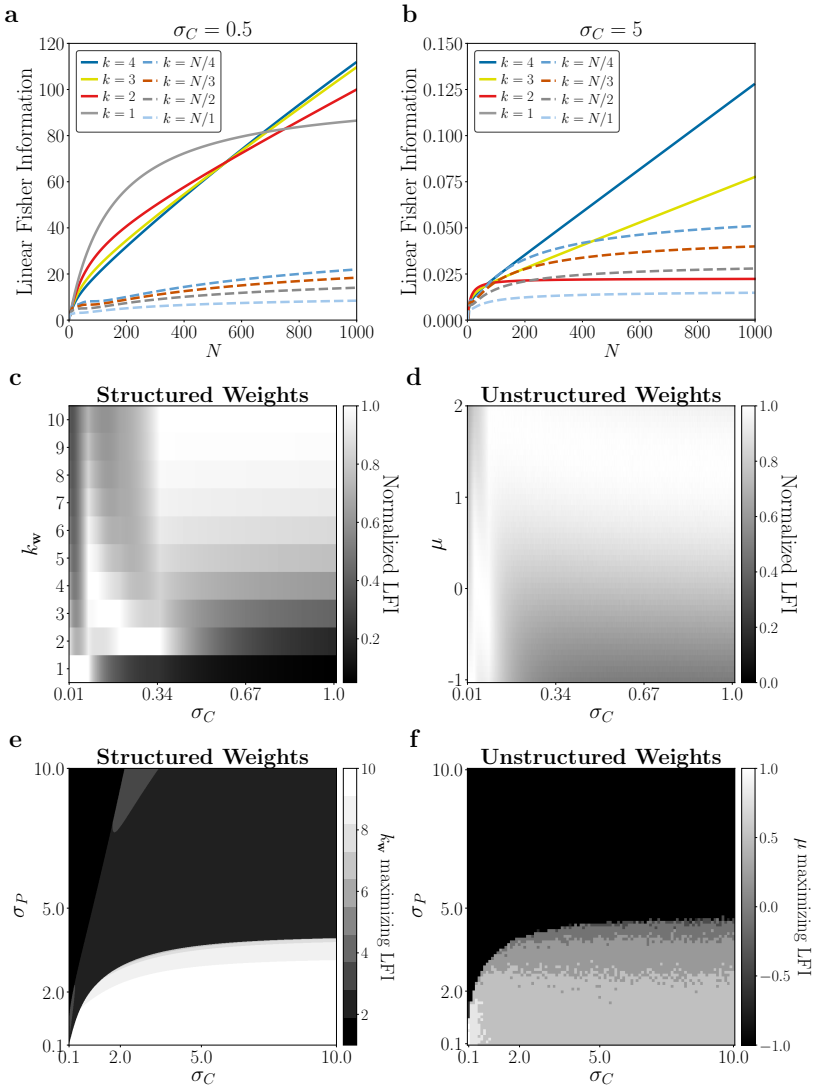


Figure 2.6: The relationship between common noise, private noise, and synaptic weight heterogeneity. **(a), (b)** Fisher information as a function of population size, N, when common noise contribution is drowned out by private noise (a), and when common noise dominates ($\sigma_P = 1$) (b). Solid lines indicate constant $k_{\mathbf{w}}$ while dashed lines refer to $k_{\mathbf{w}}$ that scales with N. **(c), (d)** Normalized Fisher information as a function of common noise for structured weights (c) and unstructured weights (d). For unstructured weights, each Fisher information is calculated by averaging over 1000 networks with their common noise weights drawn from the respective distribution. **(e)** The value of $k_{\mathbf{w}}$ that maximizes the network's Fisher information for a given choice of σ_P and σ_C. The maximum is taken over over $k_{\mathbf{w}} \in [1, 10]$. **(f)** The value of μ that maximizes the average Fisher information over 1000 draws for a given choice of σ_P and σ_C.

from increased synaptic weight heterogeneity in the presence of common noise are predicated on the size of the network (Fig. 2.4a-b, Fig. 2.6a-b) and the amount of private and shared variability (Fig. 2.4c, Fig 2.6c-d). In particular, when shared variability is the more significant contribution to the overall variability, the coding performance of the network benefits from increased heterogeneity, whether the weights are structured or unstructured (Fig. 2.6e-f). This implies that, in contrast to the linear network, there exist regimes where increasing the synaptic weight heterogeneity beyond a point will harm coding ability (Fig. 2.4d-e, Fig 2.5a-b), demonstrating that there is a tradeoff between the benefits of synaptic weight heterogeneity and the amplification of common noise it may introduce.

Mutual Information

When the network possesses a quadratic nonlinearity, the mutual information $I[s, \mathbf{r}]$ is far less tractable than for the linear case. Therefore, we computed the mutual information numerically on data simulated from the network, using an estimator built on k-nearest neighbor statistics [108]. We refer to this estimator as the KSG estimator.

We applied the KSG estimator to 100 unique datasets, each containing 100,000 samples drawn from the linear-nonlinear network. We then estimated the mutual information within each of the 100 datasets. The computational bottleneck for the KSG estimator lies in finding nearest neighbors in a kd-tree, which becomes prohibitive for large dimensions ($\sim$20), so we considered much smaller population sizes than in the case of Fisher information. Furthermore, the KSG estimator encountered difficulties when samples became too noisy, so we limited our analysis to smaller values of (σ_P, σ_C). Due to these constraints, we are only able to probe the finite network behavior of the mutual information.

Our results for the structured weights are shown in Figure 2.7. When utilizing estimators of mutual information from data, caution should be taken before comparing across different dimensions, due to bias in the KSG estimator [72]. Thus, we restrict our observations to within a specified population size. First, we evaluated the mutual information for various population sizes ($N = 8, 10, 12, 14$) in the case where $\sigma_C = \sigma_P = 0.5$. Observe that, as before, the mutual information increases with larger weight heterogeneity ($k_{\mathbf{w}}$, Fig. 2.7a). The improvement in information occurs for all four population sizes.

Decreasing the private variability increases mutual information (Fig. 2.7b). However, the network sees a greater increase in information with diverse weighting when σ_P is small. This is consistent with the small σ_P regime highlighted in Figure 2.4c: the smaller the private variability, the more the network benefits from larger synaptic weight heterogeneity. Similarly, decreasing the common variability increases mutual information (Fig. 2.7c). If the common variability is small enough (for example, $\sigma_C = 1$), then larger $k_{\mathbf{w}}$ harms the encoding. Thus, when the common noise is small enough, the amplification of noise that results when $k_{\mathbf{w}}$ is increased harms the network's encoding. It is only when the common variability becomes the dominant contribution to the variability that the diversification provided by larger $k_{\mathbf{w}}$ improves the mutual information.

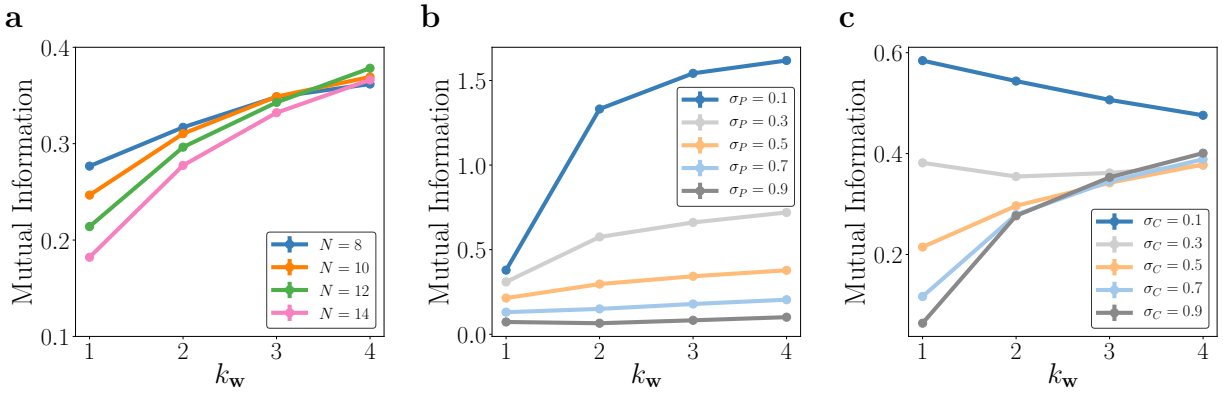


Figure 2.7: Mutual information computed by applying the KSG estimator on data simulated from the network with quadratic nonlinearity and structured weights. The estimates consist of averages over 100 datasets, each containing 100,000 samples. Standard error bars are smaller than the size of the markers. **(a)** Mutual information as a function of common noise weight heterogeneity for various population sizes N. We consider smaller N than in the case of Fisher information as computation time becomes prohibitive for larger dimensionalities. Here, $\sigma_P = \sigma_C = 0.5$. **(b)** The behavior of mutual information for various choices of σ_P, while $\sigma_C = 0.5$. **(c)** The behavior of mutual information for various choices of σ_C, while $\sigma_P = 0.5$.

As for the unstructured weights, we calculated the mutual information $I[s, \mathbf{r}]$ over 100 synaptic weight distributions drawn from the aforementioned lognormal distribution. For each synaptic weight distribution, we applied the KSG estimator to 100 unique datasets, each consisting of 10,000 samples. Thus, the mutual information estimate for a given network was computed by averaging over the individual estimates across the 100 datasets. With this procedure, we explored how the mutual information behaves as a function of the private noise variability, common noise variability, and mean of the lognormal distribution.

Similar to the normalized Fisher information, we present the normalized mutual information as a function of the private and common variances (Fig. 2.8). For a given σ_P or σ_C, the mutual information is calculated across a range of $\mu \in [-1, 1]$. The normalized mutual information is obtained by dividing each individual mutual information by the maximum value across the μ. Thus, for a given σ_P, the value of μ whose normalized mutual information is 1 specifies the lognormal distribution that maximizes the network's encoding performance. As private variability increases, the network benefits more greatly benefits diverse weighting (larger μ, Fig. 2.8a). As common variability increases, the network once again prefers more diverse weighting. If the common variability is small enough, however, the network is better suited to homogeneous weights (Fig. 2.8b). Therefore, the analysis utilizing the unstructured weights largely corroborates our findings for the structured weights shown in Figure 2.7.

Thus, these results highlight that there exist regimes where neural coding, as measured by the Shannon mutual information, benefit from increased synaptic weight heterogeneity.

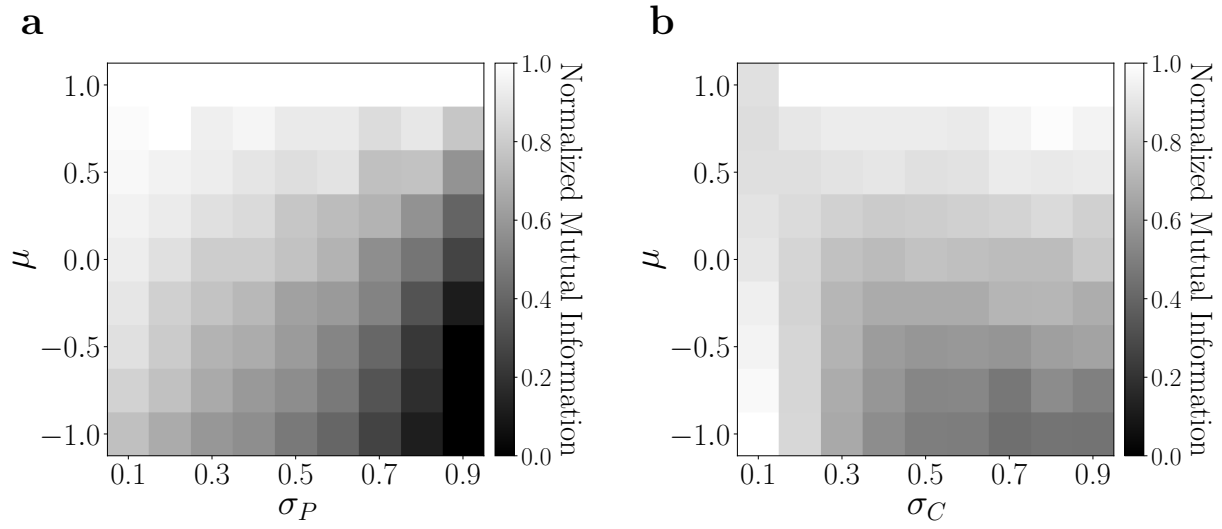


Figure 2.8: Normalized mutual information for common and private variability. For a given μ, 100 networks were created by drawing common noise weights $\mathbf{w}$ from the corresponding lognormal distribution. The mutual information shown is the average across the 100 networks. For a specified network, the mutual information was calculated by averaging KSG estimates over 100 simulated datasets, each containing 10,000 samples. Finally, for a choice of (σ_P, σ_C), mutual information is normalized to the maximum across values of μ. **(a)** Normalized mutual information as a function of μ and private variability ($\sigma_C = 0.5$). **(b)** Normalized mutual information as a function of μ and common variability ($\sigma_P = 0.5$).

Furthermore, similarly to the case of the linear Fisher information, the improvement in coding occurs more significantly when shared variability is large relative to private variability.

2.4 Discussion

We have demonstrated in a simple model of neural activity that if synaptic weighting of common noise inputs is broad and heterogeneous, coding fidelity is actually improved despite inadvertent amplification of common noise inputs. We showed that for squaring nonlinearities, there exists a regime of heterogeneous weights for which coding fidelity is maximized. We also found that the relationship between the magnitude of private and shared variability is vital for determining the ideal amount of synaptic heterogeneity. In neural circuits where shared variability is dominant, as has been reported in some parts of the cortex [59], larger weight heterogeneity results in better coding performance (Fig. 2.6e).

Why are we afforded improved neural coding under increased synaptic weight heterogeneity? An increase in heterogeneity, as we have defined it, ensures that the common noise is magnified in the network. At the same time, however, the structure of the correlated variability induced by the common noise is altered by increased heterogeneity. Previous work

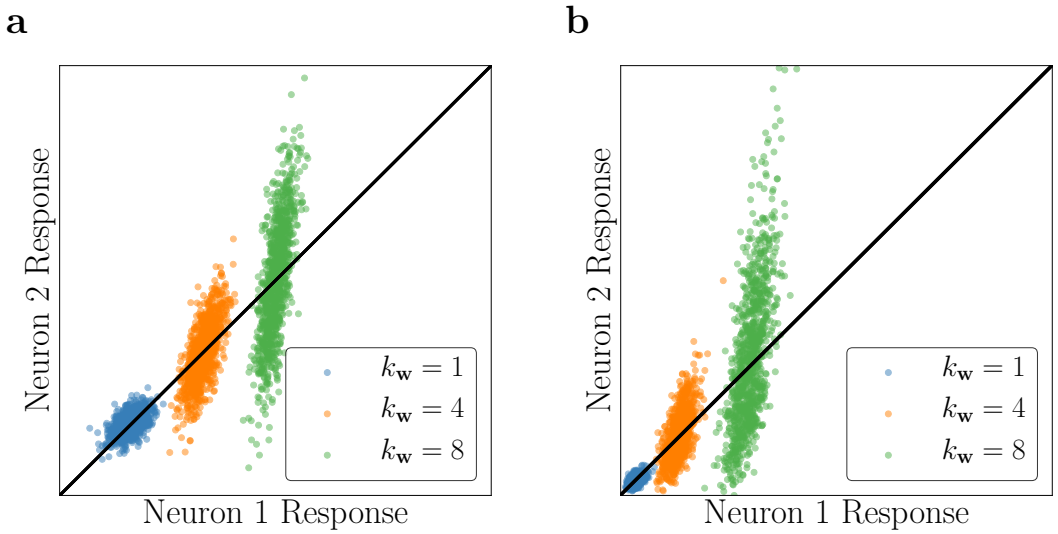


Figure 2.9: The benefits of increased synaptic weight heterogeneity. **(a)** The responses of a pair of neurons against the signal space, taken after the linear stage. Colors indicate different choices of $k_{\mathbf{w}}$ (while $k_{\mathbf{v}} = 1$). Each cloud contains 1000 sampled points. **(b)** Same as (a), but responses are taken after the quadratic nonlinearity.

has demonstrated that the relationship between signal correlations and noise correlations is important in assessing decoding ability: for example, the sign rule states that noise correlations are beneficial if they are of opposite sign as the signal correlation [84]. Geometrically, the sign rule is a consequence of the intuitive observation that decoding is easier when the noise correlations lie perpendicular to the signal manifold [12, 227, 126].

For example, consider the correlated activity for two neurons in the network against their signal space (black lines, Fig. 2.9a, b) as a function of $k_{\mathbf{w}}$. Note that the signal space is linear. After the linear stage, the larger weight heterogeneity pushes the cloud of neural activity to lie more orthogonal to the signal space. At the same time, the variance becomes observably larger due to the magnification of the common noise (Fig. 2.9a). Importantly, note that the variability for $k_{\mathbf{w}} = 1$ lies parallel to the signal space, signifying the presence of differential correlations. The correlated variability after the nonlinear stage is similar in that orthogonality to the signal space increases with $k_{\mathbf{w}}$. There is a notable difference: squaring the linear stage ensures non-negative activities, thereby limiting the response space. Thus, for large enough $k_{\mathbf{w}}$, the rectification manifests strongly enough that the network enters a regime where increased heterogeneity harms decoding. These figures only demonstrate the relationship between a pair of neurons, while the collective correlated variability structure ultimately dictates decoding performance. They do, however, shed light on how the distribution of synaptic weights can radically shape the common noise and thereby the overall structure of the shared variability.

The linear stage of the network constitutes a noisy projection of two signals (one of

which is not useful to the network) in a high-dimensional space. Thus, we can assess the entire population by examining the relationship between the projecting vectors $\mathbf{v}$ and $\mathbf{w}$. We might expect that improved decoding occurs when these signals are farther apart in the N-dimensional space [95]. For a chosen $k_{\mathbf{v}}$, this occurs as $k_{\mathbf{w}}$ is increased when the weights are structured. When the weights are unstructured, the average angle between the stimulus and weight vectors is large as either μ_v or μ_w increases. Increased heterogeneity implies access to a more diverse selection of weights, thus pushing the two signals apart. From this perspective, the nonlinear stage acts as a mapping on the high-dimensional representation. Given that no noise is added after the nonlinear processing stage in the networks, if the nonlinearities were one-to-one, the data processing inequality would ensure that the results from the linear stage would hold. But, as we observed earlier, the nonlinear stage benefits from increased heterogeneity only in certain regimes. Thus, the behavior of the nonlinearity is important: the application of the quadratic nonlinearity restricts the high-dimensional space that the neural code can occupy, and thus limits the benefits of diverse synaptic weighting. Validating and characterizing these observations for other nonlinearities (such as an exponential nonlinearity or a squared rectified linear unit) and within the framework of a linear-nonlinear-Poisson cascade model will be interesting to pursue in future studies. For example, we performed a simple experiment numerically assessing the behavior of the linear Fisher information under an exponential nonlinearity. We observed that synaptic weight heterogeneity benefits coding, but information may saturate for a wide range of k_w (Appendix 2.5). Thus, the choice of nonlinearity may impact the coding performance in the presence of common noise.

In this work, we considered the coding ability of a network in which a stimulus is corrupted by a single common noise source. However, cortical circuits receive many inputs and must likely contend with multiple common noise inputs. Thus, it is important to examine how our analysis changes as the number of inputs increases. Naively, the neural circuit could structure weights to collapse all common noise sources on a single subspace, but this strategy will fail if the circuit must perform multiple tasks (e.g., the circuit may be required to decode among many of the inputs using the same set of weights). Furthermore, there are brain regions in which the dimensionality is drastically reduced, such as cortex to striatum (10 to 1 reduction) or striatum to basal ganglia (300 to 1 reduction) [19, 171]. In these cases, the number of inputs may scale with the size of the neural circuit. In such an underconstrained system, linear decoding will be unable to properly extract estimates of the relevant stimulus. This implies that linear Fisher information, which relies on a linear decoder, may be insufficient to judge the coding fidelity of these populations. Thus, future work could examine how the synaptic weight distribution impacts neural coding with multiple common noise inputs. This includes the case when the number of common noise sources is smaller than the population size or when they are of similar scale, the latter of which may require alternative coding strategies [56, 73].

It may seem unreasonable that the neural circuit possesses the ability to weight common noise inputs. However, excitatory neurons receive many excitatory synapses in circuits throughout the brain. Some subset of common inputs across a neural population will un-

doubtedly be irrelevant for the underlying neural computation, even if these signals are not strictly speaking "noise" and could be useful for other computations. Thus, these populations must contend with common noise sources contributing to their overall shared variability and potentially hampering their ability to encode a stimulus. Our work demonstrates that neural circuits, armed with a good set of synaptic weights, need not suffer adverse impacts due to inadvertently amplifying potential sources of common noise. Instead, broad, heterogeneous weighting ensures that common noise sources will project the signal and noise into a high-dimensional space in such a way that is beneficial for decoding. This observation is in agreement with recent work that explored the relationship between heterogeneous weighting and degrees of synaptic connectivity [116]. Furthermore, synaptic input, irrelevant on one trial, may become the signal on the next: heterogeneous weighting provides a general, robust principle for neural circuits to follow.

We chose the simple network architecture in order to maintain analytic tractability, which allowed us to explore the rich patterns of behavior it exhibited. Our model is limited, however. It is worthwhile to assess how our qualitative conclusions hold with added complexity in the network. For example, interesting avenues to consider include the implementation of recurrence, spiking dynamics, In addition, these networks could also be equipped with varying degrees of sparsity and inhibitory connections. Importantly, the balance of excitation and inhibition in networks has been shown to be vital in decorrelating neural activity [154]. Past work has explored how to approximate both information theoretic quantities studied here in networks with some subset of these features [27, 216]. Thus, analyzing how common noise and synaptic weighting interact in more complex networks is of interest for future work.

We established correlated variability structure in the linear-nonlinear network by taking a linear combination of a common noise source and private noise sources (though our model ignores any noise potentially carried by the stimulus). This was sufficient to establish low-dimensional shared variability observed in neural circuits. As a consequence, our model as devised enforces stimulus-independent correlated variability. Recent work, however, has demonstrated that correlated variability is in fact stimulus-dependent. Such work used both phenomenological [114, 66] and mechanistic [227] models in producing fits to the stimulus-dependent correlated variability. These models all share a doubly stochastic noise structure, stemming from both additive and multiplicative sources of noise [77]. It is therefore worthwhile to fully examine how both additive and multiplicative modulation interact with synaptic weighting to influence neural coding. For example, [7] demonstrated that such additive and multiplicative modulation, modulated by overall population activity, can redirect information to specific neuronal assemblies, increasing information for some but decreasing it for others. Synaptic weight heterogeneity, attuned by plasticity, could serve as a mechanism for additive and multiplicative modulation, thereby gating information for specific assemblies.

2.5 Supporting Analyses

Calculation of Fisher and Mutual Information Quantities

Calculation of Fisher Information, Linear Stage

All variability after the linear stage is Gaussian; thus, the Fisher information can be expressed in the form [2, 100]:

$$I_F(s) = \mathbf{f}'(s)^T \mathbf{\Sigma}^{-1}(s)\mathbf{f}'(s) + \frac{1}{2}\operatorname{Tr}\left[\mathbf{\Sigma}'(s)\mathbf{\Sigma}^{-1}(s)\mathbf{\Sigma}'(s)\mathbf{\Sigma}^{-1}(s)\right]. \tag{2.15}$$

Our immediate goal is to calculate $\mathbf{f}(s)$, the average response of the linear stage, and $\mathbf{\Sigma}$, the covariance between the responses. The output of the ith neuron after the linear stage is

$$\ell_i = v_i s + w_i \sigma_C \xi_C + \sigma_P \xi_{P,i}, \tag{2.16}$$

so that the average response as a function of s is

$$f_i(s) = \langle \ell_i \rangle = v_i s. \tag{2.17}$$

Thus,

$$\mathbf{f}(s) = \mathbf{v}s \Rightarrow \mathbf{f}'(s) = \mathbf{v}, \tag{2.18}$$

and

$$\begin{aligned} \langle \ell_i \ell_j \rangle &= \langle (v_i s + w_i \sigma_C \xi_C + \sigma_P \xi_{P,i})(v_j s + w_j \sigma_C \xi_C + \sigma_P \xi_{P,j}) \rangle & (2.19)\\ &= v_i v_j s^2 + w_i w_j \sigma_C^2 + \sigma_P^2 \delta_{ij} & (2.20) \end{aligned}$$

so that

$$\begin{aligned} \Sigma_{ij} &= \langle \ell_i \ell_j \rangle - \langle \ell_i \rangle \langle \ell_j \rangle & (2.21)\\ &= \sigma_P^2 \delta_{ij} + w_i w_j \sigma_C^2 & (2.22)\\ \Rightarrow \mathbf{\Sigma} &= \sigma_P^2 \mathbf{I} + \sigma_C^2 \mathbf{w}\mathbf{w}^T. & (2.23) \end{aligned}$$

Notice that the covariance matrix does not depend on s, so the second term in equation (2.15) will vanish. We do, however, need the inverse covariance matrix for the first term:

$$\mathbf{\Sigma}^{-1} = \frac{1}{\sigma_P^2}\left(\mathbf{I} - \frac{\sigma_C^2}{\sigma_P^2 + \sigma_C^2 |\mathbf{w}|^2}\mathbf{w}\mathbf{w}^T\right). \tag{2.24}$$

Hence, the Fisher information is

$$\begin{aligned} I_F(s) &= \frac{1}{\sigma_P^2}\mathbf{v}^T\left(\mathbf{I} - \frac{\sigma_C^2}{\sigma_P^2 + \sigma_C^2 |\mathbf{w}|^2}\mathbf{w}\mathbf{w}^T\right)\mathbf{v} & (2.25)\\ &= \frac{1}{\sigma_P^2}\frac{(\sigma_P^2/\sigma_C^2)\,|\mathbf{v}|^2 + (|\mathbf{v}|^2|\mathbf{w}|^2 - (\mathbf{v}\cdot\mathbf{w})^2)}{(\sigma_P^2/\sigma_C^2) + |\mathbf{w}|^2}. & (2.26) \end{aligned}$$

Calculation of Mutual Information, Linear Stage

The mutual information is given by

$$I[s,\boldsymbol{\ell}] = \int d\boldsymbol{\ell} ds P[s]P[\boldsymbol{\ell}|s] \log \frac{P[\boldsymbol{\ell}|s]}{P[\boldsymbol{\ell}]} \tag{2.27}$$

$$= H[\boldsymbol{\ell}] + \int ds P[s] \int d\boldsymbol{\ell} P[\boldsymbol{\ell}|s] \log P[\boldsymbol{\ell}|s]. \tag{2.28}$$

Note that $P[\boldsymbol{\ell}]$ and $P[\boldsymbol{\ell}|s]$ are both multivariate Gaussians. The (differential) entropy of a multivariate Gaussian random variable X with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ is given by

$$H[X] = \frac{1}{2}\log\left(\det \boldsymbol{\Sigma}\right) + \frac{N}{2}(1+\log(2\pi)). \tag{2.29}$$

Therefore, by the Gaussianity of the involved distributions,

$$P[\boldsymbol{\ell}|s] = \frac{1}{\sigma_P^{N-1}\sqrt{(2\pi)^N(\sigma_P^2+\sigma_C^2|\mathbf{w}|^2)}} \times \exp\left[-\frac{1}{2\sigma_P^2}(\boldsymbol{\ell}-\mathbf{v}s)^T\left(\mathbf{I}-\frac{\sigma_C^2\mathbf{w}\mathbf{w}^T}{\sigma_P^2+\sigma_C^2|\mathbf{w}|^2}\right)(\boldsymbol{\ell}-\mathbf{v}s)\right] \tag{2.30}$$

$$P[\boldsymbol{\ell}] = \frac{1}{\sqrt{(2\pi)^N\sigma_P^{2N-4}\kappa}} \exp\left[-\frac{1}{2}\boldsymbol{\ell}^T\left(\sigma_P^2\mathbf{I}+\sigma_S^2\mathbf{v}\mathbf{v}^T+\sigma_C^2\mathbf{w}\mathbf{w}^T\right)^{-1}\boldsymbol{\ell}\right]. \tag{2.31}$$

where

$$\kappa = (\sigma_P^2+\sigma_C^2|\mathbf{w}|^2)(\sigma_P^2+\sigma_S^2|\mathbf{v}|^2) - \sigma_C^2\sigma_S^2(\mathbf{v}\cdot\mathbf{w})^2. \tag{2.32}$$

Thus,

$$H[\boldsymbol{\ell}] = \frac{1}{2}\log\left(\sigma_P^{2N-4}\kappa\right) + \frac{N}{2}(1+\log(2\pi)). \tag{2.33}$$

and

$$\int d\boldsymbol{\ell} P[\boldsymbol{\ell}|s]\log P[\boldsymbol{\ell}|s] = -\frac{1}{2}\log(\sigma_P^{2N-2}(\sigma_P^2+\sigma_C^2|\mathbf{w}|^2)) - \frac{N}{2}(1+\log(2\pi)), \tag{2.34}$$

which is notably independent of s. Thus, the integral over s will marginalize away. We are left with

$$I[s,\boldsymbol{\ell}] = \frac{1}{2}\log\left(\frac{\kappa}{\sigma_P^2(\sigma_P^2+\sigma_C^2|\mathbf{w}|^2)}\right) \tag{2.35}$$

$$= \frac{1}{2}\log\left(1+\sigma_S^2 I_F(s)\right). \tag{2.36}$$

Calculation of Linear Fisher Information, Quadratic Nonlinearity

We repeat the calculation of the first section, but after the nonlinear stage. In this case, we consider a quadratic nonlinearity. Instead of the Fisher information, we calculate the linear Fisher information (since it is analytically tractable). The output of the network is

$$\begin{aligned} r_i &= (v_i s + w_i \sigma_C \xi_C + \sigma_P \xi_{P,i})^2 && (2.37)\\ &= v_i^2 s^2 + w_i^2 \sigma_C^2 \xi_C^2 + \sigma_P^2 \xi_{P,i}^2 + 2 s v_i w_i \sigma_C \xi_C + 2 s v_i \sigma_P \xi_{P,i} + 2 w_i \sigma_C \sigma_P \xi_C \xi_{P,i}. && (2.38) \end{aligned}$$

Thus, the average is then

$$f_i(s) = \langle r_i \rangle = v_i^2 s^2 + w_i^2 \sigma_C^2 + \sigma_P^2, \tag{2.39}$$

which implies

$$\begin{aligned} \langle r_i \rangle \langle r_j \rangle &= (v_i^2 s^2 + w_i^2 \sigma_C^2 + \sigma_P^2)(v_j^2 s^2 + w_j^2 \sigma_C^2 + \sigma_P^2) && (2.40)\\ &= \sigma_P^4 + s^2 \sigma_P^2 (v_i^2 + v_j^2) + \sigma_P^2 \sigma_C^2 (w_i^2 + w_j^2) \\ &\quad + s^2 \sigma_C^2 (v_i^2 w_j^2 + v_j^2 w_i^2) + s^4 v_i^2 v_j^2 + \sigma_C^4 w_i^2 w_j^2 && (2.41) \end{aligned}$$

Next, the covariate can be written as

$$\begin{aligned} \langle r_i r_j \rangle = \sigma_P^4 &+ s^2 \sigma_P^2 (v_i^2 + v_j^2) + \sigma_P^2 \sigma_C^2 (w_i^2 + w_j^2) + s^2 \sigma_C^2 (v_i^2 w_j^2 + v_j^2 w_i^2) \\ &+ s^4 v_i^2 v_j^2 + 3 \sigma_C^4 w_i^2 w_j^2 + 4 s^2 \sigma_C^2 v_i v_j w_i w_j. \end{aligned} \tag{2.42}$$

The off diagonal terms of the covariance matrix are then

$$\langle r_i r_j \rangle - \langle r_i \rangle \langle r_j \rangle = 2 \sigma_C^4 w_i^2 w_j^2 + 4 s^2 \sigma_C^2 v_i v_j w_i w_j. \tag{2.43}$$

Lastly, the variance of r_i (the diagonal terms of the covariance matrix) is given by

$$\begin{aligned} \mathrm{Var}(r_i) &= \langle r_i^2 \rangle - \langle r_i \rangle^2 && (2.44)\\ &= 3 \sigma_P^4 + 6 s^2 \sigma_P^2 v_i^2 + 6 \sigma_P^2 \sigma_C^2 w_i^2 + 6 s^2 \sigma_C^2 v_i^2 w_i^2 + s^4 v_i^4 + 3 \sigma_C^4 w_i^4 \\ &\quad - \left(v_i^2 s^2 + w_i^2 \sigma_C^2 + \sigma_P^2 \right)^2 && (2.45)\\ &= 2 \sigma_C^4 w_i^4 + 4 s^2 \sigma_C^2 v_i^2 w_i^2 + 2 \sigma_P^4 + 4 s^2 \sigma_P^2 v_i^2 + 4 \sigma_P^2 \sigma_C^2 w_i^2. && (2.46) \end{aligned}$$

Thus, the total covariance, which takes the variance into consideration, is

$$\Sigma_{ij} = \delta_{ij} \left(2 \sigma_P^4 + 4 \sigma_P^2 (s^2 v_i^2 + \sigma_C^2 w_i^2) \right) + 4 s^2 \sigma_C^2 v_i v_j w_i w_j + 2 \sigma_C^4 w_i^2 w_j^2. \tag{2.47}$$

In vector notation, this can be expressed as

$$\mathbf{\Sigma} = 2 \sigma_P^4 \mathbf{I} + 4 \sigma_P^2 s^2 \mathrm{diag}(\mathbf{V}) + 4 \sigma_P^2 \sigma_C^2 \mathrm{diag}(\mathbf{W}) + 4 s^2 \sigma_C^2 \mathbf{X} \mathbf{X}^T + 2 \sigma_C^4 \mathbf{W} \mathbf{W}^T \tag{2.48}$$

where

$$\mathbf{V} = \mathbf{v} \odot \mathbf{v} \tag{2.49}$$
$$\mathbf{W} = \mathbf{w} \odot \mathbf{w} \tag{2.50}$$
$$\mathbf{X} = \mathbf{v} \odot \mathbf{w}, \tag{2.51}$$

where $\odot$ indicates the Hadamard product (element-wise product). We now proceed to the linear Fisher information:

$$I_{LFI}(s) = \mathbf{f}'(s)^T \mathbf{\Sigma}(s)^{-1} \mathbf{f}'(s). \tag{2.52}$$

We start by calculating the inverse covariance matrix, which we will achieve with repeated applications of the Sherman-Morrison formula [179]. We can write

$$\mathbf{\Sigma}^{-1} = (\mathbf{M} + 2\sigma_C^4 \mathbf{W}\mathbf{W}^T)^{-1} \tag{2.53}$$
$$= \mathbf{M}^{-1} - \frac{\mathbf{M}^{-1}(2\sigma_C^4 \mathbf{W}\mathbf{W}^T)\mathbf{M}^{-1}}{1 + 2\sigma_C^4 \mathbf{W}^T\mathbf{M}^{-1}\mathbf{W}} \tag{2.54}$$
$$= \mathbf{M}^{-1} - \frac{2\sigma_C^4}{1 + 2\sigma_C^4 \mathbf{W}^T\mathbf{M}^{-1}\mathbf{W}} \mathbf{M}^{-1}\mathbf{W}\mathbf{W}^T\mathbf{M}^{-1}. \tag{2.55}$$

Where

$$\begin{aligned}\mathbf{M}^{-1} \equiv & \left(2\sigma_P^4 + 4\sigma_P^2 s^2 v_i^2 + 4\sigma_P^2 \sigma_C^2 w_i^2\right)^{-1} \delta_{ij} \\ & - \frac{s^2 \sigma_C^2}{\sigma_P^4 + 2s^2\sigma_C^2\sigma_P^2 \sum_i \frac{v_i^2 w_i^2}{\sigma_P^2 + 2s^2 v_i^2 + 2\sigma_C^2 w_i^2}} \\ & \times \frac{v_i v_j w_i w_j}{\left(\sigma_P^2 + 2s^2 v_i^2 + 2\sigma_C^2 w_i^2\right)\left(\sigma_P^2 + 2s^2 v_j^2 + 2\sigma_C^2 w_j^2\right)}.\end{aligned} \tag{2.56}$$

Note that

$$\mathbf{f}'(s) = 2s\mathbf{V}, \tag{2.57}$$

so the Fisher information is

$$I_{LFI}(s) = 4s^2 \left(\mathbf{V}^T\mathbf{M}^{-1}\mathbf{V} - \frac{2\sigma_C^4}{1 + 2\sigma_C^4 \mathbf{W}^T\mathbf{M}^{-1}\mathbf{W}} \mathbf{V}^T\mathbf{M}^{-1}\mathbf{W}\mathbf{W}^T\mathbf{M}^{-1}\mathbf{V}\right) \tag{2.58}$$
$$= 4s^2 \left(\mathbf{V}^T\mathbf{M}^{-1}\mathbf{V} - \frac{2\sigma_C^4}{1 + 2\sigma_C^4 \mathbf{W}^T\mathbf{M}^{-1}\mathbf{W}} \left(\mathbf{V}^T\mathbf{M}^{-1}\mathbf{W}\right)^2\right). \tag{2.59}$$

To facilitate the matrix multiplications, we will define the following notation

$$\{v, w\}_{m,n} = \sum_i \frac{v_i^m w_i^n}{\sigma_P^2 + 2s^2 v_i^2 + 2\sigma_C^2 w_i^2}. \tag{2.60}$$

Thus,

$$\begin{aligned}\mathbf{V}^T\mathbf{M}^{-1}\mathbf{V} &= \frac{1}{2\sigma_P^2}\sum_i \frac{v_i^4}{\sigma_P^2 + 2s^2v_i^2 + 2\sigma_C^2 w_i^2} \\ &\quad - \frac{s^2\sigma_C^2}{\sigma_P^4 + 2s^2\sigma_C^2\sigma_P^2\left\{v,w\right\}_{2,2}}\left(\sum_i \frac{v_i^3 w_i}{\sigma_P^2 + 2s^2v_i^2 + 2\sigma_C^2 w_i^2}\right)^2 \qquad (2.61)\\ &= \frac{1}{2\sigma_P^2}\left\{v,w\right\}_{4,0} - \frac{s^2\sigma_C^2}{\sigma_P^4 + 2s^2\sigma_C^2\sigma_P^2\left\{v,w\right\}_{2,2}}\left\{v,w\right\}_{3,1}^2. \qquad (2.62)\end{aligned}$$

Furthermore,

$$\mathbf{W}^T\mathbf{M}^{-1}\mathbf{W} = \frac{1}{2\sigma_P^2}\left\{v,w\right\}_{0,4} - \frac{s^2\sigma_C^2}{\sigma_P^4 + 2s^2\sigma_C^2\sigma_P^2\left\{v,w\right\}_{2,2}}\left\{v,w\right\}_{1,3}^2 \qquad (2.63)$$

and finally

$$\begin{aligned}\mathbf{V}^T\mathbf{M}^{-1}\mathbf{W} &= \frac{1}{2\sigma_P^2}\left\{v,w\right\}_{2,2} \\ &\quad - \frac{s^2\sigma_C^2}{\sigma_P^4 + 2s^2\sigma_C^2\sigma_P^2\left\{v,w\right\}_{2,2}}\left\{v,w\right\}_{1,3}\left\{v,w\right\}_{3,1}. \qquad (2.64)\end{aligned}$$

Inserting this expression into equation (2.59) and simplifying, we can write the Fisher information as

$$\begin{aligned}I_{LFI}(s) = 4s^2\Bigg(&\tfrac{1}{\sigma_P^2}\left\{v,w\right\}_{4,0} - \tfrac{2s^2\sigma_C^2}{\sigma_P^2+2s^2\sigma_P^2\sigma_C^2\{v,w\}_{2,2}}\left\{v,w\right\}_{3,1}^2 + \\ &\tfrac{\sigma_P^2\sigma_C^4\{v,w\}_{2,2}+2s^2\sigma_C^6(\{v,w\}_{2,2}-2\{v,w\}_{1,3}\{v,w\}_{3,1})}{\sigma_P^4+\sigma_P^2(\sigma_C^4\{v,w\}_{0,4}+2s^2\sigma_C^2\{v,w\}_{2,2})+2s^2\sigma_C^6(\{v,w\}_{0,4}\{v,w\}_{2,2}-2\{v,w\}_{1,3}^2)}\Bigg). \qquad (2.65)\end{aligned}$$

Information Saturation and Differential Correlations

In Section 2.3, we observed that the Fisher information saturates in particular instances of the nonlinear network. Specifically, for the nonlinear network, Fisher information saturates for $k_w = 1$ and $k_w = 2$, but not for $k_w > 3$. Additionally, Fisher information saturates for $k_w \sim O(N)$. To understand why we observe saturation in some cases and not others, it is helpful to examine the eigenspectrum of the covariance matrix $\mathbf{\Sigma}$ describing the neural responses. Here, we rely on an analysis in the supplement of [128].

The linear Fisher information can be written in terms of the eigenspectrum of $\mathbf{\Sigma}$ as

$$\begin{aligned}I_{LFI} &= \mathbf{f}'^T\mathbf{\Sigma}^{-1}\mathbf{f}' \qquad (2.66)\\ &= \mathbf{f}'^T\mathbf{f}'\sum_k \frac{\cos^2\theta_k}{\sigma_k^2}, \qquad (2.67)\end{aligned}$$

where σ_k^2 is the kth eigenvalue, and θ_k is the angle between the kth eigenvector and $\mathbf{f}'$. We consider the cases in which I_{LFI} saturates with the population size N. First, note that squared norm of the tuning curve derivative $\mathbf{f}'^T\mathbf{f}'$ will scale as $O(N)$, since there are N terms in the sum. This implies that the summation must shrink at least as fast as $O(1/N)$ for information to saturate. This implies that any eigenvalues scaling as $O(1)$ must have their corresponding cosine-angles shrink faster than $O(1/N)$. If there are $O(N)$ such eigenvalues, they must shrink faster than $O(1/N^2)$.

In the case of $k_w = 1$, one eigenvalue grows as $O(N)$ while the others remain constant (Fig. 2.10a, left). Meanwhile, the cosine-angles of the constant eigenvalues are effectively zero. This case is the easiest to understand: the principal eigenvector aligns with $\mathbf{f}'$ while all other directions are effectively orthogonal to $\mathbf{f}'$. For $k_w \geq 1$, however, two eigenvalues grow as $O(N)$ while the others grow as $O(1)$ (Fig. 2.10a, middle and right). In this case, the behavior of the cosine-angles corresponding to the constant growth eigenvalues varies depending on k_w.

As in Moreno-Bote et al., we split up equation (2.67) into two groups: those with eigenvalues that scale as $O(N)$ (denoted by the set S_N), and those that scale as $O(1)$ (denoted by the set S_1):

$$I_{LFI} = \mathbf{f}'^T\mathbf{f} \sum_{m \in S_N} \frac{\cos^2 \theta_m}{\sigma_m^2} + \mathbf{f}'^T\mathbf{f}' \sum_{n \in S_1} \frac{\cos^2 \theta_n}{\sigma_n^2}. \tag{2.68}$$

The left sum contains one term when $k_w = 1$ and two terms when $k_w > 1$. Information saturation is dictated by the right sum, which we call R_{k_w}:

$$R_{k_w} = \sum_{n \in S_1} \frac{\cos^2 \theta_n}{\sigma_n^2}. \tag{2.69}$$

The addends of R_{k_w} correspond to the $O(1)$ eigenvalues, whose eigenvectors must have cosine-angles that vanish more quickly than $O(1/N)$ since there are $O(N)$ such eigenvalues. As expected, for $k_w = 1$, R_1 quickly vanishes (Fig. 2.10b: gray line). We observe similar behavior for $k_w = 2$: the summation R_2 eventually vanishes as well (Fig. 2.10b: red line). However, for $k_w > 2$, this no longer occurs: the cosine-angles scale to zero slowly enough that R_3 approaches a constant value (thereby preventing information saturation). Thus, going to larger k_w ensures that the majority of the eigenvectors of $\mathbf{\Sigma}$ do not become orthogonal to $\mathbf{f}'$ quickly enough for information saturation to occur.

In the case of $k_w \sim O(N)$, however, the behavior of the covariance matrix is different. Recall that the covariance matrix takes on the form

$$\mathbf{\Sigma} = 2\sigma_P^4\mathbf{I} + 4\sigma_P^2 s^2 \text{diag}(\mathbf{V}) + 4\sigma_P^2\sigma_C^2 \text{diag}(\mathbf{W}) + 4s^2\sigma_C^2\mathbf{X}\mathbf{X}^T + 2\sigma_C^4\mathbf{W}\mathbf{W}^T. \tag{2.70}$$

The dominant contribution to the covariance matrix is $2\sigma_C^4\mathbf{W}\mathbf{W}^T$. Thus, the scaling of the

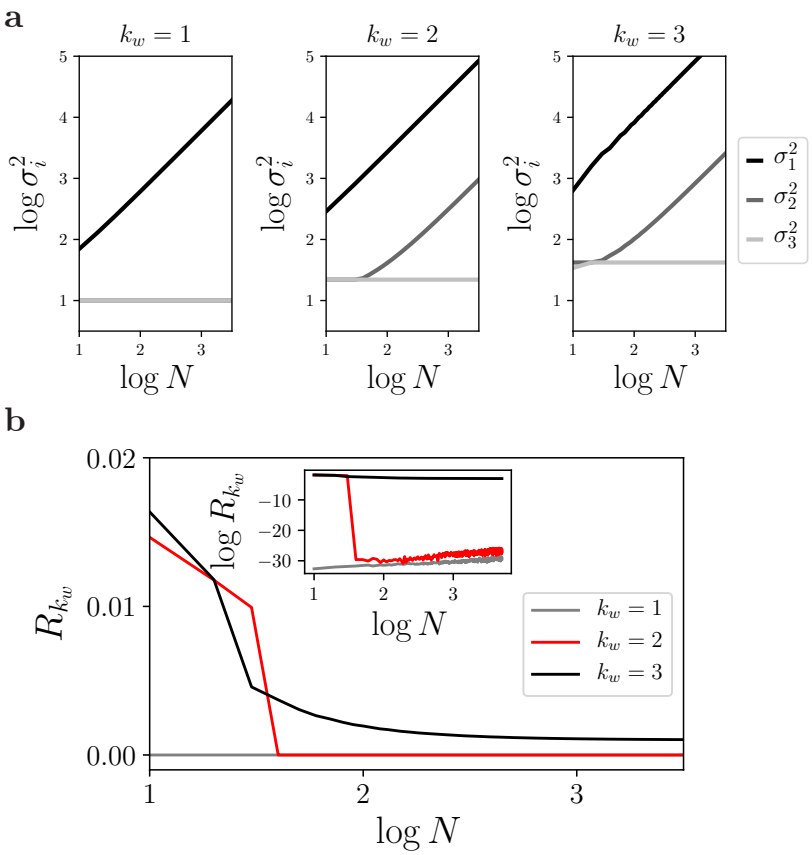


Figure 2.10: Characterizing the scaling of the eigenvalues and the shrinking of the cosine-angles for the nonlinear stage covariance. **(a)** Behavior of largest three eigenvalues σ_1^2, σ_2^2, and σ_3^2 for the cases of $k_w = 1, 2, 3$. Aspect ratio is chosen so that unit steps on each axis appear of equal length. **(b)** The behavior of cosine-angle sum R_i corresponding to the constant-growth eigenvalues, for each of $k_w = 1, 2, 3$. Inset depicts the same curves, but on a log-log scale.

trace of $\mathbf{\Sigma}$ is

$$\mathrm{Tr}[\mathbf{\Sigma}] \sim \mathrm{Tr}[\mathbf{W}\mathbf{W}^T] = \mathrm{Tr}[(\mathbf{w} \odot \mathbf{w})(\mathbf{w} \odot \mathbf{w})^T]. \tag{2.71}$$

$$= (\mathbf{w} \odot \mathbf{w})^T (\mathbf{w} \odot \mathbf{w}) \tag{2.72}$$

$$\sim \sum_{i=1}^{N} \left(i^2\right)^2 \sim O(N^5). \tag{2.73}$$

Since the trace of the covariance matrix is equal to the sum of the eigenvalues, some subset of the eigenvalues can scale as $O(N^5)$ as well. In fact, all eigenvalues scale at least as $O(N)$, with the largest eigenvalue scaling as $O(N^5)$. In this scenario, the Fisher information must saturate because the cosine-angle can at most scale to a constant. In plainer terms, the

variances of the covariance matrix scale so quickly that the differential correlation direction is irrelevant. We interpret this behavior as the neurons simply exhibiting too much variance for any meaningful decoding to occur. Note, however, that the saturation can be avoided if the behavior of $\mathbf{f}'$, which we assumed scales as $O(N)$, instead scales more quickly. This can occur, for example, when $k_v \sim O(N)$. However, it is unreasonable to expect that the synaptic weights of a neural circuit scale with the population size, making this scenario biologically implausible.

Linear Fisher Information under an Exponential Nonlinearity

The application of an exponential nonlinearity to the output of the linear stage $g_i(\ell_i) = \exp(\ell_i)$ implies that the output of the network $\mathbf{r} = \mathbf{g}(\boldsymbol{\ell})$ follows a multivariate lognormal distribution (since the linear stage is Gaussian). The linear stage is described by the distribution

$$\boldsymbol{\ell} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}^L) \tag{2.74}$$

$$\boldsymbol{\mu} = \mathbf{v}s \tag{2.75}$$

$$\boldsymbol{\Sigma}^L = \sigma_P^2 \mathbf{I} + \sigma_C^2 \mathbf{w}\mathbf{w}^T. \tag{2.76}$$

The multivariate lognormal distribution has first- and second-order statistics given by

$$\mathrm{E}\left[\mathbf{r}\right]_i = \exp\left[\mu_i + \frac{1}{2}\Sigma_{ii}^L\right] \tag{2.77}$$

$$\mathrm{Var}\left[\mathbf{r}_{ij}\right] = \exp\left[\mu_i + \mu_j + \frac{1}{2}\left(\Sigma_{ii}^L + \Sigma_{jj}^L\right)\right]\left(\exp(\Sigma_{ij}^L) - 1\right) \tag{2.78}$$

Thus, the mean activity and its derivative with respect to s are given by

$$f_i(s) = \exp\left[\frac{1}{2}\sigma_P^2 + v_i s + \frac{1}{2}\sigma_C^2 w_i^2\right] \tag{2.79}$$

$$f_i'(s) = v_i \exp\left[\frac{1}{2}\sigma_P^2 + v_i s + \frac{1}{2}\sigma_C^2 w_i^2\right]. \tag{2.80}$$

These equations provide us the tools to calculate the linear Fisher information. The inversion of the covariance matrix (equation 2.78) is not tractable, but we can proceed numerically.

We calculated the linear Fisher information numerically under the same conditions as in Figure 2.4a, but with $k_w = 1, \ldots, 5$ and for a wider range of population sizes. In Figure 2.11, we plot the linear Fisher information as a function of N for these choices of k_w. We observe that, for large enough N, synaptic weight heterogeneity results in improved coding performance. However, we also observe what appears to be saturation of the Fisher information. Since we cannot write the Fisher information as a function of N, we cannot validate this observation analytically. This does, however, suggest that the choice of nonlinearity can dramatically impact the behavior of the linear Fisher information.

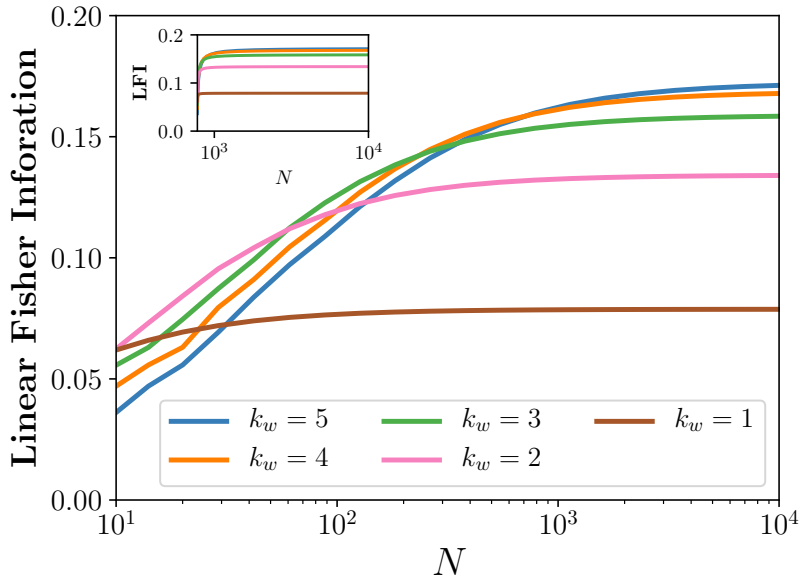


Figure 2.11: The behavior of linear Fisher information for an exponential nonlinearity as a function of population size. Colors denote different choices of k_w. Inset shows the same plot, but on a regular scale.

Conclusion

We demonstrated that diverse synaptic weighting can reduce the adverse effects of common noise even if it amplifies that common noise. Thus, our results provide a general principle by which neural circuits can shape and leverage correlated variability to improve neural decoding. This work falls in a long line of literature demonstrating whether correlated variability is beneficial or harmful for decoding. However, it does not speak to answer questions on the optimality of observed correlated variability. Assessing optimality requires a new framework, developed in the following chapter.

Chapter 3

Optimal correlated variability is biologically implausible

Chapter Co-authors
Jesse A. Livezey
Matthew Summers
Marla B. Feller
Kristofer E. Bouchard

A long line of analyses on correlated variability do not speak to whether observed correlated variability is optimal, despite implying that correlated variability might benefit neural coding. To what degree is the correlated variability observed in neural systems structured in a manner to optimize decoding? More simply, is correlated variability efficient from a decoding perspective? Answering this question requires the development of a new framework, which this chapter details.

3.1 Introduction

Variability is a prominent feature of neural activity: neural activity exhibits trial-to-trial fluctuations in response to the same stimulus. Furthermore, such variability is typically pairwise correlated (noise correlations) [12, 51]. Specifically, conditioned on repeated stimulus presentations, neural responses will covary (Fig. 3.1b). The existence of correlated variability is of paramount importance for the fidelity of a neural code.

Neural variability is believed to have several underlying sources. First, individual neurons may have their own private trial-to-trial variability (Fig 3.1a), which is a zero-correlation contribution to the total multi-unit variability. Another potential source of variability can come from ongoing neural activity across the brain that is not relevant for the decoding task [191], which can contribute to correlated variability depending on how it is loaded onto the observed neurons [164]. A third type of potentially correlated variability which is

relevant for decoding is *information limiting* variability (Fig. 3.1c). Although it can arise from different sources, without additional simultaneous measurements, it is indistinguishable from stimulus corrupting noise. This type of correlated variability leads to information limiting correlations [128]. Finally, recurrent computations in the observed area, sub-optimal computations in the preceding areas, and attention have all been related to changes in the correlated variability [227, 128, 86, 158, 29].

Assessing the impact of correlated variability on a population code has long been of theoretical interest. A host of studies have demonstrated correlated variability's diverse effects on the fidelity of a neural code, depending on the variability's sources, structure, and relationship with the tuning properties of the population [226, 2, 217, 62, 14]. Furthermore, correlated variability can potentially limit the precision with which a downstream cortical areas or brain-computer interfaces can decode the incoming stimulus information [128, 96, 106]. The question of whether correlated variability is *beneficial* for decoding can be framed in the broader question of whether the observed structure is *optimal* for decoding.

Recent large recordings have confirmed that the scale at which the class of information limiting correlations cause information saturation is approximately 1,000 neurons [93, 160, 23]. Thus, it is well understood that the structure of correlated variability at large population sizes is sub-optimal due to information-limiting correlations. Although there is evidence that some of these correlations are inevitable due to noise in the incoming stimulus or biophysical sensors [128, 96], it is possible that some part of them are due to sub-optimal computations [29]. At a smaller scale of 10s of neurons, however, the optimality of the correlated variability has not been thoroughly analyzed.

At these smaller dimensions, it has been shown that noise correlations often increase LFI compared to neural responses where the pairwise correlations have been removed [13, 62, 66, 227], although this is not always the case [85]. However, this comparison between observed correlations and zero correlations is only a weak test of optimality. The fact that the observed data exhibits higher LFI than the zero-correlations case does not imply that there are not other correlated variability structures with even higher LFI. Thus, theoretical analyses on the "optimality" of correlated neural variability depend strongly on the choice of null model (optimal relative to *what?*).

A model for correlated variability is comprised of a set of constraints (e.g., fixed marginals, fixed spectrum) and a set of degrees of freedom (e.g., pairwise correlations, synaptic loadings). These constraints and degrees of freedom reflect what we believe the biological network cannot modify and what it can modify and potentially optimize. When paired with a decoding measure such as LFI, the model degrees of freedom can be optimized to maximize the measure to potentially compare it with observed structure [85]. When paired with a null distribution of the degrees of freedom, this defines a null model to compare the observed measure against. Determining whether an observed set of neural responses is optimal, near chance, or worse-than-chance for decoding consists of examining where the observed measure lies within the null distribution (the null percentile). Across a population, the distribution of the observed percentiles under the null model can be used to asses the degree to which the biological network has optimized its hypothesized degrees of freedom for the paired measure.

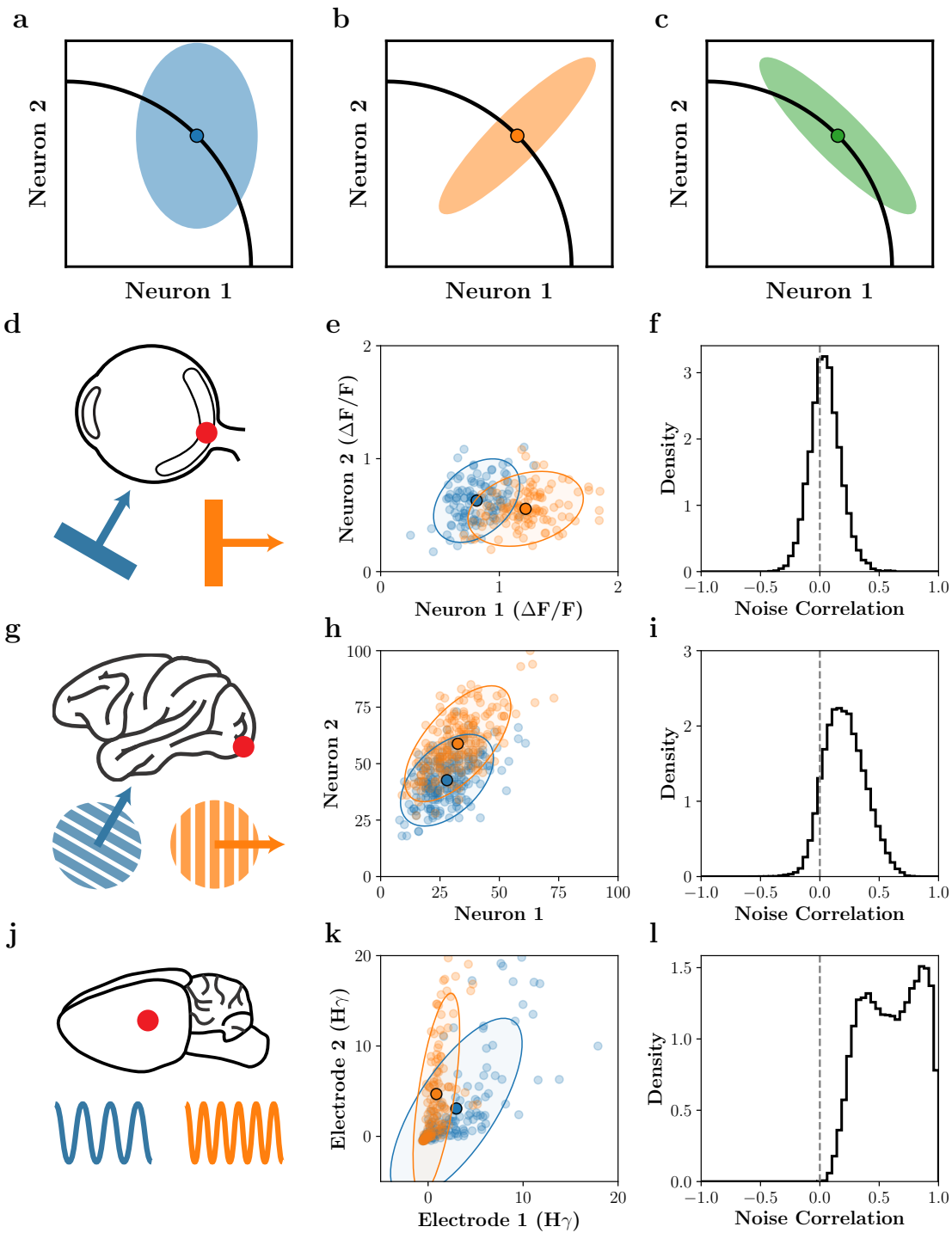


Figure 3.1: **Correlated variability is a pervasive neural phenomenon.** Continued on the following page.

Figure 3.1: **Continued from previous page. a-c: Potential components of neural variability.** Each plot depicts the neural space, whose axes correspond to the response of a specific pair of neurons to a stimulus. Black curves denote the mean responses across different stimuli (i.e., tuning curves). Variability about a specific stimulus mean response (solid points) may exhibit: **a.** Private, uncorrelated variability in each neural dimension (full rank), **b.** Correlated variability, with correlations in the neural space (potentially low-rank), and **c.** Differential correlations, which lie parallel to the mean response curve (low-rank). **d-l: Correlated variability in neural data.** Each row refers to a different dataset, while columns refer to a calculated aspect of the dataset. **Rows: d-f.** Calcium imaging recordings from mouse retinal ganglion cells in response to drifting bars. **g-i.** Single-unit activity recorded from primary visual cortex of macaque monkey in response to drifting gratings. **j-l.** Micro-electrocorticography recordings (z-scored Hγ response) from rat primary auditory cortex in response to tone pips at varying frequencies. **Columns:** First column (**d**, **g**, **j**) depicts the brain region and stimulus for each dataset. Second column (**e**, **h**, **k**) depicts the response of two randomly units in the population to two neighboring stimuli. Individual points denote the unit response on separate trials, while covariance ellipses denote the noise covariance ellipse at 2 standard deviations. Third column (**f**, **i**, **l**) plots the distribution of noise correlations, calculated for each unit and unique stimulus, across the population.

As a starting point, if the percentiles are distributed uniformly between zero and one, then the observed neural responses are no more optimized than a random-uniform setting of the degrees-of-freedom in the model. To the extent that the observed percentiles are distributed closer to one, the neural activity can be interpreted as being optimal, although real datasets are unlikely to achieve perfect optimality (all percentiles equal to one). Conversely, to the extent that the observed percentiles are distributed closer to zero, the neural activity can be interpreted as having worse optimality than a chance setting of the degrees-of-freedom.

In order to test the optimality of observed neural responses, we propose three null models, each of which have a particular biophysical interpretation. We contrast these null models with the commonly used shuffle null model, which compares the observed correlations to a distribution with equal per-unit variance, but zero correlations. The first proposed null model has the same constraints as the shuffle null model (preserves the per-unit variance), but compares the observed correlations to all possible correlational structures uniformly. The other two null models, a rotation null model and a Factor Analysis null model, attribute the correlated variability entirely to incoming shared variability and to a mixture of private variability and shared variability, respectively. Together, these null models provide methods to test the optimality of correlated variability under various biophysical assumptions.

We test the optimality of neural responses in three datasets recorded from retina, primary visual cortex (V1), and primary auditory cortex. Using the proposed null models, we find that the observed correlated variability has discriminability that is lower than chance across all datasets and null models. Furthermore, the observed percentiles quickly approach zero as a function of the dimensionality of the neural data. In order to understand this result,

we analyze the features of optimal correlational structures under the null models. We find that for a large fraction of subsamples of the recorded units, achieving optimality would push the neural responses into regimes that violate soft biophysical constraints. Together, our results demonstrate that traditional null models of correlated variability may overstate the optimality of observed neural data, and that biophysical constraints limit the ability of neural activity to achieve optimal correlated variability.

3.2 Methods

Linear Fisher information measures coding fidelity

One commonly used measure of coding fidelity in the context of decoding is the Fisher information, which is related to a limit on how accurately a readout of a neural representation can be used to determine the value of the stimulus [53]. Formally, it sets a lower bound to the variance of an unbiased estimator for the stimulus. In practice, the Fisher information is analytically intractable. An alternative measure is the *linear Fisher information* (LFI), defined as

$$\mathcal{I}(s) = \frac{d\mathbf{f}(s)^T}{ds}\mathbf{\Sigma}(s)^{-1}\frac{d\mathbf{f}(s)}{ds} \tag{3.1}$$

where $\mathbf{f}(s)$ is the neural population's average response across trials, and $\mathbf{\Sigma}(s)$ is the population's covariance across trials both for a stimulus s [106]. The LFI acts as a suitable lower bound to the Fisher information and is the most commonly used measure of coding fidelity in correlated variability analyses [2, 181, 216, 227, 66, 106, 164].

Experimental neuroscience datasets only consider discrete sets of stimuli, which is not amenable to the computation of LFI as posed in Equation 3.1. In particular, the derivative of the average neural response must be estimated by considering the neighboring pairs of stimuli. Thus, in practice, we calculate the *coarsened linear Fisher information* [93], which is defined for two stimuli s_1 and s_2 as

$$\mathcal{I}_{\text{coarse}}(\mathbf{f}_1, \mathbf{f}_2, \mathbf{\Sigma}_1, \mathbf{\Sigma}_2) = \left(\frac{\mathbf{f}_1 - \mathbf{f}_2}{\Delta s}\right)^T \left(\frac{\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2}{2}\right)^{-1} \left(\frac{\mathbf{f}_1 - \mathbf{f}_2}{\Delta s}\right) \tag{3.2}$$

where $\mathbf{f}_1 = \mathbf{f}(s_1)$, $\mathbf{f}_2 = \mathbf{f}(s_2)$, $\mathbf{\Sigma}_1 = \mathbf{\Sigma}(s_1)$, $\mathbf{\Sigma}_2 = \mathbf{\Sigma}(s_2)$, and Δs is the stimulus difference between s_1 and s_2, whose form may depend on the stimulus structure. In this work, we use the terms "coarsened LFI" and "LFI" interchangeably.

A formalism for assessing the optimality of neural data

Information theoretic analyses of neural data often ask whether the observed neural data is "optimal." In the case of correlated variability, the question can be posed as: are the observed covariances optimal from a decoding perspective? If we consider the linear Fisher

Information (LFI, Eq. 3.1) as the measure of optimality, what structure for $\mathbf{\Sigma}$ maximizes LFI? In this case, LFI can be infinitely large if $\mathbf{\Sigma} \to 0$ (or at least if the subspace of $\mathbf{\Sigma}$ defined by $\frac{d\mathbf{f}(s)}{ds}$ has zero variance). This answer is likely unsatisfying because neural systems have many sources of variability, and so expecting a neural system to become noiseless or exactly remove noise from a subspace seems implausible.

In this section, we develop the formalism that will allow us to assess the optimality of observed correlated neural variability. The formalism consists of first defining a covariance parameterization for $\mathbf{\Sigma}$, which is composed of constraints (fixed parameters) and degrees-of-freedom (free parameters). Ideally, these constraints and degrees-of-freedom have some biophysical interpretation. Then, a null model is defined by combining a covariance parameterization with a null distribution of the degrees of freedom. The distribution of some measure, such as the LFI, over the null model serves as a gauge to assess the optimality of the observed neural data.

We first consider an example to motivate our formalism. Then, we review the commonly used "fixed-marginal" model for correlated variability using our formalism and define two potential null models including the "shuffle" null model. Finally, we propose two covariance parameterizations and associated null models for assessing optimality which have more biophysical interpretability. In the following sections we will use the following terminology which we define here:

- **Covariance Parameterization:** a parameterization of $\mathbf{\Sigma}$ which can combine various constraints (fixed parameters) and degrees-of-freedom (free parameters).
- **Constraints:** elements of the covariance parameterization which are estimated from data and fixed.
- **Degrees-of-Freedom:** elements of the covariance parameterization which can potentially be modified or optimized to analyze a null model or optimality.
- **Optimality:** values for the degrees of freedom in a covariance parameterization which maximize a specified objective. Here we assess optimality using the Linear Fisher Information (LFI), although this formalism can be applied to other objectives.
- **Null Distribution:** distribution of a covariance parameterization's degrees-of-freedom.
- **Null Model:** combines a covariance parameterization with a baseline or uniform null distribution over the degrees-of-freedom.

The standard constraint considered for understanding correlated neural variability is to keep the per-neuron marginal distributions fixed. Since the LFI only depends on the covariance of the correlated variability, the fix-marginal parameterization is equivalent to constraining the per-neuron variances to be constant (equivalently, the diagonal of $\mathbf{\Sigma}$ is kept constant, $\text{diag}(\mathbf{\Sigma}) = \boldsymbol{\sigma}^2$). The corresponding degrees-of-freedom in this parameterization are the positive-definite pairwise correlation matrix, $\boldsymbol{\rho}$, specifically the symmetric, off-diagonal

Model	Constraints	DoFs	Null Distribution(s)
Fixed-Marginal	$\text{diag}(\boldsymbol{\Sigma})$	$\boldsymbol{\rho}$	shuffle, $U(\boldsymbol{\rho})$
Rotation	$\text{Evals}(\boldsymbol{\Sigma})$	$\mathbf{R}$	$U(SO(d))$
Factor Analysis	$\boldsymbol{\sigma}^2_{\text{FA}}$, $\text{Evals}(\mathbf{L}^T_{\text{FA}}\mathbf{L}_{\text{FA}})$	$\mathbf{R}$	$U(SO(d))$

Table 3.1: Existing and proposed null models and their breakdown in the constrains, degrees of freedom, distribution formalism. $\text{Evals}(\cdot)$ gives the eigenvalues of the argument and $\text{diag}(\cdot)$ gives the diagonal of the argument.

entries, $\rho_{i\neq j}$ which can vary (summarised in Table 3.1). Under this parameterization, the observed correlational structure can be compared to other proposed distributions of correlations.

When considering the structure that generates $\boldsymbol{\Sigma}$, it is desirable that the constraints and degrees-of-freedom be biophysically interpretable. Commonly, this can be achieved by considering the equations that define the mean-centered, single-trial response in terms of the degrees-of-freedom being considered. For the fixed-marginals parameterization, the distribution of the differences between the single trial responses $\mathbf{f}_t(s)$ and the mean response $\mathbf{f}(s)$ can be written in terms of a mean-zero multivariate normal distribution where the covariance is the element-wise product of the constrained marginal standard deviations, $\boldsymbol{\sigma}\boldsymbol{\sigma}^T$, and the free correlations, $\boldsymbol{\rho}$

$$\mathbf{f}_t(s) - \mathbf{f}(s) = \boldsymbol{\epsilon} \tag{3.3}$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\sigma}\boldsymbol{\sigma}^T \odot \boldsymbol{\rho}) \tag{3.4}$$

Given a parameterization (fixed-marginal) and a measure of coding fidelity (LFI), it is possible to find optimal covariance structures. In general, the values (or distribution) for the degrees-of-freedom that lead to optimality can be derived analytically or optimized numerically. This corresponds to finding the points, $\hat{\boldsymbol{\rho}}$, such that

$$\hat{\boldsymbol{\rho}} = \underset{\boldsymbol{\rho}}{\text{argmax}}\, \text{LFI}\left(\frac{d\mathbf{f}(s)}{ds}, \text{diag}(\boldsymbol{\Sigma}), \boldsymbol{\rho}\right). \tag{3.5}$$

Hu, Zylberberg, and Shea-Brown [85] show that, for the fixed-marginal model, the optimal correlational structure exist on the boundaries of the allowed values of $\boldsymbol{\rho}$ for several measures of coding fidelity including the LFI.

Novel null models allow the assessment of optimality in neural data

So far, we have have developed a formalism to define the optimal degrees-of-freedom for a specified parameterization. It is unlikely that observed neural data will precisely match the predicted optimal degrees of freedom, even if the biological system is behaving optimally, so

the predictions from Eq 3.5 cannot be used directly to assess optimality in data. In order to asses the optimality of a observed population of neurons, a *null model* must be constructed for a corresponding parameterization. In this formalism, constructing a null model corresponds to assuming a *null distribution* for the degrees-of-freedom of the covariance parameterization. The null distribution should correspond to some notion of "uniform" or "baseline" for the degrees of freedom.

For example, the *shuffle null model*, based on the fixed-marginal parameterization, posits that the baseline distribution of correlations is zero correlations. The shuffle null model compares the LFI of the observed response marginal distributions and correlations to the LFI of the responses under a distribution where the individual neural responses are independently trial shuffled, that is, with fixed-marginal variability, no underlying pairwise correlations, and empirical pairwise correlations only arising from finite sampling effects. This is analogous to defining the null distribution for the covariance as a Wishart distribution with scale matrix equal to a matrix with the diagonal entries of Σ and zeros elsewhere, although the shuffle model exactly preserves the marginal variance, unlike a Wishart distribution which allows sampling variability in both the variance and correlations. Under this choice of null model, the observed LFI can be considered optimal if it has a high percentile under the null distribution, and furthermore, optimal is considered specifically with respect to a distribution with no correlations.

Our first contribution is a related null model based on the fixed-marginal parameterization, where the correlations are chosen randomly from a uniform distribution over correlation matrices. This tests whether the observed correlation are optimal with respect to all correlations, rather than zero correlations. To our knowledge, this null model has not been considered before. Evaluating data under this null model answer the question of whether the observed correlations are optimal with respect to all possible correlations, not just zero correlations.

At another extreme, we can attribute all trial-to-trial variability to external sources that the network can shape or filter. To prevent trivial solutions, we propose a "rotation" parameterization that preserves the spectrum of the variability ($\text{Evals}(\mathbf{\Sigma}) = \boldsymbol{\lambda}$), but allows the network to change the loading of the variability onto the neurons (through a rotation, $\mathbf{R}$). This model was discussed by Hu, Zylberberg, and Shea-Brown [85], but not analyzed due it its incompatibility with the fixed-marginal constraint. Let $\mathbf{\Sigma} = \mathbf{L}^T\mathbf{L}$ be the Cholesky decomposition of the observed covariance matrix. If a rotation, $\mathbf{R}$, is applied to $\mathbf{L}$, the eigenvalues of $\mathbf{\Sigma}$ are preserved (the model constraint), while their loading onto the observed neurons is rotated (the degrees of freedom). The mean-centered single trial response can be written as a function of the full-rank external sources $\mathbf{z}$, loading matrix $\mathbf{L}$, and rotation matrix $\mathbf{R}$

$$\mathbf{f}_t(s) - \mathbf{f}(s) = \mathbf{R}^T\mathbf{L}^T\mathbf{z} \tag{3.6}$$

$$\mathbf{z} \sim \mathcal{N}(0, \mathbf{I}). \tag{3.7}$$

To optimize the rotation model, the eigenvector with the smallest eigenvalue can be rotated to align with $\frac{d\mathbf{f}(s)}{ds}$ which maximized the LFI. In addition to maximizing the LFI, the

optimal rotation can be constructed to be "minimal", that is, to have no off-axis rotation. To construct the rotation null model, a uniform distribution (Haar distribution) over special orthogonal rotations [190] is applied to the rotations (see Table 3.1 for summary).

As a parsimonious combination of these models, we propose using a Factor Analysis (FA) model to model correlated variability. Factor Analysis decomposes the observed correlated variability into two components: the first is per-neuron private variability, represented as a diagonal matrix $\text{diag}(\boldsymbol{\sigma}^2_{\text{FA}})$, and the second is a low-rank shared variability component, $\mathbf{L}^T_{\text{FA}}\mathbf{L}_{\text{FA}}$, where $\mathbf{L}_{\text{FA}} \in \mathbb{R}^{k \times d}$, $k < d$. We propose that the FA model has private variability and the spectrum of the shared component as constraints and the rotation of the shared components as the degrees-of-freedom, combining aspects of the fixed-marginal and rotation null models. The mean-centered single trial response can be written as a function of the private variances $\boldsymbol{\sigma}^2_{\text{FA}}$, low-rank external sources $\mathbf{z}$, loading matrix $\mathbf{L}_{\text{FA}}$, and rotation matrix $\mathbf{R}$

$$\mathbf{f}_t(s) - \mathbf{f}(s) = \mathbf{R}^T\mathbf{L}^T_{\text{FA}}\mathbf{z} + \boldsymbol{\epsilon} \tag{3.8}$$

$$\mathbf{z} \sim \mathcal{N}(0, \not\Vdash) \tag{3.9}$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \text{diag}(\boldsymbol{\sigma}^2_{\text{FA}})) \tag{3.10}$$

To our knowledge, there is no closed-form solution for $\mathbf{R}$ in the FA model to maximize LFI. Instead, to optimize the FA model, the rotation can be numerically optimized by gradient ascent. To construct the FA null model, a uniform distribution (Haar distribution) over special orthogonal rotations [190] is applied to the rotations (see Table 3.1 for summary).

Population statistics across dim-stims capture optimality under a null model

Each dataset can be described by a $D \times N$ design matrix $\mathbf{X}$, where D is the total number of samples and N is the number functional units in the population (Fig. 3.3a, left). We considered distributions of LFI across *dim-stims*, or sub-components of the design matrix. To create dim-stims, we first selected a *dimlet* of size d by subsampling d units from the population at random, resulting in the $D \times d$ design matrix $\mathbf{X}^d$ (Fig. 3.3, middle). Next, we created the dim-stim by further subsampling the design matrix according to a specific stimulus pairing. Specifically, we chose two neighboring stimuli, s_1 and s_2 (Fig. 3.3, middle), and isolated the samples of $\mathbf{X}^d$ corresponding to those stimuli, thereby creating a pair of design matrices $[\mathbf{X}^d_{s_1}, \mathbf{X}^d_{s_2}]$. The dim-stim maps to the task of discriminating between two neighboring stimuli using a sub-population's responses across trials to those stimuli, which can be visualized in the neural space (Fig. 3.3, right).

For each dataset, we considered dimlet dimensions $d = 3, \ldots, 15$. As we only allowed *neighboring* stimulus pairings, the number of available stimulus pairings for a dimlet was 6 (retinal), 12 (V1) and 29 (PAC). Note that the retinal and V1 stimulus sets are circular, providing an additional stimulus pairing. In the retinal and V1 datasets, we drew $d = 1000$ dimlets for each dimension d, and considered all stimulus pairings per dimlet, resulting in

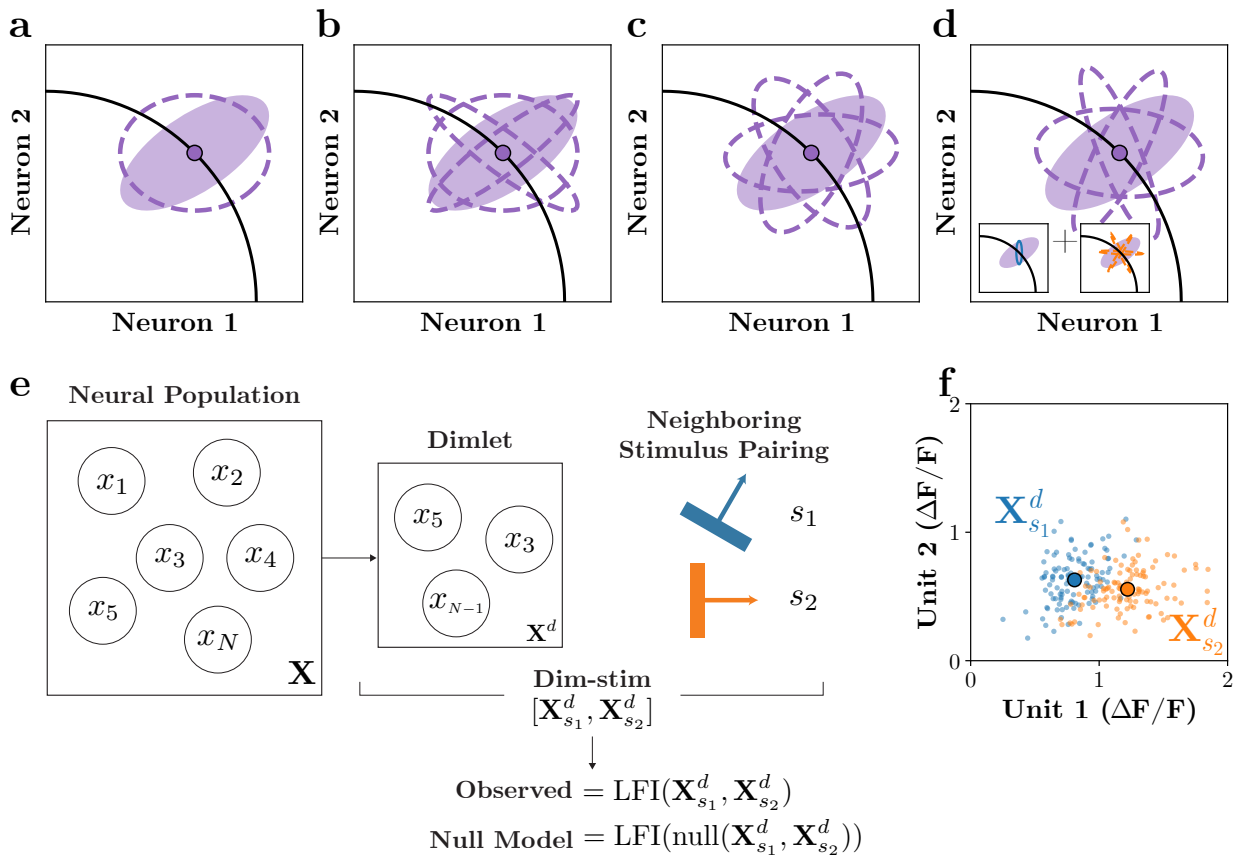


Figure 3.2: **Observed correlated variability has LFI percentiles below chance.** Each colums corresponds to one of the datasets. **a-c.** The LFI calculated on the observed data and null models for the retinal (**a**), V1 (**b**) and primary auditory cortex (**c**) datasets. Each plot depicts the LFI, plotted on a log-scale (y-axis) as a function of the dimlet dimension (x-axis). For the observed data (black), the solid line denotes the median LFI across dim-stims. For the null models (shuffle: gray, rotation: red, factor analysis: purple), solid lines denote the median across both dim-stims and repeats of the null distribution. Shaded regions bound the 40th and 60th percentiles of the LFI distribution. **d-f.** The distribution of LFIs across the shuffle, rotation, and factor analysis null models for a specific dim-stim. The observed LFI is denoted by the black dashed line in each plot. Percentiles are calculated as the fraction of the null model repeats that lie below the observed LFI, and are denoted in each plot's legend. **g-i.** Observed percentiles, for each dataset (columns) and null model (colors), across dimlet dimensions (x-axes). Solid line denotes the median observed percentile across all dim-stims, while shaded region bounds the 40th and 60th percentiles of the observed percentile distribution.

$1000 \times 6 = 6000$ dim-stims for the retinal dataset and $1000 \times 12 = 12000$ dim-stims for the V1 dataset. To manage computation time, we considered 3000 unique dim-stims for the

PAC dataset, selecting both the dimlet and stimulus pairing at random for each dim-stim.

For each dim-stim, we calculate its *observed LFI*, defined as $\mathcal{I}_{\text{coarse}}(\mathbf{f}_1, \mathbf{f}_2, \mathbf{\Sigma}_1, \mathbf{\Sigma}_2)$. Specifically, we computed

$$\mathcal{I}_{obs}(\mathbf{X}^d_{s_1}, \mathbf{X}^d_{s_2}) = \mathcal{I}_{coarse}\left(\text{mean}(\mathbf{X}^d_{s_1}), \text{mean}(\mathbf{X}^d_{s_2}), \text{cov}(\mathbf{X}^d_{s_1}), \text{cov}(\mathbf{X}^d_{s_2})\right) \tag{3.11}$$

$$= \left(\frac{\mathbf{f}^d_{s_1} - \mathbf{f}^d_{s_2}}{\Delta s}\right)^T \left(\frac{\mathbf{\Sigma}^d_{s_1} + \mathbf{\Sigma}^d_{s_2}}{2}\right)^{-1} \left(\frac{\mathbf{f}^d_{s_1} - \mathbf{f}^d_{s_2}}{\Delta s}\right) \tag{3.12}$$

where $[\mathbf{f}^d_{s_1}, \mathbf{f}^d_{s_2}]$ are the dim-stim average responses, $[\mathbf{\Sigma}^d_{s_1}, \mathbf{\Sigma}^d_{s_1}]$ are the dim-stim covariances, and Δs is the stimulus difference, or $\Delta s = |s_1 - s_2|$. When necessary, the stimulus difference was taken as a circular difference (retinal and V1 datasets). Since the LFI is scaled by the units of the stimulus differences, it is only meaningful to compare observed LFIs within a particular dataset.

Each null model acts on the design matrices of a dim-stim and outputs a distribution of covariance matrices. For example, the fixed-marginal null model shuffles the data within the design matrix, producing new design matrices $[\mathbf{X}^{d\,\prime}_{s_1}, \mathbf{X}^{d\,\prime}_{s_2}]$ and corresponding covariances $[\mathbf{\Sigma}^{d\,\prime}_{s_1}, \mathbf{\Sigma}^{d\,\prime}_{s_2}]$. We then calculate the LFI using the new covariance matrices. Each null model can be summarized as such: a sampled transformation is applied to the observed dim-stim, producing new sampled covariance matrices and therefore a sample of LFI from the null. The shuffle null model transformed the data directly, so we write its LFI as

$$\mathcal{I}_{\text{FM}}(\mathbf{X}^d_{s_1}, \mathbf{X}^d_{s_2}) = \mathcal{I}_{\text{obs}}\left(\text{shuffle}(\mathbf{X}^d_{s_1}), \text{shuffle}(\mathbf{X}^d_{s_2})\right). \tag{3.13}$$

Meanwhile, the uniform, rotation, and factor analysis null models transform the covariance or its parameterization directly, so we write their LFIs as:

$$\mathcal{I}_{\text{U}}(\mathbf{X}^d_{s_1}, \mathbf{X}^d_{s_2}) = \mathcal{I}_{\text{coarse}}\left(\mathbf{f}^d_{s_1}, \mathbf{f}^d_{s_2}, \text{sample}_{\text{U}}(\mathbf{\Sigma}^d_{s_1}), \text{sample}_{\text{U}}(\mathbf{\Sigma}^d_{s_2})\right) \tag{3.14}$$

$$\mathcal{I}_{\text{R}}(\mathbf{X}^d_{s_1}, \mathbf{X}^d_{s_2}) = \mathcal{I}_{\text{coarse}}\left(\mathbf{f}^d_{s_1}, \mathbf{f}^d_{s_2}, \text{rotate}(\mathbf{\Sigma}^d_{s_1}), \text{rotate}(\mathbf{\Sigma}^d_{s_2})\right) \tag{3.15}$$

$$\mathcal{I}_{\text{FA}}(\mathbf{X}^d_{s_1}, \mathbf{X}^d_{s_2}) = \mathcal{I}_{\text{coarse}}\left(\mathbf{f}^d_{s_1}, \mathbf{f}^d_{s_2}, \text{rotate}_{\text{FA}}(\mathbf{\Sigma}^d_{s_1}), \text{rotate}_{\text{FA}}(\mathbf{\Sigma}^d_{s_2})\right). \tag{3.16}$$

Equations (3.13-3.16) capture a single application of a null model. Specifically, shuffle() shuffles the neural data, $\text{sample}_{\text{U}}()$ samples a random off-diagonal correlation structure and applies it to the covariance, rotate() applies a rotation to the covariance, and $\text{rotate}_{\text{FA}}()$ applies a rotation to the shared component of the covariance. However, we were interested in characterizing the entire distribution of the null model. Thus, for each dim-stim, we applied 1000 repeats of the null model to obtain a null model distribution of LFIs. We then calculated *observed percentiles* as the fraction of repeats for which the observed LFI exceeded the null model LFI. Thus, each dim-stim has its own corresponding observed percentile, per null model.

Neural Recordings

We examined correlated variability in a diverse set of datasets, spanning distinct brain regions, animal models, and recording modalities. We used calcium imaging recordings from

Dataset	Animal	Recording	Stimulus	# Units	# Stimuli	# Trials/Stim
Retina	Mouse (Isolated)	Calcium Imaging	Drifting Bars	54	6	114
V1	Macaque	Single-Units	Drifting Gratings	106	12	200
PAC	Rat	μECoG	Tone Pips	65	30	60

Table 3.2: Experimental dataset summary.

mouse retinal ganglion cells, single-unit recordings from macaque primary visual cortex, and micro-electrocorticography recordings from rat auditory cortex. We briefly describe the experimental and preprocessing steps for each dataset. See Figure 3.2 and Table 3.2 for summaries of the datasets.

Recordings from mouse retina

Mouse retinal data was comprised of calcium imaging recordings from retinal ganglion cells isolated from mice. Retinal ganglion cells were presented with drifting bars at 6 unique angles (spanning 0° to 300°). Each angle was presented 114 times, for a total of 684 trials per cell. A total of 832 retinal ganglion cells were extracted, of which we analyzed 54 that exhibited tuning. Data was recorded by Summers & Feller.

Recordings from macaque primary visual cortex (V1)

Primary visual cortex data (V1) was comprised of spike-sorted units simultaneously recorded in anesthetized macaque monkey. This dataset contains recordings from three monkeys, of which the main text presents results from the first one (see Appendix for results on additional two monkeys). Recordings were obtained with a 10×10 grid of silicon microelectrodes spaced 400 μm apart and covering an area of 12.96 mm^2. A total of 106 units were isolated in the monkey. The monkey was presented with grayscale sinusoidal drifting gratings, each for 1.28 s. Twelve unique drifting angles (spanning 0° to 330°) were each presented 200 times, for a total of 2400 trials per monkey. Spike counts were obtained in a 400 ms bin after stimulus onset. The data was obtained from the Collaborative Research in Computational Neuroscience (CRCNS) data sharing website [192] and was recorded by Kohn and Smith (KS) [106]. Further details on the surgical, experimental, and preprocessing steps can be found in [180] and [102].

Recordings from rat primary auditory cortex (PAC)

Auditory cortex data (PAC) was comprised of cortical surface electrical potentials (CSEPs) recorded from rats with a custom fabricated micro-electrocorticography (μECoG) array. The μECoG array consisted of an 8×16 grid of 40 μm diameter electrodes. Anesthetized rats were presented with 50 ms tone pips of varying amplitude (8 different levels of attenuation, from 0 dB to -70 db) and frequency (30 frequencies equally spaced on a log-scale from 500 Hz to 32 kHz). We only used samples for the lowest 3 levels of attenuation since these evoked the

largest responses. Each frequency-amplitude combination was presented 20 times, for a total of $3 \times 30 \times 20 = 1800$ samples. The response for each trial was calculated as the z-scored to baseline, high-γ band analytic amplitude of the CSEP, calculated using a constant-Q wavelet transform. Of the 128 electrodes, we used 65, selecting those that recorded from primary auditory cortex. Data was recorded by Dougherty & Bouchard. Further details on the surgical, experimental, and preprocessing steps can be found in [61].

3.3 Results

An abundance of work has aimed to assess whether observed correlated variability is beneficial for neural coding. This question is a relative one, in which the observed data is compared to a specified benchmark. The benchmark must be chosen to adequately reflect what could be achievable by the neural system. If this is not the case, we may come to the conclusion that correlated variability is beneficial for neural coding, when in reality the observed correlated variability is sub-optimized. Ultimately, this becomes a question of whether correlated variability is structured optimally for decoding.

To answer this question, we developed a novel formalism to assess the optimality of correlated variability in neural data. The formalism consists of evaluating some measure of coding fidelity relative to a *null model*. We define a null model as a specific covariance parameterization (i.e., the identification of degrees of freedom for the correlated variability) coupled with a null distribution for those degrees of freedom. The goal of the null model, then, is to constrain some aspect of the data while perturbing other aspects. Thus, optimality in this framework is assessed by benchmarking the observed data relative to what could be achievable by probing the degrees of freedom.

The standard practice for evaluating coding fidelity under correlated variability is to shuffle the data across trials. The goal of trial-shuffling is to constrain the marginals (i.e., the means and variances) of the neural activities, but destroy any pairwise correlations. This approach lies within the optimality framework as the *shuffle null model*, consisting of a fixed-marginal covariance parameterization coupled with a null distribution obtained by shuffling the data (Fig. 3.2a).

However, the shuffle null model is only a weak test of optimality: it does not account for other possible correlational structures that could be achievable by neural systems. Here, we propose three null models that allow us to asses the optimality of the observed neural responses: the uniform correlation null model (Fig. 3.2b), the rotation null model (Fig. 3.2c), and the factor analysis null model (Fig. 3.2d). The uniform correlation null model maintains the marginal distributions, but allows for any off-diagonal noise correlation structure (Fig. 3.2b: dashed lines). The rotation null model maintains the eigenvalues of the correlated variability, but allows for any orientation of the covariance in the neural space (Fig. 3.2c: dashed lines). The factor analysis null model maintains the eigenvalues of a suitable shared sub-component of the covariance, but allows for any orientation of this sub-component in the neural space. Each of these null models have unique biological interpretations and provide

more suitable tests of optimality. To our knowledge, they have not been evaluated on neural data before.

We characterized the optimality of several neural datasets by evaluating their coding fidelity relative to each of the aforementioned null models. We first show that neural responses are largely suboptimal (worse than chance), across datasets. Second, we analyze the properties of derived optimal response distributions and find that biophysical constraints restrict the observed populations from achieving optimality.

Neural populations exhibit worse than chance coding fidelity according to novel null models

To characterize the optimality of a wide range of sub-population and stimulus settings, we performed a large scale experiment evaluating the LFI in both the observed data and null models. For each neural population, we randomly sampled *dimlets*, or sub-populations, of dimension d. We paired dimlets with a variety of neighboring stimulus pairings to obtain a subset of the neural responses which we call a *dim-stim* (Fig. 3.2e; see Methods). A dim-stim maps to the task of constructing a decoder for neighboring stimuli using a neural sub-population's responses across trials (Fig. 3.2f and Fig. 3.1e, h, k).

We calculated the LFI for each dim-stim, across dimensions and datasets. We refer to this quantity as the *observed LFI*. Next, we applied the null models repeatedly ($R = 1000$ times) to each dim-stim, and calculated the LFI for each repeat (see Methods). Thus, for each dim-stim, we obtain a single observed LFI, and a distribution of R LFIs for each null model. We summarized each null model by calculating the median LFI across the R repeats.

We compared the behavior of the observed LFI to those of the null models as a function of dimlet dimension (Fig. 3.3a-c). For each dataset, we generated a large number of dim-stims across a set of dimensions $d = 3, \ldots, 20$ (see Methods). The observed LFIs across dim-stims grows with dimlet dimension, as we might expect (Fig. 3.3a-c: black lines). Similarly, the null model LFIs grow with dimlet dimension. However, both the rotation and factor analysis null models clearly exhibit larger LFIs than the observed data, with the disparity increasing with dimlet dimension. The rotation null model achieves the highest median LFIs, indicating that this null model produces, in general, noise correlation structure with the highest discriminability (Fig. 3.3a-c: red lines). Meanwhile, the shuffle null model generally exhibits worse or comparable discriminability relative to the observed LFI at lower dimensions (Fig. 3.3a-c: gray lines). At higher dimensions, however, its LFIs begin to exceed the observed LFIs. We further observe differences across datasets. For example, the factor analysis null model (Fig. 3.3a-c: orchid lines) exhibits similar LFIs as the rotation null model for the retinal and PAC datasets. However, in the V1 data, its LFIs are more comparable to the observed and shuffle LFIs. Overall, Figure 3.3a-c demonstrates that the median LFIs of the rotation and factor analysis null models produce LFIs that generally exceed that of both the shuffle null model and the observed data.

We quantified the optimality of a dim-stim's representation, relative to a null model,

with its *observed percentile*. As described in the previous section, we calculated the observed LFI, as well as the LFI for each of the R repeats across the null model. The R LFIs constitute a null model distribution, which serves as a benchmark for the observed LFI. We calculated observed percentile as the fraction of the R repeats that the observed data outperformed, according to their LFIs. A larger observed percentile implies that the observed data possessed higher discriminability relative than most orientations provided by the null model, corresponding to optimal or near-optimal discriminability. On the other hand, a lower observed percentile implies that the dim-stim possesses sub-optimal discriminability.

Each null model exhibits distinct LFI distributions, with further variation depending on the dataset and dim-stim. Example null model distributions for a particular dim-stim are depicted in Figure 3.4d-f. The observed percentiles calculated in each example highlights that the performance can vary dramatically across dim-stims. For example, we observe highly sub-optimal performance (Fig. 3.3d), middling performance (Fig. 3.3e), and nearly optimal performance (Fig. 3.3f) as captured by the observed percentiles (Fig. 3.3d-f, legends).

The heterogeneity in observed percentiles motivates examining their behavior at the population level. Thus, on each dataset, we computed the distribution of observed percentiles across 1000 dim-stims per dimlet dimension, ranging from $d = 3$ to $d = 15$. The behavior of the median observed percentile (calculated across dim-stims) as a function of dimlet dimension is shown in Figure 3.3g-i. We found that, across the datasets, the shuffle null model has the largest observed percentiles, while the rotation null model has the lowest observed percentiles. This implies that, among the three null models, usage of the shuffle null model is most likely to imply optimality of the neural representations. Meanwhile, the factor analysis and rotation null models exhibit similar observed percentiles, with the factor analysis null model slightly higher across all three datasets (Fig. 3.3g-i, red and orchid lines). All observed percentiles decrease with dimlet dimension, implying that the neural representations become less optimal as the number of neurons increases. This decrease is expected as differential correlations induce information saturation in the populations.

Figure 3.3 highlights intriguing differences across datasets. In particular, the shuffle null model for the V1 data clearly exhibits the highest observed percentiles, indicating nearly optimal performance for small dimlet sizes (up to $d \approx 10$). Meanwhile, the shuffle null model in the primary auditory cortex data exhibits lower observed percentiles, with a larger spread, indicating a higher heterogeneity in the observed percentiles (Fig. 3.3i: gray shaded region). The shuffle null model for the retinal data has the lowest observed percentiles among the three datasets, exhibiting the smallest discrepancy between it and the other two null models. Meanwhile, the observed percentiles for the factor analysis and rotation null models are similar across the three datasets, with slightly different magnitudes. In particular, the retinal data exhibits the largest observed percentiles for these two null models, while the PAC data exhibits the smallest, going to zero around $d = 5$. This behavior roughly corresponds to the distribution of noise correlations amongst the three datasets (Fig. 3.1f, i, l), with the PAC data possessing the highest average noise correlation, and the retinal data possessing the lowest average noise correlation.

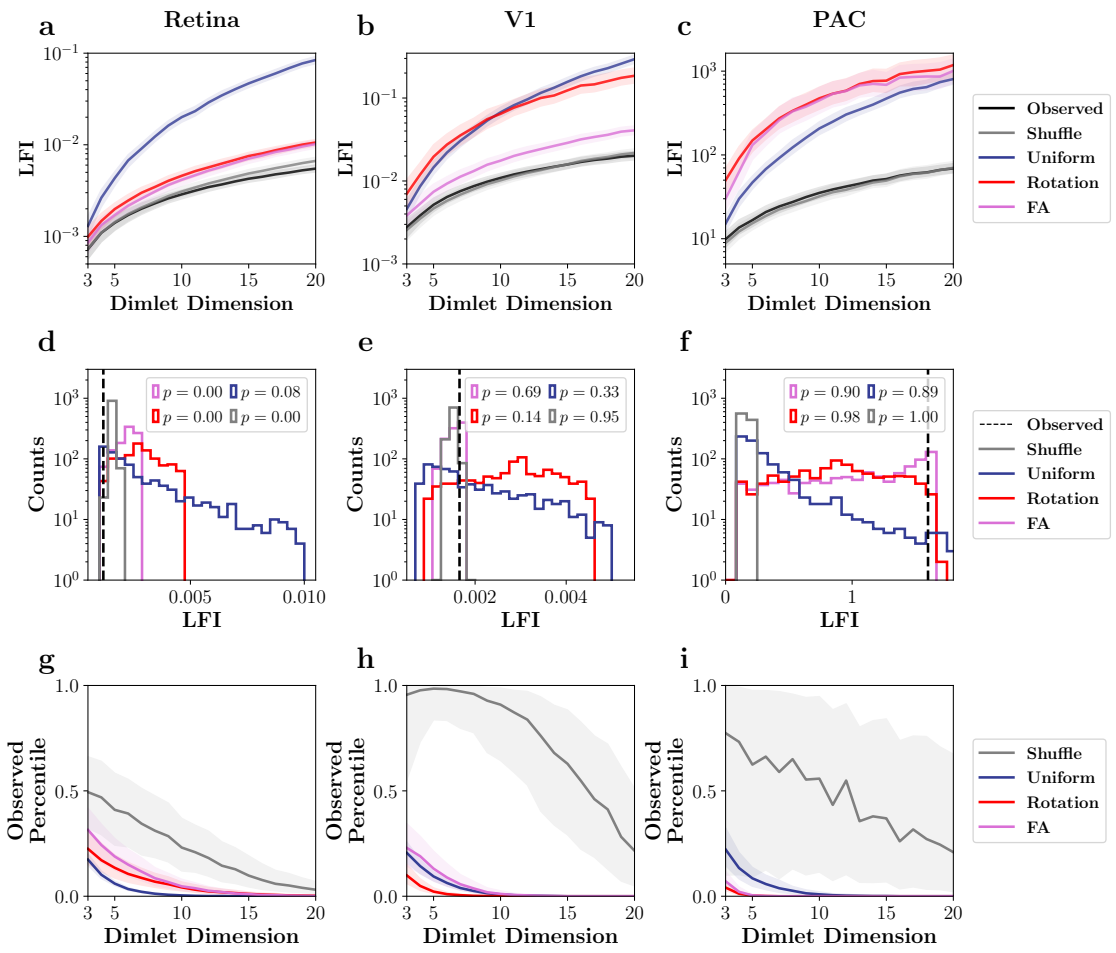


Figure 3.3: **Observed correlated variability has LFI percentiles below chance.** Each colums corresponds to one of the datasets. **a-c.** The LFI calculated on the observed data and null models for the retinal (**a**), V1 (**b**) and primary auditory cortex (**c**) datasets. Each plot depicts the LFI, plotted on a log-scale (y-axis) as a function of the dimlet dimension (x-axis). For the observed data (black), the solid line denotes the median LFI across dim-stims. For the null models (shuffle: gray, rotation: red, factor analysis: purple), solid lines denote the median across both dim-stims and repeats of the null distribution. Shaded regions bound the 40th and 60th percentiles of the LFI distribution. **d-f.** The distribution of LFIs across the shuffle, rotation, and factor analysis null models for a specific dim-stim. The observed LFI is denoted by the black dashed line in each plot. Percentiles are calculated as the fraction of the null model repeats that lie below the observed LFI, and are denoted in each plot's legend. **g-i.** Observed percentiles, for each dataset (columns) and null model (colors), across dimlet dimensions (x-axes). Solid line denotes the median observed percentile across all dim-stims, while shaded region bounds the 40th and 60th percentiles of the observed percentile distribution.

Optimal noise correlations are biologically implausible

We sought to understand why the observed correlated variability structure is highly suboptimal, as opposed to random or optimal. To do so, we compared the structure of the observed covariances to those of the optimal covariances under the rotation and factor analysis null models. Consider an example dim-stim for a dimlet of size $d = 3$, with low observed percentiles under both the null models (e.g., $p_{\mathrm{R}} = 0.0$ and $p_{\mathrm{FA}} = 0.002$). We plot the observed covariance structure, projected into two dimensions, in Figure 3.4a (black covariance denotes average covariance). Next, we compare the observed structure to that of the optimal structure, both within the rotation null model (Fig. 3.4b: red ellipse) and the factor analysis null model (Fig. 3.4c: orchid ellipse).

The observed correlated variability structure (Fig. 3.4a) clearly exhibits poor discriminability, because the variability is oriented parallel to the stimulus manifold (Fig. 3.4: black lines). The rotation null model, which has the greatest amount of flexibility in orienting the correlated variability, is oriented orthogonal to the stimulus manifold, as we might expect. Meanwhile, the optimal covariance structure under the factor analysis null model lies more orthogonal to the stimulus manifold, but not to the degree of the rotation null model. It is clear that the optimal covariance structures are markedly different from that of the observed covariance. In particular, the optimal structures project more variance into the negative neural space, which is an unattainable region for spiking units (Fig. 3.4: gray regions in marginal distributions). Furthermore, the rotation and factor analysis optimal covariances possess different per-neuron variances (Fig. 3.4: black side bars). Both of these features are biological restrictions on neural activity, and may impede a neural system from obtaining an optimal correlated variability structure.

We aimed to quantify the degree to which the biological implausibility of the optimal covariance structures correlated with the optimality of the observed firing for each dim-stim. That is, we examined whether the cases where a dim-stim exhibited optimal, or near optimal coding performance – as measured by the observed percentile – corresponded to scenarios where optimal covariances were achievable within biological constraints. We first examined the degree to which the Fano factor is preserved under optimal orientations. Then, we examined a quantity we refer to as the *excess negative density* (END), which measures the degree to which an optimal covariance structure places probability density in low or negative neural activity regions, relative to the observed data. In both cases, we found that dim-stims exhibited increased coding performance whenever the END or Fano factors were biologically plausible.

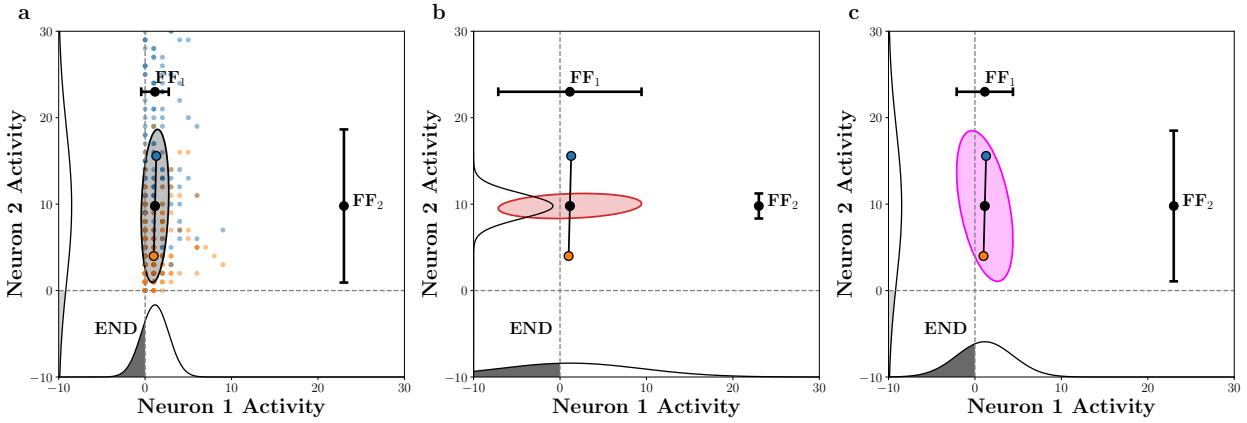


Figure 3.4: **Biological constraints may not be preserved under null model transformations.** Example dim-stim. Fits and optimal covariances are from a $d = 3$ dimlet projected into the first 2 neurons. The marginal probabilities of the multivariate gaussian fits are shown along the axes and the areas with values less than the empirical 1% are shaded grey with the maximum excess negative density in dark grey (annotated with "END"). The marginal means and standard deviations (for Fano factor calculations) are shown in the black error bars (annotated with "FF" and neuron number). **a:** Neuron responses to stimuli 1 and 2 (orange and blue circles) and the respective means (outlined circles). Their joint meant is the green circle and the observed mean covariance is in green. **b:** Covariance and marginals from the optimal rotation. **c:** Covariance and marginals from the optimal Factor Analysis rotation.

Biologically achievable Fano factors restrict optimality

The Fano factor quantifies the variability of neural units relative to their average activity. Typically, Fano factors have been observed to be around 1. However, Figure 3.4 demonstrates that the optimal covariance orientations under a null model may possess substantially different Fano factors. Thus, we aimed to assess whether biologically unachievable Fano factors shared any relation with the sub-optimality exhibited by the neural codes in our analyses.

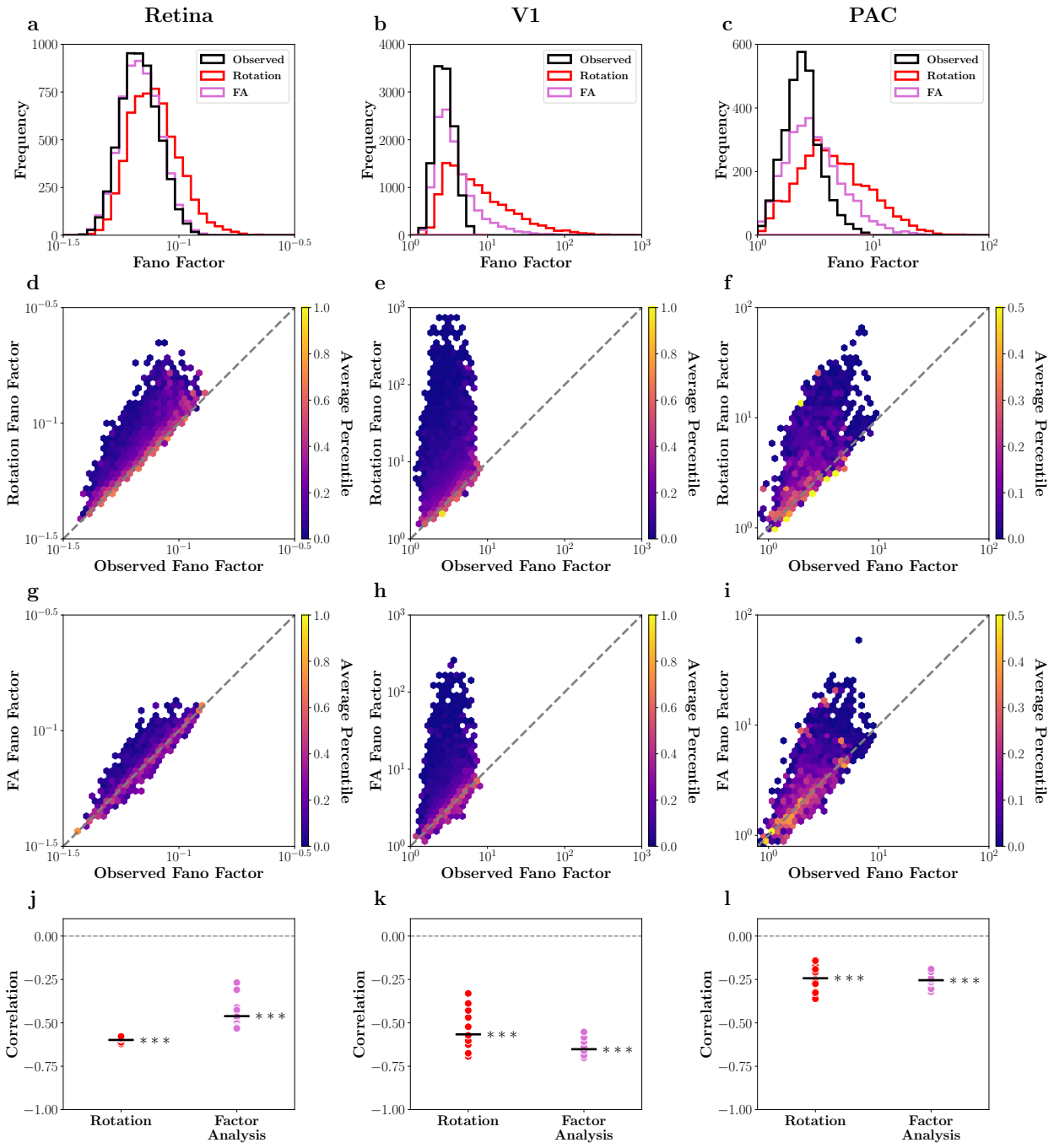


Figure 3.5: **Biologically achievable Fano factors restrict optimality.** Each column corresponds to a separate dataset. **a-c.** The distribution, across dim-stims, of Fano factors from: the observed data (black), the optimal noise covariance under the rotation null model (red), and the optimal noise covariance under the factor analysis null model (orchid). Fano factors are shown on a log-scale (x-axis). Continued on the following page.

Figure 3.5: **Continued from previous page. d-f.** Fano factors for optimal noise covariance under the factor analysis null model plotted directly against the observed Fano factor, across dim-stims. Each dim-stim is summarized by its mean Fano factor across the dimlet. Cells in each hexagonal bin are colored according to the average observed percentile, relative to the factor analysis null model (color bar to the right). Note that the colorbar for the AC data (**f**) is on a different scale. **g-i.** Same as **d-f**, but comparing the observed Fano factors to the those of the rotation null model. **j-l.** The correlation between the observed percentile and the logarithm of the Fano factor ratio, $\log\left(\mathrm{FF}_{\mathrm{null}}/\mathrm{FF}_{\mathrm{obs}}\right)$. Each point denotes a correlation for a different dimlet dimension. Significance markers denote $p < 10^{-3}$ (one-sample t-test from $\mu = 0$).

We summarized each dim-stim with an aggregate Fano factor, by averaging the Fano factors of that dim-stim's individual units. We repeated this process for the optimal noise covariances under each null model, using the variances from the diagonal of the optimal noise covariance matrix directly when calculating Fano factors. As an example, we show the distribution of Fano factors across dim-stims for $d = 3$ in Figure 3.5a-c. We observed that the rotation and factor analysis null models generally exhibited larger Fano factors, with substantially longer tails. This indicates that optimal orientations under the null models typically possess higher Fano factors, with high variance directions being assigned to units with lower activity. Furthermore, we observe that discrepancy between the observed and null model Fano factor distributions is largest for the V1 and PAC data.

We aimed to determine whether the Fano factor distribution related to the optimality of the neural code. To this end, we directly compared the null model Fano factors to the observed Fano factors in Figure 3.6d-f (rotation null model) and Figure 3.6g-i (factor analysis null model). We color-coded each bin of the 2-d histogram according to the average observed percentile of dim-stims within the bin. Thus, in Figure 3.6d-i, bins with lighter colors contain dim-stims whose coding performance is closer to optimal. We observed that dim-stims whose observed Fano factors are similar in magnitude to the optimal null model typically exhibit higher average observed percentiles, indicating that their neural codes are closer to optimal (Fig. 3.6d-i: lighter color bins near gray identity line). Meanwhile, dim-stims whose observed Fano factors were substantially smaller than the (Fig. 3.6d-i: darker color bins). This indicates that if the optimal noise covariance exhibits biologically plausible Fano factors, then the neural representations typically achieved better than sub-optimal (and in some cases, close to optimal) decoding performance. This was less true for the PAC data, which typically exhibited the lowest observed percentiles (Fig. 3.6f, i: see colorbar range).

Lastly, we quantified the correspondence between Fano factor and observed percentile. Specifically, we calculated the log-ratio of Fano factors, $\log(\mathrm{FF}_{\mathrm{null}}/\mathrm{FF}_{\mathrm{obs}})$ for each dim-stim and null model. When the null model Fano factors are substantially different from the observed Fano factors, this quantity is of larger magnitude. In the case of Fig. 3.6, the null model Fano factors were virtually never lower than the observed Fano factors. Thus, the log-ratio was almost always positive, with larger values corresponding to decreased biological

plausibility. Thus, we calculated the Spearman correlation between the log-ratio and the observed percentile, across dim-stims, and for each dimlet dimension d (Fig. 3.6j-l). For each null model and dataset, we observed negative correlations that were significantly lower than zero ($p < 10^{-3}$, one sample t-test). These correlations imply that the log-ratio decreases with increased observed percentile. Thus, when the optimal Fano factor is similar to the observed Fano factor (i.e., the log-ratio is of lower magnitude), the neural codes tend to be closer to optimal (i.e., the observed percentile is higher).

Excess negative density correlates with highly sub-optimal coding performance

A covariance arrangement that places density in negative neural space can be interpreted as less biologically plausible, because negative activity is either unachievable (for single-units) or highly unlikely (calcium imaging or μECoG). In other words, such covariance arrangements do not capture the underlying marginal statistics of a dimlet. The shuffle null model will necessarily reproduce the observed marginals, because it only changes correlational structure. The rotation and factor analysis null models, however, can produce covariance ellipses that have different marginal distributions. Thus, some optimal arrangements may orient variance in the negative or low-activity regions of the neural space.

To quantify this phenomenon, we calculated the excess negative density (END), which captures the degree to which a null model produces diverging marginal distributions for the dimlet it models. We calculate the END as follows. For each dim-stim, we calculated, r_i, the neural activity at the 1st percentile, for each neuron i. We then computed c_i, the cumulative density at r_i for a Gaussian obtained from either the observed data or the optimal orientation under the null model (Fig. 3.4: shaded regions in marginals). The END, then, was defined as the *maximum* c_i among the neurons in the dimlet (Fig. 3.4: dark gray shaded regions). Thus, a larger END implies that the covariance places an excess of density in the negative or low-activity regions for at least one dimension of the neural space. On the other hand, a lower END is more biologically plausible, as this implies there is less negative density.

We calculated the END for the observed fit, the optimal rotation fit, and the optimal factor analysis fit, across dim-stims, dimensions, and datasets. The distribution of ENDs across dim-stims at $d = 3$ is depicted in Figure 3.6a-c. We observe that the observed fits exhibit the lowest ENDs, as we might expect (black lines). Meanwhile, the optimal rotation null model covariances exhibit the largest ENDs (red lines), with the optimal factor analysis covariances lying in the middle (orchid lines). Interestingly, the V1 and primary auditory cortex data exhibit larger ENDs than the retinal dataset. This implies that the dim-stims are more likely to contain units that exhibit large differences in activity.

Next, we examined how the END behaved as a function of each null model's percentile (for $d = 3$). Specifically, we plotted the END of rotation and factor analysis null models against their corresponding observed percentiles as a 2D histogram (Fig. 3.6d-f: bottom rows). Across all datasets, we observe a clear, inverse relationship: the END generally decreases with the observed percentile (Fig. 3.6d-f: red and orchid lines). This implies that, in dim-stims where the observed data is close to optimal, the END is small, or more

biologically plausible. As a baseline, we examined the relationship between the observed END and the observed percentiles for each null model (Fig. 3.6d-f: top row). We observe either no relationship (retinal and PAC data) or a more muted inverse relationship (V1), implying that the relationship we observe is not simply by chance.

We quantified the relationship between the END and observed percentile with the Spearman correlation across dim-stims. Furthermore, we calculated the correlation at each dimlet dimension d. We compare the distribution of correlations across dimensions between the null model and its corresponding observed data in Figure 3.6g-i. We observe negative correlations for each dataset, confirming the inverse relationship between END and observed percentile. Furthermore, the rotation and factor analysis null models each exhibit significantly lower correlations than their baseline counterparts ($p < 10^{-3}$, Wilcoxon rank-sum test). Thus, dim-stims with lower, more biologically plausible ENDs are more likely to exhibit more optimal neural representations as measured by the observed percentiles. We note that at d increases, the correlations decrease, since observed percentile decreases with dimension (Fig. 3.3).

3.4 Discussion

Since correlated variability is prevalent in neural recordings, it has been the subject of studies looking to understand their mechanistic sources, implication for neural computation, and modulation by brain-state and behavior. To assess the significance of the observed correlated variability, the *shuffle* null model is typically used. This null models compares the discriminability of observed correlations, as measured by the Linear Fisher Information (LFI), to a null model which preserves the per-unit variance but zeros out the pairwise correlations. The comparison with only a distribution near zero correlations limits the value of the shuffle null model in assessing optimality. To close this gap, we proposed three null models which allow the optimality of observed correlated variability to be assessed: the uniform-correlation, rotation, and factor analysis (FA) null models.

Using these null models, we found that observed neural activity across three datasets and all null models had discriminability consistently lower than chance. As the dimensionality of the neural activity increased this effect was more pronounced. At higher neural dimensions, it is expected that the observed correlational structure would become highly sub-optimal since the differential correlations have a variance that scales with the neural dimensionality (other directions will generally have constant variance as a function of dimensionality) and they are exactly oriented in the information limiting direction. In this case, many deviations from the observed correlational structure would lead to increased discriminability. However, recent work has shown that information limiting correlations do not cause saturation until neural dimensionalities in the hundreds or several thousands [23, 93, 160], not tens, as found in this work.

In order to understand the below-chance observed discriminability across null models, we evaluated the characteristics of the optimal covariance structure across dim-stims and found

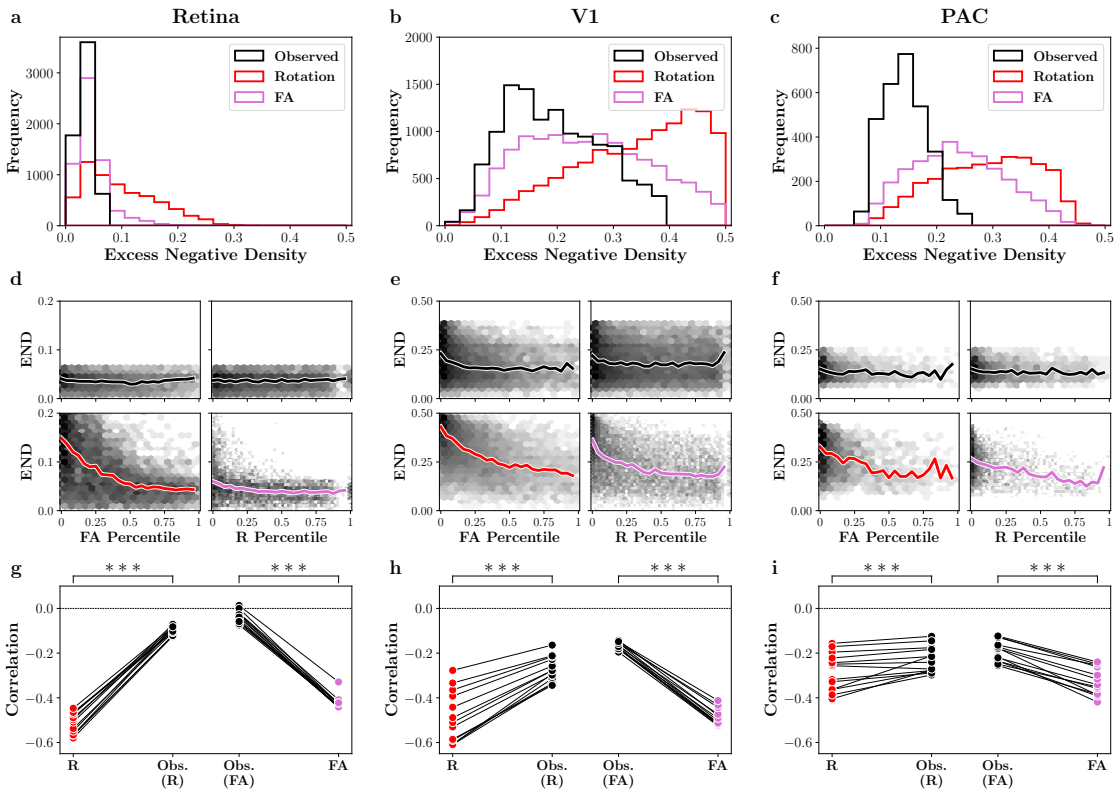


Figure 3.6: **Excess negative density correlates with worse than chance coding performance.** Each column corresponds to a separate dataset. **a-c.** The distribution, across dim-stims at $d = 3$, of excess negative densities for the observed data (black), the optimal noise covariance under the rotation null model (red), and the optimal noise covariance under the factor analysis null model (orchid). A larger excess negative density can be interpreted as less biologically plausible. **d-f.** The END from the previous subplots, compared to the observed percentile in a 2-d hexagonal histogram. For each subplot, the left column compares the the observed END (top) and rotation END (bottom) to the rotation observed percentile. The right column, meanwhile, compares the observed END (top) and FA END (bottom) to the FA observed percentile. Each bin's color is scaled according to its log-count. Colored lines denoted a rolling median of the END, binned according to the observed percentile. **g-i.** The Spearman correlation between the END and observed percentile, calculated across dim-stims. Each point denotes the correlation calculated at a different dimlet dimension, from $d = 3$ to $d = 15$. Red points denote the correlation between the rotation END and rotation observed percentile, while orchid points denote the correlation between the FA END and FA observed percentile. Each point is matched to a correlation (black lines) calculated between the observed END and corresponding null model percentile. Significance markers denote $p < 10^{-3}$ (Wilcoxon signed-rank test).

that a consistent picture emerges across datasets: when observed neural activity is more optimal, the optimal correlational structure is biophysically plausible, and when the observed neural activity has discriminability below chance, the optimal correlational structure is not plausible. When analyzed with respect to the rotation and FA null models the correlational structures that would be optimal would require marginal unit activity distributions that are highly divergent from observed biological distributions as assessed by the excess negative density and Fano factor. All three of the neural response modalities (calcium imaging, single unit spike counts, and high gamma amplitude) have highly skewed evoked responses with few or no negative values and longer tails to positive values. Achieving the optimal correlational structure might require highly bi-modal distributions of activity for a single stimulus, which is atypical for neurons in early sensory areas responding to simple parametric stimuli.

We observed worse than chance coding performance for each null model we proposed. However, the magnitude of the optimality (or lack thereof), as measured by the observed percentile, differed across null models and brain region. We consistently observed lower observed percentiles for the rotation null model compared to the factor analysis null model. In other words, the rotation null model was able to obtain orientations with greater LFI than the factor analysis null model. This observation is to be expected since the rotation null model can more flexibly achieve optimal orientations than the factor analysis null model, since the latter is limited to rotating a sub-component of the correlated variability. Interestingly, the observed correlations for the uniform null model were the lowest for the retinal data, and highest for the PAC data. This observation matches with the distribution of noise correlations in each dataset (Fig. 3.1f, i, l). The retinal dataset exhibits, on average, the smallest magnitude noise correlations, while the PAC datasets exhibits the largest. With a correct orientation in the neural space, a covariance only needs a larger condition number (i.e., the ratio of the largest to smallest eigenvalues) to improve its LFI. Since the rotation null models can freely access any orientation in neural space, it will exhibit larger LFIs in general when the data possesses higher noise correlations (thereby having larger condition number). Thus, the rotation and factor analysis null models exhibit larger LFIs when the data has larger noise correlations, explaining the discrepancies in observed percentiles across the datasets.

In this work, we have proposed three novel null models which are designed to assess the optimality of observed correlated variability. This invites the questions: is there a null model which can subsume all of these possibilities that can be applied in general, or is there and endless list of null models that need to be tested against? We suggest that the answer is "no" to both. Like picking a model, choosing a null model to assess optimality should be tailored to the particular question at hand and any potential interventions that can be applied to the neural system. For example, the mammalian retina does not receive feedback from cortex and the stimuli to the whole system can be carefully controlled. In this case, it may be most relevant to test the optimality of the recurrent processing systems rather than shared input from other areas. Therefore, the uniform correlation null model would be the relevant test. Alternatively, areas A and B are recurrently connected, and if the activity in areas B is being modulated optogenetically while areas A is being recorded, the rotation or

FA models may be more relevant to understand the impact of incoming shared variability on discriminability.

The theory of differential correlations identified a particular subspace in the neural space that limits the growth of information in a neural system. One possible source of these differential correlations lies in shared input noise, or noise carried by the stimulus. A neural system cannot do anything to prevent the onset of differential correlations. Thus, extensions of the null models could consider limiting the null distribution to avoid perturbing the differential correlation distribution. In practice, this is difficult, as identifying differential correlations requires recording on the order of thousands of neurons [128, 93, 160]. However, for a hypothesized set of differential correlations, the rotation null model could be applied only on the remaining component of the noise correlation covariance. These rotations would serve as a more suitable test of optimality, since they accurately reflect what is biologically achievable by a neural circuit.

Conclusion

We have demonstrated that neural activity is decisively sub-optimal using a novel framework. This required the development of novel null models, to which we applied large scale analyses across several datasets. This has important implications for the study of correlated variability. This concludes our analysis on correlated variability from a decoding perspective. We now turn to how correlated variability as a structure of neural activity impacts the fitting of phenomenological models of neural activity.

Chapter 4

Improved inference in coupling, encoding, and decoding models and its consequence for neuroscientific interpretation

Chapter Co-authors
Jesse A. Livezey
Maximilian E. Dougherty
Bon-Mi Gu
Joshua D. Berke
Kristofer E. Bouchard

Correlated structure generally impedes the fitting of phenomenological models, because it introduces correlations among predictive features. In this chapter, we seek to develop improved inference techniques that are stable to such structure in common systems neuroscience models. If baseline procedures suffer from the prevalence of correlated variability, then their improper parameter estimates may hamper model interpretation. Thus, we seek to determine how such improved inference procedures change neuroscientific interpretation relative to traditional approaches.

4.1 Introduction

Neuroscience is undergoing a rapid growth in the size and complexity of experimental and observational data [172, 124]. Realizing the benefits of these advances in data acquisition requires improvements in the statistical models characterizing the data, as well as the inference procedures used to fit those models [189, 44]. For example, generalized linear models are appealing because the model parameters can be interpreted to gain insight into the un-

derlying biological processes that generated the data [185, 139, 200, 152]. However, even for this ubiquitously used class of models, the impact of an inference procedure's statistical properties on neurobiological interpretation is poorly appreciated.

These issues are particularly salient in systems neuroscience, where parametric models are often used to understand how neural activity is modulated by external factors (e.g., stimuli or a behavioral task) and internal factors (e.g., other neurons) [145, 98]. The fitted parameter values, therefore, specify which factors are important in modulating neural activity, and how important they are. The specific relationships that a parametric model describes ultimately frames how the model will be interpreted in a neuroscientific context, emphasizing the importance of accurate parameter inference.

For example, functional coupling models (Fig 4.1a) capture the statistical dependencies between different functional units in the brain, at scales ranging from single units to functional areas [185, 139, 200, 152, 15, 223, 188, 182]. These models can be used to construct networks [25, 42], which are analyzed with an assortment of tools from graph theory to characterize the population [26, 20]. Additionally, functional coupling networks are related to structural connectivity [125], used to assess directed influence amongst neurons (i.e., effective connectivity) [70, 173], or related to external factors such as behavior, genetics, aging, or psychiatric conditions [103, 71, 9]. Encoding models map the dependence of a brain signal (e.g., neuronal spikes) on external factors, such as stimuli (Fig 4.1b) [57, 169, 198]. An example encoding model is a spatio-temporal receptive field of a visual cortex neuron, which maps the image space to the neuronal response (Fig 4.1b, right) [178, 195, 88]. More complex encoding models of neural population data can be used to test theoretical and computational theories of neural coding [153, 205, 225]. On the other hand, decoding models map brain signals to external factors, using the activities in, e.g., a neural population, to predict a stimulus or task-relevant behavioral condition (Fig 4.1c) [75, 131, 82, 34, 33, 118, 6]. A common linear decoding model is the extraction of a hyperplane in the neural activity space, which provides a decision boundary for one of two behavioral conditions or stimuli (Fig 4.1c, right: s_1, s_2) [103, 215, 155]. Recent work has explored more complex decoders, using artificial neural networks [118] or predictive latent representations [143]. Using decoding models for brain-computer interfaces has both clinical uses and scientific implications for understanding learning and motor control [206, 46]. Since these models are used to make scientific conclusions about the function of the brain, understanding the stability, accuracy, and parsimony of the inference procedures and resulting models is of paramount importance.

The utility of parametric models hinges on the assumption that the inference procedure used to fit them selects the correct parameters (i.e., specified as zero or non-zero) and properly estimates their values. The statistical consequences of improper selection are false positives or false negatives (Fig 4.1d), while poor estimation results in high bias (Fig 4.1e: e.g., β_1) or high variance (Fig 4.1e: e.g., β_6). The neuroscientific consequences of statistical inference lie in the interpretation of the fitted parametric model. Selection informs which internal and external factors are relevant for predicting neural activity, and estimation specifies their relative importance. Importantly, accurate selection is not a natural byproduct of predictive capacity, as cross-validated predictive accuracy is often a poor criterion for feature selection.

Specifically, model selection by held-out cross-validation predictive accuracy lacks guarantees on consistency and has been implicated in producing false positives [177, 222, 176, 209, 117]. Thus, validating that an inference procedure can reliably select and estimate a model's parameters is vital to ensure that they motivate correct conclusions about neural activity.

These issues imply that, when fitting parametric models in a scientific context, multiple goals beyond predictive performance must be balanced to produce a scientifically meaningful model. In particular, achieving a parsimonious model, which uses the fewest number of features to sufficiently predict the response variable (i.e., finding the "simplest"), has long served as a goal in statistical model selection [170]. One approach to model parsimony relies on the imposition of sparsity during feature selection, which has the added benefit of identifying a small subset of predictive features, facilitating the interpretability of the model [196, 81]. This is particularly relevant in high-dimensional settings where there are few task-relevant features and strong priors from domain knowledge for selection may not exist. Another desired property is stability, or the reliability of an inference algorithm when its inputs are slightly perturbed [218, 35]. For a model to be interpretable, its parameters must be robust to the often noisy processes that generated the data. Thus, encouraging stability in a model's parameter inference procedure will ensure that the features describing the relevant signal are selected and their correct contributions are properly estimated [113, 18]. Until recently, inference procedures that sufficiently balanced selection and estimation, predictive performance, and stability were lacking. This raises the question of whether the usage of traditional inference procedures in systems neuroscience has adversely impacted neuroscientific interpretation and data-driven discovery.

Our recently introduced Union of Intersections (UoI) is an inference framework based on stability principles which enhances inference in a variety of common parametric models [31]. The properties characterizing UoI models — sparsity, stability, and predictive accuracy — are well-suited to data-driven discovery in neuroscience, due to the high dimensionality and many sources of variability in these datasets. Furthermore, UoI is a frequentist approach, similar to the predominant traditional approaches used by neuroscientists (though we note recent development of Bayesian inference algorithms that also perform well in these settings [47, 224, 87]). Thus, we used UoI to assess whether common approaches to parameter inference in models are susceptible to improper feature selection and estimation, and if so, assess the consequences for model interpretability in a neuroscience context.

In this work, we used the UoI framework to fit functional coupling, encoding, and decoding models to diverse neural data in an effort to elucidate the impacts of precise selection and estimation on neuroscientific interpretation. We found that, compared to baseline procedures (e.g., ℓ_1-regularization), we obtained models with enhanced sparsity, improved stability, and significantly different parameter distributions, while maintaining predictive performance across recording modality, brain region, and task. Specifically, we obtained highly sparse coupling models of rat auditory cortex, macaque V1, and macaque M1 without loss in predictive performance. These models were used to construct functional networks that exhibited enhanced modularity and decreased small-worldness. We built parsimonious encoding models of mouse retinal ganglion cells and rat auditory cortex that more tightly

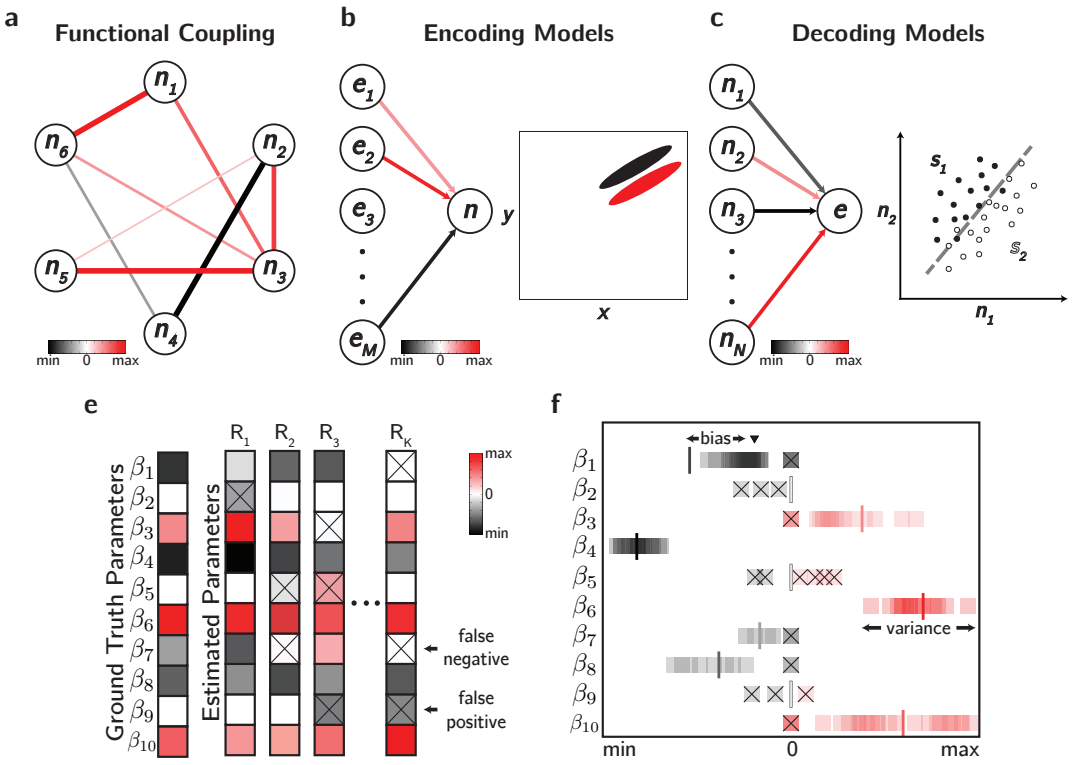


Figure 4.1: **Parametric models and statistical inference in systems neuroscience. a-c.** Three examples of parametric models widely used in systems neuroscience. **a.** Functional coupling models characterize the statistical relationships between neurons in a population. **b.** Encoding models map M internal or external factors $\{e_i\}_{i=1}^{M}$ to a neuronal response n. **c.** Decoding models map the activities of N neurons in a population $\{n_i\}_{i=1}^{N}$ to an internal or external factor e. **e-f.** Quantification of statistical selection and estimation performance. **e.** The ground truth values of the parameters in an example model, given by the first column, along with estimated values across K different resamples of the data, denoted by R_1, R_2, ..., R_K. **f.** The distribution of estimated values for each parameter in the ground truth model of the previous panel, with the true values denoted by vertical lines. For β_1, ▼ denotes the mean estimated value across resamples. False positives and false negatives are denoted with an $\times$.

matched with theory. These models were able to predict held-out neural responses with parameters that were as simple as possible, but no simpler. Lastly, we decoded task-relevant external factors from rat basal ganglia activity using fewer single units than baseline models. Overall, by utilizing improved inference algorithms during the fitting of parametric neural models, we constructed more sparse and stable models. We assessed the neuroscientific consequences of using these models, finding notable changes in secondary analyses.

4.2 Methods

Our goal is to demonstrate how the statistical properties of inference algorithms impact the fitting and interpretation of diverse parametric models commonly used in neuroscience. The main tools we use for this purpose are algorithms based on the Union of Intersections framework [31, 201, 163]. Thus, we organize the Methods as follows. First, in Section 4.2, we introduce the Union of Intersections framework, while providing other relevant background. Second, in Section 4.2, we describe a large-scale synthetic experiment comparing a specific UoI algorithm, $\text{UoI}_{\text{Lasso}}$, versus other algorithms on a synthetic dataset to motivate UoI's usage on neural datasets. Third, in Section 4.2, we describe the neural datasets on which we performed the model-fitting and subsequent analyses. Lastly, in Section 4.2, we provide the details of those analyses, outlining how model-fitting was performed for each coupling, encoding, and decoding model. We provide further details on subsequent analyses performed on the fitted models, such as statistical tests and network construction for coupling models.

The Union of Intersections framework balances sparsity, stability, and predictive performance

Union of Intersections (UoI) is not a single method or algorithm, but a flexible framework into which other algorithms can be inserted for enhanced inference. In this work, we apply the UoI framework to generalized linear models, focusing on linear regression ($\text{UoI}_{\text{Lasso}}$), Poisson regression ($\text{UoI}_{\text{Poisson}}$) and logistic regression ($\text{UoI}_{\text{Logistic}}$). We refer the reader to UoI variants of other procedures, such as non-negative matrix factorization [201] and column subset selection [31].

Consider the general problem of mapping a set of p features $\mathbf{x} \in \mathbb{R}^{p\times 1}$ to a response variable $y \in \mathbb{R}$, of which we have N samples $\{\mathbf{x}_i, y_i\}_{i=1}^N$. For convenience, we focus on linear models, which require estimating p parameters $\boldsymbol{\beta} \in \mathbb{R}^{p\times 1}$ that linearly map $\mathbf{x}_i$ to y_i. We describe the UoI framework in this context, which involves the algorithm $\text{UoI}_{\text{Lasso}}$. The steps we detail, however, extend naturally to other penalized generalized linear models [67]. Typically, the mapping in linear models is corrupted by i.i.d. Gaussian noise ϵ:

$$y = \boldsymbol{\beta}^T \mathbf{x} + \epsilon. \tag{4.1}$$

The parameters $\boldsymbol{\beta}$ can be inferred by optimizing the traditional least squares error on y:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} (y_i - \boldsymbol{\beta}^T \mathbf{x}_i)^2, \tag{4.2}$$

where i indexes the N data samples. The UoI framework combines two techniques — regularization and ensemble methods — to balance sparsity, stability, and predictive performance, thereby improving on the traditional least squares estimate (Fig 4.2a).

Structured regularization, or the inclusion of penalty terms in the objective function to restrict the model complexity, can be useful when a subset of the β_i are exactly equal to

zero, i.e., $\boldsymbol{\beta}$ is sparse. Sparsity implies that some features are not relevant for predicting the response variable. This assumption is often useful for data-driven discovery in biological settings, particularly for framing the interpretation of the model in the context of physical processes that generated the data. The identification of which β_i are non-zero can be viewed as a feature selection (or more generally, model selection) problem [81]. A common regularization penalty used for feature selection is the lasso penalty $|\boldsymbol{\beta}|_1$, or the ℓ_1-norm applied to the parameters [196]. For the case of linear regression, this creates an optimization problem of the form

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} (y_i - \boldsymbol{\beta}^T \mathbf{x}_i)^2 + \lambda |\boldsymbol{\beta}|_1. \tag{4.3}$$

Solving Eq (4.3) returns parameter estimates with some sparsity, provided that λ is appropriately chosen (Fig 4.2a, top). Typically, λ, the degree to which feature sparsity is enforced, is unknown and must be determined through cross-validation or a penalized score function such as the Bayesian information criterion (BIC) [170] across a set of J hyperparameters $\{\lambda_j\}_{j=1}^J$. Importantly, solving the lasso problem simultaneously performs model selection (identifying the non-zero features) and model estimation (determining the specific values of those parameters). However, the application of the lasso penalty suffers from shrinkage [196], or a parameter bias that erroneously reduces the magnitudes of the parameters (Fig 4.2a, top: compare opacity of parameter estimates), and often does not correctly identify the true non-zero parameters (Fig 4.2a, top: false positives).

On the other hand, ensemble procedures (e.g., bagging and boosting [36, 69]) aggregate model fits across resamples of the data to improve the stability of parameter estimates (Fig 4.2a, bottom). The more stable parameter estimates result in improved predictive accuracy. This is particularly desirable in biological settings, where model aggregation ensures that the relevant signal in noisy data is reflected in the parameter estimates. However, ensemble procedures do not perform feature selection.

UoI separates model selection and model estimation into two stages, with each stage utilizing ensemble procedures to promote stability. Specifically, model selection is performed through intersection (compressive) operations and model estimation through union (expansive) operations, in that order. This separation of parameter selection and estimation provides selection profiles that are robust and parameter estimates that have low bias and variance. Fig 4.2b and 4.2c provide a visual depiction of the UoI framework, and 4.5 provides pseudocode for the UoI algorithm in generalized linear models. For $\text{UoI}_{\text{Lasso}}$, the procedure is as follows:

Model Selection. Define the support S as the set of non-zero parameters in an estimate $\hat{\boldsymbol{\beta}}$. First, generate a regularization path of $\{\lambda_j\}_{j=1}^J$ spanning $(\epsilon\lambda_{\max}, \lambda_{\max})$ where $\lambda_{\max}$ is analytically determined to result in an empty support and $\epsilon = 10^{-3}$ [68]. For each λ_j, generate parameter estimates by solving the lasso optimization problem (Eq 4.3) on N_S resamples of the data, and calculate a support for each resample-λ_j pairing. The intersection step requires that only the features that appear in a sufficient number of resamples are

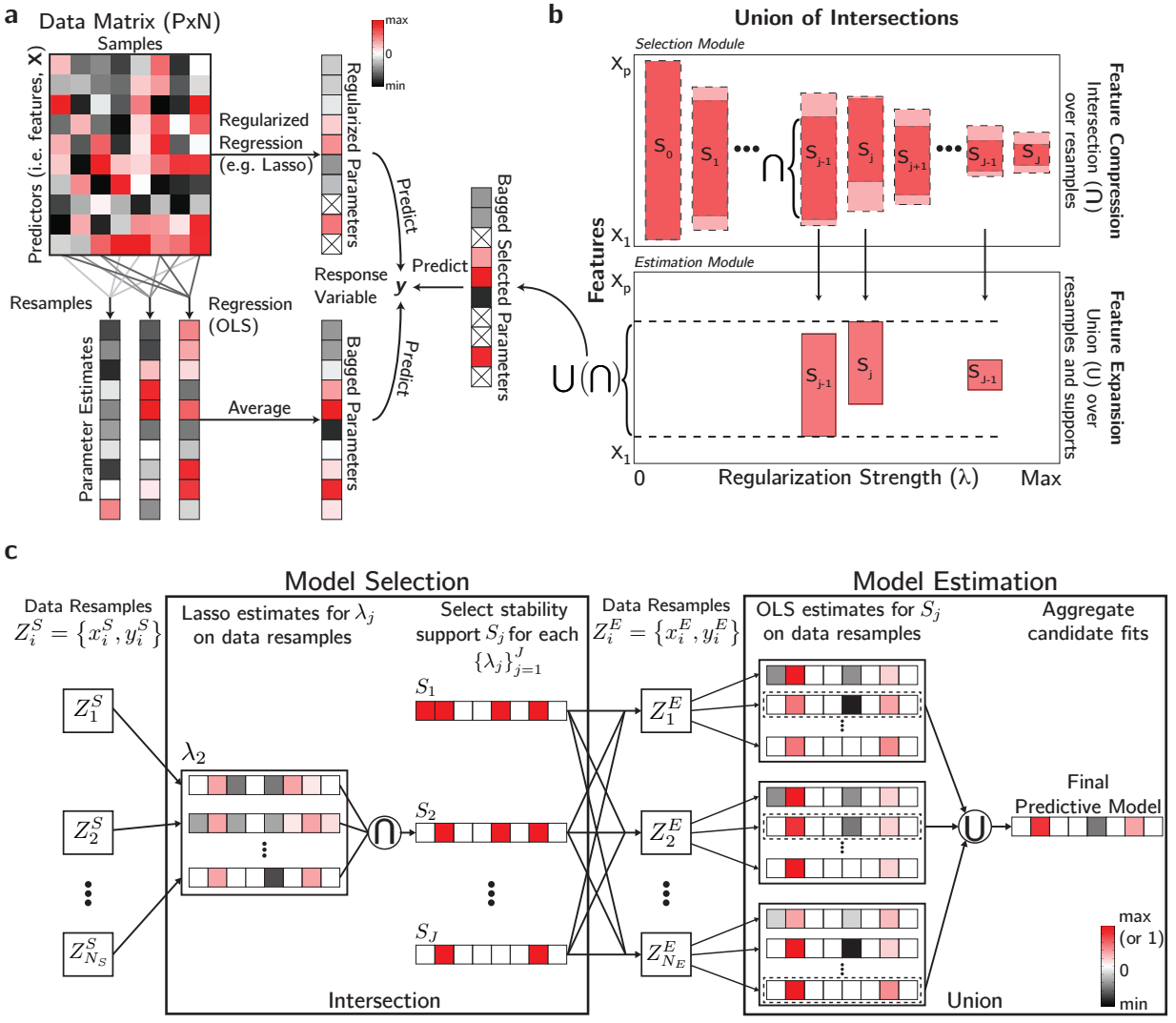


Figure 4.2: **The Union of Intersections framework combines ensemble and regularization approaches in model inference. a.** Schematic of regularization and ensemble methods. Top: Regularization can be used to perform feature selection. Features set exactly to zero are denoted by $\times$. Bottom: Ensemble procedures aggregate model fits across resamples of the data. **b.** Schematic of the UoI framework. The x-axis corresponds to the set of regularization parameters while the y-axis corresponds to the feature space $(X_1, X_2, \ldots, X_p)$. Pink bands denote features included in a support or estimated model. Dark pink bands denote features included after the intersection step. **c.** $\text{UoI}_{\text{Lasso}}$ depicted in a data-distributed fashion. Caption continued on following page.

included in the final *stability support* S_j for λ_j. We depict this in Fig 4.2b, top, where the light pink bands denote features that are included in the model due to regularization while

Figure 4.2: **Continued from previous page.** *Model Selection (left):* Left column depicts the N_S data resamples, Z_i^S, during selection. Lasso fits are obtained for each resample, at different λ_j (e.g., left column: λ_2). Right column depicts the intersected supports S_j for each λ_j, with S_2 referring specifically to the estimates in the left column. Red here denotes 1, rather than maximum value. *Model Estimation (right):* Middle column denotes each data resample, Z_i^E, for estimation. Each resample is fit using each support $\{S_j\}_{j=1}^J$ (arrows before left column), generating a set of fits per resample (left column). The fit that achieves the best predictive performance for a given resample is chosen as that resample's candidate fit (left column: dashed lines). The candidate fits are aggregated to produce the final predictive model (right column).

the dark pink bands denote features that are included in the stability support after the intersection across resamples. The bands are arranged in order of increasing regularization strength (Fig 4.2b: x-axis) and thus sparser (i.e., smaller) support sets (Fig 4.2b: y-axis). Note that the stability support may be calculated with a hard intersection (e.g., Fig 4.2c: Model Selection) or a soft intersection. In the former case, a feature must appear in the support of every resample to be included in S_j. In the latter, the feature must only appear in a sufficient fraction of supports which is a hyperparameter.

Model Estimation. Generate N_E resamples of the data, and perform an unregularized fit on each resample using each support set S_j. For each resample, the support generating the fit that performs the best according to some metric is chosen as the "candidate fit" for that resample (Fig 2b: bottom). We used the Bayesian information criterion (BIC) as this metric, which balances both predictive accuracy and model size (see Section 2.4.1 for more details on this choice). Unique supports may have the best performance across multiple resamples (e.g., only three unique supports, S_{j-1}, S_j, and S_{m-1}, are included for model averaging in Fig 4.2b). The N_E candidate fits across resamples are unionized according to some metric (e.g., median, mean, etc.), resulting in a final parameter estimate (Fig 4.2b, bottom). We use the median during the union step because it is more stable than the mean from a selection perspective. To be clear, the median will result in a parameter estimate set exactly equal to zero if that parameter is equal to zero in at least a majority of candidate fits. In contrast, the mean will likely result in a non-zero parameter estimate if even a single parameter value is non-zero. Note that, in the context of (generalized) linear models, the bagging of model parameters performed in the estimation procedure is equivalent to the bagging of model predictions. For $\text{UoI}_{\text{Lasso}}$, the estimation procedure consists of applying Ordinary Least Squares to each stability support and resample combination (Fig 4.2c: Model Estimation).

UoI's modular approach to parameter inference capitalizes on the feature selection achieved by stability selection and the unbiased, low-variance properties of the bagged OLS estimator. Furthermore, UoI's novel use of model aggregating procedures within its resampling framework allows it to achieve highly sparse (i.e., only using features robust to perturbations in

the data) and predictive (i.e., only using features that are informative) model fitting. Importantly, this is achieved without imposing an explicit prior on the model distribution, and without formulating a non-convex optimization problem. Since the optimization procedures across resamples can be performed in parallel, the UoI framework is naturally scalable, a fact that we have leveraged to facilitate parameter inference on larger datasets [163]. The application of $\text{UoI}_{\text{Lasso}}$ in a data-distributed manner is depicted in Fig 4.2c. In the selection module, the first column depicts lasso estimates across data resamples for a particular choice of regularization parameter, all of which can be fit in parallel (Fig 4.2c, Model Selection: left column). In the estimation module, OLS estimates are fit across resamples and supports, which can be done in parallel (Fig 4.2c, Model Estimation: left column).

Evaluation of Union of Intersection on synthetic data

We evaluated $\text{UoI}_{\text{Lasso}}$'s abilities as an inference procedure by assessing its performance on synthetic data generated from a linear model. The performances of UoI and five other inference procedures are depicted in Fig 4.3: $\text{UoI}_{\text{Lasso}}$ (black), ridge regression (purple) [81], lasso (green) [196], smoothly clipped absolute deviation (SCAD; red) [65], bootstrapped adaptive threshold selection (BoATS; blue) [32], and debiased lasso (dbLasso; coral) [91].

The linear model consisted of $p = 300$ total parameters, with $k = 100$ non-zero parameters (thereby having sparsity $1 - k/p = 2/3$). The non-zero ground truth parameters were drawn from a parameter distribution characterized by exponentially increasing density as a function of parameter magnitude (Fig 4.3b: gray histograms). We used $N = 1200$ samples generated according to the linear model (4.1) with noise magnitude chosen such that $\text{Var}(\epsilon) = 0.2 \times |\boldsymbol{\beta}|_1$. We report metrics according to their statistics across 100 randomized cross-validation samples of the data.

In Fig 4.3a, we show scatter plots comparing the predicted and actual values of the observation variable on held-out data samples. We visualized how well the inference procedures captured the underlying parameter distribution by comparing the histograms of (average) estimated model parameters (colors) overlaid on the ground truth model parameters (grey) (Fig 4.3b). We additionally plotted parameter bias and variance, first by comparing the mean estimated value ($\pm$ standard deviation) against the ground truth parameter value (Fig 4.3c), and then examining the standard deviation of the parameter estimates as a function of their mean estimated value (Fig 4.3d).

Fig 4.3a-d captures the improvements that the UoI framework offers in parameter inference. $\text{UoI}_{\text{Lasso}}$ is designed to maximize prediction accuracy (Fig 4.3a) by first selecting the correct features (Fig 4.3b), and then estimating their values with high accuracy (Fig 4.3c) and low variance (Fig 4.3d). By separating model selection and model estimation, $\text{UoI}_{\text{Lasso}}$ benefits from strong selection (as in BoATS and debiased Lasso), but with the low variability of the structured regularizers (Lasso, SCAD), while alleviating shrinkage with its nearly unbiased estimates.

We quantified the performance of the inference algorithms on synthetic data using a variety of metrics capturing selection, bias, variance, and prediction accuracy. Specifically,

these metrics were:

- **Selection Accuracy.** The selection accuracy, or set overlap, is a measure of how well the estimated support captures the ground truth support. Define S_β as the set of features in the ground truth support, $S_{\hat{\beta}}$ as the set of features in the estimated model's support, $|S|$ as the cardinality of S, and Δ as the symmetric set difference operator. Then the selection accuracy is defined as

$$\text{selection accuracy}\left(S_\beta, S_{\hat{\beta}}\right) = 1 - \frac{|S_\beta \Delta S_{\hat{\beta}}|}{|S_\beta| + |S_{\hat{\beta}}|}. \tag{4.4}$$

  The selection accuracy is bounded in $[0, 1]$, taking value 0 if $S_{\hat{\beta}}$ and S_β have no elements in common, and taking value 1 iff they are identical.

- **Estimation error.** The estimation error of the p fitted parameters $\hat{\beta}$, with ground truth parameters β, is defined as the root mean square error, or

$$\text{estimation error} = \sqrt{\frac{1}{p}\sum_{i=1}^{p}(\beta_i - \hat{\beta}_i)^2}. \tag{4.5}$$

- **Estimation variability.** The estimation variability for parameter β_i is defined as the parameter standard deviation $\sigma(\beta_i)$. We calculated this quantity by taking the variance of the estimated parameter $\hat{\beta}_i$ over R resamples of the data:

$$\sigma(\beta_i) = \sqrt{\frac{1}{R}\sum_{j=1}^{R}(\beta_i - \hat{\beta}_{ij})^2}, \tag{4.6}$$

  where j indexes the resample. To summarize this measure across all p parameters in a model, we took the average, i.e., $\sigma = \frac{1}{p}\sum_{i=1}^{p}\sigma(\beta_i)$.

- **Predictive performance.** To capture predictive performance, we used the coefficient of determination (R^2) evaluated on held-out data:

$$R^2 = 1 - \frac{\sum_{i=1}^{D}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{D}(y_i - \overline{y})^2} \tag{4.7}$$

  where y_i is the ground truth response for sample i, $\hat{y}_i$ its corresponding predicted value, and $\overline{y}$ the mean of the response variable over trials. R^2 has a maximum value of 1, when the model perfectly predicts the response variable across samples. R^2 values below zero indicate that the model is worse than an intercept model (i.e., simply using the mean value to predict across samples).

- **Model Parsimony.** We evaluated model parsimony using the Bayesian information criterion (BIC) [170]:

$$\mathrm{BIC} = k \log(D) - 2 \log \ell(\hat{\beta}). \tag{4.8}$$

  Here, D is the number of samples, k is the number of parameters estimated by the model, and $\log \ell(\hat{\beta}) = p(\hat{\beta}|\mathcal{D}, m)$ is the log-likelihood of the parameters $\hat{\beta}$ under data $\mathcal{D}$ and model m. Thus, the BIC includes a penalty that encourages models to be more sparse (first addend) while still accounting for predictive accuracy (second addend). Importantly, the BIC is evaluated on the data that the model was trained on (rather than held-out data). It is typically used as a model selection criterion (in lieu of, for example, cross-validation). When used as a model selection criterion, the model with lower BIC is preferred.

$\mathrm{UoI}_{\mathrm{Lasso}}$ generally resulted in the highest selection accuracy (Fig 3e, first column), parameter estimates with lowest error (Fig 3e, second column) and competitive variance (Fig 3e, third column). In addition, it led to the best prediction accuracy (Fig 3e, third column). $\mathrm{UoI}_{\mathrm{Lasso}}$ best captured the true model size (Fig 3e, fourth column), avoiding the abundance of false positives suffered by most other inference algorithms. $\mathrm{UoI}_{\mathrm{Lasso}}$'s enhanced predictive performance with fewer features resulted in superior model parsimony (Fig 3e, fifth column).

A robust set of experiments have been conducted comparing $\mathrm{UoI}_{\mathrm{Lasso}}$ to these methods in other settings. Specifically, these experiments assessed $\mathrm{UoI}_{\mathrm{Lasso}}$'s performance across a range of ground truth model sparsities, parameter distributions, and noise levels. Overall, UoI excels at parameter inference relative to other models across all these settings. These results can be found in the appendix of the original UoI paper [31].

Figure 4.3 demonstrates the superiority of UoI methods over a battery of other approaches for a linear model. However, any generalized linear model fits within the UoI framework. For example, we used the Poisson ($\mathrm{UoI}_{\mathrm{Poisson}}$) and logistic ($\mathrm{UoI}_{\mathrm{Logistic}}$) variants of the UoI algorithm in this study. Thus, we compared the performance of UoI to baseline procedures for Poisson and logistic regression. Specifically, we created similar synthetic datasets as described above, but in the Poisson and logistic contexts, and attempted to estimate the ground truth parameters using UoI and lasso-penalized Poisson and logistic regression. These results are detailed in 4.5. We arrived at similar conclusions: UoI exhibits increased selection accuracy, decreased estimation error, comparable variability, improved prediction accuracy, and enhanced model parsimony relative to the baseline procedures. These results are in line with theoretical guarantees on support recovery in generalized linear models [202, 40].

Neural recordings

We sought to demonstrate impact of improved inference on parametric models across a diversity of datasets, spanning distinct brain regions, animal models, and recording modalities. We used micro-electrocorticography recordings obtained from rat auditory cortex (for

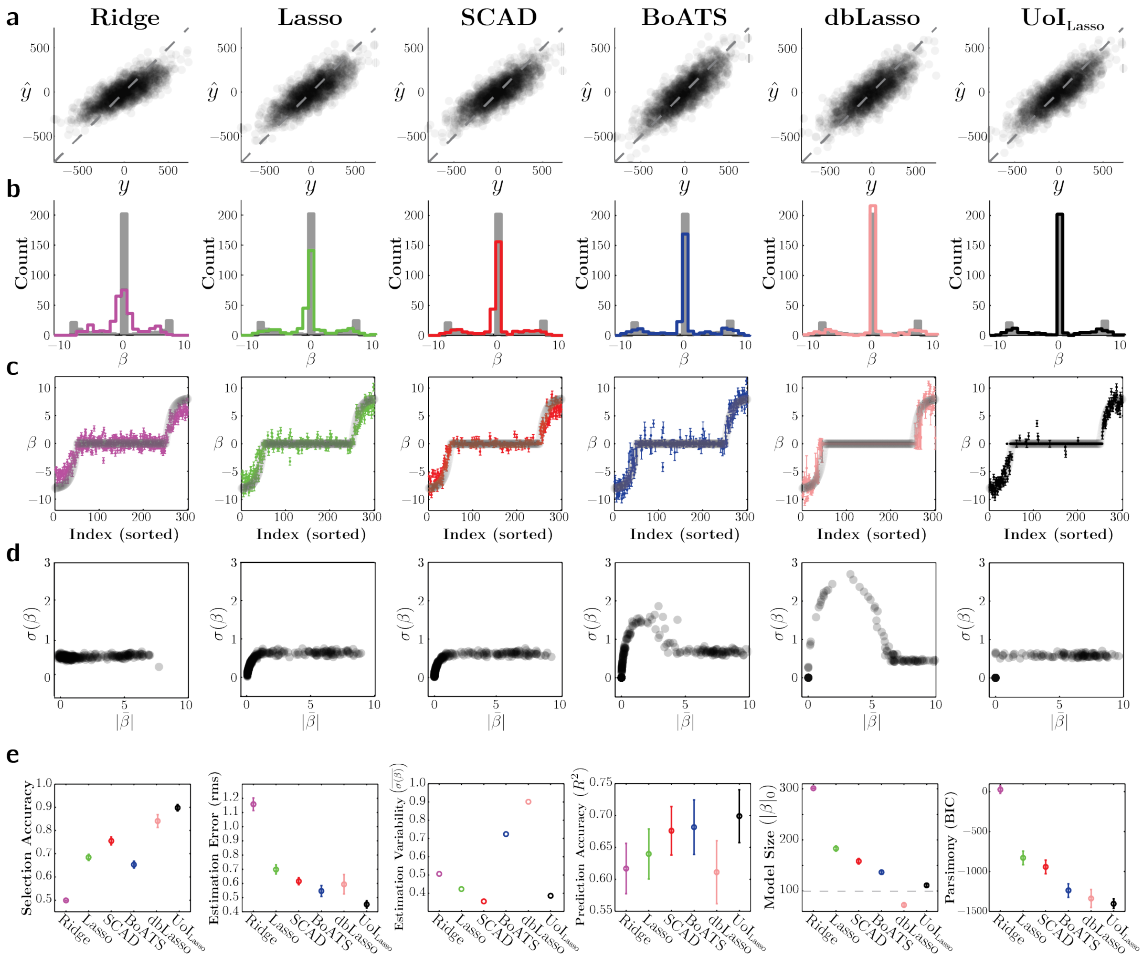


Figure 4.3: **UoI achieves superior selection and estimation performance on synthetic data over a battery of alternative inference algorithms.** The performance of ridge regression (purple), lasso regression (green), smoothly clipped absolute deviation (SCAD; red), bootstrapped adaptive threshold (BoATS; blue), debiased lasso (dbLasso; coral), and UoI$_{\text{Lasso}}$ (black) on data generated from a synthetic linear model. Panels **a-d** highlight separate measures per row, with each column referring to a separate inference algorithm. Each column in panel **e.** directly compares a summary measure across inference algorithms. **a.** Comparison of the predicted and true values of the response variable for held-out data. **b.** Histogram of the estimated parameters (colored outline) compared to the distribution of true parameters (gray). **c.** Estimated parameters (colored points; error bars denote the IQR across data resamples) compared to true parameters (gray), both sorted on the x-axis according to the value of the true parameter. **d.** Variance of the parameter estimates across data resamples as a function of the mean estimated parameter's magnitude. Caption continued on the following page.

Figure 4.3: **Continued from previous page. e.** Direct comparison of the selection accuracy, estimation error (root mean square of parameter estimates), estimation variability (mean parameter variance), prediction accuracy (coefficient of determination), the inferred model's size (number of identified non-zero parameters, with the black dashed line denoting ground truth), and parsimony (Bayesian information criterion). Error bars denote IQR across data resamples. Caption continued on following page.

coupling and encoding models), single-unit recordings from macaque visual and motor cortices (coupling models), single-unit recordings from isolated rat retina (encoding models), and single-unit recordings from basal ganglia (decoding models). We briefly describe the experimental and preprocessing steps for each dataset.

Recordings from auditory cortex

Auditory cortex (AC) data was comprised of cortical surface electrical potentials (CSEPs) recorded from rat auditory cortex with a custom fabricated micro-electrocorticography (μECoG) array. The μECoG array consisted of an 8×16 grid of 40 μm diameter electrodes. Anesthetized rats were presented with 50 ms tone pips of varying amplitude (8 different levels of attenuation, from 0 dB to -70 db) and frequency (30 frequencies equally spaced on a log-scale from 500 Hz to 32 kHz). Each frequency-amplitude combination was presented 20 times, for a total of 4200 samples. The response for each trial was calculated as the z-scored, to baseline, high-γ band analytic amplitude of the CSEP, calculated using a constant-Q wavelet transform. Of the 128 electrodes, we used 125, excluding 3 due to faulty channels. Data was recorded by Dougherty & Bouchard (DB). Further details on the surgical, experimental, and preprocessing steps can be found in [61].

Recordings from primary visual cortex

We analyzed three primary visual cortex (V1) datasets, comprised of spike-sorted units simultaneously recorded in three anesthetized macaque monkeys. Recordings were obtained with a 10×10 grid of silicon microelectrodes spaced 400 μm apart and covering an area of 12.96 mm^2. Monkeys were presented with grayscale sinusoidal drifting gratings, each for 1.28 s. Twelve unique drifting angles (spanning 0° to 330°) were each presented 200 times, for a total of 2400 trials per monkey. Spike counts were obtained in a 400 ms bin after stimulus onset. We obtained $[106, 88, 112]$ units from each monkey. The data was obtained from the Collaborative Research in Computational Neuroscience (CRCNS) data sharing website [192] and was recorded by Kohn and Smith (KS) [104]. Further details on the surgical, experimental, and preprocessing steps can be found in [180] and [102].

Recordings from primary motor cortex

Primary motor cortex (M1) data was comprised of spike-sorted units simultaneously recorded in the motor cortex of Rhesus macaque monkey. Recordings were obtained with a chronically implanted silicon microelectrode array consisting of 96 electrodes spaced at 400 μm and covering an area of 16 mm^2. We used three datasets, consisting of three recording sessions from monkey I. The behavioral task required the monkey to make self-paced reaches to targets arranged on a 8×17 grid. Spike counts were binned at 150 ms over the course of the entire recording session, resulting in $[4089, 4767, 4400]$ samples per recording session. We obtained $[136, 146, 147]$ units from each dataset. Data was recorded by O'Doherty et al. (OCMS) and obtained from Zenodo [138]. Further details on the surgical, experimental, and preprocessing steps can be found in [122].

Recordings from retina

Retina data comprised spiking activity, extracellularly recorded from isolated mice retina. Recordings were obtained using a 61-electrode array. Isolated retina were presented with a flicking black or white bar stimulus according to a pseudo-random binary sequence for a period of 16.6 ms. We utilized recordings from 23 different retinal ganglion cells. Data was obtained from CRCNS and recorded by Zhang et al [221]. Further details on the surgical, experimental, and preprocessing steps can be found in [111].

Recordings from basal ganglia

Basal ganglia data comprised tetrode recordings from two regions of rat basal ganglia: the globus pallidus pars externa (GPe: 18 units) and substantia pars nigra reticulata (SNr: 36 units). Recordings were performed during a rodent stop-signal task. Briefly, a rat was prompted to enter a center port with a light cue. The rat remained in the port until a Go cue (audio stimulus at 1 kHz or 4 kHz) which directed a lateral head movement to the left or right ports. On a subset of trials, the Go cue was followed by a Stop signal (white noise burst), indicating that the rat should remain in the center port. We utilized the successful Go trials, in which the rat was not given a Stop signal and successfully entered the correct port (186 trials). We used the spike count in the first 100 ms after the rat exited the center port to predict the behavioral condition (left or right). Further details on the surgical, experimental, and preprocessing steps can be found in [79].

Neural data analysis and model fitting

All models fit to neural data consisted of various generalized linear models, depending on the application. We trained all baseline models using either the `glmnet` [67] or `scikit-learn` [150, 41] packages. Meanwhile, we trained all UoI models using the `pyuoi` package [163]. This section is organized as follows: first, we discuss model fitting, including details on

the estimation module in UoI, followed by data analysis for coupling, encoding, and decoding models. Second, we detail model evaluation, including the measures used to assess each model, statistical tests, calculation of effect size, and the cross-validation approach for evaluation.

Model selection criterion in the estimation module

In the UoI framework, the estimation module operates by unionizing fitted stability supports across resamples. Thus, the module requires a criterion by which to choose the best fitted stability support per resample. A natural choice, akin to cross-validation, is the out-of-resample validation performance according to some measure (e.g., R^2, deviance, etc.). This is a principled approach when predictive accuracy is the only usage of the model. However, in this context, where we frame the fitted parameters of the model in the underlying neuroscience, parameter selection implicitly becomes an additional goal that must be reflected in the model criterion. Cross-validated predictive accuracy is often not sufficient in these cases. Variations of cross-validation have been shown to be model inconsistent [177, 176], with theoretical guarantees on its probability of overfitting [222]. Furthermore, empirically, it has been shown to overfit, and suffer from false positives [209, 117]. Therefore, we instead utilized the Bayesian information criterion in the estimation module for each model, which has been shown to be model selection consistent [177]. Furthermore, the BIC is a principled choice for model selection criterion because it can be couched as an approximation to Bayes factors [133].

Data analysis for coupling models

We used $\text{UoI}_{\text{Lasso}}$ (rat auditory cortex) and $\text{UoI}_{\text{Poisson}}$ (macaque V1 and M1) to fit coupling models. The auditory cortex model can be described with a linear model as

$$n_i = \beta_{i0} + \sum_{\substack{j=1 \\ j \neq i}}^{p} \beta_{ij} n_j + \epsilon \tag{4.9}$$

where n_i is the high-gamma activity of the ith electrode on a trial. The baseline procedure consisted of a lasso optimization with coordinate descent, while the UoI approach utilized $\text{UoI}_{\text{Lasso}}$. The model for the spiking datasets, which utilizes a Poisson generalized linear model, can be written as

$$\mu_i = \exp\left(\beta_{i0} + \sum_{\substack{j=1 \\ j \neq i}}^{p} \beta_{ij} n_j\right), \tag{4.10}$$

$$n_i \sim \text{Poisson}(\mu_i). \tag{4.11}$$

where n_i corresponds to the spike count of the ith neuron. The corresponding objective function for this model is the log-likelihood,

$$\mathcal{L}_i\left(\boldsymbol{\beta}|\left\{n_1^k,\ldots,n_p^k\right\}_{k=1}^D\right)=\sum_{k=1}^{D}\left[n_i^k\left(\beta_{0i}+\sum_{\substack{j=1\\j\neq i}}^{p}n_j^k\beta_{ij}\right)-\exp\left(\beta_{0i}+\sum_{\substack{j=1\\j\neq i}}^{p}n_j^k\beta_{ij}\right)\right] \tag{4.12}$$

where i denotes that this model corresponds to the ith neuron, j indexes over the remaining neurons, and k indexes over the D data samples. The baseline approach consisted of applying coordinate descent to solve this objective function with a lasso penalty:

$$\mathcal{L}_{i,\text{baseline}}\left(\boldsymbol{\beta}|\left\{n_1^k,\ldots,n_p^k\right\}_{k=1}^D\right)=\frac{1}{D}\mathcal{L}_i+\lambda_1|\boldsymbol{\beta}|_1 \tag{4.13}$$

where λ_1 is a hyperparameter specifying the strength of the ℓ_1 penalty. Note that the intercept terms were not penalized. Meanwhile, in $\text{UoI}_{\text{Poisson}}$, we utilized the same objective function Eq (4.13) in the selection module. In the estimation module, we used Eq (4.12) with a very small ℓ_2 penalty for numerical stability purposes. The specific optimization algorithm was a modified orthant-wise L-BFGS solver [76].

Data analysis for encoding models

For retinal data, we fit spatio-temporal receptive fields (STRFs) frame-by-frame. Specifically, the STRF was comprised of F frames $\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\ldots,\boldsymbol{\beta}_F$, each a vector of size M and spanning Δt seconds. For neuron i and frame k, the encoding model consisted of

$$n_i(t)=\beta_0+\boldsymbol{\beta}_k^T\mathbf{e}(t-k\Delta t) \tag{4.14}$$

where $n_i(t)$ is the spike count at timepoint t and $\mathbf{e}(t-k\Delta t)$ is flicking bar stimulus value at k bins before t. We fit the F models using lasso (baseline) and $\text{UoI}_{\text{Lasso}}$, and created the final STRF by concatenating the parameter values $\boldsymbol{\beta}=[\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\ldots,\boldsymbol{\beta}_F]$.

The tuning model for the rat auditory recordings was constructed using Gaussian basis functions. We used eight Gaussian basis functions spanning the log-frequency axis with means $\{\mu_j\}_{j=1}^8$ [185, 188]. Thus, the high-gamma activity n_i of electrode i in response to frequency f was

$$n_i(f)=\beta_{i0}+\sum_{j=1}^{8}\beta_{ij}\exp\left(-\frac{\log f-\mu_j}{2\sigma^2}\right) \tag{4.15}$$

We chose $\sigma^2=0.64$ octaves so that basis functions sufficiently spanned the plane. We chose $p=8$ basis functions because this was the minimum number of basis functions for which every electrode had a selection ratio less than 1. We fit Eq (4.15) using cross-validated lasso as the baseline. To characterize the relationship between selection ratio and predictive

performance of the rat AC tuning models, we fit trendlines across models using Gaussian process regression. Specifically, we utilized a regressor with radial basis function kernel (length scale $\ell = 0.01$) and a white noise kernel (noise level $\alpha = 0.1$).

Lastly, we fit encoding models for the macaque V1 and M1 datasets using cosine basis functions as a function of grating and reach angle, respectively [188]. Importantly, we modeled the spike count of the ith neuron after a variance stabilizing square-root transform, which is typically used when a Gaussian model is applied to data exhibiting Poisson variability [219]. For the M1 dataset, the encoding model can be written as

$$r_i = \beta_{i0} + \beta_{i1}\cos(\theta) + \beta_{i2}\sin(\theta) + \epsilon \tag{4.16}$$

where $r_i = \sqrt{n_i}$ is the square-rooted spike count of the ith neuron, and θ is the angle of the reach [186]. The encoding model for the V1 neurons was similar, aside from an adjustment in period:

$$r_i = \beta_{i0} + \beta_{i1}\cos(2\theta) + \beta_{i2}\sin(2\theta) + \epsilon, \tag{4.17}$$

where θ is the angle of the grating. We adjusted the period of the basis functions because the single-units were responsive to gratings drifting in either direction along an axis. Since these models only use two basis functions, we fit them using ordinary least squares (OLS) with no regularization. We did not perform a comparison between the baseline model and a UoI-fitted model, since the OLS estimator is a consistent estimator. These encoding models were only used when examining the relationship between network and tuning structure.

Data analysis for decoding models

We fit decoding models to basal ganglia recordings as binary logistic regression models. The model expresses the probability of one experimental condition e (e.g., the rat entering the left port) as

$$\Pr[e = \text{left}] = \text{sigmoid}\left(\beta_0 + \sum_{j=1}^{p} \beta_j n_j\right), \tag{4.18}$$

where n_i is the neural activity of the ith neuron. The corresponding objective function is the log-likelihood, or

$$\mathcal{L}_i\left(\boldsymbol{\beta} | \left\{n_1^k, \ldots, n_p^k\right\}_{k=1}^{D}\right) = \\ \sum_{k=1}^{D}\left[\log\left(1 + \exp(\beta_0 + \sum_{j=1}^{p}\beta_j n_j^k)\right) - e^k(\beta_0 + \sum_{j=1}^{p}\beta_j n_j^k)\right]. \tag{4.19}$$

The baseline approach consisted of solving this objective function with an ℓ_1 penalty. $\text{UoI}_{\text{Logistic}}$ utilized objective function (4.19) with an ℓ_1 penalty in the selection module, and Eq (4.19) alone in the estimation module.

Network creation and analysis

We performed secondary analyses on the coupling models by constructing graphs using the fitted parameters. We then analyzed these networks with standard graph theoretic measures. Here, we detail how we constructed the networks and the measures we used to analyze them.

We created directed graphs by filling the adjacency matrix A_{ij} with the coefficient β_{ij} (i.e., the coupling coefficient for neuron j in the coupling model for neuron i). Meanwhile, we created undirected networks from coupling models by symmetrizing coefficients [208]. Specifically, the symmetric adjacency matrix satisfies $A_{ij} = \frac{1}{2}(\beta_{ij} + \beta_{ji})$ where β_{ij} is the coefficient specifying neuron i's dependence on neuron j's activity, and vice versa for β_{ji}. Thus, the network lacked an edge between vertices (neurons) i and j if only if neuron i's coupling model did not depend on neuron j, and neuron j's coupling model did not depend on neuron i. This adjacency matrix is weighted in that each entry depends on the magnitudes of the coupling coefficients. However, we can also consider an unweighted, undirected graph, whose adjacency matrix is simply the binarization of A_{ij}.

We analyzed the networks with the following measures:

- **In- and out-degree.** In a directed graph, the in-degree of a vertex is the number of incoming edges to that vertex. Meanwhile, the out-degree is the number of outgoing edges from the vertex. We examine the distribution of in-degrees and out-degrees across vertices in each group, which is dependent on the sparsity of the coupling models.

- **Modularity.** The modularity Q is a scalar value that measures the degree to which a network is divided into communities [135]. We operate on the undirected graph described by the binarized adjacency matrix, described above. Suppose each vertex v is partitioned into one of c communities, where vertices within a community are more likely to be connected with each other than vertices between communities. Then, the modularity is defined as

$$Q = \frac{1}{2m} \sum_{v,w} \left[A_{vw} - \frac{k_v k_w}{2m} \right] \delta(c_v, c_w) \tag{4.20}$$

  where c_v denotes community identity, k_v is the degree of vertex v, and m is the total number of edges. Thus, the modularity is greater than zero when there exist more edges between vertices within the same community than might be expected by chance according to the degree distribution. Specifically, Q is bounded within the range $[-1/2, 1]$, where $Q > 0$ indicates the existence of community structure. We calculated modularity with the Clauset-Newman-Moore greedy modularity maximization algorithm [49]. This procedure assigns vertices to communities by greedily maximizing the modularity, and then calculating Q using the ensuing community identities.

- **Small-worldness.** Small-world networks are characterized by a high degree of clustering with a small characteristic path length [210, 26, 24]. There are multiple measures

used to quantify the degree to which a network is small-world. We use ω, which can be expressed as

$$\omega = \frac{L_r}{L} - \frac{C}{C_\ell}, \tag{4.21}$$

where L is the characteristic path length of the network, L_r is the characteristic path length for an equivalent random network, C is the clustering coefficient, and C_ℓ is the clustering coefficient of an equivalent lattice network [193]. The quantity ω is bounded within $[-1, 1]$, where ω close to 0 indicates that the graph is small-world. When ω is close to 1, the graph is closer to a random graph, while ω close to -1 implies the graph is more similar to a lattice graph.

Model evaluation

We used the following measures to evaluate the models fit to neuroscience data:

- **Selection Ratio.** We evaluate the sparsity of estimated models with the selection ratio, or the fraction of parameters fitted to be non-zero:

$$\text{selection ratio} = \frac{k}{p}, \tag{4.22}$$

where p is the total number of parameters available to the model and k is the number parameters that a model-fitting procedure explicitly sets non-zero.

- **Predictive performance.** We utilized several measures of predictive performance, depending on the model. For linear models (i.e., a generalized linear model with an identity link function), we used the coefficient of determination (R^2) evaluated on held-out data, as detailed in Section 2.2 Recall that R^2 values below zero indicate that the model is worse than an intercept model (i.e., simply using the mean value to predict across samples).

  For Poisson regression, or a generalized linear model with a logarithmic link function, we utilized the deviance, which is the difference in log-likelihood between the saturated model and the estimated model [134]. The saturated model has parameters specifically chosen to reproduce the observed values. For the Poisson log-likelihood, the expression for the deviance as a function of the estimation parameters $\hat{\beta}$ is given by

$$\text{deviance}(\hat{\beta}) = \left[\sum_{i=1}^{D} y_i \log(y_i) - y_i\right] - \left[\sum_{i=1}^{D} y_i(\hat{\beta}_0 + \hat{\beta}^T x_i) - \exp\left(\hat{\beta}_0 + \hat{\beta}^T x_i\right)\right], \tag{4.23}$$

where $\{x_i, y_i\}_{i=1}^{D}$ denote the features and response variable of the model, respectively. Note that lower deviance is preferred, in contrast to the coefficient of determination. For logistic decoding models, we used the classification accuracy on held-out data as the measure of predictive performance.

- **Model Parsimony.** As detailed in Section 4.2, we evaluated model parsimony using the Bayesian information criterion (BIC). Recall that the BIC includes a penalty that encourages models to be more sparse while still accounting for predictive accuracy. Importantly, the BIC is evaluated on the data that the model was trained on (rather than held-out data). It is typically used as a model selection criterion (in lieu of, for example, cross-validation). When used as a model selection criterion, the model with lower BIC is preferred.

- **Effect Size.** To fully capture the difference in model evaluation metrics beyond statistical significance, we measure effect size using Cohen's d [50]. For two groups of data with sample sizes D_1, D_2, means μ_1, μ_2, and standard deviations s_1, s_2, Cohen's d is given by

$$d = \left| \frac{\mu_1 - \mu_2}{s} \right| \tag{4.24}$$

  where s is the pooled standard deviation:

$$s = \sqrt{\frac{(D_1 - 1)s_1^2 + (D_2 - 1)s_2^2}{D_1 + D_2 - 2}}. \tag{4.25}$$

  We often considered cases where $D_1 = D_2$, implying that $s = \sqrt{\frac{s_1^2 + s_2^2}{2}}$. Values of d on the order of 0.01 indicate very small effect sizes, while $d > 1$ indicates a very large effect size [167].

- **Statistical Tests.** We used the Wilcoxon signed-rank test [212] to assess whether the distributions of selection ratios and predictive performances, across units, were significantly different between the UoI models and the baseline models. Importantly, we did not apply the test to the distribution of BICs, since differences in BIC are better interpreted as approximations to Bayes factors [132]. To assess whether distributions of UoI and baseline model parameters were significantly different, we used the Kolmogorov-Smirnov test. We applied a significance level of $\alpha = 0.01$ for all statistical tests.

Cross-validation, model training, and model testing

Each dataset was split into 10 folds after shuffling across samples (except for the basal ganglia data, which was split into 5 folds due to fewer samples). When appropriate, the folds were stratified to contain equal proportions of samples across experimental setting (e.g., stimulus value or behavioral condition). In each task, we fit 10 models (or five, for basal ganglia) by training each on 9 (4) folds, and using the last fold as a test set. Hyperparameter selection for baseline procedures was performed via cross-validation within the training set of 9 (4) folds. Meanwhile, all resampling for the UoI procedures was also performed within the

training set. Model evaluation statistics (selection ratio, predictive performance, Bayesian information criterion) are reported as the median across the 10 (5) models. Any measures that operate on the fitted models (e.g., coefficient value, network formation, modularity, etc.) were calculated by using the model that is formed by taking the median parameter value across folds.

4.3 Results

Parametric models are ubiquitous data analysis tools in systems neuroscience. However, their usefulness in understanding a neural system hinges on the assumption that their parameters are accurately selected and estimated. By accurate selection, we mean low false positives and false negatives in setting parameters equal to zero; by accurate estimation, we mean low-bias and low-variance in the parameter estimates. The potential neuroscientific consequences of improper selection or estimation during inference are generally not well understood. Thus, we studied selection and estimation in common systems neuroscience models by comparing the properties of models inferred by standard methods to those inferred by the Union of Intersections (UoI) framework. We fit models spanning functional coupling (coupling networks from auditory cortex, V1, and M1), sensory encoding (spatio-temporal receptive fields from retinal recordings and tuning curves from auditory cortex), and behavioral decoding (classifying behavioral condition from basal ganglia recordings). We analyzed the fitted models to assess whether improvements in inference impact the resulting neuroscientific conclusions.

Highly sparse coupling models maintain predictive performance

Functional coupling models detail the statistical interactions between the constituent units (e.g., neurons, electrodes, etc.) of a population. Such models can be used to construct networks, whose structural properties may elucidate the functional and anatomical organization of the neurons within the population [182, 125, 25]. Enhanced sparsity in these models could result in different inferred functional sub-networks reflected in the ensuing graph. Furthermore, obtaining biased parameter estimates obscures the relative importance of neuronal relationships in specific sub-populations. Therefore, precise selection and estimation in coupling models is necessary to properly relate the network structure to the statistical relationships between neurons.

We examined the possibility of building highly sparse and predictive coupling networks by fitting coupling models to data from three brain regions: recordings from auditory cortex (AC), primary visual cortex (V1), and primary motor cortex (M1). The AC data consisted of micro-electrocorticography (μECoG) recordings from rat during the presentation of tone pips (Dougherty & Bouchard, 2019: DB). The V1 data consisted of single-unit recordings in macaque during the presentation of drifting gratings (Kohn & Smith, 2016: KS). The M1 data consisted of single-unit recordings in macaque during self-paced reaches on a grid of targets (O'Doherty, Cardoso, Makin, & Sabes: OCMS). See Methods for further details on

experiments, model fitting, and metrics used for model evaluation, and see 4.1 for a model statistic summary.

We constructed coupling models consisting of either a regularized linear model (AC) or Poisson model (V1, M1) in which the activity of an electrode/single-unit (i.e., node) was modeled using the activities of the remaining electrodes/single-units in the population. Thus, each dataset had as many models as there were distinct electrodes/single-units. We quantified the size of the fitted models with the selection ratio, or the fraction of parameters that were non-zero. We compare the selection ratio between baseline and UoI coupling models across electrodes/single-units in Fig 4.4a. For all three brain regions, UoI models exhibited a marked reduction in the number of utilized electrodes/single-units. Specifically, UoI models used 2.24 (AC), 2.21 (V1), and 5.50 (M1) times fewer features than the corresponding baseline models. Across the populations of electrodes/neurons, this reduction was statistically significant ($p \ll 0.001$; see 4.2) with large effect sizes (AC: $d = 1.74$; V1: $d = 2.26$; M1: $d = 2.49$). Interestingly, while the reduction in features for AC and V1 are roughly similar, the M1 models exhibit a much larger reduction in selection ratio, an observation that holds across the three M1 datasets. Furthermore, we examined coupling fits obtained via $\text{UoI}_{\text{Lasso}}$, finding that the enhanced sparsity persists despite the change in model. Additionally, the UoI linear and Poisson models exhibited similar recovery in selection profiles not recapitulated by baseline procedures (4.5).

We assessed whether the reduction in features resulted in meaningful loss of predictive accuracy. We measured predictive accuracy using the coefficient of determination (R^2) for linear models (AC) and the deviance for Poisson models (M1, V1), both evaluated on held out data. Note that in contrast to R^2, lower deviance is preferable. The predictive performances of baseline and UoI models for each brain region are compared in Fig 4.4b. We observed that there is almost no change in the predictive performance across brain regions, with most points lying on or close to the identity line. We note that while the differences in performance across all models were statistically significant (AC: $p < 10^{-3}$; V1: $p \ll 0.001$; M1: $p \ll 0.001$; see 4.3), the effect sizes of the reduction in predictive performance were very small (AC: $d = 0.005$; V1: $d = 0.05$; M1: $d = 0.03$), making it irrelevant in practice. Thus, these results imply that it is possible to construct highly sparse coupling methods that exhibit little to no loss in predictive performance across brain regions and datasets.

We captured the two previous observations — increased sparsity and maintenance of predictive accuracy — with difference in Bayesian information criterion (BIC) between baseline and UoI methods, $\Delta\text{BIC} = \text{BIC}_{\text{baseline}} - \text{BIC}_{\text{UoI}}$. Lower BIC is preferable, so that positive ΔBIC indicates that UoI is the more parsimonious and preferred model. The distribution of ΔBIC across coupling models is depicted in Fig 4.4c. ΔBIC is positive for all models, with a large median difference (AC: 170; V1: 149; M1: 186; see 4.4). Thus, usage of BIC as a model selection criterion provides very strong evidence against the baseline models.

To characterize the functional relationships inferred by the coupling models, we examined the distribution of coefficient values. We normalized each model's coefficients by the coefficient with largest magnitude across the baseline and UoI models, and concatenated coefficients across models and datasets. We visualized the baseline and UoI coefficient val-

ues using a 2-d hexagonal histogram (Fig 4.4d). First, we observed a density of bins above (positive coefficients) and below (negative coefficients) the identity line (Fig 4.4d: red dashed line). This indicates that the magnitude of non-zero coefficients as fit by UoI are larger than the corresponding non-zero coefficient as fit by the baseline, demonstrating the amelioration of shrinkage and therefore reduction in bias. Next, we observed a density of bins on the $x = 0$ line, indicating a sizeable fraction of coefficients determined to be non-zero by baseline methods are set equal to zero by UoI. This density corroborates the reduction in selection ratio observed in Fig 4.4a. We further note that the density of bins on the $x = 0$ line encompass a wide range of baseline coefficients values, especially for the V1 and M1 datasets. This implies that utilizing a thresholding scheme based on the magnitude of the fitted parameters for a feature selection procedure will not reproduce these results. Lastly, we observe no density of bins along the $y = 0$ line, which indicates that UoI models are likely not identifying the existence of functional relationships which do not exist (i.e., suffering from false positives).

While many of the coefficients set equal to zero by UoI have large magnitude (as measured by baseline methods), the bulk of density lies in coefficients with small magnitude. We found that the difference in distributions of non-zero coefficients between the two procedures is statistically significant ($p \ll 0.001$; Kolmogorov-Smirnov test). We highlight the marginal distribution of non-zero coefficients whose magnitudes are small (Fig 4.4d, top and side histograms). While the baseline histograms (Fig 4.4d, top histograms) have the largest density of coefficients close to zero, the UoI histograms, in a similar range, exhibit a large reduction in density. Together, these results demonstrate that coupling models fit by UoI possess qualitatively different parameter distributions, and raise the possibility that these differences may reflect shrinkage and abundance of false positives.

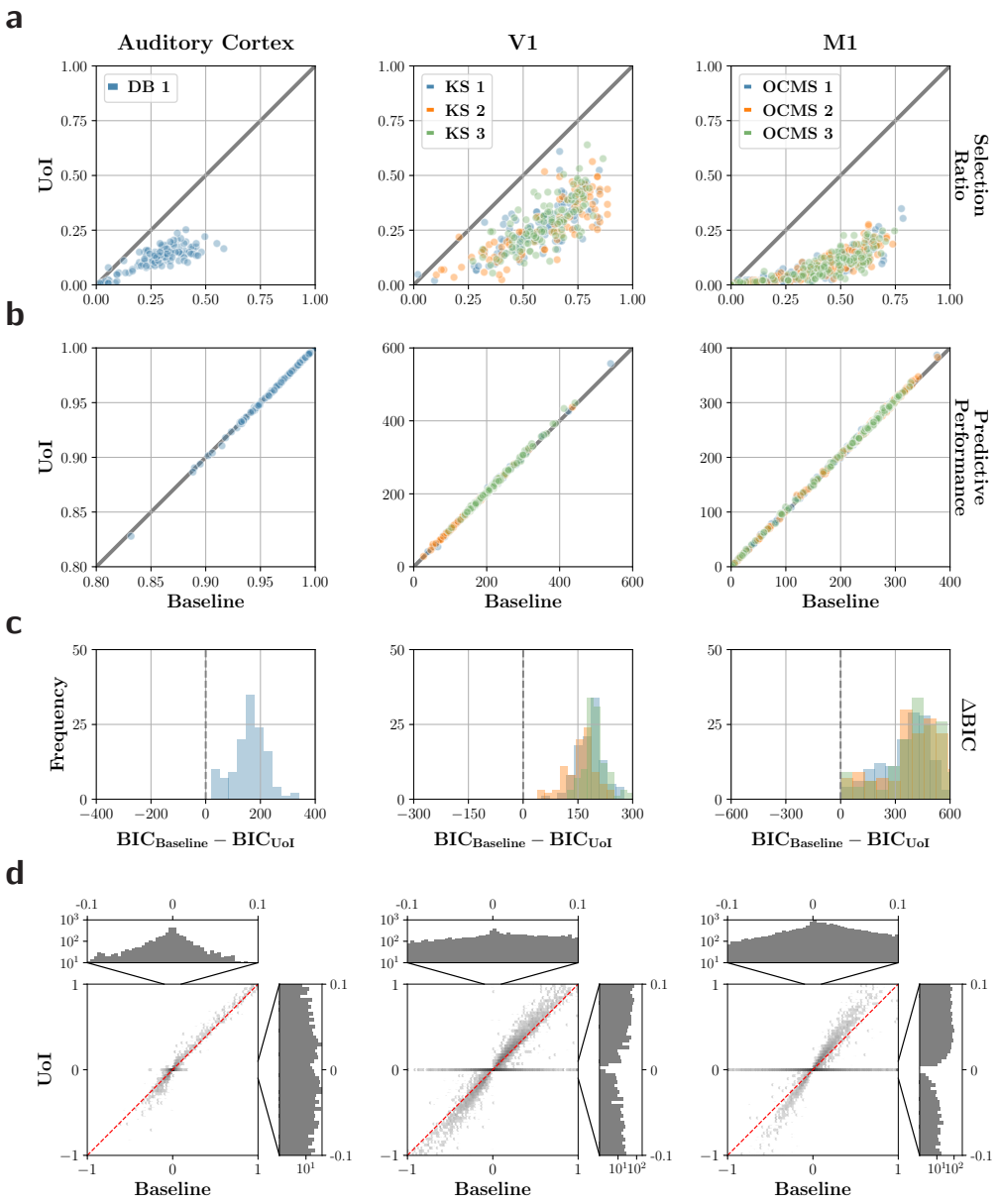


Figure 4.4: **Highly sparse coupling models maintain predictive performance. a-c.** Comparison of coupling models fit using baseline and UoI approaches to recordings from the auditory cortex (AC), primary visual cortex (V1), and primary motor cortex (M1). Each column corresponds to a different recording area (titles), and each row corresponds to a different model evaluation measure (right side labels). Colors denote different data sets within the same experiment, if available. Caption continued on following page.

Figure 4.4: **Continued from previous page. a.** The selection ratio, where the y-axis refers to the UoI model and x-axis the baseline model. Each point represents a coupling model for a specific single-unit/electrode. Gray line denotes the identity line. **b.** Comparison of predictive performance. **c.** The distribution of BIC differences across coupling models. **d.** Comparison of the coefficient values inferred by UoI (y-axis) and baseline (x-axis) methods. Data is depicted on a hexagonal 2D histogram with a log intensity scale. The marginal distributions of the non-zero coefficient values are shown on the top (baseline) and side (UoI) for each brain area. Note that the 1D histograms display the distribution in a more restricted domain than the 2D histograms, as depicted by the black lines.

Improved inference enhances visualization, increases modularity, and decreases small-worldness in functional coupling networks

Functional coupling networks are useful in that they provide opportunities to visualize the statistical relationships within a population. Furthermore, their graph structures can be analyzed to characterize global properties of the network. The previous results show that improved inference gives rise to equally predictive models, but with much greater sparsity and qualitatively different parameter distributions. Thus, we next determined the impact on network visualization and structure. To this end, we constructed networks by placing coefficient values extracted from the coupling models directly in an adjacency matrix; i.e., $A_{ij} = \beta_{ij}$, where β_{ij} is the jth parameter for the ith coupling model. The adjacency matrices constituted directed graphs with weighted edges, which served as the primary focus in the subsequent analyses. When necessary, we considered undirected graphs calculated by symmetrizing pairwise coupling coefficients (see Methods for more details).

We first visualized the AC networks by plotting the baseline and UoI networks according to their spatial organization on the μECoG grid (Fig 4.5a). Each vertex in Fig 4.5a is color-coded by preferred frequency, while the symmetrized coupling coefficients are indicated by the color (sign) and weight (magnitude) of edges between vertices. We observed that the UoI network is easier to visualize, with densities of edges clearly demarcating regions of auditory cortex. This is contrast to the baseline network, whose lack of sparsity makes it difficult to extract any meaningful structure from the visualization. For example, the UoI network exhibits a clear increase in edge density in primary auditory cortex (PAC) relative to the posterior auditory field (PAF) and ventral auditory field (VAF). Thus, the increased sparsity in UoI networks reveals graph structure that ties in closely with general anatomical structure of the recorded region.

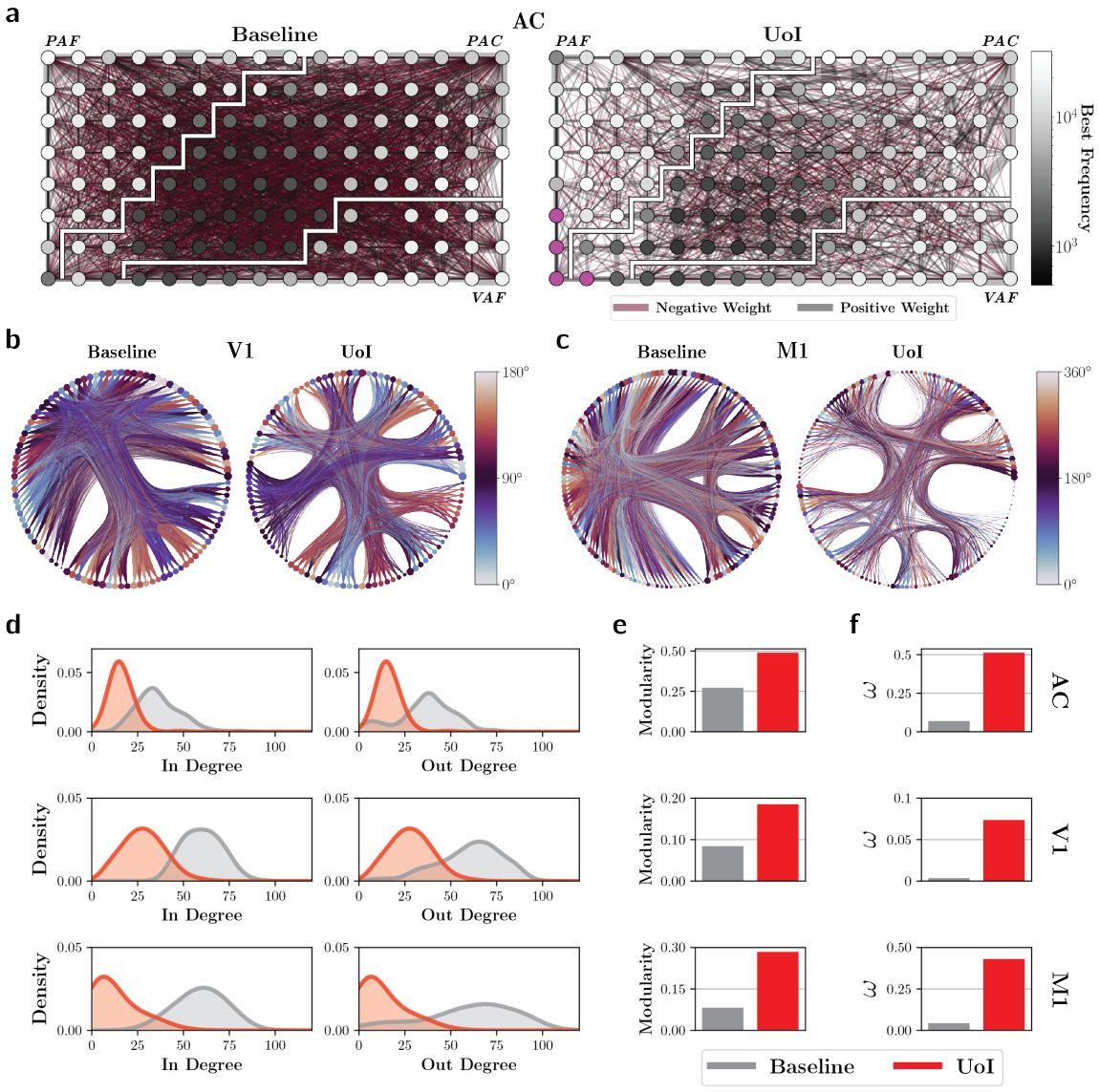


Figure 4.5: **Improved inference enhances visualization, increase modularity, and decrease small-worldness in functional coupling networks. a.** Networks obtained from auditory cortex data. Vertices are organized according to their position on the electrocorticography grid. Vertices are color coded by preferred frequency, with fuchsia vertices denoting non-tuned electrodes. Edge width increases monotonically with edge weight while edge color denotes the sign of the weight. White lines segment the grid according to the regions of auditory cortex. **b.** Visualization of example coupling networks for visual cortex recordings, with vertices color-coded by preferred tuning. Edges are bundled according to detected communities, while vertex size corresponds to its degree. Note that stimulus encoding is cyclical (static grating angle from 0° to 180°). Caption continued on the following page.

Figure 4.5: **Continued from previous page. c.** Visualization of coupling networks for motor cortex recordings, with vertices color-coded by preferred tuning. Edges are bundled according to detected communities, and vertex size corresponds to its degree. Note that stimulus encoding is cyclical (movement direction angle from 0° to 360°). **d-f.** Comparison of common graph metrics evaluated on UoI networks (red) and baseline networks (gray). Each row corresponds to a distinct brain region (left: AC, V1, and M1 from top to bottom). **d.** In-degree and out-degree densities. **e.** The graph modularity, averaged across datasets. **f.** The small-worldness as quantified by ω, and averaged across datasets.

Since we lacked the physical layout for V1 and M1 recordings, we visualized the networks by fitting nested stochastic block models to the directed graphs and plotting the ensuing structure in a circular layout with edge bundling (Fig 4.5b, c). The nested stochastic block model identifies communities of vertices (neurons) in a hierarchical manner. We color-coded the vertices of the visualized graphs according to preferred tuning inferred from fitted encoding models (drifting grating angle for V1 networks and hand movement angle for M1 networks with cosine basis functions: see Methods). The UoI V1 network exhibits clear structure, with specific communities identifying similarly tuned neurons (Fig 4.5b). The baseline V1 network does not exhibit as clear structure, and was highly unbalanced, with more than half the neurons placed in the same community. For the M1 data, the UoI communities were more balanced, though they lacked clear association with tuning properties. Together, these results demonstrate that enhanced sparsity of functional coupling networks result in cleaner visualizations and, in the case of V1, a clearer connection to functional response properties.

These plots suggest different graph structures in the networks extracted by UoI and baseline methods. Thus, we first calculated the in-degree and out-degree distributions of the vertices in both networks (Fig 4.5d). We observed that the in-degree and out-degree distributions for the UoI networks are much smaller, as one might expect due to the reduction in edges. Furthermore, the in- and out-degree distributions describing the UoI networks are nearly identical, in contrast to those of the baseline networks. Next, we calculated the modularity of the networks, which quantifies the degree to which the networks exhibit community-like structure. We found that the modularity for the UoI networks is much larger than that of baseline networks, indicating that UoI networks express more community structure than baseline networks (Fig 4.5e). These results corroborate the visual findings in Fig 4.5b. Since modularity implicitly depends on degree distribution, the enhanced community structure exhibited by the UoI networks is not simply a property of the reduction of in- and out-degrees, emphasizing that the enhanced sparsity is functionally meaningful. We found similar findings in functional networks built from linear models, rather than Poisson models, implying that the structure of coupling networks persists across the type of underlying model (4.5).

Finally, we examined the small-worldness of the networks, a graph structure commonly

used to describe brain networks. Small-world networks are characterized by a high degree of clustering and low characteristic path length, making them efficient for communication. We used ω to calculate small-worldness, whose values are bounded by the range $[-1, 1]$, with ω close to -1, 0, and 1 indicative of lattice, small-world, and random structure, respectively. Interestingly, UoI networks are considerably less small-world than the baseline networks (Fig 4.5f). However, we note that all networks are more small-world than they are random. Furthermore, the small-worldness of the networks is dependent on the brain region. For example, the V1 networks exhibit substantially more small-worldness than the auditory cortex or M1 networks. Together, these results demonstrate that several properties of networks can be substantially altered by the utilized inference procedure, and that UoI networks are more modular and less small-world.

Parsimonious tuning from encoding models

A long-standing goal of neuroscience is to understand how the activity of individual neurons are modulated by factors in the external world (e.g., how the position of a moving bar is encoded by a neuron in the retina). In such encoding models, incorrect feature selection or parameter bias may mistakenly implicate factors in the production of neural activity, or misstate their relative importance. Thus, we examined how improved inference impacts tuning models, where an external stimulus is mapped to the corresponding evoked neural activity.

We first fit spatio-temporal receptive fields (STRFs) to single-unit recordings from isolated mouse retinal ganglion cells during the presentation of a flicker black or white bar stimulus (generated by a pseudo-random sequence). We used a linear model with a lasso penalty to fit STRFs (i.e., regularized, whitened spike-triggered averaging) to recordings from 23 different cells, using a time window of 400 ms. Thus, the fitted STRFs were two dimensional, with one dimensional capturing space (location in the bar stimulus) and the other capturing the time relative to neural spiking. For further experimental and model fitting details, see Methods. See 4.5 for a dataset and model statistic summary.

The fitted STRFs for an example retinal ganglion cell are depicted in Fig 4.6a. The UoI STRF captures the ON-OFF structure exhibited by the baseline STRF. However, the UoI model is noticeably sparser, resulting in a tighter spatial receptive field. The features set to zero by UoI (relative to baseline) include regions both further from the dominant ON-OFF structure, and regions very close to the center. Additionally, the coefficient values of the UoI STRF are noticeably larger in magnitude. These observations raise the possibility that the baseline procedure could be producing false positives in both the central and distal regions of the STRF, with the remaining coefficients suffering from shrinkage.

We compared the selection ratios across fitted STRFs (Fig 4.6b) and found UoI fits to be substantially sparser, with a median reduction factor of 4.98. This reduction was statistically significant ($p \ll 0.001$; see 4.6) and had a very large effect size ($d = 3.05$). At the same time, UoI models exhibited a statistically significant improvement in R^2 ($p < 0.01$; see Table 4.7), but with a very small effect size, making the improvement irrelevant in

practice ($d = 0.05$; see 4.8). Meanwhile, the BIC differences (Fig 4.6d) were all large and positive (median difference = 654), providing very strong evidence in favor of the UoI model. Lastly, we compared the distribution of baseline and UoI encoding coefficients, normalized to the largest magnitude coefficient. We found evidence that the UoI models exhibited reduced shrinkage (Fig 4.6e: larger tails), and a substantial reduction in false positives (Fig 4.6e: reduced density at origin). Thus, improved inference resulted in STRFs with tighter structure.

Across neurons, we observed selection ratios and predictive performances spanning a wide range of values (Fig 4.6b, c). We might expect that models with little predictive accuracy utilize fewer features, since poor predictive performance indicates that the provided features are inadequate. For example, in the limit that the model has no predictive capacity ($R^2 \leq 0$), the model should utilize no features, since such an R^2 indicates that none of the available features are relevant for reproducing the response statistics better than the mean response value. Therefore, we sought to determine whether inaccurate inference mistakenly identifies models as tuned (i.e., non-zero tuning features), when in fact a "non-tuned" model is appropriate (e.g., an intercept model: all features set equal to zero). To this end, we utilized a dataset in which the feature space dimensionality is small. This scenario provides a suitable test bed for assessing whether an intercept model could arise, and if such a model is appropriate given the response statistics of the data. We examined a dataset consisting of μECoG recordings from rat auditory cortex during the presentation of tone pips at various frequencies. We employed a linear tuning model mapping frequency to the peak (z-scored relative to baseline, see Methods), high-γ band analytic amplitude of each electrode. The model features consisted of 8 Gaussian basis functions that tiled the log-frequency space.

We first examined whether improved inference resulted in any qualitative changes in the fitted encoding models. We plotted the fitted tuning curves as a function of log-frequency for a subset of electrodes arranged according to their location on the μECoG grid (Fig 4.6f). Interestingly, the baseline and UoI tuning curves exhibit similar structure for a large fraction of the electrodes on the grid, in many cases matching closely (e.g., Fig 4.6f: anterior side of grid). In other cases, particularly on the posterior side of the grid, the UoI tuning curves exhibit similar broad structure with noticeable smoothing, indicating that UoI has simplified the tuning model.

We compared the selection ratio of the models (Fig 4.6g), and found that the UoI tuning curves utilize fewer features than those fit by baseline, with a median reduction factor of 2.5 that was statistically significant ($p \ll 0.001$; see 4.6) and a large effect size ($d = 2.19$). Furthermore, despite a statistically significant decrease in R^2 (Fig 4.6h) across electrodes ($p \ll 0.001$; see 4.7), the observed effect size is very small (median $\Delta R^2 = 0.001$; $d = 0.05$). Meanwhile, we observed a median BIC difference of 19.4, providing evidence in favor of the UoI models (Fig 4.6i; Table 4.8). Taken together, these results imply that the reduction in features did not harm the predictive performance of the tuning models, thereby enhancing their parsimony.

We highlight four electrodes whose selection ratios, according to UoI, are exactly zero in Figure 4.6g (purple points; some points overlap). The tuning curves of two of these

electrodes are depicted in the posterior region of Figure 4.6f (purple axis boundaries). These four "non-tuned" electrodes are among the least predictive, with R^2 close to or below zero for both baseline and UoI methods (Fig 4.6h: purple points; some points overlap). Interestingly, the baseline selection ratio for one of these models was close to 0.4, indicating that UoI is generating models with substantially different support. We visually examined the frequency-response areas (FRAs) of these four electrodes, which detail the mean response values as a function of sound frequency and amplitude. We compared them to the FRAs of two randomly chosen electrodes, finding they had little discernible structure relative to the "tuned" FRAs (4.5). Thus, while increased sparsity in encoding models may not always result in perceptible changes in their appearance (e.g., Fig 4.6f), there are cases where an inference procedure may mistakenly imply that a constituent unit is tuned, when in fact the stimulus features are not relevant for capturing its response statistics. To understand the behavior of the selection ratio as $R^2 \to 0$, we examined the relationship between selection ratio and R^2 for baseline (gray) and UoI (red) models (Fig 4.6j). We found that the sparser models exhibit lower predictive power, with model trends predicting that at $R^2 = 0$, the selection ratio for the baseline model would be 0.35 ± 0.10 (mean $\pm$ 1 s.d.) while the UoI selection ratio would be 0.12 ± 0.10. This demonstrates that baseline procedures can suffer from false positives even when their fitted models exhibit little to no explanatory power. Overall, our results reveal that UoI can identify units as non-tuned when their encoding models lack predictive ability.

Decoding behavioral condition from neural activity with a small number of single-units

Decoding models describe which neuronal sub-populations contain information relevant for an external factor, such as a stimulus or a behavioral feature. Such models can identify which neurons may be useful to a downstream population for a task that requires knowledge of an external factor. Specifically, a decoding model's non-zero parameters can be interpreted as the sub-population of neurons containing the task-relevant information, emphasizing the need for precise selection. Additionally, the model details how specific neurons describe the decoded variable through the magnitudes of its parameters, requiring unbiased estimation. Thus, we sought to assess the degree to which an improved inference algorithm might impact data-driven discovery in neural decoding models.

For this analysis we examined 54 single units in the rat basal ganglia (18 units from globus pallidus pars externa, GPe, and 36 from the substantia nigra pars reticulata, SNr) that were recorded simultaneously during performance of a behavioral task involving rapid leftward or rightward head movements in response to cues. Details of the task are given in [79]; the analysis was restricted to trials in which a correct head movement was made. Thus, the decoding model consisted of binary logistic regression predicting trial outcome (left or right) using the single-unit spike counts as features. We fit the logistic regression with an ℓ_1 penalty (baseline) and the UoI framework ($\text{UoI}_{\text{Logistic}}$). Further details on the experimental

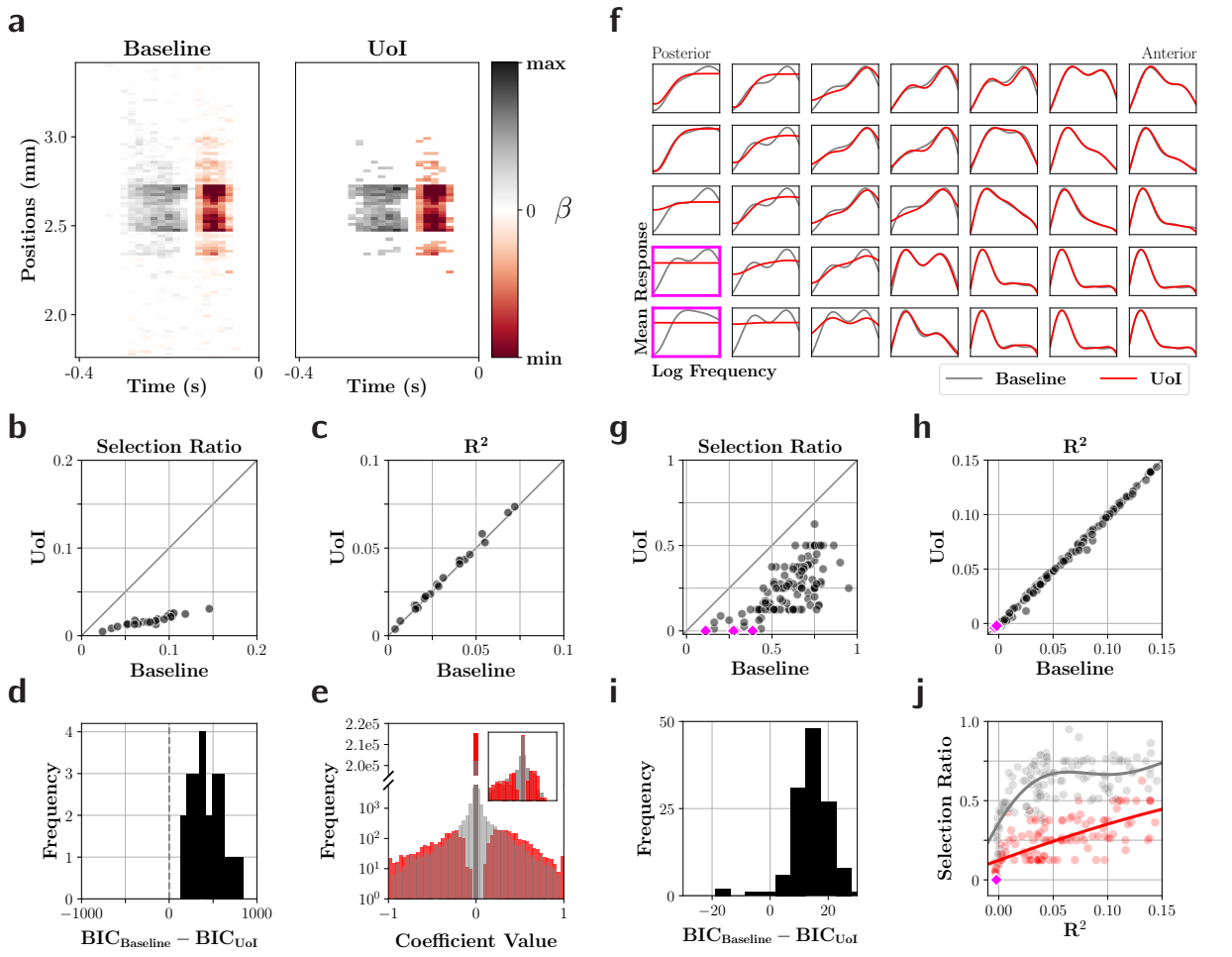


Figure 4.6: **Parsimonious tuning from encoding models. a-e.** Analysis of spatiotemporal receptive fields (STRFs) fit to spikes from mouse retinal ganglion cells using a 1d white noise stimulus. **a.** Example STRF extracted by baseline (left) and UoI (right) procedures. **b-c.** Quantitative comparison of STRFs extracted with UoI (y-axis) and baseline (x-axis) procedures. Each point represents a unique STRF. Gray line denotes identity. **b.** Comparison of selection ratios for each STRF. **c.** Comparison of coefficient of determination (R^2) on held out data. **d.** The distribution in BIC differences across STRFs. Dashed line denotes equal BIC, i.e. $\Delta\text{BIC} = 0$. **e.** Distribution of normalized coefficient values across all baseline (gray) and UoI (red) STRFs. The broken y-axis is log-scale below the break and a regular scale above the break. Inset shows distribution of coefficients for STRFs depicted in **a.** **f-j.** Analysis of tuning curves extracted from micro-electrocorticography recordings on rat auditory cortex during the presentation of tone pips. Caption continued on following page.

setup and model fitting can be found in Methods. See 4.9 for a summary of the dataset and

Figure 4.6: **Continued from previous page. f.** Examples of tuning curves for a subset of electrodes on the grid, as fit by baseline (gray) and UoI (red) procedures. Purple outlines denote tuning curves set exactly equal to zero by UoI (of the four electrodes, only two are shown). **g-h.** Quantitative comparison of tuning curves extracted with UoI (y-axis) and baseline (x-axis) procedures. Panels are structured similarly as panels **b-c.**. Purple points highlight tuning curves that had a selection ratio of zero, as determined by UoI. **i.** The distribution in BIC differences across electrode tuning curves, structured similarly as panel **d. j.** Selection ratio plotted against coefficient of determination for baseline (gray) and UoI (red) procedures. Each point denotes a STRF. Trendlines are fit with Gaussian process regression.

fitted model statistics.

The selection ratios for GPe and SNr, as obtained by baseline and UoI procedures, are depicted in Fig 4.7a. In GPe, the UoI decoding models utilized about half the number of parameters as the baseline procedures. Meanwhile, in SNr, the UoI models utilized four times fewer parameters than the baseline. Furthermore, we observe that UoI model sizes were more consistent across folds of the data. For example, the baseline SNr decoding models estimated anywhere from 1 to 21 parameters (out of 36) depending on the data fold, while UoI models consistently used only 2 or 3 parameters (Fig 4.7a: IQR bars). These results validate that neural decoding models are capable of utilizing fewer features to predict relevant behavioral features. Furthermore, the stability principle ensures that these features are more robust to perturbations of the data (e.g., random subsamples).

To examine whether the use of fewer single-units decreased predictive performance, we evaluated the decoding models' classification accuracy on held-out data, depicted in Fig 4.7b. The classification accuracy of the UoI models is equal to that of the baseline models for both GPe (67%) and SNr (100%). Furthermore, in both regions, the classification accuracy is greater than chance (56%), implying that the models are extracting meaningful information about the behavioral condition from the neural response. Interestingly, the median SNr models achieve perfect classification accuracy. The UoI model achieves this performance utilizing only 2 neurons, in contrast to median baseline model, which utilizes 8. Therefore, the activities of only a small subset of neurons are required to predict the behavioral condition, an observation that required an improved inference algorithm to consistently capture.

We examined the fitted coefficient values for each brain region and fitting procedure (Fig 4.7c). First, we observed that all coefficients set equal to zero by the baseline procedure are also set equal to zero by UoI. Additionally, the coefficients set equal to zero by UoI, but not the baseline procedure, typically have smaller magnitudes than the coefficients that are non-zero for both procedures. Finally, the coefficients set non-zero by UoI have larger magnitudes relative to their value under the baseline procedure. These observations imply that the UoI procedure, for this task, consistently utilized only the most important neurons to predict the behavioral condition. At the same time, UoI elevated the coefficient

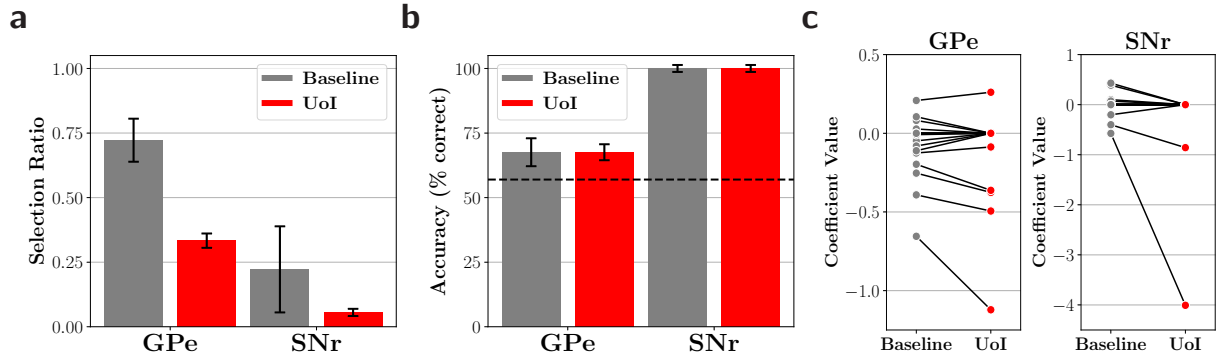


Figure 4.7: **Behavioral condition can be decoded with a small number of single-units at no loss in accuracy.** Decoding models were applied to single-unit recordings from rat basal ganglia. The models, consisting of binary logistic regression, predicted whether the rat went left or right on a stop signal task. Single-unit recordings were used from the globus pallidus pars externa (GPe: 18 units) and the substantia nigra pars reticulata (SNr: 36 units). **a-b.** Evaluation of decoding models fit via the UoI (red) and baseline (gray) procedures. Bar heights indicate median across five data folds, while error bars denote IQR. **a.** Comparison of the selection ratios. **b.** Comparison of left/right classification accuracy on held-out data. Dashed line denotes accuracy by chance. **c.** Comparison of fitted coefficient values extracted by baseline and UoI procedures in GPe and SNr. The decoding models fit by UoI utilize about 3 times fewer single-units at no cost to accuracy, indicating that task relevant information is contained in a small number of single-units.

values relative to the baseline procedure, implying that baseline procedures suffered from substantial parameter shrinkage. In contrast to the coupling models, the parameters with the lowest magnitude in the baseline decoding models were set to zero by UoI. This could reflect differences in the task (classification vs. regression) or collinearity between the neural responses. Overall, these results demonstrate that task-relevant information is conveyed by a sparse subset of basal ganglia neurons, especially in SNr.

4.4 Discussion

Parametric models are used pervasively in systems neuroscience to characterize neural activity. The parameters of these models must be precisely selected and estimated in order to ensure accurate interpretation, particularly in the sparse parameter regime that is desirable for neural data. Motivated by the advent of new inference procedures capable of improved inference, we used the UoI framework to assess the degree to which poor parameter inference may impact neuroscientific interpretation of parametric models. We fit functional coupling, encoding, and decoding models to a battery of neural datasets, using standard and UoI inference procedures. We found that, across all models, the number of non-zero parameters could

be reduced by a factor of 2–5, while maintaining predictive performance. Furthermore, we found broader, structural differences in the models beyond enhanced sparsity, which resulted in concrete changes to secondary analyses that inform neuroscientific interpretation.

The parameters obtained in coupling models denote the existence and strength of functional relationships between constituent units in a population. We fit coupling models that exhibited a marked reduction in model size, providing evidence that the baseline models suffered from false positives. Interestingly, the amount of feature reduction varied across brain areas, which may reflect differences in the nature of neural activity for these regions. Furthermore, we observed striking differences in the distribution of these parameter estimates, which impacted the graph structure. These changes produced cleaner visualizations, which in the case of V1, could related to the functional response properties of the neurons. In M1, we found no such relationship, inline with the dynamics view of this region [48]. Additionally, these networks were characterized by increased modularity and decreased small-worldness. These results do not directly contradict previous work characterizing brain networks as small-world, but do reduce the magnitude of the characterization [26, 83, 24] (though see [123]). Furthermore, coupling model parameters have been assessed for their recapitulation of synaptic weight distributions in neural circuits, in some cases identifying parameter biases induced by specific dynamical regimes of neural activity [55]. These biases could be due to shrinkage of large parameter values, as observed here, and are mitigated by UoI. Furthermore, the coupling parameters extracted by UoI better reflect the weight distribution as observed in neural circuits, which is characterized by sparse connectivity with a heavy-tailed distribution [183, 21]. This was not achieved by baseline procedures, suggesting that the previously identified biases are due to inaccurate inference. The salient differences in the inferred coupling parameter distributions we observed motivates similar examination in models that capture neural dynamics, such as vector auto-regressive models [16, 159], which could be further assessed by controllability metrics used in recent work on fMRI networks [80].

The parameters in encoding models detail which features modulate neural activity. We observed that the application of UoI to the encoding models highlighted cases where the fitted model had only zero parameter values, other than the intercept. Such an intercept model implies that a tuning model may be inappropriate for capturing the response statistics of the constituent units. This observation can be understood as a natural consequence of stability enforcement during parameter inference: UoI benefits from the stability principle by only utilizing selected features that persist across data resamples [129]. The use of data resamples mimics perturbing the dataset, ensuring that features are included only if they are robust to those perturbations. Thus, the stability principle enforces model parsimony by encouraging the use of fewer features, eliminating those that offer no predictive accuracy throughout the resamples. However, UoI prioritizes predictive accuracy in the model averaging step. Therefore, models will only be made "as simple as possible" (e.g., removing all features) when they possesses no predictive ability. Furthermore, the fit spatio-temporal receptive fields on the retinal ganglion cells had typical ON-OFF structure, characteristic of the linear model [152]. In contrast to the auditory cortex, the stability enforcement of UoI resulted

in models that were more spatially constrained, in better agreement with theoretical work characterizing the receptive fields and more accurately reflecting the visual features that explain the production of neural activity in retinal ganglion cells [141, 191]. More broadly, these results indicate that such improvements in parameter inference could serve to close the gap between experiment and theory in systems neuroscience.

Likewise, decoding models can inform which internal factors contain information about a task-relevant external factor. We found that decoding models could be fit using fewer single-units, at no cost to classification accuracy, implying that task-relevant information can be confined to a very small fraction of a neural circuit. In the context of this work, we note that SNr is a basal ganglia output nucleus, receiving converging inputs from multiple basal ganglia structures including GPe. Thus, the finding that SNr decodes behavioral output more selectively and accurately compared to GPe is consistent with the idea that SNr is closer to post-decision behavioral outputs, whereas GPe represent internal preparatory states. Our observations have more general implications for communication between brain areas: wiring constraints often restrict information transmission through a relatively smaller number of projection neurons, suggesting that these neurons contain the relevant information required to "decode" a given signal [168, 89]. These results, in which we identified a very small fraction of the neurons capable of accurate decoding, raise the possibility that these selected neurons are, in fact, the projection neurons. Decoding using fewer single-units also has practical implications. An abundance of work has considered the fidelity of a neural code by assessing the decoding ability of neural populations as a function of population size [107]. These decoding analyses can be informed by knowledge of the sparse sub-populations predictive of an external factor, which these results indicate are smaller than previously thought. Brain-machine interfaces (BMIs), which rely on accurate decoding from neural activity to operate, could reduce their power consumption by using a decoder relying on fewer single-units. Together, these results imply that accurate inference procedures, more capable of discovering specific task-relevant neuronal sub-populations, could drive the development of normative theories of neural communication and decoding.

Across brain regions and models, UoI resulted in more parsimonious models with differences in predictive performance that were irrelevant in practice, as measured by Cohen's d. However, statistical tests comparing the distribution of predictive performance between the baseline and UoI models revealed a statistically significant decrease in predictive performance for some cases (coupling models, AC tuning) and statistically significant increase in others (retinal STRF, decoding). Depending on one's goals, relying solely on predictive performance to judge a model may be unreliable [177, 207]. In particular, because model interpretability depends crucially on the included features and their estimates, we prioritized feature selection and estimation. Cross-validated predictive accuracy is often a poor criterion for accurate feature selection [176, 222]. In these cases, the BIC, which captures model parsimony, serves as a more suitable criterion [170], and universally favored the UoI models (though we note there is no single preferred model selection criterion [170, 4, 157, 74, 222]). Furthermore, the increase in sparsity of the UoI models imply that the baseline fits suffered from an abundance of false positives. The similar predictive performance between the UoI

and baseline fits could imply that false positives can have little impact on predictive performance in models of neural data, further emphasizing the need to consider multiple criteria in model selection. Alternatively, UoI fits may have suffered from false negatives, thereby lowering predictive performance, though previous experiments in synthetic data make this unlikely [31].

We considered models of neural activity exclusively in terms of coupling, encoding, or decoding. However, past studies have built parametric models of neural activity by using other features or model structures. For example, the combination of coupling and encoding in a single model is a natural extension which has been examined extensively in previous work [188, 152, 204, 200, 139]. Other features that are not constrained within coupling, encoding, or decoding — such as spike-time history or global fluctuations — are also important to incorporate [200, 140, 145]. Additionally, latent variable models have been used to great success to capture, in particular, the dynamics of neural activity [43, 147, 121]. In this work, the stability principles used by UoI resulted in a significant difference in the model sparsity and estimated parameter distribution, which impacted interpretation. Future work should assess whether similar results can be achieved in these extended models, and if so, determine the neuroscientific consequences.

UoI is formulated in a frequentist context, which guided the baseline methods we chose as comparisons. This was done to have maximal utilization by neuroscientists, which predominantly utilize frequentist inference. However, a multitude of inference approaches, including Bayesian methods [47, 224, 87], have been introduced in recent years which all excel at parameter inference. This new class of inference approaches will fundamentally change the interpretation of parametric models on neural data, and therefore should preferentially be used in future studies to improve data-driven discovery. As these inference procedures continue to be improved, additional assessments on neural data will be informative to better understand how neuroscientific interpretation further develops.

We restricted our analysis to generalized linear models, because their structure lends itself well to interpretation, making them ubiquitous in neuroscience. However, the improvements we obtained by encouraging stability and sparsity in these models may extend to other classes of models. For example, dimensionality reduction methods have also played an important role in systems neuroscience [54]. The UoI framework is naturally extendable to such methods, including column subset selection [31] and non-negative matrix factorization [201]. Furthermore, recent work has found success in using artificial neural networks (ANNs) to model neural activity, which excel at predictive performance [101]. Since ANNs are highly parameterized, these models are not interpretable in the sense that their parameter values do not directly convey neuroscientific meaning. Instead, these models are often interpreted through the lens of learned representations or recapitulation of emergent properties of neural activity. Future work could assess whether the inference principles described in this work could have similar effects for ANNs modeling neural activity, especially given recent advancements in compressing such models [1].

4.5 Supporting Analyses

Pseudocode for the Union of Intersections

We provide pseudocode to the Union of Intersections framework. This pseudocode is generalized to all UoI algorithms discussed in the paper, with the objective $L(\beta; X^k, y^k)$ corresponding to the choice of algorithm.

Algorithm 1 Union of Intersections inference in generalized linear models

Input: Data $X \in \mathbb{R}^{n \times p}$, $y \in \mathbb{R}^n$
Regularization strengths $\{\lambda_j\}_{j=1}^q$
Number of resamples N_S and N_E
Loss function $L(\beta; X, y)$
Model Selection
for $k = 1$ to N_S **do**
  Generate resample X^k, y^k
  for $j = 1$ to q **do**
    $\hat{\beta}^{jk} \leftarrow$ Optimize $L(\beta; X^k, y^k) + \lambda_j |\beta|_1$ (ℓ_1-penalized objective)
    $S_j^k \leftarrow \{i\}$ where $\hat{\beta}_i^{jk} \neq 0$
  end for
end for
for $j = 1$ to q **do**
  $S_j \leftarrow \bigcap_{k=1}^{N_S} S_j^k$ ▷ *Intersection*
end for
Model Estimation
for $k = 1$ to N_E **do**
  Generate training $\left(X_T^k, y_T^k\right)$ resample
  for $j = 1$ to q **do**
    $X_{T,j}^k \leftarrow X_T^k$ with features S_j extracted.
    $\hat{\beta}^{jk} \leftarrow$ Optimize $L(\beta; X_{T,j}^k, y_T^k)$
    $\ell^{jk} \leftarrow \mathrm{BIC}(\hat{\beta}^{jk}, X_{T,j}^k, y_T^k)$
  end for
  $\hat{\beta}^k \leftarrow \underset{\hat{\beta}^{jk}}{\operatorname{argmin}}\ \ell^{jk}$
end for
$\hat{\beta}^* = \underset{k}{\operatorname{median}} \left(\hat{\beta}^k\right)$ ▷ *Union*
return $\hat{\beta}^*$

Extended synthetic results for Poisson and logistic variants

We conducted additional synthetic experiments evaluating the performance of UoI in the context of Poisson and logistic regression. As in the case of the linear model (detailed in Section 4.2), we generated data with $p = 300$ total parameters of which $k = 100$ were non-zero. The non-zero ground truth parameters were drawn from a parameter distribution characterized by exponentially increasing density as a function of parameter magnitude (Fig. 4.3b: gray histograms). We used $N = 1200$ samples generated according to the Poisson model and $N = 2400$ samples generated according to the logistic model (logistic regression typically requires more samples for convergence). We compared UoI against ℓ_1-penalized Poisson regression, fit by `glmnet` [68], and ℓ_1-penalized logistic regression, fit by `scikit-learn` [150]. We did not conduct an analysis against a full battery of methods, as done in Section 4.2, because solvers beyond the ℓ_1-penalty are not readily available in the Poisson and logistic contexts.

We summarized our findings with similar metrics detailed in Section 4.2. Specifically, we quantified selection performance with the selection accuracy, bias with estimation error, variance with the estimation variability, predictive performance with the deviance and log-likelihood, and model parsimony with the Bayesian information criterion. Our results are shown in Figure 4.8, with boxplots capturing the distribution of metrics across 30 synthetic datasets (in the case of estimation variability, the distribution across parameters). We found that UoI exhibits increased selection accuracy (Fig. 4.8a, b) and decreased bias (Fig. 4.8c,d). At the same time, UoI exhibits comparable variability, with most parameters seeing an improvement, but a longer tail exhibiting increased variance (Fig. 4.8e, f). These improvements in model sparsity and bias translated to improved prediction performance, as quantified by the deviance for Poisson regression (Fig. 4.8g: lower is better) and the log-likelihood for logistic regression (Fig. 4.8h: higher is better). Lastly, UoI exhibited markedly improved model parsimony as quantified by the Bayesian information criterion (Fig. 4.8i: lower is better).

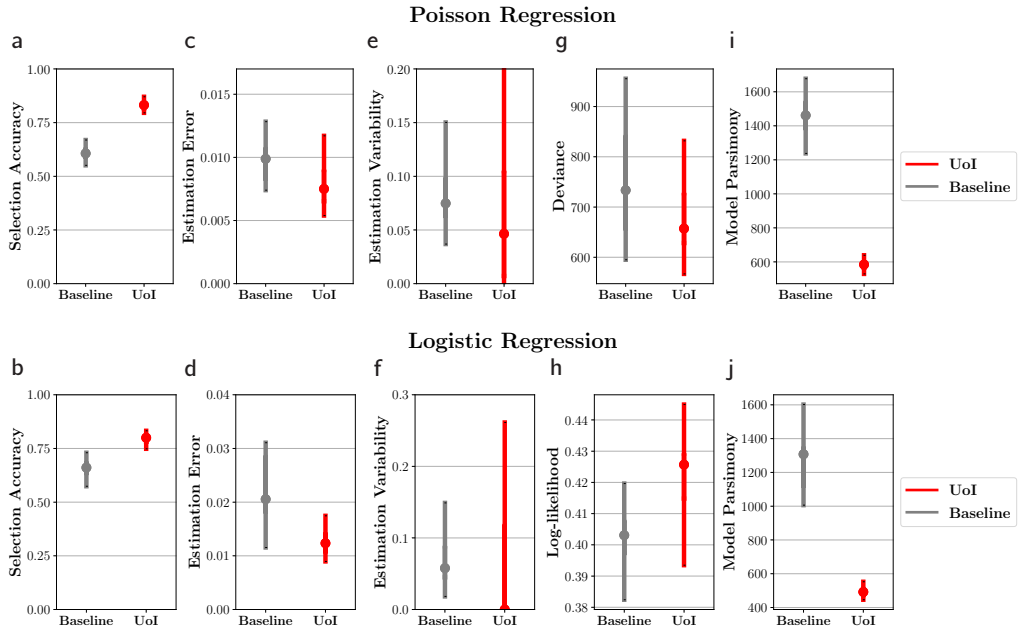


Figure 4.8: **UoI exhibits improved selection, decreased bias, comparable variance, and superior model parsimony on Poisson and logistic variants.** Top row depicts results for Poisson regression, while bottom row depicts results for logistic regression. Each column corresponds to a different metric, with red box plots denoting UoI performance and gray boxplots denoting baseline performance. Circular points denote median, with whiskers denoting the total spread of responses. **a, b.** Comparison of selection accuracies (higher is better). **c, d.** Comparison of estimation errors, quantifying bias (lower is better). **e, f.** Comparison of estimation variabilities (lower is better). **g, h.** Comparison of predictive performance. For Poisson models, this is quantified by deviance (**g**: lower is better). For logistic models, this is quantified by log-likelihood (**h**: higher is better). **i, j.** Comparison of model parsimonies, as quantified by Bayesian information criterion (lower is better).

Supplementary Tables

Functional coupling dataset summary and fitted model statistics

Dataset summary (Table 4.1) provides details on the datasets used to fit functional coupling models, including the number of units and samples across brain region and recording session. The following tables provide statistics summarizing aspects of the fitted baseline and UoI models, including selection ratio (Table 4.2), predictive performance (Table 4.3), and Bayesian information criterion (Table 4.4).

Brain Region	Number of Units	Number of Samples
AC	125	4200
VC 1	106	2400
VC 2	88	2400
VC 3	112	2400
MC 1	136	4089
MC 2	146	4767
MC 3	147	4400

Table 4.1: Dataset summary for functional coupling models

Brain Region	Baseline	UoI	Reduction Factor	p-value	d
AC	0.30 ± 0.07	0.13 ± 0.03	2.24	5×10^{-23}	1.74
VC 1	0.59 ± 0.11	0.27 ± 0.07	2.21	2×10^{-19}	2.26
VC 2	0.66 ± 0.17	0.26 ± 0.11	2.56	2×10^{-16}	1.87
VC 3	0.59 ± 0.10	0.28 ± 0.07	2.13	2×10^{-20}	2.57
MC 1	0.43 ± 0.12	0.07 ± 0.04	5.85	3×10^{-24}	2.49
MC 2	0.46 ± 0.12	0.08 ± 0.04	5.50	3×10^{-25}	2.37
MC 3	0.46 ± 0.10	0.09 ± 0.04	5.15	1×10^{-25}	2.58

Table 4.2: Selection ratios for functional coupling models

Brain Region	Baseline	UoI	p-value	d
AC	0.98 ± 0.02	0.98 ± 0.02	5×10^{-4}	0.005
VC 1	203 ± 35	205 ± 36	3×10^{-16}	0.05
VC 2	166 ± 44	168 ± 43	3×10^{-12}	0.03
VC 3	213 ± 42	217 ± 41	2×10^{-18}	0.05
MC 1	232 ± 65	233 ± 65	1×10^{-23}	0.03
MC 2	263 ± 60	266 ± 59	1×10^{-23}	0.03
MC 3	248 ± 57	250 ± 58	2×10^{-25}	0.03

Table 4.3: Predictive performance for functional coupling models

Brain Region	Baseline	UoI	Median Difference
AC	-6240 ± 1884	-6491 ± 1859	170
VC 1	-15349 ± 20946	-15516 ± 20970	149
VC 2	-2864 ± 11046	-3007 ± 11087	131
VC 3	-35164 ± 34069	-35309 ± 34066	161
MC 1	481 ± 124	321 ± 59	162
MC 2	542 ± 150	354 ± 75	186
MC 3	564 ± 120	362 ± 67	190

Table 4.4: Bayesian information criteria for functional coupling models

Encoding model dataset summary and fitted model statistics.

Dataset summary (Table 4.5) provides details on the datasets used to fit encoding models, including the number of units and samples across dataset. The following tables provide statistics summarizing aspects of the fitted baseline and UoI models, including selection ratio (Table 4.6), predictive performance (Table 4.7), and Bayesian information criterion (Table 4.8).

Brain Region	Number of Units	Number of Samples
Retina	125	4200
AC	23	89896

Table 4.5: Dataset summary for encoding models

Brain Region	Baseline	UoI	Reduction Factor	p-value	d
Retina	0.084 ± 0.020	0.017 ± 0.004	4.98	3×10^{-5}	2.21
AC	0.625 ± 0.116	0.250 ± 0.114	2.50	7×10^{-23}	2.19

Table 4.6: Selection ratios for encoding models

Brain Region	Baseline	UoI	p-value	d
Retina	0.028 ± 0.013	0.028 ± 0.013	0.004	0.05
AC	0.042 ± 0.034	0.041 ± 0.033	2×10^{-19}	0.05

Table 4.7: Predictive performance for encoding models

Brain Region	Baseline	UoI	Median Difference
Retina	-1646606 ± 44936	-1647261 ± 44857	654
AC	4371 ± 2495	4351 ± 2493	19.4

Table 4.8: Bayesian information criteria for encoding models

Decoding model dataset summary and fitted model statistics

Dataset summary (Table C.3a) provides details on the datasets used to fit decoding models, including the number of units and samples across dataset. The following tables provide statistics summarizing aspects of the fitted baseline and UoI models, including selection ratio (Table C.3b), and predictive performance (Table C.3c).

Brain Region	Number of Units	Number of Samples
GPe	18	186
SNr	36	186

Table 4.9: Dataset summary for decoding models

Brain Region	Baseline	UoI	Reduction Factor
GPe	0.722 ± 0.083	0.333 ± 0.028	2.167
SNr	0.222 ± 0.167	0.056 ± 0.014	4

Table 4.10: Selection ratios for decoding models

Brain Region	Baseline	UoI
GPe	0.676 ± 0.054	0.676 ± 0.031
SNr	1.000 ± 0.014	1.000 ± 0.014

Table 4.11: Prediction performance for decoding models

Comparison of Poisson and linear coupling models for single-unit activity

We used a Poisson distribution to model single-unit spike count activity in the M1 and V1 datasets. However, past work has modeled single-unit activity with a linear-Gaussian model, after applying variance-stabilizing square root transform to the spike count responses. The degree to which a linear model can capture the functional relationships identified by a Poisson model for spiking data is unclear. Thus, we sought to characterize this capability, and its dependence on the inference procedure. We modeled the neural activity after a square-root transform $\sqrt{n_i}$ using a linear model

$$\sqrt{n_i} = \beta_0 + \sum_{j=1}^{M} \beta_{ij} \sqrt{n_j} + \epsilon. \tag{4.26}$$

We fit this model with lasso optimization by coordinate descent (baseline) and $\text{UoI}_{\text{Lasso}}$.

We compared the fitted selection profiles, i.e. the set $S = \{i | \beta_i \neq 0\}$ between the linear and Poisson models. To do so, we used the hypergeometric distribution, which describes the probability that k objects with a particular feature are drawn from a population of size M that has K total objects with that feature, using m draws without replacement. To frame this in terms of selection, suppose the Poisson model has $|S_{\text{Poisson}}| = K$ non-zero parameters out of the M possible features. Then, the probability the linear model, which has $|S_{\text{linear}}| = m$ non-zero parameters, would match the Poisson model on k such features is given by the hypergeometric distribution:

$$\Pr(k) = \frac{\binom{K}{k}\binom{M-k}{m-k}}{\binom{M}{m}}. \tag{4.27}$$

Thus, the probability that the selection profile would overlap at most as well by chance as the linear model is given by $p_{\text{overlap}} = 1 - \Pr(k < k_{\text{linear}})$. We compared the distribution of p_{overlap} across coupling models, calculated for both baseline and UoI procedures (Fig. 4.1, panel a). For the V1 data, the UoI linear models better reproduced the Poisson selection profiles, with 98% of the profiles fit by UoI satisfying $p_{\text{overlap}} < 0.001$, compared to only 74% of the baseline selection profiles. In contrast, in the M1 data, both inference procedures fit linear models that closely matched the selection profiles of the Poisson models, with 99% of the selection profiles satisfying $p_{\text{overlap}} < 0.001$. Therefore, improved inference results in

more consistent selection across models, and furthermore this consistency depends on brain region.

We constructed networks from the linear models as described in Methods, and calculated their in-degree and out-degree distributions. We compared the in-degree and out-degree distributions of the linear networks to the Poisson networks, finding a closer correspondence between the UoI models than for the baseline models in most cases (Fig. 4.1, panel b). Specifically, the in-degrees of UoI models had a correlation of 0.742 (V1) and 0.969 (M1) compared to 0.717 (V1) and 0.924 (M1) for the baseline procedures. Similarly, we obtained out-degree correlations of 0.806 (UoI) and 0.531 (baseline) for V1 and 0.918 (UoI) and 0.924 (baseline) for M1. Lastly, we found that the UoI linear networks were more modular than the baseline linear networks. Interestingly, both were more modular than their Poisson counterparts (4.1, panel c). Taken together, these results imply that a more precise inference framework better preserves structure across model types.

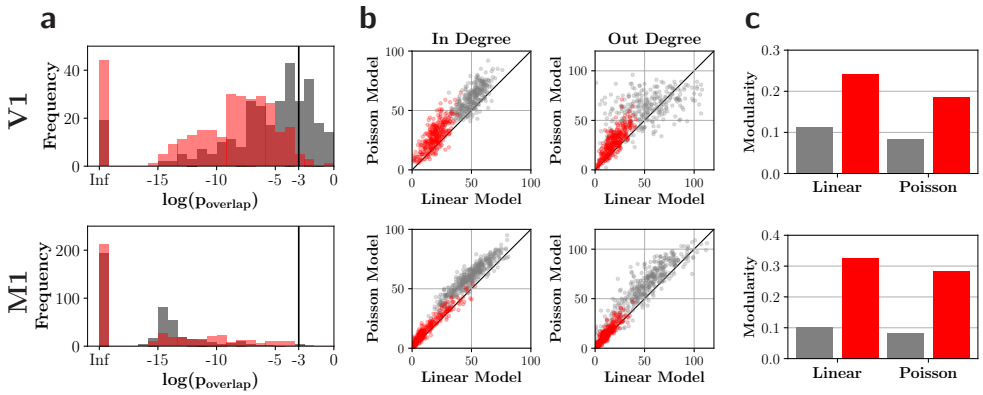


Figure 4.1: **Improved inference ensures that the structure of fitted coupling networks persists across the type of underlying model.** Linear and Poisson coupling models were fit to the datasets with single-unit recordings (V1 and M1) using baseline (gray) and UoI (red) procedures. Top row corresponds to results from networks fit to V1 recordings, while bottom row corresponds to networks fit to M1 recordings. **a.** The (log) probability distribution of extracting a support (set of non-zero parameters) by the linear model that matches with the Poisson model support, according to a hypergeometric distribution. Vertical line denotes a p-value of 0.001 **b.** Comparison of the in-degree and out-degree distributions between the Poisson network (y-axis) and linear network (x-axis). Each point represents a single unit. Black line denotes identity. **c.** Graph modularity of linear and Poisson networks.

Frequency response area analysis for tuned and non-tuned electrodes

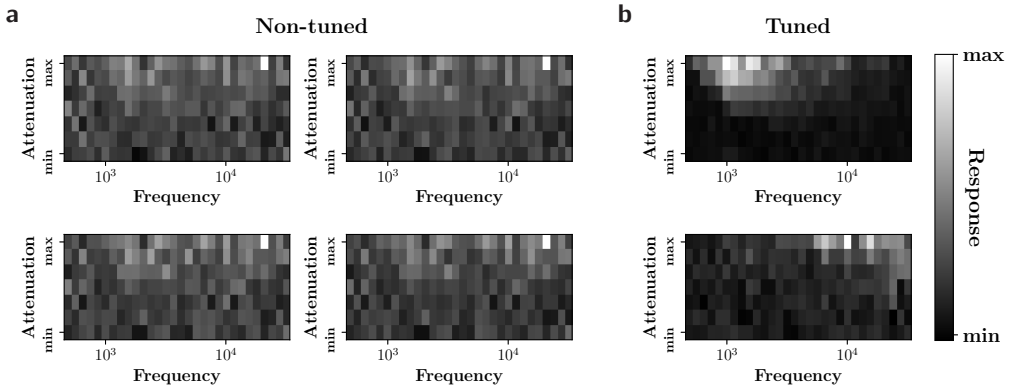


Figure 4.2: **Frequency response area (FRA) analysis of non-tuned electrodes, as determined by UoI, confirm that a frequency tuning model captures no discernible structure in their responses.** Each plot depicts the FRA, or the mean responses, across trials, to frequency-attenuation stimulus pairings. The plots are normalized to their maximum value. In each plot, the y-axis denote attenuation (ranging from -70 db to 0 db) while the x-axis denotes frequency (ranging from 500 Hz to 32 kHz). **a.** FRAs corresponding to the four electrodes that UoI determined to be non-tuned (pink points in Fig 4.6). **b.** FRAs for two randomly chosen tuned electrodes. The non-tuned electrodes exhibit no discernible structure in the FRAs (in contrast to the clear structure depicted by the tuned electrodes), validating that UoI correctly determined them to be non-tuned.

Conclusion

In this work, we demonstrated that a novel inference procedures can fit parametric models that are robust to correlated variability in the data. In particular, we created encoding and functional coupling models. A next iteration would be combining the models into a single model capturing both. However, doing so is difficult without capturing correlated variability at the same time. In the next chapter, we follow this line of work, and construct models capable of modeling both encoding and functional coupling.

Chapter 5

Identifying and mitigating statistical biases in neural models of tuning and functional coupling

Chapter Co-authors
Jesse A. Livezey
Sharmodeep Bhattacharyya
Kristofer E. Bouchard

We have already separately modeled encoding, which captures how external input drives neural activity, and functional coupling, which captures how internal input drives neural activity. How can we simultaneously model a neuron's dependence on external input and internal input given that we only record from a subset of the complete neural population? The unobserved neural activity – a source of correlated variability – will bias parameter estimates in simple phenomenological models. How do we account for correlated variability in such systems neuroscience models? In this chapter, we develop new models and inference procedures to answer these questions.

5.1 Introduction

Statistical models are a central tool in systems neuroscience for understanding neural activity [146, 99]. In particular, parametric models, such as generalized linear models, are appealing because the model parameters are interpreted to gain insight into the underlying neurobiological processes that generated the data [199, 144, 152, 188, 149]. For example, model parameters can represent external factors (e.g., stimuli or a behavioral task) and internal factors (e.g., other neurons). The fitted parameter values, therefore, specify which factors are important and how important they are. In many cases, this amounts to describing the

statistical nature of stimulus-neuron relationships (tuning: Fig. 5.1b) and neuron-neuron relationships (functional coupling: Fig. 5.1c) [185, 38, 184].

Assessing the degree to which these models actually capture the underlying neurobiological processes that generated the data is imperative to ensure that these models can serve scientific use beyond prediction. In particular, understanding what features these models fail to capture aids in constructing more complete models that are less prone to error. For example, tuning models neglect to consider how neighboring neurons impact the modeled neuron, while coupling models neglect to capture external drive. A tuning and coupling model (TC), which aims to remedy both these issues, combines both tuning and coupling into a single model (Fig. 5.1d). However, all these models neglect the fact that unobserved activity, not recorded by the experimental apparatus, will influence the observed neural activity (Fig. 5.1a).

Previous work has examined how the inclusion of functional coupling in a model modulates the magnitude of tuning in neuronal populations [188, 152, 185]. When a tuning and coupling model model is fitted to data, the ensuing tuning modulation has been observed to be downplayed compared to a tuning model alone. For example, we observe such "explaining away" in a population of neurons from macaque primary visual cortex during the presentation of drifting sinusoidal gratings (Fig. 5.2a) [105]. As shown by Figure 5.2b, neuronal activity is dependent on the angle of the grating. Tuning curves as a function of the drifting angle can be constructed using cosine basis functions (Fig. 5.2c: black curves). If we include functional coupling in the model, the tuning modulation, or min-to-max distance of the tuning curve, is reduced (Fig. 5.2c: gray curves).

Extracting conclusions about neural computation by interpreting the parameters of a fitted model requires unbiased parameter estimation, in addition to predictive power. On top of this, precise selection of the relevant, non-zero parameters necessarily impacts interpretability [31]. If structure in the data is not captured by the model, or the improper parameters are selected in the first place, the resulting parameter estimates may be sufficiently biased to jeopardize these conclusions. The TC model obtained different parameter fits relative to a tuning alone model, resulting in different neuroscientific conclusions. This "explaining away" effect can be viewed as a consequence of utilizing a more complete model relative to the tuning model alone [185]. Recent work has examined how model misspecification may introduce biases in models of neural activity [187, 55].

Thus, it is imperative to assess the degree to which models are misspecified, and the consequences for their interpretation. Just as the tuning model and coupling model are incomplete, the tuning and coupling model also suffers from misspecification. In particular, it omits two important features of the underlying neural activity. First, it neglects to consider that tuning may jointly impact both the observed neurons and the target neuron of interest. Second, as mentioned above, it neglects to consider that unobserved activity also jointly influence the observed neurons and the target neuron. Incorporating both of these features into a model is necessary to assess whether the TC model paints an accurate picture of the underlying neural computation.

In this work, we sought to understand whether a TC model accurately captures tuning

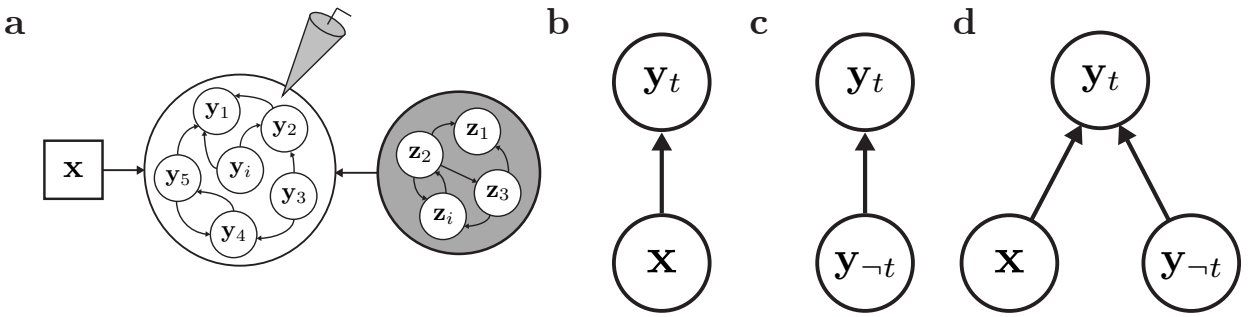


Figure 5.1: **Systems neuroscience models capture the impact of tuning and functional coupling on neural activity. a.** Neural datasets are comprised of recordings from observed neurons **y** that respond to an external stimulus **x**. However, the recording apparatus fails to capture unobserved activity **z**. **b.** Tuning model. **c.** Coupling model. **d.** Tuning and coupling model.

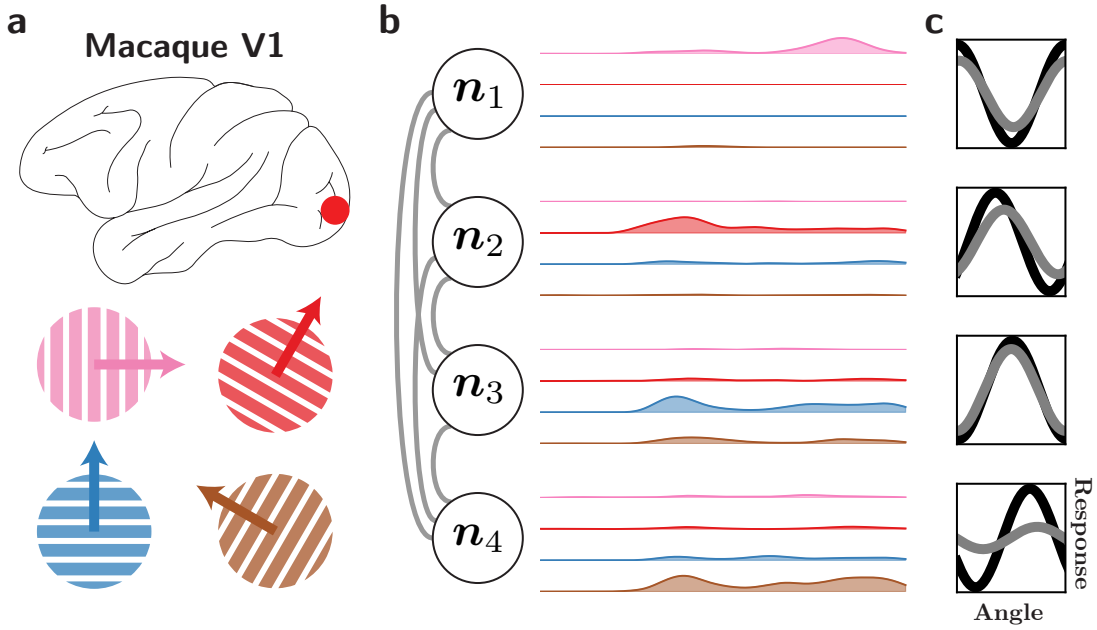


Figure 5.2: **Explaining away in a tuning and coupling model. a.** Single-unit recordings from macaque V1 during the presentation of drifting gratings. Colors denote four unique angles. **b.** Filtered firing rates for four example neurons in response to the four angles. **c.** Tuning curves, as a function of angle, for each neuron as fitted by a tuning model (black) and a TC model (gray).

and coupling parameters that reflect the data generation process. To do so, we first introduce the *triangular model*, a novel model of neural activity that is more complete than the TC model (Fig. 5.3). The triangular model allows stimulus information to flow through both a

tuning pathway and a coupling pathway. Additionally, the triangular model incorporates a latent space [203, 110, 148, 137] that jointly influences the entire neural population, thereby reproducing the phenomenon of correlated variability [11]. Lastly, posed as a graphical model, the triangular model allows the generation of synthetic data, allowing us to assess the degree to which the TC model sufficiently captures ground truth parameter values.

We demonstrated, using synthetic data generated from the triangular model, that the TC model suffers from the *simultaneous equations bias* due to the fact that it omits unobserved activity. Furthermore, we characterized that bias as underestimated tuning and overestimating coupling, implying that past observations of explaining may simply be a side effect of the simultaneous equations bias. Next, we develop inference procedures, using expectation-maximization, to fit the triangular model to data. We further demonstrate that the triangular model suffers from structural non-identifiability, demonstrate that sufficient sparsity in the model can mitigate this issue. We characterize the identifiability and loss surface of the triangular model. Lastly, we apply our inference procedure to neural data, and demonstrate the elevation of tuning modulations relative to the tuning and coupling model.

5.2 Methods

Triangular model definition

The triangular model is defined by the graphical model depicted in in Figure 5.3. The observed neural population consists of $N+1$ neurons $\mathbf{y} = [\mathbf{y}_{\neg t}, y_t]$, jointly influenced by an M-dimensional stimulus $\mathbf{x}$ and an latent K-dimensional population $\mathbf{z}$. Importantly, we make a distinction between a "target neuron" y_t and "non-target neurons" $\mathbf{y}_{\neg t}$. The latent state acts as a low-dimensional representation of the unobserved neurons in the neural population. Thus, the graphical model allows the target neuron to be influenced by a an external factor (the stimulus), an internal factor (the observed neurons), an unobserved internal factor (the latent state), while accounting for the fact that both the stimulus and unobserved activity jointly influence the observed neurons.

The main parameters of interest the N coupling parameters $\mathbf{a}$ and the M target tuning parameters, $\mathbf{b}_t$. These have direct analogues with the tuning and coupling model, and describe how the target neuron depends on the non-target neurons and stimulus, respectively. However, because the triangular model is more complete than the tuning and coupling model, it requires inferring additional parameters. These include the non-target parameters $\mathbf{B}_{\neg t}^{M \times N}$ and parameters describing how the latent state influences the observed neurons.

In this work, we operate in the linear-gaussian setting, where all relationships are linear, and the latent state operates in the gaussian settings. Thus, the graphical model in Figure 5.3

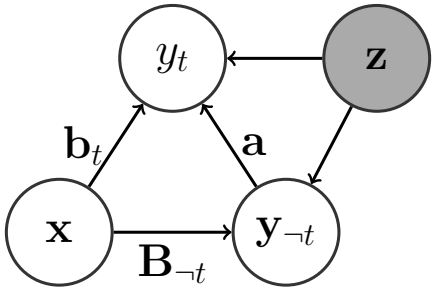


Figure 5.3: **Graphical model describing the triangular model.** The target neuron is denoted by y_t, non-target neurons as $\mathbf{y}_{\neg t}$, stimulus as $\mathbf{x}$, and latent state as $\mathbf{z}$. Latent state is shaded gray to emphasize that it is unobserved. The main parameters of interest are the target tuning parameters $\mathbf{b}_t$, coupling parameters $\mathbf{a}$, and non-target tuning parameters $\mathbf{B}_{\neg t}$.

can be written as

$$\begin{aligned}
\mathbf{y}_{\neg t} &= \mathbf{B}_{\neg t}^T \mathbf{x} + \boldsymbol{\epsilon}_{\neg t} && (5.1)\\
&= \mathbf{B}_{\neg t}^T \mathbf{x} + \mathbf{L}_{\neg t}^T \mathbf{z} + \boldsymbol{\psi}_{\neg t} && (5.2)\\
y_t &= \mathbf{b}_t^T \mathbf{x} + \mathbf{a}^T \mathbf{y}_{\neg t} + \epsilon_t && (5.3)\\
&= \mathbf{b}_t^T \mathbf{x} + \mathbf{a}^T \mathbf{y}_{\neg t} + \mathbf{l}_t^T \mathbf{z} + \psi_t. && (5.4)
\end{aligned}$$

Note that we have used additional terms, $\epsilon_{\epsilon t}$ and ϵ_t to refer to unobserved variability that cannot be captured by the observed parameters. Then, we rewrite these terms in terms of shared and private components in Equations (5.2) and (5.4), similar to a factor analysis model. Thus, the additional parameters include latent factors $\mathbf{L} = [\mathbf{L}_{\neg t}, \mathbf{l}_t]$ and private variances $\boldsymbol{\Psi} = [\boldsymbol{\Psi}_{\neg t}, \boldsymbol{\Psi}_t]$, which describe how the neural population depends on the unobserved influences.

When we have D data samples, we can write the data generation process across all samples as

$$\begin{aligned}
\mathbf{Y}_{\neg t} &= \mathbf{X}\mathbf{B}_{\neg t} + \mathbf{Z}\mathbf{L}_{\neg t} + \boldsymbol{\psi}_{\neg t} && (5.5)\\
\mathbf{y}_t &= \mathbf{X}\mathbf{b}_t + \mathbf{Y}_{\neg t}\mathbf{a} + \mathbf{Z}\mathbf{l}_t + \boldsymbol{\psi}_t && (5.6)
\end{aligned}$$

where instead the data is rewritten as $\mathbf{x}^{M\times 1} \to \mathbf{X}^{D\times M}$, $\mathbf{y}_{\neg t}^{N\times 1} \to \mathbf{Y}_{\neg t}^{D\times N}$, $y_t \to \mathbf{y}_t^{D\times 1}$, $\mathbf{z}^{K\times 1} \to \mathbf{Z}^{D\times K}$. We further note that the variability stems from a factor analysis model with latent factors $\mathbf{L} = [\mathbf{L}_{\neg t}, \mathbf{l}_t]$ and private variances $\boldsymbol{\Psi} = [\boldsymbol{\Psi}_{\neg t}, \Psi_t]$ which characterize the specific private noise terms on a trial $\boldsymbol{\psi} = [\boldsymbol{\psi}_{\neg t}, \psi_t]$. Thus, inference across all parameters in the model requires estimation of the set $\theta = [\mathbf{a}, \mathbf{b}_t, \mathbf{B}_{\neg t}, \mathbf{l}_t, \mathbf{L}_{\neg t}, \Psi_t, \boldsymbol{\Psi}_{\neg t}]$.

Parametric Inference in the Triangular Model

Since the triangular model is a latent variable model, we can perform parametric inference via the expectation-maximization algorithm. At the same time, the linear-gaussian instantiation

of the model lends itself well to analytic derivations of the joint and marginal distributions. Here, we calculate these distributions as pre-requisites for deriving the EM-update rules for optimization.

Joint distribution of the neural activity and latent state

A full likelihood expression of the triangular model incorporates the parameters $\mathbf{L} = [\mathbf{l}_t, \mathbf{L}_{\neg t}]$ and $\mathbf{\Psi} = [\Psi_t, \mathbf{\Psi}_{\neg t}]$ that define the shared and private variability. Recall that the neural activities are defined as

$$y_t = \mathbf{x}^T\mathbf{b}_t + \mathbf{y}_{\neg t}^T\mathbf{a} + \mathbf{z}^T\mathbf{l}_t + \psi_t \tag{5.7}$$

$$\mathbf{y}_{\neg t} = \mathbf{B}_{\neg t}^T\mathbf{x} + \mathbf{L}_{\neg t}^T\mathbf{z} + \boldsymbol{\psi}_{\neg t}. \tag{5.8}$$

The joint distribution of the data, including the latent variables, can be written as

$$p(y_t, \mathbf{y}_{\neg t}, \mathbf{x}, \mathbf{z}; \theta) = p(y_t|\mathbf{y}_{\neg t}, \mathbf{x}, \mathbf{z}; \theta)p(\mathbf{y}_{\neg t}|\mathbf{x}, \mathbf{z}; \theta)p(\mathbf{x})p(\mathbf{z}) \tag{5.9}$$

where θ specifies the parameter set. In the linear-gaussian setting, each of these densities takes on the form

$$p(y_t|\mathbf{y}_{\neg t}, \mathbf{x}, \mathbf{z}; \theta) \sim \mathcal{N}(\mathbf{x}^T\mathbf{b}_t + \mathbf{y}_{\neg t}^T\mathbf{a} + \mathbf{z}^T\mathbf{l}_t, \Psi_t) \tag{5.10}$$

$$p(\mathbf{y}_{\neg t}|\mathbf{x}, \mathbf{z}; \theta) \sim \mathcal{N}(\mathbf{B}_{\neg t}^T\mathbf{x} + \mathbf{L}_{\neg t}^T\mathbf{z}, \mathbf{\Pi}_{\neg t}) \tag{5.11}$$

$$p(\mathbf{z}) \sim \mathcal{N}(0, \mathbf{I}), \tag{5.12}$$

where $\mathbf{\Pi}_{\neg t} \coloneqq \text{diag}(\mathbf{\Psi}_{\neg t})$. By the gaussianity of the above distributions, we can write the joint distribution of y_t, $\mathbf{y}_{\neg t}$, and $\mathbf{z}$ (conditioned on $\mathbf{x}$) as a multivariate Gaussian distribution. Specifically, we have

$$\begin{pmatrix} y_t \\ \mathbf{y}_{\neg t} \\ \mathbf{z} \end{pmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma}) \tag{5.13}$$

where

$$\boldsymbol{\mu} = \begin{pmatrix} \mathbf{x}^T\mathbf{b}_t + \mathbf{x}^T\mathbf{B}_{\neg t}\mathbf{a} \\ \mathbf{B}_{\neg t}^T\mathbf{x} \\ \mathbf{0} \end{pmatrix}, \tag{5.14}$$

$$\mathbf{\Sigma} = \begin{pmatrix} \Psi_t + \mathbf{a}^T\mathbf{\Pi}_{\neg t}\mathbf{a} + (\mathbf{l}_t + \mathbf{L}_{\neg t}\mathbf{a})^T(\mathbf{l}_t + \mathbf{L}_{\neg t}\mathbf{a}) & \mathbf{a}^T\mathbf{\Pi}_{\neg t} + (\mathbf{l}_t + \mathbf{L}_{\neg t}\mathbf{a})^T\mathbf{L}_{\neg t} & (\mathbf{l}_t + \mathbf{L}_{\neg t}\mathbf{a})^T \\ \mathbf{\Pi}_{\neg t}\mathbf{a} + \mathbf{L}_{\neg t}^T(\mathbf{l}_t + \mathbf{L}_{\neg t}\mathbf{a}) & \mathbf{\Pi}_{\neg t} + \mathbf{L}_{\neg t}^T\mathbf{L}_{\neg t} & \mathbf{L}_{\neg t}^T \\ \mathbf{l}_t + \mathbf{L}_{\neg t}\mathbf{a} & \mathbf{L}_{\neg t} & \mathbf{I} \end{pmatrix} \tag{5.15}$$

and an associated precision matrix with analytic form given by

$$\mathbf{\Sigma}^{-1} = \begin{pmatrix} \Psi_t^{-1} & -\Psi_t^{-1}\mathbf{a}^T & -\Psi_t^{-1}\mathbf{l}_t^T \\ -\Psi_t^{-1}\mathbf{a} & \mathbf{\Pi}_{\neg t}^{-1} + \Psi_t^{-1}\mathbf{a}\mathbf{a}^T & \Psi_t^{-1}\mathbf{a}\mathbf{l}_t^T - \mathbf{\Pi}_{\neg t}^{-1}\mathbf{L}_{\neg t}^T \\ -\Psi_t\mathbf{l}_t & \Psi_t^{-1}\mathbf{l}_t\mathbf{a} - \mathbf{L}_{\neg t}\mathbf{\Pi}_{\neg t}^{-1} & \mathbf{I} + \Psi_t^{-1}\mathbf{l}_t\mathbf{l}_t^T + \mathbf{L}_{\neg t}\mathbf{\Pi}_{\neg t}^{-1}\mathbf{L}_{\neg t}^T \end{pmatrix}. \tag{5.16}$$

Since we have the complete joint distribution, we can easily extract marginals of the neural activity by taking the corresponding blocks of the mean and covariance matrices.

Maximum likelihood via expectation-maximization

The triangular model is a latent state model. Thus, parameter inference can be achieved by performing expectation-maximization (EM). In this section, we derive the update rules for EM optimization. To do so, we first must determine the complete log-likelihood. From this, we derive the E-step, followed by the M-step.

Complete likelihood

Using the joint distribution calculated above, we can write the log-likelihood over all random variables as

$$\ell_c(y_t, \mathbf{y}_{\neg t}, \mathbf{x}, \mathbf{z}; \theta) = -\frac{1}{2}\sum_{d=1}^{D}\Bigg[\log \Psi_t + \frac{1}{\Psi_t}\left(y_t^{(d)} - \mathbf{x}^{(d)^T}\mathbf{b}_t - \mathbf{y}_{\neg t}^{(d)^T}\mathbf{a} - \mathbf{z}^T\mathbf{l}_t\right)^2 + \log\det\mathbf{\Pi}_{\neg t} + \left(\mathbf{y}_{\neg t}^{(d)} - \mathbf{B}_{\neg t}^T\mathbf{x}^{(d)} - \mathbf{L}_{\neg t}^T\mathbf{z}\right)^T\mathbf{\Pi}_{\neg t}^{-1}\left(\mathbf{y}_{\neg t}^{(d)} - \mathbf{B}_{\neg t}^T\mathbf{x}^{(d)} - \mathbf{L}_{\neg t}^T\mathbf{z}\right)\Bigg] - \frac{1}{2}\mathbf{z}^T\mathbf{z} + \log p(\mathbf{x}^{(d)}). \tag{5.17}$$

In general we will ignore the contribution from the density of $\mathbf{x}$ since it is observed and has no parents in the graphical model.

E-step update

To perform the E-step, we need to calculate the averaging distribution $q(\mathbf{z}|\mathcal{D}^{(d)};\theta)$ with a dataset $\mathcal{D}^{(d)} = \left(\mathbf{x}^{(d)}, \mathbf{y}_{\neg t}^{(d)}, y_t^{(d)}\right)$. Note that

$$\begin{aligned} q(\mathbf{z}|\mathcal{D}^{(d)};\theta) &= p(\mathbf{z}|\mathcal{D}^{(d)};\theta) \\ &\propto p(\mathbf{x}^{(d)}, \mathbf{y}_{\neg t}^{(d)}, y_t^{(d)}|\mathbf{z};\theta)p(\mathbf{z}) \qquad (5.18) \\ &= p(y_t^{(d)}|\mathbf{y}_{\neg t}^{(d)}, \mathbf{x}^{(d)}, \mathbf{z};\theta)p(\mathbf{y}_{\neg t}^{(d)}|\mathbf{x}^{(d)}, \mathbf{z};\theta)p(\mathbf{x}^{(d)})p(\mathbf{z}). \qquad (5.19) \end{aligned}$$

Ultimately, this expression can be written as a Gaussian in $\mathbf{z}$ with mean $\boldsymbol{\mu}^{(d)}$ and covariance $\mathbf{\Sigma}$, i.e.

$$q(\mathbf{z}|\mathcal{D}^{(d)};\theta) \propto \exp\left(-\frac{1}{2}[\mathbf{z} - \boldsymbol{\mu}^{(d)}]^T\mathbf{\Sigma}^{-1}[\mathbf{z} - \boldsymbol{\mu}^{(d)}]\right). \tag{5.20}$$

Collecting the quadratic terms gives us the inverse covariance matrix:

$$\begin{aligned} \mathbf{z}^T\mathbf{\Sigma}^{-1}\mathbf{z} &= \mathbf{z}^T\mathbf{z} + \mathbf{z}^T\mathbf{l}_t\Psi_t^{-1}\mathbf{l}_t^T\mathbf{z} + \mathbf{z}^T\mathbf{L}_{\neg t}\mathbf{\Pi}_{\neg t}^{-1}\mathbf{L}_{\neg t}^T\mathbf{z} \qquad (5.21) \\ \Rightarrow \mathbf{\Sigma}^{-1} &= \mathbf{I} + \mathbf{L}\mathbf{\Pi}^{-1}\mathbf{L}^T. \qquad (5.22) \end{aligned}$$

Next, we examine all the linear terms in $\mathbf{z}$:

$$\mathbf{z}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}^{(d)} = \mathbf{z}^T\mathbf{l}_t\Psi_t^{-1}(y_t^{(d)} - \mathbf{x}^{(d)^T}\mathbf{b}_t - \mathbf{y}_{\neg t}^{(d)^T}\mathbf{a}) + \mathbf{z}^T\mathbf{L}_{\neg t}\boldsymbol{\Pi}_{\neg t}^{-1}(\mathbf{y}_{\neg t}^{(d)} - \mathbf{B}_{\neg t}^T\mathbf{x}^{(d)}) \tag{5.23}$$

$$\Rightarrow \boldsymbol{\mu}^{(d)} = \boldsymbol{\Sigma}\left[\mathbf{l}_t\Psi_t^{-1}(y_t^{(d)} - \mathbf{x}^{(d)^T}\mathbf{b}_t - \mathbf{y}_{\neg t}^{(d)^T}\mathbf{a}) + \mathbf{L}_{\neg t}\boldsymbol{\Pi}_{\neg t}^{-1}(\mathbf{y}_{\neg t}^{(d)} - \mathbf{B}_{\neg t}^T\mathbf{x}^{(d)})\right]. \tag{5.24}$$

The statistics of the unobserved variables, given by

$$\langle\mathbf{z}\rangle_{q^{(d)}} = \boldsymbol{\mu}^{(d)} \tag{5.25}$$

$$\langle\mathbf{z}\mathbf{z}^T\rangle_{q^{(d)}} = \boldsymbol{\Sigma} + \boldsymbol{\mu}^{(d)}\boldsymbol{\mu}^{(d)^T} \tag{5.26}$$

will become relevant in the M-step.

M-step update

To calculate the M-step, we take the expectation of the complete log-likelihood over the averaging distribution $q(\mathbf{z}|\mathcal{D})$. Note that we ignore the prior distribution for $\mathbf{x}$ as it will not be relevant for any gradients. The expected complete log-likelihood is given by

$$\begin{aligned}
\langle\ell_c(y_t, \mathbf{y}_{\neg t}, \mathbf{x}, \mathbf{z}; \theta)\rangle = -\frac{1}{2}\sum_{d=1}^{D}\Bigg[&\log\det\boldsymbol{\Pi} + \frac{1}{\Psi_t}\left\langle\left(y_t^{(d)} - \mathbf{x}^{(d)^T}\mathbf{b}_t - \mathbf{y}_{\neg t}^{(d)^T}\mathbf{a} - \mathbf{z}^T\mathbf{l}_t\right)^2\right\rangle_{q^{(d)}} \\
&\left\langle\left(\mathbf{y}_{\neg t}^{(d)} - \mathbf{B}_{\neg t}^T\mathbf{x}^{(d)} - \mathbf{L}_{\neg t}^T\mathbf{z}\right)^T\boldsymbol{\Pi}_{\neg t}^{-1}\left(\mathbf{y}_{\neg t}^{(d)} - \mathbf{B}_{\neg t}^T\mathbf{x}^{(d)} - \mathbf{L}_{\neg t}^T\mathbf{z}\right)\right\rangle_{q^{(d)}}\Bigg] \\
= -\frac{1}{2}\sum_{d=1}^{D}\Bigg[&\log\det\boldsymbol{\Pi} + \frac{1}{\Psi_t}\left(y_t^{(d)} - \mathbf{x}^{(d)^T}\mathbf{b}_t - \mathbf{y}_{\neg t}^{(d)^T}\mathbf{a}\right)^2 \\
&-\frac{2}{\Psi_t}\left(y_t^{(d)} - \mathbf{x}^{(d)^T}\mathbf{b}_t - \mathbf{y}_{\neg t}^{(d)^T}\mathbf{a}\right)\mathbf{l}_t^T\langle\mathbf{z}\rangle_q \qquad (5.27) \\
&+\frac{1}{\Psi_t}\mathbf{l}_t^T\langle\mathbf{z}\mathbf{z}^T\rangle_q\mathbf{l}_t + \left(\mathbf{y}_{\neg t}^{(d)} - \mathbf{B}_{\neg t}^T\mathbf{x}^{(d)}\right)^T\boldsymbol{\Pi}_{\neg t}^{-1}\left(\mathbf{y}_{\neg t}^{(d)} - \mathbf{B}_{\neg t}^T\mathbf{x}^{(d)}\right) \\
&-2\left(\mathbf{y}_{\neg t}^{(d)} - \mathbf{B}_{\neg t}^T\mathbf{x}^{(d)}\right)^T\boldsymbol{\Pi}_{\neg t}^{-1}\mathbf{L}_{\neg t}^T\langle\mathbf{z}\rangle_q + \langle\mathbf{z}^T\mathbf{L}_{\neg t}\boldsymbol{\Pi}_{\neg t}^{-1}\mathbf{L}_{\neg t}^T\mathbf{z}\rangle_q\Bigg]. \qquad (5.28)
\end{aligned}$$

Note that the last expectation can be written as

$$\langle\mathbf{z}^T\mathbf{L}_{\neg t}\boldsymbol{\Pi}_{\neg t}^{-1}\mathbf{L}_{\neg t}^T\mathbf{z}\rangle_q = \mathrm{Tr}\left[\mathbf{L}_{\neg t}\boldsymbol{\Pi}_{\neg t}^{-1}\mathbf{L}_{\neg t}^T\boldsymbol{\Sigma}\right] + \langle\mathbf{z}\rangle_q^T\mathbf{L}_{\neg t}\boldsymbol{\Pi}_{\neg t}^{-1}\mathbf{L}_{\neg t}^T\langle\mathbf{z}\rangle_q \tag{5.29}$$

Intercepts and standardizing

In practice, we often want to include an intercept term in our models. With an intercept term, the triangular model becomes:

$$y_t = b_{0,t} + \mathbf{x}^T\mathbf{b}_t + \mathbf{y}_{\neg t}^T\mathbf{a} + \mathbf{z}^T\mathbf{l}_t + \psi_t \tag{5.30}$$

$$\mathbf{y}_{\neg t} = \mathbf{b}_{0,\neg t} + \mathbf{B}_{\neg t}^T\mathbf{x} + \mathbf{L}_{\neg t}^T\mathbf{z} + \boldsymbol{\psi}_{\neg t}. \tag{5.31}$$

In this case, the marginal distribution (conditioned on the stimulus $\mathbf{x}$) of the neural activity is given by

$$\mathbf{y} = \begin{pmatrix} y_t \\ \mathbf{y}_{\neg t} \end{pmatrix} \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y) \tag{5.32}$$

where

$$\boldsymbol{\mu}_y = \begin{pmatrix} b_{0,t} + \mathbf{a}^T \mathbf{b}_{0,\neg t} + \mathbf{x}^T \mathbf{b}_t + \mathbf{x}^T \mathbf{B}_{\neg t} \mathbf{a} \\ \mathbf{b}_{0,\neg t} + \mathbf{B}_{\neg t}^T \mathbf{x} \end{pmatrix} \tag{5.33}$$

and $\boldsymbol{\Sigma}_y$ is unchanged. This gives log-likelihood:

$$\ell(y_t, \mathbf{y}_{\neg t}; \mathbf{x}, \theta) = -\frac{1}{2} \sum_{d=1}^{D} \left[\log \det \boldsymbol{\Sigma}_y + (\mathbf{y}^{(d)} - \boldsymbol{\mu}_y)^T \boldsymbol{\Sigma}_y^{-1} (\mathbf{y}^{(d)} - \boldsymbol{\mu}_y) \right] \tag{5.34}$$

Taking the gradient of the marginal log-likelihood with respect to the intercept terms gives

$$\frac{1}{D} \sum_{d=1}^{D} y_t - (b_{0,t} + \hat{\mathbf{a}}^T \mathbf{b}_{0,\neg t} + \mathbf{x}^T \mathbf{b}_t + \mathbf{x}^T \mathbf{B}_{\neg t} \mathbf{a}) = 0 \tag{5.35}$$

$$\Rightarrow b_{0,t} + \mathbf{a}^T \mathbf{b}_{0,\neg t} = \bar{y} - \bar{\mathbf{x}}^T \hat{\mathbf{b}}_t - \bar{\mathbf{x}}^T \hat{\mathbf{B}}_{\neg t} \hat{\mathbf{a}} \tag{5.36}$$

and

$$\sum_{d=1}^{D} \mathbf{y}_{\neg t} - \mathbf{b}_{0,\neg t} - \mathbf{B}_{\neg t}^T \mathbf{x} = 0 \tag{5.37}$$

$$\Rightarrow \hat{\mathbf{b}}_{0,\neg t} = \bar{\mathbf{y}}_{\neg t} - \hat{\mathbf{B}}_{\neg t}^T \bar{\mathbf{x}}. \tag{5.38}$$

implying that

$$b_{0,t} = \bar{y} - \bar{\mathbf{x}}^T \hat{\mathbf{b}}_t - \bar{\mathbf{x}}^T \hat{\mathbf{B}}_{\neg t} \hat{\mathbf{a}} - \hat{\mathbf{a}}^T \left(\bar{\mathbf{y}}_{\neg t} - \hat{\mathbf{B}}_{\neg t}^T \bar{\mathbf{x}} \right) \tag{5.39}$$

$$= \bar{y} - \bar{\mathbf{x}}^T \hat{\mathbf{b}}_t - \bar{\mathbf{y}}_{\neg t}^T \hat{\mathbf{a}} \tag{5.40}$$

Note that if we center the inputs, i.e. $(y_t, \mathbf{y}_{\neg t}, \mathbf{x}) \rightarrow (y - \bar{y}_t, \mathbf{y}_{\neg t} - \bar{\mathbf{y}}_{\neg t}, \mathbf{x} - \bar{\mathbf{x}}) = (y'_t, \mathbf{y}'_{\neg t}, \mathbf{x}')$, we would find that the intercepts are zero. Thus, performing triangular model inference on the data $(y'_t, \mathbf{y}'_{\neg t}, \mathbf{x}')$ requires that we need not fit an intercept. However, we must transform back to the non-centered space. In such a case, we have

$$\mathbf{y'}_{\neg t} = 0 + \mathbf{B'}_{\neg t}^T \mathbf{x}' + \mathbf{L}_{\neg t}^T \mathbf{z} + \boldsymbol{\psi}_{\neg t}. \tag{5.41}$$

$$\Rightarrow \mathbf{y}_{\neg t} - \bar{\mathbf{y}}_{\neg t} = \mathbf{B'}_{\neg t}^T (\mathbf{x} - \bar{\mathbf{x}}) + \mathbf{L}_{\neg t}^T \mathbf{z} + \boldsymbol{\psi}_{\neg t} \tag{5.42}$$

$$\Rightarrow \mathbf{y}_{\neg t} = \left(\bar{\mathbf{y}}_{\neg t} - \hat{\mathbf{B}}_{\neg t}^T \bar{\mathbf{x}} \right) + \mathbf{B'}_{\neg t}^{\ T} \mathbf{x} + \mathbf{L}_{\neg t}^T \mathbf{z} + \boldsymbol{\psi}_{\neg t}. \tag{5.43}$$

Thus, we have the same intercept formula as above, and $\mathbf{B}_{\neg t} = \mathbf{B}'_{\neg t}$. Similarly,

$$y_t - \bar{y}_t = 0 + (\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{b}'_t + (\mathbf{y}_{\neg t} - \bar{\mathbf{y}}_{\neg t})^T \mathbf{a}' + \mathbf{z}^T \mathbf{l}_t + \psi_t \tag{5.44}$$

$$\Rightarrow y_t = \left(\bar{y}_t - \bar{\mathbf{x}}^T \mathbf{b}'_t - \bar{\mathbf{y}}_{\neg t}^T \mathbf{a}'\right) + \mathbf{x}^T \mathbf{b}'_t + \mathbf{y}_{\neg t}^T \mathbf{a}' + \mathbf{z}^T \mathbf{l}_t + \psi_t \tag{5.45}$$

Implying that $b_{0,t} = \bar{y}_t - \bar{\mathbf{x}}^T \mathbf{b}'_t - \bar{\mathbf{y}}_{\neg t}^T \mathbf{a}'$, as above.

If we standardize the data in addition to centering, then we have the following:

$$\frac{\mathbf{y}_{\neg t} - \bar{\mathbf{y}}_{\neg t}}{s_y} = \mathbf{B}'^T_{\neg t} \left(\frac{\mathbf{x} - \bar{\mathbf{x}}}{s_x}\right) + \mathbf{L}_{\neg t}^T \mathbf{z} + \boldsymbol{\psi}_{\neg t} \tag{5.46}$$

$$\Rightarrow \mathbf{y}_{\neg t} = \left(s_x \bar{\mathbf{y}}_{\neg t} - \frac{s_y}{s_x} \hat{\mathbf{B}}_{\neg t}^T \bar{\mathbf{x}}\right) + \left(\frac{s_y}{s_x} \mathbf{B}'_{\neg t}\right)^T \mathbf{x} + \left(\frac{s_y}{s_x} \mathbf{L}_{\neg t}\right)^T \mathbf{z} + \frac{s_y}{s_x} \boldsymbol{\psi}_{\neg t}. \tag{5.47}$$

Structural non-identifiability in the triangular model

In this section, we show that the triangular model is structurally non-identifiable. Furthermore, we prove that sufficient sparsity in the triangular model remedies the non-identifiability.

Deriving the identifiability subspace

Recall that the marginal log-likelihood is given by

$$\ell(y_t, \mathbf{y}_{\neg t}; \mathbf{x}, \theta) = -\frac{1}{2} \sum_{d=1}^{D} \left[\log\det \boldsymbol{\Sigma}_y + (\mathbf{y}^{(d)} - \boldsymbol{\mu}_y)^T \boldsymbol{\Sigma}_y^{-1} (\mathbf{y}^{(d)} - \boldsymbol{\mu}_y) \right] \tag{5.48}$$

$$= -\frac{D}{2} \log\det \boldsymbol{\Sigma}_y - \frac{1}{2} \operatorname{Tr}\left[\mathbf{R}_y \boldsymbol{\Sigma}_y^{-1} \mathbf{R}_y^T\right] \tag{5.49}$$

where

$$\boldsymbol{\mu}_y = \begin{pmatrix} \mathbf{x}^T \mathbf{b}_t + \mathbf{x}^T \mathbf{B}_{\neg t} \mathbf{a} \\ \mathbf{B}_{\neg t}^T \mathbf{x} \end{pmatrix}, \tag{5.50}$$

$$\boldsymbol{\Sigma}_y = \begin{pmatrix} \Psi_t + \mathbf{a}^T \boldsymbol{\Pi}_{\neg t} \mathbf{a} + (\mathbf{l}_t + \mathbf{L}_{\neg t}\mathbf{a})^T (\mathbf{l}_t + \mathbf{L}_{\neg t}\mathbf{a}) & \mathbf{a}^T \boldsymbol{\Pi}_{\neg t} + (\mathbf{l}_t + \mathbf{L}_{\neg t}\mathbf{a})^T \mathbf{L}_{\neg t} \\ \boldsymbol{\Pi}_{\neg t} \mathbf{a} + \mathbf{L}_{\neg t}^T (\mathbf{l}_t + \mathbf{L}_{\neg t}\mathbf{a}) & \boldsymbol{\Pi}_{\neg t} + \mathbf{L}_{\neg t}^T \mathbf{L}_{\neg t} \end{pmatrix}. \tag{5.51}$$

Here, we consider an identifiability issue in the "separable" sense, i.e. offsets that can be applied to the parameters such that the mean $\boldsymbol{\mu}_y$ and covariance $\boldsymbol{\Sigma}_y$ are separately unchanged. If these quantities remain unchanged, the log-likelihood will necessarily be unchanged. The "separable" case is in contrast to the scenario in which we change the parameters such that the log-likelihood is unchanged, but through its overall computation (rather than due to the mean and covariance remaining unchanged).

We apply offsets to $\mathbf{l}_t$, Ψ_t, $\mathbf{a}$, and $\mathbf{b}_t$. Specifically, suppose we apply some offset $\boldsymbol{\delta}$ to $\mathbf{l}_t$:

$$\mathbf{l}'_t \leftarrow \mathbf{l}_t + \boldsymbol{\delta} \tag{5.52}$$

and define the quantity

$$\boldsymbol{\Delta} = -(\boldsymbol{\Pi}_{\neg t} + \mathbf{L}_{\neg t}^T \mathbf{L}_{\neg t})^{-1} \mathbf{L}_{\neg t}^T \boldsymbol{\delta}. \tag{5.53}$$

Then, we apply the following transformations to $\Psi_t, \mathbf{a}$ and $\mathbf{b}_t$:

$$\mathbf{a}' \leftarrow \mathbf{a} + \boldsymbol{\Delta} \tag{5.54}$$

$$\mathbf{b}_t' \leftarrow \mathbf{b}_t - \mathbf{B}_{\neg t} \boldsymbol{\Delta} \tag{5.55}$$

$$\begin{aligned} \Psi_t' \leftarrow \Psi_t &- 2\boldsymbol{\Delta}^T \boldsymbol{\Pi}_{\neg t} \mathbf{a} - \boldsymbol{\Delta}^T \boldsymbol{\Pi}_{\neg t} \boldsymbol{\Delta} \\ &- (\boldsymbol{\delta} + \mathbf{L}_{\neg t} \boldsymbol{\Delta})^T (\boldsymbol{\delta} + \mathbf{L}_{\neg t} \boldsymbol{\Delta}) - 2(\mathbf{l}_t + \mathbf{L}_{\neg t} \mathbf{a})^T (\boldsymbol{\delta} + \mathbf{L}_{\neg t} \boldsymbol{\Delta}) \end{aligned} \tag{5.56}$$

The bottom block of the mean is necessarily unchanged, since we have not modified $\mathbf{B}_{\neg t}$. Meanwhile, the top block, with the new parameter configuration, becomes

$$\begin{aligned} \mathbf{x}^T (\mathbf{b}_t' + \mathbf{B}_{\neg t} \mathbf{a}') &= \mathbf{x}^T (\mathbf{b}_t - \mathbf{B}_{\neg t} \boldsymbol{\Delta} + \mathbf{B}_{\neg t} (\mathbf{a} + \Delta)) && (5.57) \\ &= \mathbf{x}^T \mathbf{b}_t + \mathbf{x}^T \mathbf{B}_{\neg t} \mathbf{a}. && (5.58) \end{aligned}$$

Thus, $\boldsymbol{\mu}_y$ is unchanged with the new parameter configuration. In the covariance matrix $\boldsymbol{\Sigma}_y$, the bottom right quadrant is necessarily unchanged, since we do not modify any of the constituent parameters. The bottom left component (which is identical in content to the top right component), under the new configuration, is given by

$$\begin{aligned} \boldsymbol{\Pi}_{\neg t} \mathbf{a}' + \mathbf{L}_{\neg t}^T (\mathbf{l}_t' + \mathbf{L}_{\neg t} \mathbf{a}') &= \boldsymbol{\Pi}_{\neg t} (\mathbf{a} + \boldsymbol{\Delta}) + \mathbf{L}_{\neg t}^T ((\mathbf{l}_t + \boldsymbol{\delta}) + \mathbf{L}_{\neg t} (\mathbf{a} + \boldsymbol{\Delta})) && (5.59) \\ &= \boldsymbol{\Pi}_{\neg t} \mathbf{a} + \mathbf{L}_{\neg t}^T (\mathbf{l}_t + \mathbf{L}_{\neg t} \mathbf{a}) + \boldsymbol{\Pi}_{\neg t} \boldsymbol{\Delta} + \mathbf{L}_{\neg t}^T (\boldsymbol{\delta} + \mathbf{L}_{\neg t} \boldsymbol{\Delta}) && (5.60) \\ &= \boldsymbol{\Pi}_{\neg t} \mathbf{a} + \mathbf{L}_{\neg t}^T (\mathbf{l}_t + \mathbf{L}_{\neg t} \mathbf{a}) + \left(\boldsymbol{\Pi}_{\neg t} + \mathbf{L}_{\neg t}^T \mathbf{L}_{\neg t}\right) \boldsymbol{\Delta} + \mathbf{L}_{\neg t}^T \boldsymbol{\delta} && (5.61) \\ &= \boldsymbol{\Pi}_{\neg t} \mathbf{a} + \mathbf{L}_{\neg t}^T (\mathbf{l}_t + \mathbf{L}_{\neg t} \mathbf{a}) - \mathbf{L}_{\neg t}^T \boldsymbol{\delta} + \mathbf{L}_{\neg t}^T \boldsymbol{\delta} && (5.62) \\ &= \boldsymbol{\Pi}_{\neg t} \mathbf{a} + \mathbf{L}_{\neg t}^T (\mathbf{l}_t + \mathbf{L}_{\neg t} \mathbf{a}) && (5.63) \end{aligned}$$

and so is unchanged. Lastly, we consider the top-left component of the covariance matrix:

$$\begin{aligned} \Psi_i' + \mathbf{a}'^T \boldsymbol{\Pi}_{\neg t} \mathbf{a}' + (\mathbf{l}_t' + \mathbf{L}_{\neg t} \mathbf{a}')^T (\mathbf{l}_t' + \mathbf{L}_{\neg t} \mathbf{a}') &= \Psi_t' + (\mathbf{a} + \boldsymbol{\Delta})^T \boldsymbol{\Pi}_{\neg t} (\mathbf{a} + \boldsymbol{\Delta}) \\ &+ \left[(\mathbf{l}_t + \boldsymbol{\delta}) + \mathbf{L}_{\neg t} (\mathbf{a} + \boldsymbol{\Delta})\right]^T \left[(\mathbf{l}_t + \boldsymbol{\delta}) + \mathbf{L}_{\neg t} (\mathbf{a} + \boldsymbol{\Delta})\right] && (5.64) \\ &= \Psi_t' + \mathbf{a}^T \boldsymbol{\Pi}_{\neg t} \mathbf{a} + 2\boldsymbol{\Delta}^T \boldsymbol{\Pi}_{\neg t} \mathbf{a} + \boldsymbol{\Delta}^T \boldsymbol{\Pi}_{\neg t} \boldsymbol{\Delta} \\ + (\mathbf{l}_t + \mathbf{L}_{\neg t} \mathbf{a})^T (\mathbf{l}_t + \mathbf{L}_{\neg t} \mathbf{a}) + (\boldsymbol{\delta} + \mathbf{L}_{\neg t} \boldsymbol{\Delta})^T (\boldsymbol{\delta} + \mathbf{L}_{\neg t} \boldsymbol{\Delta}) &+ 2(\mathbf{l}_t + \mathbf{L}_{\neg t} \mathbf{a})^T (\boldsymbol{\delta} + \mathbf{L}_{\neg t} \boldsymbol{\Delta}) && (5.65) \\ &= \Psi_t + \mathbf{a}^T \boldsymbol{\Pi}_{\neg t} \mathbf{a} + (\mathbf{l}_t + \mathbf{L}_{\neg t} \mathbf{a})^T (\mathbf{l}_t + \mathbf{L}_{\neg t} \mathbf{a}). && (5.66) \end{aligned}$$

Thus, the variance for the target neuron is unchanged. These offsets specify a family of solutions, given a $\mathbf{B}_{\neg t}$, $\mathbf{L}_{\neg t}$, and $\boldsymbol{\Pi}_{\neg t}$. Importantly, however, this family is restricted to where Ψ_t is positive.

Model sparsity sufficiently constrains identifiability

In this section, we determine under what conditions having sparse support in $\mathbf{a}$ and $\mathbf{b}$ removes enough degrees of freedom in the identifiability subspace to have a unique solution. We aim to show the following: given a procedure for sparse estimation of $\mathbf{a}$ and $\mathbf{b}$, there is a required level of sparsity in $\mathbf{a}$ and $\mathbf{b}$ (and an additional condition on the rank of a matrix) such that any identifiability transform modifies the support of $\mathbf{a}$ and $\mathbf{b}$ and so support-preserving estimation is unique.

Theorem 1. *Consider an identifiability transformation of the tuning and coupling parameters:*

$$\begin{aligned}
\mathbf{a}' &= \mathbf{a} + \boldsymbol{\Delta} && (5.67)\\
&= \mathbf{a} - (\boldsymbol{\Pi}_{\neg t} + \mathbf{L}_{\neg t}^T \mathbf{L}_{\neg t})^{-1} \mathbf{L}_{\neg t}^T \boldsymbol{\delta} && (5.68)\\
\mathbf{b}' &= \mathbf{b}_t - \mathbf{B}_{\neg t} \boldsymbol{\Delta} && (5.69)\\
&= \mathbf{b}_t - \mathbf{B}_{\neg t} (\boldsymbol{\Pi}_{\neg t} + \mathbf{L}_{\neg t}^T \mathbf{L}_{\neg t})^{-1} \mathbf{L}_{\neg t}^T \boldsymbol{\delta} && (5.70)
\end{aligned}$$

where there are N coupling parameters in $\mathbf{a}$, M tuning parameters in $\mathbf{b}$, and K latent factors. $\boldsymbol{\Pi}_{\neg t}$, $\mathbf{L}_{\neg t}$, and $\mathbf{B}_{\neg t}$ are fixed. Let k_C be the sparsity of $\mathbf{a}$ so that Nk_C parameters are exactly zero. Similarly, let k_T be the sparsity of $\mathbf{b}_t$ so that Mk_T are exactly zero. Let $\mathbf{P} = (\boldsymbol{\Pi}_{\neg t} + \mathbf{L}_{\neg t}^T \mathbf{L}_{\neg t})^{-1} \mathbf{L}_{\neg t}^T$ and $\mathbf{Q} = \mathbf{B}_{\neg t} (\boldsymbol{\Pi}_{\neg t} + \mathbf{L}_{\neg t}^T \mathbf{L}_{\neg t})^{-1} \mathbf{L}_{\neg t}^T$ and let $\mathbf{P}_{sub}$ and $\mathbf{Q}_{sub}$ be their respective matrices with only the rows that are not in the selection profiles of $\mathbf{a}$ and $\mathbf{b}$.

If $K \leq Nk_C + Mk_T$ and $\mathbf{R} = \begin{pmatrix} \mathbf{P}_{sub} \\ \mathbf{Q}_{sub} \end{pmatrix}$ is full-rank, then the only $\boldsymbol{\delta}$ in the identifiability subspace which preserves the selection profile of $\mathbf{a}$ and $\mathbf{b}$ is $\boldsymbol{\delta} = 0$.

Proof. The only free parameters lie in the K-dimensional subspace determined by $\boldsymbol{\delta}$ or equivalently $\boldsymbol{\Delta}$. We can rewrite the identifiability transform equations as

$$\begin{aligned}
\mathbf{a}' - \mathbf{a} = \mathbf{a}'' &= -(\boldsymbol{\Pi}_{\neg t} + \mathbf{L}_{\neg t}^T \mathbf{L}_{\neg t})^{-1} \mathbf{L}_{\neg t}^T \boldsymbol{\delta} && (5.71)\\
&= \mathbf{P}\boldsymbol{\delta} && (5.72)\\
\mathbf{b}' - \mathbf{b}_t = \mathbf{b}'' &= -\mathbf{B}_{\neg t} (\boldsymbol{\Pi}_{\neg t} + \mathbf{L}_{\neg t}^T \mathbf{L}_{\neg t})^{-1} \mathbf{L}_{\neg t}^T \boldsymbol{\delta} && (5.73)\\
&= \mathbf{Q}\boldsymbol{\delta}. && (5.74)
\end{aligned}$$

In these equations, the subset of $\mathbf{a}''$ and $\mathbf{b}''$ which are not included in the selection profile must be set to 0 to preserve the support. Thus, we are only concerned with the subset of the parameters that are constrained to be 0. Call this subset (of rows)

$$\begin{aligned}
\mathbf{P}_{\text{sub}} \boldsymbol{\delta} = \mathbf{a}''_{\text{sub}} = 0 && (5.75)\\
\mathbf{Q}_{\text{sub}} \boldsymbol{\delta} = \mathbf{b}''_{\text{sub}} = 0 && (5.76)
\end{aligned}$$

where $\mathbf{P}_{\text{sub}}$ and $\mathbf{Q}_{\text{sub}}$ are the $Nk_C \times K$ and $Mk_T \times K$ linear transforms. This linear system can collectively be written as the $(Nk_C + Mk_T) \times K$ linear system

$$\begin{pmatrix} \mathbf{P}_{\text{sub}} \\ \mathbf{Q}_{\text{sub}} \end{pmatrix} \boldsymbol{\delta} = \mathbf{R}\boldsymbol{\delta} = \begin{pmatrix} \mathbf{a}''_{\text{sub}} \\ \mathbf{b}''_{\text{sub}} \end{pmatrix} = 0. \quad (5.77)$$

When $K \leq Nk_C + Mk_T$ and $\mathbf{R}$ is full-rank, the only solution is $\boldsymbol{\delta} = 0$. □

Otherwise, there will be a family of non-trivial solutions which live in the kernel of $\mathbf{R}$. So, solutions with higher sparsity permit a higher latent dimensionality which still having a unique support-preserving solution. Thus, for a sufficiently sparse estimate with low enough latent dimensionality, a check of the rank of a matrix is sufficient to determine whether the identifiability subspace has been constrained through fixing the support. Note that this determines conditions on identifiability for both ground-truth parameters or estimated parameters in the triangular model.

Modularized inference procedures

The structural non-identifiability elevates the issue of selection in parameter inference for the triangular model. Specifically, a sufficiently sparse number of non-zero tuning and coupling parameters must be identified to ensure that their values can be accurately estimated, thereby avoiding the simultaneous equations bias. Thus, parameter inference in the triangular model can be modularized into a selection procedure, which identifies the non-zero parameters, and an estimation procedure, which estimates their values given a selection profile. In this section, we detail each components of the overall modularized inference procedure.

Selection procedures

Sparse TM inference. One natural approach to performing selection in the triangular model is to apply an ℓ_1 penalty to the tuning and coupling parameters during expectation-maximization. However, since these parameters may have different sparsity levels, a different ℓ_1 penalty must be applied to each set. In practice, this would occur during the M-step of the EM optimization when maximizing the expected complete log-likelihood. Thus, the M-step would consist of minimizing the expression

$$\ell_M(\theta; \mathcal{D}) = -\langle \ell_c(\theta; \mathcal{D}) \rangle + \lambda_1 |\mathbf{a}|_1 + \lambda_2 |\mathbf{b}_t|_1 \tag{5.78}$$

where we assume that the tuning penalties are applied equally across both target and non-target parameters. Such an optimization would require cross-validating over a grid of (λ_1, λ_2) combinations.

In practice, only one λ penalty can be used at a time. We can sidestep this issue by rescaling the parameters during optimization. Let $r = \lambda_2/\lambda_1$ and

$$\mathbf{b}'_t = r\mathbf{b}_t \tag{5.79}$$

so that

$$\ell_M(\mathbf{a}, \mathbf{b}_t, \theta; \mathcal{D}) = -\langle \ell_c(\mathbf{a}, \mathbf{b}_t, \theta; \mathcal{D}) \rangle + \lambda_1 |\mathbf{a}|_1 + \lambda_2 |\mathbf{b}_t|_1 \tag{5.80}$$

$$= -\langle \ell_c(\mathbf{a}, \frac{1}{r}\mathbf{b}'_t, \theta; \mathcal{D}) \rangle + \lambda_1 |\mathbf{a}|_1 + \frac{\lambda_2}{r} |\mathbf{b}'_t|_1 \tag{5.81}$$

$$= -\langle \ell_c(\mathbf{a}, \frac{1}{r}\mathbf{b}'_t, \theta; \mathcal{D}) \rangle + \lambda_1 |\mathbf{a}|_1 + \lambda_1 |\mathbf{b}'_t|_1 \tag{5.82}$$

$$= \ell'_M(\mathbf{a}, \mathbf{b}'_t, \theta; \mathcal{D}). \tag{5.83}$$

The expressions ℓ_M and ℓ'_M are equivalent aside from a reparameterization. Thus, the minimization we want to achieve,

$$\mathbf{a}^*, \mathbf{b}_t^* = \underset{\mathbf{a}, \mathbf{b}_t}{\operatorname{argmin}} \, \ell_M(\mathbf{a}, \mathbf{b}_t, \theta; \mathcal{D}) \tag{5.84}$$

can be achieved by instead minimizing

$$\mathbf{a}^*, \mathbf{b}_t'^* = \underset{\mathbf{a}, \mathbf{b}'_t}{\operatorname{argmin}} \, \ell'_M(\mathbf{a}, \mathbf{b}'_t, \theta; \mathcal{D}). \tag{5.85}$$

However, we want the solution that minimizes ℓ_M, so after performing the optimization, we need to transform back to the desirable parameters:

$$\mathbf{b}_t^* = \mathbf{b}_t'^* / r. \tag{5.86}$$

Sparse TC inference. If we enforce sparsity in this fashion via the triangular model, the most natural comparison to the tuning and coupling model is via a similar sparse optimizer. Specifically, the loss function would be simply be the mean-squared error, with the additional penalties:

$$\ell_{TC}(\mathbf{a}, \mathbf{b}_i; \mathcal{D}) = \sum_{d=1}^{D} \left(y^{(d)} - \mathbf{x}^{(d)^T} \mathbf{b}_i - \mathbf{y}_{\neg t}^{(d)^T} \mathbf{a} \right)^2 + \lambda_1 |\mathbf{a}|_1 + \lambda_2 |\mathbf{b}_i|_1 \tag{5.87}$$

This optimization problem is ultimately a linear regression with lasso penalty, with some parameters penalized differently than others. Thus, it can easily be solved using a cross-validation grid to determine the best (λ_1, λ_2) configuration.

TC Selection. This approach is the same as sparse TC inference, but instead of one penalty, a single penalty is applied to both the tuning and coupling parameters. Thus, only one loop of cross-validation needs to be performed.

Separate Selection. Lastly, we can perform separate tuning and coupling selection fits using an inference procedure of choice to obtain corresponding selection profiles.

Estimation procedures

Estimation procedures accept a selection profile and estimate the values of the non-zero parameters. If we use a tuning and coupling model, we can simply use ordinary least squares. If we use a triangular model, we can simply use EM inference, while masking the updates for the non-zero parameters. We propose an additional estimation procedure called Iterated Two-Stage Factor Analysis (ITSFA).

Iterated Two-Stage Factor Analysis. This approach is based on the the usual econometric approach to combat the SEB, known as *two-stage least squares*, where the endogenous features are replaced with their projections onto the column spaces of alternative features known as *instrumental variables* [5, 194]. These instruments are necessary because simultaneous equations models can be unidentifiable, preventing maximum likelihood estimation [78]. The instruments are specifically chosen such that they are correlated with the response variable only through the endogenous features. In a canonical example, the relationship between smoking and general health can be understood using the tax rate on cigarettes as an instrument, as tax rates should only be correlated with health outcomes through smoking [112].

It may seem natural to use the stimulus as an instrument for the non-target neurons, since this reflects their data generation process. However, in the triangular model, this is problematic due to their direct correlation with y_i through the tuning parameters $\mathbf{b}_i$. Instead, we note that projecting the non-target neurons on the stimulus gives us access to an estimate of the shared variability (i.e., $\mathbf{z}^T\mathbf{L}_{\neg i}$), which is the cause of the SEB. Our approach is founded on goal of isolating this shared variability and removing it from the dataset, accomplished in two stages.

First Stage. If the shared variability (i.e., the contribution from the latent space) is removed from the data for the non-target neurons, a second-stage regression would provide unbiased estimation of the tuning and coupling parameters. Thus, the first stage consists of estimating the noise terms by regressing the non-target neural responses $\mathbf{Y}_{\neg i}$ on the stimulus $\mathbf{X}$ with $\text{UoI}_{\text{Lasso}}$ and calculating residuals (Lines 5-6). We apply factor analysis to the residuals to obtain an estimate of the shared variability, and remove it from $\mathbf{Y}_{\neg i}$ to obtain a modified dataset $\mathbf{Y}'_{\neg i}$ (Line 8). This first stage is analogous to profile likelihood approach for $\mathbf{B}_{\neg i}$ followed by an application of the EM-algorithm to the residuals in order to deduce the shared variability contribution to uncaptured variance [130, 109].

(Iterated) Second Stage. In practice, the factor analysis cannot truly capture the shared variability, as the E-step will only provide the expected latent state values. Thus, some shared variability will remain in the non-target design matrix and the SEB will persist. Therefore, we utilize a second stage regression as an error correction step. Specifically, we perform a TC model regression (target neuron $\mathbf{y}_i$ on the regressor set $(\mathbf{X}, \mathbf{Y}'_{\neg i})$) to extract estimates $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}_i$ of the fitted parameters, and obtain a set of residuals from this fit. We then perform factor analysis on the concatenated residuals between the first and second stages, and subtract the estimated shared variability from the target neuron to obtain $\mathbf{y}'_i$. The second stage regression is performed once again, now regressing $\mathbf{y}'_i$ on $(\mathbf{X}, \mathbf{Y}'_{\neg i})$ and

Algorithm 2 IteratedTwoStageFactorAnalysis($\mathbf{X}, \mathbf{Y}_{\neg i}, \mathbf{y}_i$).

Input: $\mathbf{X}$, $D \times M$ stimulus design matrix
$\mathbf{Y}_{\neg i}$, $D \times N$ non-target neural activity matrix
$\mathbf{y}_i$, $D \times 1$ target neuron activity vector

$\hat{\mathbf{B}}_{\neg i} \leftarrow$ Regress $\mathbf{Y}_{\neg i}$ on $\mathbf{X}$
$\mathbf{R}_{\neg i} \leftarrow \mathbf{Y}_{\neg i} - \mathbf{X}\hat{\mathbf{B}}_{\neg i}$ ▷ *non-target neuron residuals*
$\hat{\mathbf{L}}_{\neg i}, \hat{\mathbf{\Psi}}_{\neg i}, \hat{\mathbf{Z}}, K \leftarrow$ FactorAnalysisCV($\mathbf{R}_{\neg i}$) ▷ *cross-validated factor analysis*
$\mathbf{Y}'_{\neg i} \leftarrow \mathbf{Y}_{\neg i} - \hat{\mathbf{Z}}\hat{\mathbf{L}}_{\neg i}$ ▷ *remove shared variability*
$(\hat{\mathbf{a}}, \hat{\mathbf{b}}_i) \leftarrow$ OLS regress $\mathbf{y}_i$ on $(\mathbf{X}, \mathbf{Y}'_{\neg i})$
while parameters not converged **or** max iterations not reached **do**
  $\mathbf{r}_i \leftarrow \mathbf{y}_i - \left(\mathbf{X}\hat{\mathbf{b}}_i + \mathbf{Y}_{\neg i}\hat{\mathbf{a}}\right)$ ▷ *target neuron residuals*
  $\mathbf{R} \leftarrow [\mathbf{r}_i, \mathbf{R}_{\neg i}]$ ▷ *concatenate residuals*
  $\hat{\mathbf{L}}, \hat{\mathbf{\Psi}}, \hat{\mathbf{z}} \leftarrow$ FactorAnalysis($\mathbf{R}, K$) ▷ *factor analysis*
  $\mathbf{y}'_i \leftarrow \mathbf{y}_i - \hat{\mathbf{Z}}\hat{\mathbf{l}}_i$ ▷ *remove variability*
  $(\hat{\mathbf{a}}, \hat{\mathbf{b}}_i) \leftarrow$ OLS regress $\mathbf{y}'_i$ on $(\mathbf{X}, \mathbf{Y}'_{\neg i})$ ▷ *iterated regression*
end while
return $\hat{\mathbf{a}}, \hat{\mathbf{b}}_i$

obtaining estimates $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}_i$. This process can be repeated until the parameters converge to some tolerance or a maximum number of iterations has been reached. Empirically, we observe consistent convergence of the parameters. A pseudocode for ITSFA is shown in Algorithm 2.

Neural recordings and data analysis

Recordings from auditory cortex

Auditory cortex (AC) data was comprised of cortical surface electrical potentials (CSEPs) recorded from rat auditory cortex with a custom fabricated micro-electrocorticography (μECoG) array. The μECoG array consisted of an 8×16 grid of 40 μm diameter electrodes. Anesthetized rats were presented with 50 ms tone pips of varying amplitude (8 different levels of attenuation, from 0 dB to -70 db) and frequency (30 frequencies equally spaced on a log-scale from 500 Hz to 32 kHz). Each frequency-amplitude combination was presented 20 times, for a total of 4200 samples. The response for each trial was calculated as the z-scored, to baseline, high-γ band analytic amplitude of the CSEP, calculated using a constant-Q wavelet transform. Of the 128 electrodes, we used 125, excluding 3 due to faulty channels. Data was recorded by Dougherty & Bouchard (DB). Further details on the surgical, experimental, and preprocessing steps can be found in [61].

Recordings from primary visual cortex

We analyzed three primary visual cortex (V1) datasets, comprised of spike-sorted units simultaneously recorded in three anesthetized macaque monkeys. Recordings were obtained with a 10×10 grid of silicon microelectrodes spaced 400 μm apart and covering an area of 12.96 mm^2. Monkeys were presented with grayscale sinusoidal drifting gratings, each for 1.28 s. Twelve unique drifting angles (spanning 0° to 330°) were each presented 200 times, for a total of 2400 trials per monkey. Spike counts were obtained in a 400 ms bin after stimulus onset. We obtained $[106, 88, 112]$ units from each monkey. The data was obtained from the Collaborative Research in Computational Neuroscience (CRCNS) data sharing website [192] and was recorded by Kohn and Smith (KS) [104]. Further details on the surgical, experimental, and preprocessing steps can be found in [180] and [102].

Modeling tuning with basis functions

We used basis functions to model the influence of tuning in the neural data. We use a set of M basis functions $g_i(s)$, which form the M-dimensional stimulus representation $\mathbf{x}$. The contribution to the activity of a specific neuron provided by the stimulus is encoded by tuning parameters $\mathbf{b}$, where

$$g(s) = \mathbf{x}^T\mathbf{b} = \sum_{i=1}^{M} b_i \cdot g_i(s) \tag{5.88}$$

For the primary visual cortex data, we use cosine basis function (using $M = 2$). For the auditory cortex data, we used Gaussian basis functions. Specifically, we chose $M = 8$ basis functions tiling the log-frequency plane with a standard deviation of $\sigma^2 = 0.64$ octaves.

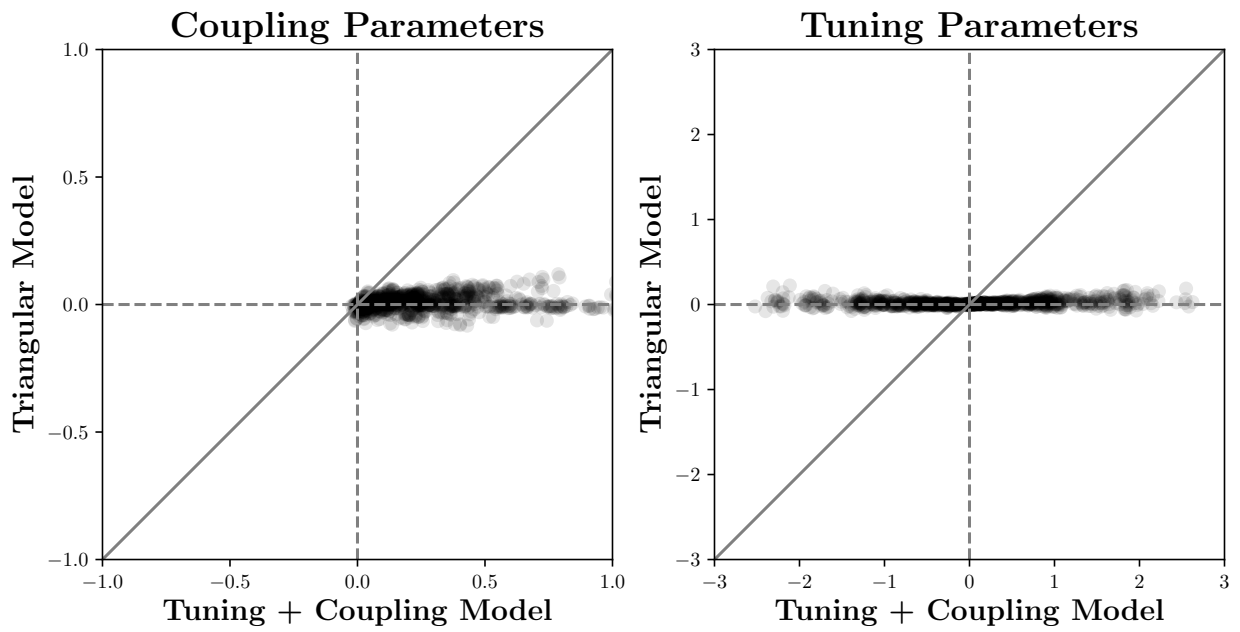
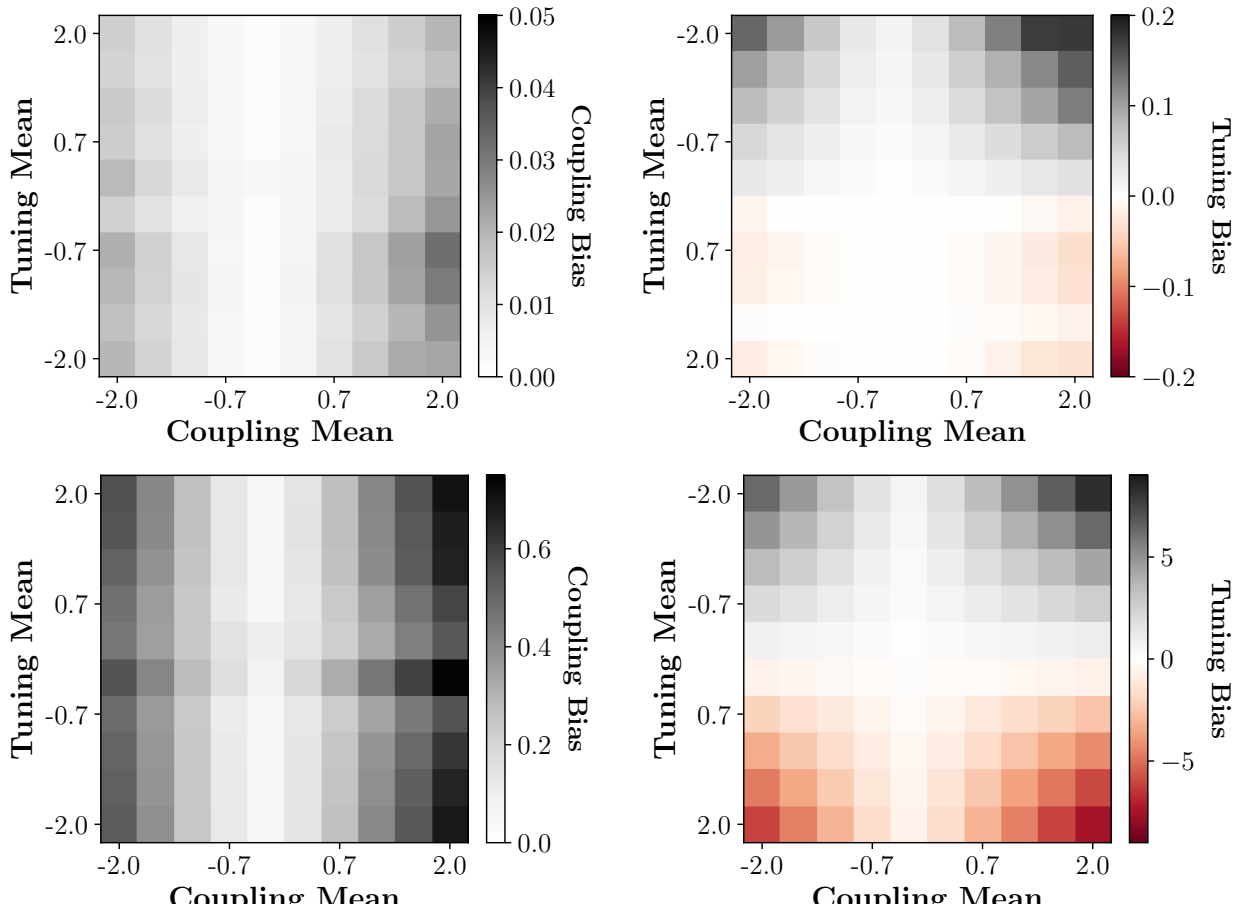


Figure 5.4: **Triangular model inference alleviates the simultaneous equations bias.** Each point is a different model and hyperparameter configuration. **Left.** The bias of the coupling parameters, in the triangular model versus the tuning and coupling model. **Right.** The bias of the tuning parameters, in the triangular model versus the tuning and coupling model.



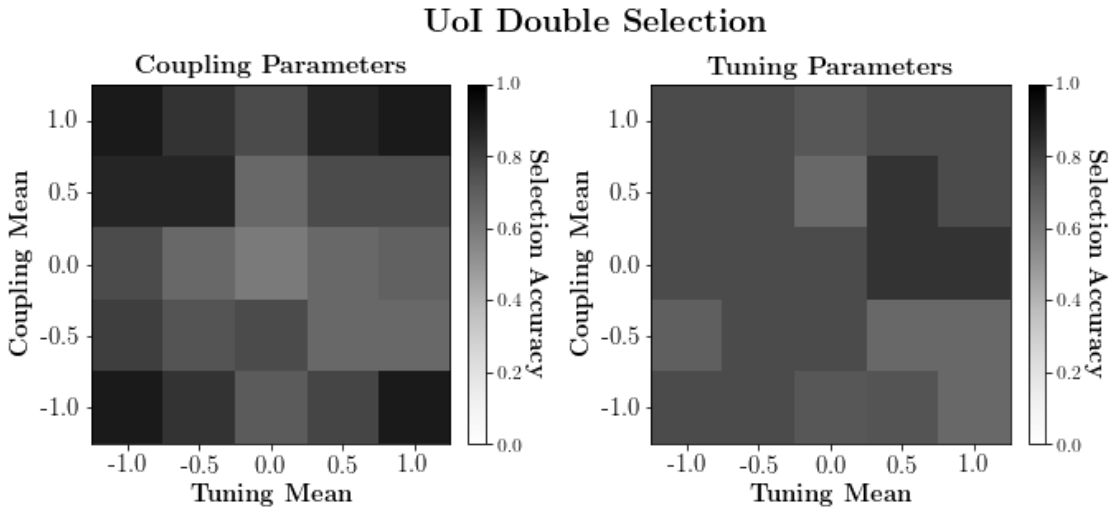


Figure 5.6: **Selection performance in the synthetic data.** Selection accuracy is summarized across models, datasets, and cross-validation folds. The procedure consisted of applying separate $\text{UoI}_{\text{Lasso}}$ fits for the tuning and coupling parameters. **Left.** The selection accuracy of the coupling parameters as a function of hyperparameter configuration. **Right.** The selection accuracy of the tuning parameters as a function of hyperparameter configuration.

5.3 Results

Triangular model inference mitigates biases in synthetic data

We evaluated the performance of our inference procedures in synthetic data generated from the triangular model to assess whether they mitigate the simultaneous equations bias. Our main comparison was between inference procedures for the tuning and coupling model and the triangular model. We considered a large-scale synthetic experiment with two hyperparameters of interest: the means of the coupling and tuning parameters. Specifically, we enforced $N = M = 10$ coupling and tuning parameters, sparsities of $k_T = k_C = 0.5$, and a noise correlation of $\rho_C = 0.25$ with $K = 1$ latent factor. Then, we drew both tuning parameters and coupling parameters from Gaussian distributions (with variance $\sigma^2 = 0.5$), allowing the means to vary across $\mu \in [-2, 2]$.

Since we modularized the inference procedure, we consider two cases. First, we utilize oracle selection, where each inference procedure was provided the true selection profile of the triangular model. For the triangular model, this consists of performing expectation-maximization with no regularization. For the TCM, this consists of ordinary least squares. We considered 10 values per hyperparameter (for a total sweep of 10×10 settings), and for each hyperparameter setting, we considered 10 models. For each model, we drew 30 datasets and performed inference over 3 folds of the data. We estimated statistics by taking averages or variances across datasets, averaging across folds, and taking a median across models and

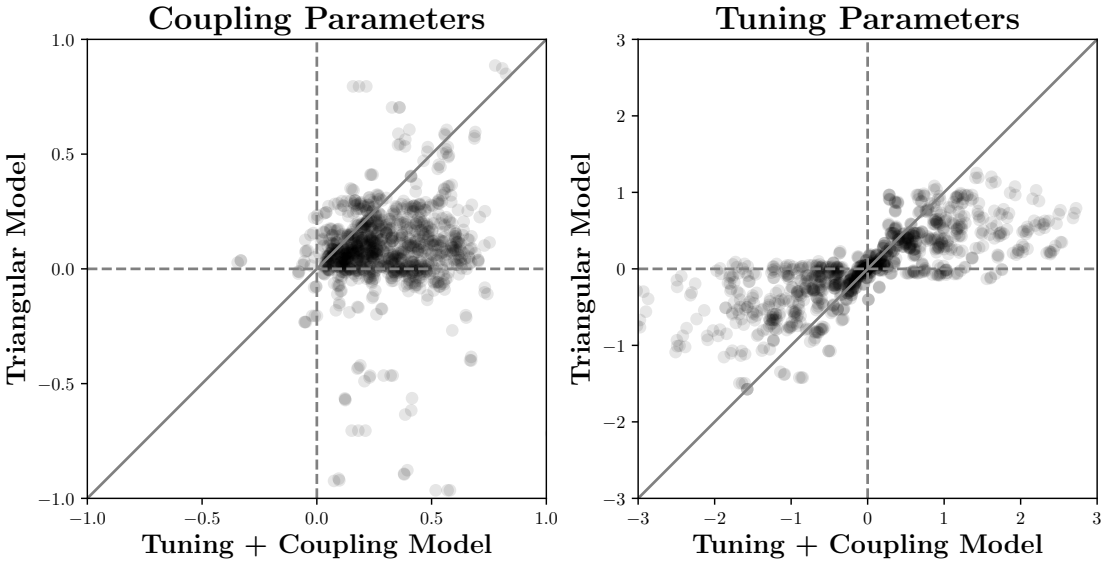


Figure 5.7: **Estimation performance with inferred selection profiles in synthetic data.** Each point is a different model and hyperparameter configuration. **Left.** The bias of the coupling parameters, in the triangular model versus the tuning and coupling model. **Right.** The bias of the tuning parameters, in the triangular model versus the tuning and coupling model.

parameters within a model.

Our results are shown in Figure 5.4. We find that triangular model inference achieves nearly unbiased parameter estimates for both the tuning and coupling parameters. Meanwhile, the simultaneous equations bias is evident for the tuning and coupling model. Interestingly, we observed only positive biases for the coupling parameters, while the tuning parameters exhibit both negative and positive biases. To better assess this difference in sign, we examined aggregate biases across models, for each hyperparameter configuration (Fig. 5.5). We find that, as expected, bias magnitude increases with both the tuning or coupling mean, for both sets of parameters. Furthermore, the tuning parameter bias corresponds to the sign of the underlying hyperparameters: if the hyperparameter is positive, the bias will be negative, and vice versa (Fig. 5.5, right). Furthermore, we similarly observe that triangular model inference exhibits lower bias than the tuning and coupling model. In particular, its biases are an order of magnitude smaller, indicating what is effectively unbiased estimation. Interestingly, however, the signs of the biases exhibit the same structure as depicted in the tuning and coupling case.

Next, we evaluated how our inference procedure performed when selection was required. We considered double selection using $\text{UoI}_{\text{Lasso}}$, which typically achieved the best selection performance (Fig. 5.6). We observed that selection performance was not strongly modulated by the underlying hyperparameters, with the procedure exhibiting moderate to good selection

performance (selection accuracy greater than 0.5) in all regimes. The procedure tended to suffer from false positives, which is to be expected, since the procedure cannot disentangle influence between tuning and coupling parameters. In the context of the triangular model, however, false positives are less detrimental than false negatives, as the latter can induce additional omitted variables bias.

We then evaluated how the inferred selection profiles influenced the estimation module. Once again, we compared the triangular model to the tuning and coupling model, but this time using the selection profiles obtained by the double selection procedure. Once again, we evaluated the bias across a wide range of model conditions (Fig. 5.7). We observe that the bias is substantially worse in this scenario, demonstrating that selection is important for the success of estimation. While the triangular model still outperforms the tuning and coupling model on the whole, it generally performs worse relative to the case of oracle selection.

Triangular model elevates tuning modulations in neural data

We applied triangular model inference to neural data to assess whether it resulted in any discernible changes in the structure of tuning and functional coupling. We considered two datasets: single-unit activity from macaque primary visual cortex in response to drifting gratings, and μECoG recordings from rat auditory cortex in response to tone pips. We fit linear Gaussian models, encoding the tuning parameters using basis functions. We performed fits for each functional unit of both datasets. We performed selection using a double fit with $\text{UoI}_{\text{Lasso}}$. With the fitted selection profiles, we performed estimation using either the triangular model or the tuning and coupling model. We then examined the parameter estimates to assess whether there were differences between the two.

We examined the tuning modulation for each set of fits. Specifically, we examined the fitted tuning parameters, constructed the tuning curves, and examined the minimum-to-maximum distance. We then compared these tuning modulations between the triangular model and the tuning and coupling model. Our results are shown in Figure 5.8. We observe that, in the case of the auditory cortex data, the tuning modulations of the triangular model are elevated relative to the tuning and coupling model. In the case of the visual cortex data, we generally observe the elevation of tuning modulations. However, in the case of some functional units, we observe that some are *decreased* relative to the TC model. This implies that there is a heterogeneity of changes in the tuning modulation. However, on the whole, our results demonstrate that application of the novel triangular model inference procedures results in alleviation of the simultaneous equations bias, which removes some of the explaining away effect observed in Figure 5.2, but not all of it.

Failure to enforce identifiability reproduces biases

The triangular model suffers from structural non-identifiability. We demonstrated that applying sparsity to the fitted parameter values removes the structural non-identifiability (see Methods). Since structural non-identifiability is often unmitigated in neural models, we

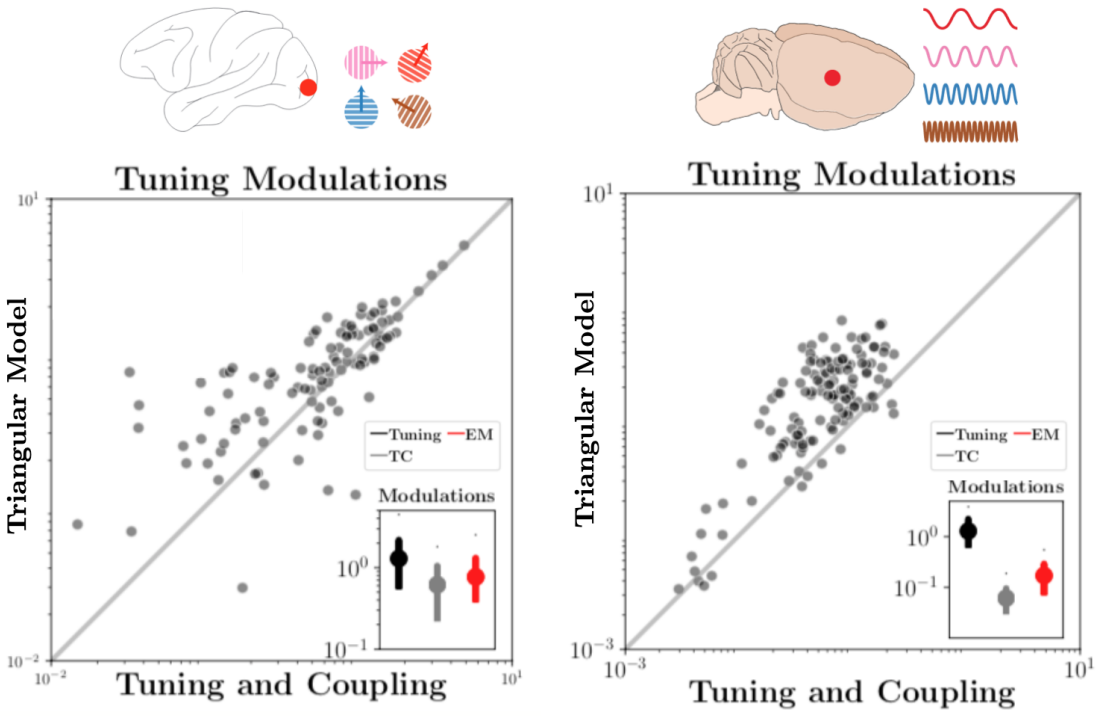


Figure 5.8: **Triangular model inference elevates tuning modulation relative to baseline procedures.** Each point is a different model. Axes denote the tuning modulation of the fitted models, or the minimum-to-maximum distance of the tuning curve. **Left.** Comparison of tuning modulations for macaque V1 recordings. **Right.** Comparison of tuning modulations for rat μECoG recordings.

aimed to assess the degree to which it might parameter inference without correction. To recap, a structural non-identifiability exists when an infinite number of transformations can be applied to any parameter set, resulting in a new parameter set that has equal likelihood. This implies that for a given parameter set, there always exists an infinite number of other parameter sets with the same likelihood. In realistic scenarios, such as the triangular model, the identifiability family exists in a high-dimensional space. We can apply dimensionality reduction to better understand the relationship amongst the variables.

The identifiability family in the triangular model is linear in the target tuning, coupling, and latent factor parameters, and quadratic in the target private variance. The identifiability family is specified by the K-dimensional vector $\boldsymbol{\delta}$, with transform $(\mathbf{a}, \mathbf{b}_t, \mathbf{l}_t, \Psi_t) \rightarrow$

$(\mathbf{a}', \mathbf{b}'_t, \mathbf{l}'_t, \Psi'_t)$ given by

$$\mathbf{l}'_t \leftarrow \mathbf{l}_t + \boldsymbol{\delta} \qquad \text{(linear)} \qquad (5.89)$$

$$\mathbf{a}' \leftarrow \mathbf{a} - \mathbf{P}\mathbf{L}_{\neg t}^T \boldsymbol{\delta} \qquad \text{(linear)} \qquad (5.90)$$

$$\mathbf{b}' \leftarrow \mathbf{b} + \mathbf{B}_{\neg t}\mathbf{P}\mathbf{L}_{\neg t}^T \boldsymbol{\delta} \qquad \text{(linear)} \qquad (5.91)$$

$$\Psi'_t \leftarrow \Psi_t - 2\mathbf{l}_t^T(\mathbf{I} - \mathbf{L}_{\neg t}\mathbf{P}\mathbf{L}_{\neg t}^T)\boldsymbol{\delta} - \boldsymbol{\delta}^T\boldsymbol{\delta} \qquad \text{(quadratic)} \qquad (5.92)$$

where $\mathbf{P} \equiv (\boldsymbol{\Pi}_{\neg t} + \mathbf{L}_{\neg t}^T \mathbf{L}_{\neg t})^{-1}$.

If we have $K = 1$, or one latent dimension, then this implies that the identifiability family can be projected down to fewer dimensions: for example, two which capture projections of the linear subspace, and the other which captures the quadratic subspace denoted by the private variance. Thus, the identifiability family looks like a parabola. Importantly, this quadratic subspace is truncated at two ends, because the target private variance cannot be negative. The identifiability family is visualized in Figure 5.9.

Without enforcing sparsity, optimizations are drawn to the solution on an identifiability family that exhibits the largest private target variance. We performed an experiment to validate this: we initialized a series of fits across an identifiability family, and observed where they end up on a fitted identifiability family. The fitted solutions crowded the apex of the parabola, implying that the target private variance maximizing solution is preferred, regardless of the initialization. The experiment setup is depicted in Figure 5.9 (left), with actual experiment results shown in the figure on the right. The solution with maximum private variance will have the lowest shared variability, which is the closest model to the tuning and coupling model. This implies that the solution will reproduce the simultaneous equations bias.

5.4 Discussion

In this work, we identified that a tuning and coupling model is prone to the simultaneous equations bias. We provided a novel null model that is more complete than the tuning and coupling model, and provided inference procedures that mitigate two issues: the simultaneous equations bias and structural non-identifiability. We validated these procedures on synthetic data and applied them to neural data, observing significant changes in the ensuing tuning modulations.

The triangular model we examined is linear, and thus is limited in how well it characterizes neural activity. Future work could build on this model using a Poisson generalized linear model as done in previous studies [152, 189]. Furthermore, while the model is time-instantaneous, and thus cannot capture dynamics, it incorporates a data generation process for the non-target neurons, more accurately reflecting the flow of tuning through neural populations. Consequentially, this allows the interaction of unobserved activity, modeled as a lower-dimensional latent state, with the coupling parameters. Such interactions introduce identifiability issues that hamper joint estimation of the parameters.

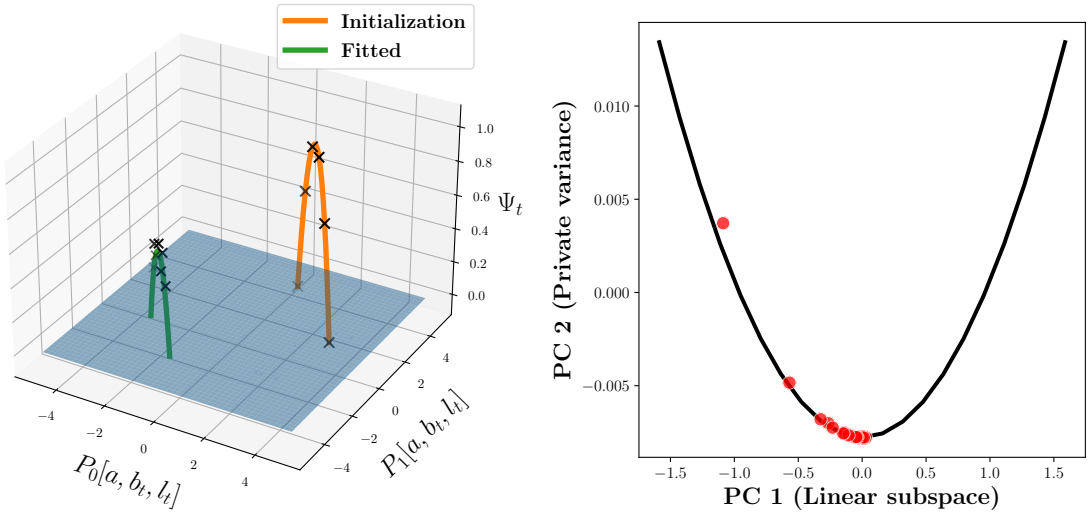


Figure 5.9: **Failure to enforce identifiability reproduces biases. Left:** An series of optimizations are initialized are various points along an identifiability family (orange curve). The resulted fitted parameters are examined to assess where they lie on a fitted identifiability family (green curve). The fitted points cluster toward the solution maximizing private target variance, despite the fact that the initializations were equally spread along the identifiability family. **Right:** The first two principal components of the fitted identifiability family (black line). Within this PC space, the fitted solutions across the initializations are denoted by red points. The red points cluster toward the extrema of the parabola, in accordance with the left plot.

The usage of the unobserved activity effectively acts a way to account for omitted neurons in the model. The inclusion of other omitted variables that are not accounted by the latent state, such as spike-time history or additional tuning parameters (e.g., spatial frequency of the gratings) are also important in order to extract unbiased and interpretable parameter estimates from the model [187].

Lastly, while the triangular model allows the exploration of how tuning can flow through two different pathways, its unidirectional setup leaves the target neuron in a position of privilege relative to the non-target neurons. We aim to pursue a natural extension of the triangular model, the dynamic simultaneous equations model (DSEM), that incorporates the data generation process for all neurons and further captures dynamics. The DSEM is similar to recent work examining autoregressive models [115, 120, 17], but can be rewritten in a "structural form" whose parameters represent instantaneous functional coupling as distinct from tuning and lagged functional coupling [78, 220]. However, the DSEM is also prone to simultaneous equations biases and identifiability concerns. Future work will require developing estimators that mitigate these issues, particularly in the context of selection.

Conclusion

We have demonstrated that omitting correlated variability in neural models of tuning and functional coupling biases their parameter estimates. We proposed a novel model to mitigate this issue, and solved the model's structural non-identifiability by inducing sparsity within it. Thus, we developed novel inference procedures to perform parameter estimations in models that incorporate correlated variability. This concludes our analysis on correlated variability. We now turn to examining the software tools required to complete the analyses described in the thesis thus far.

Chapter 6

Software Engineering Tools for Science

Chapter Co-authors
Jesse A. Livezey
Andrew J. Tritt
Kristofer E. Bouchard

The projects discussed in this thesis required the development of software engineering tools to facilitate their analyses. In this chapter, we discussed a framework to conduct software engineering in science, and provide a case study of this framework in a polished scientific software package. This package, PyUoI, was instrumental in conducting the analyses of correlated variability described in this thesis.

6.1 Introduction

Computational and data-driven research increasingly requires developing complex codebases. At the same time, many scientists don't receive training in software engineering practices, resulting in, for some, the perception that scientists write poor software. More importantly, this lack of training impedes scientific progress, as good software can accelerate scientific work and facilitate its reproducibility. However, achieving such proficiency is not trivial, and generally requires knowledge of and experience with a variety of programming tools.

Each project discussed in this thesis required developing an accompanying software package to facilitate the involved analyses and generate the results and figures. Furthermore, these software packages are readily available in an effort to support the reproduction of all figures in this thesis. Since the development of scientific software was of paramount importance for the research presented here, an entire chapter is dedicated to developing software for science. My hope is that future scientists can use this chapter as a helpful resource in developing good software for their research.

This chapter is organized as follows. In Section 6.2, we start by discussing virtual environments, which should serve as a precursor to actual software developing. Next, in Section 6.3, we discuss how to build a software package for a research project. Then, in Section 6.4, we discuss extensions to professionalize software repositories, making them "production-ready." Lastly, we conclude in Section 6.5 by presenting a case study on software packaging for science called `PyUoI`. Note that these sections are written from the perspective of someone using the Python programming language, but the principles are language agnostic.

6.2 Preparing Virtual Environments

Imagine you have a codebase for one of your projects. You submit a paper, and go work on other projects while waiting on reviews. When the reviews come back, you need to do additional analyses. You open up the codebase for the original project only to find that your code runs into errors. As it turns out, while working on the newer projects, you updated package `A` from version 1.0 to version 2.0. Unfortunately, this update made changes to specific functions your codebase relied on, which expected version 1.0.

This scenario is one motivation for using a *virtual environment*. A virtual environment is an isolated copy of Python and any external packages, all installed at specific versions. Ideally, you'd have a virtual environment for each one of your research projects (or group of closely related research projects). Thus, when using a virtual environment, you can be confident that any changes you make will not impact your other projects. In the scenario described above, your codebase would have its own virtual environment, in which package `A` would be version 1.0. Meanwhile, your other projects would have their own virtual environment(s), in which package `A` could be updated to version 2.0. When you return to the old codebase, you'd switch back to the virtual environment for it, and everything should work as expected.

Building and running a virtual environment requires external software that is easily installable. In Python, the most commonly used distributor of virtual environments is Anaconda (often referred to as `conda`). Anaconda is specifically tailored toward scientific programming, making it an ideal choice for researchers. Furthermore, it is well-supported, actively developed, and has extensive documentation. Setting up a `conda` environment for your research project is the first step you should take before you begin developing any software.

6.3 Setting up a Package for Scientific Programming

The next step is to build a code repository for your research project. In order to manage the development of the repository as it is changed, version control must be implemented. Then, we discuss how to set up the repository specifically as a *software package*, which will facilitate its usage throughout your (and others') analyses. Lastly, we discuss an often overlooked aspect of setting up a code repository: file organization. Structuring repositories according

to a predictable template will make your science easier, cleaner, and more reproducible. This section is accompanied with an example template, which can serve as a model for future code repositories. This template is designed with a "paper repository" in mind: that is, it is intended for projects whose output is a written report (e.g., publication, thesis, technical report). The template I present is accessible on Github. It's forked from another template that is used by my lab. This original template was developed by Jesse Livezey.

Version control: Setting Up a Github repository

Imagine that you are collaborating with one of your labmates on a project. You are both concurrently making changes to functions in the codebase. At one point, you both have changed the same lines in a particular function. How do you go about merging your changes so that you both are using a consistent function? This is the rationale for version control: a system that manages and records changes to a codebase. The most commonly used version control system is called `git` (others include Mercurial and SVN). `git` is often used in tandem with a cloud-based hosting platform–the most common is Github (but others include Gitlab and Bitbucket). The benefit to using Github is that it makes it easier to collaborate on code with others via its web platform. Thus, your next step after creating the environment should be to initialize a git repository for your project, and make sure it's hosted somewhere like Github, Gitlab, or Bitbucket.

Instantiating Your Project as a Package

Now that you have created a repository, you need to start populating it. The first file to create is the `setup.py` file. This file provides instructions to Python on how to treat your repository like a package. The benefit to having your repository be an installable package is that you can access the code within the package–any classes or utility functions–anywhere you might be coding, as long as you import the package. This is much easier than having to set your working directory every time you need to import a class or function from your codebase. The `setup.py` file tells a virtual environment how to install your package. Furthermore, it contains descriptive information about the package as well as any dependencies, which are other packages that need to be installed before you can run your code.

Once you have a setup file, you can install your package onto your `conda` environment using `pip`, which is a Python package index and installer (`pip` and `conda` can work together, each capable of installing packages into a `conda` environment). In particular, you can use `pip` to install an editable version of the package. This means that any changes you make during development will automatically update the package. So, if you are testing some code and find a bug, you can fix the bug and expect the package to update on its own, without having to reinstall it.

Choosing the best folder structure

Now that you have an installable package, you are ready to begin developing. The next step is to decide where to place all the files in the package. It would not be productive to have all your files in the same folder–having some organization will make it easier for you and others to efficiently use the repository. The folder organization for the template linked to above is as follows:

- `codebase`: The name of this folder is set by the `setup.py` file. This is your main codebase: any code that you expect to be imported when this package is imported should go in here. This includes any classes and functions that are consistently used during analyses relevant for the project. Some suggested files are included in the template: these include `analysis.py`, `plotting.py`, and `utils.py`, which contains, as one might expect, analysis, plotting, and utility functions, respectively.
- `scripts`: Contains scripts that perform the important analyses for the project. These scripts will depend on the functions in the codebase, but should not contain functions themselves. For example, a script may apply functions in `analysis.py` on specific datasets, producing outputs that are used in the figures of the paper.
- `notebooks`: Contains Jupyter notebooks that perform important analyses for the project. Note that there may be some flexibility between what goes in scripts and notebooks. This is often up to personal style. As a general rule of thumb, if the output is a plot, use a notebook. If the output is processed data, use a script.
- `figures`: A separate folder, often consisting of Jupyter notebooks that generate the figures (or at least each figure's subpanels) of the paper that the project leads to. Ideally, each figure should have its own notebook. This way, any user can download your repository, install the package, and easily generate the figures in your paper.
- `tests`: An important component of good software engineering is unit testing, where you develop simple tests for the classes and functions in the package. You often do some sort of unit testing as you debug your code. However, storing these tests in their own folder - which can be run with an external package, like pytest, increases confidence in the quality and correctness of the code.

Ultimately, the only required folder here is `codebase`, since it contains the package code. The rest can be tailored to your preferences, but you can use this organization as a starting point.

6.4 Extensions

The above steps should be considered the minimum amount needed to produce a software package for a research project. However, depending on the use case, there are a variety

of tools and practices from software engineering that may be beneficial. These include the following:

- **Code coverage and continuous integration.** As discussed above, unit testing is very important to ensuring that users trust the correctness of your package. Github provides tools to facilitate the effectiveness of these unit tests. The first is code coverage: this is a measure of how well your tests cover your code. That is, it checks what parts of you software package actually are involved in your unit tests, allowing you to assess whether you may have incomplete tests. The second is continuous integration, which continuously runs unit tests as new changes are integrated into the package. This a centralized way of making sure any changes respect the old unit tests, but also conveys to other users the health of a software package (e.g., a repository that is failing its continuous integration is likely not production ready). Both code coverage and continuous integration rely on third party services that automatically run via Github anytime updates are made to the repository.

- **Code linting:** In addition to unit testing, code style is instrumental to ensuring that your code is readable and clean. Different languages have style standards that you should follow (e.g., when to indent, when to have spaces, restrictions on variable names, etc.). In Python, there are packages that can automatically *lint* your code, to point out instances where style is not being adhered to. Such packages include `flake8` or `pylint`. You can include a protocol in your Github repository that details the custom style guide it follows (e.g., a `.flake8` file). Furthermore, variously integrated development environments have add-ons that will run the linting *as* you code, allowing you to make fixes in real-time.

- **Documentation:** The last key component to code reproducibility is documentation. In Python, classes and functions should each be accompanied by docstrings, which provide important information on the inputs, outputs, and what the functions and classes do. There are tools that will automatically compile all docstrings into an easy-to-read website (e.g., a "ReadTheDocs"). In Python, you can use a package called Sphinx to generate these websites.

- **Docker:** A virtual environment helps reproducibility by providing a record of exactly what packages are installed, and what their version numbers are. However, this might not be good enough, particularly if users are running different operating systems. For example, packages often require slightly different dependencies, making cross-platform building of virtual environments tricky. This is where Docker comes in: Docker provides a platform for constructing a container that is, quite literally, a barebones virtual OS capable of running your code. That way, another user can simply run your code within a Docker container, without having to worry about the details of the underlying environment.

6.5 PyUoI: The Union of Intersections Framework in Python

In this section, we describe a software package called `PyUoI`, a Python package that implements the Union of Intersections framework, described earlier in this thesis [31]. `PyUoI` is open-sourced, carefully designed, rigorously tested, thoroughly documented, and fully parallelized, making it easy and flexible to use in the general scientific setting [162]. It was intentionally designed for general use, and structured similarly as `scikit-learn` in order to facilitate its adoption. Thus, `PyUoI` has supported multiple, often unrelated, research projects. We present this package as a case study for scientific software engineering [163].

Summary

The increasing size and complexity of scientific data requires statistical analysis methods to produce models that are both interpretable and predictive. Interpretability implies one can interpret the output of the model in terms of processes generating the data. This typically requires identification of a small number of features in the actual data and accurate estimation of their contributions. Meanwhile, achieving predictive power requires optimizing the performance of some machine learning measure such as precision, mean squared error, etc. There is often a trade-off between interpretability and predictive power. This trade-off is particularly acute for scientific applications, where the output of the model is used to provide insight into the underlying physical processes that generated the data.

The recently introduced Union of Intersections (UoI) is a flexible, modular, and scalable framework designed to enhance both the identification of features (model selection) as well as the estimation of the contributions of these features (model estimation). UoI-based methods leverage stochastic data resampling and a range of sparsity-inducing regularization parameters to build families of potential feature sets robust to perturbations of the data, and then average nearly unbiased parameter estimates of selected features to maximize predictive accuracy. Models inferred through the UoI framework are characterized by their usage of fewer parameters with little or no loss in predictive accuracy relative to benchmark approaches.

`PyUoI` is a Python package containing implementations of a variety of UoI-based algorithms, encompassing regression, classification, and dimensionality reduction. In order to better facilitate its usage, `PyUoI`'s API is structured similarly to the `scikit-learn` package, which is commonly used to build models on scientific data. Additionally, because the UoI framework is naturally scalable, `PyUoI` is equipped with `mpi4py` functionality to parallelize model fitting on large datasets.

Background

The Union of Intersections is not a single method or algorithm, but a flexible statistical framework into which other algorithms can be inserted. In this section, we briefly describe

$\text{UoI}_{\text{Lasso}}$, the UoI implementation of lasso penalized regression. $\text{UoI}_{\text{Lasso}}$ is similar in structure to the UoI versions of other lasso or elastic net penalized generalized linear models. We refer the user to existing literature on the UoI variants of column subset selection and non-negative matrix factorization [31, 201].

Linear regression consists of estimating parameters $\beta \in \mathbb{R}^p$ that map a p-dimensional vector of features $x \in \mathbb{R}^p$ to the observation variable $y \in \mathbb{R}$, when the n samples are corrupted by i.i.d Gaussian noise:

$$y = \beta^T x + \epsilon \tag{6.1}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ for each sample. When the true β is thought to be sparse (i.e., some subset of the β are exactly zero), an estimate of β (i.e., $\hat{\beta}$) can be found by solving a constrained optimization problem of the form

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\text{argmin}} \sum_{i=1}^{n} (y_i - \beta x_i)^2 + \lambda |\beta|_1 \tag{6.2}$$

where $|\beta|_1$ is the ℓ_1-norm of the parameters. The ℓ_1-norm is a convenient penalty because it will tend to force parameters to be set exactly equal to zero, performing feature selection. Typically, λ, the degree to which feature sparsity is enforced, is unknown and must be determined through cross-validation across a set of hyperparameters $\{\lambda_j\}_{j=1}^{k}$.

The key mathematical idea underlying UoI is to perform model selection through intersection (compressive) operations and model estimation through union (expansive) operations, in that order. For $\text{UoI}_{\text{Lasso}}$, the procedure is as follows (see Algorithm 1 for a more detailed pseudocode):

- **Model Selection:** For each λ_j, generate Lasso estimates on N_S resamples of the data (Line 2). The support S_j (i.e., the set of non-zero parameters) for λ_j consists of the features that persist in all model fits across the resamples (Line 7).
- **Model Estimation:** For each support S_j, perform Ordinary Least Squares (OLS) on N_E resamples of the data. The final model is obtained by averaging across the supports chosen according to some model selection criteria, such as optimally predicting on data according to an information criterion (Lines 20-21).

Thus, the selection module ensures that, for each λ_j, only features that are stable to perturbations in the data (resamples) are allowed in the support S_j. Meanwhile, the estimation module ensures that only the predictive supports are averaged together in the final model. The degree of feature compression via intersections (quantified by N_S) and the degree of feature expansion via unions (quantified by N_E) can be balanced to maximize prediction accuracy for the response variable y.

Features

`PyUoI` is split up into two modules, with the following UoI algorithms:

- `linear_model` (generalized linear models)
  - Lasso penalized linear regression ($\text{UoI}_{\text{Lasso}}$).
  - Elastic-net penalized linear regression ($\text{UoI}_{\text{ElasticNet}}$).
  - Logistic regression (binary and multinomial) ($\text{UoI}_{\text{Logistic}}$).
  - Poisson regression ($\text{UoI}_{\text{Poisson}}$).
- `decomposition` (dimensionality reduction)
  - CUR decomposition (UoI_{CUR}).
  - Non-negative matrix factorization (UoI_{NMF}).

Similar to `scikit-learn`, each UoI algorithm has its own class. Instantiations of these classes are created with specific hyperparameters and are fit to user-provided datasets. The hyperparameters allow the user to fine-tune the number of resamples, fraction of data in each resample, and the model selection criteria used in the estimation module (in Algorithm 1, Bayesian information criterion is used, but test set accuracy and the Akaike Information Criteria are also available). Additionally, `PyUoI` is agnostic to the specific solver used for a given model. For example, for $\text{UoI}_{\text{Lasso}}$, `PyUoI` comes equipped with a coordinate descent solver (from `scikit-learn`), a built-in Orthant-Wise Limited memory Quasi-Newton solver, and the `pycasso` solver. The choice of solver is left to the user as a hyperparameter.

Applications

We have used `PyUoI` largely in the realm of neuroscience and genomics. A few applications include:

- Interpretable functional connectivity networks from neural populations in the visual, auditory, and motor cortices of various animal models;
- Sparse decoding of behavioral activity from spiking neural activity;
- Parts-based decomposition of electrocorticography recordings in rat auditory cortex that reflect functional cortical organization;
- Extraction of characteristic single nucleotide polymorphisms for the prediction of phenotypes in mice.

However, the algorithms implemented in `PyUoI` are broadly applicable and not limited to these contexts.

Extensions to standard estimators

`PyUoI` builds on commonly used optimization procedures provided by `scikit-learn`, but there are scenarios in which `scikit-learn` lacks proper estimators, which required custom implementations. Specifically, there are two points where the `scikit-learn` estimators cannot be used for fitting. The first is during estimation when a zero-feature support has been found during the selection step. In this case, a model should be fit with no input features, but only an intercept. This is not supported with the `scikit-learn` estimators. The second is after selection and estimation, when the final intercept in the model needs to be fit without adjusting the coefficients.

For linear and Poisson regression, there are closed form solutions for fitting intercepts in both of these situations. For logistic regression with no features, there is also a closed form solution. For the fixed-feature logistic regression cases, we have to solve an optimization problem. We have implemented fitting functions or classes for these two cases. They require calculations of the loss and gradient of the loss, described in the following sections.

Bernoulli logistic regression

For Bernoulli logistic regression (binary choice), the model is defined as

$$\begin{aligned} P(y=1|x;w,b) = \hat{y} = \sigma(x\cdot w + b) \text{ with} \\ \sigma(r) = \frac{1}{1+\exp(-r)}. \end{aligned} \tag{6.3}$$

The negative log-likelihood (nll) for one sample is

$$\begin{aligned} \text{nll}(x,y) &= -y\log\left(\frac{1}{1+\exp(-x\cdot w-b)}\right) - (1-y)\log\left(1-\frac{1}{1+\exp(-x\cdot w-b)}\right) \\ &= y\log(1+\exp(-x\cdot w-b)) + (1-y)(x\cdot w+b+\log(1+\exp(-x\cdot w-b)) \\ &= \log(1+\exp(-x\cdot w-b)) + (1-y)(x\cdot w+b). \end{aligned} \tag{6.4}$$

The derivative with respect to b is

$$\begin{aligned} \frac{\partial \text{nll}}{\partial b} &= -\frac{\exp(-x\cdot w-b)}{1+\exp(-x\cdot w-b)} + (1-y) \\ &= \sigma(x\cdot w+b) - y \end{aligned} \tag{6.5}$$

which is the update for the fixed-feature case. For a dataset, both the nll and gradient should be averaged over samples.

In the case where we have no features, the nll simply becomes:

$$\begin{aligned} \text{nll}(x,y) &= \log(1+\exp(-b)) + (1-y)b \\ \frac{\partial \text{nll}}{\partial b} &= \sigma(b) - y \end{aligned} \tag{6.6}$$

which can be solved for the intercept as:

$$b = \log \frac{\langle y \rangle}{1 - \langle y \rangle}, \tag{6.7}$$

where $\langle y \rangle$ is the average of the response variable over samples. This needs to be clipped to prevent the log from blowing up if the dataset contains only one class.

Multinomial logistic regression

In the multinomial case, having one intercept per class is overparameterized, since an additive constant will get normalized out of the softmax. So, without loss of generality, we choose to set the first element of the intercept to zero and solve for the rest.

For multinomial logistic regression (multiclass), the model is defined as

$$P(y_i = 1|x; w, b) = \hat{y}_i = \begin{cases} \frac{\exp(x \cdot \beta_i)}{\exp(x \cdot \beta_j) + \sum_{j>1} \exp(x \cdot \beta_j + b_j)}, & i = 1 \\ \frac{\exp(x \cdot \beta_i + b_i)}{\exp(x \cdot \beta_j) + \sum_{j>1} \exp(x \cdot \beta_j + b_j)}, & i > 1 \end{cases} \tag{6.8}$$

The negative log-likelihood for one sample is

$$\begin{aligned} \text{nll}(x, y_i = 1) &= \begin{cases} -\log \left(\frac{\exp(x \cdot \beta_i)}{\exp(x \cdot \beta_j) + \sum_{j>1} \exp(x \cdot \beta_j + b_j)} \right), & i = 1 \\ -\log \left(\frac{\exp(x \cdot \beta_i + b_i)}{\exp(x \cdot \beta_j) + \sum_{j>1} \exp(x \cdot \beta_j + b_j)} \right), & i \neq 1 \end{cases} \\ &= \begin{cases} -x \cdot \beta_i + \log \left(\exp(x \cdot \beta_j) + \sum_{j>1} \exp(x \cdot \beta_j + b_j) \right), & i = 1 \\ -x \cdot \beta_i - b_i + \log \left(\exp(x \cdot \beta_j) + \sum_{j>1} \exp(x \cdot \beta_j + b_j) \right), & i > 1 \end{cases}. \end{aligned} \tag{6.9}$$

Thus, the derivative with respect to b_k is

$$\frac{\partial \text{nll}(x, y_i = 1)}{\partial b_k} = \begin{cases} \frac{\exp(x \cdot \beta_k + b_k)}{\exp(x \cdot \beta_j) + \sum_{j>1} \exp(x \cdot \beta_j + b_j)}, & i = 1 \\ -\delta_{ij} + \frac{\exp(x \cdot \beta_k + b_k)}{\exp(x \cdot \beta_j) + \sum_{j>1} \exp(x \cdot \beta_j + b_j)}, & i > 1 \end{cases} \tag{6.10}$$

which is the update for the fixed-feature case. For a dataset, both the nll and gradient should be averaged over samples.

For the no-feature case, this expression is the same without the $x \cdot w$

$$\begin{aligned} \text{nll}(x, y_i = 1) &= \begin{cases} \log \left(1 + \sum_{j>1} \exp(b_j) \right), & i = 1 \\ -b_i + \log \left(1 + \sum_{j>1} \exp(b_j) \right), & i > 1 \end{cases} \\ \frac{\partial \text{nll}(x, y_i = 1)}{\partial b_k} &= \begin{cases} \frac{\exp(b_k)}{1 + \sum_{j>1} \exp(b_j)}, & i = 1 \\ -\delta_{ik} + \frac{\exp(b_k)}{1 + \sum_{j>1} \exp(b_j)}, & i > 1 \end{cases} \end{aligned} \tag{6.11}$$

which can be solved. If there are any zero-occurrence classes, those class probabilities will need to be clipped away from zero.

Determining the largest L_1 coefficient

For linear models, model selection is done by fitting a family of L_1 regularized models over many bootstraps. The family is defined by a range of regularization parameters. A desirable range of regularization parameters is one that does not always set $\beta = 0$ but provides enough regularization so that many different supports are selected. Here, we describe the strategy for selecting the strongest regularization parameter.

For a L_1 (generalized) regression model, the loss function that is being minimized can be written as

$$\ell = \frac{1}{N}\text{nll}(X, y, \beta, b) + \lambda \sum_{ij} |\beta_{ij}| \tag{6.12}$$

where N is the number of samples and nll is the negative log-likelihood. The gradient of the L_1 part will always be proportional to λ

$$\frac{\partial}{\partial \beta_{nm}} \lambda \sum_{ij} |\beta_{ij}| = \lambda \cdot \text{sign}(\beta_{nm}) \tag{6.13}$$

for $\beta_{nm} \neq 0$. The value for the largest λ is the largest element of the absolute value of the derivative of the nll term when $\beta = 0$.

$$\lambda_{\max} = \max_{n,m} \left| \frac{1}{N} \frac{\partial \text{nll}(X, y, \beta, b)}{\partial \beta_{nm}} |_{\beta=0} \right|. \tag{6.14}$$

At this value, a gradient descent step away from $\beta = 0$ due to the nll will be brought back to 0 by the L_1 term. For models with intercepts (b), $\lambda_{\max}$ is derived with b equal to the value it would take with $\beta = 0$. Said another way, this is the regularization parameter such that the intercept-only model will be chosen.

Lasso. For the multi-target Lasso problem, the average nll is

$$\frac{1}{N}\text{nll}(X, y, \beta, b) = \frac{1}{2N} \sum_{i,k} (y_i^k - \sum_j \beta_{ij} X_j^k - b)^2 \tag{6.15}$$

where subscripts are feature dimensions and superscripts are over samples. $\lambda_{\max}$ can be

solved for as

$$\begin{aligned}
\lambda_{\max} &= \max_{n,m} \left| \frac{1}{N} \frac{\partial \text{nll}(X, y, \beta, b)}{\partial \beta_{nm}} |_{\beta=0} \right| \\
&= \max_{n,m} \left| \frac{1}{2N} \sum_{i,k} \frac{\partial}{\partial \beta_{nm}} (y_i^k - \sum_j \beta_{ij} X_j^k - b_i)^2 |_{\beta=0} \right| \\
&= \max_{n,m} \left| \frac{-1}{N} \sum_{i,j,k} \delta_{in} \delta_{jm} X_j^k (y_i^k - \sum_r \beta_{ir} X_r^k - b_i) |_{\beta=0} \right| \\
&= \max_{n,m} \left| \frac{-1}{N} \sum_k X_m^k (y_n^k - \sum_r \beta_{nr} X_r^k - b_n) |_{\beta=0} \right| \\
&= \max_{n,m} \left| \frac{1}{N} \sum_k X_m^k (y_n^k - b_n |_{\beta=0}) \right|.
\end{aligned} \tag{6.16}$$

L_1-Logistic, Bernoulli. For the L_1 Logistic regression with a Bernoulli noise model (binary features), the average nll is

$$\begin{aligned}
\frac{1}{N} \text{nll}(X, y, \beta, b) &= -\frac{1}{N} \sum_{i,k} \left(y_i^k \log(P(y_i^k = 1 | x, \beta, b)) + (1 - y_i^k) \log(1 - P(y_i^k = 1 | x, \beta, b)) \right) \\
&= -\frac{1}{N} \sum_{i,k} \left(\log(1 - P(y_i^k = 1 | x, \beta, b)) + y_i^k \log \left(\frac{P(y_i^k = 1 | x, \beta, b)}{1 - P(y_i^k = 1 | x, \beta, b)} \right) \right) \\
&= \frac{1}{N} \sum_{i,k} \left(\log(1 + \exp \left(\sum_j \beta_{ij} X_j^k + b_i \right) \right) - y_i^k (\sum_j \beta_{ij} X_j^k + b_i) \right)
\end{aligned} \tag{6.17}$$

where subscripts are feature dimensions and superscripts are over samples, and

$$P(y_i^k = 1 | x, \beta, b) = \frac{1}{1 + \exp(-\sum_j \beta_{ij} X_j^k - b_i)} \tag{6.18}$$

$$1 - P(y_i^k = 1 | x, \beta, b) = \frac{1}{1 + \exp(\sum_j \beta_{ij} X_j^k + b_i)}. \tag{6.19}$$

$\lambda_{\max}$ can be solved for as

$$\begin{aligned}
\lambda_{\max} &= \max_{n,m} \left| \frac{1}{N} \frac{\partial \text{nll}(X, y, \beta, b)}{\partial \beta_{nm}} |_{\beta=0} \right| \\
&= \max_{n,m} \left| \frac{1}{N} \sum_{i,k} \frac{\partial}{\partial \beta_{nm}} \left(\log(1 + \exp(\sum_j \beta_{ij} X_j^k + b_i)) - y_i^k (\sum_j \beta_{ij} X_j^k + b_i) \right) |_{\beta=0} \right| \\
&= \max_{n,m} \left| \frac{1}{N} \sum_{i,k} \delta_{ni} \left(\frac{\exp(\sum_j \beta_{ij} X_j^k + b_i)}{1 + \exp(\sum_j \beta_{ij} X_j^k + b_i)} \sum_j \delta_{mj} X_j^k - y_i^k \sum_j \delta_{mj} X_j^k \right) |_{\beta=0} \right| \\
&= \max_{n,m} \left| \frac{1}{N} \sum_k \left(\frac{X_m^k}{1 + \exp(-\sum_j \beta_{nj} X_j^k - b_n)} - y_n^k X_m^k \right) |_{\beta=0} \right| \\
&= \max_{n,m} \left| \frac{1}{N} \sum_k \left(\frac{X_m^k}{1 + \exp(-b_n|_{\beta=0})} - y_n^k X_m^k \right) \right|.
\end{aligned} \tag{6.20}$$

L_1-Logistic, Multinomial. For the L_1 Logistic regression with Multinomial noise model (classes), the average nll is

$$\begin{aligned}
\frac{1}{N} \text{nll}(X, y, \beta, b) &= -\frac{1}{N} \sum_{i,k} y_i^k \log(P(y_i^k = 1 | x, \beta, b)) \\
&= -\frac{1}{N} \sum_{i,k} y_i^k \log \left(\frac{\exp(\sum_j \beta_{ij} X_j^k + b_i)}{\sum_n \exp(\sum_j \beta_{nj} X_j^k + b_n)} \right)
\end{aligned} \tag{6.21}$$

where subscripts are feature dimensions or classes and superscripts are over samples, and

$$P(y_i^k = 1 | x, \beta, b) = \frac{\exp(\sum_j \beta_{ij} X_j^k + b_i)}{\sum_n \exp(\sum_j \beta_{nj} X_j^k + b_n)}. \tag{6.22}$$

$\lambda_{\max}$ can be solved for as

$$\begin{aligned}
\lambda_{\max} &= \max_{n,m} \left| \frac{1}{N} \frac{\partial \text{nll}(X, y, \beta, b)}{\partial \beta_{nm}} |_{\beta=0} \right| \\
&= \max_{n,m} \left| \frac{-1}{N} \sum_{i,k} \frac{\partial}{\partial \beta_{nm}} \left(y_i^k \log \left(\frac{\exp(\sum_j \beta_{ij} X_j^k + b_i)}{\sum_r \exp(\sum_j \beta_{rj} X_j^k + b_n)} \right) \right) |_{\beta=0} \right| \\
&= \max_{n,m} \left| \frac{-1}{N} \sum_{i,k} y_i^k \frac{\partial}{\partial \beta_{nm}} \left(\sum_j \beta_{ij} X_j^k + b_i - \log(\sum_r \exp(\sum_j \beta_{rj} X_j^k + b_n)) \right) |_{\beta=0} \right| \\
&= \max_{n,m} \left| \frac{-1}{N} \sum_{i,k} y_i^k \left(\sum_j \delta_{in} \delta_{mj} X_j^k - \frac{\sum_r \exp(\sum_j \beta_{rj} X_j^k + b_r) \sum_j \delta_{rn} \delta_{mj} X_j^k}{\sum_r \exp(\sum_j \beta_{rj} X_j^k + b_r)} \right) |_{\beta=0} \right| \\
&= \max_{n,m} \left| \frac{-1}{N} \sum_{i,k} y_i^k X_m^k \left(\delta_{in} - \frac{\exp(b_n|_{\beta=0})}{\sum_r \exp(b_r|_{\beta=0})} \right) \right|
\end{aligned} \tag{6.23}$$

Poisson Regression

The average negative log-likelihood for Poisson regression is given by

$$\frac{1}{N} \text{nll}(x, y, \beta, b) = -\frac{1}{N} \sum_{k=1}^{D} \left[y^k \left(b + \sum_j x_j^k \beta_j \right) - \exp \left(b + \sum_j x_j^k \beta_j \right) \right]. \tag{6.24}$$

Thus, $\lambda_{\max}$ can be solved for as

$$\lambda_{\max} = \max_m \left| \frac{1}{N} \frac{\partial \text{ nll}(x, y, \beta, b)}{\partial \beta_m} \Big|_{\beta=0} \right| \tag{6.25}$$

$$= \max_m \left| -\frac{1}{N} \sum_{k=1}^{D} y^k x_m^k - x_m^k \exp(b) \right| \tag{6.26}$$

$$= \max_m \left| \frac{1}{N} \sum_{k=1}^{D} x_m^k (y^k - \exp(b)) \right|. \tag{6.27}$$

The value of the intercept we use is the b that results from a featureless model, i.e. $b = \log(\bar{y})$. Thus,

$$\lambda_{\max} = \max_m \left| \frac{1}{N} \sum_{k=1}^{D} x_m^k (y^k - \bar{y}) \right|. \tag{6.28}$$

Conclusion

We have provided a framework for developing software engineering packages in scientific settings. These principles were used to great effect to conduct the analyses in this thesis. In particular, PyUoI was presented as a case study, which was used to conduct the correlated variability analyses discussed in the latter half of this thesis.

Bibliography

[1] Reza Abbasi-Asl and Bin Yu. *Structural Compression of Convolutional Neural Networks Based on Greedy Filter Pruning*. 2017. arXiv: 1705.07356.

[2] Larry F Abbott and Peter Dayan. "The effect of correlated variability on the accuracy of a population code". In: *Neural computation* 11.1 (1999), pp. 91–101.

[3] Edward H Adelson and James R Bergen. "Spatiotemporal energy models for the perception of motion". In: *Josa a* 2.2 (1985), pp. 284–299.

[4] Hirotugu Akaike. "A new look at the statistical model identification". In: *IEEE transactions on automatic control* 19.6 (1974), pp. 716–723.

[5] T. W. Anderson and Herman Rubin. "Estimation of the Parameters of a Single Equation in a Complete System of Stochastic Equations". In: *The Annals of Mathematical Statistics* 20.1 (1949), pp. 46–63. DOI: 10.1214/aoms/1177730090. arXiv: arXiv:1011.1669v3. URL: http://projecteuclid.org/euclid.aoms/1177730090.

[6] Gopala K Anumanchipalli, Josh Chartier, and Edward F Chang. "Speech synthesis from neural decoding of spoken sentences". In: *Nature* 568.7753 (2019), pp. 493–498.

[7] Iñigo Arandia-Romero et al. "Multiplicative and additive modulation of neuronal tuning with population activity affects encoded information". In: *Neuron* 89.6 (2016), pp. 1305–1316.

[8] Amos Arieli et al. "Dynamics of ongoing activity: explanation of the large variability in evoked cortical responses". In: *Science* 273.5283 (1996), pp. 1868–1871.

[9] Katelyn L Arnemann et al. "Metabolic brain networks in aging and preclinical Alzheimer's disease". In: *NeuroImage: Clinical* 17 (2018), pp. 987–999.

[10] Fred Attneave. "Some informational aspects of visual perception." In: *Psychological review* 61.3 (1954), p. 183.

[11] Bruno B Averbeck, Peter E Latham, and Alexandre Pouget. "Neural correlations, population coding and computation". In: *Nature reviews neuroscience* 7.5 (2006), pp. 358–366.

[12] Bruno B Averbeck, Peter E Latham, and Alexandre Pouget. "Neural correlations, population coding and computation". In: *Nature reviews neuroscience* 7.5 (2006), p. 358.

[13] Bruno B Averbeck and Daeyeol Lee. "Effects of noise correlations on information encoding and decoding". In: *Journal of neurophysiology* 95.6 (2006), pp. 3633–3644.

[14] Rava Azeredo da Silveira and Fred Rieke. "The Geometry of Information Coding in Correlated Neural Populations". In: *Annual Review of Neuroscience* 44.1 (July 2021), pp. 403–424. ISSN: 1545-4126. DOI: 10.1146/annurev-neuro-120320-082744. URL: http://dx.doi.org/10.1146/annurev-neuro-120320-082744.

[15] Baktash Babadi et al. "A generalized linear model of the impact of direct and indirect inputs to the lateral geniculate nucleus". In: *Journal of Vision* 10.10 (2010), pp. 22–22.

[16] Mahesh Balasubramanian et al. "Optimizing the Union of Intersections LASSO (UoI-LASSO) and Vector Autoregressive (UoI-VAR) Algorithms for Improved Statistical Estimation at Scale". In: *arXiv:1808.06992* (2018).

[17] Mahesh Balasubramanian et al. "Optimizing the Union of Intersections LASSO ($\text{UoI}_{\text{LASSO}}$) and Vector Autoregressive (UoI_{VAR}) Algorithms for Improved Statistical Estimation at Scale". In: *arXiv* (2018).

[18] Luca Baldassarre, Massimiliano Pontil, and Janaina Mourão-Miranda. "Sparsity is better with stability: Combining accuracy and stability for model selection in brain decoding". In: *Frontiers in neuroscience* 11 (2017), p. 62.

[19] Izhar Bar-Gad, Genela Morris, and Hagai Bergman. "Information processing, dimensionality reduction and reinforcement learning in the basal ganglia". In: *Progress in neurobiology* 71.6 (2003), pp. 439–473.

[20] Albert-László Barabási and Réka Albert. "Emergence of scaling in random networks". In: *science* 286.5439 (1999), pp. 509–512.

[21] Boris Barbour et al. "What can we learn from synaptic weight distributions?" In: *TRENDS in Neurosciences* 30.12 (2007), pp. 622–629.

[22] Horace B Barlow. "Possible principles underlying the transformation of sensory messages". In: *Sensory communication* 1 (1961), pp. 217–234.

[23] Ramon Bartolo et al. "Information-limiting correlations in large neural populations". In: *Journal of Neuroscience* 40.8 (2020), pp. 1668–1678.

[24] Danielle S Bassett and Edward T Bullmore. "Small-world brain networks revisited". In: *The Neuroscientist* 23.5 (2017), pp. 499–516.

[25] Danielle S Bassett and Olaf Sporns. "Network neuroscience". In: *Nature neuroscience* 20.3 (2017), p. 353.

[26] Danielle Smith Bassett and ED Bullmore. "Small-world brain networks". In: *The neuroscientist* 12.6 (2006), pp. 512–523.

[27] Jeffrey Beck, Vikranth R Bejjanki, and Alexandre Pouget. "Insights from a simple expression for linear fisher information in a recurrently connected population of spiking neurons". In: *Neural computation* 23.6 (2011), pp. 1484–1502.

[28] Jeffrey M Beck et al. "Not noisy, just wrong: the role of suboptimal inference in behavioral variability". In: *Neuron* 74.1 (2012), pp. 30–39.

[29] Jeffrey M Beck et al. "Not noisy, just wrong: the role of suboptimal inference in behavioral variability". In: *Neuron* 74.1 (2012), pp. 30–39.

[30] Anthony J Bell and Terrence J Sejnowski. "The "independent components" of natural scenes are edge filters". In: *Vision research* 37.23 (1997), pp. 3327–3338.

[31] Kristofer Bouchard et al. "Union of Intersections (UoI) for Interpretable Data Driven Discovery and Prediction". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

[32] Kristofer E Bouchard. "Bootstrapped adaptive threshold selection for statistical model selection and estimation". In: *arXiv preprint arXiv:1505.03511* (2015).

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

[34] Kristofer E Bouchard et al. "Functional organization of human sensorimotor cortex for speech articulation". In: *Nature* 495.7441 (2013), p. 327.

[35] Olivier Bousquet and André Elisseeff. "Stability and generalization". In: *Journal of machine learning research* 2.Mar (2002), pp. 499–526.

[36] Leo Breiman. "Bagging predictors". In: *Machine learning* 24.2 (1996), pp. 123–140.

[37] Braden AW Brinkman et al. "How do efficient coding strategies depend on origins of noise in neural circuits?" In: *PLoS computational biology* 12.10 (2016), e1005150.

[38] Emery N. Brown et al. "An analysis of neural receptive field plasticity by point process adaptive filtering". In: *Proceedings of the National Academy of Sciences* 98.21 (2001), pp. 12261–12266.

[39] Nicolas Brunel and Jean-Pierre Nadal. "Mutual information, Fisher information, and population coding". In: *Neural computation* 10.7 (1998), pp. 1731–1757.

[40] Peter Bühlmann and Sara Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

[41] Lars Buitinck et al. "API design for machine learning software: experiences from the scikit-learn project". In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 2013, pp. 108–122.

[42] Ed Bullmore and Olaf Sporns. "Complex brain networks: graph theoretical analysis of structural and functional systems". In: *Nature reviews neuroscience* 10.3 (2009), pp. 186–198.

[43] M Yu Byron et al. "Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity". In: *Advances in neural information processing systems*. 2009, pp. 1881–1888.

[44] Danilo Bzdok and BT Thomas Yeo. "Inference in the age of big data: Future perspectives on neuroscience". In: *Neuroimage* 155 (2017), pp. 549–564.

[45] Jon Cafaro and Fred Rieke. "Noise correlations improve response fidelity and stimulus encoding". In: *Nature* 468.7326 (2010), p. 964.

[46] Jose M Carmena et al. "Learning to control a brain–machine interface for reaching and grasping by primates". In: *PLoS biology* 1.2 (2003).

[47] Zhe Chen et al. "Statistical inference for assessing functional connectivity of neuronal ensembles with sparse spiking data". In: *IEEE transactions on neural systems and rehabilitation engineering* 19.2 (2010), pp. 121–135.

[48] Mark M Churchland et al. "Neural population dynamics during reaching". In: *Nature* 487.7405 (2012), pp. 51–56.

[49] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. "Finding community structure in very large networks". In: *Physical review E* 70.6 (2004), p. 066111.

[50] Jacob Cohen. *Statistical power analysis for the behavioral sciences*. Academic press, 2013.

[51] Marlene R Cohen and Adam Kohn. "Measuring and interpreting neuronal correlations". In: *Nature neuroscience* 14.7 (2011), p. 811.

[52] Marlene R Cohen and John HR Maunsell. "Attention improves performance primarily by reducing interneuronal correlations". In: *Nature neuroscience* 12.12 (2009), p. 1594.

[53] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[54] John P Cunningham and M Yu Byron. "Dimensionality reduction for large-scale neural recordings". In: *Nature neuroscience* 17.11 (2014), p. 1500.

[55] Abhranil Das and Ila R. Fiete. "Systematic errors in connectivity inferred from activity in strongly coupled recurrent circuits". In: *bioRxiv* (2019).

[56] Mark A Davenport et al. "Introduction to compressed sensing". In: *preprint* 93.1 (2011), p. 2.

[57] Peter Dayan and Laurence F Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. MIT press, 2001.

[58] AF Dean. "The variability of discharge of simple cells in the cat striate cortex". In: *Experimental Brain Research* 44.4 (1981), pp. 437–440.

[59] Michael R Deweese and Anthony M Zador. "Shared and private variability in the auditory cortex". In: *Journal of neurophysiology* 92.3 (2004), pp. 1840–1855.

[60] Brent Doiron et al. "The mechanics of state-dependent neural correlations". In: *Nature neuroscience* 19.3 (2016), pp. 383–393.

[61] M. E. Dougherty et al. "Laminar origin of evoked ECoG high-gamma activity". In: *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. July 2019, pp. 4391–4394. DOI: 10.1109/EMBC.2019.8856786.

[62] Alexander S Ecker et al. "The effect of noise correlations in populations of diversely tuned neurons". In: *Journal of Neuroscience* 31.40 (2011), pp. 14272–14283.

[63] Robert C Emerson, Michael J Korenberg, and Mark C Citron. "Identification of complex-cell intensive nonlinearities in a cascade model of cat visual cortex". In: *Biological cybernetics* 66.4 (1992), pp. 291–300.

[64] A Aldo Faisal, Luc PJ Selen, and Daniel M Wolpert. "Noise in the nervous system". In: *Nature reviews neuroscience* 9.4 (2008), p. 292.

[65] Jianqing Fan and Runze Li. "Variable selection via nonconcave penalized likelihood and its oracle properties". In: *Journal of the American statistical Association* 96.456 (2001), pp. 1348–1360.

[66] Felix Franke et al. "Structures of neural correlation and how they favor coding". In: *Neuron* 89.2 (2016), pp. 409–422.

[67] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. "Regularization paths for generalized linear models via coordinate descent". In: *Journal of statistical software* 33.1 (2010), p. 1.

[68] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. "Regularization paths for generalized linear models via coordinate descent". In: *Journal of statistical software* 33.1 (2010), p. 1.

[69] Jerome H Friedman. "Greedy function approximation: a gradient boosting machine". In: *Annals of statistics* (2001), pp. 1189–1232.

[70] Karl Friston, Rosalyn Moran, and Anil K Seth. "Analysing connectivity with Granger causality and dynamic causal modelling". In: *Current opinion in neurobiology* 23.2 (2013), pp. 172–178.

[71] Ben D Fulcher and Alex Fornito. "A transcriptional signature of hub connectivity in the mouse connectome". In: *Proceedings of the National Academy of Sciences* 113.5 (2016), pp. 1435–1440.

[72] Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. "Efficient estimation of mutual information for strongly dependent variables". In: *Artificial intelligence and statistics*. 2015, pp. 277–286.

[73] Charles J Garfinkle and Christopher J Hillar. "On the uniqueness and stability of dictionaries for sparse representation of noisy signals". In: *IEEE Transactions on Signal Processing* 67.23 (2019), pp. 5884–5892.

[74] EdwardI George and Dean P Foster. "Calibration and empirical Bayes variable selection". In: *Biometrika* 87.4 (2000), pp. 731–747.

[75] Joshua I Glaser et al. "Machine learning for neural decoding". In: *arXiv preprint arXiv:1708.00909* (2017).

[76] Pinghua Gong and Jieping Ye. "A Modified Orthant-Wise Limited Memory Quasi-Newton Method with Convergence Analysis". In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. 2015, pp. 276–284.

[77] Robbe LT Goris, J Anthony Movshon, and Eero P Simoncelli. "Partitioning neuronal variability". In: *Nature neuroscience* 17.6 (2014), p. 858.

[78] W. H. Greene. *Econometric Analysis*. Pearson International Edition. Pearson Education, Limited, 2012.

[79] Bon-Mi Gu, Robert Schmidt, and Joshua D. Berke. "Globus pallidus dynamics reveal covert strategies for behavioral inhibition". In: *bioRxiv* (2020).

[80] Shi Gu et al. "Controllability of structural brain networks". In: *Nature communications* 6.1 (2015), pp. 1–10.

[81] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. Chapman and Hall/CRC, 2015.

[82] Christopher R Holdgraf et al. "Encoding and decoding models in cognitive electrophysiology". In: *Frontiers in systems neuroscience* 11 (2017), p. 61.

[83] Christopher J Honey et al. "Network structure of cerebral cortex shapes functional connectivity on multiple time scales". In: *Proceedings of the National Academy of Sciences* 104.24 (2007), pp. 10240–10245.

[84] Yu Hu, Joel Zylberberg, and Eric Shea-Brown. "The sign rule and beyond: boundary effects, flexibility, and noise correlations in neural population codes". In: *PLoS computational biology* 10.2 (2014), e1003469.

[85] Yu Hu, Joel Zylberberg, and Eric Shea-Brown. "The sign rule and beyond: boundary effects, flexibility, and noise correlations in neural population codes". In: *PLoS computational biology* 10.2 (2014), e1003469.

[86] Chengcheng Huang et al. "Circuit models of low-dimensional shared variability in cortical networks". In: *Neuron* 101.2 (2019), pp. 337–348.

[87] Shuai Huang and Trac D Tran. "Sparse signal recovery via generalized entropy functions minimization". In: *IEEE Transactions on Signal Processing* 67.5 (2018), pp. 1322–1337.

[88] D. H. Hubel and T. N. Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". In: *The Journal of Physiology* 160.1 (1962), pp. 106–154.

[89] Guy Isely, Christopher Hillar, and Fritz Sommer. "Deciphering subsampled data: adaptive compressive sampling as a principle of brain communication". In: *Advances in neural information processing systems*. 2010, pp. 910–918.

[90] Ramakrishnan Iyer et al. "The influence of synaptic weight distribution on neuronal population dynamics". In: *PLoS computational biology* 9.10 (2013).

[91] Adel Javanmard and Andrea Montanari. "Confidence intervals and hypothesis testing for high-dimensional regression". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 2869–2909.

[92] Krešimir Josić et al. "Stimulus-dependent correlations and population codes". In: *Neural computation* 21.10 (2009), pp. 2774–2804.

[93] MohammadMehdi Kafashan et al. "Scaling of information in large neural populations reveals signatures of information-limiting correlations". In: *bioRxiv* (2020).

[94] MohammadMehdi Kafashan et al. "Scaling of sensory information in large neural populations shows signatures of information-limiting correlations". In: *Nature communications* 12.1 (2021), pp. 1–16.

[95] Pentti Kanerva. "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors". In: *Cognitive computation* 1.2 (2009), pp. 139–159.

[96] Ingmar Kanitscheider, Ruben Coen-Cagli, and Alexandre Pouget. "Origin of information-limiting noise correlations". In: *Proceedings of the National Academy of Sciences* 112.50 (2015), E6973–E6982.

[97] Yan Karklin and Eero P Simoncelli. "Efficient coding of natural images with a population of noisy linear-nonlinear neurons". In: *Advances in neural information processing systems*. 2011, pp. 999–1007.

[98] Robert E. Kass et al. "Computational Neuroscience: Mathematical and Statistical Perspectives". In: *Annual Review of Statistics and Its Application* 5.1 (2018), pp. 183–214.

[99] Robert E. Kass et al. "Computational Neuroscience: Mathematical and Statistical Perspectives". In: *Annual Review of Statistics and Its Application* 5.1 (2018), pp. 183–214.

[100] Steven M Kay. *Fundamentals of statistical signal processing*. Prentice Hall PTR, 1993.

[101] Alexander JE Kell and Josh H McDermott. "Deep neural network models of sensory systems: windows onto the role of task constraints". In: *Current opinion in neurobiology* 55 (2019), pp. 121–132.

[102] Ryan C Kelly et al. "Local field potentials indicate network state and account for neuronal response variability". In: *Journal of computational neuroscience* 29.3 (2010), pp. 567–579.

[103] Roozbeh Kiani et al. "Dynamics of neural population responses in prefrontal cortex indicate changes of mind on single trials". In: *Current Biology* 24.13 (2014), pp. 1542–1547.

[104] Adam Kohn and Matthew A Smith. *Utah array extracellular recordings of spontaneous and visually evoked activity from anesthetized macaque primary visual cortex (V1).* 2016. URL: `http://dx.doi.org/10.6080/K0NC5Z4X`.

[105] Adam Kohn and Matthew A. Smith. "Utah array extracellular recordings of spontaneous and visually evoked activity from anesthetized macaque primary visual cortex (V1)". In: *CRCNS.org* (2016). URL: `http://dx.doi.org/10.6080/K0NC5Z4X`.

[106] Adam Kohn et al. "Correlations and neuronal population information". In: *Annual review of neuroscience* 39 (2016), pp. 237–256.

[107] Adam Kohn et al. "Correlations and neuronal population information". In: *Annual review of neuroscience* 39 (2016), pp. 237–256.

[108] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. "Estimating mutual information". In: *Physical review E* 69.6 (2004), p. 066138.

[109] Clemens Kreutz et al. "Profile likelihood in systems biology". In: *The FEBS Journal* 280.11 (2013), pp. 2564–2571.

[110] Jayant E. Kulkarni and Liam Paninski. "Common-input models for multiple neural spike-train data". In: *Network: Computation in Neural Systems* 18.4 (2007), pp. 375–407.

[111] Julie L. Lefebvre et al. "γ-Protocadherins regulate neuronal survival but are dispensable for circuit formation in retina". In: *Development* 135.24 (2008), pp. 4141–4151.

[112] J.Paul Leigh and Michael Schembri. "Instrumental variables technique: cigarette price provided better estimate of effects of smoking on SF-12". In: *Journal of Clinical Epidemiology* 57.3 (2004), pp. 284–293.

[113] Chinghway Lim and Bin Yu. "Estimation stability with cross-validation (ESCV)". In: *Journal of Computational and Graphical Statistics* 25.2 (2016), pp. 464–492.

[114] I-Chun Lin et al. "The nature of shared cortical variability". In: *Neuron* 87.3 (2015), pp. 644–656.

[115] Scott Linderman, Ryan P Adams, and Jonathan W Pillow. "Bayesian latent structure discovery from multi-neuron recordings". In: *Advances in Neural Information Processing Systems 29* 29 (2016), pp. 2002–2010.

[116] Ashok Litwin-Kumar et al. "Optimal degrees of synaptic connectivity". In: *Neuron* 93.5 (2017), pp. 1153–1164.

[117] Han Liu, Kathryn Roeder, and Larry Wasserman. "Stability approach to regularization selection (StARS) for high dimensional graphical models". In: *24th Annual Conference on Neural Information Processing Systems 2010, NIPS 2010*. 2010.

[118] Jesse A Livezey, Kristofer E Bouchard, and Edward F Chang. "Deep learning as a tool for neural data analysis: speech classification and cross-frequency coupling in human sensorimotor cortex". In: *PLoS computational biology* 15.9 (2019), e1007091.

[119] Wei Ji Ma et al. "Bayesian inference with probabilistic population codes". In: *Nature neuroscience* 9.11 (2006), p. 1432.

[120] Jakob H Macke, Lars Buesing, and Maneesh Sahani. "Estimating state and parameters in state space models of spike trains". In: *Advanced State Space Methods for Neural and Clinical Data*. Vol. 137. Cambridge University Press, 2015.

[121] Jakob H Macke et al. "Empirical models of spiking in neural populations". In: *Advances in neural information processing systems*. 2011, pp. 1350–1358.

[122] Joseph G Makin et al. "Superior arm-movement decoding from cortex with a new, unsupervised-learning algorithm". In: *Journal of Neural Engineering* 15.2 (Jan. 2018), p. 026010. DOI: 10.1088/1741-2552/aa9e95.

[123] Nikola T Markov et al. "Cortical high-density counterstream architectures". In: *Science* 342.6158 (2013), p. 1238406.

[124] Vivien Marx. *Biology: The big challenges of big data*. 2013.

[125] Francesca Melozzi et al. "Individual structural features constrain the mouse functional connectome". In: *Proceedings of the National Academy of Sciences* 116.52 (2019), pp. 26961–26969.

[126] Jorrit S Montijn et al. "Population-level neural codes are robust to single-neuron variability from a multidimensional coding perspective". In: *Cell reports* 16.9 (2016), pp. 2486–2498.

[127] Jorrit Steven Montijn et al. "Strong information-limiting correlations in early visual areas". In: *bioRxiv* (2019), p. 842724.

[128] Rubén Moreno-Bote et al. "Information-limiting correlations". In: *Nature neuroscience* 17.10 (2014), p. 1410.

[129] W. James Murdoch et al. "Definitions, methods, and applications in interpretable machine learning". In: *Proceedings of the National Academy of Sciences* 116.44 (2019), pp. 22071–22080. ISSN: 0027-8424. DOI: 10.1073/pnas.1900654116.

[130] S. A. Murphy and A. W. van der Vaart. "On Profile Likelihood". In: *Journal of the American Statistical Association* 95.450 (2000), pp. 449–465.

[131] Thomas Naselaris et al. "Encoding and decoding in fMRI". In: *Neuroimage* 56.2 (2011), pp. 400–410.

[132] Andrew A Neath and Joseph E Cavanaugh. "The Bayesian information criterion: background, derivation, and applications". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 4.2 (2012), pp. 199–203.

[133] Andrew A. Neath and Joseph E. Cavanaugh. "The Bayesian information criterion: background, derivation, and applications". In: *WIREs Computational Statistics* 4.2 (2012), pp. 199–203. DOI: https://doi.org/10.1002/wics.199. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wics.199. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/wics.199.

[134] John Ashworth Nelder and Robert WM Wedderburn. "Generalized linear models". In: *Journal of the Royal Statistical Society: Series A (General)* 135.3 (1972), pp. 370–384.

[135] Mark EJ Newman. "Modularity and community structure in networks". In: *Proceedings of the national academy of sciences* 103.23 (2006), pp. 8577–8582.

[136] Ramon Nogueira et al. "The effects of population tuning and trial-by-trial variability on information encoding and behavior". In: *Journal of Neuroscience* 40.5 (2020), pp. 1066–1083.

[137] Duane Q. Nykamp. "A mathematical framework for inferring connectivity in probabilistic neuronal networks". In: *Mathematical Biosciences* 205.2 (2007), pp. 204–251.

[138] Joseph E. O'Doherty et al. *Nonhuman Primate Reaching with Multichannel Sensorimotor Cortex Electrophysiology*. May 2017. DOI: 10.5281/zenodo.583331. URL: https://doi.org/10.5281/zenodo.583331.

[139] Murat Okatan, Matthew A Wilson, and Emery N Brown. "Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity". In: *Neural computation* 17.9 (2005), pp. 1927–1961.

[140] Michael Okun et al. "Diverse coupling of neurons to populations in sensory cortex". In: *Nature* 521.7553 (2015), pp. 511–515. DOI: 10.1038/nature14273.

[141] Bruno A Olshausen and David J Field. "Emergence of simple-cell receptive field properties by learning a sparse code for natural images". In: *Nature* 381.6583 (1996), pp. 607–609.

[142] Marino Pagan, Eero P Simoncelli, and Nicole C Rust. "Neural quadratic discriminant analysis: Nonlinear decoding with V1-like computation". In: *Neural computation* 28.11 (2016), pp. 2291–2319.

[143] Chethan Pandarinath et al. "Inferring single-trial neural population dynamics using sequential auto-encoders". In: *Nature methods* 15.10 (2018), pp. 805–815.

[144] Liam Paninski. "Maximum likelihood estimation of cascade point-process neural encoding models". In: *Network: Computation in Neural Systems* 15.4 (2004), pp. 243–262.

[145] Liam Paninski, Jonathan Pillow, and Jeremy Lewi. "Statistical models for neural encoding, decoding, and optimal stimulus design". In: *Computational Neuroscience: Theoretical Insights into Brain Function*. Vol. 165. Progress in Brain Research. Elsevier, 2007, pp. 493–507.

[146] Liam Paninski, Jonathan Pillow, and Jeremy Lewi. "Statistical models for neural encoding, decoding, and optimal stimulus design". In: *Computational Neuroscience: Theoretical Insights into Brain Function*. Vol. 165. Progress in Brain Research. Elsevier, 2007, pp. 493–507.

[147] Liam Paninski et al. "A new look at state-space models for neural data". In: *Journal of computational neuroscience* 29.1-2 (2010), pp. 107–126.

[148] Liam Paninski et al. "A new look at state-space models for neural data". In: *Journal of Computational Neuroscience* 29.1 (2010), pp. 107–126.

[149] Il Memming Park et al. "Encoding and decoding in parietal cortex during sensorimotor decision-making". In: *Nature Neuroscience* 17 (Aug. 2014).

[150] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[151] Jonathan W Pillow et al. "Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model". In: *Journal of Neuroscience* 25.47 (2005), pp. 11003–11013.

[152] Jonathan W Pillow et al. "Spatio-temporal correlations and visual signalling in a complete neuronal population". In: *Nature* 454.7207 (2008), p. 995.

[153] Rajesh PN Rao and Dana H Ballard. "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects". In: *Nature neuroscience* 2.1 (1999), pp. 79–87.

[154] Alfonso Renart et al. "The asynchronous state in cortical circuits". In: *science* 327.5965 (2010), pp. 587–590.

[155] Erin L Rich and Jonathan D Wallis. "Decoding subjective decisions from orbitofrontal cortex". In: *Nature Neuroscience* 19.7 (2016), pp. 973–980. DOI: 10.1038/nn.4320. URL: https://doi.org/10.1038/nn.4320.

[156] Fred Rieke et al. *Spikes: exploring the neural code*. Vol. 7. 1. MIT press Cambridge, 1999.

[157] Jorma Rissanen. "Modeling by shortest data description". In: *Automatica* 14.5 (1978), pp. 465–471.

[158] Douglas A Ruff and Marlene R Cohen. "Attention increases spike count correlations between visual cortical areas". In: *Journal of Neuroscience* 36.28 (2016), pp. 7523–7534.

[159] Trevor Ruiz et al. "Sparse, Low-bias, and Scalable Estimation of High Dimensional Vector Autoregressive Models via Union of Intersections". In: *arXiv:1908.11464* (2019).

[160] Oleg I Rumyantsev et al. "Fundamental bounds on the fidelity of sensory cortical coding". In: *Nature* 580.7801 (2020), pp. 100–105.

[161] P.S. Sachdeva. *neuronoise*. https://github.com/pssachdeva/neuronoise. 2018.

[162] Pratik Sachdeva et al. *PyUoI: The Union of Intersections Framework in Python*. https://github.com/BouchardLab/pyuoi. 2019.

[163] Pratik Sachdeva et al. "PyUoI: The Union of Intersections Framework in Python". In: *Journal of Open Source Software* 4.44 (2019), p. 1799.

[164] Pratik S Sachdeva, Jesse A Livezey, and Michael R DeWeese. "Heterogeneous synaptic weighting improves neural coding in the presence of common noise". In: *Neural computation* 32.7 (2020), pp. 1239–1276.

[165] Ko Sakai and Shigeru Tanaka. "Spatial pooling in the second-order spatial structure of cortical complex cells". In: *Vision Research* 40.7 (2000), pp. 855–871.

[166] Peter B Sargent et al. "Rapid vesicular release, quantal variability, and spillover contribute to the precision and reliability of transmission at a glomerular synapse". In: *Journal of Neuroscience* 25.36 (2005), pp. 8173–8187.

[167] Shlomo S Sawilowsky. "New effect size rules of thumb". In: *Journal of Modern Applied Statistical Methods* 8.2 (2009), p. 26.

[168] Almut Schüz et al. "Quantitative aspects of corticocortical connections: a tracer study in the mouse". In: *Cerebral cortex* 16.10 (2006), pp. 1474–1486.

[169] Odelia Schwartz et al. "Spike-triggered neural characterization". In: *Journal of vision* 6.4 (2006), pp. 13–13.

[170] Gideon Schwarz. "Estimating the Dimension of a Model". In: *Ann. Statist.* 6.2 (Mar. 1978), pp. 461–464.

[171] Carol A Seger. "How do the basal ganglia contribute to categorization? Their roles in generalization, response selection, and learning via feedback". In: *Neuroscience & Biobehavioral Reviews* 32.2 (2008), pp. 265–278.

[172] Terrence J Sejnowski, Patricia S Churchland, and J Anthony Movshon. "Putting big data to good use in neuroscience". In: *Nature neuroscience* 17.11 (2014), p. 1440.

[173] Anil K Seth, Adam B Barrett, and Lionel Barnett. "Granger causality analysis in neuroscience and neuroimaging". In: *Journal of Neuroscience* 35.8 (2015), pp. 3293–3297.

[174] Michael N Shadlen and William T Newsome. "The variable discharge of cortical neurons: implications for connectivity, computation, and information coding". In: *Journal of neuroscience* 18.10 (1998), pp. 3870–3896.

[175] Maoz Shamir and Haim Sompolinsky. "Implications of neuronal diversity on population coding". In: *Neural computation* 18.8 (2006), pp. 1951–1986.

[176] Jun Shao. "An asymptotic theory for linear model selection". In: *Statistica sinica* (1997), pp. 221–242.

[177] Jun Shao. "Linear model selection by cross-validation". In: *Journal of the American statistical Association* 88.422 (1993), pp. 486–494.

[178] Tatyana O. Sharpee. "Computational Identification of Receptive Fields". In: *Annual Review of Neuroscience* 36.1 (2013), pp. 103–120.

[179] Jack Sherman and Winifred J Morrison. "Adjustment of an inverse matrix corresponding to a change in one element of a given matrix". In: *The Annals of Mathematical Statistics* 21.1 (1950), pp. 124–127.

[180] Matthew A. Smith and Adam Kohn. "Spatial and Temporal Scales of Neuronal Correlation in Primary Visual Cortex". In: *Journal of Neuroscience* 28.48 (2008), pp. 12591–12603.

[181] Haim Sompolinsky et al. "Population coding in neuronal systems with correlated noise". In: *Physical Review E* 64.5 (2001), p. 051904.

[182] Dong Song et al. "Identification of sparse neural functional connectivity using penalized likelihood estimation and basis functions". In: *Journal of computational neuroscience* 35.3 (2013), pp. 335–357.

[183] Sen Song et al. "Highly nonrandom features of synaptic connectivity in local cortical circuits". In: *PLoS biology* 3.3 (2005).

[184] Klaas E Stephan et al. "Biophysical models of fMRI responses". In: *Current Opinion in Neurobiology* 14.5 (2004), pp. 629–635.

[185] Ian H Stevenson et al. "Inferring functional connections between neurons". In: *Current opinion in neurobiology* 18.6 (2008), pp. 582–588.

[186] Ian H Stevenson et al. "Statistical assessment of the stability of neural movement representations". In: *Journal of neurophysiology* 106.2 (2011), pp. 764–774.

[187] Ian H. Stevenson. "Omitted Variable Bias in GLMs of Neural Spiking Activity". In: *Neural Computation* 30.12 (2018), pp. 3227–3258.

[188] Ian H. Stevenson et al. "Functional Connectivity and Tuning Curves in Populations of Simultaneously Recorded Neurons". In: *PLoS Computational Biology* 8.11 (2012). DOI: 10.1371/journal.pcbi.1002775.

[189] Ian H. Stevenson et al. "Statistical assessment of the stability of neural movement representations". In: *Journal of Neurophysiology* 106.2 (2011), pp. 764–774.

[190] Gilbert W Stewart. "The efficient generation of random orthogonal matrices with an application to condition estimators". In: *SIAM Journal on Numerical Analysis* 17.3 (1980), pp. 403–409.

[191] Carsen Stringer et al. "High-dimensional geometry of population responses in visual cortex". In: *Nature* 571.7765 (2019), pp. 361–365.

[192] Jeffrey L. Teeters et al. "Data Sharing for Computational Neuroscience". In: *Neuroinformatics* 6.1 (Mar. 2008), pp. 47–55.

[193] Qawi K Telesford et al. "The ubiquity of small-world networks". In: *Brain connectivity* 1.5 (2011), pp. 367–375.

[194] Henri Theil. "Estimation of Parameters of Econometric Models". In: *Henri Theil's Contributions to Economics and Econometrics: Econometric Theory and Methodology*. Dordrecht: Springer Netherlands, 1992, pp. 109–116. ISBN: 978-94-011-2546-8. DOI: 10.1007/978-94-011-2546-8_7. URL: https://doi.org/10.1007/978-94-011-2546-8_7.

[195] F.E. Theunissen et al. "Estimating spatio-temporal receptive fields of auditory and visual neurons from their responses to natural stimuli". In: *Network: Computation in Neural Systems* 12.3 (2001), pp. 289–316.

[196] Robert Tibshirani. "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), pp. 267–288.

[197] DJ Tolhurst, J Anthony Movshon, and ID Thompson. "The dependence of response amplitude and variance of cat visual cortical neurones on stimulus contrast". In: *Experimental brain research* 41.3 (1981), pp. 414–419.

[198] Marcus A. Triplett and Geoffrey J. Goodhill. "Probabilistic Encoding Models for Multivariate Neural Data". In: *Frontiers in Neural Circuits* 13 (2019), p. 1.

[199] Wilson Truccolo et al. "A Point Process Framework for Relating Neural Spiking Activity to Spiking History, Neural Ensemble, and Extrinsic Covariate Effects". In: *Journal of Neurophysiology* 93.2 (2004), pp. 1074–1089. DOI: 10.1152/jn.00697.2004. arXiv: NIHMS150003.

[200] Wilson Truccolo et al. "A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects". In: *Journal of neurophysiology* 93.2 (2005), pp. 1074–1089.

[201] S. Ubaru, K. Wu, and K. E. Bouchard. "UoI-NMF Cluster: A Robust Nonnegative Matrix Factorization Algorithm for Improved Parts-Based Decomposition and Reconstruction of Noisy Data". In: *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Dec. 2017, pp. 241–248. DOI: 10.1109/ICMLA.2017.0-152.

[202] Sara A Van de Geer et al. "High-dimensional generalized linear models and the lasso". In: *The Annals of Statistics* 36.2 (2008), pp. 614–645.

[203] Michael Vidne et al. "Modeling the impact of common noise inputs on the network activity of retinal ganglion cells". In: *Journal of computational neuroscience* 33.1 (2012), pp. 97–121.

[204] Michael Vidne et al. "Modeling the impact of common noise inputs on the network activity of retinal ganglion cells". In: *Journal of computational neuroscience* 33.1 (2012), pp. 97–121.

[205] William E Vinje and Jack L Gallant. "Sparse coding and decorrelation in primary visual cortex during natural vision". In: *Science* 287.5456 (2000), pp. 1273–1276.

[206] Jeremiah D Wander et al. "Distributed cortical adaptation during learning of a brain–computer interface task". In: *Proceedings of the National Academy of Sciences* 110.26 (2013), pp. 10818–10823.

[207] Hansheng Wang, Runze Li, and Chih-Ling Tsai. "Tuning parameter selectors for the smoothly clipped absolute deviation method". In: *Biometrika* 94.3 (2007), pp. 553–568.

[208] Huifang E Wang et al. "A systematic framework for functional connectivity measures". In: *Frontiers in neuroscience* 8 (2014), p. 405.

[209] Larry Wasserman and Kathryn Roeder. "High dimensional variable selection". In: *Annals of statistics* 37.5A (2009), p. 2178.

[210] Duncan J Watts and Steven H Strogatz. "Collective dynamics of 'small-world'networks". In: *nature* 393.6684 (1998), p. 440.

[211] Xue-Xin Wei and Alan A Stocker. "Mutual information, Fisher information, and efficient coding". In: *Neural computation* 28.2 (2016), pp. 305–326.

[212] Frank Wilcoxon. "Individual comparisons by ranking methods". In: *Breakthroughs in statistics*. Springer, 1992, pp. 196–202.

[213] Stefan D Wilke and Christian W Eurich. "Representational accuracy of stochastic neural populations". In: *Neural computation* 14.1 (2002), pp. 155–189.

[214] Si Wu, Hiroyuki Nakahara, and Shun-Ichi Amari. "Population coding with correlation and an unfaithful model". In: *Neural Computation* 13.4 (2001), pp. 775–797.

[215] Okito Yamashita et al. "Sparse estimation automatically selects voxels relevant for the decoding of fMRI activity patterns". In: *NeuroImage* 42.4 (2008), pp. 1414–1429.

[216] Stuart Yarrow, Edward Challis, and Peggy Seriès. "Fisher and Shannon information in finite neural populations". In: *Neural computation* 24.7 (2012), pp. 1740–1780.

[217] Hyoungsoo Yoon and Haim Sompolinsky. "The effect of correlations on the Fisher information of population codes". In: *Advances in neural information processing systems*. 1999, pp. 167–173.

[218] Bin Yu et al. "Stability". In: *Bernoulli* 19.4 (2013), pp. 1484–1500.

[219] Shan Yu et al. "A small world of neuronal synchrony". In: *Cerebral cortex* 18.12 (2008), pp. 2891–2901.

[220] Arnold Zellner and Franz Palm. "Time series analysis and simultaneous equation econometric models". In: *Journal of Econometrics* 2.1 (1974), pp. 17–54.

[221] Yi-Feng Zhang, Hiroki Asari, and Markus Meister. *Multi-electrode recordings from retinal ganglion cells*. 2014. URL: `http://dx.doi.org/10.6080/K0RF5RZT`.

[222] Yiyun Zhang, Runze Li, and Chih-Ling Tsai. "Regularization parameter selections via generalized information criterion". In: *Journal of the American Statistical Association* 105.489 (2010), pp. 312–323.

[223] Mengyuan Zhao et al. "An L 1-regularized logistic model for detecting short-term neuronal interactions". In: *Journal of computational neuroscience* 32.3 (2012), pp. 479–497.

[224] Bo Zhou et al. "A dynamic bayesian model for characterizing cross-neuronal interactions during decision-making". In: *Journal of the American Statistical Association* 111.514 (2016), pp. 459–471.

[225] Mengchen Zhu and Christopher J Rozell. "Visual nonclassical receptive field effects emerge from sparse coding in a dynamical system". In: *PLoS computational biology* 9.8 (2013).

[226] Ehud Zohary, Michael N Shadlen, and William T Newsome. "Correlated neuronal discharge rate and its implications for psychophysical performance". In: *Nature* 370.6485 (1994), p. 140.

[227] Joel Zylberberg et al. "Direction-selective circuits shape noise to ensure a precise population code". In: *Neuron* 89.2 (2016), pp. 369–383.

[228] Joel Zylberberg et al. "Robust information propagation through noisy neural circuits". In: *PLoS computational biology* 13.4 (2017), e1005497.