Neural network adaptive coding efficiency and stochastic representational geometry

by

Lyndon Duong

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Center for Neural Science

New York University

September, 2023

Dr. Eero P. Simoncelli

Dr. David J. Heeger

DEDICATION

To my parents and siblings, whose journey in life has been my constant source of inspiration.

Acknowledgments

I am deeply grateful to my advisor, Eero Simoncelli. His humbling ingenuity and unconditional support have helped me grow both as a scientist and person. Eero's talent for clearly communicating scientific ideas has profoundly impacted my development as a researcher, and sits second only to the espresso brewing wisdom he's imparted. I'd also like to extend my gratitude to my co-advisor, David Heeger, whose advice and practical approach to science have been instrumental in my scientific development. David's guidance and insights have not only enriched my understanding but also inspired me to pursue innovative ideas and embrace a pragmatic mindset.

At NYU, my sincere thanks go to my committee members, Mike Landy and Tony Movshon, for their insightful comments and career advice. Their thoughtful feedback on my work has been instrumental in refining my approach to research. I'd also like to thank Jess Holman for administrative support, and Paul Fan for technical support through my PhD. Thanks to Shivang Rawat and Stefano Martiniani for their insightful discussions and constructive critiques of my research. Lastly, I'm immensely grateful for the camaraderie of my NYU Neuroscience cohort: Anya Krok, Ionatan Kuperwajs, Andrew Mah, Pat O'Neill, Ravi Pancholi, Lauren Ryan, Shannon Schiereck, and Klavdia Zemlianova, each of whom have made this journey all the more fulfilling with our shared struggle and friendship.

To those in the Lab for Computational Vision who I leaned on most: Hope Lutwak, a smart and sweet soul, and one of my dearest friends whom I've known since day one (NYU CNS interviews!); Teddy Yerxa, my brother and top-2 favorite Flatiron office-mate; Nikhil Parthasarathy,

my Big Tech comrade and rock climbing Sherpa; Pierre-Étienne Fiquet, my tall encyclopédie mathématique of an office neighbor, always warmly welcoming my frequent procrastination-driven distraction; Colin Bredenberg, an inspirational productivity machine (who was instrumental to Chapter 4 of this dissertation); Jenelle Feather, our newest addition, who is always fun to chat with; and Zahra Kadkhodaie, an incredibly talented researcher and irreplaceable lunch/-coffee/tea companion.

Thanks to the Simons Foundation for scientific, administrative, and funding support. Much of the work in this thesis was made possible by teaming up with the meticulous mathematician and all-round good guy, David Lipshutz; I'm proud to call him both a collaborator and a friend. I'm also thankful to have worked with Alex Williams, a fire hose of modern statistics knowledge, who was the driving force behind Chapter 5 of this thesis. I am forever grateful to our world-class admin team, who are all wonderfully friendly and kind: Brooklyn buddy and travel wizard, Jessica Hauser; fellow early-morning warrior, Noah Dlugacz; and the unstoppable force of nature and kindred sci-fi fan that is Matthew Turner.

To my friends, old and new: Rob & Brooke, Chao, Rishi, Mat & Ally, Sara & Andy, Jules, Brendan, Nora, Kylee, Colleen, Borna, Ben, Megan, John, Mark, Francis & Maryse, Billy, and James. Thank you for your love and shared laughter over the years, and reminding me of life outside of the lab. My parents, Bac Ai Duong and Huong Le, deserve the biggest thank you for their boundless support and love. Their encouragement has been instrumental in shaping my journey. I also extend my heartfelt appreciation to my big sister and brother, Kim and Tino, for their continuous support.

Finally, I want to thank Paige Leary for her unwavering love and support throughout this journey. She has been a constant source of patience, understanding, motivation, and encouragement, fueling my determination to overcome every hurdle and achieve my goals.

Abstract

This dissertation investigates two fundamental aspects of neural population coding: adaptive coding efficiency and stochastic representational geometry. We introduce a theory for adaptive statistical whitening revolving around a gain control mechanism, based on a novel overcomplete matrix factorization of the whitening transform. From this theory, we derive an online whitening algorithm that maps directly onto a recurrent neural network with primary neurons and an overcomplete, auxiliary set of gain-modulating interneurons. Further elaborating on this framework, we integrate adaptive gain control with existing theories of adaptive whitening into a single unified adaptation objective using synaptic plasticity in a multi-timescale mechanistic model. This model adapts to changing sensory statistics by modifying gains and synapses at varying rates, resulting in improved adaptive whitening responses that is robust to non-stationary environments. Leveraging V1 population adaptation data, we demonstrate that propagation of single neuron gain changes through recurrent network structures is sufficient to explain the entire set of observed adaptation effects. Finally, we shift our focus to stochastic representational geometry, and introduce a family of distance metrics for comparing geometry between stochastic neural networks. These metrics are based on concepts from optimal transport theory and provide unique insights into the representations of noisy artificial and biological neural networks. Taken together, this thesis advances our understanding of neural population coding by examining the adaptive coding efficiency and the stochastic geometry of neural representations, with possible implications to the fields of neuroscience and machine learning.

Contents

List of Figures

List of Appendices

1 Introduction

1.1 Adaptive Coding Efficiency in Neural Populations

1.1.1 The efficient coding hypothesis

Coding efficiency is deeply rooted in the principles of information theory, and is a common goal in the design of machine systems. At its conception, information theory was used to determine fundamental bounds for signal processing algorithms in compressing, transmitting, and storing data (Cover, 1999). Interestingly, these concepts were recognized as applicable to perceptual and biological systems as well. Early studies proposed that our sensory and perceptual systems should leverage statistical regularities in the natural environment to optimally encode sensory information (Attneave, 1954; Barlow, 1961).

The efficient coding hypothesis proposed by Barlow (1961), postulates that sensory systems have evolved to maximize the efficiency of information coding, subject to metabolic and physical constraints. From an information-theoretic perspective, the hypothesis states that sensory neurons are optimized to represent the most information possible about their probabilistic input distribution (e.g., the statistics of the natural environment), subject to constraints such as energy expenditure from spiking.

The efficient coding hypothesis can intuitively be understood at the level of a single neuron encoding a scalar input using a continuous response function (e.g. visual contrast). This response

function has a minimum and maximum value, and the neuron's response is assumed to be deterministic. The occurrence frequency of input values in the environment is represented by a probability distribution. In this scenario, the optimal response function that maximizes information is directly proportional to the cumulative probability distribution of the input (Figure 1.1). The neuron thereby dedicates more sensitive regions of its response range to accurately encode frequently encountered inputs, but sacrifices fidelity in encoding less common inputs. The foundational study by Laughlin (1981) measured the probability distribution of stimulus intensity levels in natural scenes and discovered that the intensity-response function of a large monopolar cell in a fly closely resembled the cumulative distribution.

While Barlow's theory was predicated on the notion that sensory systems have evolved to efficiently encode the statistical properties of the natural environment (Ganguli and Simoncelli, 2014), it remains unclear how his theory can be reconciled with the fact that natural environments are in general *non-stationary*. Statistical properties of the natural world dynamically vary with time (Młynarski and Hermundstad, 2021); this can be due to slow-timescale changes (day-night cycles), more immediate changes in the environment (e.g. stepping into mirror fun-house at the carnival), or task demands. The existence of dynamic sensory statistics suggests that an efficient code at one moment may not necessarily be efficient for the next, and that a truly efficient system would need to be constantly adapting to these changes (Barlow and Foldiak, 1989).

1.1.2 The neural basis of adaptation

For nearly a century, it has been known that individual neurons rapidly regulate their sensitivity, also known as their *gain*, based on their recent response history (Adrian and Matthews, 1928a). This adaptive behavior enables neurons to normalize the variance of their outputs (Bonin et al., 2006; Nagel and Doupe, 2006), thereby maximizing the information transmitted about sensory inputs (Barlow, 1961; Fairhall et al., 2001; Laughlin, 1981). This is illustrated in Figure 1.1. With changing input distribution (blue densities), a neuron adjusts its gain (the slope of its input-

output function) to normalize its response statistics (e.g. Nagel and Doupe, 2006).

At the neural population level, beyond variance normalization, there have been reports of adaptive transformations across different species and sensory modalities. These observed effects encompass various phenomena, such as reductions in response maxima and minima (Movshon and Lennie, 1979), tuning curve repulsion (Hershenhoren et al., 2014; Shen et al., 2015; Yaron et al., 2012), and stimulus-driven decorrelation (Benucci et al., 2013; Friedrich, 2013; Gschwend et al., 2015; Muller et al., 1999; Wanner and Friedrich, 2020). While coding efficiency and gain-mediated adaptation have been extensively studied in single neurons, these nuanced empirical observations appear to necessitate a more intricate adaptation mechanism that involves *coordinated interactions* among neurons within the population. Previous studies have indeed relied on adaptive changes in feedforward or recurrent synaptic efficacy to account for these phenomena (e.g., by altering the synaptic weights of the entire network; Młynarski and Hermundstad, 2021; Rast and Drugowitsch, 2020; Wainwright et al., 2001; Westrick et al., 2016). However, this approach requires synaptic weights to *continually remap* under different statistical contexts, which can undergo significant and transient changes at short timescales. This proposed adaptation mechanism stands in stark contrast to the single neuron scenario that simply relies solely on gain rescaling.

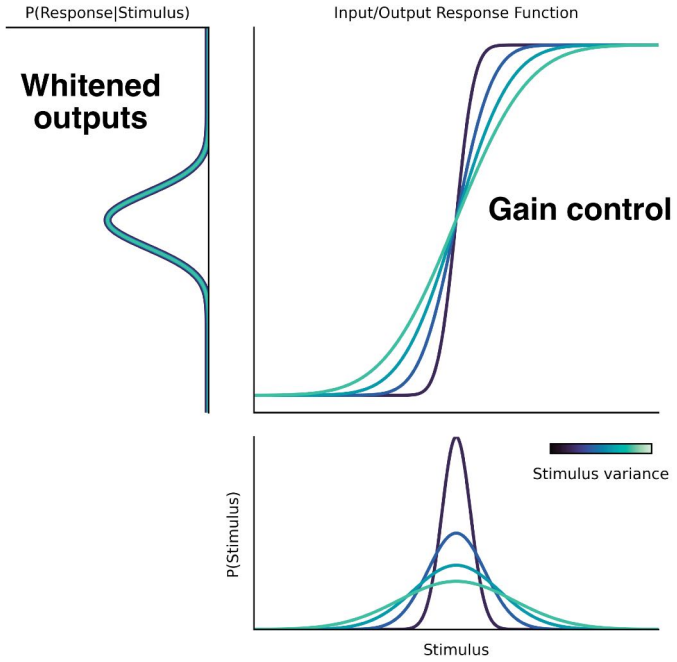

Figure 1.1: Schematic illustrating single-neuron adaptation results from (Fairhall et al., 2001; Nagel and Doupe, 2006). Gain control confers adaptive coding efficiency in single neurons. Despite changing input statistics (densities with different variances, bottom), neuron output statistics are unaffected (left) due to the neuron modulating its input-output function slope, i.e. gain.

In this thesis, we introduce novel theories which generalize single-neuron gain adaptation to the level of a population.

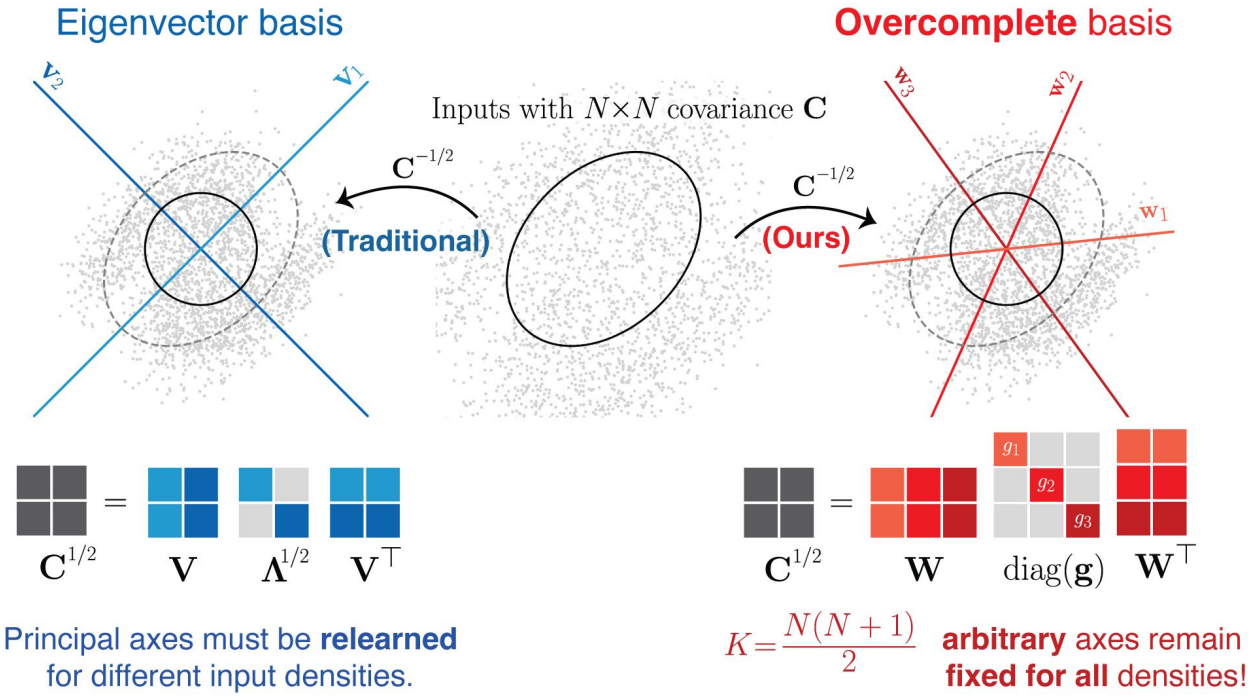

Figure 1.2: Factorizations of the (inverse) symmetric whitening matrix. Left: conventional eigenvector-based whitening. Right: an example of the overcomplete factorization framework proposed in this thesis.

1.1.3 Statistical whitening transformations

Redundancy reduction, i.e. reducing inter-channel statistical dependencies in a multi-channel neural code, is core to Barlow's efficient coding hypothesis. Empirical observations have shown that neural populations exhibit adaptive decorrelation and *statistical whitening* (e.g. Wanner and Friedrich, 2020), which is a specific form of redundancy reduction. For multivariate Gaussian variables, which are fully characterized by their first and second statistical moments, decorrelation is necessary and sufficient to conclude that each channel is *independent* (Bishop, 2006). By decorrelating neural responses, the population ensures that each neuron conveys unique and independent information. With this lack of redundancy, neural populations maximize the infor-

mation content conveyed by the ensemble of neurons, enabling more efficient representation of sensory inputs (Cover, 1999).

Statistical whitening is a commonly used linear transformation in signal processing and statistical machine learning. Formally, an N-dimensional neural population response, $\mathbf{r} \sim p(\mathbf{r})$, is white if and only if its covariance $\mathbf{C} = \mathbb{E}[\mathbf{r}\mathbf{r}^\top] - \mathbb{E}[\mathbf{r}]\mathbb{E}[\mathbf{r}]^\top = \mathbf{I}_N$. From this we can see that statistical whitening serves to both decorrelate neurons (as indicated by off-diagonal entries of $\mathbf{C}$ being zero), in addition to normalizing variances (diagonal entries are 1). Thus, a neural system whose objective is to adaptively whiten its outputs can be interpreted as a population-level generalization of the adaptive variance normalization effects observed in single neurons.

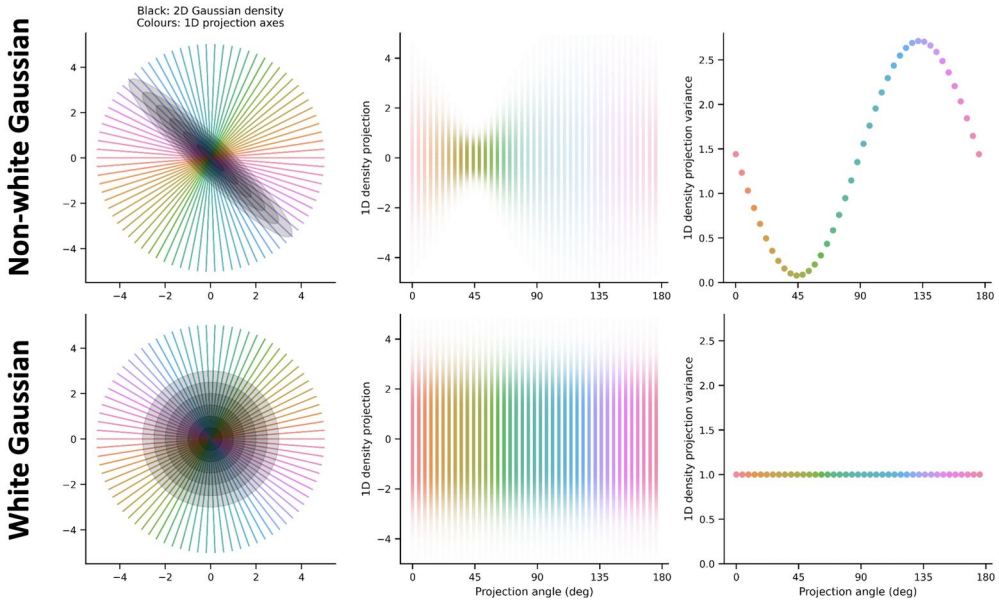

Figure 1.3: Representing a Gaussian using 1D projections. **Top:** A non-white 2D Gaussian, and a set of 1D projection axes (colors). These 1D marginal densities (middle column) encode the original 2D density. Because the density is *not* white, the marginal variances (right panel) are not all equal to 1. **Bottom:** A white Gaussian has variance of 1 along all possible directions. However, only $K = N(N+1)/2$ projections are necessary to conclude a density is white. In this figure where $N = 2$, 3 projections are required. See Chapter 2, Proposition 2.1 for details.

Whitening is not a unique transformation: any orthogonal rotation of a random vector with an identity covariance matrix will also result in an identity covariance matrix. Various approaches exist to address this rotational ambiguity, each offering its own benefits (Kessy et al., 2018). In

this thesis, we specifically focus on the symmetric whitening transformation, commonly known as Zero-phase Component Analysis (ZCA) whitening or Mahalanobis whitening. This transform is the unique solution to the objective seeking to minimize the mean squared error between the inputs and the whitened outputs (Appendix A.1).

The symmetric whitening transform is a matrix that can obtained through the eigendecomposition of the covariance matrix, $\mathbf{C} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathbf{T}}$, where $\mathbf{V}$ is a matrix of eigenvectors (i.e. principal axes) and $\mathbf{\Lambda}$ is a diagonal matrix of corresponding eigenvalues. To obtain the unique, symmetric whitening transform, we take the inverse PSD matrix square root of the covariance matrix, $\mathbf{M} \coloneqq \mathbf{V}\mathbf{\Lambda}^{-1/2}\mathbf{V}^{\top}$. Then, the whitened responses are computed as $\mathbf{Mr}$ (Figure 1.2, left).

Neural circuit models of adaptive whitening typically operate via synaptic plasticity, modifying between-neuron connections in response to novel stimulus statistics (Lipshutz et al., 2023; Pehlevan et al., 2015). These models effectively encode the eigenvectors $\mathbf{V}$ into the synaptic weights of the network. However, because the statistics of natural environments are dynamic, this implies that the principal axes $\mathbf{V}$, and therefore the synaptic weights, must *constantly change* to adaptively whiten newly-observed contexts. If input statistics were to return to a previously-observed state (e.g. statistical context $A \longrightarrow B \longrightarrow A$), then the network must *re-learn* the optimal weights corresponding to the original condition. This lack of stability and reversibility poses a challenge for achieving long-term and robust adaptive whitening in neural circuit models.

In Chapter 2, we introduce a completely different approach to adaptive whitening which obviates adapting synaptic weights altogether. We propose a novel, *overcomplete* factorization of the (inverse) whitening matrix, $\mathbf{W}\,\mathrm{diag}\,(\mathbf{g})\,\mathbf{W}^{\top}$, where $\mathbf{W} \in \mathbb{R}^{N \times N(N+1)/2}$, which allows the network to adaptively whiten its inputs exclusively using single-neuron *gains* (elements of $\mathbf{g}$), thereby allowing the synaptic weights, $\mathbf{W}$, to remain *fixed* during adaptation.

Intuitively this factorization relies on the fact that any (zero-mean) Gaussian is entirely described by its covariance, and thus has $N(N+1)/2$ degrees of freedom (the number of unique entries of a symmetric matrix), and can therefore be encoded in an overcomplete set of $N(N+1)/2$

scalar projections. This is referred to as "overcomplete" because $N(N+1)/2 > N$, the dimensionality of the space. Thus, measuring variances along $N(N+1)/2$ unique axes is sufficient to encode the covariance of an N-dimensional density. For the case where $N = 2$ (Figure 1.3), only 3 projections are required. Furthermore, as we will demonstrate in Chapter 2, measuring unity variance along $N(N+1)/2$ unique projections is *necessary and sufficient* to conclude that a density is statistically white. We exploit this geometric insight in forming our novel matrix factorization for adaptive whitening, an example of which is shown in Figure 1.2 (right). Elaborating on this idea in Chapter 3, we *unify* the two seemingly disparate concepts of gain control and synaptic plasticity into a single adaptive whitening framework using both synaptic plasticity and gain control, each operating over separate timescales.

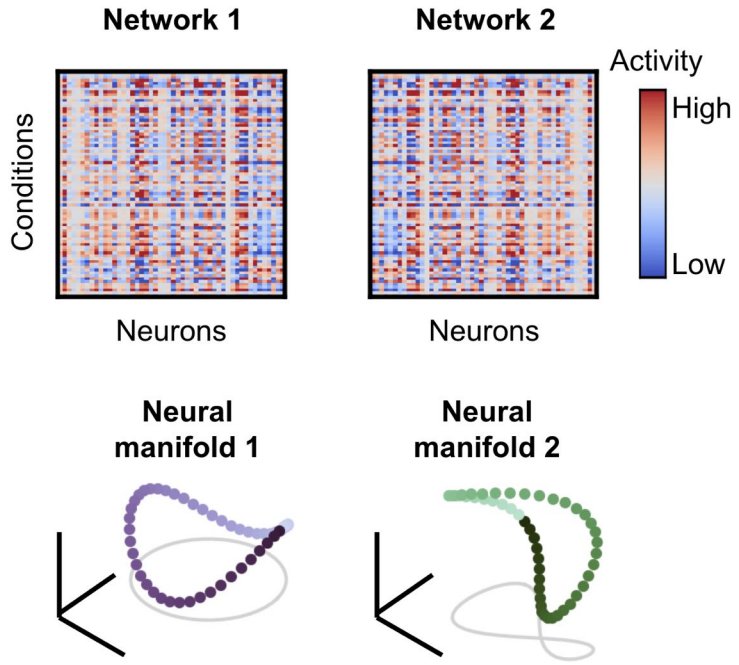

Figure 1.4: Two non-stochastic multi-neuron network responses to several conditions (upper), and their low-dimensional response representations, depicted as Pringle-shaped manifolds (lower). Each dot is a single condition. The green Pringle is a rotated, slightly warped version of the purple Pringle. Modified from Williams et al. (2021).

1.2 Stochastic Representational Geometry

1.2.1 Comparing neural representational geometry

Neural representational geometry refers to the arrangement of response patterns that encode information about stimuli or other variables (e.g. behavioral state, time, etc.) in the activations neural networks (Figure 1.4). This generally involves studying how the responses of neural populations are organized in a high-dimensional space (Chung and Abbott, 2021; Kriegeskorte and Wei, 2021). With the advent of modern recording tools capable of capturing signals from hundreds to thousands of neurons simultaneously, the study of representational geometry has catalyzed our understanding of neural computations at the population level. This framework has revealed that individual networks - be they different animals, distinct brain areas, or artificial neural networks trained with different initializations - can form unique representational geometries even for a common task. Thus, despite these advancements, a key challenge persists: quantitatively *comparing* the representational geometry between neural populations.

Numerous methods have been proposed to facilitate such comparisons. Pioneering work introduced Representational Similarity Analysis (RSA) (Kriegeskorte et al., 2008a), drawing inspiration from cognitive psychology analyses (Edelman et al., 1998). Alternative techniques have also been explored, including Canonical Correlations Analysis (CCA) (Gallego et al., 2020; Raghu et al., 2017) and its generalizations (reviewed in Zhuang et al., 2020), Procrustes alignment (Degenhart et al., 2020), and hyperalignment (Haxby et al., 2020). In machine learning, studies have examined representational geometry of deep neural networks using similar approaches, notably Centered Kernel Alignment (CKA) (Kornblith et al., 2019), which has been noted to share similarities with a variant of RSA (Williams et al., 2021, Appendix C). Notably, all of these techniques ignore trial-to-trial variability (stochasticity) in neural responses.

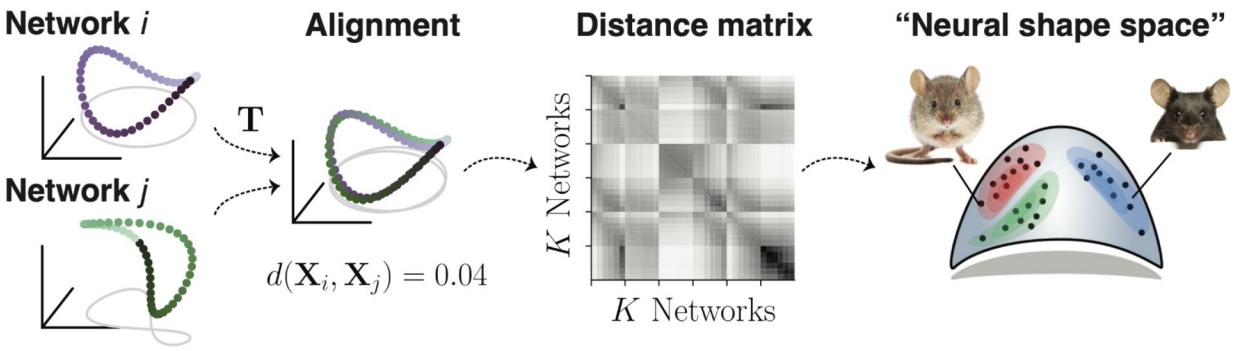

Figure 1.5: Aligning (deterministic) neural representations using shape metrics. Two network representations are aligned through some nuisance transformation (here, a rotation) T (first and second column). The remaining difference after accounting for this nuisance transformation is the distance $d(\mathbf{X}_i, \mathbf{X}_j)$ between them. We can repeat this procedure for $\binom{K}{2}$ network pairs in our dataset to form a $K \times K$ distance matrix (third column). We can visualize each K network as a point in "shape space" by embedding the distance matrix using standard tools such as multidimensional scaling (right column). Because the distances in this matrix were computed using a bona fide metric, this enables downstream analyses such as nearest-neighbors clustering with theoretical guarantees on correctness (Cover and Hart, 1967).

1.2.2 Shape metrics on deterministic neural representations

Here, we review a framework for comparing deterministic (i.e. non-stochastic) neural networks, called *shape metrics*, recently proposed by Williams et al. (2021). Consider a large dataset of K sets of simultaneously recorded neurons, which could denote different animals, recording sessions from the same animal, or neurons from varied brain regions. Each set consists of N neurons, observed over M task conditions, with L repeated trials for every condition. In practice, different recording sessions may yield different neural population sizes, implying variation in the value of N; here, we consider equal numbers of neurons, but the framework is general enough to handle unequal populations.

The analysis starts by computing trial-averaged neural responses within each session. The data can then be interpreted as a series of $M \times N$ matrices, designated as $\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_K$. Conceptually, each column of $\boldsymbol{X}_k$ corresponds to a neuron's tuning curve measured in neural network k (Figure 1.4). Shape metrics devise functions $d(\boldsymbol{X}_i, \boldsymbol{X}_j)$ that evaluate the difference between two sets of tuning curves.

It is useful to consider concepts of distance that comply with the following three properties:

$$\text{Equivalence:} \quad d\left(\boldsymbol{X}_i, \boldsymbol{X}_j\right) = 0 \quad \text{if and only if} \quad \boldsymbol{X}_i \text{ is equivalent to } \boldsymbol{X}_j \tag{1.1}$$

$$\text{Symmetry:} \quad d\left(\boldsymbol{X}_i, \boldsymbol{X}_j\right) = d\left(\boldsymbol{X}_j, \boldsymbol{X}_i\right) \tag{1.2}$$

$$\text{Triangle Inequality:} \quad d\left(\boldsymbol{X}_i, \boldsymbol{X}_j\right) \leq d\left(\boldsymbol{X}_i, \boldsymbol{X}_k\right) + d\left(\boldsymbol{X}_k, \boldsymbol{X}_j\right) \tag{1.3}$$

Distances adhering to these properties are formally referred to as *metrics*, and are used to define metric spaces. Most of the methods listed in the previous subsection are *not* metrics. These spaces are useful because they allow us to conceptualize each network responses as a single point in some abstract space of all possible network responses. Specifically, we can imagine our recordings from K neural networks $\boldsymbol{X}_1, ..., \boldsymbol{X}_K$ as points within the space of all potential neural recordings, which we term "neural shape space". There are many readily available statistical methods that function on a matrix of pairwise distances, $\boldsymbol{D}_{ij} = d(\boldsymbol{X}_i, \boldsymbol{X}_j)$. For instance, we might be interested in identifying clusters of animals with similar neural responses. To accomplish this, we can use hierarchical clustering methods, which work with theoretical guarantees in metric spaces. Moreover, we can use multidimensional scaling to visualize data across animals in low-dimensional spaces and rigorously quantify how well these vector embeddings represent distances in the original metric space (Figure 1.5, right). Finally, we might measure a behavioral variable for each animal (e.g., performance on a task) that we want to predict from neural data.

Defining distance functions $d(\boldsymbol{X}_i, \boldsymbol{X}_j)$ is a complex task because simple measures like Euclidean distance are effectively meaningless whenever neurons are not identified and matched one-to-one across recordings. This suggests a need to at the very least account for *misalignment* between neural representations before quantifying their distance. Williams et al. (2021) suggest a constrained set of linear isometric transformations (e.g. permutations, rotations, etc.), parameterized by a matrix $\boldsymbol{Q}$, to achieve this alignment. Furthermore, since it's typical to preprocess neural data (for instance, by z-scoring each neuron), they propose applying a user-defined function $\phi(\cdot)$

before fitting the alignment. Putting these together, we arrive at a general formula for a neural representational distance:

$$d(\boldsymbol{X}_i, \boldsymbol{X}_j) = \min_{Q \in \mathcal{G}} \|\phi(\boldsymbol{X}_i) - \phi(\boldsymbol{X}_j)\mathbf{Q}\|_F \tag{1.4}$$

where $\mathcal{G}$ is the set of allowable linear transformations. This equation defines a family of metrics referred to as generalized shape metrics, as they expand on classical notions of shape distance, such as Procrustes distance (Dryden and Mardia, 2016). Each set of network responses is interpreted as a high-dimensional geometric *shape*. Quantifying similarity between these shapes should be agnostic to a defined set of nuisance transformations defined in $\mathcal{G}$ (determined by the researcher), such as neuron permutations, or arbitrary rotations/reflections (Figure 1.5).

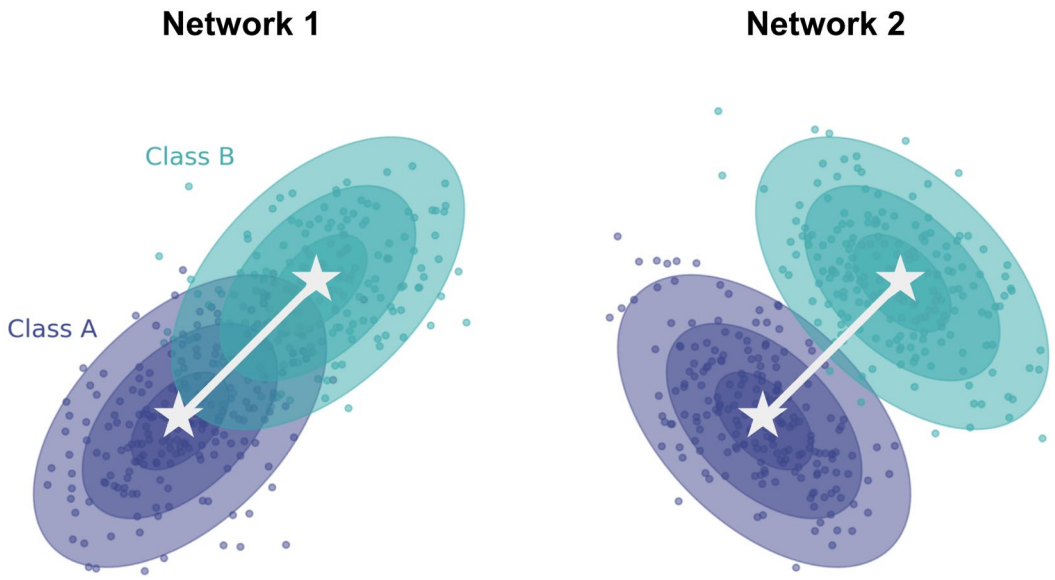

Figure 1.6: Noise correlations impact neural population coding. Two networks (left and right) have the same mean responses for a navy and cyan input (white stars), but different noise correlations, as indicated by the orientation of the ellipses. The white line indicates the direction of the linear discriminant.

1.2.2.1 Stochasticity and noise correlations

Existing methods comparing representational geometry, including the shape metrics framework described above, focus on deterministic responses, and therefore *ignore* stochasticity in neural responses; however, it is well known that neural responses are generally correlated on a trial-to-trial basis (Averbeck et al., 2006). Figure 1.6 shows two different neural populations with identical class-conditional means and different noise correlations. In this example, noise correlations parallel to the linear discriminant of the two classes hinder downstream linear classification performance (Moreno-Bote et al., 2014). Given how noise correlations are pervasive throughout the brain, we consider it of paramount importance to be able to quantify differences in *noisy* representational geometry between networks.

1.2.3 Optimal transport

Optimal transport theory provides a powerful mathematical framework for understanding and quantifying differences between different probability densities (Cuturi, 2013). By modeling neural responses as (conditional) probability distributions, optimal transport allows us to analyze the similarities and differences between these distributions, enabling insights into the underlying mechanisms of information processing in the brain. In Chapter 5, we integrate ideas from optimal transport into the shape metrics methodology to create a general framework for comparing the representational geometry of *stochastic* neural networks. The two optimal transport distances we focus on in this thesis are the Wasserstein and Energy distances.

1.2.3.1 Wasserstein distance

The Wasserstein distance, also known as the Earth Mover's distance or the Kantorovich-Rubinstein distance, is an example of a metric used to quantify the dissimilarity between probability distributions (Villani, 2009). It provides a measure of how much "mass" must be moved

from one distribution to another to transform the former into the latter.

The p-Wasserstein distance between two probability densities P and Q on a metric space $\mathcal{X}$ is

$$W_p(P,Q) = \left(\inf_{\gamma \in \Pi(P,Q)} \int_{\mathcal{X}\times\mathcal{X}} d(x,y)^p \, \gamma(x,y) \, dx \, dy \right)^{1/p}, \tag{1.5}$$

where $\Pi(P,Q)$ represents the set of all joint probability measures on $\mathcal{X}\times\mathcal{X}$ with marginals P and Q, and $d(x,y)$ is the metric or distance function on $\mathcal{X}$. Note that the integrals are taken over the entire space $\mathcal{X}$ and $dx\,dy$ represents the integration with respect to the joint measure γ.

In this thesis, we consider when $p = 2$ for *multivariate Gaussian densities*, and distance functions $d(x,y) = |x - y|$. The 2-Wasserstein distance for multivariate Gaussians is particularly attractive due to its analytically tractable form that decomposes into the sum of a metric measuring the distance on means, and a metric measuring the distance on covariances. For two multivariate Gaussian densities, $P = \mathcal{N}(\boldsymbol{\mu}_P, \Sigma_P)$ and $Q = \mathcal{N}(\boldsymbol{\mu}_Q, \Sigma_Q)$, with mean vectors $\boldsymbol{\mu}_P$ and $\boldsymbol{\mu}_Q$, and covariance matrices Σ_P and Σ_Q, the 2-Wasserstein distance is

$$\mathcal{W}_2(\boldsymbol{\mu}_P, \boldsymbol{\mu}_Q)^2 = \|\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q\|_2^2 + \operatorname{trace}\left(\Sigma_P + \Sigma_Q - 2\left(\Sigma_Q^{1/2} \Sigma_P \Sigma_Q^{1/2} \right)^{1/2} \right) \tag{1.6}$$

The first term is simply the squared Euclidean distance between the means of each distribution, while the second term is precisely the squared Bures distance between Σ_P and Σ_Q.

1.2.3.2 The Bures distance on covariance matrices

The Bures distance is a metric used to quantify the dissimilarity between symmetric positive definite matrices, such as covariance matrices. When considering zero-mean multivariate Gaussian distributions, the Bures distance provides a way to compare these distributions in terms of their covariance matrices. Specifically, for two zero-mean multivariate Gaussian distributions P and Q with covariance matrices Σ_P and Σ_Q, the Bures distance, denoted as $\mathcal{B}(P,Q)$, and its square

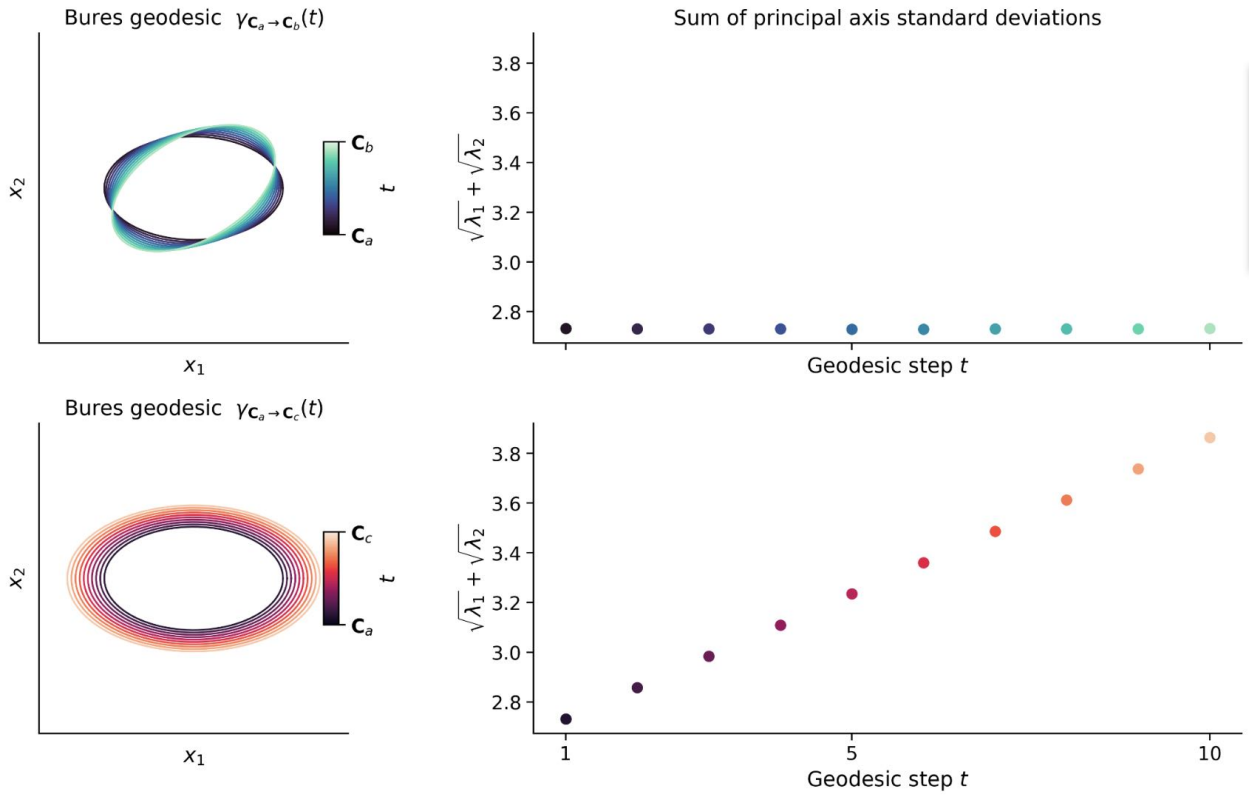

Figure 1.7: Bures metric interpolation. Families of densities which linearly interpolate the Bures distance between two covariances $\mathbf{C}_a \to \mathbf{C}_b$, and $\mathbf{C}_a \to \mathbf{C}_c$ (left). The top shows a 2D simple rotation, while the bottom shows an isotropic scaling. This geodesic path in Bures distance linearly interpolates the principal standard deviations of the densities (right). Colors on the right correspond to the sum of square root eigenvalues of the densities on the left.

is defined as

$$\mathcal{B}^2(P, Q) = \text{trace}\left(\Sigma_P + \Sigma_Q - 2\left(\Sigma_Q^{1/2}\Sigma_P\Sigma_Q^{1/2}\right)^{1/2}\right). \tag{1.7}$$

The Bures distance provides a measure of the work required to "reshape" one pile of dirt with elliptical volume into another. Geometrically, a straight line as measured via Bures distance (i.e. a Bures geodesic; Thanwerdas and Pennec, 2022) traces out a sequence of densities with fixed mean, and varying covariance matrices (Figure 1.7). The sum of principal standard deviations of each density (sum of square root eigenvalues) along this path, forms a straight line. Thus, linear interpolation of the Bures distance can be interpreted as linear interpolation of principal standard deviations between densities.

In this thesis, we use an alternative, equivalent form of Equation 1.7 (Bhatia et al., 2019),

$$\mathcal{B}^2(P, Q) = \min_{U \in O(n)} \|\Sigma_P^{1/2} - \Sigma_Q^{1/2} \boldsymbol{U}\|_F^2. \tag{1.8}$$

Equation 1.8 states that the Bures distance between two covariances Σ_P and Σ_Q can be interpreted as the least squares solution to a problem involving finding the optimal rotation $\mathbf{U}$ between their PSD square roots. This least squares optimization can easily be solved in closed form using the singular value decomposition (Appendix D.6.4). We exploit this formulation to derive an alternating least squares algorithm for our Wasserstein-based shape metrics in Chapter 5.

1.2.3.3 Energy distance

The Energy distance is a non-parametric distance, widely used in various fields, including statistics and machine learning (Feydy et al., 2019; Székely and Rizzo, 2013). Given two probability distributions P and Q defined on a metric space $\mathcal{X}$, the (squared) q-Energy distance between P and Q, denoted as $\mathcal{E}_q^2(P, Q)$, is defined as follows:

$$\mathcal{E}_q^2(P, Q) = \mathbb{E}[X - Y]^q - \frac{1}{2}\mathbb{E}[X - X']^q - \frac{1}{2}\mathbb{E}[Y - Y']^q, \tag{1.9}$$

where $X, X' \sim P$, $Y, Y' \sim Q$, and $0 < q < 2$. Concretely, we form all possible pairwise differences of data points within and between distributions, and compute an average between-distribution similarity term (first term) and within-distribution similarity terms (last two terms). By subtracting the within-distribution similarity terms from the average between-distribution similarity, the Energy distance captures the discrepancy between the two distributions, considering both the location and spread of the data. In practice, the Wasserstein distance suffers from the curse of dimensionality, while estimates of Energy distance converge at a faster rate, making it an attractive alternative (Gretton et al., 2012; Sejdinovic et al., 2013).

1.3 Thesis Organization

The remaining chapters of this thesis are organized as follows. Chapters 2, 3, and 4 propose new theories and models of neural population adaptation. These specifically focus on adaptive gain control, and how classical concepts of single-neuron gain adaptation can be *generalized* to explain population-level adaptation. In Chapter 5, we outline a new statistical framework for comparing and aligning multi-dimensional *stochastic* neural responses. Finally, the discussion in Chapter 6 closes out this dissertation with concluding remarks.

2 Adaptive Whitening in Neural Populations with Gain-modulating Interneurons

2.1 Overview

A version of this work was presented at Computational and Systems Neuroscience (2023), and the main findings are published in the Proceedings of the 40th International Conference on Machine Learning (Duong et al., 2023c).

Statistical whitening transformations play a fundamental role in many computational systems, and may also play an important role in biological sensory systems. Existing neural circuit models of adaptive whitening operate by modifying synaptic interactions; however, such modifications would seem both too slow and insufficiently reversible. Motivated by the extensive neuroscience literature on gain modulation, we propose an alternative model that adaptively whitens its responses by modulating the gains of individual neurons. Starting from a novel whitening objective, we derive an online algorithm that whitens its outputs by adjusting the marginal variances of an *overcomplete* set of projections. We map the algorithm onto a recurrent neural network with fixed synaptic weights and gain-modulating interneurons. We demonstrate numerically that sign-constraining the gains improves robustness of the network to ill-conditioned inputs,

and a generalization of the circuit achieves a form of local whitening in convolutional populations, such as those found throughout the visual or auditory systems.

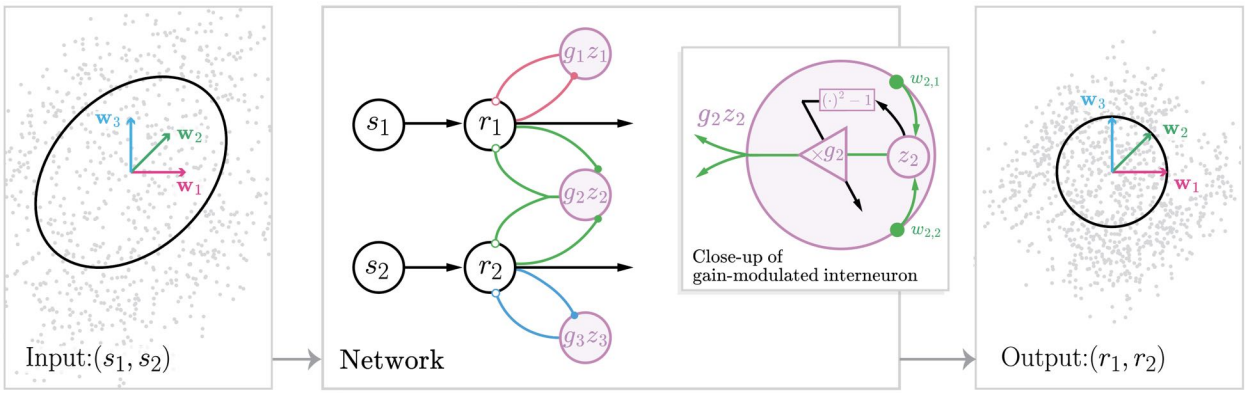

Figure 2.1: Schematic of a recurrent statistical whitening network with 2 primary neurons and 3 interneurons. **Left**: 2D Scatter plot of network inputs $\mathbf{s} = [s_1, s_2]^\top$ (e.g. post-synaptic currents), with covariance indicated by the ellipse. **Center**: Primary neurons, with outputs $\mathbf{r} = [r_1, r_2]^\top$, receive external feedforward inputs, $\mathbf{s}$, and recurrent feedback from an overcomplete population of interneurons, $-\sum_{i=1}^{3} g_i z_i \mathbf{w}_i$. Projection vectors $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\} \in \mathbb{R}^2$ encode feedforward synaptic weights connecting primary neurons to interneuron $i = 1, 2, 3$, with *symmetric* feedback connections. Weight vectors are shown in the left and right panels with corresponding colors. In general, there can exist all-to-all connectivity; we use a reduced subset of weights here for diagram clarity. **Inset:** The i^{th} interneuron (e.g. here $i = 2$) receives input $z_i = \mathbf{w}_i^\top \mathbf{r}$, which is multiplied by its gain g_i to produce output $g_i z_i$. Its gain, g_i, is adjusted s.t. $\Delta g_i \propto z_i^2 - 1$. The dark arrow indicates that the gain update operates on a slower time scale. **Right:** Scatter plots of the whitened network outputs $\mathbf{r}$. Outputs have unit variance along all $\mathbf{w}_i$'s, which is equivalent to having identity covariance matrix, i.e., $\mathbf{C}_{rr} = \mathbf{I}_N$ (black circle).

2.2 Introduction

Statistical whitening transformations, in which multi-dimensional inputs are decorrelated and normalized to have unit variance, are common in signal processing and machine learning systems. For example, they are integral to many statistical factorization methods (Bell and Sejnowski, 1996; Hyvärinen and Oja, 2000; Olshausen and Field, 1996), they provide beneficial preprocessing during neural network training (Krizhevsky et al., 2009), and they can improve unsupervised feature learning (Coates et al., 2011). More recently, self-supervised learning methods have used decorrelation transformations such as whitening to prevent representational collapse (Bardes

et al., 2022; Ermolov et al., 2021; Hua et al., 2021; Zbontar et al., 2021). While whitening has mostly been used for training neural networks in the offline setting, it is also of interest to develop adaptive (run-time) variants that can adjust to dynamically changing input statistics with minimal changes to the network (e.g. Hu et al., 2022; Mohan et al., 2021).

Single neurons in early sensory areas of many nervous systems rapidly adjust to changes in input statistics by scaling their input-output gains (Adrian and Matthews, 1928b). This allows neurons to adaptively normalize the variance of their outputs (Bonin et al., 2006; Nagel and Doupe, 2006), maximizing information transmitted about sensory inputs (Barlow, 1961; Fairhall et al., 2001; Laughlin, 1981). At the neural *population* level, in addition to variance normalization, adaptive decorrelation and whitening transformations have been observed across species and sensory modalities, including: macaque retina (Atick and Redlich, 1992); cat primary visual cortex (Benucci et al., 2013; Muller et al., 1999); and the olfactory bulbs of zebrafish (Friedrich, 2013) and mice (Giridhar et al., 2011; Gschwend et al., 2015). These population-level adaptations reduce redundancy in addition to normalizing neuronal outputs, facilitating *dynamic* efficient multi-channel coding (Barlow and Foldiak, 1989; Schwartz and Simoncelli, 2001). However, the mechanisms underlying such adaptive whitening transformations remain unknown, and would seem to require coordinated synaptic adjustments amongst neurons, as opposed to the single neuron case which relies only on gain rescaling.

Here, we propose a novel recurrent network architecture for online statistical whitening that exclusively relies on gain modulation. Specifically, the primary contributions of our study are as follows:

1. We introduce a novel factorization of the (inverse) whitening matrix, using an *overcomplete, arbitrary, but fixed* basis, and a diagonal matrix with statistically optimized entries. This is in contrast with the conventional factorization using the eigendecomposition of the input covariance matrix.

2. We introduce an unsupervised online learning objective using this factorization to express the whitening objective solely in terms of the *marginal* variances within the overcomplete representation of the input signal.

3. We derive a recursive algorithm to optimize the objective, and show that it corresponds to an unsupervised recurrent neural network (RNN), comprised of primary neurons and an auxiliary overcomplete population of interneurons, whose synaptic weights are fixed, but whose gains are adaptively modulated. The network responses converge to the classical symmetric whitening solution without backpropagation.

4. We show how enforcing non-negativity on the gain modulation provides a novel approach for dealing with ill-conditioned or noisy data. Further, we relax the global whitening constraint in our objective and provide a method for *local* decorrelation of convolutional neural populations.

2.3 A Novel Objective for Symmetric Whitening

Consider a neural network with N primary neurons. For each $t = 1, 2, \ldots,$ let $\mathbf{s}_t$ and $\mathbf{r}_t$ be N-dimensional vectors whose components respectively denote the inputs (e.g. post-synaptic currents), and outputs of the primary neurons at time t (Figure 2.1). Without loss of generality, we assume the inputs $\mathbf{s}_t$ are centered.

2.3.1 Conventional objective

Statistical whitening aims to linearly transform inputs $\mathbf{s}_t$ so that the covariance of the outputs $\mathbf{r}_t$ is the identity, i.e.,

$$\mathbf{C}_{rr} = \langle \mathbf{r}_t \mathbf{r}_t^\top \rangle_t = \mathbf{I}_N, \tag{2.1}$$

where $\langle \cdot \rangle_t$ denotes the expectation operator over t, and $\mathbf{I}_N$ denotes the $N \times N$ identity matrix.

It is well known that whitening is not unique: any orthogonal rotation of a random vector with identity covariance matrix also has identity covariance matrix. There are several common methods of resolving this rotational ambiguity, each with their own advantages (Kessy et al., 2018). Here, we focus on the symmetric whitening transformation, often referred to as Zero-phase Component Analysis (ZCA) whitening or Mahalanobis whitening, which minimizes the mean-squared error between the inputs and the whitened outputs (alternatively, the one whose transformation matrix is symmetric). The symmetric whitened outputs are the optimal solution to the minimization problem

$$\min_{\{\mathbf{r}_t\}} \langle \|\mathbf{s}_t - \mathbf{r}_t\|_2^2 \rangle_t \quad \text{s.t.} \quad \langle \mathbf{r}_t \mathbf{r}_t^\top \rangle_t = \mathbf{I}_N, \tag{2.2}$$

where $\| \cdot \|_2$ denotes the Euclidean norm on $\mathbb{R}^N$. Assuming the covariance of the inputs $\mathbf{C}_{ss} := \langle \mathbf{s}_t \mathbf{s}_t^\top \rangle_t$ is positive definite, the unique solution to the optimization problem in Equation 2.2 is $\mathbf{r}_t = \mathbf{C}_{ss}^{-1/2} \mathbf{s}_t$ for $t = 1, 2, \ldots$, where $\mathbf{C}_{ss}^{-1/2}$ is the symmetric inverse matrix square root of $\mathbf{C}_{ss}$ (see Appendix A.1).

Previous approaches to *online* symmetric whitening have optimized Equation 2.2 by deriving RNNs whose *synaptic weights* adaptively adjust to learn the eigendecomposition of the (inverse) whitening matrix, $\mathbf{C}_{ss}^{1/2} = \mathbf{V} \mathbf{\Lambda}^{1/2} \mathbf{V}^\top$, where $\mathbf{V}$ is an orthogonal matrix of eigenvectors and $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues (Pehlevan and Chklovskii, 2015). We propose an entirely different decomposition: $\mathbf{C}_{ss}^{1/2} = \mathbf{W} \operatorname{diag}(\mathbf{g}) \mathbf{W}^\top + \mathbf{I}_N$, where $\mathbf{W}$ is a *fixed* overcomplete matrix of synaptic weights, and $\mathbf{g}$ is a vector of *gains* that adaptively adjust to match the whitening matrix. For more details on this factorization, see Appendix A.3.

2.3.2 A NOVEL OBJECTIVE USING *MARGINAL* STATISTICS

We formulate an objective for learning the symmetric whitening transform via gain modulation. Our innovation exploits the fact that a random vector has identity covariance matrix (i.e., Equation 2.1 holds) if and only if it has unit marginal variance along *all possible 1D projections* (a form of tomography; see Figure 1.3 and Related Work). We derive a tighter statement for a finite but *overcomplete* set of at least $K \geqslant K_N := N(N+1)/2$ distinct axes ('overcomplete' means that the number of axes exceeds the dimensionality of the input, i.e., $K > N$). Intuitively, this equivalence holds because an $N \times N$ symmetric matrix has K_N degrees of freedom, so the marginal variances along $K \geqslant K_N$ distinct axes are sufficient to constrain an $N \times N$ covariance matrix. We formalize this equivalence in the following proposition, whose proof is provided in Appendix A.2.

Proposition 2.1. *Fix $K \geq K_N$. Suppose $\mathbf{w}_1, \ldots, \mathbf{w}_K \in \mathbb{R}^N$ are unit vectors*[1] *such that*

$$\mathit{span}(\{\mathbf{w}_1\mathbf{w}_1^\top, \ldots, \mathbf{w}_K\mathbf{w}_K^\top\}) = \mathbb{S}^N, \tag{2.3}$$

where $\mathbb{S}^N$ denotes the K_N-dimensional vector space of $N \times N$ symmetric matrices. Then Equation 2.1 holds if and only if the projection of $\mathbf{r}_t$ onto each unit vector $\mathbf{w}_1, \ldots, \mathbf{w}_K$ has unit variance, i.e.,

$$\langle(\mathbf{w}_i^\top \mathbf{r}_t)^2\rangle_t = 1 \quad \textit{for} \quad i = 1, \ldots, K. \tag{2.4}$$

Assuming Equation 2.3 holds, we can interpret the set of vectors $\{\mathbf{w}_1, \ldots, \mathbf{w}_K\}$ as a *frame* (i.e., an overcomplete basis; Casazza et al., 2013) in $\mathbb{R}^N$ such that the covariance of the outputs $\mathbf{C}_{rr}$ can be computed from the variances of the K-dimensional projection of the outputs onto the set of frame vectors. Thus, we can replace the whitening constraint in Equation 2.2 with the equivalent

[1] The unit-length assumption is imposed, without loss of generality, for notational convenience.

marginal variance constraint to obtain the following objective:

$$\min_{\{\mathbf{r}_t\}} \langle \|\mathbf{s}_t - \mathbf{r}_t\|_2^2 \rangle_t \quad \text{s.t.} \quad \text{Equation 2.4 holds.} \tag{2.5}$$

2.4 An RNN with Gain Modulation for Adaptive Symmetric Whitening

In this section, we derive an online algorithm for solving the optimization problem in Equation 2.5 and map the algorithm onto an RNN with adaptive gain modulation. Assume we have an overcomplete frame $\{\mathbf{w}_1, \ldots, \mathbf{w}_K\}$ in $\mathbb{R}^N$ satisfying Equation 2.3. We concatenate the frame vectors into an $N \times K$ synaptic weight matrix $\mathbf{W} := [\mathbf{w}_1, \ldots, \mathbf{w}_K]$. In our network, primary neurons project onto a layer of K interneurons via the synaptic weight matrix to produce the K-dimensional vector $\mathbf{z}_t := \mathbf{W}^\top \mathbf{r}_t$, encoding the interneurons' post-synaptic inputs at time t (Figure 2.1). We emphasize that the synaptic weight matrix $\mathbf{W}$ remains *fixed*.

2.4.1 Enforcing the marginal variance constraints with scalar gains

We introduce Lagrange multipliers $g_1, \ldots, g_K \in \mathbb{R}$ to enforce the K constraints in Equation 2.4. These are concatenated as the entries of a K-dimensional vector $\mathbf{g} := [g_1, \ldots, g_K]^\top \in \mathbb{R}^K$, and express the whitening objective as a saddle point optimization:

$$\max_{\mathbf{g}} \min_{\{\mathbf{r}_t\}} \langle \ell(\mathbf{s}_t, \mathbf{r}_t, \mathbf{g}) \rangle_t, \tag{2.6}$$
$$\text{where} \quad \ell(\mathbf{s}, \mathbf{r}, \mathbf{g}) := \|\mathbf{s} - \mathbf{r}\|_2^2 + \sum_{i=1}^{K} g_i \left\{ (\mathbf{w}_i^\top \mathbf{r})^2 - 1 \right\}.$$

Here, we have exchanged the order of maximization over $\mathbf{g}$ and minimization over $\mathbf{r}_t$, which is justified because $\ell(\mathbf{s}_t, \mathbf{r}_t, \mathbf{g})$ satisfies the saddle point property with respect to $\mathbf{r}$ and $\mathbf{g}$, see sec-

tion A.4.

In our RNN implementation, there are K interneurons and g_i corresponds to the multiplicative gain associated with the i^{th} interneuron, so that its output at time t is $g_i z_{i,t}$ (Figure 2.1, Inset). Equation 2.6, shows that the gain of the i^{th} interneuron, g_i, encourages the marginal variance of $\mathbf{r}_t$ along the axis spanned by $\mathbf{w}_i$ to be unity. Importantly, the gains are not hyper-parameters, but rather they are optimization variables which statistically whiten the outputs $\{\mathbf{r}_t\}$, preventing the neural outputs from trivially matching the inputs $\{\mathbf{s}_t\}$.

2.4.2 Deriving RNN neural dynamics and gain updates

To solve Equation 2.6 in the online setting, we assume there is a time-scale separation between 'fast' neural dynamics and 'slow' gain updates, so that at each time step the neural dynamics equilibrate before the gains are adjusted. This allows us to perform the inner minimization over $\{\mathbf{r}_t\}$ before the outer maximization over the gains $\mathbf{g}$. This is consistent with biological networks in which a given neuron's responses operate on a much faster time-scale than its intrinsic input-output gain, which is driven by slower processes such as changes in Ca^{2+} concentration gradients and Na^{+}-activated K^{+} channels (Ferguson and Cardin, 2020; Wang et al., 2003).

2.4.2.1 Fast neural activity dynamics

For each time step $t = 1, 2, \ldots$, we minimize the objective $\ell(\mathbf{s}_t, \mathbf{r}_t, \mathbf{g})$ over $\mathbf{r}_t$ by recursively running gradient-descent steps to equilibrium:

$$\begin{aligned} \mathbf{r}_t &\leftarrow \mathbf{r}_t - \frac{\gamma}{2}\nabla_{\mathbf{r}}\ell(\mathbf{s}_t, \mathbf{r}_t(\tau), \mathbf{g}) \\ \mathbf{r}_t &\leftarrow \mathbf{r}_t + \gamma\left\{\mathbf{s}_t - \mathbf{W}(\mathbf{g} \circ \mathbf{z}_t) - \mathbf{r}_t\right\}, \end{aligned} \tag{2.7}$$

where $\gamma > 0$ is a small constant, $\mathbf{z}_t = \mathbf{W}^\top \mathbf{r}_t$, the circle '$\circ$' denotes the Hadamard (element-wise) product, $\mathbf{g} \circ \mathbf{z}_t$ is a vector of K gain-modulated interneuron outputs, and we assume the primary

cell outputs are initialized at zero.

We see from the right-hand-side of Equation 2.7 that the 'fast' dynamics of the primary neurons are driven by three terms (within the curly braces): 1) constant feedforward external input $\mathbf{s}_t$; 2) recurrent gain-modulated feedback from interneurons $-\mathbf{W}(\mathbf{g} \circ \mathbf{z}_t)$; and 3) a leak term $-\mathbf{r}_t$. Because the neural activity dynamics are linear, we can analytically solve for their equilibrium (i.e. steady-state), $\bar{\mathbf{r}}_t$, by setting the update in Equation 2.7 to zero:

$$\begin{aligned} \bar{\mathbf{r}}_t &= \left[\mathbf{I}_N + \mathbf{W}\,\mathrm{diag}\,(\mathbf{g})\,\mathbf{W}^\top\right]^{-1}\mathbf{s}_t \\ &= \left[\mathbf{I}_N + \sum_{i=1}^{K} g_i \mathbf{w}_i \mathbf{w}_i^\top\right]^{-1}\mathbf{s}_t, \end{aligned} \tag{2.8}$$

where $\mathrm{diag}\,(\mathbf{g})$ denotes the $K \times K$ diagonal matrix whose $(i,i)^{\text{th}}$ entry is g_i, for $i = 1, \ldots, K$. The equilibrium feedforward interneuron inputs are then given by

$$\bar{\mathbf{z}}_t = \mathbf{W}^\top \bar{\mathbf{r}}_t. \tag{2.9}$$

The gain-modulated outputs of the K interneurons, $\mathbf{g} \circ \mathbf{z}_t$, are then projected back onto the primary cells via symmetric weights, $-\mathbf{W}$ (Figure 2.1). After $\mathbf{g}$ adapts to optimize Equation 2.6 (provided Proposition 2.1 holds), the matrix within the brackets in Equation 2.8 will equal $\mathbf{C}_{ss}^{1/2}$, and the circuit's equilibrium responses are symmetrically whitened. The result is a novel *overcomplete* symmetric matrix factorization in which $\mathbf{W}$ is arbitrary and fixed, while $\mathbf{C}_{ss}^{1/2}$ is adaptively learned and encoded in the gains $\mathbf{g}$.

2.4.2.2 Slow gain dynamics

After the fast neural activities reach steady-state, the interneuron gains are updated with a stochastic gradient-ascent step with respect to $\mathbf{g}$:

$$\begin{aligned} \mathbf{g} &\leftarrow \mathbf{g} + \frac{\eta}{2}\nabla_{\mathbf{g}}\ell(\mathbf{s}_t, \bar{\mathbf{r}}_t, \mathbf{g}) \\ \mathbf{g} &\leftarrow \mathbf{g} + \eta\left(\bar{\mathbf{z}}_t^{\circ 2} - \mathbf{1}\right), \end{aligned} \tag{2.10}$$

where $\eta > 0$ is the learning rate, $\bar{\mathbf{z}}_t^{\circ 2} = [\bar{z}_{t,1}^2, \ldots, \bar{z}_{t,K}^2]^\top$, and $\mathbf{1} = [1, \ldots, 1]^\top$ is the K-dimensional vector of ones[2]. Remarkably, the update to the i^{th} interneuron's gain g_i (Equation 2.10) depends only on the online estimate of the *variance* of its equilibrium input $\bar{z}_{t,i}^2$, and its distance from 1 (i.e. the target variance). Since the interneurons adapt using local signals, this circuit is a suitable candidate for hardware implementations using low-power neuromorphic chips (Pehlevan and Chklovskii, 2019). Intuitively, each interneuron adjusts its gain to modulate the amount of suppressive (inhibitory) feedback onto the joint primary neuron responses. In Appendix A.3, we provide conditions under which $\mathbf{g}$ can be solved analytically. Thus, while statistical whitening inherently involves a transformation on a joint density, our solution operates solely using single neuron gain changes in response to *marginal* statistics of the joint density.

2.4.2.3 Online unsupervised algorithm

By combining Equations 2.7 and 2.10, we arrive at our online RNN algorithm for adaptive whitening via gain modulation (Algorithm 1). We also provide batched and offline versions of the algorithm in Appendix A.6.

There are two points worth noting about this network: 1) $\mathbf{W}$ remains *fixed* in Algorithm 1. Instead, $\mathbf{g}$ adapts to statistically whiten the outputs. 2) In practice, since network dynamics are

[2] Appendix A.5 generalizes the gain update to allowing for temporal-weighted averaging of the variance over past samples.

Algorithm 1: Adaptive whitening via gain modulation

1: **Input:** Centered inputs $\mathbf{s}_1, \mathbf{s}_2, \cdots \in \mathbb{R}^N$
2: **Initialize:** $\mathbf{W} \in \mathbb{R}^{N\times K}$; $\mathbf{g} \in \mathbb{R}^K$; $\eta, \gamma > 0$
3: **for** $t = 1, 2, \ldots$ **do**
4: $\quad \mathbf{r}_t \leftarrow \mathbf{0}$
5: $\quad$ **while** not converged **do**
6: $\quad\quad \mathbf{z}_t \leftarrow \mathbf{W}^\top \mathbf{r}_t$
7: $\quad\quad \mathbf{r}_t \leftarrow \mathbf{r}_t + \gamma \left\{\mathbf{s}_t - \mathbf{W}(\mathbf{g} \circ \mathbf{z}_t) - \mathbf{r}_t\right\}$
8: $\quad$ **end while**
9: $\quad \mathbf{g} \leftarrow \mathbf{g} + \eta \left(\mathbf{z}_t^{\circ 2} - \mathbf{1}\right)$
10: **end for**

linear, we can bypass the inner loop (the fast dynamics of the primary cells, lines 5–8), by directly computing $\bar{\mathbf{r}}_t$, and $\bar{\mathbf{z}}_t$ (Eqs. 2.8, 2.9).

2.5 Numerical Experiments and Applications

We provide different applications of our adaptive symmetric whitening network via gain modulation, emphasizing that gain adaptation is distinct from, and *complementary to*, synaptic weight learning (i.e. learning $\mathbf{W}$). We therefore side-step the goal of learning the frame $\mathbf{W}$, and assume it is fixed (for example, through longer time scale learning). This allows us to decouple and analyze the general properties of our proposed gain modulation framework, independently of the choice of frame. Python code for this study can be located at `github.com/lyndond/frame_whitening`.

We evaluate the performance of our adaptive whitening algorithm using the matrix operator norm, $\|\cdot\|_{\mathrm{Op}}$, which measures the largest eigenvalue,

$$\text{Error} := \|\mathbf{C}_{rr} - \mathbf{I}_N\|_{\mathrm{Op}}.$$

As a performance criterion, we use $\|\mathbf{C}_{rr} - \mathbf{I}_N\|_{\mathrm{Op}} \leq 0.1$, the point at which the principal axes of $\mathbf{C}_{rr}$ are within 0.1 of unity. Geometrically, this means the ellipsoid corresponding to the covariance matrix lies between the circles with radii 0.9 and 1.1.

For visualization of output covariance matrices, we plot 2D ellipses representing the 1 standard deviation probability level-set contour of the density. These ellipses are defined by the set of points $\{\|\mathbf{C}_{rr}^{1/2}\mathbf{v}\| \mathbf{v} : \|\mathbf{v}\| = 1\}$.

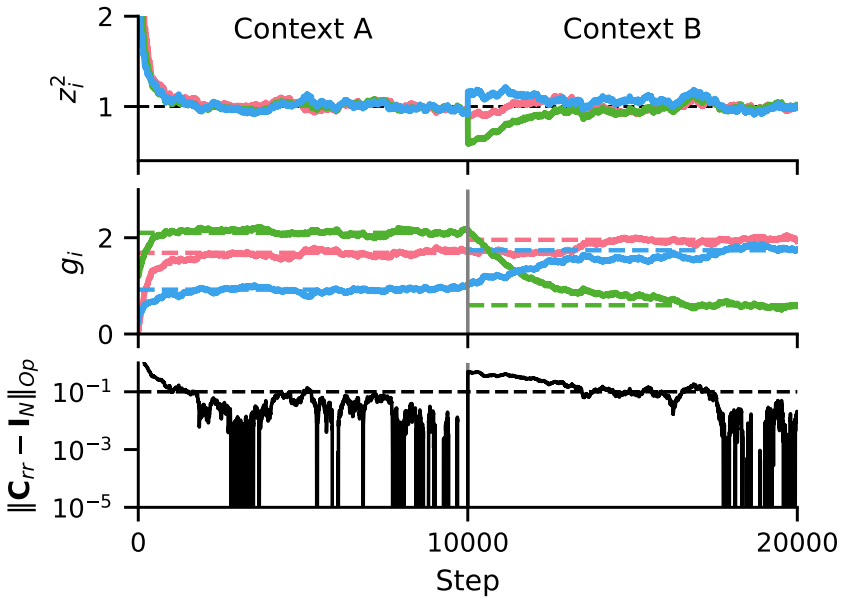

Figure 2.2: Network from Figure 2.1 (with corresponding colors; N=2, K=K_N=3, η=2E-3) adaptively whitening samples from two randomly generated statistical contexts online (10K steps each). **Top:** Marginal variances measured by interneurons approach 1 over time. **Middle:** Dynamics of interneuron gains, which are applied to z_i before feeding back onto the primary cells. Dashed lines are optimal gains (Appendix A.3). **Bottom:** Error over time, as measured by the maximal difference between the standard deviation along the principal axes of $\mathbf{C}_{rr}$ and unity.

2.5.1 Adaptive symmetric whitening via gain modulation

We first demonstrate that our algorithm successfully whitens its outputs. We initialize a network with fixed interneuron weights, $\mathbf{W}$, corresponding to the frame illustrated in Figure 2.1 (N=2, K=K_N=3). Figure 2.2 shows the network adapting to inputs from two successively presented contexts with randomly-generated underlying input covariances $\mathbf{C}_{ss}$ (10K gain update steps each). As update steps progress, all marginal variances converge to unity, as expected from

the objective (top panel). Since the number of interneurons satisfies $K{=}K_N$, the optimal gains to achieve symmetric whitening can be solved analytically (Appendix A.3), and are shown in the middle panel (dashed lines).

Figure 2.2 illustrates the *online, adaptive* nature of the network; it whitens inputs from novel statistical contexts at run-time, without supervision. By Proposition 2.1, measuring unit variance along K_N unique axes, as in this example, guarantees that the underlying joint density is statistically white. Indeed, the whitening error (bottom panel), approaches zero as all K_N marginal variances approach 1. Thus, with interneurons monitoring their respective *marginal* input variances z_i^2, and re-scaling their gains to modulate feedback onto the primary neurons, the network adaptively whitens its outputs in each context.

2.5.2 Algorithmic convergence rate depends on $\mathbf{W}$

Our model assumes that the frame, $\mathbf{W}$, is fixed and known (e.g., optimized via pre-training or development). This distinguishes our method from existing symmetric whitening methods, which typically operate by estimating and transforming to the eigenvector basis. By contrast, our network obviates learning the principal axes of the data altogether, and instead uses a statistical sampling approach along the fixed set of measurement axes spanned by $\mathbf{W}$. While the result expressed in Proposition 2.1 is exact, and the *optimal solution* to the whitening objective Equation 2.5 is independent of $\mathbf{W}$ (provided Equation 2.3 holds), we hypothesize that the *algorithmic convergence rate* would depend on $\mathbf{W}$.

Figure 2.3 summarizes an experiment assessing the convergence rate of different networks whitening inputs with a random covariance, $\mathbf{C}_{ss}$, with $N = 2$ (the results are consistent when $N > 2$). We initialize three kinds of frames $\mathbf{W} \in \mathbb{R}^{N \times K_N}$ with 100 repetitions each: '**Random**', a frame with i.i.d. Gaussian entries; '**Optimized**', a randomly initialized frame whose columns are then optimized to have minimum mutual coherence and cover the ambient space; and '**Spectral**', a frame whose first N columns are the eigenvectors of the data and the remaining $K_N - N$

columns are zeros. For clarity, we remove the effects of input sampling stochasticity by running the offline version of our network, which assumes having direct access to the input covariance (Appendix A.6); the online version is qualitatively similar.

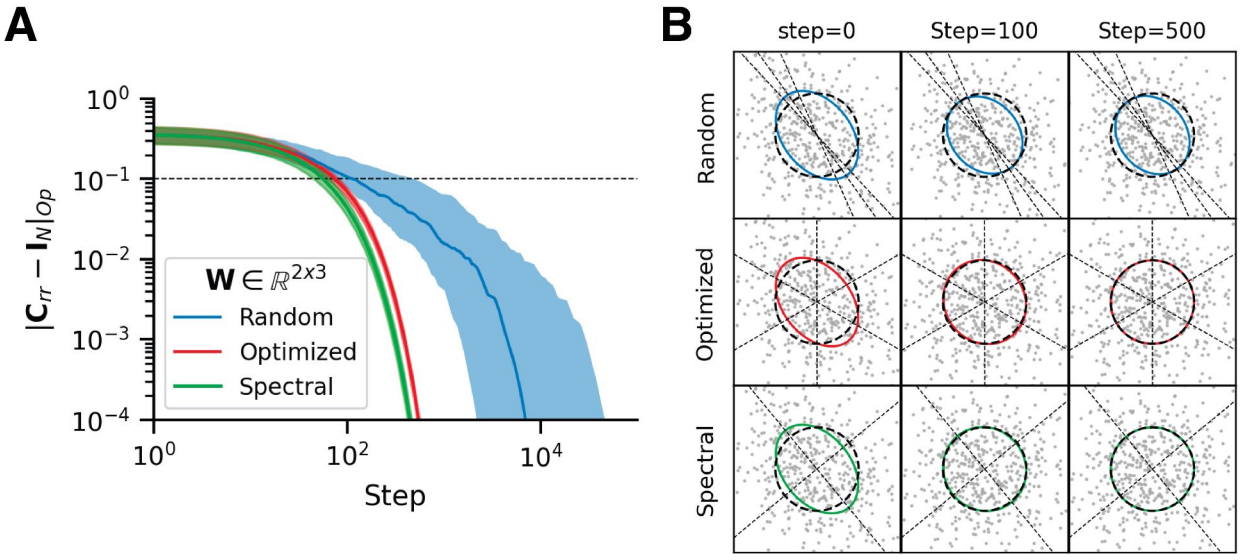

Figure 2.3: Convergence rate depends on structure of $\mathbf{W}$. For each network, η=1E-2. **A:** Error over time. Curves are median and [25%, 75%] quantile regions over 100 repeats. Dashed line indicates when the principal axes of 1-standard deviation ellipse representing $\mathbf{C}_{rr}$ are within 0.1 of unity. **B:** Scatter plots and covariance ellipses of $\mathbf{r}$ for a single experiment with each frame type at different steps. Gray dashed lines are axes spanned by $\mathbf{W}$.

When the input distribution is known, then using the input covariance eigenvectors, as with the Spectral frame, defines a bound on achievable performance, converging faster, on average, than the Random and Optimized frames (Figure 2.3A,B). This is because the frame is aligned with the input covariance's principal axes, and a simple gain scaling along those directions is sufficient to achieve a whitened response. We find that the networks with Optimized frames converge at similar rates to those with Spectral frames, despite the frame vectors not being aligned with the principal axes of the data (Figure 2.3B). Comparing the Random to Optimized frames gives a better understanding of how one might choose a frame in the more realistic scenario when the input distribution is unknown. The networks with Optimized frames systematically converge faster than Random frames. Thus, when the input distribution is unknown, we empirically find that

the convergence rate of Algorithm 1 benefits from a frame that is optimized to splay the ambient space. Increased coverage of the space by the frame vectors facilitates whitening with our gain re-scaling mechanism. Sec. 2.5.5 elaborates on how underlying signal structure can be exploited to inform more efficient choices of frames.

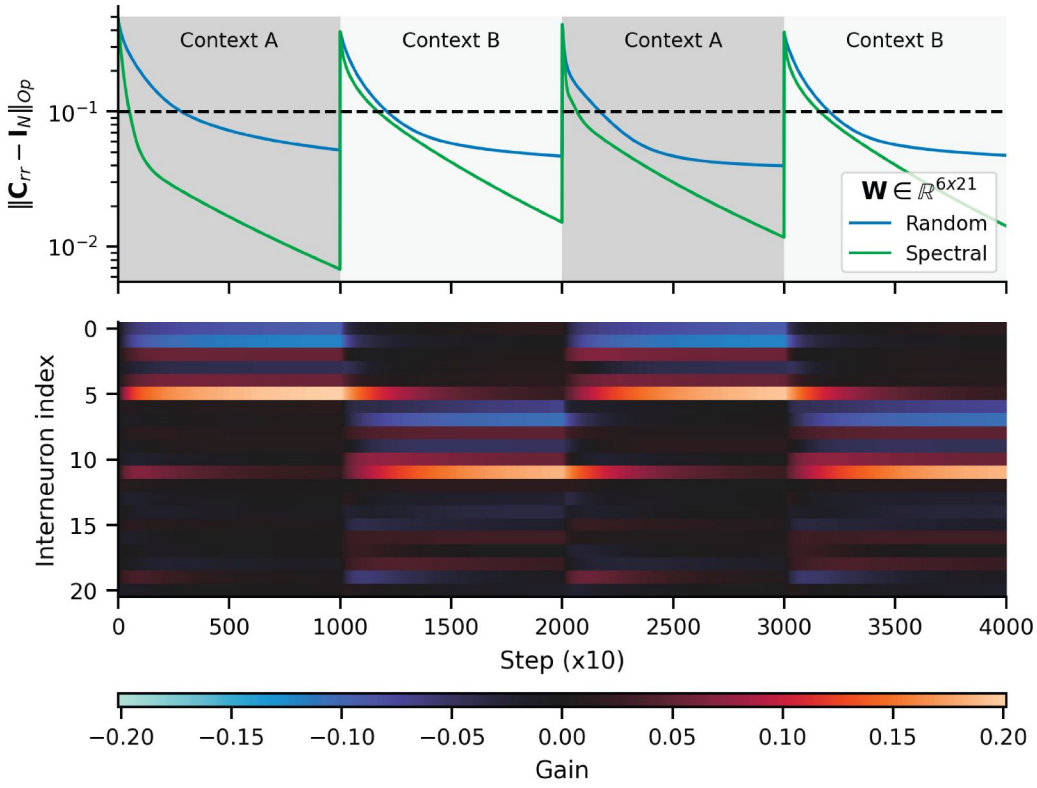

Figure 2.4: Gain modulation as a fast implicit sparse gating mechanism. **Top**: Error over time for Spectral vs. Random networks (N=6; K=K_N=21; η=1E-3) adapting to 2 alternating statistical contexts with different input covariances. Dashed line indicates when the principal axes of 1-standard deviation ellipsoid representing $\mathbf{C}_{rr}$ are within 0.1 of unity. **Bottom**: Gains act as implicit context switches, sparsely gating the respective eigenbases embedded in the Spectral frame to optimally whiten each context.

2.5.3 Implicit sparse gating via gain modulation

Motivated by the findings in Sec 2.5.2, and concepts from sparse coding (Olshausen and Field, 1996), we explore how adaptive gain modulation can complement or augment a 'pre-trained' network with context-dependent weights. Figure 2.4 shows an experiment using either a pre-

trained Spectral, or Random $\mathbf{W}$ (N=6, K=K_N=21) adaptively whitening inputs from two random, alternating statistical contexts, A and B, for 10K steps each. The first and second N columns of the Spectral frame are the eigenvectors of context A and B's covariance matrix, respectively, and the remaining elements are random i.i.d. Gaussian; the Random frame has all i.i.d. Gaussian elements. Figure 2.4 (top panel) shows that both networks successfully adapt to whiten the inputs from each context, with the Spectral frame converging faster than the Random frame (as in Sec 2.5.2).

Inspecting the Spectral frame's K interneuron gains during run-time (bottom panel) reveals that they sparsely 'select' the frame vectors corresponding to the eigenvectors of each respective condition (indicated by the blue/red intensity). This effect arises *without* a sparsity penalty or modifying the objective. Gain modulation thus *sparsely gates* context-dependent information without an explicit context signal.

2.5.4 Normalizing ill-conditioned data

Foundational work by Atick and Redlich (1992) showed that neural populations in the retina may encode visual inputs by optimizing mutual information in the presence of noise. For natural images with $1/f$ spectra, the optimal transform is approximately a product of a whitening filter and a low-pass filter. This is a particularly effective solution because when inputs are low-rank, $\mathbf{C}_{ss}$ is ill-conditioned (Figure 2.5A), and classical whitening leads to noise amplification along axes with small variance. In this section, we show how a simple modification to the objective allows our gain-modulating network to handle these types of inputs.

We prevent amplification of inputs below a certain variance threshold by replacing the unit marginal variance equality constraints with upper bound constraints[3]:

$$\langle (\mathbf{w}_i^\top \mathbf{r}_t)^2 \rangle_t \leqslant 1 \quad \text{for} \quad i = 1, \ldots, K. \tag{2.11}$$

[3]We set the threshold to 1 to remain consistent with the whitening objective, but it can be any arbitrary variance.

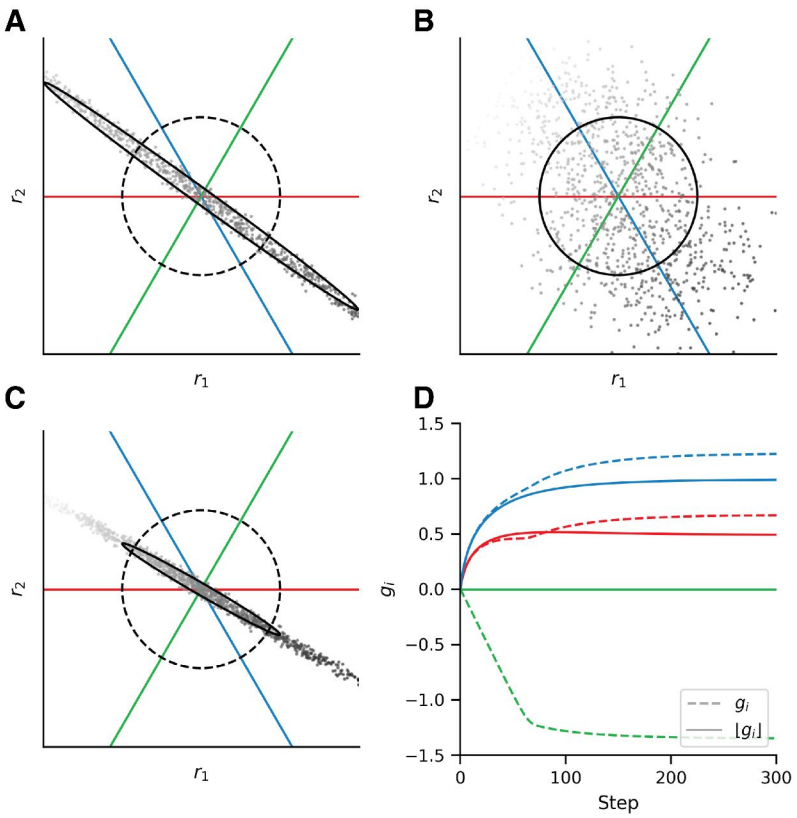

Figure 2.5: Two networks (N=2, K=3, η=0.02) whitening ill-conditioned inputs. **A:** Outputs without whitening. 2D scatterplot of a non-Gaussian density whose underlying signal lies close to a latent 1D axis. Many points lie outside of the axis limits in this panel. Signal magnitude along that axis is denoted by the grayscale gradient. The 1-standard deviation covariance matrix is depicted as a black ellipse. Colored lines are axes spanned by Optimal frame (see Sec 2.5.2). **B:** Symmetric whitening boosts noise along the uninformative direction. **C:** Modulating gains according to Eq. 2.14 rescales the data *without* amplifying noise. **D:** Gains updated with Eq. 2.10 vs. Eq. 2.14. Colors correspond to frame axes in panels A–C.

Our modified network objective then becomes

$$\min_{\{\mathbf{r}_t\}} \langle \|\mathbf{s}_t - \mathbf{r}_t\|_2^2 \rangle_t \quad \text{s.t.} \quad \text{Equation 2.11 holds.} \tag{2.12}$$

Intuitively, if the projected variance along a given direction is already less than or equal to unity, then it will not affect the overall loss. Interneuron gain should accordingly *stop adjusting* once the marginal variance along its projection axis is less than or equal to one. To enforce these upper bound constraints, we introduce gains as Lagrange multipliers, but restrict the domain of $\mathbf{g}$ to be

the non-negative orthant $\mathbb{R}_+^K$, resulting in non-negative optimal gains:

$$\max_{\mathbf{g}\in\mathbb{R}_+^K} \min_{\{\mathbf{r}_t\}} \langle \ell(\mathbf{s}_t, \mathbf{r}_t, \mathbf{g}) \rangle_t, \tag{2.13}$$

where $\ell(\mathbf{s}, \mathbf{r}, \mathbf{g})$ is defined as in Equation 2.6. At each time step t, we optimize Equation 2.13 by first taking gradient-descent steps with respect to $\mathbf{r}_t$, resulting in the same neural dynamics (Equation 2.7) and equilibrium solution (Equation 2.8) as before. To update $\mathbf{g}$, we modify Equation 2.10 to take a *projected* gradient-ascent step with respect to $\mathbf{g}$:

$$\mathbf{g} \leftarrow \lfloor \mathbf{g} + \eta(\bar{\mathbf{z}}_t^{\circ 2} - \mathbf{1}) \rfloor \tag{2.14}$$

where $\lfloor \cdot \rfloor$ denotes the element-wise half-wave rectification operation that projects its inputs onto the non-negative orthant $\mathbb{R}_+^K$, i.e., $\lfloor \mathbf{v} \rfloor := [\max(v_1, 0), \ldots, \max(v_K, 0)]^\top$.

Figure 2.5 shows a simulation of a network whitening ill-conditioned inputs with an Optimized frame (N=2, K=K_N; see Sec. 2.5.2) where gains are either unconstrained (Equation 2.10), or rectified (Equation 2.14). We observe that these two models converge to two different solutions (Figure 2.5B, C). When g_i is unconstrained, the network achieves global whitening, as before, but in doing so it amplifies noise along the axis orthogonal to the latent signal axis. The gains constrained to be non-negative converged to different values than the unconstrained gains (Figure 2.5D), with one of them (green) converging to zero rather than becoming negative. In general, with constrained g_i, the whitening error network converges to a non-zero value (see Appendix A.7 for details). Thus, with a non-negative constraint, the network normalizes the responses $\mathbf{r}$, and *does not amplify the noise*. In Appendix A.7 we show additional cases that provide further geometric intuition on differences between symmetric whitening with and without non-negative constrained gains.

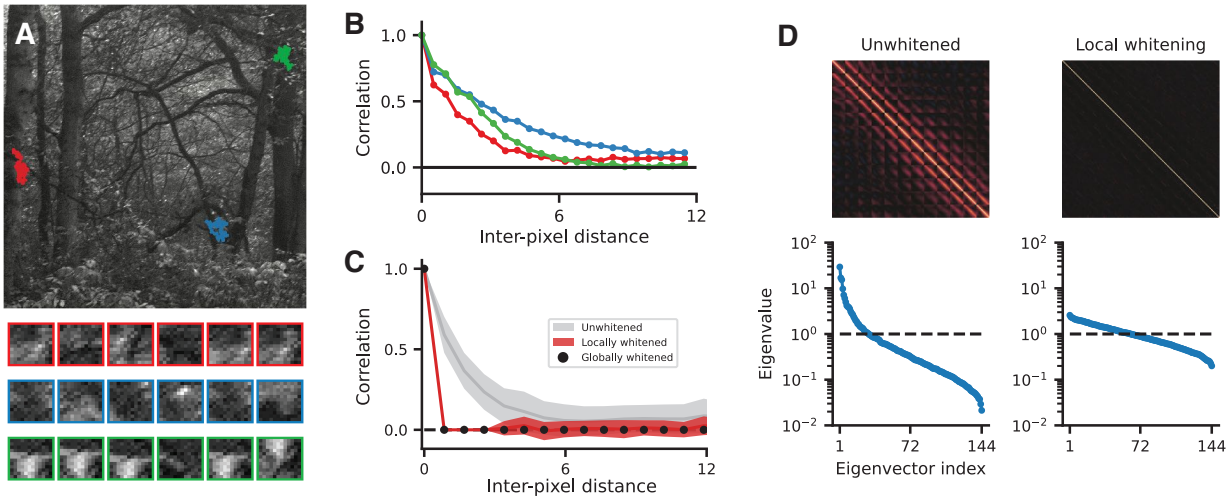

Figure 2.6: Local spatial whitening. **A)** Large grayscale image from which we draw 12×12 image patch samples. Colors represent random-walk sampling from regions of the image corresponding to contexts with different underlying statistics. Six samples from each context are shown below. **B)** Without whitening, pixel correlations decay rapidly with spatial distance in each context, suggesting that local whitening may be effective. **C)** Binned pairwise output pixel correlation of patches from the red context before (gray) and after global (black dots) vs. local whitening with overlapping 4×4 neighborhoods (red). Shaded regions represent standard deviations. **D)** Top: Correlation matrices of flattened patches from the red context before whitening (left), and after local symmetric whitening (right). Both panels use the same color scale. Bottom: Corresponding covariance eigenspectra. Dashed lines are spectra after global whitening.

2.5.5 Gain modulation enables local spatial decorrelation

Requiring K_N interneurons to guarantee a statistically white output (Proposition 2.1) becomes prohibitively costly for high-dimensional inputs: the number of interneurons scales as $O(N^2)$. This leads us to ask: how many interneurons are needed in practice? For natural sensory inputs such as images, it is well-known that inter-pixel correlation is highly structured, decaying as a function of distance. We simulate an experiment of visual gaze fixations and micro-saccadic eye movements using a Gaussian random walk, drawing 12×12 patch samples from a region of a natural image (Figure 2.6A; van Hateren and van der Schaaf, 1998); this can be interpreted as a form of video-streaming dataset where each frame is a patch sample. We repeat this for different randomly selected regions of the image (Figure 2.6A colors). The image content of each region is quite different, but the inter-pixel correlation within each context consistently falls rapidly with

distance (Figure 2.6B).

We *relax* the $O(N^2)$ marginal variance constraint to instead whiten *spatially local neighborhoods* of primary neurons whose inputs are the image patches. We construct a frame $\mathbf{W}$ that exploits spatial structure in the image patches, and spans $K < K_N$ axes in $\mathbb{R}^N$. $\mathbf{W}$ is convolutional, such that *overlapping* neighborhoods of 4×4 primary neurons are decorrelated, each by a population of interneurons that is 'overcomplete' with respect to that neighborhood (see Appendix A.8 for details). Importantly, taking into account local structure dramatically reduces the interneuron complexity from $O(N^2) \rightarrow O(N)$, thereby making our framework practically feasible for high-resolution image inputs and video streams. This frame is still overcomplete ($K > N$), but because $K < K_N$, we no longer guarantee at equilibrium that $\mathbf{C}_{rr} = \mathbf{I}_N$ (Proposition 2.1).

After the network converges to the inputs drawn from the red context (Figure 2.6C): i) interpixel correlations drop within the region specified by the local neighborhood; and ii) surprisingly, correlations at longer-range (i.e. outside the window of the defined spatial neighborhood) are also dramatically reduced. Accordingly, the eigenspectrum of the locally whitened outputs is significantly flatter compared to the inputs (Figure 2.6D left vs. right columns). We also provide an example using 1D inputs in Appendix A.8. This empirical result is not obvious — that whitening individual *overlapping local* neighborhoods of neurons should produce a more *globally* whitened output covariance. Indeed, exactly how or when a globally whitened solution is possible from whitening of spatial overlapping neighborhoods of the inputs is a problem worth pursuing.

2.6 Related Work

2.6.1 Biologically plausible whitening networks

Biological circuits operate in the online setting and, due to physical constraints, must learn exclusively using local signals. Therefore, to plausibly model neural computation, a neural net-

work model must operate in the online setting (i.e., streaming data) and use local learning rules (Pehlevan and Chklovskii, 2019). There are a few existing normative models of adaptive statistical whitening and related transformations; however, these models use synaptic plasticity mechanisms (i.e., changing $\mathbf{W}$) to adapt to changing input statistics (Chapochnikov et al., 2021; Lipshutz et al., 2023; Młynarski and Hermundstad, 2021; Pehlevan and Chklovskii, 2015; Westrick et al., 2016). Adaptation of neural population responses to changes in sensory input statistics occurs rapidly, on the order of hundreds of milliseconds to seconds (Muller et al., 1999; Wanner and Friedrich, 2020), so it could potentially arise from short-term synaptic plasticity, which operates on the timescale of tens of milliseconds to minutes (Zucker and Regehr, 2002), but not by long-term synaptic plasticity, which operates on the timescale of minutes or longer (Martin et al., 2000). Here, we have proposed an alternative hypothesis: that modulation of neural gains, which operates on the order of tens of milliseconds to minutes (Ferguson and Cardin, 2020), facilitates rapid adaptation of neural populations to changing input statistics.

2.6.2 Tomography and "sliced" density measurements

Leveraging 1D projections to compute the symmetric whitening transform is reminiscent of approaches taken in the field of tomography. Geometrically, our method represents an ellipsoid (i.e., the N dimensional covariance matrix) using noisy 1D projections of the ellipsoid onto axes spanned by frame vectors (i.e., estimates of the marginal variances). This is a special case of reconstruction problems studied in geometric tomography (Gardner, 1995; Karl et al., 1994). A distinction between tomography and our approach to symmetric whitening is that we are not reconstructing the multi-dimensional inputs; instead, we are utilizing the univariate measurements to transform an ellipsoid into a hyper-sphere.

In optimal transport, "sliced" methods offer a way to measure otherwise intractable Wasserstein distances in high dimensions (Bonneel et al., 2015), thereby enabling their use in optimization loss functions. Sliced methods estimate Wasserstein distance by taking series of 1D pro-

jections of two densities, then computing the expectation over all 1D Wasserstein distances, for which there exists an analytic solution. The 2-Wasserstein distance between a 1D zero-mean Gaussian with variance σ^2 and a standard normal density is

$$W_2\left(\mathcal{N}\left(0, \sigma^2\right); \mathcal{N}\left(0, 1\right)\right) = \|\sigma - 1\| .$$

This is strikingly similar to Equation 2.10. However, distinguishing characteristics of our approach include: 1) minimizing distance between *variances* rather than standard deviations; 2) directions along which we compute slices are fixed, while sliced methods compute a new set of projections at each optimization step; 3) our network operates online, *without* backpropagation.

2.7 Discussion

Our study introduces a recurrent circuit for adaptive whitening using *gain modulation* to transform joint second-order statistics of their inputs based on *marginal* variance measurements. We demonstrate that, given sufficiently many marginal measurements along unique axes, the network produces symmetric whitened outputs. Our objective (Equation 2.5) provides a novel way to think about the classical problem of statistical whitening, and draws connections to old concepts from tomography and transport theory. This framework is *flexible and extensible*, with some possible generalizations explored in Appendix A.9. For example, we show that our model provides a way to prevent representational collapse in the analytically tractable example of online principal subspace learning (Appendix A.9.1). By replacing the unity marginal variance constraint by a set of target variances differing from 1, the network can be used to transform its input density to one matching the corresponding (non-white) covariance (Appendix A.9.2).

2.7.1 Implications for machine learning

Decorrelation and whitening are canonical transformations in signal processing, widely used in compression and channel coding. Deep nets are generally not trained to whiten, although their response variances are generally normalized during training through batch normalization, and recent methods (e.g. Bardes et al., 2022) do impose global whitening properties in their objective functions. Modulating feature gains has proven effective in adapting pre-trained neural networks to novel inputs with out-of-training distribution statistics (Ballé et al., 2020; Duong et al., 2023b; Mohan et al., 2021). Future architectures may benefit from adaptive run-time adjustments to changing input statistics (e.g. Hu et al., 2022). Our framework provides an unsupervised, online mechanism that avoids 'catastrophic forgetting' in neural networks during continual learning.

2.7.2 Implications for neuroscience

It has been known for nearly 100 years (Adrian and Matthews, 1928b) that single neurons rapidly adjust their sensitivity (gain) adaptively, based on recent response history. Experiments suggest that neural populations *jointly* adapt, adjusting both the amplitude of their responses, as well as their correlations (e.g. Benucci et al., 2013; Friedrich, 2013) to confer dynamic, efficient multi-channel coding. The natural thought is that they achieve this by adjusting the strength of their interactions (synaptic weights). Our work provides a *fundamentally different* solution: these effects can arise solely through gain changes, thereby generalizing rapid and reversible single neuron adaptive gain modulation to the level of a neural population.

Support for our model will ultimately require careful experimental measurement and analysis of responses and gains of neurons in a circuit during adaptation (e.g. Wanner and Friedrich, 2020). Our model predicts: 1) Specific architectural constraints, such as reciprocally connected interneurons (Kepecs and Fishell, 2014), with consistency between their connectivity and population size (e.g. in the olfactory bulb). 2) Synaptic strengths that remain stable during adaptation,

which would adjudicate between our model and more conventional adaptation models relying on synaptic plasticity (e.g. Lipshutz et al., 2023). 3) Interneurons that modulate their gains according to the difference between the variance of their post-synaptic inputs and some *target variance* (Equation 2.10; also see Appendix A.9.2). Experiments could assess whether interneuron input variances converge to the same values after adaptive whitening. 4) Interneurons that increase their gains with the variance of their inputs (i.e. $\tilde{z}_{i,t}^2$). Input variance-dependent gain modulation may be mediated by changes in slow Na^+ currents (Kim and Rieke, 2003). This predicts a mechanistic role for interneurons during adaptation, and complements the observed gain effects found in excitatory neurons described in classical studies (Fairhall et al., 2001; Nagel and Doupe, 2006).

2.7.3 Conclusion

Whitening is an effective constraint for preventing feature collapse in representation learning (Ermolov et al., 2021; Zbontar et al., 2021). The networks developed here provide a whitening solution that is particularly well-suited for applications prioritizing streaming data and low-power consumption.

3 | Adaptive Whitening with Fast Gain Modulation and Slow Synaptic Plasticity

3.1 Overview

The work in this chapter unifies our adaptive gain control framework presented in Chapter 2 with existing, disparate models of adaptive whitening based on synaptic plasticity. We show that these two mechanisms are complementary forms of adaptation, and can operate in tandem over different timescales. Our results are published as a preprint (currently under review; Duong et al., 2023d).

Neurons in early sensory areas rapidly adapt to changing sensory statistics, both by normalizing the variance of their individual responses and by reducing correlations between their responses. Together, these transformations may be viewed as an adaptive form of statistical whitening. Existing mechanistic models of adaptive whitening exclusively use either synaptic plasticity or gain modulation as the biological substrate for adaptation; however, on their own, each of these models has significant limitations. In this work, we unify these approaches in a normative multi-timescale mechanistic model that adaptively whitens its responses with complementary computational roles for synaptic plasticity and gain modulation. Gains are modified

on a fast timescale to adapt to the current statistical context, whereas synapses are modified on a slow timescale to learn structural properties of the input statistics that are invariant across contexts. Our model is derived from a novel multi-timescale whitening objective that factorizes the inverse whitening matrix into basis vectors, which correspond to synaptic weights, and a diagonal matrix, which corresponds to neuronal gains. We test our model on synthetic and natural datasets and find that the synapses learn optimal configurations over long timescales that enable the circuit to adaptively whiten its responses on short timescales exclusively using gain modulation.

3.2 Introduction

Individual neurons in early sensory areas rapidly adapt to changing sensory statistics by normalizing the variance of their responses (Fairhall et al., 2001; Nagel and Doupe, 2006). At the population level, neurons also adapt by reducing correlations between their responses (Benucci et al., 2013; Muller et al., 1999). These adjustments enable the neurons to maximize the information that they transmit by utilizing their entire dynamic range and reducing redundancies in their representations (Attneave, 1954; Barlow, 1961; Barlow and Foldiak, 1989; Laughlin, 1981). A natural normative interpretation of these transformations is *adaptive whitening*, a context-dependent linear transformation of the sensory inputs yielding responses that have unit variance and are uncorrelated.

Decorrelation of the neural responses requires coordination between neurons and the neural mechanisms underlying such coordination are not known. Since neurons communicate via synaptic connections, it is perhaps unsurprising that most existing mechanistic models of adaptive whitening decorrelate neural responses by modifying the strength of these connections (Chapochnikov et al., 2021; King et al., 2013; Lipshutz et al., 2023; Pehlevan and Chklovskii, 2015; Pehlevan et al., 2018; Westrick et al., 2016; Wick et al., 2010). However, synaptic plasticity is gen-

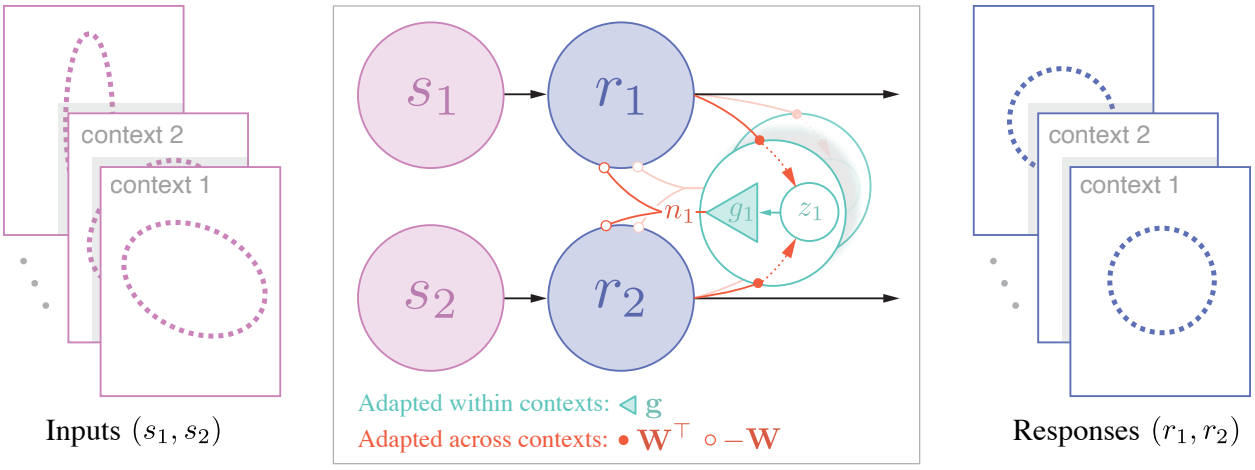

Figure 3.1: Adaptive whitening circuit, illustrated with $N = 2$ primary neurons and $K = 2$ interneurons. **Left**: Dashed ellipses representing the covariance matrices of 2D stimuli $\mathbf{s}$ drawn from different statistical contexts. **Center**: Primary neurons (shaded blue circles) receive feedforward stimulus inputs (shaded purple circles), $\mathbf{s}$, and recurrent weighted inputs, $-\mathbf{Wn}$, from the interneurons (teal circles), producing responses $\mathbf{r}$. The interneurons receive weighted inputs, $\mathbf{z} = \mathbf{W}^\top\mathbf{r}$, from the primary neurons, which are then multiplied elementwise by gains $\mathbf{g}$ to generate their outputs, $\mathbf{n} = \mathbf{g} \circ \mathbf{z}$. The gains $\mathbf{g}$ are modulated at a fast timescale to adaptively whiten within a specific stimulus context. Concurrently, the synaptic weights are optimized at a slower timescale to learn structural properties of the inputs across contexts. **Right:** Dashed unit circles representing the whitened circuit responses $\mathbf{r}$ in each statistical context.

erally associated with long-term learning and memory (Martin et al., 2000), and thus may not be a suitable biological substrate for adaptive whitening (though short-term synaptic plasticity has been reported, Zucker and Regehr, 2002). On the other hand, there is extensive neuroscience literature on rapid and reversible gain modulation (Abbott et al., 1997; Chance et al., 2002; Ferguson and Cardin, 2020; Polack et al., 2013; Salinas and Thier, 2000; Schwartz and Simoncelli, 2001; Wyrick and Mazzucato, 2021). Motivated by this, Duong et al. (2023c) proposed a mechanistic model of adaptive whitening in a neural circuit with *fixed* synaptic connections that adapts exclusively by modifying the gains of interneurons that mediate communication between the primary neurons. They demonstrate that an appropriate choice of the fixed synaptic weights can both accelerate adaptation and significantly reduce the number of interneurons that the circuit requires. However, it remains unclear how the circuit *learns* such an optimal synaptic configuration, which would seem to require synaptic plasticity.

In this study, we combine the learning and adaptation of synapses and gains, respectively, in a unified mechanistic neural circuit model that adaptively whitens its inputs over multiple timescales (Fig. 3.1). Our main contributions are as follows:

1. We introduce a novel multi-timescale adaptive whitening objective in which the (inverse) whitening matrix is factorized into a synaptic weight matrix that is optimized across contexts and a diagonal (gain) matrix that is optimized within each statistical context.

2. With this objective, we derive a multi-timescale online algorithm for adaptive whitening that can be implemented in a neural circuit comprised of primary neurons and an auxiliary population of interneurons with slow synaptic plasticity and fast gain modulation (Fig. 3.1).

3. We test our algorithm on synthetic and natural datasets, and demonstrate that the synapses learn optimal configurations over that enable the circuit to adaptively whiten its responses on short timescales exclusively using gain modulation.

Beyond the biological setting, multi-timescale learning and adaptation may also prove important in machine learning tasks. For example, Mohan et al. (2021) adjust the gains of channels in a deep denoising neural network (with pre-trained synaptic weights) to improve performance on samples with out-of-distribution noise corruption. The normative multi-timescale framework developed here offers a new approach to continual learning and test-time adaptation problems such as this.

3.3 Adaptive Symmetric Whitening

Consider a neural population with N primary neurons (Fig. 3.1). The stimulus inputs to the primary neurons are represented by a random N-dimensional vector $\mathbf{s}$ whose distribution $p(\mathbf{s}|c)$ depends on a latent context variable c. The stimulus inputs $\mathbf{s}$ can be inputs to peripheral sensory neurons (e.g., the rate at which photons hit N cones) or the postsynaptic inputs to neurons in

an early sensory area (e.g., glomerulus inputs to N mitral cells in the olfactory bulb). Context variables can include location (e.g., a forest or a meadow) and time (e.g., season or time of day). For simplicity, we assume the context-dependent inputs are centered; that is, $\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s}|c)}[\mathbf{s}] = \mathbf{0}$, where $\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s}|c)}[\cdot]$ denotes the expectation over the conditional distribution $p(\mathbf{s}|c)$ and $\mathbf{0}$ denotes the vector of zeros.

The goal of adaptive whitening is to linearly transform the inputs $\mathbf{s}$ so that, conditioned on the context variable c, the N-dimensional neural responses $\mathbf{r}$ have identity covariance matrix; that is,

$$\mathbf{r} = \mathbf{F}_c\mathbf{s} \qquad \text{such that} \qquad \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s}|c)}\left[\mathbf{r}\mathbf{r}^\top\right] = \mathbf{I}_N,$$

where $\mathbf{F}_c$ is a context-dependent $N \times N$ whitening matrix. Whitening is not a unique transformation; left multiplication of the whitening matrix $\mathbf{F}_c$ by any $N \times N$ orthogonal matrix results in another whitening matrix. We focus on symmetric whitening (also referred to as Zero-phase Components Analysis (ZCA) whitening or Mahalanobis whitening), in which the whitening matrix for context c is uniquely defined as

$$\mathbf{F}_c = \mathbf{C}_{ss}^{-1/2}(c), \qquad \mathbf{C}_{ss}(c) := \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s}|c)}\left[\mathbf{s}\mathbf{s}^\top\right]. \tag{3.1}$$

This is the unique whitening transformation that minimizes the mean-squared difference between the inputs and the outputs (Eldar and Oppenheim, 2003).

To derive an algorithm that learns the symmetric whitening matrix $\mathbf{F}_c$, we express $\mathbf{F}_c$ as the solution to an appropriate optimization problem. For a context c, we can write the *inverse* symmetric whitening matrix $\mathbf{M}_c := \mathbf{F}_c^{-1}$ as the unique optimal solution to the minimization problem

$$\mathbf{M}_c = \underset{\mathbf{M}\in\mathbb{S}_{++}^N}{\arg\min}\, f_c(\mathbf{M}), \qquad f_c(\mathbf{M}) := \mathrm{Tr}\left(\mathbf{M}^{-1}\mathbf{C}_{ss}(c) + \mathbf{M}\right), \tag{3.2}$$

where $\mathbb{S}^N_{++}$ denotes the set of $N \times N$ positive definite matrices.[1] This follows from the fact that $f_c(\mathbf{M})$ is strictly convex (under the assumption $\mathbf{C}_{ss}(c)$ is positive definite for all contexts c) with its unique minimum achieved at $\mathbf{M}_c$, where $f_c(\mathbf{M}_c) = 2\mathbf{M}_c$. Existing neural circuit models of adaptive whitening solve the minimization problem in Eq. 3.2 by choosing a matrix factorization of $\mathbf{M}_c$ and then optimizing the components (Duong et al., 2023c; Lipshutz et al., 2023; Pehlevan and Chklovskii, 2015; Pehlevan et al., 2018).

3.4 Adaptive Whitening in Neural Circuits: a Matrix Factorization Perspective

Here, we review two adaptive whitening objectives, which we unify into a single objective that adaptively whitens responses across multiple timescales.

3.4.1 Objective for adaptive whitening via synaptic plasticity

Pehlevan and Chklovskii (2015) proposed a neural circuit model that whitens neural responses by adjusting the synaptic weights between the N primary neurons and $K \geqslant N$ auxiliary interneurons according to a Hebbian update rule. Their circuit can be derived by factorizing the context-dependent matrix $\mathbf{M}_c$ into a symmetric product $\mathbf{M}_c = \mathbf{W}_c\mathbf{W}_c^\top$ for some context-dependent $N \times K$ matrix $\mathbf{W}_c$ (Lipshutz et al., 2023). Substituting this factorization into Eq. 3.2 results in the synaptic plasticity objective in Table 3.1. In the circuit implementation, $\mathbf{W}_c^\top$ denotes the weight matrix of synapses connecting primary neurons to interneurons and the matrix $-\mathbf{W}_c$ denotes the weight matrix of synapses connecting interneurons to primary neurons. Importantly, this requires the synapses $\mathbf{W}_c$ to adaptively reconfigure each time the context c changes, counter to the prevailing view that synaptic plasticity implements long-term learning and memory (Martin et al., 2000).

[1] For technical purposes, we extend the definition of f_c to all $\mathbb{S}^N$ by setting $f_c(\mathbf{M}) = \infty$ if $\mathbf{M} \notin \mathbb{S}^N_{++}$.

Table 3.1: Factorizations of the context-dependent inverse whitening matrix $\mathbf{M}_c$, and corresponding objectives for adaptive whitening circuits. The first was proposed by Pehlevan and Chklovskii (2015), the second was by Duong et al. (2023c), and the third is our proposed factorization and objective which unifies the two across timescales.

Model	Matrix factorization	Objective
Synaptic plasticity	$\mathbf{W}_c\mathbf{W}_c^\top$	$\min_{\mathbf{W}} f_c\left(\mathbf{W}\mathbf{W}^\top\right)$
Gain modulation	$\mathbf{I}_N + \mathbf{W}_{\text{fix}}\text{diag}(\mathbf{g}_c)\mathbf{W}_{\text{fix}}^\top$	$\min_{\mathbf{g}} f_c\left(\mathbf{I}_N + \mathbf{W}_{\text{fix}}\text{diag}(\mathbf{g})\mathbf{W}_{\text{fix}}^\top\right)$
Multi-timescale	$\alpha\mathbf{I}_N + \mathbf{W}\text{diag}(\mathbf{g}_c)\mathbf{W}^\top$	$\min_{\mathbf{W}} \mathbb{E}_{c\sim p(c)}\left[\min_{\mathbf{g}} f_c\left(\alpha\mathbf{I}_N + \mathbf{W}\text{diag}(\mathbf{g})\mathbf{W}^\top\right)\right]$

3.4.2 Objective for adaptive whitening via gain modulation

Duong et al. (2023c) proposed a neural circuit model with *fixed* synapses that whitens the N primary responses by adjusting the multiplicative gains in a set of K auxiliary interneurons. To derive a neural circuit with gain modulation, they considered a novel diagonalization of the inverse whitening matrix, $\mathbf{M}_c = \mathbf{I}_N + \mathbf{W}_{\text{fix}}\text{diag}(\mathbf{g}_c)\mathbf{W}_{\text{fix}}^\top$, where $\mathbf{W}_{\text{fix}}$ is an arbitrary, but fixed $N \times K$ matrix of synaptic weights (with $K \geq K_N := N(N+1)/2$) and $\mathbf{g}_c$ is an adaptive, context-dependent real-valued K-dimensional vector of gains. Note that unlike the conventional eigendecomposition, the number of elements along the diagonal matrix is significantly larger than the dimensionality of the input space. Substituting this factorization into Eq. 3.2 results in the gain modulation objective in Table 3.1. As in the synaptic plasticity model, $\mathbf{W}_{\text{fix}}^\top$ denotes the weight matrix of synapses connecting primary neurons to interneurons while $-\mathbf{W}_{\text{fix}}$ connects interneurons to primary neurons. In contrast to the synaptic plasticity model, the interneuron outputs are modulated by context-dependent multiplicative gains, $\mathbf{g}_c$, that are adaptively adjusted to whiten the circuit responses.

Duong et al. (2023c) demonstrate that an appropriate choice of the fixed synaptic weight matrix can both accelerate adaptation and significantly reduce the number of interneurons in the circuit. In particular, the gain modulation circuit can whiten *any* input distribution provided the gains vector $\mathbf{g}_c$ has dimension $K \geqslant K_N$ (the number of degrees of freedom in an $N \times N$ symmet-

ric covariance matrix). However, in practice, the circuit need only adapt to input distributions corresponding to *natural* input statistics (Barlow, 1961; Ganguli and Simoncelli, 2014; Młynarski and Hermundstad, 2021; Simoncelli and Olshausen, 2001). For example, the statistics of natural images are approximately translation-invariant, which significantly reduces the degrees of freedom in their covariance matrices, from $O(N^2)$ to $O(N)$. Therefore, while the space of all possible correlation structures is K_N-dimensional, the set of natural statistics likely has far fewer degrees of freedom and an optimal selection of the weight matrix $\mathbf{W}_{\text{fix}}$ can potentially offer dramatic reductions in the number of interneurons K required to adapt. As an example, Duong et al. (2023c) specify a weight matrix for performing "local" whitening with $O(N)$ interneurons when the input correlations are spatially-localized (e.g., as in natural images). However, they do not prescribe a method for *learning* a (synaptic) weight matrix that is optimal across the set of natural input statistics.

3.4.3 Unified objective for adaptive whitening via synaptic plasticity and gain modulation

We unify and generalize the two disparate adaptive whitening approaches (Duong et al., 2023c; Pehlevan and Chklovskii, 2015) in a single *multi-timescale* nested objective in which gains $\mathbf{g}$ are optimized within each context and synaptic weights $\mathbf{W}$ are optimized across contexts. In particular, we optimize, with respect to $\mathbf{W}$, the expectation of the objective from (Duong et al., 2023c) (for some fixed $K \geqslant 1$) over the distribution of contexts $p(c)$:

$$\min_{\mathbf{W}\in\mathbb{R}^{N\times K}} \mathbb{E}_{c\sim p(c)} \left[\min_{\mathbf{g}\in\mathbb{R}^K} f_c \left(\alpha \mathbf{I}_N + \mathbf{W}\text{diag}(\mathbf{g})\mathbf{W}^\top \right) \right], \tag{3.3}$$

where we have also generalized the objective from (Duong et al., 2023c) by including a fixed multiplicative factor $\alpha \geqslant 0$ in front of the identity matrix $\mathbf{I}_N$, and we have relaxed the requirement that $K \geqslant K_N$.

What is an optimal solution of Eq. 3.3? Since the convex function f_c is (uniquely) minimized at $\mathbf{M}_c$, a sufficient condition for the optimality of a synaptic weight matrix $\mathbf{W}$ is that for each context c, there is a gains vector $\mathbf{g}_c$ such that $\alpha\mathbf{I}_N + \mathbf{W}\mathrm{diag}(\mathbf{g}_c)\mathbf{W}^\top = \mathbf{M}_c$. Importantly, under such a synaptic configuration, the function f_c can attain its minimum exclusively by adjusting the gains vector $\mathbf{g}$. In the space of covariance matrices, we can express the statement as

$$\mathbf{C}_{ss}(c) \in \mathbb{F}(\mathbf{W}) := \left\{ \left[\alpha\mathbf{I}_N + \mathbf{W}\mathrm{diag}(\mathbf{g})\mathbf{W}^\top\right]^2 : \mathbf{g} \in \mathbb{R}^K \right\} \cap \mathbb{S}^N_{++} \quad \text{for every context } c,$$

where $\mathbb{F}(\mathbf{W})$ contains the set of covariance matrices that can be whitened with fixed synapses $\mathbf{W}$ and adaptive gains $\mathbf{g}$. Fig. 3.2 provides an intuitive Venn diagram comparing a non-optimal synaptic configuration $\mathbf{W}_0$ and an optimal synaptic configuration $\mathbf{W}_T$.

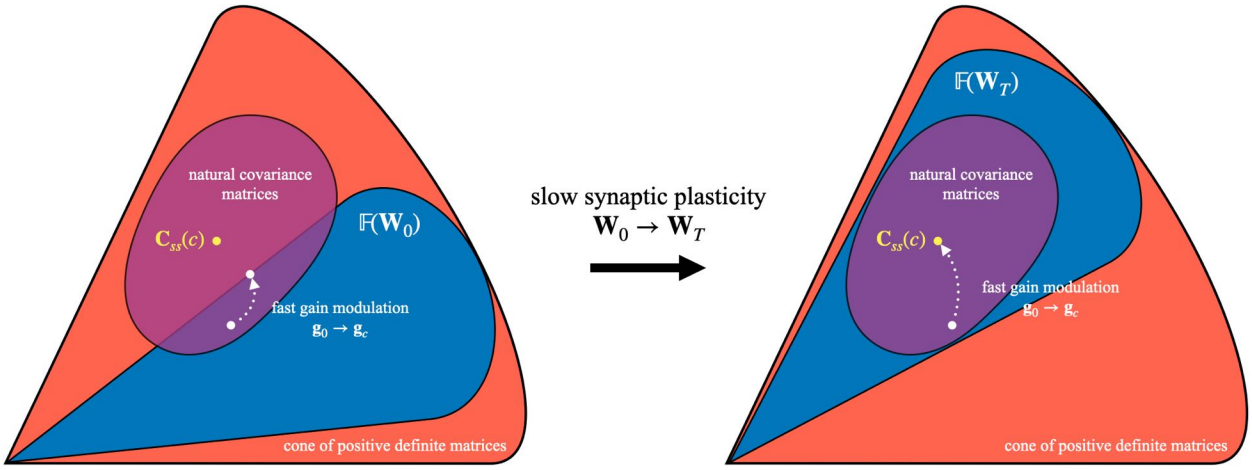

Figure 3.2: Illustration of multi-timescale learning in the space of covariance matrices. Orange and purple regions (identical on the left and right) respectively represent the cone of all positive definite matrices $\mathbb{S}^N_{++}$, and the subset of naturally-occurring covariance matrices $\{\mathbf{C}_{ss}(c)\}$. Blue regions represent the set of covariance matrices that can be whitened with adaptive gains for a particular synaptic weight matrix. On each side, the yellow circle denotes a naturally-occurring input covariance matrix $\mathbf{C}_{ss}(c)$ and the dotted white curve illustrates the trajectory of covariance matrices the circuit is adapted to whiten as the gains are modulated (with fixed synapses, note the dotted white curve remains in the blue region). **Left:** With initial synaptic weights $\mathbf{W}_0$ the circuit cannot whiten some natural input distributions exclusively via gain modulation, i.e., $\{\mathbf{C}_{ss}(c)\} \not\subset \mathbb{F}(\mathbf{W}_0)$. **Right:** After learning optimal synaptic weights $\mathbf{W}_T$, the circuit can match any naturally-occurring covariance matrix using gain modulation, i.e., $\{\mathbf{C}_{ss}(c)\} \subset \mathbb{F}(\mathbf{W}_T)$.

3.5 Multi-timescale Adaptive Whitening Algorithm and Circuit Implementation

In this section, we derive an online algorithm for optimizing the multi-timescale objective in Eq. 3.3, then map the algorithm onto a neural circuit with fast gain modulation and slow synaptic plasticity. To derive an online algorithm that includes neural dynamics, we first add neural responses $\mathbf{r}$ to the objective, which introduces a third timescale to the objective. We then derive a multi-timescale gradient-based algorithm for optimizing the objective.

Adding neural responses to the objective. First, observe that we can write $f_c(\mathbf{M})$, for $\mathbf{M} \in \mathbb{S}^N_{++}$, in terms of the neural responses $\mathbf{r}$:

$$f_c(\mathbf{M}) = \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s}|c)}\left[\max_{\mathbf{r}\in\mathbb{R}^N} \mathrm{Tr}\left(2\mathbf{r}\mathbf{s}^\top - \mathbf{M}\mathbf{r}\mathbf{r}^\top + \mathbf{M}\right)\right].$$

To see this, maximize over $\mathbf{r}$ to obtain $\mathbf{r} = \mathbf{M}^{-1}\mathbf{s}$ and then use the definition of $\mathbf{C}_{ss}(c)$ from Eq. 3.1. Substituting this expression for f_c, with $\mathbf{M} = \alpha\mathbf{I}_N + \mathbf{W}\mathrm{diag}(\mathbf{g})\mathbf{W}^\top$, into Eq. 3.3, dropping the constant term $\alpha\mathbf{I}_N$ term and using the cyclic property of the trace operator results in the following objective with 3 nested optimizations:

$$\min_{\mathbf{W}\in\mathbb{R}^{N\times K}} \mathbb{E}_{c\sim p(c)}\left[\min_{\mathbf{g}\in\mathbb{R}^K} \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s}|c)}\left[\max_{\mathbf{r}\in\mathbb{R}^N} \ell(\mathbf{W},\mathbf{g},\mathbf{r},\mathbf{s})\right]\right], \tag{3.4}$$

$$\text{where}\quad \ell(\mathbf{W},\mathbf{g},\mathbf{r},\mathbf{s}) := 2\mathbf{r}^\top\mathbf{s} - \alpha\|\mathbf{r}\|^2 - \sum_{i=1}^K g_i\left[(\mathbf{w}_i^\top\mathbf{r})^2 - \|\mathbf{w}_i\|^2\right].$$

The inner-most optimization over $\mathbf{r}$ corresponds to neural responses and will lead to recurrent neural dynamics. The outer 2 optimizations correspond to the optimizations over the gains $\mathbf{g}$ and synaptic weights $\mathbf{W}$ from Eq. 3.3.

To solve Eq. 3.4 in the online setting, we assume there is a timescale separation between

neural dynamics and the gain/weight updates. This allows us to perform the optimization over $\mathbf{r}$ before optimizing $\mathbf{g}$ and $\mathbf{W}$ concurrently. This is biologically sensible: neural responses (e.g., action potential firing) operate on a much faster timescale than gain modulation and synaptic plasticity (Ferguson and Cardin, 2020; Wang et al., 2003). In Appx. B.1, we also consider the case there is also a timescale separation between the gain updates and weight updates, so that the weights are optimized after the gains have equilibrated.

Recurrent neural dynamics. At each iteration, the circuit receives a stimulus $\mathbf{s}$. We maximize $\ell(\mathbf{W}, \mathbf{g}, \mathbf{r}, \mathbf{s})$ with respect to $\mathbf{r}$ by iterating the following gradient-ascent steps that correspond to repeated timesteps of the recurrent circuit (Fig. 3.1) until the responses equilibrate:

$$\mathbf{r} \leftarrow \mathbf{r} + \eta_r \left(\mathbf{s} - \sum_{i=1}^{K} n_i \mathbf{w}_i - \alpha \mathbf{r} \right), \tag{3.5}$$

where $\eta_r > 0$ is a small constant, $z_i = \mathbf{w}_i^\top \mathbf{r}$ denotes the weighted input to the i^{th} interneuron, $n_i = g_i z_i$ denotes the gain-modulated output of the i^{th} interneuron. For each i, synaptic weights, $\mathbf{w}_i$, connect the primary neurons to the i^{th} interneuron and symmetric weights, $-\mathbf{w}_i$, connect the i^{th} interneuron to the primary neurons. From Eq. 3.5, we see that the neural responses are driven by feedforward stimulus inputs $\mathbf{s}$, recurrent weighted feedback from the interneurons $-\mathbf{W}\mathbf{n}$, and a leak term $-\alpha\mathbf{r}$.

Fast gain modulation and slow synaptic plasticity. After the neural activities equilibrate, we minimize $\ell(\mathbf{W}, \mathbf{g}, \mathbf{r}, \mathbf{s})$ by taking concurrent gradient-descent steps

$$\Delta g_i = \eta_g \left(z_i^2 - \|\mathbf{w}_i\|^2 \right)$$
$$\Delta \mathbf{w}_i = \eta_w \left(\mathbf{r} n_i - \mathbf{w}_i g_i \right),$$

where η_g and η_w are the respective learning rates for the gains and synaptic weights. By choosing $\eta_g \gg \eta_w$, we ensure that the gains are updated at a faster timescale than the synaptic weights.

The update to the i^{th} interneuron's gain g_i depends on the difference between online estimate of the variance of its input, z_i^2, and the squared-norm of the i^{th} synaptic weight vector, $\|\mathbf{w}_i\|^2$, quantities that are both locally available to the i^{th} interneuron. Using the fact that $z_i = \mathbf{w}_i^\top \mathbf{r}$, we can rewrite the gain update as $\Delta g_i = \eta_g[\mathbf{w}_i^\top(\mathbf{r}\mathbf{r}^\top - \mathbf{I}_N)\mathbf{w}_i]$. From this expression, we see that the gains equilibrate when the marginal variance of the responses along the direction $\mathbf{w}_i$ is 1, for $i = 1, \ldots, K$.

The update to the $(i, j)^{\text{th}}$ synaptic weight w_{ij} is proportional to the difference between $r_i n_j$ and $w_{ij} g_j$, which depends only on variables that are available in the pre- and postsynaptic neurons. Since $r_i n_j$ is the product of the pre- and postynaptic activities, we refer to this update as *Hebbian*. In Appx. B.3.2, we decouple the feedforward weights $\mathbf{w}_i^\top$ and feedback weights $-\mathbf{w}_i$ and provide conditions under which the symmetry asymptotically holds.

Multi-timescale online algorithm. Combining the neural dynamics, gain modulation and synaptic plasticity yields our online multi-timescale adaptive whitening algorithm, Alg. 2, which we express is vector-matrix form with '$\circ$' denoting the Hadamard (elementwise) product of two vectors.

Algorithm 2: Multi-timescale adaptive whitening via synaptic plasticity and gain modulation

Input: $\mathbf{s}_1, \mathbf{s}_2, \dots \in \mathbb{R}^N$
Initialize: $\mathbf{W} \in \mathbb{R}^{N \times K}$; $\mathbf{g} \in \mathbb{R}^K$; $\eta_r > 0$; $\eta_g \gg \eta_w > 0$
for $t = 1, 2, \dots$ **do**
  $\mathbf{r}_t \leftarrow \mathbf{0}$
  while not converged **do**
    $\mathbf{z}_t \leftarrow \mathbf{W}^\top \mathbf{r}_t$
    $\mathbf{n}_t \leftarrow \mathbf{g} \circ \mathbf{z}_t$
    $\mathbf{r}_t \leftarrow \mathbf{r}_t + \eta_r \left(\mathbf{s}_t - \mathbf{W}\mathbf{n}_t - \alpha \mathbf{r}_t\right)$
  end while
  $\mathbf{g} \leftarrow \mathbf{g} + \eta_g \left(\mathbf{z}_t \circ \mathbf{z}_t - \mathrm{diag}\left(\mathbf{W}^\top \mathbf{W}\right)\right)$
  $\mathbf{W} \leftarrow \mathbf{W} + \eta_w \left(\mathbf{r}_t \mathbf{n}_t^\top - \mathbf{W}\mathrm{diag}(\mathbf{g})\right)$
end for

Alg. 2 is naturally viewed as a *unification* and generalization of previously proposed neural circuit models for adaptation. When $\alpha = 0$ and the gains $\mathbf{g}$ are constant (e.g., $\eta_g = 0$) and identically equal to the vector of ones $\mathbf{1}$ (so that $\mathbf{n}_t = \mathbf{z}_t$), we recover the synaptic plasticity algorithm from (Lipshutz et al., 2023). Similarly, when $\alpha = 1$ and the synaptic weights $\mathbf{W}$ are fixed (e.g., $\eta_w = 0$), we recover the gain modulation algorithm from (Duong et al., 2023c).

3.6 Numerical Experiments

We test Alg. 2 on stimuli $\mathbf{s}_1, \mathbf{s}_2, \dots$ drawn from slowly fluctuating latent contexts $c_1, c_2, \dots$; that is, $\mathbf{s}_t \sim p(\mathbf{s}|c_t)$ and $c_t = c_{t-1}$ with high probability.[2] To measure performance, we evaluate the operator norm on the difference between the expected response covariance and the identity matrix:

$$\mathrm{Error}(t) = \|\mathbf{M}_t^{-1} \mathbf{C}_{ss}(c_t) \mathbf{M}_t^{-1} - \mathbf{I}_N\|_{\mathrm{op}}, \qquad \mathbf{M}_t := \alpha \mathbf{I}_N + \mathbf{W}_t \mathrm{diag}(\mathbf{g}) \mathbf{W}_t^\top. \tag{3.6}$$

[2]Note: A Python (numPy) implementation of our algorithm is included with our submission. Code to reproduce all figures will be released on GitHub at the time of publication.

Geometrically, this "worst-case" error measures the maximal Euclidean distance between the ellipsoid corresponding to $\mathbf{M}_t^{-1}\mathbf{C}_{ss}(c_t)\mathbf{M}_t^{-1}$ and the $(N-1)$-sphere along all possible axes. To compare two synaptic weight matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N\times K}$, we evaluate $\|\hat{\mathbf{A}} - \mathbf{B}\|_F$, where $\hat{\mathbf{A}} = \mathbf{AP}$ and $\mathbf{P}$ is the permutation matrix (with possible sign flips) that minimizes the error.

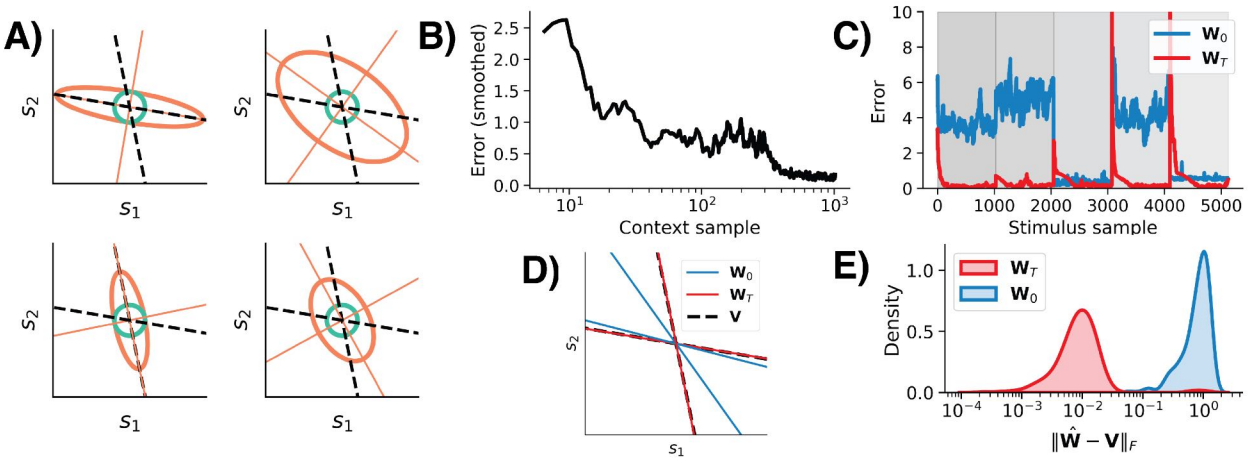

Figure 3.3: Adaptive whitening of a synthetic dataset with $N = 2$, η_w = 1E-5, η_g = 5E-2. **A)** Covariance ellipses (orange) of 4 out of 64 synthesized contexts. Black dashed lines are axes corresponding to the column vectors of $\mathbf{V}$. The unit circle is shown in green. Since the column vectors of $\mathbf{V}$ are not orthogonal, these covariance matrices do *not* share a common set of eigenvectors (orange lines). **B)** Whitening error at the end of each context presentation of 1E3 samples. We apply a moving average window of 10 stimulus samples. **C)** Error at each stimulus presentation within five different contexts (gray panels), presented with $\mathbf{W}_0$, or $\mathbf{W}_T$. **D)** Column vectors of $\mathbf{W}_0$, $\mathbf{W}_T$, $\mathbf{V}$ (each axis corresponds to the span of one column vector in $\mathbb{R}^2$). **E)** Smoothed distributions of error (in Frobenius norm) between $\hat{\mathbf{W}}$ and $\mathbf{V}$ across 250 random initializations of $\mathbf{W}_0$.

3.6.1 Synthetic dataset

To validate our model, we first consider a 2-dimensional synthetic dataset in which an optimal solution is known. Suppose that each context-dependent inverse whitening matrix is of the form $\mathbf{M}_c = \mathbf{I}_N + \mathbf{V}\mathbf{\Lambda}(c)\mathbf{V}^\top$, where $\mathbf{V}$ is a fixed 2×2 matrix and $\mathbf{\Lambda}(c) = \mathrm{diag}(\lambda_1(c), \lambda_2(c))$ is a context-dependent diagonal matrix. Then, in the case $\alpha = 1$ and $K = 2$, an optimal solution of the objective in Eq. 3.3 is when the column vectors of $\mathbf{W}$ align with the column vectors of $\mathbf{V}$.

To generate this dataset, we chose the column vectors of $\mathbf{V}$ uniformly from the unit circle, so

they are *not* generally orthogonal. For each context $c = 1, \ldots, 64$, we assume the diagonal entries of $\mathbf{\Lambda}(c)$ are sparse and i.i.d.: with probability 1/2, $\lambda_i(c)$ is set to zero and with probability 1/2, $\lambda_i(c)$ is chosen uniformly from the interval $[0, 4]$. Example covariance matrices from different contexts are shown in Fig. 3.3A (note that they do *not* share a common eigen-decomposition). Finally, for each context, we generate 1E3 i.i.d. samples with context-dependent distribution $\mathbf{s} \sim \mathcal{N}(\mathbf{0}, \mathbf{M}_c^2)$.

We test Alg. 2 with $\alpha = 1$, $K = 2$, $\eta_w =$ 1E-5, and $\eta_g =$ 5E-2 on these sequences of synthetic inputs with the column vectors of $\mathbf{W}_0$ chosen uniformly from the unit circle. The model successfully learns to whiten the different contexts, as indicated by the decreasing whitening error with the number of contexts presented (Fig. 3.3B). At the end of training, the synaptic weight matrix $\mathbf{W}_T$ is optimized such that the circuit can adapt to changing contexts exclusively by adjusting its gains. This is evidenced by the fact that when the context changes, there is a brief spike in error as the gains adapt to the new context (Fig. 3.3C, red line). By contrast, the error remains high in many of the contexts when using the initial random synaptic weight matrix $\mathbf{W}_0$ (Fig. 3.3C, blue line). In particular, the synapses learn (across contexts) an optimal configuration in the sense that the column vectors of $\mathbf{W}$ learn to align with the column vectors of $\mathbf{V}$ over the course of training (Fig. 3.3DE).

3.6.2 Natural images dataset

By hand-crafting a particular set of synaptic weights, Duong et al. (2023c) showed that their adaptive whitening network can approximately whiten a dataset of natural image patches with $O(N)$ gain-modulating interneurons instead of $O(N^2)$. Here, we show that our model can exploit spatial structure across natural scenes to *learn* an optimal set of synaptic weights by testing our algorithm on 56 high-resolution natural images (van Hateren and van der Schaaf, 1998) (Fig. 3.4A, top). For each image, which corresponds to a separate context c, 5×5 pixel image patches are randomly sampled and vectorized to generate context-dependent samples $\mathbf{s} \in \mathbb{R}^{25}$ with covariance matrix $\mathbf{C}_{ss}(c) \in \mathbb{S}_{++}^{25}$ (Fig. 3.4A, bottom). We train our algorithm with a timescale separation

between gain and weight updates (Appx. B.1, Alg. 5, η_w =5E-2) with $K = N = 25$ and random $\mathbf{W}_0 \in O(25)$ on a training set of 50 of the images, presented uniformly at random 2E4 total times. We find that the model successfully learns a basis that enables adaptive whitening *across* different visual contexts via gain modulation, as shown by the decreasing training error (Eq. 3.6) in Fig. 3.4B.

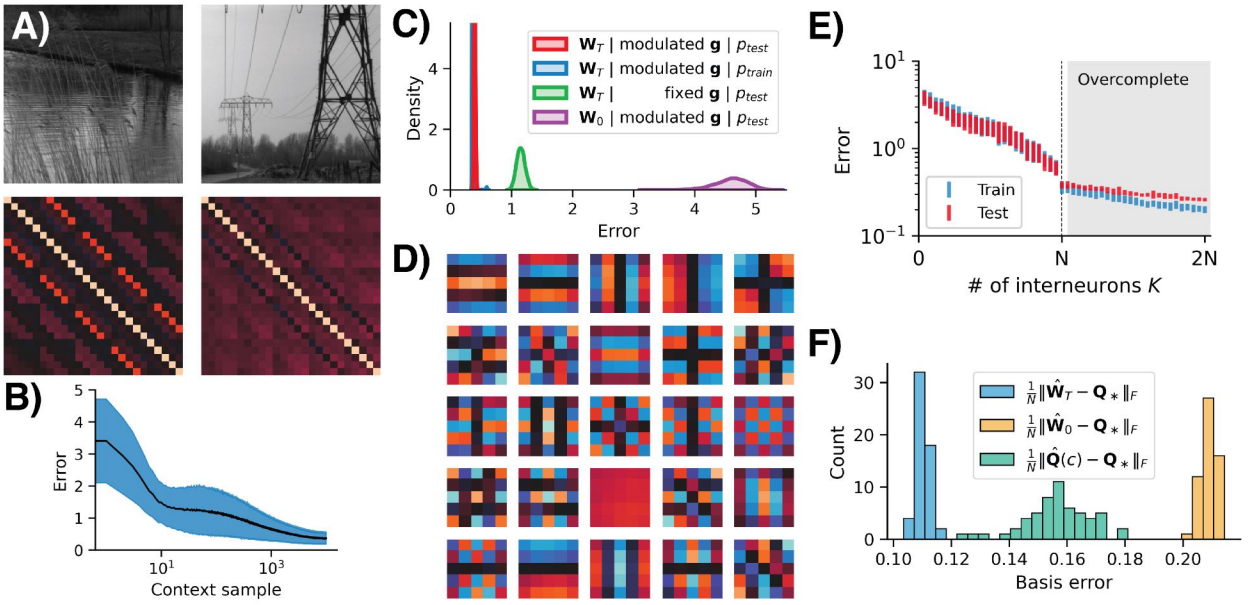

Figure 3.4: Adaptive whitening of natural images. **A)** Examples of 2 out of 56 high-resolution images (top) with each image corresponding to a separate context. For each image, 5×5 pixel patches are randomly sampled to generate context-dependent stimuli with covariance matrix $\mathbf{C}_{ss}(c) \in \mathbb{S}^{25}_{++}$ (bottom). **B)** Mean error during training (Eq. 3.6) with $K = N = 25$. Shaded region is standard deviation over 2E3 random initializations $\mathbf{W}_0 \in O(25)$. **C)** Smoothed distributions of average adaptive whitening error over all 2E3 initializations. The red distribution corresponds to the error on the held-out images with fixed learned synapses $\mathbf{W}_T$ and modulated gains $\mathbf{g}$. The blue (resp. green, purple) distribution corresponds to the same error, but tested on the training images (resp. with fixed gains equal to the average gains over the final 100 iterations, with fixed random synapses $\mathbf{W}_0$). **D)** The learned weights (re-shaped columns of $\mathbf{W}_T$) approximate orthogonal 2D sinusoids. **E)** Final error (after T = 5E4 iterations) as a function of number of interneurons K. Bars are standard deviations centered on the mean error at each K. **F)** Frobenius norm between the eigenbasis of $\mathbb{E}_{c\sim p(c)}[\mathbf{C}_{ss}(c)]$ (i.e. across all contexts), $\mathbf{Q}_*$, with $\mathbf{W}_T$, $\mathbf{W}_0$, and eigenbasis of each individual context covariance, $\mathbf{Q}(c)$, when $K = N = 25$. See Appx. B.2 for details.

How does the network learn to leverage statistical structure that is consistent across contexts? We test the circuit with fixed synaptic weights $\mathbf{W}_T$ and modulated (adaptive) gains $\mathbf{g}$ on stimuli from the held-out images (Fig. 3.4C, red distribution shows the smoothed error over 2E3

random initializations $\mathbf{W}_0$). The circuit performs as well on the held-out images as on the training images (Fig. 3.4C, red versus blue distributions). In addition, the circuit with learned synaptic weights $\mathbf{W}_T$ and modulated gains $\mathbf{g}$ outperforms the circuit with learned synaptic weights $\mathbf{W}_T$ and fixed gains (Fig. 3.4C, green distribution), and significantly outperforms the circuit with random synaptic weights $\mathbf{W}_0$ and modulated gains (Fig. 3.4C, purple distribution). Together, these results suggest that the circuit learns features $\mathbf{W}_T$ that enable the circuit to adaptively whiten across statistical contexts exclusively using gain modulation, and that gain modulation is crucial to the circuit's ability to adaptively whiten. In Fig. 3.4D, we visualize the learned filters (columns of $\mathbf{W}_T$), and find that they are approximately equal to the 2D discrete cosine transform (DCT, Appx. B.2), an orthogonal basis that is known to approximate the eigenvectors of natural image patch covariances (Ahmed et al., 1974; Bull and Zhang, 2021).

To test how the number of interneurons K impacts the performance of the circuit, we train the algorithm with $K = 1, \ldots, 2N$ and report the final error in Fig. 3.4E. There is a steady drop in error as K ranges from 1 to N, at which point there is a (discontinuous) drop in error followed by a continued, but more gradual decay in *both* training and test images error as K ranges from N to $2N$ (the overcomplete regime). To understand this behavior, note that the covariance matrices of image patches *approximately* share an eigen-decomposition (Bull and Zhang, 2021). To see this, let $\mathbf{Q}(c)$ denote the orthogonal matrix of eigenvectors corresponding to the context-dependent covariance matrix $\mathbf{C}_{ss}(c)$. As shown in Fig. 3.4F (green histogram), there is a small, but non-negligible, difference between the eigenvectors $\mathbf{Q}(c)$ and the eigenvectors $\mathbf{Q}_*$ of the *average* covariance matrix $\mathbb{E}_{c\sim p(c)}[\mathbf{C}_{ss}(c)]$. When $K = N$, the column vectors of $\mathbf{W}_T$ learn to align with $\mathbf{Q}_*$ (as shown in Fig. 3.4F, blue histogram), and the circuit *approximately* adaptively whitens the context-dependent stimulus inputs via gain modulation. As K ranges from 1 to N, $\mathbf{W}_T$ progressively learns the eigenvectors of $\mathbf{Q}_*$ (Appx. B.2). Since $\mathbf{W}_T$ achieves a full set of eigenvectors at $K = N$, this results in a large drop in error when measured using the operator norm. Finally, as mentioned, there is a non-negligible difference between the eigenvectors $\mathbf{Q}(c)$ and the eigenvec-

tors $\mathbf{Q}_*$. Therefore, increasing the number of interneurons from N to $2N$ allows the circuit to discover basis vectors $\mathbf{W}_T$ to account for the small deviations between $\mathbf{Q}(c)$ and $\mathbf{Q}_*$, resulting in improved whitening error (Appx. B.2).

3.7 Discussion

Our normative derivation relies on a novel multi-timescale objective (Eq. 3.3) in which the (inverse) whitening matrix is factorized into components that are optimized at different timescales. This model draws inspiration from the extensive neuroscience literature on rapid gain modulation (Ferguson and Cardin, 2020) and long-term synaptic plasticity (Martin et al., 2000), and concretely proposes complementary roles for these computations: synaptic plasticity facilitates learning features that are invariant *across* statistical contexts while gain modulation facilitates adaptation *within* a statistical context. Experimental support for this will come from detailed understanding of natural sensory statistics across statistical contexts and estimates of (changes in) synaptic connectivity from wiring diagrams (e.g. Wanner and Friedrich, 2020) or neural activities (e.g. Linderman et al., 2014).

Our circuit uses local learning rules for the gain and synaptic weight updates, so it serves as a plausible model of neural computation and can potentially be implemented in low-power neuromorphic hardware (Pehlevan and Chklovskii, 2019). However, there are aspects of our circuit that are not biologically realistic. For example, we do not sign-constrain the gains or synaptic weight matrices, so our circuit can violate Dale's law. In addition, the feedforward synaptic weights $\mathbf{W}^\top$ and feedback weights $-\mathbf{W}$ are constrained to be symmetric. In Appx. B.3, we consider modifications of our model to be more biologically realistic. Additionally, while we focus on the potential joint function of gain modulation and synaptic plasticity in adaptation, short-term synaptic plasticity, which operates on similar timescales as gain modulation, has also been reported (Zucker and Regehr, 2002). Theoretical studies suggest that short-term synaptic plasticity

is useful in multi-timescale learning tasks (Aitken and Mihalas, 2023; Masse et al., 2019; Tsodyks et al., 1998) and it may also contribute to multi-timescale adaptive whitening. Ultimately, support for different adaptation mechanisms will be adjudicated by experimental observations.

The work we present here may also be relevant beyond the biological setting. Decorrelation and whitening transformations are common preprocessing steps in statistical and machine learning methods (Bell and Sejnowski, 1996; Coates et al., 2011; Hyvärinen and Oja, 2000; Krizhevsky et al., 2009; Olshausen and Field, 1996), and are useful for preventing representational collapse in recent self-supervised learning methods (Bardes et al., 2022; Ermolov et al., 2021; Hua et al., 2021; Zbontar et al., 2021). Therefore, our online multi-timescale algorithm may be useful for developing online adaptive self-supervised learning algorithms. In addition, our work is related to the general problem of online meta-learning (Finn et al., 2019; Thrun and Pratt, 2012); that is, learning methods that can rapidly adapt to new tasks. Our solution—which is closely related to mechanisms of test-time feature gain modulation developed for machine learning models for denoising (Mohan et al., 2021), compression (Ballé et al., 2020; Duong et al., 2023b), and classification (Wang et al., 2020)—suggests a general approach to meta-learning inspired by neuroscience: structural properties of the tasks (contexts) are encoded in synaptic weights and adaptation to the current task (context) is achieved by adjusting the gains of individual neurons.

4 Adaptive Coding Efficiency in Recurrent Cortical Circuits via Gain Control

4.1 Overview

An earlier version work in this chapter was presented at Computational and Systems Neuroscience (2022), and is published in preprint form (currently under review; Duong et al., 2023a).

Sensory systems across all modalities and species exhibit adaptation to continuously changing input statistics. Individual neurons have been shown to modulate their response gains so as to maximize information transmission in different stimulus contexts. Experimental measurements have revealed additional, nuanced sensory adaptation effects including changes in response maxima and minima, tuning curve repulsion from the adapter stimulus, and stimulus-driven response decorrelation. Existing explanations of these phenomena rely on changes in inter-neuronal synaptic efficacy, which, while more flexible, are unlikely to operate as rapidly or reversibly as single neuron gain modulations. Using published V1 population adaptation data, we show that propagation of single neuron gain changes in a recurrent network is sufficient to capture the entire set of observed adaptation effects. We propose a novel adaptive efficient coding objective with which single neuron gains are modulated, maximizing the fidelity of the stimulus

representation while minimizing overall activity in the network. From this objective, we analytically derive a set of gains that optimize the trade-off between preserving information about the stimulus and conserving metabolic resources. Our model generalizes well-established concepts of single neuron adaptive gain control to recurrent populations, and parsimoniously explains experimental adaptation data.

4.2 Introduction

Some of the earliest neurophysiological recordings showed that repeated or prolonged stimulus presentation leads to a relative decrease in neural responses (Adrian and Zotterman, 1926). Indeed, neurons across different species, brain areas, and sensory modalities adjust their gains (i.e. input-output sensitivity) in response to recent stimulus history (Kohn, 2007; Weber et al., 2019, for reviews). Gain control provides a mechanism for single neurons to rapidly and reversibly adapt to different stimulus contexts (Abbott et al., 1997; Brenner et al., 2000; Fairhall et al., 2001; Muller et al., 1999; Młynarski and Hermundstad, 2021) while preserving synaptic weights that serve to represent features that remain consistent across contexts (Ganguli and Simoncelli, 2014). From a normative standpoint, this allows a single neuron to adjust the dynamic range of its responses to accommodate changes in input statistics (Fairhall et al., 2001; Laughlin, 1981) – a core tenet of theories of efficient sensory coding (Attneave, 1954; Barlow, 1961).

Experimental measurements, however, reveal that adaptation induces additional complex changes in neural responses, including tuning-dependent reductions in both response maxima and minima (Movshon and Lennie, 1979), tuning curve repulsion (Hershenhoren et al., 2014; Shen et al., 2015; Yaron et al., 2012), and stimulus-driven decorrelation (Benucci et al., 2013; Gutnisky and Dragoi, 2008; Muller et al., 1999; Wanner and Friedrich, 2020). Although coding efficiency and gain-mediated adaptation is well studied in single neurons, it appears as though these nuanced empirical observations require a more complex adaptation mechanism, involving *joint* coordi-

nation among neurons in the population. Indeed, to explain these phenomena, previous studies have relied on adaptive changes in feedforward or recurrent synaptic efficacy (i.e. by changing the entire network's set of synaptic weights; Młynarski and Hermundstad, 2021; Rast and Drugowitsch, 2020; Wainwright et al., 2001; Westrick et al., 2016). However, this requires synaptic weights to continuously remap under different statistical contexts, which may change significantly and transiently at short time scales.

Here, we hypothesize that adaptation effects reported in neural population recording data can be explained by combining normative theory with a mechanistic recurrent population model that includes single neuron gain modulation. The primary contributions of our study are as follows:

1. We introduce an analytically tractable recurrent neural network (RNN) architecture for adaptive gain control, in which single neurons adjust their gains in response to novel stimulus statistics. The model respects experimental evidence that cortical anatomy is dominated by recurrence (Douglas and Martin, 2007), allowing the effects of single neuron gain changes to propagate through lateral connections.

2. We propose a novel *adaptive efficient coding* objective for adjustment of the single neuron gains, which optimizes coding fidelity of the stimulus ensemble, subject to metabolic and homeostatic constraints.

3. Through numerical simulations, we compare model predictions to experimental measurements of cat V1 neurons responding to a sequence of gratings drawn from an ensemble with either uniform or biased orientation probability (Benucci et al., 2013). We show that adaptive adjustment of neural gains, with no changes in synaptic strengths, parsimoniously captures the full set of adaptation phenomena observed in the data.

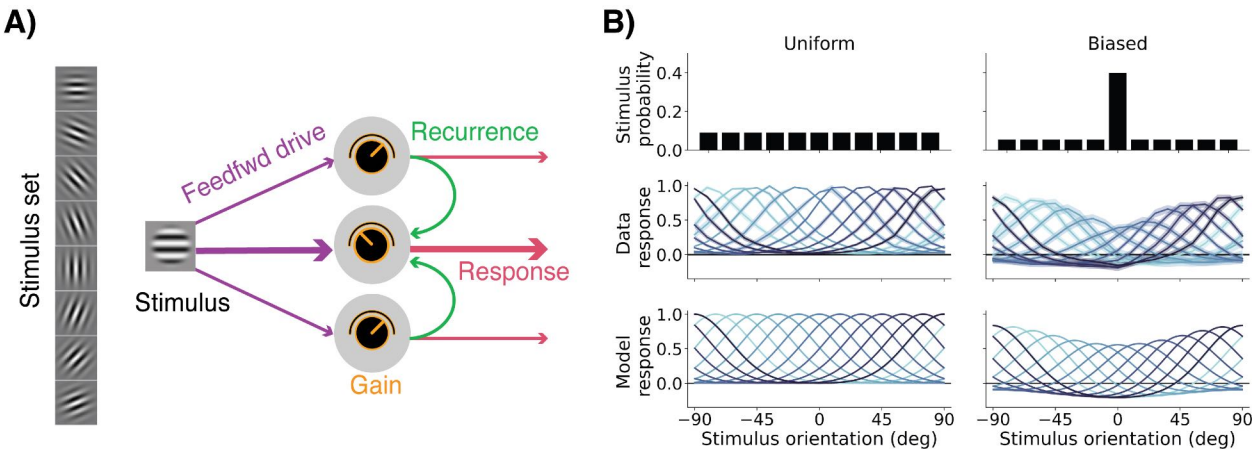

Figure 4.1: Recurrent adaptation model. **A)** A population of recurrently-connected orientation-tuned cells receives external feedforward drive (purple arrows) from a presented oriented grating stimulus, randomly sampled from a set of possible orientations. The width of the arrow denotes the strength of the drive, and indicates that the center neuron is tuned towards the horizontal-oriented stimuli. The feedforward drive of each neuron is multiplicatively modulated by its a scalar gain (orange dials). Lateral recurrent input between neurons is denoted by green arrows. Recurrent connectivity is all-to-all, with synaptic strengths determined by the distance between neurons' preferred feedforward orientation. Output responses (red) of each neuron are a function of both feedforward drive and recurrent drive. **B)** Response tuning curves for orientation-tuned units to stimuli presented with uniform probability (left column), or biased probability (right column). Middle row shows recordings of neurons in visual area V1 of cats, aggregated over 11 sessions. Bottom row shows model responses. Shaded regions are standard error of the mean (SEM).

4.3 Related Work

Models of statistical adaptation in neural populations. While evidence for adaptive efficient coding via gain modulation in single neurons is relatively well understood (Fairhall et al., 2001; Młynarski and Hermundstad, 2021; Nagel and Doupe, 2006), the question of whether neural *population* adaptation can be explained by efficient coding and gain modulation remains underexplored. Normative models of population adaptation have generally relied on synaptic plasticity (i.e. between-neuron synaptic weight adjustments) as the mechanism mediating adaptation (Lipshutz et al., 2023; Młynarski and Hermundstad, 2021; Pehlevan and Chklovskii, 2015; Rast and Drugowitsch, 2020; Wainwright et al., 2001; Westrick et al., 2016). For example, Westrick et al. (2016) argue that empirical observations of V1 neural populations (Benucci et al., 2013)

can be explained by adapting normalization weights (parameterized by all-to-all synaptic connections) to different stimulus statistical contexts. The major downside of this approach is that changes in synaptic weights require $O(N^2)$ adaptation parameters, for a population of size N. Here, we examine the effects of classical single-neuron adaptive gain modulation on responses of a recurrently-connected population, and demonstrate that these are sufficient to explain adaptation phenomena, while requiring only $O(N)$ adaptation parameters. Holding the synaptic weights fixed prevents overfitting, and allows the network to remain stable across input contexts. Network stability is also relevant for contemporary machine learning applications that rely on adaptive adjustments to changing input statistics (e.g. Ballé et al., 2020; Hu et al., 2022; Mohan et al., 2021).

The adaptation model most similar to ours, developed by Gutierrez and Denève (2019), proposes an adaptive recurrent spiking neural network whose dynamics are derived from an efficient coding objective. Our model is complementary to this, but is simpler and more tractable, providing an analytic solution for population steady-state responses that facilitates comparisons to experimental data. Finally, recent work (published while this manuscript was being written) uses gain control as a normative population adaptation mechanism, but with the central goal of statistically whitening neural responses, while ignoring the means of responses (i.e. redundancy reduction via decorrelation and variance equalization; Duong et al., 2023c). Here, we demonstrate that our model captures adaptive effects involving mean responses as well as population response redundancy reduction, but that its steady-state responses are not whitened. We show that these deviations from whitening are similar to those seen in the neural recordings analyzed here.

Recurrent circuitry in sensory cortex. It is well known that recurrent excitation dominates cortical circuits (Douglas and Martin, 2007). In early sensory areas, a series of optogenetic inactivation experiments showed that recurrent excitation in cortex serves to progressively am-

plify thalamic inputs (Lien and Scanziani, 2013; Reinhold et al., 2015). In the context of sensory adaptation, King et al. (2016) performed silencing experiments in mice to show that the majority of adaptation effects seen in V1 arise from *local* activity-dependent processes, rather than being inherited from depressed thalamic responses upstream. Similarly, in monkey V1 neurophysiological recordings, Westerberg et al. (2019) used current source density analyses to show that stimulus-driven adaptation is primarily due to recurrent intracortical effects rather than feedforward effects. We leverage these functional observations, along with anatomical measurements of intracortical synaptic connectivity (Ko et al., 2011; Lee et al., 2016; Rossi et al., 2020) to inform the recurrent architecture used in our study.

4.4 An Analytically Tractable RNN with Gain Modulation

4.4.1 Adaptive gain modulation in a population without recurrence

We first consider the steady-state response of N neurons, $\mathbf{r}_\mathrm{f} \in \mathbb{R}^N$, receiving sensory stimulus inputs $\mathbf{s} \in \mathbb{R}^M$, with feedforward drive, $\mathbf{f}(\mathbf{s}) = [f_1(\mathbf{s}), f_2(\mathbf{s}), \ldots, f_N(\mathbf{s})]^\top$, which are each multiplicatively scaled by gains, $\mathbf{g} = [g_1, g_2, \ldots, g_N]^\top$:

$$\mathbf{r}_\mathrm{f}(\mathbf{s}, \mathbf{g}) = \mathbf{g} \circ \mathbf{f}(\mathbf{s}). \tag{4.1}$$

The gains, $\mathbf{g}$, have the effect of adjusting the amplitudes of responses $\mathbf{f}(\mathbf{s})$, and therefore the dynamic range of each neuron. As we demonstrate in Section 4.7, these simple multiplicative gain scalings are incapable of shifting the peaks of tuning curves, as seen in physiological data (Movshon and Lennie, 1979; Muller et al., 1999; Saul and Cynader, 1989). Previous approaches modeling neural population adaptation in cortex modify the structure of $\mathbf{f}(\mathbf{s})$ in response to changes in input statistics (e.g. Wainwright et al., 2001; Westrick et al., 2016). Here, we propose a fundamentally different approach, requiring *no* changes in synaptic weighting between

neurons.

4.4.2 Gain modulation in a recurrent neural population

We show that by incorporating single neuron gain modulation into a recurrent network, adaptive effects in each neuron propagate laterally to affect other cells in the population. Consider a model of N *recurrently* connected neurons with fixed feedforward and recurrent weights (Fig. 4.1A), presumed to have been learned over timescales much longer than the adaptive timescales examined in this study. We assume that the population of neural responses $\mathbf{r} \in \mathbb{R}^N$, driven by input stimuli $\mathbf{s} \in \mathbb{R}^M$ presented with probability $p(\mathbf{s})$, are governed by linear dynamics:

$$\frac{d\mathbf{r}(\mathbf{s}, \mathbf{g})}{dt} = -\mathbf{r} + \mathbf{g} \circ \mathbf{f}(\mathbf{s}) + \mathbf{W}\mathbf{r}, \tag{4.2}$$

where $\mathbf{W} \in \mathbb{R}^{N \times N}$ is a matrix of recurrent synaptic connection weights; and neuronal gains, $\mathbf{g} \in \mathbb{R}^N$, are adaptively optimized to a given $p(\mathbf{s})$. Both the feedforward functions $f_i(\mathbf{s})$ and recurrent weights $\mathbf{W}$ are assumed to be fixed despite varying stimulus contexts (i.e. *non-adaptive*). For notational convenience, we omit explicit time-dependence of the responses and stimuli (i.e. $\mathbf{r}(\mathbf{s}, \mathbf{g}, t), \mathbf{s}(t)$).

Empirical studies typically consider neural activity at steady-state before and after adapting to changes in stimulus statistics (Clifford et al., 2007). We therefore analyze the responses of our network at steady-state, $\mathbf{r}_*(\mathbf{s}, \mathbf{g})$, to facilitate comparison with data. The network dynamics of Equation 4.2 are linear in $\mathbf{r}$, and computing its steady-state is analytically tractable. Setting Eq. 4.2 to zero and isolating $\mathbf{r}$ (with the mild assumptions on invertibility; see Appendix C.1), yields the steady-state solution,

$$\mathbf{r}_*(\mathbf{s}, \mathbf{g}) = [\mathbf{I} - \mathbf{W}]^{-1} (\mathbf{g} \circ \mathbf{f}(\mathbf{s})) . \tag{4.3}$$

We can interpret these equilibrium responses as a modification of the gain-modulated feedforward drive, $\mathbf{g} \circ \mathbf{f}(\mathbf{s})$, which is propagated to other cells in the network via recurrent interactions, $[\mathbf{I} - \mathbf{W}]^{-1}$. When $\mathbf{W}$ is the zeros matrix (i.e. no recurrence), Equation 4.3 reduces to Equation 4.1, and adjusting neuronal gains simply rescales the feedforward responses without affecting the shape of response curves. The presence of the recurrent weight matrix $\mathbf{W}$ allows changes in neuronal gains to alter the effective tuning of other neurons in the network *without* changes to any synaptic weights.

4.4.3 Structure of recurrent connectivity matrix $\mathbf{W}$

Importantly, in our recurrent network, there are no explicit excitatory and inhibitory neurons – the recurrent activity term (last term in Eq. 4.2) represents the *net* lateral input to a neuron (i.e. the combination of both excitatory and inhibitory inputs). In addition, model simulations in this study use a $\mathbf{W}$ that is translation invariant (i.e. convolutional) in preferred orientation space, with strong net recurrent excitation near the preferred orientation of the cell, and relatively weak net excitation far away. This structure is motivated by functional and anatomical measurements in V1, indicating that orientation-tuned cells receive excitatory and inhibitory presynaptic inputs from cells tuned to every orientation, with disproportionate excitatory bias from similarly-tuned neurons (Lee et al., 2016; Rossi et al., 2020; Rubin et al., 2015). We elaborate on specific choices of $\mathbf{W}$ in Appendix C.1.

4.5 A Novel Objective for Adaptive Efficient Coding via Gain Modulation

Theories of efficient coding postulate that sensory neurons optimally encode the statistics of the natural environment (Barlow, 1961; Laughlin, 1981), subject to constraints on finite metabolic

resources (e.g. energy expenditure from firing spikes; Ganguli and Simoncelli, 2014; Olshausen and Field, 1996). However, sensory input statistics vary with context, and the means by which a neural population might confer an *adaptive and dynamic* efficient code remains an open question (Barlow and Foldiak, 1989; Duong et al., 2023c; Gutierrez and Denève, 2019; Młynarski and Hermundstad, 2021). How should our network (Equation 4.3) adaptively modulate its gains, $\mathbf{g}$, according to the statistics of a novel stimulus ensemble? We assume an initial stimulus ensemble, with probability density $p_0(\mathbf{s})$ (Fig. 4.1B), with a corresponding set of optimal gains, $\mathbf{g}_0$, toward which adaptive gains are homeostatically driven; and an optimal linear decoder, $\mathbf{D} \in \mathbb{R}^{N \times M}$. $\mathbf{D}$ is fixed and set to the pseudoinverse of $\mathbf{r}_*(\mathbf{s}, \mathbf{g})$ under the initial stimulus ensemble (see Appendix C.3).

Given a novel stimulus ensemble with probability density $p(\mathbf{s})$, we propose an adaptive efficient coding objective that neurons minimize by adjusting their gains,

$$\mathcal{L}(\mathbf{g}, p(\mathbf{s})) = \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})} \left\{ \|\mathbf{s} - \mathbf{D}^\top \mathbf{r}_*(\mathbf{s}, \mathbf{g})\|_2^2 + \alpha \|\mathbf{r}_*(\mathbf{s}, \mathbf{g})\|_2^2 \right\} + \gamma \| \mathbf{g} - \mathbf{g}_0 \|_2^2, \tag{4.4}$$

where α and γ are scalar hyperparameters. Intuitively, as the stimulus ensemble changes $p_0(\mathbf{s}) \rightarrow p(\mathbf{s})$, the gains $\mathbf{g}$ are adaptively adjusted to maximize the fidelity of the representation (first term), while minimizing overall activity in the network (second term), and minimally deviating from the initial gain state (third term). The gain homeostasis term serves to prevent catastrophic forgetting in the network under different stimulus contexts (Kirkpatrick et al., 2017): minimizing the gains' deviation from their optimal state under $p_0(\mathbf{s})$ allows the system to stably maintain reasonable performance on previously presented data and prevents the system from radically reorganizing itself on a fast time scale. In Appendix C.2, we show that adapting to $p(\mathbf{s})$ with gain homeostasis allows the network to maintain improved stimulus representation error under the $p_0(\mathbf{s})$ ensemble relative to a network optimized without gain homeostasis. We also perform ablations to show that the three terms in the objective are *jointly* necessary to produce the adaptation effects observed

in data.

4.5.1 Objective optimization

The objective given in Equation 4.4 is bi-convex in $\mathbf{g}$ and $\mathbf{D}$, and we can *analytically* solve for either variable independently or in alternation (i.e., coordinate descent via alternating least squares). See Appendix C.3 for the complete derivation. We initialize the network under the uniform stimulus density $p_0(\mathbf{s})$ to obtain a homeostatic gain target, $\mathbf{g}_0$, and a fixed decoder, $\mathbf{D}$.

4.6 V1 Neural Population Adaptation Data Reanalysis

In the following section, we compare our simulated adaptation model responses to reanalyzed neural population recordings from cat primary visual cortex (data obtained with permission from Benucci et al., 2013). Here, we provide an overview of our data analysis procedure which we also apply to our simulated model responses. Some of our analysis plots are new and are not in the original study[1]. For details on the recordings and preprocessing, we refer the reader to the original paper.

In the experiment, oriented stimuli were briefly presented randomly in rapid succession, with presentation probability determined by one of two contextual distributions: a uniform distribution $p_0(\mathbf{s})$, or a *biased* distribution, in which one orientation was presented significantly more frequently than the others, $p(\mathbf{s})$ (Figure 4.1B, top row). Figure 4.1B (middle row) shows responses for $N = 13$ units, aggregated over 11 recording sessions. For N units and K distinct stimuli, the authors fit orientation tuning curves to neural responses to produce matrices of orientation tuning curves, $\mathbf{R} \in \mathbb{R}^{N \times K}$ for each of the uniform and biased stimulus ensembles.

We normalize each unit's response curves under both contexts according to its minimum and maximum response during the $p_0(\mathbf{s})$ context, such that all responses lie in the interval $[0, 1]$ for

[1] Additionally, our plots are derived from steady-state fitted response curves, whereas the original publication used temporal information.

$p_0(\mathbf{s})$. That is, zero is the minimum stimulus-evoked response under the uniform ensemble, and one is the maximum. For responses to the biased ensemble, $p(\mathbf{s})$, a minimum response less than 0 indicates that the evoked response after adaptation has decreased relative to the uniform ensemble; similarly, a maximum response less than 1 indicates the response maximum after adaptation has decreased relative to that of the uniform ensemble (Figure 4.1B).

We compute response means, $\boldsymbol{\mu} \in \mathbb{R}^N$, and signal (as opposed to noise) covariance matrices, $\Sigma \in \mathbb{S}_+^N$,

$$\boldsymbol{\mu} = \mathbb{E}[\mathbf{R}], \qquad \Sigma = \mathbb{E}[\mathbf{R}\mathbf{R}^\top] - \boldsymbol{\mu}\boldsymbol{\mu}^\top, \tag{4.5}$$

where the expectation is over $p_0(\mathbf{s})$ or $p(\mathbf{s})$. To facilitate comparisons between response covariances under the uniform and biased stimulus ensembles, we scale response covariance matrices by the variances of the neurons under the uniform stimulus probability condition, $\boldsymbol{\sigma}_0^2 \in \mathbb{R}_+^N$,

$$\hat{\Sigma} = \operatorname{diag}(\boldsymbol{\sigma}_0)^{-1} \Sigma \operatorname{diag}(\boldsymbol{\sigma}_0)^{-1}. \tag{4.6}$$

As in the original work, we symmetrize recorded responses before computing changes in response amplitude, maxima, and minima; and we anti-symmetrized the data to compute changes in preferred tuning. Amplitudes before and after adaptation are computed by taking the ratio of the peak-to-trough heights of the response curves. Finally, we compute shifts in minimum response relative to the response curves under the uniform ensemble condition. Because the stimuli $\mathbf{s} \sim p(\mathbf{s})$ under consideration are oriented gratings, we compute non-parametric circular statistics to characterize response changes with adaptation.

4.7 Numerical Simulations and Comparisons to Neural Data

We compare numerical simulations of our normative adaptation model with reanalyzed cat V1 population recording data (Benucci et al., 2013).

4.7.1 Model and simulation parameters

For all simulation results and figures in this study, we consider a network comprised of $N = 255$ recurrently connected neurons, with $K = M = 511$ orientation stimuli as inputs. The neuronal gains, $\mathbf{g}$, adapt to changes in stimulus ensemble statistics ($p_0(\mathbf{s}) \rightarrow p(\mathbf{s})$), while the feedforward synaptic weights, $\mathbf{f}(\mathbf{s})$, and recurrent synaptic weights, $\mathbf{W}$, remain fixed. We set the homeostatic target gains, $\mathbf{g}_0$, to the optimal values of $\mathbf{g}$ under the uniform probability stimulus ensemble, $p_0(\mathbf{s})$. Feedforward orientation-tuning functions, $\mathbf{f}(\mathbf{s})$, are evenly distributed in the stimulus domain, and are broadly-tuned Gaussians with full-width half-max (FWHM) of 30° (Benucci et al., 2013). The recurrent weight matrix, $\mathbf{W}$, is a Gaussian with 10° FWHM, summed with a weaker, broad, untuned excitatory component (see Appendix C.1).

To determine appropriate values of α and γ in the objective (Equation 4.4), we performed a grid search hyperparameter sweep, minimizing the deviation between model and experimentally measured tuning curves for the biased stimulus ensemble. The figures here all use model responses from a simulation using α = 1E-3, γ = 1E-2. We find that qualitative effects are insensitive to small changes in these parameters. The key finding from this parameter sweep is that the gain homeostasis penalty weight must be sufficiently greater than the activity penalty weight (i.e. $\gamma > \alpha$). After initializing the network gains to the statistics of $p_0(\mathbf{s})$, we adapt the gains to $p(\mathbf{s})$ by optimizing Equation 4.4, then compare our model predicted responses to cat V1 population recordings Figure 4.1B.

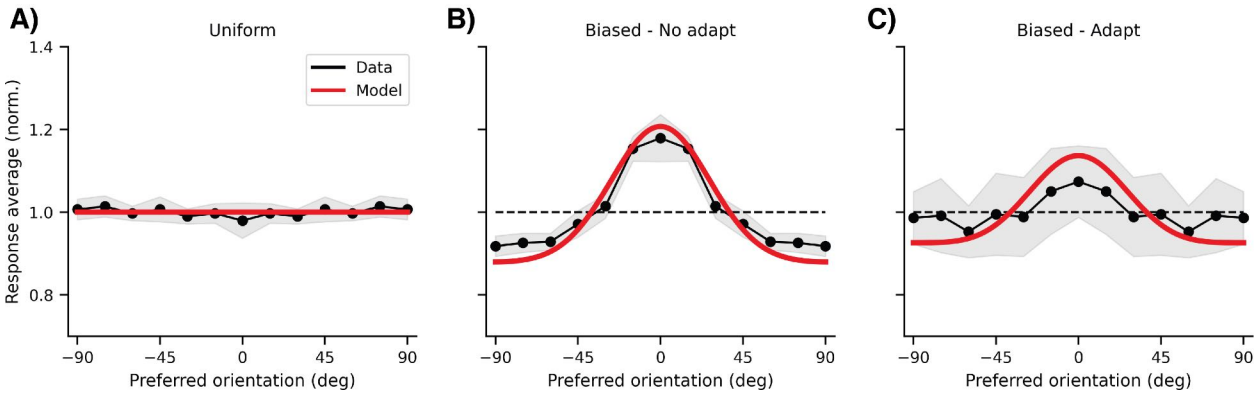

Figure 4.2: Adaptive response equalization. Each dot is the average response of a neuron. **A)** Response averages under the uniform stimulus ensemble condition. **B)** *Without* adaptation, response averages under the biased stimulus ensemble show substantial deviation from equalization (which corresponds to the dashed black line). **C)** After adaptation, response averages to the biased ensemble are nearly equalized. Shaded regions are SEM.

4.7.2 Adaptive gain modulation predicts response equalization

Population response equalization is an adaptive mechanism first proposed in psychophysics (Anstis et al., 1998). The authors argued that adaptation should serve as a "graphic equalizer" in response to alterations in environmental statistics. Others have have described equalization as a mechanism that centers a population response by subtracting the responses to the prevailing stimulus ensemble (Clifford et al., 2000), to rescale responses such that the average of a measured signal remains constant (Ullman and Schechtman, 1982). Figure 4.2 shows how our model recapitulates mean firing rates across all stimuli under the uniform and biased ensembles without adaptation, along with adaptive population response equalization under the biased ensemble. Figure 4.2A shows how the average response of each pre-adapted neuron under the uniform ensemble is equal. By contrast, Figure 4.2B demonstrates that our model predicts how the pre-adaptation tuning curves under the biased stimulus ensemble would produce a substantial deviation from equalization. Finally, adaptively optimizing neuron gains via Equation 4.4 predicts the compensatory response equalization under the biased stimulus ensemble observed in data (panel C).

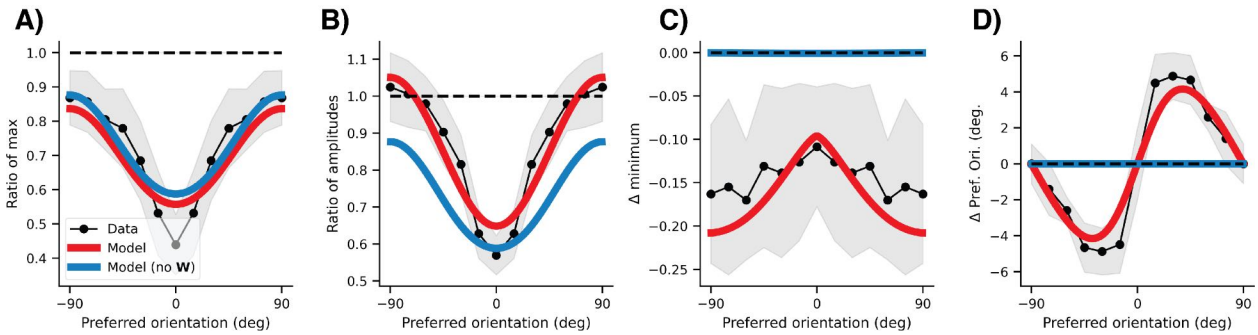

Figure 4.3: Recurrent network model with adaptive gain modulation (red) captures the full set of post-adaptation first-order response changes observed in data (black points), while a network without recurrence (blue) does not. **A)** Ratios of after/before adaptation response maxima. **B)** Ratios of response amplitudes (|| max − min || response) after/before adaptation. **C)** Changes in average minimum evoked response. **D)** Shifts in tuning away from the adapter. Shaded regions are SEM. Dashed lines indicate predictions for a non-adaptive model.

4.7.3 Adaptive gain modulation predicts nuanced changes in first-order statistics of responses

Figure 4.3 summarizes adaptive changes in neural responses by comparing tuning curve responses under the biased stimulus ensemble compared to responses under the uniform stimulus ensemble (i.e. right vs. left columns of Fig. 4.1B). Our gain-modulating efficient coding model can capture this entire array of observed adaptation effects.

Changes in response maxima, amplitudes, and minima. Stimulus-dependent response reductions are a ubiquitous finding in adaptation experiments (Weber et al., 2019). Figure 4.3A,B show changes in response maxima, and response amplitudes (peak-to-trough height) following adaptation to the biased stimulus ensemble. Ratios less than 1 indicate a reduction in maxima or amplitudes following adaptation. Under the biased stimulus ensemble, the model optimizes its gains according to the objective (Eq. 4.4) to preferentially reduce activity near the over-presented adapter stimulus. By optimizing gains to the adaptive efficient coding objective, our linear model undershoots the magnitude of change around the adapter (Fig. 4.3A,B red curve near 0°), but captures the overall effect of adaptive amplitude and maxima reduction.

Figure 4.3C shows that adaptation induces a tuning-dependent, global reduction in minimum stimulus-evoked response across the population. These minima typically occur at the anti-preferred orientation for each neuron (Fig. 4.1B). Previous work has attributed this to an untuned reduction in thalamic inputs, or a drop in base firing (Benucci et al., 2013; Westrick et al., 2016). Our model proposes a different mechanism: gain reductions in neurons tuned for the adapter propagate laterally through the network, and result in commensurate reductions in the broad-/untuned recurrent excitation to other neurons in the population. This ultimately leads to a reduction in minimum evoked response across the entire population (Fig. 4.3C); importantly, the model also captures the qualitative shape of the change. Our mechanistic prediction that this effect arises due to recurrent contributions is in concordance with the broad literature on recurrent cortical circuitry, its role in amplification (Reinhold et al., 2015), and in sensory adaptation (Hershenhoren et al., 2014; King et al., 2016).

Shifts in tuning preference. Tuning curve shifts following adaptation have been reported across many visual and auditory adaptation studies (Clifford et al., 2007; Whitmire and Stanley, 2016, for reviews). Figure 4.3D quantifies changes in neuron preferred orientation (i.e. the orientation at which response maximum occurs) after adapting to the biased stimulus ensemble. The sinusoidal shape of the curve indicates that adapted tuning curves are *repelled* from the over-presented adapter stimulus. This rearrangement of tuning curve density is consistent with efficient coding studies that argue that a sensory neural population should optimally allocate its finite resources toward encoding information about the current stimulus ensemble (Ganguli and Simoncelli, 2014; Gutnisky and Dragoi, 2008). Here, we show that these effects can mechanistically be explained by optimizing neuronal gains to maintain a high fidelity representation of the stimulus under the biased ensemble.

Objective and network ablations. In Appendix C.2, we assess the importance of each term of the adaptation objective (Eq. 4.4) by ablating them from the objective and re-simulating the

network adapting to the biased stimulus ensemble. We show that each of the three terms is jointly necessary to capture the adaptation effects shown here. In terms of network architectural ablations, the blue curves in Figure 4.3 demonstrate how removing recurrence (i.e. $\mathbf{W} = 0$; Equation 4.1) impacts adaptive changes in neural responses. While this single stage feedforward model can reproduce reductions in response maxima Fig. 4.3A, it is incapable of producing the appropriate change in response amplitudes (Fig. 4.3B), and completely fails at producing adaptive reductions in minimum response, or shifts in tuning preference (Fig. 4.3C,D). Intuitively, this is because the gains in this reduced model serve to set the amplitude of the output, and cannot alter the qualitative shape of the tuning curve without propagating through the recurrent circuitry. The structure of $\mathbf{W}$ used in our model is informed by functional and anatomical studies in cortical circuits (Lee et al., 2016; Rossi et al., 2020), comprising strong net excitation from similarly-tuned neurons and untuned weak net excitation from dissimilarly-tuned neurons. In Appendix C.1, we study the impact of $\mathbf{W}$'s structure on model adapted responses. The structure of $\mathbf{W}$ can be quite flexible while still producing the effects shown here, so long as recurrent input includes weak net excitation from dissimilarly-tuned neurons.

4.7.4 Adaptive gain modulation predicts homeostasis in second-order statistics of responses

The principle of redundancy reduction is core to the efficient coding hypothesis (Barlow, 1961), and evidence supporting *adaptive* redundancy reduction has been reported across multiple brain regions and modalities (Atick and Redlich, 1992; Muller et al., 1999; Wanner and Friedrich, 2020). In the task modeled in our study, over-presenting the adapter stimulus can be viewed as increasing redundancy in the stimulus ensemble (Figure 4.1B, top). This manifests as a "hot spot" in the center of $\hat{\Sigma}$ if the neural responses were to remain unadapted to $p(\mathbf{s})$ (Fig. 4.4A, middle column). However, when the model adapts its gains according to the objective (Eq. 4.4),

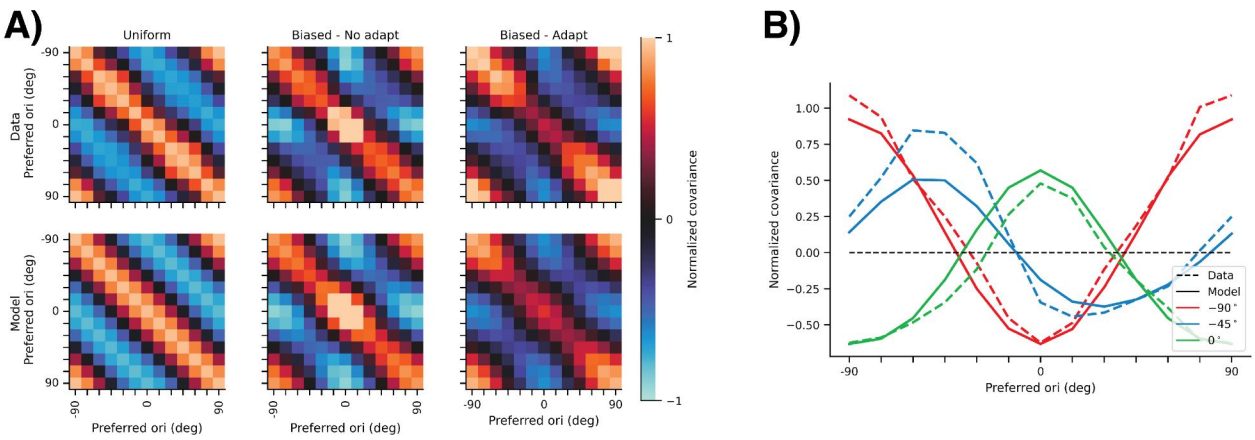

Figure 4.4: Population response redundancy reduction and signal covariance homeostasis. **A)** Scaled response covariance matrices, $\hat{\Sigma}$ (Eq. 4.6), for V1 data (top row) and model simulations (bottom row), for unadapted tuning curves and uniform stimulus ensemble (left column), unadapted tuning curves and biased stimulus ensemble (middle column), and adapted tuning curves and biased stimulus ensemble (right column). **B)** Three example horizontal slices of the data (dashed) and model (solid) $\hat{\Sigma}$ from **A**, at 0, -45, and -90 degrees orientation (colors).

the covariance near the adapter stimulus is reduced, and the predicted signal covariance is well matched to data (Fig. 4.4A, right column, Fig. 4.4B).

A signal covariance matrix devoid of redundancy would be one that is statistically white (i.e. the identity matrix). However, under both the uniform and biased stimulus ensemble conditions (Figure 4.4A, top left and right), we note that the experimentally observed signal covariance matrix *is not* statistically white[2]. Thus, previous normative approaches to population adaptation that explicitly whiten neural responses may not be suitable models for this data (e.g. Młynarski and Hermundstad, 2021; Pehlevan and Chklovskii, 2015). By contrast, our adaptation objective, which emphasizes stimulus signal fidelity subject to metabolic and homeostatic constraints predicts an adapted signal covariance matrix whose deviations from the identity matrix are similar to those observed in data. Notably, this effect naturally emerges from our model *without* additional parameter-tuning.

[2]In their study, Benucci et al. (2013, Fig. 3) replaced negative entries of $\hat{\Sigma}$ with zeros.

4.8 Discussion

Study limitations. The network considered here is a rate model whose tractable linear dynamics allow us to examine adaptation responses at steady-state. Response dynamics during adaptation are rich (Dragoi et al., 2000; Patterson et al., 2013; Quiroga et al., 2016), and are relatively understudied. Developing our model and objective into a biologically plausible online network with explicit excitatory and inhibitory neurons, while adapting gains according to only local signals (Duong et al., 2023c; Gutierrez and Denève, 2019) is an interesting direction worth pursuing. Furthermore, because we model trial-averaged experimental data in this study, our model does not account for stochasticity in neural responses. Thus, our model cannot explain adaptive changes in trial-to-trial variability (Gutnisky and Dragoi, 2008). Finally, there exist adaptive changes to simultaneously-presented stimuli, usually explained via divisive normalization (Aschner et al., 2018; Solomon and Kohn, 2014; Yiltiz et al., 2020), which is not included in our model (see Appendix C.4). One possible way to bridge this gap would be to combine our normative approach with recently-proposed recurrent models of normalization (Heeger and Mackey, 2018; Heeger and Zemlianova, 2020).

Alternative network architectures. There are alternative, equivalent formulations of our model that may give rise to the same steady-state responses as Eq. 4.3, which we illustrate in Appendix C.4. Firstly, our model is equivalent to a two-stage feedforward network with gain modulation preceding the inputs of the second stage. Since orientation tuning arises in V1, these two stages could be two different layers within V1; the core mechanism of our framework can thus be related to studies describing adaptive gain changes being inherited from one group of neurons to the next (Dhruv and Carandini, 2014; Kohn and Movshon, 2003; Stocker and Simoncelli, 2009). Secondly, gain modulation in our model, which serves to multiplicatively scale input drive, $\mathbf{f}(\mathbf{s})$, can equivalently be interpreted as multiplicatively *attenuating* the recurrent drive of

the network. In this sense, our model resembles that of Heeger and Zemlianova (2020), in which divisive normalization is mediated by gating recurrent amplification.

Experimental predictions. We propose that rapid neural population adaptation in cortex can be mediated by single neuron adaptive gain modulation. Validating this hypothesis would require careful experimental measurements of neurons during adaptation. First, our framework predicts that between-neuron synaptic connectivity (i.e. $\mathbf{W}$) remains stable through adaptation. Second, our normative objective suggests that gain homeostasis plays a central role in population adaptation (see Appendix C.2). Evidence for stimulus-dependent gain control such as this can possibly be found by measuring neuron membrane conductance during adaptation, mediated by changes in slow hyperpolarizing Ca^{2+}- and Na^{+}-induced K^{+} currents (Sanchez-Vives et al., 2000). Lastly, while there has been considerable progress in mapping the circuits involved in sensory adaptation (Wanner and Friedrich, 2020), determining the exact structure of functional recurrent connectivity remains an open problem. Indeed, we show how different (but not all) forms of $\mathbf{W}$ can give rise to the same qualitative results shown here (Appendix C.1). Performing adaptation experiments with richer sets of stimulus ensembles, $p(\mathbf{s})$, can provide better constraints for solving this functional inverse problem.

4.8.1 Conclusion

We demonstrate that adaptation effects observed in cortex – changes in response maxima and minima, tuning curve repulsion, and stimulus-dependent response decorrelation – can be explained as arising from the recurrent propagation of single neuron gain adjustments aimed at coding efficiency. This adaptation mechanism is general, and can be applied to modalities other than vision. For example, studies of neural adaptation in auditory cortex have shown that adaptive responses such tuning curve shifts cannot be explained by feedforward mechanisms, and likely arise from adaptive changes to intracortical recurrent interactions (Hershenhoren et al.,

2014; Lohse et al., 2020). Previous population adaptation models rely on changes in all-to-all synaptic weights to explain these phenomena (e.g. Westrick et al., 2016), but our results suggest that single neuron gain modulations may provide a more plausible mechanism which uses $O(N)$ instead of $O(N^2)$ adaptive parameters. Adaptation in cortex happens on the order of hundreds of milliseconds, and is just as quickly *reversible* (Muller et al., 1999); a network whose synaptic weights were constantly remapping would be undesirable due to a lack of stability, while a mechanism such as adaptive single neuron gain modulation can be local, fast, and reversible (Ferguson and Cardin, 2020). Taken together, our study offers a simple mechanistic explanation for observed adaptation effects at the level of a neural population, and expands upon well-established concepts of adaptive coding efficiency with single neuron gain control.

5 Representational Dissimilarity Metric Spaces for Stochastic Neural Networks

5.1 Overview

In this chapter, we shift our focus away from adaptive coding efficiency in neural populations to introduce a new statistical tool for comparing representational geometry, and aligning *stochastic* neural population responses. An earlier version of this work appeared in Computational and Systems Neuroscience 2023. These findings presented in this chapter published in the Proceedings of the Eleventh International Conference on Learning Representations (Duong et al., 2023e).

Quantifying similarity between neural representations—e.g. hidden layer activation vectors—is a perennial problem in deep learning and neuroscience research. Existing methods compare deterministic responses (e.g. artificial networks that lack stochastic layers) or averaged responses (e.g., trial-averaged firing rates in biological data). However, these measures of *deterministic* representational similarity ignore the scale and geometric structure of noise, both of which play important roles in neural computation. To rectify this, we generalize previously proposed shape metrics (Williams et al., 2021) to quantify differences in *stochastic* representations. These new

distances satisfy the triangle inequality, and thus can be used as a rigorous basis for many supervised and unsupervised analyses. We show that this approach is practical for large-scale data and provides insights that cannot be measured with existing metrics. Leveraging this novel framework, we find that the stochastic geometries of neurobiological representations of oriented visual gratings and naturalistic scenes respectively resemble untrained and trained deep network representations. Further, we are able to more accurately predict certain network attributes (e.g. training hyperparameters) from its position in stochastic (versus deterministic) shape space.

5.2 Introduction

Comparing high-dimensional neural responses—neurobiological firing rates or hidden layer activations in artificial networks—is a fundamental problem in neuroscience and machine learning (Chung and Abbott, 2021; Dwivedi and Roig, 2019). There are now many methods for quantifying representational dissimilarity including canonical correlations analysis (CCA; Raghu et al., 2017), centered kernel alignment (CKA; Kornblith et al., 2019), representational similarity analysis (RSA; Kriegeskorte et al., 2008a), shape metrics (Williams et al., 2021), and Riemannian distance (Shahbazi et al., 2021) . Intuitively, these measures quantify similarity in the geometry of neural responses while removing expected forms of invariance, such as permutations over arbitrary neuron labels.

However, these methods only compare *deterministic representations*—i.e. networks that can be represented as a function $f : \mathcal{Z} \mapsto \mathbb{R}^n$, where n denotes the number of neurons and $\mathcal{Z}$ denotes the space of network inputs. For example, each $z \in \mathcal{Z}$ could correspond to an image, and $f(z)$ is the response evoked by this image across a population of n neurons (Fig. 5.1A). Biological networks are essentially never deterministic in this fashion. In fact, the variance of a stimulus-evoked neural response is often larger than its mean (Goris et al., 2014). Stochastic responses also arise in the deep learning literature in many contexts, such as in deep generative modeling (Kingma and

Welling, 2019), Bayesian neural networks (Wilson, 2020), or to provide regularization (Srivastava et al., 2014).

Stochastic networks may be conceptualized as functions mapping each $z \in \mathcal{Z}$ to a probability distribution, $F(\cdot \mid z)$, over $\mathbb{R}^n$ (Fig. 5.1B, Kriegeskorte and Wei 2021). Although it is easier to study the representational geometry of the average response, it is well understood that this provides an incomplete and potentially misleading picture (Kriegeskorte and Douglas, 2019). For instance, the ability to discriminate between two inputs $z, z' \in \mathcal{Z}$ depends on the overlap of $F(z)$ and $F(z')$, and not simply the separation of their means (Fig. 5.1C-D). A rich literature in neuroscience has built on top of this insight (Abbott and Dayan, 1999; Rumyantsev et al., 2020; Shadlen et al., 1996). However, to our knowledge, no studies have compared noise correlation structure across animal subjects or species, as has been done with trial-averaged responses. In machine learning, many studies have characterized the effects of noise on model predictions (An, 1996; Sietsma and Dow, 1991), but only a handful have begun to characterize the geometry of stochastic hidden layers (Dapello et al., 2021). Further, the recent surge of interest in neural network symmetries and permutation-invariant mode connectivity has been largely limited to the deterministic setting (Ainsworth et al., 2022; Bronstein et al., 2021; Tatro et al., 2020). Altogether, this points to a need for systematic frameworks for analyzing large ensembles of stochastic networks in terms of representational geometry.

To address these gaps, we formulate a novel class of *metric spaces* over stochastic neural representations. That is, given two stochastic networks F_i and F_j, we construct distance functions $d(F_i, F_j)$ that are symmetric, satisfy the triangle inequality, and are equal to zero if and only if F_i and F_j are equivalent according to a pre-defined criterion. In the deterministic limit—i.e., as F_i and F_j map onto Dirac delta functions—our approach converges to well-studied metrics over *shape spaces* (Dryden and Mardia, 1993; Srivastava and Klassen, 2016), which were proposed by Williams et al. (2021) to measure distances between deterministic networks. The triangle inequality is required to derive theoretical guarantees for many downstream analyses (e.g. nonparamet-

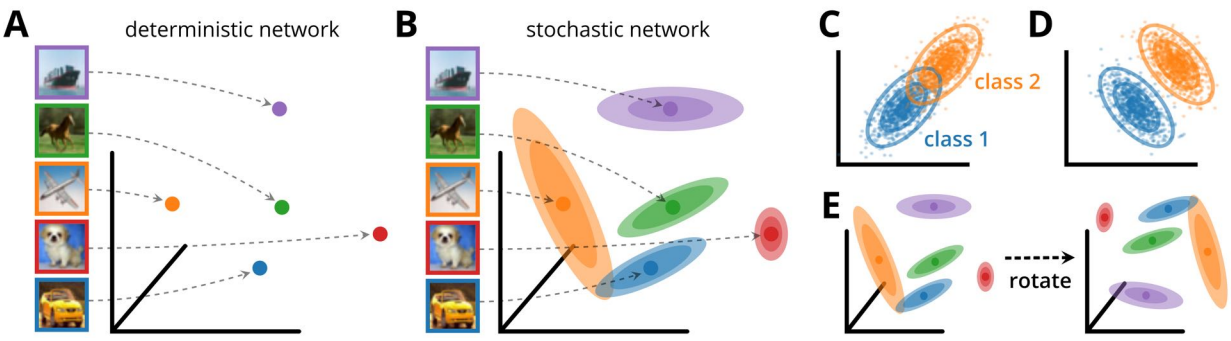

Figure 5.1: *(A)* Illustration of a deterministic network mapping inputs, (color-coded images) into points in $\mathbb{R}^n$. *(B)* Illustration of a stochastic network, where each input, $z \in \mathcal{Z}$, maps onto a distribution, $F(\cdot \mid z)$, over $\mathbb{R}^n$. *(C)* Example where noise correlations impair discriminability between two image classes. *(D)* Example where noise correlations improve discriminability (see Abbott and Dayan, 1999). *(E)* Illustration of two stochastic networks with equivalent representational geometry.

ric regression, Cover and Hart 1967, and clustering, Dasgupta and Long 2005). Thus, we lay an important foundation for analyzing stochastic representations, akin to results shown by Williams et al. (2021) in the deterministic case.

To demonstrate the utility of this framework, we apply stochastic shape metrics to neurobiological recordings of mouse visual cortex, and show that contributions of mean and covariance to stochastic representational geometry differ across simple, artificial stimulus sets (oriented gratings) and rich stimulus sets (natural scenes). Further, we analyze stochastic representations in artificial networks, and find that a network's position in stochastic shape space can be used to accurately predict its hyperparameters, including the random seed. Interestingly, these predictions are often more accurate if contributions from the average representational geometry are ignored, suggesting that the features of neural "noise" carries salient information about network structure and computations.

5.3 Methods

Deterministic Shape Metrics

We begin by reviewing how shape metrics quantify representational dissimilarity in the deterministic case. In the *Discussion* (section 5.5), we review other related prior work.

Let $\{f_1, \ldots, f_K\}$ denote K deterministic neural networks, each given by a function $f_k : \mathcal{Z} \mapsto \mathbb{R}^{n_k}$. Representational similarity between networks is typically defined with respect to a set of M inputs, $\{z_1, \ldots, z_M\} \in \mathcal{Z}^M$. We can collect the representations of each network into a matrix:

$$\boldsymbol{X}_k = \begin{bmatrix} — & f_k(z_1) & — \\ & \vdots & \\ — & f_k(z_M) & — \end{bmatrix}. \tag{5.1}$$

A naïve dissimilarity measure would be the Euclidean distance, $\|\boldsymbol{X}_i - \boldsymbol{X}_j\|_F$. This is nearly always useless. Since neurons are typically labelled in arbitrary order, our notion of distance should—at the very least—be invariant to permutations. Intuitively, we desire a notion of distance such that $d(\boldsymbol{X}_i, \boldsymbol{X}_j) = d(\boldsymbol{X}_i, \boldsymbol{X}_j\boldsymbol{\Pi})$ for any permutation matrix, $\boldsymbol{\Pi} \in \mathbb{R}^{n \times n}$. Linear CKA and RSA achieve this by computing the dissimilarity between $\boldsymbol{X}_i\boldsymbol{X}_i^\top$ and $\boldsymbol{X}_j\boldsymbol{X}_j^\top$ instead of the raw representations.

Generalized shape metrics are an alternative approach to quantifying representational dissimilarity. The idea is to compute the distance after minimizing over nuisance transformations (e.g. permutations or rotations in $\mathbb{R}^n$). Let $\phi_k : \mathbb{R}^{M \times n_k} \mapsto \mathbb{R}^{M \times n}$ be a fixed, "preprocessing function" for each network and let $\mathcal{G}$ be a set of nuisance transformations on $\mathbb{R}^n$. Williams et al. (2021) showed that:

$$d(\boldsymbol{X}_i, \boldsymbol{X}_j) = \min_{\boldsymbol{T} \in \mathcal{G}} \|\phi_i(\boldsymbol{X}_i) - \phi_j(\boldsymbol{X}_j)\boldsymbol{T}\|_F \tag{5.2}$$

is a *metric* over equivalent neural representations provided two technical conditions are met. The first is that $\mathcal{G}$ is a *group* of linear transformations. This means that: (a) the identity is in the set

of nuisance transformations ($\boldsymbol{I} \in \mathcal{G}$), (b) every nuisance transformation is invertible by another nuisance transformation (if $\boldsymbol{T} \in \mathcal{G}$ then $\boldsymbol{T}^{-1} \in \mathcal{G}$), and (c) nuisance transformations are closed under composition ($\boldsymbol{T}_1\boldsymbol{T}_2 \in \mathcal{G}$ if $\boldsymbol{T}_1 \in \mathcal{G}$ and $\boldsymbol{T}_2 \in \mathcal{G}$). The second condition is that every nuisance transformation is an *isometry*, meaning that $\|\boldsymbol{X}_i\boldsymbol{T} - \boldsymbol{X}_j\boldsymbol{T}\|_F = \|\boldsymbol{X}_i - \boldsymbol{X}_j\|_F$ for every $\boldsymbol{T} \in \mathcal{G}$. Several choices of $\mathcal{G}$ satisfy these conditions including the permutation group, $\mathcal{P}$, and the orthogonal group, $\mathcal{O}$, which respectively correspond to the set of all permutations and rotations on $\mathbb{R}^n$.

Equation 5.2 provides a recipe to construct many notions of distance. To enumerate some examples, we will assume for simplicity that $\phi_1 = \ldots = \phi_K = \phi$ and all networks have n neurons. Then, to obtain a metric that is invariant to translations and permutations, we can set $\phi(\boldsymbol{X}) = (1/n)(\boldsymbol{I} - \boldsymbol{1}\boldsymbol{1}^\top)\boldsymbol{X}$ and $\mathcal{G} = \mathcal{P}(n)$. If we instead set $\mathcal{G} = \mathcal{O}(n)$, we recover the well-known *Procrustes distance*, which is invariant to rotations. Finally, if we choose $\phi(\cdot)$ to whiten the covariance of $\boldsymbol{X}$, we obtain notions of distance that are invariant to linear transformations and are closely related to CCA. Williams et al. (2021) provides further discussion and examples.

An attractive property of Equation 5.2 is that it establishes a metric space over deterministic representations. That is, distances are symmetric $d(\boldsymbol{X}_i, \boldsymbol{X}_j) = d(\boldsymbol{X}_j, \boldsymbol{X}_i)$ and satisfy the triangle inequality $d(\boldsymbol{X}_i, \boldsymbol{X}_k) \leq d(\boldsymbol{X}_i, \boldsymbol{X}_j) + d(\boldsymbol{X}_j, \boldsymbol{X}_k)$. Further, the distance is zero if and only if there exists a $\boldsymbol{T} \in \mathcal{G}$ such that $\phi_i(\boldsymbol{X}_i) = \phi_j(\boldsymbol{X}_j)\boldsymbol{T}$. These fundamental properties are needed to rigorously establish many statistical analyses (Cover and Hart, 1967; Dasgupta and Long, 2005).

Stochastic Shape Metrics

Let $\{F_1, \ldots, F_K\}$ denote a collection of K stochastic networks. That is, each F_k is a function that maps each input $\boldsymbol{z} \in \mathcal{Z}$ to a conditional probability distribution $F_k(\cdot \mid \boldsymbol{z})$. How can Equation 5.2 be generalized to measure representational distances in this case? In particular, the minimization in Equation 5.2 is over a Euclidean "ground metric," and we would like to choose a compatible metric over probability distributions. Concretely, let $\mathcal{D}(P, Q)$ be a chosen "ground metric" between two distributions P and Q. Let $\delta_{\boldsymbol{x}}$ and $\delta_{\boldsymbol{y}}$ denote Dirac masses at $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ and consider the limit

that $P \to \delta_{\boldsymbol{x}}$ and $Q \to \delta_{\boldsymbol{y}}$. In this limit, we seek a ground metric for which $\mathcal{D}(\delta_{\boldsymbol{x}}, \delta_{\boldsymbol{y}})$ is related to $\|\boldsymbol{x} - \boldsymbol{y}\|$. Many probability metrics and divergences fail to meet this criterion. For example, if $\boldsymbol{x} \neq \boldsymbol{y}$, then the Kullback-Leibler (KL) divergence approaches infinity and the total variation distance and Hellinger distance approach a constant that does not depend on $\|\boldsymbol{x} - \boldsymbol{y}\|$.

In this work, we explored two ground metrics. First, the *p-Wasserstein distance* (Villani, 2009):

$$\mathcal{W}_p(P, Q) = (\inf \mathbb{E}\left[\|X - Y\|^p\right])^{1/p} \tag{5.3}$$

where $p \geq 1$, and the infimum is taken over all random variables (X, Y) whose marginal distributions coincide with P and Q. Second, the *energy distance* (Székely and Rizzo, 2013):

$$\mathcal{E}_q(P, Q) = (\mathbb{E}\left[\|X - Y\|^q\right] - \tfrac{1}{2}\mathbb{E}\left[\|X - X'\|^q\right] - \tfrac{1}{2}\mathbb{E}\left[\|Y - Y'\|^q\right])^{1/2} \tag{5.4}$$

where $0 < q < 2$ and $X, X' \overset{\text{i.i.d.}}{\sim} P$ and $Y, Y' \overset{\text{i.i.d.}}{\sim} Q$. As desired, we have $\mathcal{W}_p(\delta_{\boldsymbol{x}}, \delta_{\boldsymbol{y}}) = \|\boldsymbol{x} - \boldsymbol{y}\|$ for any p, and $\mathcal{E}_q(\delta_{\boldsymbol{x}}, \delta_{\boldsymbol{y}}) = \|\boldsymbol{x} - \boldsymbol{y}\|^{q/2}$. Thus, when $q = 1$ for example, the energy distance converges to the square root of Euclidean distance in the deterministic limit. Interestingly, when $q = 2$, the energy distance produces a deterministic metric on trial-averaged responses (see subsection D.6.1).

The Wasserstein and energy distances are intuitive generalizations of Euclidean distance. Both can be understood as being proportional to the amount of energy it costs to transport a pile of dirt (a probability density P) to a different configuration (the other density Q). Wasserstein distance is based on the cost of the optimal transport plan, while energy distance is based on the the cost of a random (i.e. maximum entropy) transport plan (see Fig. 5.2, and Feydy et al. 2019).

Our main proposition shows that these two ground metrics can be used to generalize Equation 5.2.

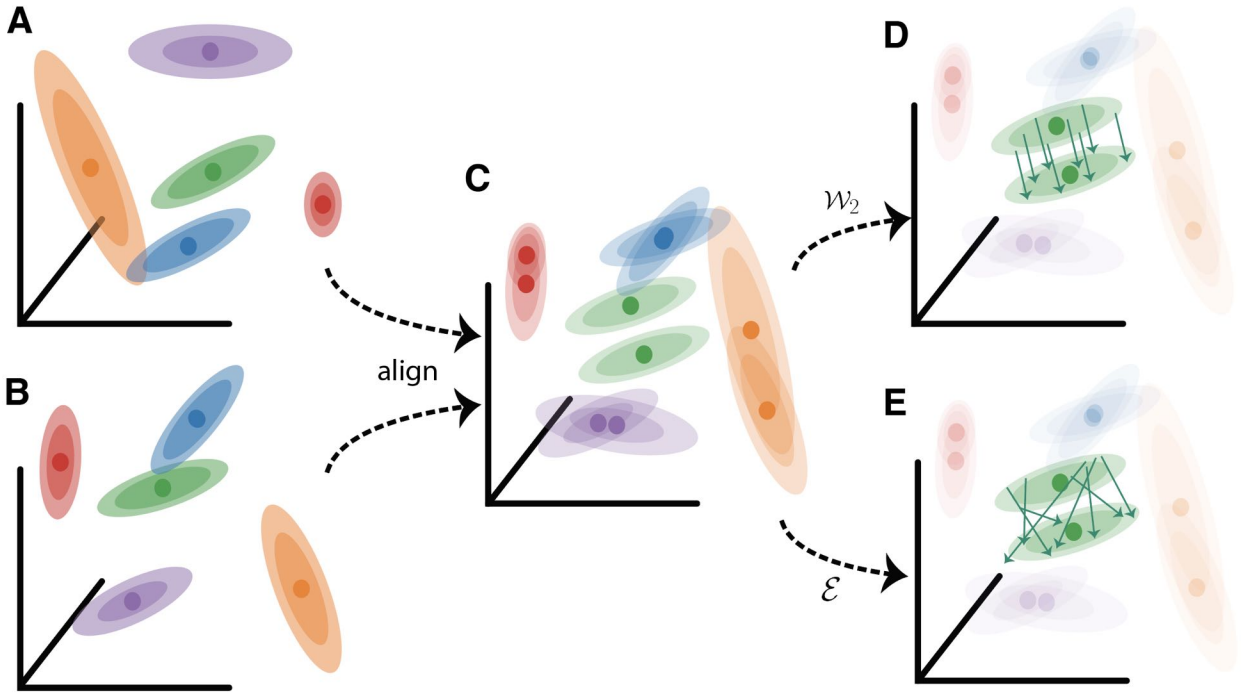

Figure 5.2: Proposed method intuition using distances based on either $\mathcal{W}_2$ or $\mathcal{E}$ ground metrics. *(A)* and *(B)* Two example stochastic network representations to five stimuli (colors). *(C)* The optimal alignment of the representations over nuisance transformations (e.g. rotations, $\mathcal{G} = O$). *(D)* Intuitively, the 2-Wasserstein distance ($\mathcal{W}_2$) is the minimum cost of turning one density (pile of dirt) to another (Peyré and Cuturi, 2019; Villani, 2009). Here we highlight the distance between the two green densities to reduce clutter. *(E)* Energy distance is based on the maximum-entropy transport map between the two distributions (Feydy et al., 2019).

Proposition 5.1 (Stochastic Shape Metrics)**.** *Let Q be a distribution on the input space, $\phi_1, \ldots, \phi_K$ be measurable functions mapping onto $\mathbb{R}^n$ and let $F_i^\phi = F_i \circ \phi_i^{-1}$. Let $\mathcal{D}^2$ denote the squared "ground metric," and let $\mathcal{G}$ be a group of isometries with respect to $\mathcal{D}$. Then,*

$$d(F_i, F_j) = \min_{T \in \mathcal{G}} \left(\mathop{\mathbb{E}}_{z \sim Q} \left[\mathcal{D}^2 \left(F_i^\phi(\cdot \mid z), F_j^\phi(\cdot \mid z) \circ \boldsymbol{T}^{-1} \right) \right] \right)^{1/2} \tag{5.5}$$

defines a metric over equivalence classes, where F_i is equivalent to F_j if and only if there is a $\boldsymbol{T} \in \mathcal{G}$ such that $F_i^\phi(\cdot \mid z)$ and $F_j^\phi(\cdot \mid z) \circ \boldsymbol{T}^{-1}$ are equal for all $z \in supp(Q)$.

Above, we use the notation $P \circ \phi^{-1}$ to denote the pushforward measure—i.e. the measure defined by the function composition, $P(\phi^{-1}(A))$ for a measurable set A, where P is a distribution and

ϕ is a measurable function. A proof is provided in section D.3. Intuitively, $\boldsymbol{T}$ plays the same role as in Equation 5.2, which is to remove nuisance transformations (e.g. rotations or permutations; see Fig. 5.1E). The functions $\phi_1, \ldots, \phi_K$ also play the same role as "preprocessing functions," implementing steps such as whitening, normalizing by isotropic scaling, or projecting data onto a principal subspace. For example, to obtain a translation-invariant distance, we can subtract the grand mean response from each conditional distribution. That is, $\phi_k(\boldsymbol{x}) = \boldsymbol{x} - \mathbb{E}_{z\sim Q}[\mathbb{E}_{\boldsymbol{x}\sim F_k(\boldsymbol{x}\mid z)}[\boldsymbol{x}]]$.

Practical Estimation of Stochastic Shape Metrics

Stochastic shape distances (eq. 5.5) are generally more difficult to estimate than deterministic distances (eq. 5.2). In the deterministic case, the minimization over $\boldsymbol{T} \in \mathcal{G}$ is often a well-studied problem, such as linear assignment (Burkard et al., 2012) or the orthogonal Procrustes problem (Gower and Dijksterhuis, 2004). In the stochastic case, the conditional distributions $F_k(\cdot \mid \boldsymbol{z})$ often do not even have a parametric form, and can only be accessed by drawing samples—e.g. by repeated forward passes in an artificial network. Moreover, Wasserstein distances suffer a well-known curse of dimensionality: in n-dimensional spaces, the plug-in estimator converges at a very slow rate proportional to $s^{-1/n}$, where s is the number of samples (Niles-Weed and Rigollet, 2022).

Thus, to estimate shape distances with Wasserstein ground metrics, we assume that, $F_i^{\phi}(\cdot \mid \boldsymbol{z})$, is well-approximated by a Gaussian for each $\boldsymbol{z} \in \mathcal{Z}$. The 2-Wasserstein distance has a closed form expression in this case (Remark 2.31 in Peyré and Cuturi 2019 and Theorem 1 in Bhatia et al. 2019):

$$\mathcal{W}_2(\mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i), \mathcal{N}(\boldsymbol{\mu}_j, \Sigma_j)) = \left(\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2 + \min_{\boldsymbol{U}\in O(n)} \|\Sigma_i^{1/2} - \Sigma_j^{1/2}\boldsymbol{U}\|_F^2\right)^{1/2} \tag{5.6}$$

where $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ denotes a Gaussian density. It is important not to confuse the minimization over $\boldsymbol{U} \in O(n)$ in this equation with the minimization over nuisance transformations, $\boldsymbol{T} \in \mathcal{G}$, in the shape metric (eq. 5.5). These two minimizations arise for entirely different reasons, and the

Wasserstein distance is not invariant to rotations. Intuitively, we can estimate the Wasserstein-based shape metric by minimizing over $\boldsymbol{U} \in \mathcal{O}(n)$ and $\boldsymbol{T} \in \mathcal{G}$ in alternation (for full details, see subsection D.4.1).

In biological data, we often only have enough trials to estimate the first two moments of a neural response, and one may loosely appeal to the principle of maximum entropy to justify the Gaussian approximation (Uffink, 1995). In certain artificial networks the Gaussian assumption is satisfied exactly, such as in variational autoencoders (see subsection 5.4.3). Finally, even if the Gaussian assumption is violated, Equation 5.6 can still be a reasonable ground metric that is only sensitive to the first two moments (mean and covariance) of neural responses (see subsection D.5.3).

The Gaussian assumption is also unnecessary if we use the energy distance (eq. 5.4) as the ground metric instead of Wasserstein distance. Plug-in estimates of this distance converge at a much faster rate in high-dimensional spaces (Gretton et al., 2012; Sejdinovic et al., 2013). In this case, we propose a two-stage estimation procedure using iteratively reweighted least squares (Kuhn, 1973), followed by a "metric repair" step (Brickell et al., 2008) which resolves small triangle inequality violations due to distance estimation error (see Appendix D.4.2 for full details).

We discuss computational complexity in Appendix D.4.1.1 and provide user-friendly implementations of stochastic shape metrics at: `github.com/ahwillia/netrep`.

Interpolating Between Mean- and Covariance-only Metrics

An appealing feature of the 2-Wasserstein distance for Gaussian measures (eq. 5.6) is its decomposition into two terms that respectively depend on the mean and covariance. We reasoned that it would be useful to isolate the relative contributions of these two terms. Thus, we considered the following generalization of the 2-Wasserstein distance parameterized by a scalar,

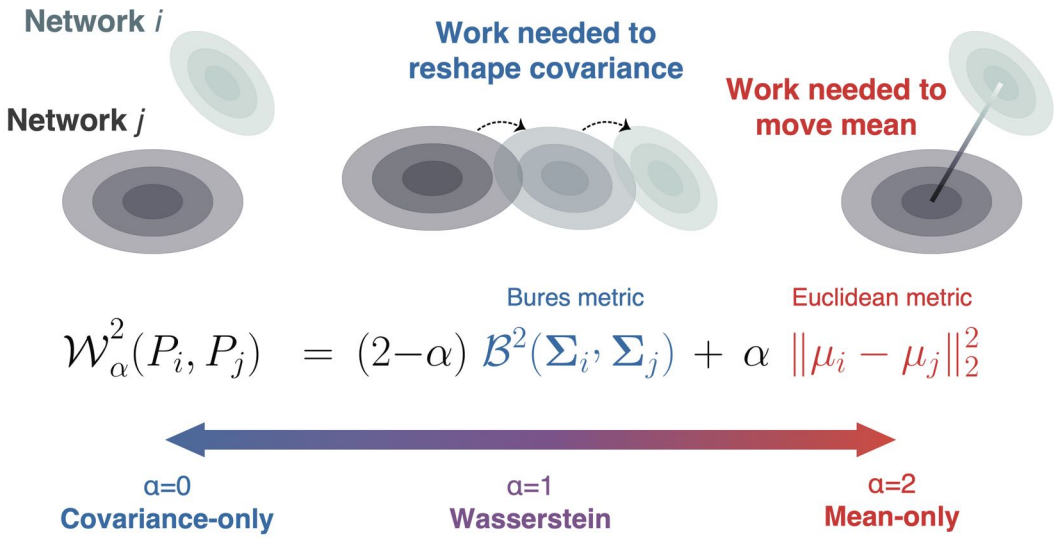

Figure 5.3: Interpolating between mean- and covariance-only metrics with the 2-Wasserstein distance.

$0 \leq \alpha \leq 2$:

$$\overline{\mathcal{W}}_2^{\alpha}(P_i, P_j) = \left(\alpha\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2 + (2-\alpha)\min_{\boldsymbol{U}\in O(n)} \|\Sigma_i^{1/2} - \Sigma_j^{1/2}\boldsymbol{U}\|_F^2\right)^{1/2} \tag{5.7}$$

where P_i, P_j are distributions with means $\boldsymbol{\mu}_i, \boldsymbol{\mu}_j$ and covariances Σ_i, Σ_j. In section D.5 we show that this defines a metric and, by extension, a shape metric when plugged into Equation 5.5.

We can use α to interpolate between a Euclidean metric on the mean responses and a metric on covariances known as the Bures metric (Bhatia et al., 2019). When $\alpha = 1$ and the distributions are Gaussian, we recover the 2-Wasserstein distance. Thus, by sweeping α, we can utilize a spectrum of stochastic shape metrics ranging from a distance that isolates differences in trial-average geometry ($\alpha = 2$) to a distance that isolates differences in noise covariance geometry ($\alpha = 0$). Figure 5.3 shows that distances along this spectrum can all be understood as generalizations of the usual "earth mover" interpretation of Wasserstein distance—the covariance-insensitive metric ($\alpha = 2$) only penalizes transport due to differences in the mean while the mean-insensitive metric ($\alpha = 0$) only penalizes transport due to differences in the orientation and scale of covariance. Simulation results in Supp. Figure D.1 provide additional intuition for the behavior of these shape distances as α is adjusted between 0 to 2.

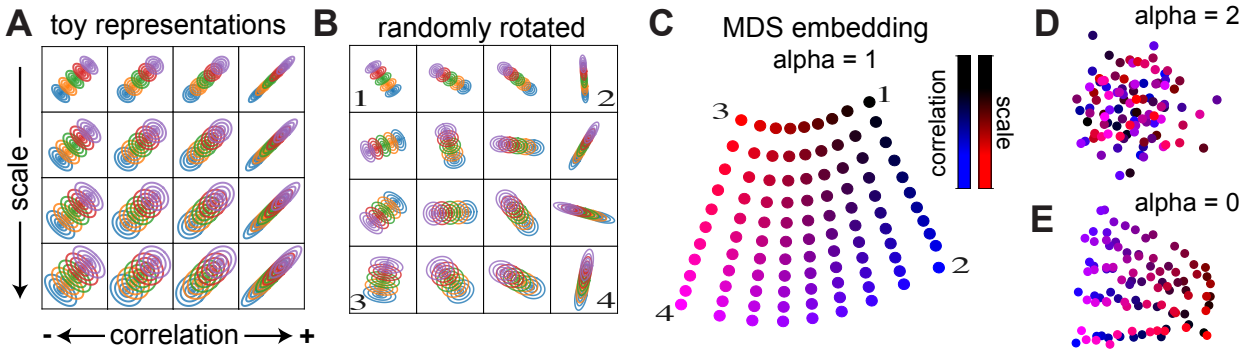

Figure 5.4: Toy Dataset. *(A)* 16 out of 99 "toy networks" with different correlation structure (horizontal axis) and covariance scale (vertical axis). Colors indicate distributions conditioned on different network inputs (as in Fig. 5.1B). *(B)* Same as A, with random rotations applied in neural activation space. These rotated representations are used in subsequent panels. *(C)* 2D embedding of networks in stochastic shape space ($\alpha = 1$ ground metric, $\mathcal{G} = O$). Numbered points correspond to labeled representations in panel B. Colormap indicates ground truth covariance parameters. *(D)* Same as C, but with $\alpha = 2$ (covariance-insensitive). *(E)* Same as C, but with $\alpha = 0$ (mean-insensitive).

5.4 Results and Applications

5.4.1 Toy Dataset

We begin by building intuition on a synthetic dataset in $n = 2$ dimensions with $M = 5$ inputs. Each response distribution was chosen to be Gaussian, and the mean responses were spaced linearly along the identity line. We independently varied the scale and correlation of the covariance, producing a 2D space of "toy networks." Figure 5.4A shows a sub-sampled 4×4 grid of toy networks. To demonstrate that stochastic shape metrics are invariant to nuisance transformations, we applied a random rotation to each network's activation space (Fig. 5.4B). The remaining panels show analyses for 99 randomly rotated toy networks spaced over a 11×9 grid (11 correlations and 9 scales).

Because the mean neural responses were constructed to be identical (up to rotation) across networks, existing measures of representational dissimilarity (CKA, RSA, CCA, etc.) all fail to capture the underlying structure of this toy dataset (Fig. D.2). In contrast, stochastic shape distances

can elegantly recover the 2D space of networks we constructed. In particular, we computed the 99×99 pairwise distance matrix between all networks (2-Wasserstein ground metric and rotation invariance, $\mathcal{G} = O$) and then performing multi-dimensional scaling (MDS; Borg and Groenen, 2005) to obtain a 2D embedding. This reveals a 2D grid of networks that maps onto our constructed arrangement (Fig. 5.4C). Again, since the toy networks have equivalent geometries on average, a deterministic metric obtained by setting $\alpha = 2$ in eq. 5.7 (covariance-insensitive metric) fails to recover this structure (Fig. 5.4D). Setting $\alpha = 0$ in eq. 5.7 (mean-insensitive stochastic metric) also fails to recover a sensible 2D embedding (Fig. 5.4E), since covariance ellipses of opposite correlation can be aligned by a 90° rotation. Thus, we are only able to fully distinguish the toy networks in Figure 5.4A by taking *both the mean and covariance* into account when computing shape distances.

In Fig. D.3 we show that using energy distance (eq. 5.4, $q = 1$) as the ground metric produces a similar result. Similar to Fig. 5.4C, MDS visualizations reveal the expected 2D manifold of toy networks. Indeed, these alternative distances correlate—but do not coincide exactly—with the distances shown in Figure 5.4 that were computed with a Wasserstein ground metric (Fig. D.3D).

5.4.2 Biological Data

Quantifying representational similarity is common in visual neuroscience (Kriegeskorte et al., 2008b; Shi et al., 2019). To our knowledge, past work has only quantified similarity in the geometry of trial-averaged responses and has not explored how the population geometry of noise varies across animals or brain regions (e.g. how the scale and shape of the response covariance changes). We leveraged stochastic shape metrics to perform a preliminary study on primary visual cortical recordings (VISp) from $K = 31$ mice in the Allen Brain Observatory.[1] The results we present below suggest: (a) across-animal variability in covariance geometry is comparable in magnitude to variability in trial-average geometry, (b) across-animal distances in covariance and

[1] See subsection D.2.2 for full details. Data are available at: `observatory.brain-map.org/visualcoding/`

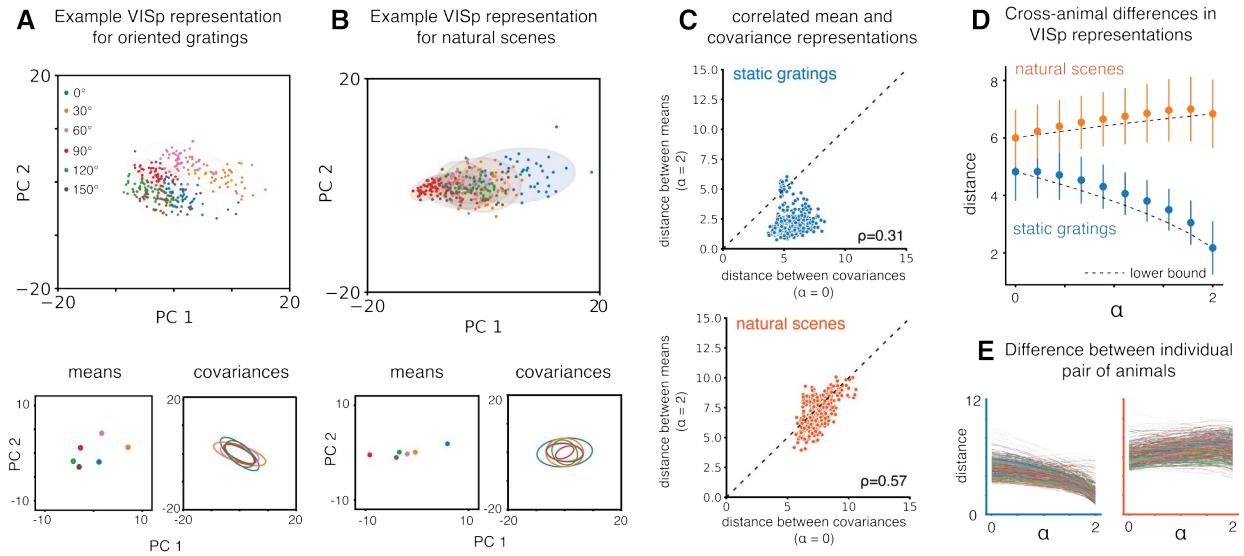

Figure 5.5: *(A)* In an example mouse VISp, neuronal responses form different means and covariances for six grating orientations. *(B)* In an example mouse VISp, neuronal responses form different means and covariances for six different natural scenes. *(C)* Covariance distances dominate differences between recording sessions for artificial grating stimuli, but not for natural scenes. *(D)* Averaged distance (across sessions) between mean responses for natural scenes is larger compared to for gratings. *(E)* Observations in *(C)* generally hold for individual pairs of recording sessions.

trial-average geometry are not redundant statistics as they are only weakly correlated, and (c) the relative contributions of mean and covariance geometry to inter-animal shape distances are stimulus-dependent. Together, these results suggest that neural response distributions contain nontrivial geometric structure in their higher-order moments, and that stochastic shape metrics can help dissect this structure.

We studied population responses (evoked spike counts, see Appendix D.2.2) to two stimulus sets: a set of 6 static oriented grating stimuli and a set of 119 natural scene images. Figure 5.5A shows neural responses from one animal to the oriented gratings within a principal component subspace (top), and the isolated mean and covariance geometries (bottom). Figure 5.5B similarly summarizes neural responses to six different natural scenes. In both cases, the scale and orientation of covariance within the first two PCs varies across stimuli. Furthermore, the scale of trial-to-trial variance was comparable to across-condition variance in the response means. These observations can be made individually within each animal, but a stochastic shape metric

(2-Wasserstein ground metric, and rotation invariance $\mathcal{G} = O$) enables us to quantify differences in covariance geometry *across animals*. We observed that the overall shape distance between two animals reflected a mixture of differences in trial-average and covariance geometry. By leveraging Equation 5.7, we observed that mean-insensitive ($\alpha = 0$) and covariance-insensitive ($\alpha = 2$) distances between animals have similar magnitudes and are weakly correlated (Fig. 5.5C-E).

Interestingly, the ratio of these two distances was reversed across the two stimulus sets—differences in covariance geometry across animals were larger relative to differences in average for oriented gratings, while the opposite was true for natural scenes (Fig. 5.5C-E). Later we will show an intriguingly similar trend when comparing representations between trained and untrained deep networks.

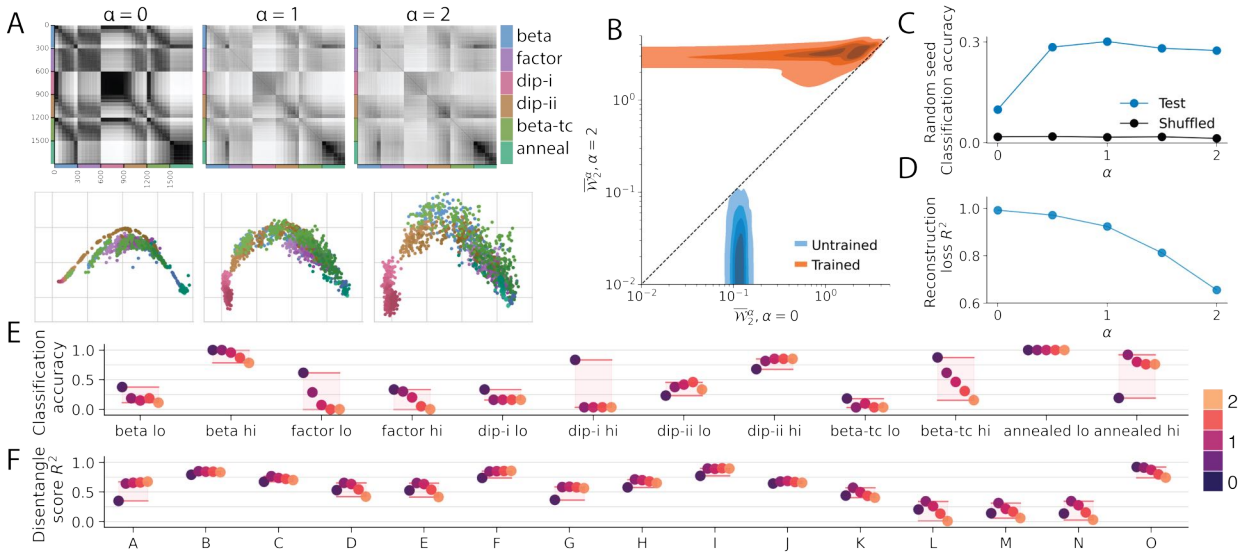

Figure 5.6: *(A)* Dissimilarity matrices with varying α (top) and corresponding 2D embeddings (bottom) for 1800 VAEs trained on `dSprites`. Six different VAE objectives (color hue) were used, each with six possible regularization strengths (tint) repeated over 50 random seeds. *(B)* Untrained networks were farther apart in mean-insensitive distance ($\alpha = 0$) while trained networks were largely separated by covariance-insensitive distance ($\alpha = 2$). *(C)* kNN prediction of a network's random seed. *(D)* Predicting reconstruction loss using kNN regression. *(E)* kNN prediction accuracy of VAE objective and regularization strength (hi vs lo) for metrics with different α (color scale). *(F)* Regression performance predicting factor disentanglement scores (see Apdx. D.2.3 for details).

5.4.3 Variational autoencoders and latent factor disentanglement

Variational autoencoders (VAEs; Kingma and Welling, 2019) are a well-known class of deep generative models that map inputs, $z \in \mathcal{Z}$ (e.g. images), onto conditional latent distributions, $F(\cdot \mid z)$, which are typically parameterized as Gaussian. Thus, for each high-dimensional input z_i, the encoder network produces a distribution $\mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$ in a relatively low-dimensional latent space ("bottleneck layer"). Because of this, VAEs are a popular tool for unsupervised, nonlinear dimensionality reduction (Batty et al., 2019; Goffinet et al., 2021; Higgins et al., 2021; Seninge et al., 2021). However, the vast majority of papers only visualize and analyze the means, $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M\}$, and ignore the covariances, $\{\Sigma_1, \ldots, \Sigma_M\}$, generated by these models. Stochastic shape metrics enable us to compare both the mean and covariance structure learned by different VAEs. Such comparisons can help us understand how modeling choices impact learned representations (Locatello et al., 2019) and how reproducible or identifiable learned representations are in practice (Khemakhem et al., 2020).

We studied a collection of 1800 trained networks [2] spanning six variants of the VAE framework at six regularization strengths and 50 random seeds (Locatello et al., 2019). Networks were trained on a synthetic image dataset called `dSprites` (Matthey et al., 2017), which is a well-established benchmark within the VAE disentanglement literature. Each image has a set of ground truth latent factors which Locatello et al. (2019) used to compute various disentanglement scores for all networks.

We computed stochastic shape distances between over 1.6 million network pairs, demonstrating the scalability of our framework. We computed rotation-invariant distances ($\mathcal{G} = O$) for the generalized Wasserstein ground metric ($\alpha = 0, 0.5, 1, 1.5, 2$ in equation 5.7; Fig. 5.6A) and for energy distance ($q = 1$ in equation 5.4; Fig. D.6A, Apdx. D.2.3.3). In all cases, different VAE variants visibly clustered together in different regions of the stochastic shape space (Fig. 5.6B, Fig. D.6B).

[2] Released by Locatello et al. (2019) at `https://github.com/google-research/disentanglement_lib`

Interestingly, the covariance-insensitive ($\alpha = 2$) shape distance tended to be larger than the mean-insensitive ($\alpha = 0$) shape distance (Fig. 5.6B), in agreement with the biological data on natural images (Fig. 5.5C, bottom). Even more interestingly, this relationship was reversed in untrained VAEs (Fig. 5.6B), similar to the biological data on artificial gratings (Fig. 5.5B, top). We trained several hundred VAEs on MNIST and CIFAR-10 to confirm these results persisted across more complex datasets (Fig. D.8; Apdx. D.2.3). Overall, this suggests that the ratio of $\alpha = 0$ and $\alpha = 2$ shape distances may be a useful summary statistic of representational complexity. We leave a detailed investigation of this to future work.

Since stochastic shape distances define proper metric spaces without triangle inequality violations, we can identify the k-nearest neighbors (kNN) of each network within this space, and use these neighborhoods to perform nonparametric classification and regression (Cover and Hart, 1967). This simple approach was sufficient to predict most characteristics of a network, including its random seed (Fig. 5.6C), average training reconstruction loss (Fig. 5.6D), its variant of the VAE objective including regularization strength (Fig. 5.6E), and various disentanglement scores (Fig. 5.6F). Detailed procedures for these analyses are provided in Appendix D.2.3. Notably, many of these predictions about network identity (Fig. 5.6E) were *more accurate* for the novel stochastic shape metrics ($0 \leq \alpha < 2$), compared to existing shape metrics ($\alpha = 2$, deterministic metric on the mean responses; Williams et al. 2021). Similarly, many disentanglement score predictions (Fig. 5.6F) improved when considering both covariance and means together ($0 < \alpha < 2$).

The fact that we can often infer a network's random seed from its position in stochastic shape space suggests that VAE features may have limited interpretability on this dataset. These limitations appear to apply both to the mean ($\alpha = 2$) and the covariance ($\alpha = 0$) representational geometries, as well as to intermediate interpolations. Future work that aims to use VAEs to identify interpretable structure within scientific data may find it useful to repeat this analysis to quantify the consistency of the learned representations across different optimization runs or different VAE architectures. Future work that aims to assess the identifiability of VAE representations may find

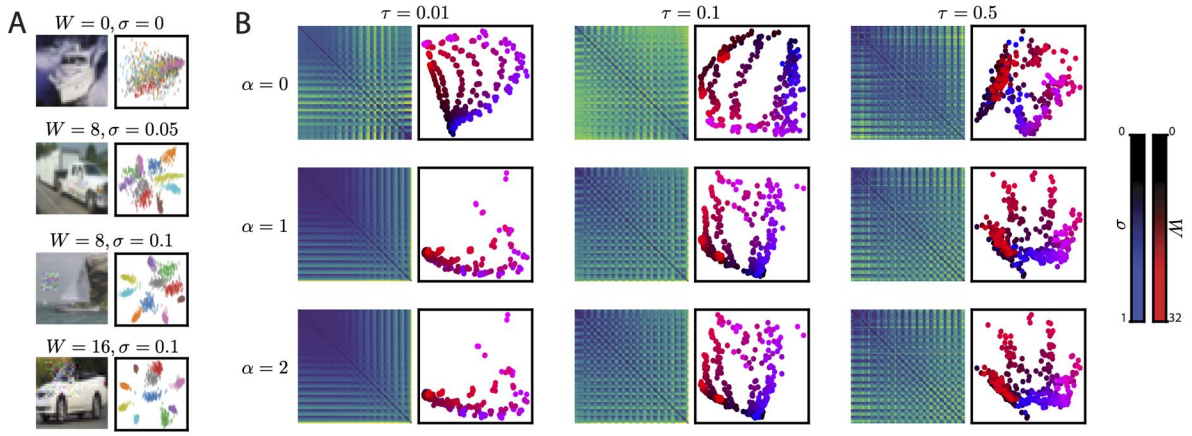

Figure 5.7: *(A) Left*, Patch-Gaussian augmented images for different patch width, W, and train-time noise level, σ. *Right*, MDS embedded activation vectors from Patch-Gaussian trained networks. Colors correspond to a different images, points correspond to independent samples, $\tau = 0.1$. *(B)* Distance matrices (left) and low-dimensional MDS embedding (right) of networks for different shape distances parameterized by α, and different levels of Gaussian corruption at test time, τ.

it useful to use stochastic shape metrics to perform similar analyses.

5.4.4 Effects of Patch-Gaussian data augmentation on artificial deep networks

Despite the success of artificial neural networks on vision tasks, they are still susceptible to small input perturbations (Hendrycks and Dietterich, 2019). A simple and popular approach to induce robustness in deep networks is Patch-Gaussian augmentation (Lopes et al., 2019), which adds Gaussian noise drawn from $\mathcal{N}(0, \sigma^2)$ to random image patches of width W during training (Fig. 5.7A, left column). At test time, network robustness is assessed with images with spatially uniform noise drawn from $\mathcal{N}(0, \tau^2)$. Importantly, the magnitude of noise at test time, τ, may be distinct from noise magnitude during training, σ. From Fig. 5.7A (right column), we see that using Patch-Gaussian augmentation (second-fourth rows) qualitatively leads to more robust hidden layer representations on noisy data compared to networks trained without it (first row). While Patch-Gaussian augmentation is empirically successful (for quantitative details, see Lopes et al.,

2019), how W and σ change hidden layer representations to confer robustness remains poorly understood.

To investigate, we trained a collection of 339 ResNet-18 networks (He et al., 2016) on CIFAR-10 (Krizhevsky et al., 2009), sweeping over 16 values of W, 7 values of σ, and 3 random seeds (see subsection D.2.4 for details). While the architecture is deterministic, we can consider it to be a stochastic mapping by absorbing the random Gaussian perturbation—parameterized by τ—into the first layer of the network and allowing the stochasticity to percolate through the network. Representations from a fully connected layer following the final average pooling layer were used for this analysis. We computed stochastic shape distances across all 57,291 pairs of networks across six values of τ and three shape metrics parameterized by $\alpha \in \{0, 1, 2\}$ defining the ground metric in Equation 5.7.

Sweeping across α and τ revealed a rich set of relationships across these networks (Fig. 5.7B and Fig. D.10). While a complete investigation is beyond the scope of this paper, several points are worthy of mention. First, the mean-insensitive ($\alpha = 0$, top row) and covariance-insensitive ($\alpha = 2$, bottom row) metrics produce clearly distinct MDS embeddings. Thus, the new notions of stochastic representational geometry developed in this paper (corresponding to $\alpha = 0$) provide new information to existing distance measures (corresponding to $\alpha = 2$). Second, the arrangement of networks in stochastic shape space reflects both W and σ, sometimes in a 2D grid layout that maps nicely onto the hyperparameter sweep (e.g. $\alpha = 0$ and $\tau = 0.01$). Networks with the same hyperparameters but different random seeds tend to be close together in shape space. Third, the test-time noise, τ, also intricately impacts the structure revealed by all metrics. Finally, embeddings based on 2-Wasserstein metric ($\alpha = 1$) qualitatively resemble embeddings based on the covariance-insensitive metric ($\alpha = 2$) rather than the mean-insensitive metric ($\alpha = 0$).

5.5 Discussion and Relation to Prior Work

We have proposed stochastic shape metrics as a novel framework to quantify representational dissimilarity across networks that respond probabilistically to fixed inputs. Very few prior works have investigated this issue. To our knowledge, methods within the deep learning literature like CKA (Kornblith et al., 2019) have been exclusively applied to deterministic networks. Of course, the broader concept of measuring distances between probability distributions appears frequently. For example, to quantify distance between two distributions over natural images, Fréchet inception distance (FID; Heusel et al., 2017) computes the 2-Wasserstein distance within a hidden layer representation space. While FID utilizes similar concepts to our work, it addresses a very different problem—i.e., how to compare two stimulus sets within the deterministic feature space of a single neural network, rather than how to compare the feature spaces of two stochastic networks over a single stimulus set.

A select number of reports in neuroscience, particularly within the fMRI literature, have addressed how measurement noise impacts RSA (an approach very similar to CKA). Diedrichsen et al. (2020) discuss how measurement noise induces positive bias in RSA distances, and propose approaches to correct for this bias. Similarly, Cai et al. (2016) propose a Bayesian approach to RSA that performs well in low signal-to-noise regimes. These papers essentially aim to develop methods that are robust to noise, while we were motivated to directly quantify differences in noise scale and geometric structure across networks. It is also common to use Mahalanobis distances weighted by inverse noise covariance to compute intra-network representation distances (Walther et al., 2016). This procedure does not appear to quantify differences in noise structure between networks, which we verified on a simple "toy dataset" (compare Fig. 5.4C to Supp. Fig. D.2). Furthermore, the Mahalanobis variant of RSA typically only accounts for a single, stimulus-independent noise covariance. In contrast, stochastic shape metrics account for noise statistics that change across stimuli.

While our primary contribution is to develop a new theoretical framework, several aspects of our experiments are noteworthy. Applying supervised learning methods (e.g. k-nearest neighbors) to study representational geometry is rare within the literature, and is only rigorously justified by our proof that stochastic shape metrics satisfy the triangle inequality. We also performed representational analysis across thousands of networks, which exceeds many other papers and matches the scale of experiments in Williams et al. (2021), despite our approach requiring novel and more computationally demanding algorithms.

Overall, our work meaningfully broadens the toolbox of representational geometry to quantify stochastic neural responses. The strengths and limitations of our work are similar to other approaches within this toolbox. A limitation in neurobiological recordings is that we only observe a subset of the total neurons in each network. Shi et al. (2019) document the effect of subsampling neurons on representational geometry. Intuitively, when the number of recorded neurons is large relative to the representational complexity, geometric features are not badly distorted (Kriegeskorte and Diedrichsen, 2016; Trautmann et al., 2019). We show that our results are not qualitatively affected by subsampling neurons in Supp. Fig. D.5. Another limitation is that representational geometry does not directly shed light on the algorithmic principles of neural computation (Maheswaranathan et al., 2019). Despite these challenges, representational dissimilarity measures are one of the few quantitative tools available to compare activations across large collections of complex, black-box models, and will be a mainstay of artificial and biological network analysis for the foreseeable future.

In practical applications, we found that stochastic shape metrics can be used as targeted diagnostic tool—e.g. for quantifying the sensitivity of learned representations in VAEs to arbitrary parameters like the random seed (Section 5.4.3)—or as a flexible exploratory analysis tool—e.g. in revealing different low-dimensional visualizations (Section 5.4.4) or identifying conditions under which the mean or covariance of neural responses dominates representational similarity (Section 5.4.2).

Our work demonstrates that large-scale, systematic analyses of stochastic representational geometry is possible and may shed light on a variety of application areas, including neurobiological noise correlations (Section 5.4.2), deep feature learning (Section 5.4.3), and robustness to noise in deep networks (Section 5.4.4).

6 Discussion

6.1 Adaptive Coding Efficiency in Neural Populations

Efficient coding has been foundational in the exploration of neural coding since its inception (Attneave, 1954; Barlow, 1961). It has led to the creation of numerous influential theories of neural coding, achieved by deriving the best ways to communicate information regarding unchanging stimulus distributions (Ganguli and Simoncelli, 2014). Nonetheless, natural environments are perpetually in flux, and this is mirrored in the adaptive and dynamic nature of sensory codes in the brain. Furthermore, while some normative studies focus on adaptive coding efficiency from a single neuron standpoint (e.g. Młynarski and Hermundstad, 2021), our theoretical understanding of population-level adaptive coding remains in its infancy. In this dissertation, we construct normative theories that interpret adaptive population coding through the lens of gain control, a mechanism with which a neuron adjusts its input-output sensitivity. We show that joint gain control, *without* synaptic plasticity, may orchestrate efficient neural population adaptation under dynamic sensory environments. Our approach offers a richer and more nuanced understanding of the adaptive processes in neural computation, and contributes a complementary viewpoint to the existing research on adaptive synaptic plasticity.

Specifically, Chapters 2 and 3 introduce a framework for adaptive statistical whitening. Using a novel overcomplete matrix factorization of the whitening transform, we devise an online whitening algorithm that directly maps onto a biologically plausible recurrent neural circuit with

primary neurons and gain-modulating interneurons. We further expand on this framework by integrating adaptive gain control with existing, seemingly disparate normative theories of adaptation relying on synaptic plasticity (Lipshutz et al., 2023; Pehlevan and Chklovskii, 2015) into a unified *multi-timescale* mechanistic model of adaptive whitening. This new model modifies gains and synapses at different rates, thereby enabling the network to effectively adapt to changing sensory statistics, while enhancing the robustness of the adaptation to an array of statistical contexts. Finally, in Chapter 4, we show that related ideas of adaptive gain control propagating through recurrent network circuitry can effectively explain the entire set of observed adaptation effects in V1 population adaptation data.

There remain open questions to be explored in future studies. In contemporary machine learning research, self-supervised learning methods are increasingly using decorrelation transformations like whitening to prevent representational collapse (Bardes et al., 2022; Ermolov et al., 2021; Hua et al., 2021; Zbontar et al., 2021). In a similar vein, it would be interesting to understand the effectiveness of *stacked* whitening layers on learning. Transformations such as batch normalization layers in deep neural networks have proven effective during train and test time for a variety of tasks (e.g. Krizhevsky et al., 2009). The decorrelation properties obtained by introducing unsupervised, online *whitening layers* in place of batch normalization layers in a network may prove useful in representation learning. Furthermore, while whitening has been predominantly focused on offline neural network training, there is a growing interest in devising adaptive (runtime) versions. Models proposed in this dissertation could potentially allow a network to adapt to changes in input statistics dynamically, requiring minimal alterations to the existing network (e.g. Ballé et al., 2020; Duong et al., 2023b; Hu et al., 2022; Mohan et al., 2021).

In the context of neuroscience, our framework poses intriguing predictions and opens up exciting avenues for future exploration in the study of neural population adaptation. Firstly, one of the core predictions that our framework presents is the stability of between-neuron synaptic connectivity, denoted as $\mathbf{W}$, through the process of adaptation. To evaluate this prediction, future

studies could investigate changes in synaptic connectivity patterns under varying conditions of neuronal adaptation, enabling us to understand how stable these connections are.

Secondly, our normative objective points towards the central role of gain control in population adaptation, whether that be via gain-modulating interneurons within a circuit (Chapters 2,3), or within the primary neurons themselves (Chapter 4). An intriguing direction of research could focus on seeking empirical evidence for this postulated stimulus-dependent gain control in either cell type. This could potentially be achieved by examining changes in neuron membrane conductance during adaptation, which may be mediated by fluctuations in slow hyperpolarizing Ca^{2+}- and Na^{+}-induced K^{+} currents, as suggested by Sanchez-Vives et al. (2000).

Lastly, despite significant strides in uncovering the circuits involved in sensory adaptation (Wanner and Friedrich, 2020), the precise structure of functional recurrent connectivity in adapting circuits remains unknown. In Chapters 2 and 3, we derive networks with symmetric feedforward and feedback weights between primary and interneurons. This architectural prediction can be validated using anatomical measurements in networks known to compute whitening. For instance, neurons in the zebrafish olfactory bulb are known to adaptively whiten their responses (Friedrich and Wanner, 2021). It has been shown that subsets of excitatory and inhibitory neurons in this area have reciprocal connections (Friedrich and Wiechert, 2014). Future work can examine these reciprocally-connected neurons during adaptation directly, while validating whether the interneuron gains adapt according to our theory.

Moreover in Chapter 4, we discuss how different variations of $\mathbf{W}$ can give rise to qualitatively similar V1 population adaptation effects as those shown in our main findings (Appendix C.1). However, not all forms of $\mathbf{W}$ lead to identical outcomes. Indeed, by analyzing the responses before and after adaptation, we can gain valuable insights into the functional recurrent connectivity between neurons. Conducting adaptation experiments using a wider array of probabilistic stimulus ensembles, $p(\mathbf{s})$, could provide further constraints for solving this complex functional inverse problem. In summary, the insights and predictions offered by our framework open up avenues

of research in understanding the role and mechanisms of synaptic connectivity and gain control in adaptation.

6.2 Stochastic Shape Metrics

The last part of this dissertation derives and presents stochastic shape metrics as an effective framework for comparing and quantifying the representational dissimilarity across networks that produce *stochastic* responses to fixed inputs. However, as with any new statistical tool, several research directions warrant exploration to further refine and apply this approach.

While our method operates in "feature space" (i.e. in the neural response domain), other methods in the literature, such as Centered Kernel Alignment (Kornblith et al., 2019) and Representational Similarity Analysis (Kriegeskorte and Diedrichsen, 2019), deal with comparing stimulus $\times$ stimulus representational similarity matrices. These methods have the benefit of being agnostic to the number of neurons in each dataset being compared, so long as they each have the same number of presented stimuli. While Williams et al. (2021) show that these kernel-based methods can be viewed as a special case of shape metrics, they still only deal with deterministic responses. Future research might seek to investigate potential extensions these techniques in the context of stochastic networks.

Previous methods have considered noise in representational geometry comparisons as well, but have treated noise as a nuisance variable. For example, the impact of *measurement noise* on RSA, particularly within the realm of human functional brain imaging literature, has been investigated. These studies predominantly aim to develop noise-robust methodologies, whereas our work is motivated by the goal to quantify differences in noise scale and structure across networks. However, there of course exists measurement noise in the neural recordings we are interested in as well, which may impact the estimation of each response shape. Future work could thus attempt to reconcile these two approaches, perhaps leading to a hybrid methodology that is

not only robust to noise but can also quantify its impact. Moreover, despite being powerful tools, representational dissimilarity measures do not directly shed light on the underlying algorithmic principles of neural computation. Consequently, future research should strive to connect these measures with theoretical foundations of neural computation.

While we demonstrated that our results are not qualitatively affected by subsampling neurons, there is a need for studies documenting the effect of subsampling neurons on representational geometry in more depth. Additionally, researchers might explore how these effects can be minimized or compensated for to yield more accurate and reliable results. Our work also opens the door for applying supervised learning methods to the study of representational geometry, an approach that has seldom been explored in the literature. The computational implications and benefits of this approach could be an intriguing avenue for future research.

Taken together, the development of stochastic shape metrics adds a powerful tool to the toolbox for the analysis of representational geometry, allowing for the quantification of stochastic neural population responses.

6.2.1 Concluding remarks

This dissertation proposes new theories of neural population adaptation and new neural population statistical data analysis tools. First, we study adaptive coding efficiency in sensory populations with an emphasis on gain control as a key mechanism. Our models provide a new perspective on how dynamic sensory codes in the brain respond to fluctuating environmental statistics, offering insight into the complementary roles of synaptic plasticity and gain control in adaptation. Second, we introduce stochastic shape metrics, a general approach to comparing *noisy* representational geometry across different neural networks. We show that this framework has the potential to improve our understanding of how trial variability impacts neural coding in different networks. In sum, this work lays the foundation for future research on understanding dynamic neural sensory population codes and population geometry.

A | Adaptive Whitening with Overcomplete Gain Control

A.1 Optimal Solution to Symmetric Whitening Objective

In this section, we prove that the optimal solution to the optimization problem in equation 2.2 is given by $\mathbf{r}_t = \mathbf{C}_{ss}^{-1/2}\mathbf{s}_t$ for $t = 1, \dots, T$ (we treat the case that $T < \infty$).

We first recall Von Neumann's trace inequality (see, e.g., Carlsson, 2021, Theorem 3.1).

Lemma A.1 (Von Neumann's trace inequality)**.** *Suppose* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times m}$ *with* $n \leqslant m$*. Let* $\sigma_1^A \geqslant \cdots \geqslant \sigma_n^A \geqslant 0$ *and* $\sigma_1^B \geqslant \cdots \geqslant \sigma_n^B \geqslant 0$ *denote the respective singular values of* $\mathbf{A}$ *and* $\mathbf{B}$*. Then*

$$\operatorname{Tr}(\mathbf{A}\mathbf{B}^\top) \leqslant \sum_{i=1}^{n} \sigma_i^A \sigma_i^B.$$

Furthermore, equality holds if and only if $\mathbf{A}$ *and* $\mathbf{B}$ *share left and right singular vectors.*

We can now proceed with the proof of our result. We first concatenate the inputs and outputs into data matrices $\mathbf{X} = [\mathbf{s}_1, \dots, \mathbf{s}_T] \in \mathbb{R}^{N\times T}$ and $\mathbf{Y} = [\mathbf{r}_1, \dots, \mathbf{r}_T] \in \mathbb{R}^{N\times T}$. We can write equation 2.2 as follows:

$$\min_{\mathbf{Y}} \|\mathbf{X} - \mathbf{Y}\|_F^2 \qquad \text{subject to} \qquad \mathbf{Y}\mathbf{Y}^\top = T\mathbf{I}_N.$$

Expanding, substituting in with the constraint $\mathbf{Y}\mathbf{Y}^\top = T\mathbf{I}_N$ and dropping terms that do not depend on $\mathbf{Y}$ results in the objective

$$\max_{\mathbf{Y}} \operatorname{Tr}(\mathbf{X}\mathbf{Y}^\top) \qquad \text{subject to} \qquad \mathbf{Y}\mathbf{Y}^\top = T\mathbf{I}_N.$$

By Von Neumann's trace inequality, the trace is maximized when the singular vectors of $\mathbf{Y}$ are aligned with the singular vectors of $\mathbf{X}$. In particular, if the SVD of $\mathbf{X}$ is given by $\mathbf{X} = \mathbf{U}_x\mathbf{S}_x\mathbf{V}_x^\top$, then the optimal $\mathbf{Y}$ is given by $\mathbf{Y} = \sqrt{T}\mathbf{U}_x\mathbf{V}_x^\top$, which is precisely $\mathbf{C}_{ss}^{-1/2}\mathbf{X}$, where $\mathbf{C}_{ss} := \frac{1}{T}\mathbf{X}\mathbf{X}^\top = \mathbf{U}_x\mathbf{S}_x^2\mathbf{U}_x^\top$.

A.2 Proof of Proposition 2.1

Proof of Proposition 2.1. Suppose Equation 2.1 holds. Then, for $i = 1, \ldots, K$,

$$\langle (\mathbf{w}_i^\top \mathbf{r}_t)^2 \rangle_t = \langle \mathbf{w}_i^\top \mathbf{r}_t \mathbf{r}_t^\top \mathbf{w}_i \rangle_t = \mathbf{w}_i^\top \mathbf{w}_i = 1.$$

Therefore, Equation 2.4 holds.

Now suppose Equation 2.4 holds. Let $\mathbf{v} \in \mathbb{R}^N$ be an arbitrary unit vector. Then $\mathbf{v}\mathbf{v}^\top \in \mathbb{S}^N$ and by Equation 2.3, there exist $g_1, \ldots, g_K \in \mathbb{R}$ such that

$$\mathbf{v}\mathbf{v}^\top = g_1\mathbf{w}_1\mathbf{w}_1^\top + \cdots + g_K\mathbf{w}_K\mathbf{w}_K^\top. \tag{A.1}$$

We have

$$\mathbf{v}^\top \langle \mathbf{r}_t\mathbf{r}_t^\top \rangle_t \mathbf{v} = \operatorname{Tr}(\mathbf{v}\mathbf{v}^\top \langle \mathbf{r}_t\mathbf{r}_t^\top \rangle_t) = \sum_{i=1}^K g_i \operatorname{Tr}(\mathbf{w}_i\mathbf{w}_i^\top \langle \mathbf{r}_t\mathbf{r}_t^\top \rangle_t) = \sum_{i=1}^K g_i \operatorname{Tr}(\mathbf{w}_i\mathbf{w}_i^\top) = \operatorname{Tr}(\mathbf{v}\mathbf{v}^\top) = 1. \tag{A.2}$$

The first equality is a property of the trace operator. The second and fourth equalities follow from Equation A.1 and the linearity of the trace operator. The third equality follows from Equation 2.4, the cyclic property of the trace, and the fact that each $\mathbf{w}_i$ is a unit vector. The final equality holds

because $\mathbf{v}$ is a unit vector. Since Equation A.2 holds for every unit vector $\mathbf{v} \in \mathbb{R}^N$, Equation 2.1 holds. □

A.3 Frame Factorizations of Symmetric Matrices

A.3.1 Analytic solution for the optimal gains

Recall that the optimal solution of the symmetric objective in Equation 2.5 is given by $\mathbf{r}_t = \mathbf{C}_{ss}^{-1/2}\mathbf{s}_t$ for $t = 1, 2, \ldots$. In our neural circuit with interneurons and gain control, the outputs of the primary neurons at equilibrium is (given in Equation 2.8, but repeated here for clarity),

$$\bar{\mathbf{r}}_t = \left[\mathbf{I}_N + \mathbf{W}\,\mathrm{diag}\,(\mathbf{g})\,\mathbf{W}^\top\right]^{-1}\mathbf{s}_t,$$

where $\mathbf{W} \in \mathbb{R}^{N\times K}$ is overcomplete, arbitrary (provided Equation 2.3 holds), and *fixed*; and elements of $\mathbf{g} \in \mathbb{R}^K$ can be interpreted as learnable scalar gains. The circuit performs symmetric whitening when the gains $\mathbf{g}$ satisfy the relation

$$\mathbf{I}_N + \mathbf{W}\,\mathrm{diag}\,(\mathbf{g})\,\mathbf{W}^\top = \mathbf{C}_{ss}^{1/2}. \tag{A.3}$$

It is informative to contrast this with conventional approaches to symmetric whitening, which rely on eigendecompositions,

$$\mathbf{V}\,\mathrm{diag}\,(\boldsymbol{\lambda})^{1/2}\,\mathbf{V}^\top = \mathbf{C}_{ss}^{1/2},$$

where $\mathbf{V} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{\lambda}$ are the eigenvectors and eigenvalues of $\mathbf{C}_{ss}$, respectively. Note that in this eigenvector formulation, both vector quantities (columns of $\mathbf{V}$) and scalar quantities (elements of $\boldsymbol{\lambda}$) need to be learned, whereas in our formulation (Equation A.3), *only scalars* need to be learned

(elements of $\mathbf{g}$).

When $K \geq N(N+1)/2$, we can explicitly solve for the optimal gains $\mathbf{g}^*$ (derived in the next subsection):

$$\mathbf{g}^* = \left[\left(\mathbf{W}^\top \mathbf{W} \right)^{\circ 2} \right]^\dagger \left[\mathbf{w}_1^\top \mathbf{C}_{ss}^{1/2} \mathbf{w}_1 - 1, \ldots, \mathbf{w}_K^\top \mathbf{C}_{ss}^{1/2} \mathbf{w}_K - 1 \right]^\top. \tag{A.4}$$

A.3.2 Isolating $\mathbf{g}$ embedded in a diagonal matrix

In the upcoming subsection, our variable of interest, $\mathbf{g}$, is embedded along the diagonal of a matrix, then wedged between two fixed matrices, i.e. $\mathbf{A}_1 \operatorname{diag}(\mathbf{g}) \mathbf{A}_2$. We employ the following identity to isolate $\mathbf{g}$,

$$\operatorname{diag}\left(\mathbf{A}_1 \operatorname{diag}(\mathbf{g}) \mathbf{A}_2 \right) = \left(\mathbf{A}_1 \circ \mathbf{A}_2^\top \right) \mathbf{g}, \tag{A.5}$$

where, on the left-hand-side, the inner $\operatorname{diag}(\cdot)$ forms a diagonal matrix from a vector, the outer $\operatorname{diag}(\cdot)$ returns the diagonal of a matrix as a vector, and $\circ$ is the element-wise Hadamard product.

A.3.3 Deriving optimal gains

Let $\mathbf{C} \in \mathbb{S}^N$, where $\mathbb{S}^N$ is the set of symmetric $N \times N$ matrices. Suppose $\mathbf{g} \in \mathbb{R}^K$ is such that the following holds:

$$\mathbf{W} \operatorname{diag}(\mathbf{g}) \mathbf{W}^\top = \mathbf{C} \tag{A.6}$$

where $\mathbf{W} \in \mathbb{R}^{N \times K}$ is some fixed, arbitrary, frame with $K \geq \frac{N(N+1)}{2}$ (i.e. a representation that is $O(N^2)$ overcomplete). To solve for $\mathbf{g}$, we multiply both sides of Equation A.6 from the left and

right by $\mathbf{W}^\top$ and $\mathbf{W}$, respectively, then take the diagonal[1] of the resultant matrices,

$$\operatorname{diag}\left(\mathbf{W}^\top\mathbf{W}\operatorname{diag}\left(\mathbf{g}\right)\mathbf{W}^\top\mathbf{W}\right) = \operatorname{diag}\left(\mathbf{W}^\top\mathbf{C}\mathbf{W}\right). \tag{A.7}$$

Finally, employing the identity in Equation A.5 yields

$$(\mathbf{W}^\top\mathbf{W})^{\circ 2}\mathbf{g} = \operatorname{diag}\left(\mathbf{W}^\top\mathbf{C}\mathbf{W}\right), \tag{A.8}$$

$$\mathbf{g} = \left[(\mathbf{W}^\top\mathbf{W})^{\circ 2}\right]^\dagger \operatorname{diag}\left(\mathbf{W}^\top\mathbf{C}\mathbf{W}\right), \tag{A.9}$$

where $(\cdot)^{\circ 2}$ denotes element-wise squaring, $(\mathbf{W}^\top\mathbf{W})^{\circ 2}$ is positive semidefinite by the Schur product theorem and $(\cdot)^\dagger$ denotes the Moore-Penrose pseudoinverse. Thus, *any* $N \times N$ symmetric matrix, can be encoded as a vector, $\mathbf{g}$, with respect to an arbitrary fixed frame, $\mathbf{W}$, by solving a standard linear system of K equations of the form $\mathbf{A}\mathbf{g} = \mathbf{b}$. Importantly, when $K = \frac{N(N+1)}{2}$ and the columns of $\mathbf{W}$ are not collinear, we have empirically found the matrix on the LHS, $(\mathbf{W}^\top\mathbf{W})^{\circ 2}$, to be positive definite, so the vector $\mathbf{g}$ is uniquely defined.

Without loss of generality, assume that the columns of $\mathbf{W}$ are unit-norm (otherwise, we can always normalize them by absorbing their lengths into the elements of $\mathbf{g}$). Furthermore, assume without loss of generality that $\mathbf{C} \in \mathbb{S}^N_{++}$, the set of all symmetric positive definite matrices (e.g. covariance, precision, PSD square roots, etc.). When $\mathbf{C}$ is a covariance matrix, then $\operatorname{diag}(\mathbf{W}^\top\mathbf{C}\mathbf{W})$ can be interpreted as a vector of projected variances of $\mathbf{C}$ along each axis spanned by $\mathbf{W}$. Therefore, Equation A.8 states that the vector $\mathbf{g}$ is linearly related to the vector of projected variances via the element-wise squared frame Gramian, $(\mathbf{W}^\top\mathbf{W})^{\circ 2}$.

[1] Similar to commonly-used matrix libraries, the $\operatorname{diag}(\cdot)$ operator here is overloaded and can map a vector to a matrix or vice versa.

A.4 Saddle Point Property

In this section, we prove the following minimax property (for the case $t = 1, \ldots, T$ with T finite):

$$\min_{\{\mathbf{r}_t\}} \max_{\mathbf{g}} \langle \ell(\mathbf{s}_t, \mathbf{r}_t, \mathbf{g}) \rangle_t = \max_{\mathbf{g}} \min_{\{\mathbf{r}_t\}} \langle \ell(\mathbf{s}_t, \mathbf{r}_t, \mathbf{g}) \rangle_t. \tag{A.10}$$

The proof relies on the following minimax property for a function that satisfies the saddle point property (Boyd and Vandenberghe, 2004, section 5.4).

Theorem A.2. *Let $V \subseteq \mathbb{R}^n$, $W \subseteq \mathbb{R}^m$ and $f : V \times W \to \mathbb{R}$. Suppose f satisfies the saddle point property; that is, there exists $(\mathbf{a}^*, \mathbf{b}^*) \in V \times W$ such that*

$$f(\mathbf{a}^*, \mathbf{b}) \leqslant f(\mathbf{a}^*, \mathbf{b}^*) \leqslant f(\mathbf{a}, \mathbf{b}^*), \qquad \textit{for all } (\mathbf{a}, \mathbf{b}) \in V \times W.$$

Then

$$\min_{\mathbf{a} \in V} \max_{\mathbf{b} \in W} f(\mathbf{a}, \mathbf{b}) = \max_{\mathbf{b} \in W} \min_{\mathbf{a} \in V} f(\mathbf{a}, \mathbf{b}) = f(\mathbf{a}^*, \mathbf{b}^*).$$

In view of Theorem A.2, it suffices to show there exists $(\mathbf{r}_1^*, \ldots, \mathbf{r}_T^*, \mathbf{g}^*)$ such that

$$\ell(\mathbf{r}_1^*, \ldots, \mathbf{r}_T^*, \mathbf{g}) \leqslant \ell(\mathbf{r}_1^*, \ldots, \mathbf{r}_T^*, \mathbf{g}^*) \leqslant \ell(\mathbf{r}_1, \ldots, \mathbf{r}_T, \mathbf{g}^*), \qquad \text{for all } \mathbf{r}_1, \ldots, \mathbf{r}_T \in \mathbb{R}^N \text{ and } \mathbf{g} \in \mathbb{R}^K. \tag{A.11}$$

Define $\mathbf{r}_t^* := \mathbf{C}_{ss}^{-1/2} \mathbf{s}_t$ for all $t = 1, \ldots, T$ and define $\mathbf{g}^*$ as in equation A.4 so that equation A.3 holds. Then, for all $\mathbf{g} \in \mathbb{R}^K$,

$$\ell(\mathbf{r}_1^*, \ldots, \mathbf{r}_T^*, \mathbf{g}) = \frac{1}{T} \sum_{t=1}^{T} \|\mathbf{s}_t - \mathbf{r}_t^*\|_2^2.$$

Therefore, the first inequality in equation A.11 holds (in fact it is an equality for all $\mathbf{g}$). Next, we have

$$\begin{aligned}
\ell(\mathbf{r}_1, \ldots, \mathbf{r}_T, \mathbf{g}^*) &= \frac{1}{T}\sum_{t=1}^{T} \|\mathbf{s}_t - \mathbf{r}_t\|_2^2 + \frac{1}{T}\sum_{t=1}^{T} \mathrm{Tr}\left[\mathbf{W}\mathrm{diag}(\mathbf{g}^*)\mathbf{W}^\top(\mathbf{r}_t\mathbf{r}_t^\top - \mathbf{I}_N)\right] \\
&= \frac{1}{T}\sum_{t=1}^{T} (\mathbf{s}_t^\top\mathbf{s}_t - 2\mathbf{s}_t^\top\mathbf{r}_t) + \frac{1}{T}\sum_{t=1}^{T} \mathrm{Tr}\left[(\mathbf{I}_N + \mathbf{W}\mathrm{diag}(\mathbf{g}^*)\mathbf{W}^\top)(\mathbf{r}_t\mathbf{r}_t^\top - \mathbf{I}_N)\right] \\
&= \frac{1}{T}\sum_{t=1}^{T} (\mathbf{s}_t^\top\mathbf{s}_t - 2\mathbf{s}_t^\top\mathbf{r}_t) + \frac{1}{T}\sum_{t=1}^{T} \mathrm{Tr}\left[\mathbf{C}_{ss}^{1/2}(\mathbf{r}_t\mathbf{r}_t^\top - \mathbf{I}_N)\right]
\end{aligned}$$

Since $\mathbf{C}_{ss}^{1/2}$ is positive definite, $\ell(\mathbf{r}_1, \ldots, \mathbf{r}_T, \mathbf{g}^*)$ is strictly convex in $(\mathbf{r}_1, \ldots, \mathbf{r}_T)$ with its unique minimum obtained at $\mathbf{r}_t = \mathbf{C}_{ss}^{-1/2}\mathbf{s}_t$ for all $t = 1, \ldots, T$ (to see this, differentiate with respect to $\mathbf{r}_1, \ldots, \mathbf{r}_T$, set the derivatives equal to zero and solve for $\mathbf{r}_1, \ldots, \mathbf{r}_T$). This establishes the second inequality in equation A.11 holds. Therefore, by Theorem A.2, equation A.10 holds.

A.5 Weighted Average Update Rule for Gains

The update for $\mathbf{g}$ in Equation 2.10 can be generalized to allow for a weighted average over past samples. In particular, the general update is given by

$$\mathbf{g} \leftarrow \mathbf{g} + \eta\left(\frac{1}{Z}\sum_{s=1}^{t} \gamma^{t-s}\mathbf{z}_s^{\circ 2} - \mathbf{1}\right),$$

where $\gamma \in [0, 1]$ determines the decay rate and $Z := 1 + \gamma + \cdots + \gamma^{t-1}$ is a normalizing factor.

A.6 Batched and Offline Algorithms for Whitening with RNNs via Gain Modulation

In addition to the fully-online algorithm provided in the main text (Algorithm 1), we also provide two variants below. In many applications, streaming inputs arrive in batches rather than one at a time (e.g. video streaming frames). Similarly for conventional offline stochastic gradient descent training, data is sampled in batches. Algorithm 3 would be one way to accomplish this in our framework, where the main difference between the fully online version is taking the mean across samples in the batch to yield average gain update $\Delta \mathbf{g}$ term. Furthermore, in the fully offline setting when the covariance of the inputs, $\mathbf{C}_{ss}$ is known, Algorithm 4 presents a way to whiten the covariance directly.

Algorithm 3: Batched symmetric whitening

Input: Data matrix $\mathbf{X} \in \mathbb{R}^{N \times T}$ (centered)
Initialize: $\mathbf{W} \in \mathbb{R}^{N \times K}$; $\mathbf{g} \in \mathbb{R}^{K}$; η; batch size B
while not converged **do**
  $\mathbf{X}_B \leftarrow \texttt{sample\_batch}(\mathbf{X}, B)$
  $\mathbf{Y}_B \leftarrow [\mathbf{I}_N + \mathbf{W}\,\mathrm{diag}\,(\mathbf{g})\,\mathbf{W}^\top]^{-1}\mathbf{X}_B$
  $\mathbf{Z}_B \leftarrow \mathbf{W}^\top \mathbf{Y}_B$
  $\Delta \mathbf{g} \leftarrow \frac{1}{T}\mathrm{diag}(\mathbf{Z}_B \mathbf{Z}_B^\top) - \mathbf{1}$
  $\mathbf{g} \leftarrow \mathbf{g} + \eta\ \texttt{mean}(\Delta \mathbf{g}, \texttt{axis=1})$
end while

Algorithm 4: Offline symmetric whitening

Input: Input covariance $\mathbf{C}_{ss}$
Initialize: $\mathbf{W} \in \mathbb{R}^{N \times K}$; $\mathbf{g} \in \mathbb{R}^{K}$; η
while not converged **do**
  $\mathbf{M} \leftarrow [\mathbf{I}_N + \mathbf{W}\,\mathrm{diag}\,(\mathbf{g})\,\mathbf{W}^\top]^{-1}$
  $\mathbf{C}_{rr} \leftarrow \mathbf{M}\mathbf{C}_{ss}\mathbf{M}$
  $\Delta \mathbf{g} \leftarrow \mathrm{diag}\,(\mathbf{W}^\top \mathbf{C}_{rr} \mathbf{W}) - \mathbf{1}$
  $\mathbf{g} \leftarrow \mathbf{g} + \eta \Delta \mathbf{g}$
end while

A.7 Normalizing Ill-conditioned Inputs with Non-negative Constrained Gains

A.7.1 Quantifying whitening error

Whitening with non-negative gains does not, in general, produce an output with identity covariance matrix; therefore, quantifying algorithm performance with the error defined in the main text would not be informative. Because this extension shares similarities with ideas of regularized whitening, in which principal axes whose eigenvalues are below a certain threshold are unaffected by the whitening transform, we quantify algorithmic performance using thresholded Spectral Error,

$$\text{Spectral Error} := \frac{1}{N}\sum_{i}^{N} \max(\lambda_i - 1, 0)^2,$$

where λ_i is the i^{th} eigenvalue of $\mathbf{C}_{rr}$. Here, as in the main text, we set the threshold to 1. Figure A.1 shows that this network reduces spectral error. Importantly, the converged solution depends on the initial choice of frame (see next subsection).

A.7.2 Geometric intuition behind thresholded whitening with non-negative gains

In general, the modified objective with rectified gains (Equation 2.14) does not statistically whiten the inputs $\mathbf{s}_1, \mathbf{s}_2, \ldots$, but rather adapts the non-negative gains $g_1, \ldots, g_K$ to ensure that the variances of the outputs $\mathbf{r}_1, \mathbf{r}_2, \ldots$ in the directions spanned by the frame vectors $\{\mathbf{w}_1, \ldots, \mathbf{w}_K\}$ are bounded above by unity (Figure A.2). This one-sided normalization carries interesting implications for how and when the circuit statistically whitens its outputs, which can be compared

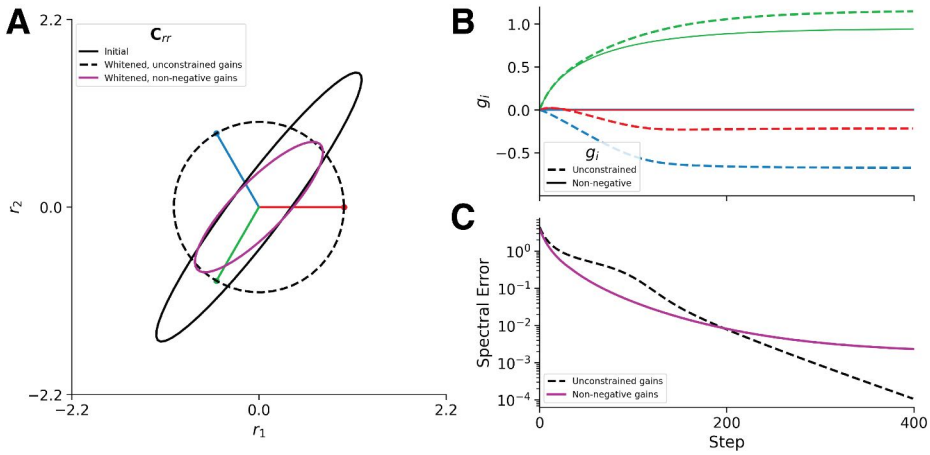

Figure A.1: Whitening ill-conditioned inputs with non-negative gains. **A)** An equi-angular frame (red, blue, green; see Sec. 2.5.2) whitening ill-conditioned inputs. **B)** Gains as algorithm progresses, using updates with either rectified or unrectified constraints. **C)** Spectral Error (see text).

with experimental observations. For instance, the circuit performs symmetric whitening if and only if there are non-negative gains such that Equation A.3 holds (see, e.g., the top right example in Figure A.2), which corresponds to cases such that the matrix $\mathbf{C}_{ss}^{1/2}$ is an element of the following cone (with its vertex translated by $\mathbf{I}_N$):

$$\left\{ \mathbf{I}_N + \sum_{i=1}^{K} g_i \mathbf{w}_i \mathbf{w}_i^\top : \mathbf{g} \in \mathbb{R}_+^K \right\}.$$

On the other hand, if the variance of an input projection is less than unity — i.e., $\mathbf{w}_i^\top \mathbf{C}_{ss} \mathbf{w}_i \leqslant 1$ for some i — then the corresponding gain g_i remains zero. When this is true for all $i = 1, \ldots, K$, the gains all remain zero and the circuit output is equal to its input (see, e.g., the bottom middle panel of Figure A.2).

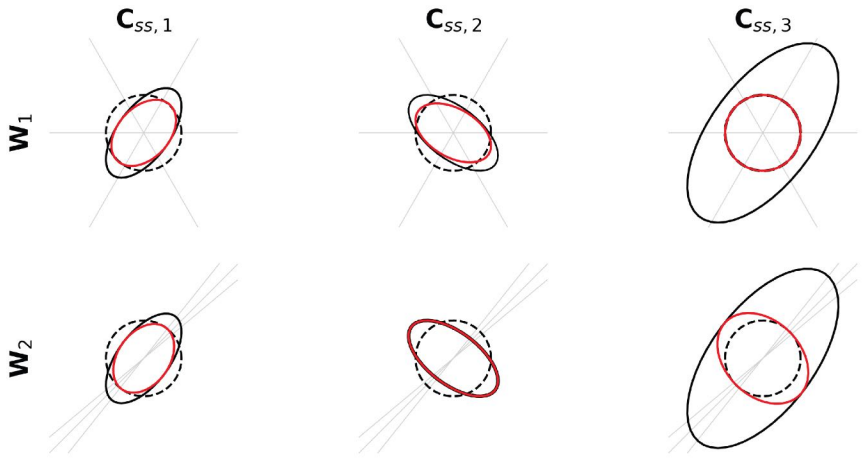

Figure A.2: Geometric intuition of whitening with/without inequality constraint. Whitening efficacy using non-negative gains depends on $\mathbf{W}$ and $\mathbf{C}_{ss}$. For $N = 2$ and $K = 3$, examples of covariance matrices $\mathbf{C}_{rr}$ (red ellipses) corresponding to optimal solutions $\mathbf{r}$ of objective 2.12, for varying input covariance matrices $\mathbf{C}_{ss}$ (black ellipses) and frames $\mathbf{W}$ (spanning axes denoted by gray lines). Unit circles, which correspond to the identity matrix target covariance, are shown with dashed lines. Each row corresponds to a different frame $\mathbf{W}$ and each column corresponds to a different input covariance $\mathbf{C}_{ss}$.

A.8 Whitening Spatially Local Neighborhoods

A.8.1 Spatially local whitening in 1D

For an N-dimensional input, we consider a network that whitens spatially local neighborhoods of size $M < N$. To this end, we can construct N filters of the form

$$\mathbf{w}_i = \mathbf{e}_i, \qquad i = 1, \ldots, N$$

and $M(N - \frac{M+1}{2})$ filters of the form

$$\mathbf{w}_{ij} = \frac{\mathbf{e}_i + \mathbf{e}_j}{\sqrt{2}}, \qquad i, j = 1, \ldots, N, \qquad 1 \leqslant |i - j| \leqslant M.$$

The total number of filters is $(M+1)(N - \frac{M}{2})$, so for fixed M the number of filters scales linearly in N rather than quadratically.

We simulated a network comprising $N = 10$ primary neurons, and a convolutional weight matrix connecting each interneuron to spatial neighborhoods of three primary neurons. Given input data with covariance $\mathbf{C}_{ss}$ illustrated in Figure A.3A (left panel), this modified network succeeded to statistically whiten local neighborhoods of size of primary 3 neurons (right panel). Notably, the eigenspectrum (Figure A.3B) after local whitening is much closer to being equalized. Furthermore, while the global whitening solution produced a flat spectrum as expected, the local whitening network did not amplify the axis with very low-magnitude eigenvalues (Figure A.3B right panel).

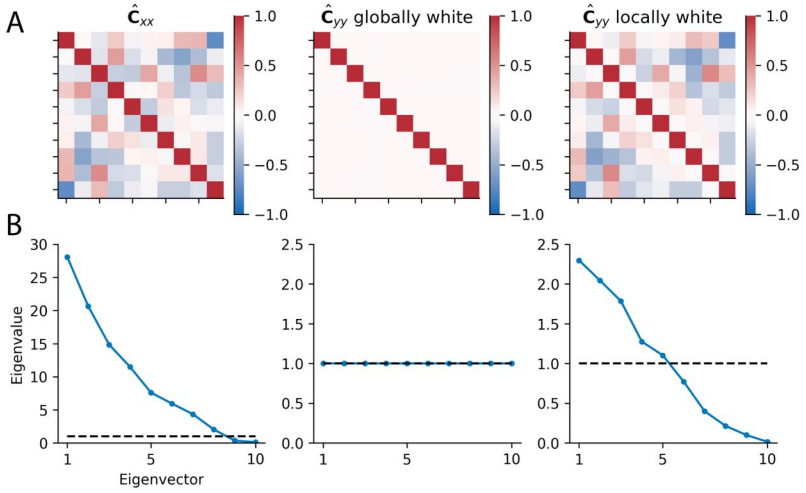

Figure A.3: Statistically adapting local neighborhoods of neurons. **A)** $\hat{\mathbf{C}}_{ss}$ denotes correlation matrix, which are shown here for display purposes only, to facilitate comparisons. Network with 10-dimensional input correlation (left) 10-dimensional output correlation matrix after global whitening (middle); and output correlation matrix after statistically whitening local neighborhoods of size 3. The output correlation matrix of the locally adapted circuit has block-identity structure along the diagonal. **B)** Corresponding eigenspectra of *covariance* matrices of unwhitened (left), global whitened (middle), and locally whitened (right) network outputs. The y-axis limits of the middle and right columns are the same, but different than the left column. The black dashed line denotes unity.

A.8.2 Filter bank construction in 2D

Here, we describe one way of constructing a set of convolutional weights for overlapping spatial neighborhoods (e.g. image patches) of neurons. Given an $n \times m$ input and overlapping

neighborhoods of size $h \times w$ to be statistically whitened, the samples are therefore matrices $X \in \mathbb{R}^{n \times m}$. In this case, filters $\mathbf{w} \in \mathbb{R}^{1 \times n \times m}$ can be indexed by pairs of pixels that are in the same patch:

$$((i,j),(k,\ell)), \qquad 1 \leqslant i \leqslant n, \qquad 1 \leqslant j \leqslant m, \qquad 0 \leqslant |i-k| \leqslant h, \qquad 0 \leqslant |j-\ell| \leqslant w$$

We can then construct the filters as,

$$\mathbf{w}_{(i,j),(k,\ell)}(X) = \begin{cases} x_{i,j} & \text{if } (i,j) = (k,\ell), \\ \frac{x_{i,j}+x_{k,\ell}}{\sqrt{2}} & \text{if } (i,j) \neq (k,\ell). \end{cases}$$

In this case there are

$$nm + wh\left[(n-w)(m-h) + (n-w)\frac{(h+1)}{2} + (m-h)\frac{(w+1)}{2} + (h+1)\frac{(w+1)}{2}\right]$$

such filters, so the number of filters required scales linearly with nm rather than quadratically.

A.9 Additional Applications

A.9.1 Preventing representational collapse in online principal subspace learning

Here, similar to Lipshutz et al. (2023), we show how whitening can prevent representational collapse using the analytically tractable example of online principal subspace learning. Recent approaches to self-supervised learning have used decorrelation transforms such as whitening to prevent collapse during training (e.g. Zbontar et al., 2021). Future architectures may benefit from online, adaptive whitening to allow for continual learning and test-time adaptation.

Consider a primary neuron whose *pre-synaptic* input at time t is $\mathbf{u}_t \in \mathbb{R}^D$, and corresponding post-synaptic input is $s_t := \mathbf{v}^\top \mathbf{u}_t$, where $\mathbf{v} \in \mathbb{R}^D$ are the synaptic weights connecting the pre-synaptic inputs to the neuron. An online variant of power iteration algorithm learns the top principal component of the inputs by updating the vector $\mathbf{v}$ as follows:

$$\mathbf{v} \leftarrow \mathrm{v} + \zeta \left(s_t \mathbf{u}_t - s_t^2 \mathbf{v} \right)$$
$$\mathbf{v} \leftarrow \frac{1}{\|\mathbf{v}\|} \mathbf{v}$$

where $\zeta > 0$ is small.

Next, consider a population of $2 \leq N \leq D$ primary neurons with outputs $\mathbf{r}_t \in \mathbb{R}^N$ and feedforward synaptic weight vectors $\mathbf{v}_1, \ldots, \mathbf{v}_N \in \mathbb{R}^D$ connecting the pre-synaptic inputs $\mathbf{u}_t$ to the N neurons. Running N parallel instances of the power iteration algorithm defined above *without* a decorrelation process results in representational collapse, because each synaptic weight vector $\mathbf{v}_i$ converges to the top principal component (Figure A.4, orange). We demonstrate that our whitening algorithm via gain modulation readily solves this problem. Here, it is important that the whitening happen on a faster timescale than the principal subspace learning, to avoid collapse (see Lipshutz et al., 2023, for details).

For this simulation, we set $D = 3, N = 2$ and randomly sample i.i.d. pre-synaptic inputs $\mathbf{u}_t \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}(5, 2, 1))$. We randomly initialize two vectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^3$ with i.i.d. Gaussian entries. At each time step t, we project pre-synaptic inputs to form the post-synaptic primary neuron inputs, $\mathbf{s}_t := \left[\mathbf{v}_1^\top \mathbf{u}_t, \mathbf{v}_2^\top \mathbf{u}_t \right]^\top$, forming the input to Algorithm 1. Let $\mathbf{r}_t$ be the primary neuron steady-state output; that is, $\mathbf{r}_t = \left(\mathbf{I}_N + \mathbf{W} \, \mathrm{diag}\left(\mathbf{g} \right) \mathbf{W}^\top \right)^{-1} \mathbf{s}_t$ (Equation 2.8). For $i = 1, 2$, we update $\mathbf{v}_i$ according to the above-defined update rules, with $\zeta = 10^{-3}$. We update the gains $\mathbf{g}$ according to Algorithm 1 with $\eta = 10\zeta$. To measure the online subspace learning performance,

we define

$$\text{Subspace error} := \left\| \mathbf{V} \left(\mathbf{V}^\top \mathbf{V} \right)^{-1} \mathbf{V}^\top - \text{diag}\left([1, 1, 0] \right) \right\|_{\text{Frob}}^2, \quad \mathbf{V} := [\mathbf{v}_1, \mathbf{v}_2] \in \mathbb{R}^{3\times 2}$$

Figure A.4 (blue) shows that our adaptive whitening algorithm with gain modulation successfully facilitates subspace learning and prevents representational collapse.

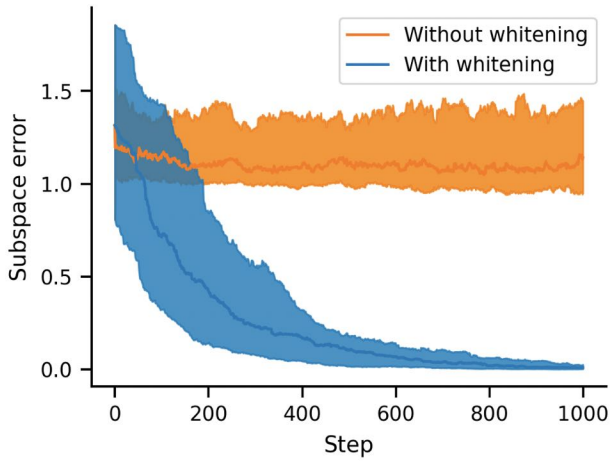

Figure A.4: Adaptive symmetric whitening with gain modulation prevents representational collapse during online principal subspace learning. Without whitening, subspace error stabilizes at a non-zero value, indicating that the network has converged to a collapsed representation. Shaded curves are median and [25%, 75%] quantiles over 50 random intializations.

A.9.2 Generalized adaptive covariance transformations

Our framework for adaptive whitening via gain modulation can easily be generalized to adaptively transform a signal with some initial covariance matrix to one with *any target covariance* (i.e. not just the identity matrix). This demonstrates that our adaptive gain modulation framework has implications beyond statistical whitening. This could, for example, allow online systems to stably maintain some initial/target (non-white) output covariance under changing input statistics (i.e. covariance homeostasis, Benucci et al., 2013; Westrick et al., 2016). The key insight, similar to the main text, is that a full-rank covariance matrix has K_N degrees of freedom, and therefore

marginal measurements along K_N distinct axes is necessary and sufficient to represent the matrix (Karl et al., 1994).

Let $\mathbf{C}_{\text{target}}$ be some arbitrary target covariance matrix. Then the general objective is

$$\min_{\{\mathbf{r}_t\}} \langle \|\mathbf{s}_t - \mathbf{r}_t\|_2^2 \rangle_t \quad \text{s.t.} \quad \langle \mathbf{r}_t \mathbf{r}_t^\top \rangle_t = \mathbf{C}_{\text{target}}. \tag{A.12}$$

Following the same logic as in the main text, the Lagrangian becomes

$$\begin{aligned} &\max_{\mathbf{g}} \min_{\{\mathbf{r}_t\}} \langle \ell(\mathbf{s}_t, \mathbf{r}_t, \mathbf{g}) \rangle_t, \\ &\text{where} \quad \ell(\mathbf{s}, \mathbf{r}, \mathbf{g}) := \|\mathbf{s} - \mathbf{r}\|_2^2 + \sum_{i=1}^{K} g_i \left\{ (\mathbf{w}_i^\top \mathbf{r})^2 - \sigma_i^2 \right\}, \end{aligned} \tag{A.13}$$

where $\sigma_i^2 = \mathbf{w}_i^\top \mathbf{C}_{\text{target}} \mathbf{w}_i$ is the marginal variance along the axis spanned by $\mathbf{w}_i$. When $\mathbf{C}_{\text{target}} = \mathbf{I}_N$, then $\sigma_i^2 = 1$ for all i, and this reduces to our original overcomplete whitening objective (Equation 2.5). The only difference in the recursive algorithm optimizing this generalized objective is the gain update rule,

$$\begin{aligned} g_i &\leftarrow g_i + \frac{\eta}{2} \nabla_{g_i} \ell(\mathbf{s}_t, \bar{\mathbf{r}}_t, \mathbf{g}) \\ &= g_i + \eta \left(\bar{z}_{i,t}^2 - \sigma_i^2 \right). \end{aligned} \tag{A.14}$$

We can interpret this formulation as each interneuron having a pre-determined target input variance (perhaps learned over long time-scales), and adjusting its gains to modulate the joint responses of the primary neurons until its input variance matches the target.

B Adaptive Whitening with Fast Gain Modulation and Slow Synaptic Plasticity

B.1 Separation of Timescales for Gain and Synaptic Weight Updates

In this section, we consider an algorithm where we directly optimize the objective in equation 3.3. In particular, for each context c, we first optimize over the gains $\mathbf{g}$ and then take a gradient descent step with respect to $\mathbf{W}$.

We first compute the gains using the formula for the optimal gains derived in (Duong et al., 2023c, equation 18):

$$\mathbf{g} = \left[(\mathbf{W}^\top \mathbf{W})^{\circ 2}\right]^\dagger \operatorname{diag}\left(\mathbf{W}^\top \mathbf{C}_{ss}^{1/2}(c)\mathbf{W} - \mathbf{W}^\top \mathbf{W}\right).$$

We then update the synaptic weights by taking the following gradient descent step:

$$\Delta \mathbf{w}_i = \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s}|c)}\left[\eta_w(\mathbf{r}\mathbf{r}^\top \mathbf{w}_i g_i - \mathbf{w}_i g_i)\right] = \eta_w\left(\mathbf{M}(\mathbf{W},\mathbf{g})^{-1}\mathbf{C}_{ss}(c)\mathbf{M}(\mathbf{W},\mathbf{g})^{-1} - \mathbf{I}_N\right)\mathbf{w}_i g_i,$$

where $\mathbf{M}(\mathbf{W}, \mathbf{g}) := \alpha \mathbf{I}_N + \mathbf{W}\mathrm{diag}(\mathbf{g})\mathbf{W}^\top$. Combining these updates yields Algorithm 5, which takes context-dependent covariance matrices $\mathbf{C}_{ss}(c)$ as its input.

Algorithm 5: Adaptive Whitening via Synaptic Plasticity and Gain Modulation

Input: Covariance matrices $\mathbf{C}_{ss}(1), \mathbf{C}_{ss}(2), \ldots$
Initialize: $\mathbf{W} \in \mathbb{R}^{N \times K}$; $\eta_w > 0$
for $c = 1, 2, \ldots$ **do**
  $\mathbf{g} \leftarrow \left[(\mathbf{W}^\top \mathbf{W})^{\circ 2}\right]^\dagger \mathrm{diag}\left(\mathbf{W}^\top \mathbf{C}_{ss}^{1/2}(c)\mathbf{W} - \mathbf{W}^\top \mathbf{W}\right)$
  $\mathbf{G} \leftarrow \mathrm{diag}(\mathbf{g})$
  $\mathbf{W} \leftarrow \mathbf{W} + \eta_w \left((\mathbf{W}\mathbf{G}\mathbf{W}^\top)^{-1} \mathbf{C}_{ss}(c) \, (\mathbf{W}\mathbf{G}\mathbf{W}^\top)^{-1} \mathbf{W}\mathbf{G} - \mathbf{W}\mathbf{G}\right)$
end for

B.2 Adaptive Whitening of Natural Images

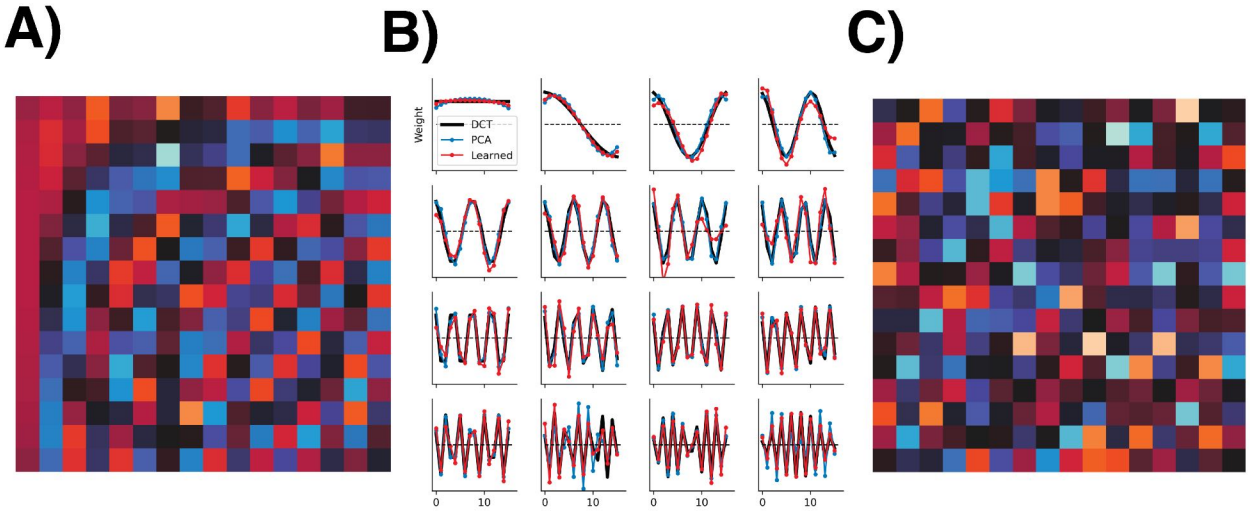

Figure B.1: Control experiment accompanying Sec. 3.6.2. **A)** $\mathbf{W}_T$ learned from natural image patches. **B)** Basis vectors from **A** displayed as line plots, compared to the 1D DCT, and principal components of $\mathbb{E}_{c\sim p(c)}[\mathbf{C}_{ss}(c)]$. **C)** Control condition. $\mathbf{W}_T$ learned from spectrally-matched image patches with random eigenvectors.

In this section, we elaborate on the converged structure of $\mathbf{W}_T$ using natural image patches. To better visualize the relationship between the learned columns of $\mathbf{W}$ and sinusoidal basis functions

(e.g. DCT), we focus on 1-dimensional image patches (rows of pixels). The results are similar with 2D image patches.

It is well known that eigenvectors of natural images are well-approximated by sinusoidal basis functions (e.g. the DCT; Ahmed et al., 1974; Bull and Zhang, 2021). Using the same images from the main text (van Hateren and van der Schaaf, 1998), we generated 56 contexts by sampling 16×1 pixel patches from separate images, with 2E4 samples each. We train Algorithm 5 with $K = N = 16$, $\eta_w = 5\text{E}{-2}$, and random $\mathbf{W}_0 \in O(16)$ on a training set of X of the images, presented uniformly at random $T = 1\text{E}5$ times. Fig B.1A,B shows that $\mathbf{W}_T$ approximates the principal components of the aggregated context-dependent covariance, $\mathbb{E}_{c\sim p(c)}[\mathbf{C}_{ss}(c)]$, which are closely aligned with the DCT. To show that this structure is inherent in the spatial statistics of natural images, we generated control contexts, $\mathbf{C}_{ss}(c)$, by forming covariance matrices with matching eigenspectra, but each with *random* and distinct eigenvectors. This destroys the structure induced by natural image statistics. Consequently, the learned vectors in $\mathbf{W}_T$ are no longer sinusoidal (Fig B.1C). As a result, whitening error with $\mathbf{W}_T$ is much higher on the training set, with 0.3 ± 0.02 error (mean ± standard error over 10 random initializations; Eq. 3.6) on natural image contexts and 2.7 ± 0.1 on the control contexts. While for the natural images, a basis approximating the DCT was sufficient to adaptively whiten all contexts in the ensemble, this is not the case for the generated control contexts.

Finally, we find that as K increases from $K = 1$ to $K = 16$, the basis vectors in $\mathbf{W}_T$ *progressively* learn higher frequency components of the DCT (Fig. B.2). This is a sensible solution, due to the ℓ_2 reconstruction error of our objective, and the $1/f$ spectral content of natural image statistics. With more flexibility, as K increases past N (i.e. the overcomplete regime), the network continues to improve its whitening error (Fig. B.3A) by learning a basis, $\mathbf{W}_T$, that can account for within-context information that is insufficiently captured by the DCT (Fig. B.3B). Taken together, our model successfully learns a basis $\mathbf{W}_T$ that exploits the spatial structure present in natural images.

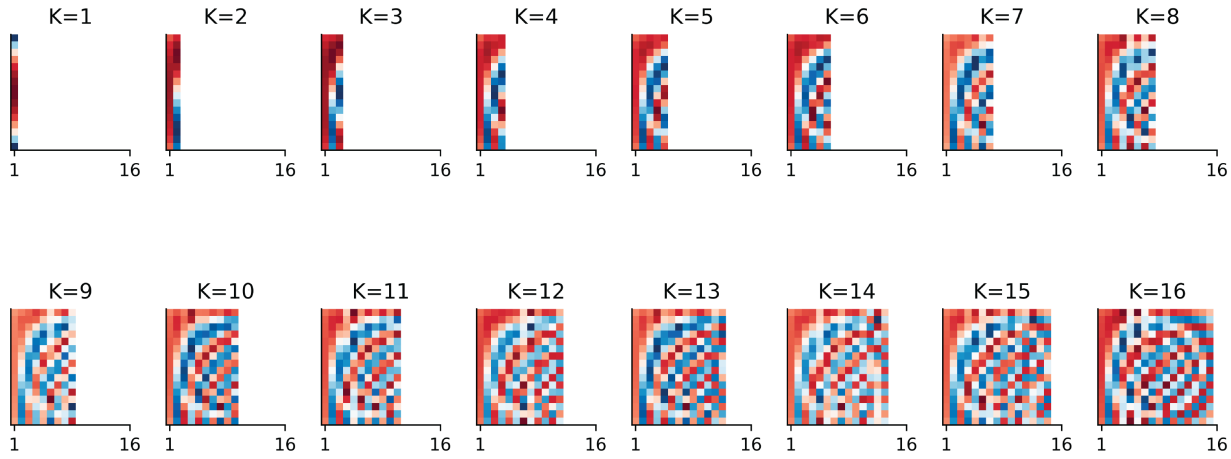

Figure B.2: As K increases, columns of $\mathbf{W}$ progressively learn higher frequency components of the DCT.

B.3 Modifications for Increased Biological Realism

In this section, we modify Algorithm 1 to be more biologically realistic.

B.3.1 Enforcing unit norm basis vectors

In our algorithm, there is no constraint on the magnitude of the column vectors of $\mathbf{W}$. We can enforce a unit norm (here measured using the Euclidean norm) constraint by adding Lagrange multipliers to the objective in equation 3.3:

$$\min_{\mathbf{W}\in\mathbb{R}^{N\times K}} \max_{\mathbf{m}\in\mathbb{R}^K} \mathbb{E}_{c\sim p(c)} \left[\min_{\mathbf{g}\in\mathbb{R}^K} \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s}|c)} \left[g\left(\mathbf{W}, \mathbf{g}, \mathbf{r}, \mathbf{s}\right) \right] \right], \tag{B.1}$$

where

$$g(\mathbf{W}, \mathbf{g}, \mathbf{r}, \mathbf{s}) = \ell(\mathbf{W}, \mathbf{g}, \mathbf{r}, \mathbf{s}) + \sum_{i=1}^{K} m_i(\|\mathbf{w}_i\|^2 - 1).$$

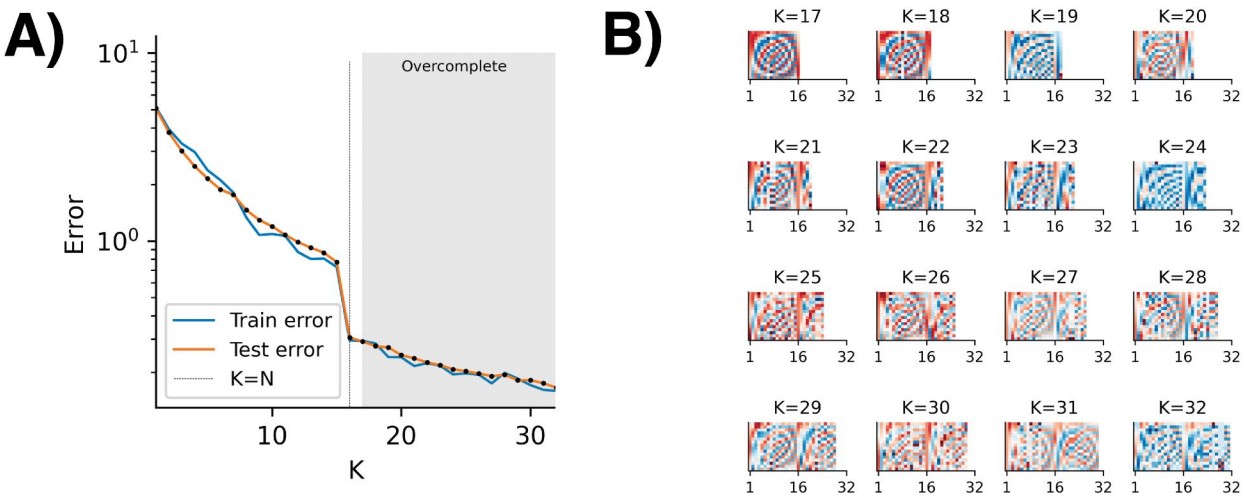

Figure B.3: **A)** Error on training and test set as a function of K. **B)** In the overcomplete regime, the network converges to a $\mathbf{W}_T$ that helps to improve error compared to the the $K \leq N$ regime.

Taking partial derivatives with respect to $\mathbf{w}_i$ and $\mathbf{m}_i$ results in the updates:

$$\Delta \mathbf{w}_i = \eta_w (n_i \mathbf{r} - (g_i + m_i)\mathbf{w}_i)$$

$$\Delta m_i = \|\mathbf{w}_i\|^2 - 1.$$

Furthermore, since the weights are constrained to have unit norm, we can replace $\|\mathbf{w}_i\|^2$ with 1 in the gain update:

$$\Delta g_i = \eta_g (z_i^2 - 1).$$

B.3.2 Decoupling the feedforward and feedback weights

We replace the primary neuron to interneuron weight matrix $\mathbf{W}^\top$ (resp. interneuron to primary neuron weight matrix $-\mathbf{W}$) with $\mathbf{W}_{rn}$ (resp. $-\mathbf{W}_{nr}$). In this case, the update rules are

$$\begin{aligned}\mathbf{W}_{rn} &\leftarrow \mathbf{W}_{rn} + \eta_w \left(\mathbf{n}_t \mathbf{r}_t^\top - \operatorname{diag}(\mathbf{g}+\mathbf{m})\mathbf{W}_{rn}\right)\\ \mathbf{W}_{nr} &\leftarrow \mathbf{W}_{nr} + \eta_w \left(\mathbf{r}_t \mathbf{n}_t^\top - \mathbf{W}_{nr}\operatorname{diag}(\mathbf{g}+\mathbf{m})\right).\end{aligned}$$

Let $\mathbf{W}_{rn,t}$ and $\mathbf{W}_{nr,t}$ denote the values of the weights $\mathbf{W}_{rn}$ and $\mathbf{W}_{nr}$, respectively, after $t = 0, 1, \ldots$ iterates. Then for all $t = 0, 1, \ldots,$

$$\mathbf{W}_{rn,t}^\top - \mathbf{W}_{nr,t} = \left(\mathbf{W}_{rn,0}^\top - \mathbf{W}_{nr,0}\right)\left(\mathbf{I}_N - \eta_w \operatorname{diag}(\mathbf{g}+\mathbf{m})\right)^t.$$

Thus, if $g_i + m_i \in (0, 2\eta_w^{-1})$ for all i (e.g., by enforcing non-negative g_i, m_i and choosing $\eta_w > 0$ sufficiently small), then the difference decays exponentially in t and the feedforward and feedback weights are asymptotically symmetric.

B.3.3 Sign-constraining the synaptic weights and gains

The synaptic weight matrix $\mathbf{W}$ and gains vector $\mathbf{g}$ are not sign-constrained in Algorithm 1, which is not consistent with biological evidence. We can modify the algorithm to enforce the sign constraints by rectifying the weights and gains at each step. Here $[\cdot]_+$ denote the elementwise rectification operation. This results in the updates

$$\begin{aligned}\mathbf{g} &\leftarrow \left[\mathbf{g} + \eta_g(\mathbf{z}\circ\mathbf{z} - \mathbf{1})\right]_+\\ \mathbf{W}_{rn} &\leftarrow \left[\mathbf{W}_{rn} + \eta_w \left(\mathbf{n}_t \mathbf{r}_t^\top - \operatorname{diag}(\mathbf{g}+\mathbf{m})\mathbf{W}_{rn}\right)\right]_+\\ \mathbf{W}_{nr} &\leftarrow \left[\mathbf{W}_{nr} + \eta_w \left(\mathbf{r}_t \mathbf{n}_t^\top - \mathbf{W}_{nr}\operatorname{diag}(\mathbf{g}+\mathbf{m})\right)\right]_+.\end{aligned}$$

B.3.4 Online algorithm with improved biological realism

Combining these modifications yields our more biologically realistic multi-timescale online algorithm, Algorithm 6.

Algorithm 6: Biologically realistic multi-timescale adaptive whitening

Input: $\mathbf{s}_1, \mathbf{s}_2, \dots \in \mathbb{R}^N$
Initialize: $\mathbf{W}_{nr} \in \mathbb{R}^{N \times K}$; $\mathbf{W}_{rn} \in \mathbb{R}^{K \times N}$; $\mathbf{m}, \mathbf{g} \in \mathbb{R}^K$; $\eta_r, \eta_m > 0$; $\eta_g \gg \eta_w > 0$
for $t = 1, 2, \dots$ **do**
  $\mathbf{r}_t \leftarrow \mathbf{0}$
  while not converged **do**
    $\mathbf{z}_t \leftarrow \mathbf{W}_{rn}\mathbf{r}_t$
    $\mathbf{n}_t \leftarrow \mathbf{g} \circ \mathbf{z}_t$
    $\mathbf{r}_t \leftarrow \mathbf{r}_t + \eta_r \left(\mathbf{s}_t - \mathbf{W}_{nr}\mathbf{n}_t - \alpha \mathbf{r}_t\right)$
  end while
  $\mathbf{m} \leftarrow \left[\mathbf{m} + \eta_m(\mathrm{diag}(\mathbf{W}_{rn}\mathbf{W}_{nr}) - \mathbf{1})\right]_+$
  $\mathbf{g} \leftarrow \left[\mathbf{g} + \eta_g \left(\mathbf{z}_t \circ \mathbf{z}_t - \mathbf{1}\right)\right]_+$
  $\mathbf{W}_{rn} \leftarrow \left[\mathbf{W}_{rn} + \eta_w \left(\mathbf{n}_t \mathbf{r}_t^\top - \mathrm{diag}(\mathbf{g} + \mathbf{m})\mathbf{W}_{rn}\right)\right]_+$
  $\mathbf{W}_{nr} \leftarrow \left[\mathbf{W}_{nr} + \eta_w \left(\mathbf{r}_t \mathbf{n}_t^\top - \mathbf{W}_{nr}\mathrm{diag}(\mathbf{g} + \mathbf{m})\right)\right]_+$
end for

C | Propagating Single Neuron Gains Through Recurrent Circuitry

C.1 Details on Model Recurrent Connectivity Matrix

C.1.1 Initializing the recurrent connectivity matrix

We restrict $\mathbf{W} \in \mathbb{R}^{N \times N}$ to the space of circularly symmetric (i.e. convolutional) positive definite matrices. In our model, the recurrent weight kernel forming the convolutional matrix is (net) positive everywhere, with higher probability between similarly tuned excitatory neurons than between dissimilarly tuned neurons (Ko et al., 2011; Lee et al., 2016). The kernel we use is a Gaussian (10° FWHM) summed with a uniform density. To prevent recurrence from diverging (Eq. 4.3), the operator norm of $\mathbf{W}$ (i.e. the max eigenvalue) must be less than 1. We fixed $\|\mathbf{W}\|_{\text{Op}} = 0.8$ for all recurrent weight matrices in this study.

C.1.2 Recurrent synaptic connectivity influences adaptation effects

The structure of the recurrent weight matrix $\mathbf{W}$ greatly impacts the adaptive changes in neural responses. Specifically, we find that, with our recurrent model and objective (Eq. 4.3 and Eq. 4.4), weak net excitatory inputs from dissimilarly tuned neurons is needed to capture the observed effects in data. This recurrent composition is line with broad, untuned excitatatory signal ampli-

fication contributing to overall background activity in cortical circuits (Reinhold et al., 2015). For reference, we reproduce a subset of the post-adaptation tuning curves from the main text here in Fig. C.1A.

Figure C.1B shows response curves from an example model using a convolutional $\mathbf{W}$ derived from the Mexican hat weight kernel, which is an excitatory Gaussian (10° FWHM) minus a wider Gaussian (60° FWHM) (Carandini and Ringach, 1997; Quiroga et al., 2016; Teich and Qian, 2010). We re-scaled the weight matrix to have operator norm of 0.8. The responses in Fig. C.1B here are from a model minimizing ℓ_2 error between the data and model adapted responses after a hyper-parameter sweep (α = 3E-4, γ = 2E-3). In contrast to the recurrent weight matrix we use in the main text (described above), the Mexican hat kernel has recurrent net excitation from neurons with similar tuning, and *net inhibition* from neurons with dissimilar tuning. A model with this recurrent weight kernel is unable to capture the effects of response maxima, minima, and amplitudes, and produces tuning curve *attraction* rather than repulsion from the adapter Fig. C.2A-D. Taken together, this suggests that broad, untuned weak recurrent excitation is necessary for our model to capture the wide array of post-adaptation effects found in this dataset.

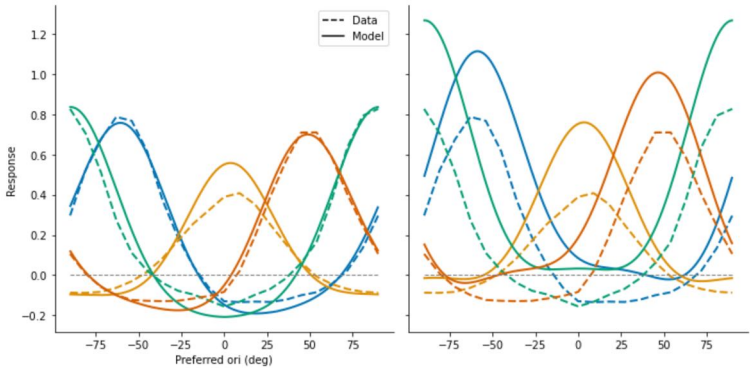

Figure C.1: Data (dashed) vs model (solid) with different forms of $\mathbf{W}$. Dashed lines (identical in left and right panels) are observed post-adaptation response curves for a subset of the population. **Left** Model from main text, using a convolutional $\mathbf{W}$ with recurrent net excitation from similarly tuned neurons, and broad/untuned net excitation from dissimilarly tuned neurons. **Right** Simulated post-adaptation responses from a model with $\mathbf{W}$ comprising recurrent net excitation from similarly tuned neurons, and net *inhibition* from dissimilarly tuned neurons.

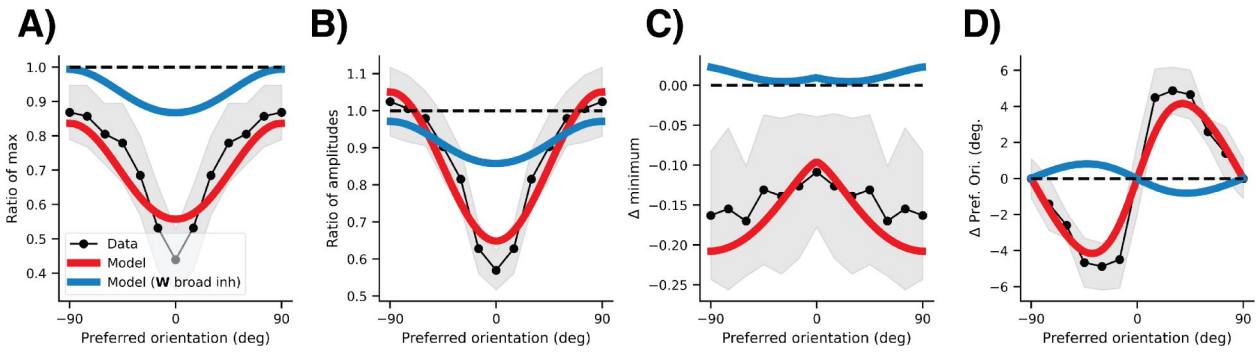

Figure C.2: All panels are the same as Fig. 4.3 in the main text, but blue is now a model with **W** set to a convolutional matrix with a Mexican hat kernel. Notably, this model cannot reproduce the adaptive maxima, minima, and amplitude effects observed in data. Furthermore, the tuning curves are no longer repelled from the adapter (panel **D)**, but are instead *attracted* toward it.

C.2 Objective Ablation

Here, we assess the contribution of each term of the objective (Equation 4.4) to explain the adaptation effects found in data.

C.2.1 Gain homeostasis confers representation stability with adaptation

Fig. C.3 shows the impact of removing the gain homeostasis term from Eq 4.4. Gain homeostasis prevents the network from drastically re-configuring its representation after adaptation, and allows the network to maintain a stable representation of the stimulus ensemble (compare orange to green).

C.2.2 contributions of each term to adaptation

We assess the importance of the three terms in the objective (Eq. 4.4) and show that they are all jointly necessary to produce the effects shown in main text. Figure C.4 shows the adapted model responses using Eq. 4.4 to adapt in red. Without the gain homeostasis term (i.e. $\gamma = 0$, green), the

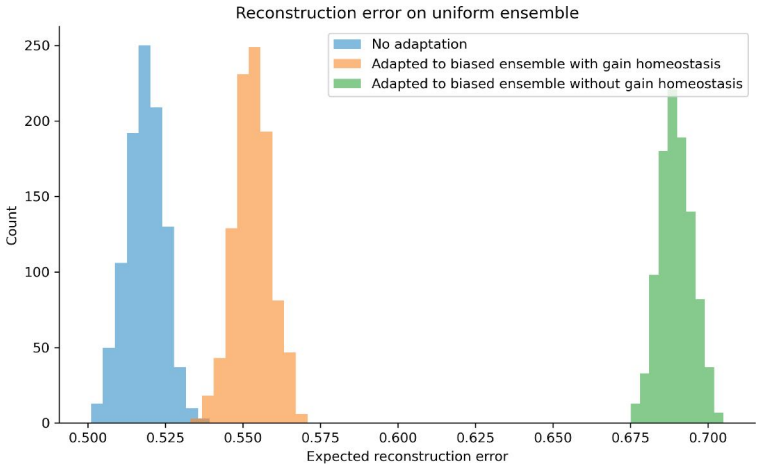

Figure C.3: Gain homeostasis induces stability across statistical contexts. Histograms are bootstrap samples (1000 repeats) of the average stimulus reconstruction error under the uniform stimulus ensemble without adaptation, after adaptation with gain homeostasis, and after adaptation without gain homeostasis.

gains radically change after adapting to the biased stimulus ensemble. This produces higher responses in neurons tuned for orientations along the flank (far from the adapter at zero degrees), and completely fails to reproduce any of the adaptation effects observed in data. Without the activity penalty (i.e. $\alpha = 0$, blue), the model's responses are equivalent to one with no adaptation. Finally, without the ℓ_2 reconstruction penalty (first term of objective; purple), the maxima and minima undershoot what is observed in data; however, the model does reasonably well at capturing the shifts in tuning preference and response amplitude. The reconstruction term of the objective encourages the network to maintain a high fidelity representation of the stimulus after adaptation. This ablation finding suggests that the shifts in tuning preference observed in many previous studies (Clifford et al., 2007) may arise from adaptive sensory information-preserving properties of the system. Taken together, each component of the objective works in concert to yield the adaptation response phenomena seen in the data.

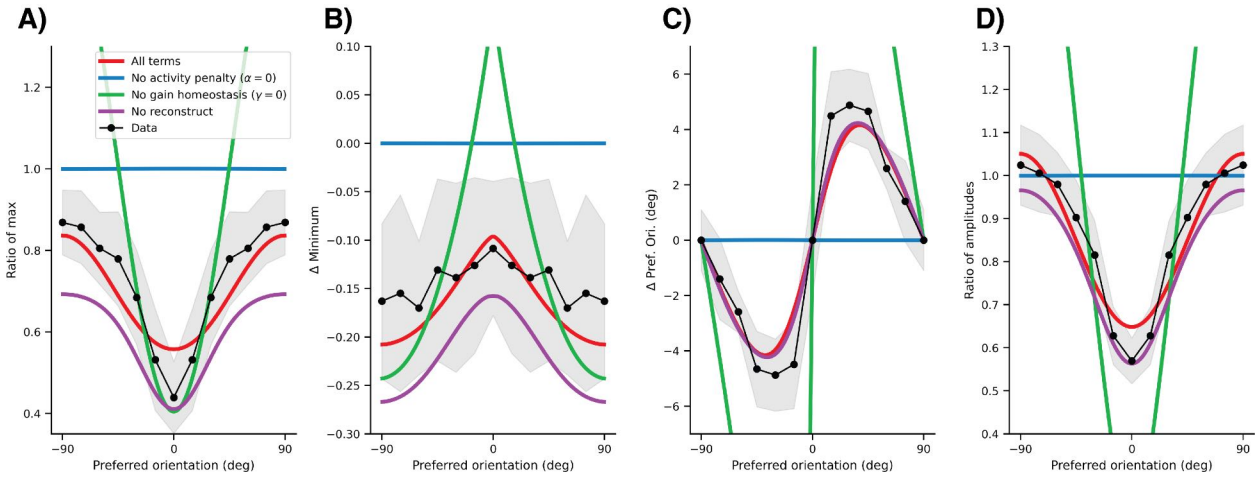

Figure C.4: Each term in the objective (Equation 4.4) is necessary to account for the full array of adaptation effects observed in data.

C.3 Analytic Solution to the Adaptation Objective

From Eq. 4.3, the steady state response of a network is given by

$$\begin{aligned} \mathbf{r}_*(\mathbf{s}, \mathbf{g}) &= [\mathbf{I} - \mathbf{W}]^{-1} (\mathbf{g} \circ \mathbf{f}(\mathbf{s})) \\ \mathbf{r}_*(\mathbf{s}, \mathbf{g}) &= \mathbf{M} (\mathbf{g} \circ \mathbf{f}(\mathbf{s})) \end{aligned} \tag{C.1}$$

where $\mathbf{W} \prec \mathbf{I}$, and $\mathbf{M} \coloneqq [\mathbf{I} - \mathbf{W}]^{-1}$ is a matrix capturing the effect of leak and lateral recurrence in the network. The feedforward and recurrent weights of our model are assumed to be fixed through adaptation. We can isolate $\mathbf{g}$ using the identity $\text{diag}(\mathbf{a})\,\mathbf{b} = \text{diag}(\mathbf{b})\,\mathbf{a}$, for two vectors $\mathbf{a}$ and $\mathbf{b}$, to get:

$$\begin{aligned} \mathbf{M} (\mathbf{g} \circ \mathbf{f}(\mathbf{s})) &= \mathbf{M}\,\text{diag}(\mathbf{f}(\mathbf{s}))\,\mathbf{g} \\ &\equiv \mathbf{H}(\mathbf{s})\mathbf{g}, \end{aligned}$$

where we define $\mathbf{H} : \mathbb{R}^N \mapsto \mathbb{R}^{N\times N}$ as a linear operator that maps $\mathbf{s}$ to a matrix using $\mathbf{M}$ and $\mathbf{f}(\mathbf{s})$.

In the main text, the loss functional (Eq. 4.4) omitted dependence on the decoder $\mathbf{D}$ for clarity. The full objective is

$$\mathcal{L}(p(\mathbf{s}), \mathbf{g}, \mathbf{D}) = \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}\left[\underbrace{\| \mathbf{s} - \mathbf{D}^\top \mathbf{H}(\mathbf{s})\mathbf{g} \|_2^2}_{\text{reconstruction}} + \alpha \underbrace{\| \mathbf{H}(\mathbf{s})\mathbf{g} \|_2^2}_{\text{activation}}\right] + \gamma \underbrace{\| \mathbf{g} - \mathbf{g}_0 \|_2^2}_{\text{homeostatic gain}} + \delta \underbrace{\| \mathbf{D} \|_{\mathrm{F}}^2}_{\text{decoder}}, \tag{C.2}$$

where δ is a hyperparameter controlling the decoder weights $\mathbf{D}$. The results in the main text have δ set to zero, and our findings do not qualitatively change with small deviations away fro $\delta = 0$. This objective is bi-convex in $\mathbf{D}$ and $\mathbf{g}$ (i.e. convex when one of the two variables is held fixed). Indeed, with $\mathbf{g}$ fixed, the loss simply becomes an ℓ_2-regularized least-squares problem in $\mathbf{D}$. One can show that the linear decoder regularization term, δ, is equivalent to assuming noisy outputs with additive isotropic Gaussian noise. We solve for each optimization variable in alternation until they reach convergence. As our stimulus ensemble comprises a discrete set of K stimuli, we can write our objective explicitly as a weighted summation,

$$\mathcal{L}(\mathbf{g}) = \frac{1}{2}\sum_{k=1}^{K} p(\mathbf{s}_k)\left\{\| \mathbf{s}_k - \mathbf{D}^\top \mathbf{H}(\mathbf{s}_k)\mathbf{g}_s \|_2^2 + \alpha \| \mathbf{H}(\mathbf{s}_k)\mathbf{g} \|_2^2\right\} + \gamma \| \mathbf{g} - \mathbf{g}_0 \|_2^2 . \tag{C.3}$$

Computing $\nabla\mathcal{L}_{\mathbf{g}} = 0$, and isolating for $\mathbf{g}$ yields a linear system of equations,

$$\left[\sum_k p(\mathbf{s}_k)\left\{\mathbf{H}(\mathbf{s}_k)^\top\left(\mathbf{D}\mathbf{D}^\top + \alpha\mathbf{I}\right)\mathbf{H}(\mathbf{s}_k)\right\} + \gamma\mathbf{I}\right]\mathbf{g} = \left[\sum_k p(\mathbf{s}_k)\left\{\mathbf{H}(\mathbf{s}_k)^\top\mathbf{D}\hat{\mathbf{s}}_k\right\}\right] + \gamma\mathbf{g}_0. \tag{C.4}$$

This is in the form of $\mathbf{A}\mathbf{g} = \mathbf{b}$ and can therefore be solved exactly. A similar derivation can be done for the optimal $\mathbf{D}$. To initialize $\mathbf{g}_0$ and $\mathbf{D}$, we alternated optimization between $\mathbf{D}$ and $\mathbf{g}$ (using the control context stimulus ensemble) until convergence using co-ordinate descent.

C.4 Alternative Models

C.4.1 Equivalent circuit with adaptive recurrent gain

Here, we explore an alternative network parameterization which has identical steady-state behavior as the network we have selected for our model: consequently, adaptation under our training procedure will have identical behavior at a network level for both parameterizations. The dynamics of our network, replicated from Equation 4.2 for clarity, are given by

$$\begin{aligned}\frac{d\mathbf{r}(\mathbf{s},\mathbf{g})}{dt} &= -\mathbf{r} + \mathbf{W}\mathbf{r} + \mathbf{g} \circ \mathbf{f}(\mathbf{s}), \\ &= [-\mathbf{I} + \mathbf{W}]\mathbf{r} + \mathbf{g} \circ \mathbf{f}(\mathbf{s}),\end{aligned} \tag{C.5}$$

where we denote both the leak term,$-\mathbf{I}\mathbf{r}$, and the recurrent weight term, $\mathbf{W}\mathbf{r}$, as the recurrent drive. As an alternative to this parameterization, consider instead multiplicatively scaling each neuron's recurrent drive with $\mathbf{g} \in \mathbb{R}_+^N$,

$$\frac{d\mathbf{r}(\mathbf{s},\mathbf{g})}{dt} = \mathbf{g}^{-1} \circ [-\mathbf{I} + \mathbf{W}]\mathbf{r} + \mathbf{f}(\mathbf{s}), \tag{C.6}$$

where $\mathbf{g}^{-1} = [1/g_1, 1/g_2, ..., 1/g_N]^\top$. Intuitively, Equation C.6 states that an increase in each neuron's g_i attenuates its recurrent drive. Gain changes in this model adjust the overall sensitivity to recurrent (including self-recurrence, i.e. leak) drive. To solve for this new network's steady-state, $\mathbf{r}_*(\mathbf{s},\mathbf{g})$, we set Equation C.6 to zero and solve,

$$\mathbf{g}^{-1} \circ [\mathbf{I} - \mathbf{W}] \left(\mathbf{r}_*(\mathbf{s}, \mathbf{g})\right) = \mathbf{f}(\mathbf{s}) \tag{C.7}$$

$$\mathbf{r}_*(\mathbf{s}, \mathbf{g}) = \left[\text{diag}\left(\mathbf{g}^{-1}\right) [\mathbf{I} - \mathbf{W}]\right]^{-1} \mathbf{f}(\mathbf{s}) \tag{C.8}$$

$$\mathbf{r}_*(\mathbf{s}, \mathbf{g}) = [\mathbf{I} - \mathbf{W}]^{-1} \left(\mathbf{g} \circ \mathbf{f}(\mathbf{s})\right),$$

which is the same as our steady-state equation given by Equation 4.3 in the main text. Therefore, our original formulation of multiplicatively scaling the network's feedforward drive is mathematically equivalent to inversely scaling (attenuating) its recurrent drive. This means that upscaling gain on feedforward inputs is equivalent to downscaling net inhibition on each neuron.

C.4.2 Equivalent circuit with two-layer feedforward architecture

The overall action of the network at steady-state (Equation 4.3) is to mix the gain-modulated feedforward responses, $\mathbf{g} \circ \mathbf{f}(\mathbf{s})$, by a linear transformation dependent on the recurrent circuitry $[\mathbf{I}-\mathbf{W}]^{-1}$. The steady-state response of this recurrent network is *equivalent* to a two-layer feedforward network with gain modulation after the first layer. This means that viewing the steady-state responses alone, as is frequently done in neurophysiological adaptation experiments (Weber et al., 2019), it is impossible to tell whether the system was exclusively feedforward or recurrent. This is well-known (Dayan and Abbott, 2005) and is a fundamental result in signal processing theory of linear feedback systems. Our model can be interpreted as a cascade of transformations, propagating adaptive response changes downstream (Dhruv and Carandini, 2014; Kohn and Movshon, 2003).

C.4.3 Relationship to divisive normalization

Divisive normalization is a canonical computation reported across species, sensory modalities, and brain areas (Carandini and Heeger, 2012; Duong et al., 2019). Our model is linear and

not divisive. Writing the steady-state for the i^{th} neuron explicitly (omitting $\mathbf{g}$ to reduce clutter) yields

$$
\begin{aligned}
r_i &= f_i(\mathbf{s}) + \sum_{j=1}^{N} w_j r_j \\
r_i &= f_i(\mathbf{s}) + \sum_{j \neq i} w_j r_j + w_i r_i \\
(1 - w_i) r_i &= f_i(\mathbf{s}) + \sum_{j \neq i} w_j r_j \\
r_i &= \frac{1}{1 - w_i} \left[\sum_{j} f_j(\mathbf{s}) + \sum_{j \neq i} w_j r_j \right].
\end{aligned}
$$

Thus, there is no nonlinear interaction between r_i and other neurons in the population.

D | Stochastic Shape Metrics

D.1 Supplementary Figures

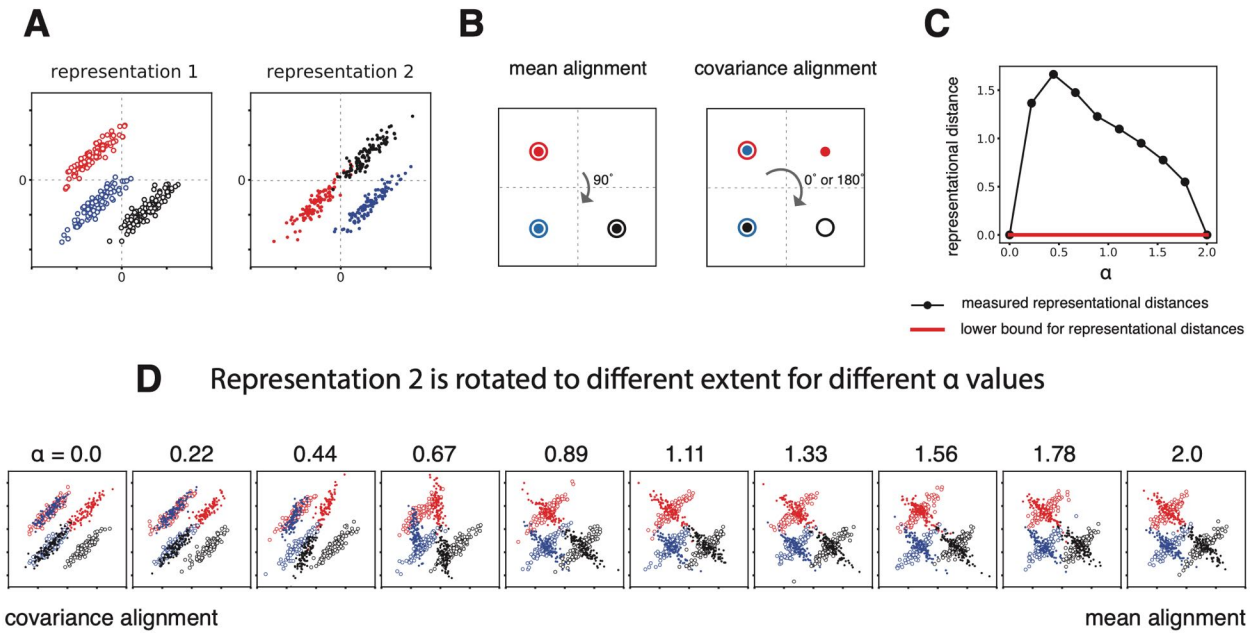

Figure D.1: Simulated example showing how varying the α parameter in $\overline{\mathcal{W}}_2^{\alpha}$ induces different rotational alignments between neural representations ($\mathcal{G} = \mathcal{O}$). *(A)* Two stimulated stochastic representations for three stimulus inputs. Colors represent different input conditions ($M = 3$), hollow points represent sampled representations from the first network and filled points represent sampled representations from the second network. The example is constructed so that no rotation can simultaneously align both the means and covariances. *(B)* If the stochastic metric only takes means into account ($\alpha = 2$), after rotating one of the representations by $90°$, two sets of representational means completely overlap, and the distance becomes 0. If the stochastic metric only takes covariances into account ($\alpha = 0$), the optimal alignment between the two sets of covariances is either $0°$ or $180°$, and after this rotation, distance between representations again is 0. *(C)* When both $\alpha = 0$ and $\alpha = 2$, distance between the two representations is 0, so the lower bound for the distance for α in the range between 0 and 2 is also 0. We computed the stochastic metric within this range of α, and the final distance is generally above the lower bound. *(D)* Optimal rotation between the two representations at different values of α.

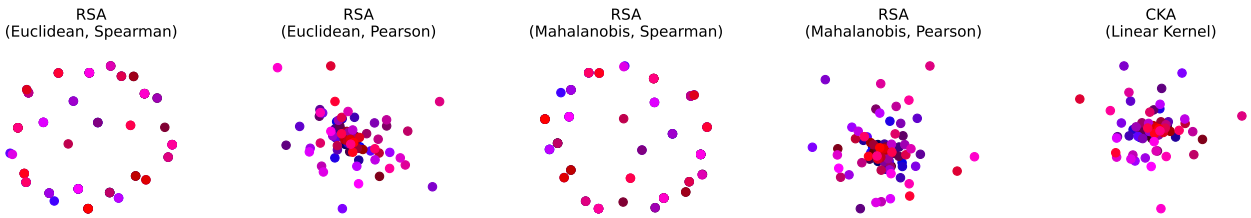

Figure D.2: Associated with Figure 5.4 from the main text. Embeddings of "toy dataset" networks (see Fig. 5.4A-B) visualized by multi-dimensional scaling of existing dissimilarity measures. Each point represents a network, the color scheme is the same as in Fig. 5.4C. All methods fail to recover a reasonable embedding which captures representational differences (compare with stochastic shape metric embeddings in Fig. 5.4C and Supp. Fig. D.3C). Starting from the left, the first two plots use representational similarity analysis (RSA; Kriegeskorte et al. 2008a) with two forms of correlation distance (Spearman and Pearson) applied to Euclidean representational similarity matrices. The next two plots use Mahalanobis distance re-weighted by the noise covariance (Walther et al., 2016) rather than Euclidean distance. The final plot shows an embedding by centered kernel alignment with a linear filter (Kornblith et al., 2019).

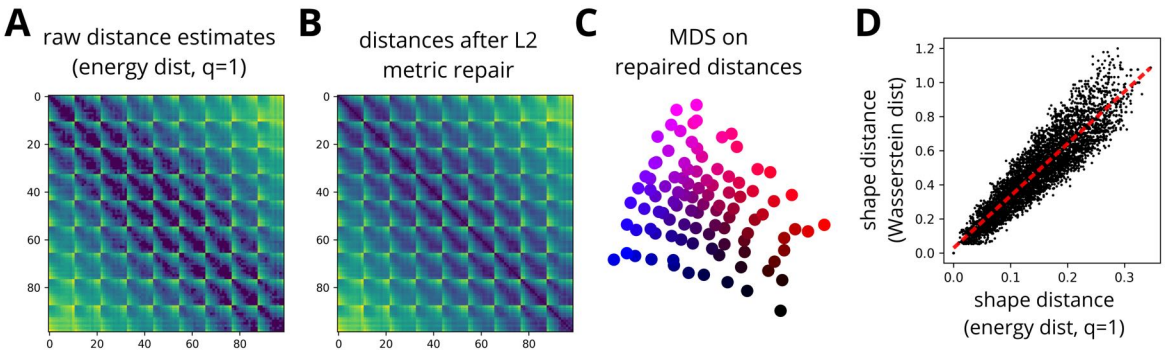

Figure D.3: Associated with Figure 5.4 from the main text. Stochastic shape metrics with energy distance also recover the "ground truth" structure of synthetic "toy data". *(A)* Matrix of estimated pairwise distances computed with $\mathcal{E}_1$ ground metric on the synthetic data shown in Fig. 5.4A. *(B)* Matrix of pairwise distances after quadratic metric repair (see subsection D.6.3) was performed to correct for minor triangle inequality violations. *(C)* Multidimensional scaling embedding of the distance matrix in panel B into 2D Euclidean space. Compare with Fig. 5.4C. *(D)* Linear correlation between stochastic shape distances with energy distance ground metric (i.e. off-diagonal entries of panel B) and 2-Wasserstein ground metric (i.e. off-diagonal entries of Fig. 5.4B). Red dashed line denotes the best linear model according to a least-squares criterion.

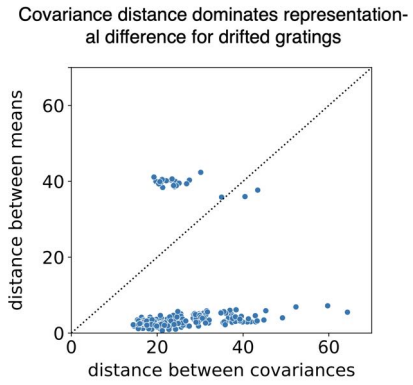

Figure D.4: Associated with Figure 5.5 from the main text. Drifted gratings (4 drifting directions, 75 repeats each) were presented in a different set of experimental sessions. Like (artificial) static gratings, representational distances across sessions for drifted gratings are dominated by covariance differences.

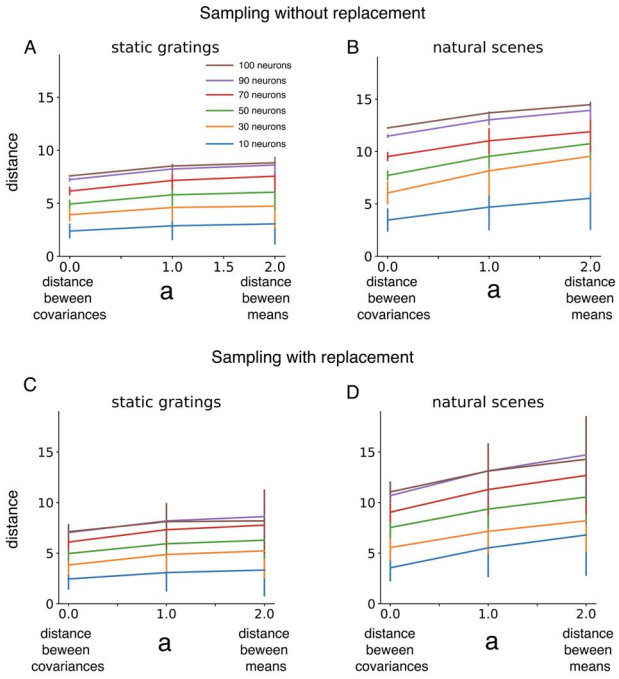

Figure D.5: We use a simulation to explore how the size of the neural population recording affects our conclusions about representational distances. In particular, does the ratio of mean-insensitive to covariance-insensitive across-animal distances ($\alpha = 0$ vs. $\alpha = 2$) change when we sub-sample neurons? For this simulation, we chose two mice that have 102 and 110 neurons recorded from their respective VISps. We randomly sample a subset of n neurons among these recorded neurons ($n = 10, 30, 50, 70, 90, 100$), and computed representational distance ($\alpha = 0, 1, 2$) using only the subset. For panel A and B, we sampled the neurons without replacement, and for panel C and D, we sample with replacement (bootstrapping). We observe that for all tested α, representational distance increases with the number of neurons within the subset. This is expected because distances will generically increase with the dimension (e.g. the Euclidean distance between two random vectors in a high dimensions will tend to be large, relative to low dimensions). However, the ratio of $\alpha = 0$ and $\alpha = 2$ shape distances is preserved when subsampling neurons (all lines are trending upward as a function of α). Error bars capture how the computed distances vary across 15 random draws of n neurons from the recorded population.

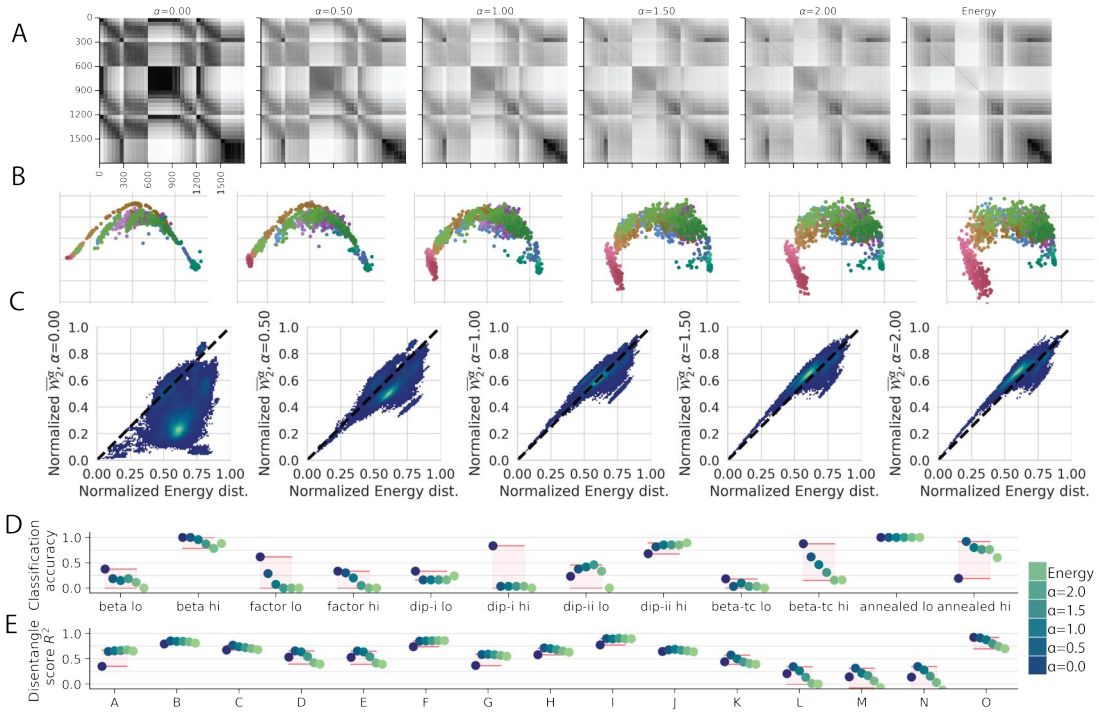

Figure D.6: Associated with VAE analyses (Figure 5.6) in the main text. *(A)* Dissimilarity matrices measured for 1800 dSprites-trained VAEs from Locatello et al. (2019) using generalized interpolated 2-Wasserstein (Equation 5.7) with varying α (first five columns), and using energy distance (Equation 5.4) with 64 samples for each unique input (right-most column). Row/column ordering of each matrix is the same as in Figure 5.6. *(B)* 2D embeddings corresponding to distance matrices in *(A)*. Colors are the same as in Figure 5.6. We aligned to the left-most panel using Procrustes analysis, allowing for scaling and rotations/reflections. *(C)* Re-scaling $\overline{\mathcal{W}}_2^{\alpha}$ distances and energy distances such that they lie between $[0, 1]$ reveals that the distribution of energy distances agrees best with $\overline{\mathcal{W}}_2^{\alpha=1.0}$ (middle column). *(D)* Predicting objective and regularization strength using distance matrices in *(A)*. *(E)* Predicting disentanglement scores using distance matrices in *(A)*. See Supp. D.2.3 for more details.

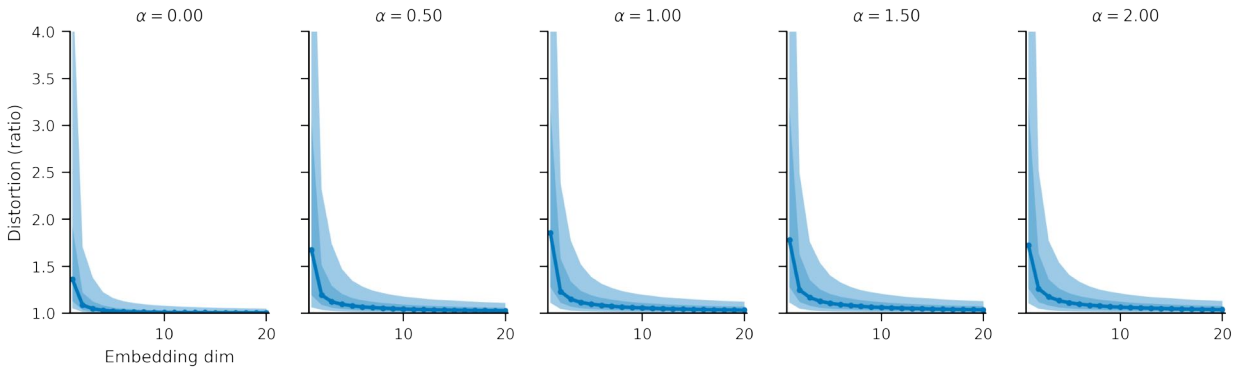

Figure D.7: Associated with VAE analyses (Figure 5.6) in the main text. Distortion induced by multidimensional scaling of 1800×1800 dissimilarity matrices with varying embedding dimensionality. Different shading represents (10th-90th) and (25th-75th) percentiles. See Supp. D.2.3 for details.

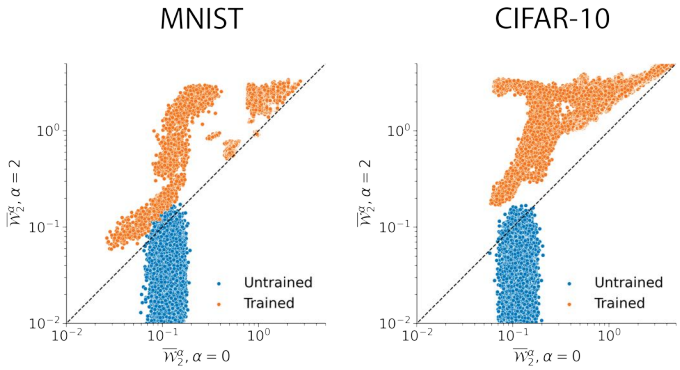

Figure D.8: Associated with VAE analyses (Figure 5.6) in the main text. We initialized 350 β-VAEs and trained them on MNIST (left) and CIFAR-10 (right) with different values of β in the loss function. Training led to inter-network distances being dominated by covariance-insensitive ($\alpha = 2$) dissimilarity, in agreement with Figure 5.6B of the main text. See Supp. D.2.3 for training details.

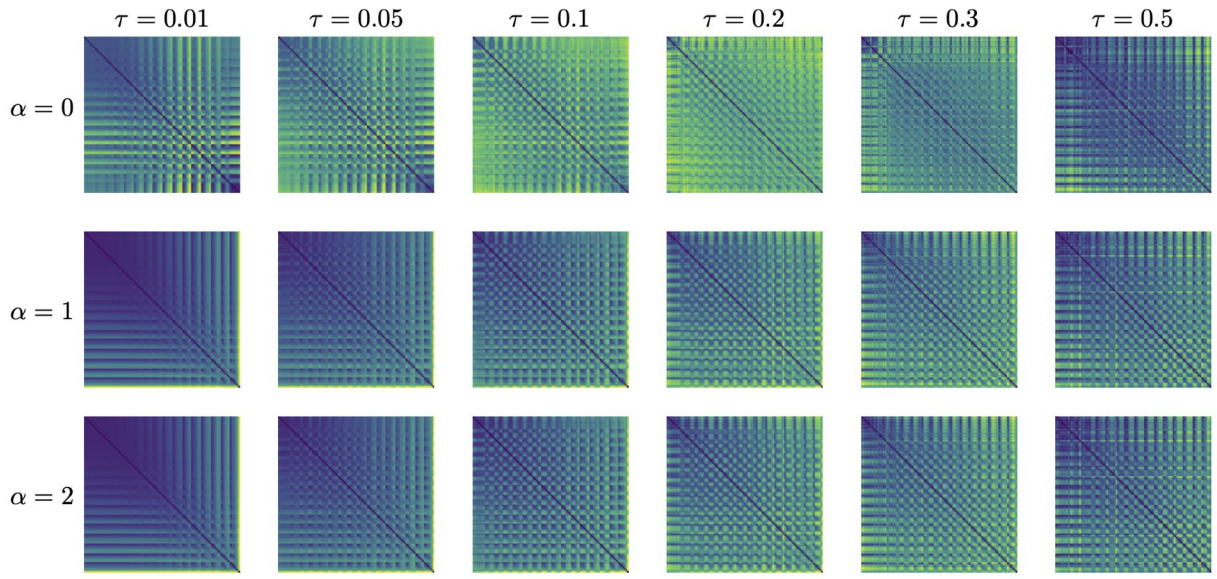

Figure D.9: Distance matrices for different values of α and τ.

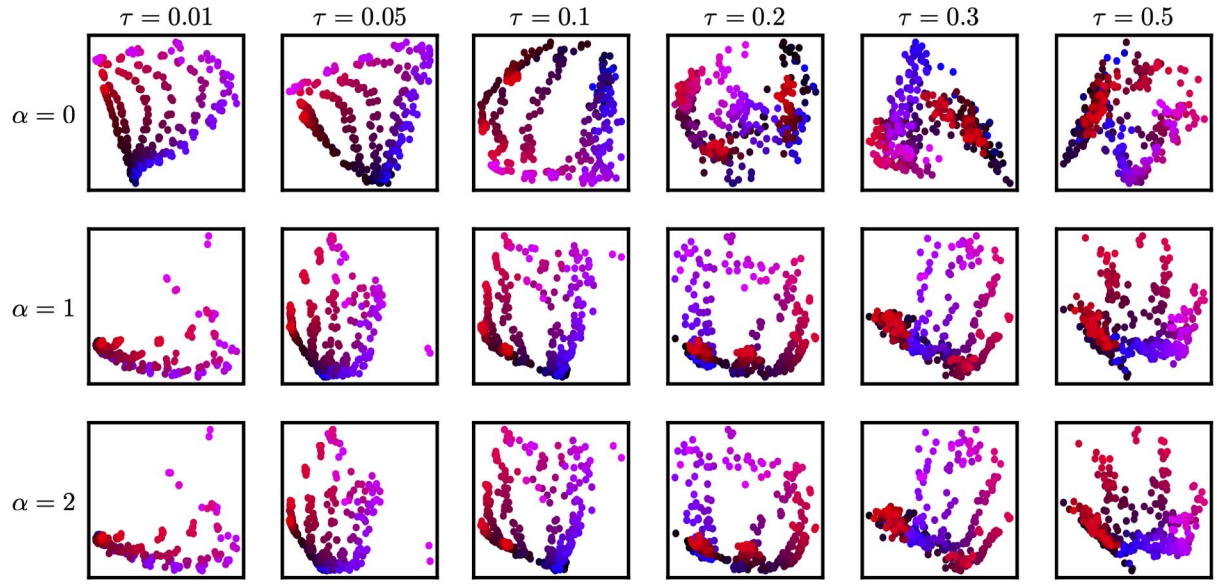

Figure D.10: Two dimensional embedding of the distance matrices in Fig. D.9 for different values of α (row) and τ (column).

D.2 Supplemental Methods

D.2.1 Code and reproducibility

Additional analysis and source code will be located at github.com/ahwillia/netrep.

D.2.2 Allen Brain Observatory Data

D.2.2.1 Data pre-processing.

In each recording session, gratings (6 orientations) and natural scenes (119 images) were presented to one mouse, and between 19 to 110 neurons in VISp were recorded by extracellular microelectrode arrays (Neuropixels Visual Coding dataset). Each neuron's response to an image was measured as the sum of action potentials (spikes) emitted within a 250 millisecond time window (the duration of the stimulus presentation in this dataset). To compare how similar a single set of stimuli were represented across sessions, we Gaussian-approximated the data recorded for each stimulus. Then for each stimulus class (gratings and scenes), we compared between two sets of Gaussians (one per stimulus).

Our metrics compared between sets of Gaussians that have the same dimensionality, so we performed PCA to equalize dimensionality across all sessions. We concatenated data recorded for different images within a single stimulus class, and extracted the first 19 principal components in replacement of the total number of recorded neurons for further analysis. On average, the extracted 19 principal components explained 83% of the data variance in response to gratings, and 76% of the variance to natural scenes.

D.2.2.2 Estimating response mean and covariance.

To compare between two neural representations, our stochastic metrics take two sets of Gaussian means and covariances as inputs, where each of which is estimated from the principal components (PCs) extracted from the data.

In each session, a stimulus (either a grating or a scene) was presented over 50 repeats. The number of repeats is large compared to conventional neuroscience experiments, but it is still small compared to the total number of recorded neurons (e.g. 110 neurons in one session), or the total number of PCs, which introduces challenges to covariance estimation. For the mean of each stimulus representation, we used the sample mean from the PCs. When number of samples is relatively small, sample covariance has one known bias: it tends to over-estimate large eigenvalues, and under-estimate small eigenvalues of the population covariance. One standard and effective fix in the literature is to use a shrinkage estimator (S^*) – a linear interpolation between an identity matrix (I) and the sample covariance (S) (e.g. (Ledoit and Wolf, 2004; Tong et al., 2018)):

$$S^* = \gamma I + (1 - \gamma)S. \tag{D.1}$$

This interpolation reduces the eigenvalue bias by balancing between eigenvalues of the sample covariance (overly skewed eigenvalue spectrum), and that of the identity matrix (flat eigenvalue spectrum). γ of the shrinkage covariance estimator was chosen using cross-validation. To obtain the cross-validation training set, we randomly sample half of the epochs from each data trial, and for test set, we used the remaining half.

D.2.3 Variational autoencoders and latent factor disentanglement (Supplement to subsection 5.4.3)

D.2.3.1 VAE objectives and architectures used in this study

Because conventional VAE encoders output a latent Gaussian conditional mean and covariance, this makes them an ideal framework with which to apply stochastic shape metrics. In particular, the interpolated Wasserstein distance (equation 5.7) is exact in this case. We used a set of 1800 VAEs trained on `dSprites` from the extensive study by Locatello et al. (2019). These include six variants of the VAE objective (β, Factor, β-TC, DIP-I, DIP-II, Annealed), each with six different levels of regularization strength and 50 repetitions at different random seeds. The dimensionality of the latent representation, from which we obtained activations used in this study, was 10D. We refer the reader to their supplemental document for more details about each architecture and training scheme. The authors provided metadata associated with each network such as training hyperparameters as well as factor disentanglement scores (see below).

In addition to the VAEs trained on `dSprites`, we trained 350 β VAEs on MNIST and CIFAR-10. To remain consistent with the study by Locatello et al. (2019), we trained β-VAEs at 6-8 different levels of regularization strength ($1 \leq \beta \leq 16$) and 50 random initialization seeds. We used the standard VAE symmetric encoder-decoder architecture with L layers ($L = 3$ for MNIST and $L = 4$ for CIFAR-10), each with 64 4×4 convolutional filters with stride 2, followed by a fully-connected layer with 256 hidden units and ReLU activations. The latent representations of these networks were diagonal Gaussians, and were 10D (MNIST) or 50D (CIFAR-10). The final 2D convolution-transpose layer of the decoder used a sigmoid nolinearity to ensure outputs were between $[0, 1]$.

We zero-padded the height and width of MNIST images from 28×28 $\rightarrow$ 32×32. Model training used batch sizes of 64 images, and up to 1000 training epochs. We used the Adam optimizer with 1E-4 learning rate and model checkpoints at each epoch. Models used in this study were from

checkpoints corresponding to the lowest validation loss during training. Latent activations used for shape metric analysis in this study were obtained using a held-out test set of 3500 images.

D.2.3.2 $\overline{\mathcal{W}}_2^{\alpha=0}$ vs. $\overline{\mathcal{W}}_2^{\alpha=2}$ before and after training

Fig. 5.6B of the main text shows that, prior to training, VAEs are primarily separated by mean-insensitive distance ($\overline{\mathcal{W}}_2^{\alpha}, \alpha = 0$, equation 5.7), whereas after training they are separated by covariance-insensitive distance ($\alpha = 2$). We sought to confirm whether this effect persisted across different datasets using VAEs trained on MNIST and CIFAR-10 (described above). SuppFig. D.8 shows that pairwise network $\overline{\mathcal{W}}_2^{\alpha}$ distances before and after training indeed exhibit this effect on these more complex datasets. We reproduced these effects using both default PyTorch weight initialization and Kaiming weight initialization.

D.2.3.3 Computing energy distance between trained VAEs

In addition to measuring interpolated Wasserstein distances (equation 5.7), we also repeated our analyses using energy distance (equation 5.4). Rather than requiring computing means and covariances, this method operates directly on samples. Since VAE latents are parameterized as Gaussian, we generated data by randomly sampling from the Gaussian defined by the model's conditional mean and covariance for a given input. We sampled 64 samples for 2048 images and computed pairwise energy distances between all 1800 networks in the (Locatello et al., 2019) `dSprites` dataset (SuppFig. D.6). Interestingly, the energy dissimilarity matrix was qualitatively different than all of the $\overline{\mathcal{W}}_2^{\alpha}$ dissimilarity matrices (SuppFig. D.6A). The geometry of the embedded points was accordingly different than embeddings derived from $\overline{\mathcal{W}}_2^{\alpha}$ distances (SuppFig. D.6B).

The energy dissimilarity matrix seemed to correlate with those derived from $\overline{\mathcal{W}}_2^{\alpha}$ distances (SuppFig. D.6C). We noted, however, that after re-scaling the dissimilarity matrices such that they lie between $[0, 1]$, the distribution of pairwise energy distances was most in line with interpolated

Wasserstein distances when $\alpha = 1$ (SuppFig. D.6C middle panel).

We repeated the classification and disentanglement kNN analyses done in the main text using neighborhoods defined by energy distance (SuppFig. D.6D,E). In most cases, using energy distance performed as well as, but sometimes worse than $\overline{\mathcal{W}}_2^{\alpha=2}$, the covariance-insensitive Wasserstein metric. It is possible that computing energy distance using a higher number of samples per image than 64 would improve estimates and downstream regression/classification performance. In general it would be interesting to examine the effects of sample size and empirical energy distance estimate convergence. We leave a deeper investigation into this for future work.

D.2.3.4 Low-dimensional projections

To determine a reasonable embedding dimensionality for K networks, we performed the following analysis. Given a symmetric $K \times K$ distance matrix D, with elements $d(i, j)$, we used multidimensional scaling to embed K networks into a low M-dimensional space. Networks in this embedded space can be encoded by a new, Euclidean distance matrix $\tilde{D}$ with elements $\tilde{d}(i, j)$. For each element on the upper-triangle of these matrices, we computed a distortion ratio,

$$\Delta(i, j) = d(i, j)/\tilde{d}(i, j) \tag{D.2}$$

$$\text{Distortion}(i, j) = \max(\Delta(i, j), 1/\Delta(i, j)). \tag{D.3}$$

By sweeping embedding dimensionality M from 1-20, we determined that using an MDS embedding dimensionality of M=15 produced reasonably minimal distortions (SuppFig. D.7) for all the distance matrices. After embedding the networks into 15D, we then performed principal components analysis to obtain the scatterplots in Fig. 5.6A and SuppFig. D.6. In the main text, we used orthogonal Procrustes to align the principal components of each subpanel to the left-most panel. For SuppFig. D.6B, we again aligned all panels to the left-most panel using Procrustes analysis, but allowed for re-scaling in order to compensate for energy distances being on an arbitrary scale

compared with $\overline{\mathcal{W}}_2^{\alpha}$ distances.

D.2.3.5 VAE disentanglement metrics

For each of the 1800 VAEs trained on `dSprites`, Locatello et al. (2019) computed a large array of factor disentanglement scores proposed by previous studies. The scores abbreviated in Fig. 5.6F are listed below, using the same naming convention as in the work of Locatello et al. (2019). We refer the reader to their supplement for more details on each of these scores.

A β-VAE eval accuracy

B Disentanglement, Informativeness, Completeness (DCI) disentanglement

C DCI completeness

D DCI informativeness

E Factor VAE eval accuracy

F Logistic regression mean test accuracy

G Boosted trees mean test accuracy

H Discrete mutual information gap (MIG)

I Modularity score

J Explicitness test score

K Separated Attribute Predictability (SAP) score

L Gaussian total correlation

M Gaussian Wasserstein correlation

N Gaussian Wasserstein normalized correlation

O Mutual information score

D.2.3.6 k-nearest neighbors analyses

Because the stochastic metrics used in this study satisfy the triangle inequality, this permitted non-parametric analyses using k-nearest neighbors (kNN) to determine whether network

similarity carried information about model hyper-parameters and task performance. We used scikit-learn's `KNeighborsClassifier` and `KNeighborsRegressor` for classification and regression analyses, respectively. We withheld a test set and performed 6-fold cross-validation on the remaining data to determine k, the number of neighbors to use for classification/regression. We reported final performance using the average score on the held-out test set.

For classification analyses, we trained models to decode random initial seed (1/50 chance, Fig. 5.6C), and model objective along with regularization strength (6 objective $\times$ 6 regularization strengths in the Locatello study, i.e. 1/36 chance, Fig. 5.6E). In terms of regression analyses, we trained models to predict training reconstruction loss Fig. 5.6D and disentanglement scores Fig. 5.6F and reported average R^2 on the held-out test set.

D.2.4 Additional Details for Patch-Gaussian Augmentation Experiments

D.2.4.1 Training and architecture details

We use the ResNet-18 architecture (He et al., 2016) where an intermediate fully-connected layer of dimension 100 is added after the final average pooling layer, followed by a linear readout layer. All analyses were done on the representations produced of this intermediate fully-connected layer.

Following standard practice, images were randomly cropped, followed by a random horizontal flip. A modified version of the Patch-Gaussian augmentation was applied, where the entire noisy patch is constrained to reside in the image. Lastly, we subtract off the per-channel mean and divide by the per-channel standard deviation. For the Patch-Gaussian augmentation, we swept over 16 values of patch width, $W \in \{2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32\}$ and 7 different values of noise scale, $\sigma \in \{0.05, 0.1, 0.2, 0.3, 0.5, 0.8, 1.\}$, leading to $17 \times 6 = 112$ possible (W, σ) combinations. For each (W, σ) pair, we trained 3 networks, each with a different random

seed, leading to $16 \times 7 \times 3 = 336$ networks. As a baseline, we also trained networks with no Patch-Gaussian augmentation over three random seeds, giving us 339 total networks.

We used stochastic gradient descent with a momentum of 0.9, batch size of 128 and weight decay of 1E-4. Networks were trained for 200 epochs where the learning rate was initially set to 0.1 and halved every 60 epochs.

D.2.5 Visualization of Hidden Layer Representations

To visualize the effect of Patch-Gaussian hyper-parameters on hidden layer representations as shown in Fig. 5.7A, we randomly selected one image from each of the 10 classes, e.g. $z_1, \ldots, z_{10}$. For each image i—and a given value of τ—we drew 100 samples from $\mathcal{N}(z_i, \tau)$ and collected the hidden layer representations, leading to 1,000 points total. Mutli-dimensional scaling was then applied to embed the representations into two dimensions.

D.2.5.1 Stochastic Shape Metric Computation

2,000 images were used for computing the stochastic shape metric. To estimate the conditional mean and covariance for each image, 1,000 samples were first drawn from $\mathcal{N}(z_i, \tau)$. The conditional mean was estimated via a Monte Carlo estimator. The conditional covariance was computed by first computing the Monte Carlo estimator and then adding 0.0001 to the diagonal to ensure the covariance is well-conditioned.

To visualize the metric shape induced by the stochastic shape metric, multi-dimensional scaling was used to embed the networks into 20 dimensions. Principal component analysis was then done to linearly project the MDS embeddings onto the top 2 principal components.

We used three different values for the interpolated Wasserstein distance, $\alpha \in \{0, 1, 2\}$ and 6 values for the magnitude of the input perturbation, $\tau \in \{0.01, 0.05, 0.1, 0.2, 0.3, 0.5\}$. All distance matrices are shown in Fig. D.9. The corresponding two-dimensional embedding are shown in Fig. D.10.

D.3 Proof of Proposition 5.1

We first prove two lemmas, from which the main proposition immediately follows.

Lemma D.1. *If $\mathcal{G}$ is a group of isometries on a metric space (d_1, S) then*

$$d(x, y) = \min_{T \in \mathcal{G}} d_1(x, \boldsymbol{T}(y)) \tag{D.4}$$

is a pseudometric which can be used to define a metric over equivalence classes $[x] = \{y \mid y \sim x\}$ where the equivalence relation is defined as:

$$x \sim y \iff \exists\, \boldsymbol{T} \in \mathcal{G} \quad \textit{such that} \quad x = \boldsymbol{T}(y) \tag{D.5}$$

Proof. This proof is more or less reproduced from Williams et al. (2021), and similar arguments can be found elsewhere within the statistical shape analysis literature.

The equivalence relation in Equation D.5 is self-evident. This simply states that $d(x, y) = 0$ if and only if $x = \boldsymbol{T}(y)$ for some alignment transformation $\boldsymbol{T} \in \mathcal{G}$. Then we define our equivalence relation as: $x \sim y$ if and only if $d(x, y) = 0$. In other words, although d technically only defines a pseudometric on S, it is easily associated to a proper metric on a set of equivalence classes, i.e. the quotient space $(S/\sim)$. See Howes (1995) for more background details (page 27, in particular).

Now we prove that d is symmetric. Let $\boldsymbol{T}_{xy}$ denote the optimal transformation from Y to X. That is, $\boldsymbol{T}_{xy} = \arg\min_{T \in \mathcal{G}} d_1(x, \boldsymbol{T}(y))$ and $\boldsymbol{T}_{yx} = \arg\min_{T \in \mathcal{G}} d_1(y, \boldsymbol{T}(x))$. Then, using the fact that d_1 is symmetric and that $\mathcal{G}$ defines a group of isometries, we have

$$d(x, y) = d_1(x, \boldsymbol{T}_{xy}(y)) = d_1(\boldsymbol{T}_{xy}(y), x) = d_1(y, \boldsymbol{T}_{xy}^{-1}(x))) \leq d_1(y, \boldsymbol{T}_{yx}(x))) = d(y, x)$$

but also

$$d(y,x) = d_1(y, \boldsymbol{T}_{yx}(x)) = d_1(\boldsymbol{T}_{yx}(x), y) = d_1(x, \boldsymbol{T}_{yx}^{-1}(y))) \leq d_1(x, \boldsymbol{T}_{xy}(y))) = d(x,y).$$

The only way for both inequalities to hold is for $d(x,y) = d(y,x)$. Also, we see that $\boldsymbol{T}_{xy} = \boldsymbol{T}_{yx}^{-1}$, which we will exploit below.

It remains to prove the triangle inequality. This is done as follows:

$$\begin{aligned}
d(x,y) &= d_1(x, \boldsymbol{T}_{xy}(y)) && \text{(D.6)}\\
&\leq d_1(x, \boldsymbol{T}_{xz}(\boldsymbol{T}_{zy}(y))) && \text{(D.7)}\\
&\leq d_1(x, \boldsymbol{T}_{xz}(z)) + d_1(\boldsymbol{T}_{xz}(z), \boldsymbol{T}_{xz}(\boldsymbol{T}_{zy}(y))) && \text{(D.8)}\\
&= d_1(x, \boldsymbol{T}_{xz}(z)) + d_1(Z, \boldsymbol{T}_{zy}(y)) && \text{(D.9)}\\
&= d(x,z) + d(z,y) && \text{(D.10)}
\end{aligned}$$

The first inequality follows from replacing the optimal alignment, $\boldsymbol{T}_{xy}$, with a sub-optimal alignment, given by function composition $\boldsymbol{T}_{xz} \circ \boldsymbol{T}_{zy}$. (Recall that $\mathcal{G}$ is a group and so is closed under function compositions.) The second inequality follows from the triangle inequality on d_1, after choosing $\boldsymbol{T}_{xz}(z)$ as the midpoint. The penultimate step follows from $\boldsymbol{T}_{xz}^{-1}$ being an isometry on d_1 and since $\boldsymbol{T}_{xz} \in \mathcal{G}$, we have $\boldsymbol{T}_{xz}^{-1} \in \mathcal{G}$ by the group properties of $\mathcal{G}$.

□

Lemma D.2. *Let (d_2, S_2) be a metric space, let $f(\cdot)$ and $g(\cdot)$ be functions mapping $\mathcal{Z} \mapsto S_2$, and let Q be a probability distribution supported on $\mathcal{Z}$. Then,*

$$d_1(f,g) = \left(\mathop{\mathbb{E}}_{z \sim Q} d_2^2(f(z), g(z)) \right)^{1/2} \tag{D.11}$$

is a metric over the set of functions mapping $\mathcal{Z} \mapsto S_2$.

Proof. Since d_2 is a metric, we have $d_2(x, y) > 0$ if $x \neq y$. Recall our assumption that the support of Q equals $\mathcal{Z}$. Thus, if there exists a $z \in \mathcal{Z}$ for which $f(z) \neq g(z)$, the expectation will evaluate to a positive number and we have $d_1(f, g) > 0$. So we conclude $d_1(f, g) = 0$ if and only if f and g define the exact same mapping from $\mathcal{Z} \mapsto S_2$.

It is also obvious that $d_1(f, g) = d_1(g, f)$, due to the symmetry of d_2. Thus, it only remains to prove the triangle inequality.

Fix any function $h : \mathcal{Z} \mapsto S$. Due to the triangle inequality on d_2, we have:

$$d_1(f, g) = \left(\mathop{\mathbb{E}}_{z \sim Q} d_2^2(f(z), g(z)) \right)^{1/2} \leq \left(\mathop{\mathbb{E}}_{z \sim Q} \left(d_2(f(z), h(z)) + d_2(h(z), g(z)) \right)^2 \right)^{1/2} \tag{D.12}$$

Now let $X = d_2(f(z), h(z))$ and $Y = d_2(h(z), g(z))$. Note that z is a random variable (sampled from Q), and so X and Y are also random variables. We now recall two elementary facts: $\|X\|_2 = (\mathbb{E}[X^2])^{1/2}$ defines a norm over random variables, and $\|X + Y\|_2 \leq \|X\|_2 + \|Y\|_2$ for any two random variables (Minkowski's inequality). Our definitions of X and Y imply that the right hand side of Equation D.12 can be re-written as $\|X + Y\|_2$. And we can therefore conclude the proof since:

$$d_1(f, g) \leq \|X + Y\|_2 \leq \|X\|_2 + \|Y\|_2 = d_1(f, h) + d_1(h, g) \,. \tag{D.13}$$

□

Main proof. Let us restate and then prove Theorem 5.1. We want to show that the following:

$$d(F_i, F_j) = \min_{T \in \mathcal{G}} \left(\mathop{\mathbb{E}}_{z \sim Q} \left[\mathcal{D}^2 \left(F_i^{\phi_i}(\cdot \mid z), F_j^{\phi_j}(\cdot \mid z) \circ \boldsymbol{T}^{-1} \right) \right] \right)^{1/2} \tag{D.14}$$

is a pseudometric over stochastic networks—i.e, a pseudometric over functions F that map inputs $z \in \mathcal{Z}$ onto probability distributions. Recall that $F^{\phi}(\cdot \mid z)$ is a shorthand notation for $F(\phi^{-1}(\cdot) \mid z)$ where ϕ^{-1} is the pre-image of ϕ.

Our key assumptions are that $\mathcal{D}(\cdot, \cdot)$ is a metric over probability distributions and that $\mathcal{G}$ is

a group of isometry transformations with respect to this metric—i.e., for any pair of probability distributions F and G, we have that:

$$\mathcal{D}(F, G) = \mathcal{D}(F \circ \boldsymbol{T}^{-1}, G \circ \boldsymbol{T}^{-1}) \tag{D.15}$$

for any $\boldsymbol{T} \in \mathcal{G}$. It is well-known that the Wasserstein distance (Villani, 2009) and energy distance (Sejdinovic et al., 2013; Székely and Rizzo, 2017) are probability metrics. Further, it is easy to show that orthogonal pushforward transformations are isometries for both metrics. For example, we have for the 2-Wasserstein distance that:

$$\mathcal{W}_2^2(P, Q) = \inf \mathbb{E} \|X - Y\|^2 = \inf \mathbb{E} \|\boldsymbol{T}X - \boldsymbol{T}Y\|^2 = \mathcal{W}_2^2(P \circ \boldsymbol{T}^{-1}, Q \circ \boldsymbol{T}^{-1}) \tag{D.16}$$

for any orthogonal transformation $\boldsymbol{T}$. Thus, for our purposes we can think of $\mathcal{G}$ as being any subgroup of the orthogonal group.

Now that we have reminded ourselves of the main proposition, let us turn to the proof.

Proof. Let us define:

$$d_1(F_i^\phi, F_j^\phi) = \left(\mathop{\mathbb{E}}_{z \sim Q} \left[\mathcal{D}^2 \left(F_i^\phi(\cdot \mid z), F_j^\phi(\cdot \mid z) \right) \right] \right)^{1/2}. \tag{D.17}$$

Plugging this into Equation D.14, we have:

$$d(F_i, F_j) = \min_{\boldsymbol{T} \in \mathcal{G}} d_1(F_i^\phi, F_j^\phi \circ \boldsymbol{T}^{-1}). \tag{D.18}$$

Theorem D.2 tells us that d_1 is a metric. Thus, Theorem D.1 applies to Equation D.18. This permits us to conclude that d is a pseudometric and defines a metric over sets of equivalent neural representations, as claimed. □

D.4 Practical Estimation of Stochastic Shape Metrics

In both biological and artificial networks, we do not have parametric forms for the conditional distributions over neural population responses. Instead, we can only draw samples from these distributions—e.g., by feeding an input into an artificial network and performing a stochastic forward pass, or by recording evoked spike counts to a sensory stimulus in biological data. We consider a simple experimental setup: we are given K stochastic neural networks $\{F_1, \ldots, F_K\}$, M network inputs or conditions $\{z_1, \ldots, z_M\}$, and L repeated observations or measurements of the neural responses to each input. For example, in an artificial network that ingests image data, M would denote the number of images in a test set and L denotes the number of samples per image. Let $\boldsymbol{x}_\ell^{(km)} \in \mathbb{R}^n$ to denote sample ℓ, from network k, to condition m. That is,

$$\boldsymbol{x}_\ell^{(km)} \sim F_k^\phi(\boldsymbol{x} \mid \boldsymbol{z}_m) \quad \text{i.i.d. for } (\ell, m, k) \in \{1, \ldots, L\} \times \{1, \ldots, M\} \times \{1, \ldots, K\}. \tag{D.19}$$

D.4.1 Metrics based on 2-Wasserstein distance and Gaussian assumption

Our main assumption in this section is that distributions over neural activations are multivariate Gaussians. That is, for each stochastic network and every input $\boldsymbol{z} \in \mathcal{Z}$, we have $F_i^\phi(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{\mu}_i(\boldsymbol{z}), \Sigma_i(\boldsymbol{z}))$, where $\boldsymbol{\mu}_i : \mathcal{Z} \mapsto \mathbb{R}^n$ and $\Sigma_i : \mathcal{Z} \mapsto \mathbb{S}^{n \times n}$. If $\boldsymbol{T} : \mathbb{R}^n \mapsto \mathbb{R}^n$ is a linear pushforward map, then the pushforward measure is still Gaussian and is defined by

$$F_j^\phi(\boldsymbol{z}) \circ \boldsymbol{T}^{-1} = \mathcal{N}(\boldsymbol{T}\boldsymbol{\mu}_j(\boldsymbol{z}), \boldsymbol{T}\Sigma_j(\boldsymbol{z})\boldsymbol{T}^\top).$$

The 2 Wasserstein distance between two multivariate Gaussian distributions has a well known

closed form expression (Peyré and Cuturi, 2019, Remark 2.31):

$$\mathcal{W}_2\big(\mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i), \mathcal{N}(\boldsymbol{\mu}_j, \Sigma_j)\big) = \big(\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2 + \mathcal{B}(\Sigma_i, \Sigma_j)^2\big)^{1/2} \tag{D.20}$$

where $\mathcal{B}(\cdot, \cdot)$ is the Bures metric between positive definite matrices. Typically one sees the Bures metric defined as:

$$\mathcal{B}(\Sigma_i, \Sigma_j) = \Big(\operatorname{Tr}[\Sigma_i] + \operatorname{Tr}[\Sigma_j] - 2\operatorname{Tr}[\Sigma_i^{1/2}\Sigma_j\Sigma_i^{1/2}]^{1/2}\Big)^{1/2}. \tag{D.21}$$

If we use this expression, minimizing $\mathcal{B}(\Sigma_i, \boldsymbol{T}\Sigma_j\boldsymbol{T}^\top)$ over nuisance transformations $\boldsymbol{T} \in \mathcal{G}$ is not straightforward.[1] However, an equivalent formulation of the Bures metric is:

$$\mathcal{B}(\Sigma_i, \Sigma_j) = \min_{\boldsymbol{U}} \ \|\Sigma_i^{1/2} - \Sigma_j^{1/2}\boldsymbol{U}\|_F \tag{D.22}$$

where the minimization is over $\boldsymbol{U} \in \mathcal{O}(n)$. The equivalence between Eqs. D.21,D.22 is already established in the literature (see Theorem 1 of Bhatia et al. 2019). For the sake of completeness we have included a proof in subsection D.6.4.

Recall that we are given sampled neural responses $\{\boldsymbol{x}_\ell^{(km)}\}_{k,m,\ell}^{K,M,L}$ as specified in Equation D.19. Using these, we can estimate the mean and covariance of each distribution:

$$\hat{\boldsymbol{\mu}}_k(\boldsymbol{z}_m) = \frac{1}{L}\sum_\ell \boldsymbol{x}_\ell^{(km)} \quad \text{and} \quad \hat{\Sigma}_k(\boldsymbol{z}_m) = \frac{1}{L}\sum_\ell \boldsymbol{x}_\ell^{(km)}\boldsymbol{x}_\ell^{(km)\top} - \hat{\boldsymbol{\mu}}_k(\boldsymbol{z}_m)\hat{\boldsymbol{\mu}}_k(\boldsymbol{z}_m)^\top. \tag{D.23}$$

Here, we've used the typical maximum likelihood estimators. However, any consistent estimator will suffice.

[1] Although there are certain tricks one can exploit to compute the gradient (Newton-Schulz iterations), the constraint that $\boldsymbol{T} \in \mathcal{G}$ is non-trivial. When $\mathcal{G}$ is a continuous manifold (e.g. the orthogonal or special orthogonal group), one can resort to manifold optimization algorithms. These algorithms are somewhat cumbersome but nonetheless a plausible approach. However, even this would not cover the case where $\mathcal{G}$ is a discrete set (e.g., the permutation group).

The proposition below summarizes the main result of this section. Using this proposition, we produce an estimate of the distance between two stochastic networks by alternating minimization (i.e. block coordinate descent) over $\boldsymbol{T}, \boldsymbol{U}_1, \ldots, \boldsymbol{U}_M$. Each parameter update can often be solved exactly. For example, we typically consider the case of orthogonal nuisance transformations, i.e. $\mathcal{G} = O(n)$, in which case all parameter updates correspond to solving an orthogonal Procrustes problem (Gower and Dijksterhuis, 2004). Further, the minimizations over $\{\boldsymbol{U}_1, \ldots, \boldsymbol{U}_M\}$ can be done in parallel.

Proposition D.3. *If $F_i^{\phi}(z)$ and $F_j^{\phi}(z)$ are both Gaussian for all $z \in \mathcal{Z}$, then:*

$$\hat{d}(F_i, F_j) = \min_{\boldsymbol{T}, \boldsymbol{U}_1, \ldots, \boldsymbol{U}_M} \left(\frac{1}{M} \sum_{m=1}^{M} \|\hat{\boldsymbol{\mu}}_i(z_m) - \boldsymbol{T}\hat{\boldsymbol{\mu}}_j(z_m)\|^2 + \|\hat{\Sigma}_i(z_m)^{1/2} - \boldsymbol{T}\hat{\Sigma}_j(z_m)^{1/2}\boldsymbol{U}_m\|_F^2 \right)^{1/2}$$

is a consistent estimator of a stochastic shape distance (eq. 5.5) with the 2-Wasserstein distance used as the "ground metric." The minimization in the above equation is performed over $\boldsymbol{T} \in \mathcal{G}$ and $\boldsymbol{U}_m \in O(n)$ for all $m \in \{1, \ldots, M\}$.

Proof. Plugging Eqs. D.20,D.22 into our definition of stochastic distance (eq. 5.5 from Theorem 5.1), we have:

$$d(F_i, F_j) = \min_{\boldsymbol{T}} \left(\mathop{\mathbb{E}}_{z \sim Q} \|\boldsymbol{\mu}_i(z) - \boldsymbol{T}\boldsymbol{\mu}_j(z)\|^2 + \min_{\boldsymbol{U}} \|\Sigma_i(z)^{1/2} - \boldsymbol{T}\Sigma_j(z)^{1/2}\boldsymbol{T}^\top \boldsymbol{U}\|_F^2 \right)^{1/2}. \tag{D.24}$$

Given M i.i.d. samples $z_m \sim Q$ for $m \in \{1, \ldots, M\}$, we can estimate the expectation with an empirical average:

$$\min_{\boldsymbol{T}} \left(\frac{1}{M} \sum_{m=1}^{M} \|\boldsymbol{\mu}_i(z_m) - \boldsymbol{T}\boldsymbol{\mu}_j(z_m)\|^2 + \min_{\widetilde{\boldsymbol{U}}_m} \|\Sigma_i(z_m)^{1/2} - \boldsymbol{T}\Sigma_j(z_m)^{1/2}\boldsymbol{T}^\top \widetilde{\boldsymbol{U}}_m\|_F^2 \right)^{1/2} \tag{D.25}$$

Next, we pull out the minimization over each $\widetilde{\boldsymbol{U}}_m \in O(n)$ outside the sum. Additionally, since $\mathcal{G}$ is a group of isometries on $\mathbb{R}^n$, we know that $\mathcal{G}$ is a subgroup of the orthogonal group. Thus,

every $T \in \mathcal{G}$ is an orthogonal matrix, so $T^\top \widetilde{U}_m$ is also an orthogonal matrix. Thus we introduce a change of variables $U_m = T^\top \widetilde{U}_m$ and minimize over $U_m \in O(n)$, as this attains the same minimum value. In summary, we have:

$$\min_{T, U_1, \ldots, U_M} \left(\frac{1}{M} \sum_{m=1}^{M} \|\mu_i(z_m) - T\mu_j(z_m)\|^2 + \|\Sigma_i(z_m)^{1/2} - T\Sigma_j(z_m)^{1/2} U_m\|_F^2 \right)^{1/2}. \tag{D.26}$$

The only remaining step is to replace every $\mu(z)$ and $\Sigma(z)$ with some consistent estimator, such as the empirical mean and covariance (see eq. D.23). In the limit as $M \to \infty$ and $L \to \infty$, we have convergence to the true distance due to the law of large numbers. □

D.4.1.1 Algorithmic complexity and computational considerations

To compute distances using the 2 Wasserstein ground metric between two Gaussian distributed stochastic network representations (Eq. 5.6), we used closed form updates of the orthogonal procrustes problem (Gower and Dijksterhuis, 2004) for $T \in O$ and $\{U_m\}_{m=1}^M$ in alternation, using S iterations. Importantly, T and each U_m can be solved exactly at each alternation (described above). For K stochastic networks, we must consider $O(K^2)$ total pairwise comparisons.

Computing the optimal T at each step involves aligning two stochastic representations, each comprising a stacked matrix of M n-dimensional means and M $n \times n$ covariances using Procrustes alignment (Gower and Dijksterhuis, 2004). This involves a matrix multiplication and singular value decomposition with $O(Mn^3)$ combined complexity, assuming $n < M$. Similarly, at each step, computing the Bures metric requires solving for M $n \times n$ orthogonal matrices, $\{U_m\}_{m=1}^M$, with total complexity $O(Mn^3)$. Thus, the total algorithmic worst-case time complexity is $O(K^2 S M n^3)$.

Notably, this computation is highly parallelizable over the K^2 pairwise comparisons. For the VAE and patch-Gaussian results in the main text (Figures 5.6 and 5.7), we distributed the distance matrix calculation by distributing pairwise comparisons over single CPU cores, with each comparison taking a few seconds to complete.

D.4.2 Metrics based on Energy distance

This section outlines an alternative measure of stochastic representational distance that does not require any parametric assumption (e.g. Gaussian) on stochastic neural responses. We use *energy distance*, $\mathcal{E}_q$ defined in Equation 5.4, as the ground metric appearing in Theorem 5.1. As explained in the main text, this distance has favorable estimation properties in high dimensional spaces relative to the Wasserstein distances. It remains an open problem to develop estimation procedures for Wasserstein-based stochastic shape distances without the assumption of Gaussianity.

To compute the stochastic shape distance, we need to solve the following optimization problem:

$$\underset{T\in\mathcal{G}}{\operatorname{argmin}} \underset{z\sim Q}{\mathbb{E}} \left[\mathcal{E}_q^2 \left(F_i^\phi(z), F_j^\phi(z) \circ \boldsymbol{T}^{-1} \right) \right] \tag{D.27}$$

Note that we have squared the expression occurring in the main proposition – i.e., we have dropped the $(\cdot)^{1/2}$ operation—as this does not effect the optimal alignment transformation.

First, let's focus on the innermost term of Equation D.27. Let $X_i, X_i' \sim F_i^\phi(\cdot \mid \boldsymbol{z})$ and $X_j, X_j' \sim F_j^\phi(\cdot \mid \boldsymbol{z})$, independently and treating the input $\boldsymbol{z}$ as fixed for now. Now, since $\mathcal{G}$ is a group of isometries with respect to the Euclidean norm, we have:

$$\begin{aligned}
\mathcal{E}_q^2(F_i^\phi(\cdot \mid \boldsymbol{z}), F_j^\phi(\cdot \mid \boldsymbol{z}) \circ \boldsymbol{T}^{-1}) &= \mathbb{E}\|X_i - \boldsymbol{T}X_j\|^q - \tfrac{1}{2}\mathbb{E}\|X_i - X_i'\|^q - \tfrac{1}{2}\mathbb{E}\|\boldsymbol{T}X_j - \boldsymbol{T}X_j'\|^q \\
&= \mathbb{E}\|X_i - \boldsymbol{T}X_j\|^q - \tfrac{1}{2}\mathbb{E}\|X_i - X_i'\|^q - \tfrac{1}{2}\mathbb{E}\|X_j - X_j'\|^q
\end{aligned}$$

The final two terms are constant with respect to $\boldsymbol{T}$, so we can drop them from the objective function without effecting the result. Thus, Equation D.27 can be simplified to:

$$\underset{T\in\mathcal{G}}{\operatorname{argmin}} \underset{z\sim Q}{\mathbb{E}} \left[\mathbb{E}\|X_i - \boldsymbol{T}X_j\|^q \right] \tag{D.28}$$

Here, the outer expectation is over network inputs $\boldsymbol{z}$ and the inner expectation is over conditional distributions, $X_i \sim F_i^{\phi}(\cdot \mid \boldsymbol{z})$ and $X_j \sim F_j^{\phi}(\cdot \mid \boldsymbol{z})$.

Recall again that we are given sampled responses $\{\boldsymbol{x}_\ell^{(km)}\}_{k,m,\ell}^{K,M,L}$ as specified in Equation D.19. To construct a consistent estimator, we evoke the law of large numbers to replace the expectations with empirical averages. Equation D.28 becomes:

$$\underset{T \in \mathcal{G}}{\operatorname{argmin}} \frac{1}{ML^2} \sum_{m=1}^{M} \sum_{\ell=1}^{L} \sum_{p=1}^{L} \|\boldsymbol{x}_\ell^{(im)} - \boldsymbol{T}\boldsymbol{x}_p^{(jm)}\|^q \tag{D.29}$$

When $q = 2$, the optimal $\boldsymbol{T} \in \mathcal{G}$ can often be identified efficiently—e.g., by solving a Procrustes problem when $\mathcal{G} = \mathcal{O}$ (Gower and Dijksterhuis, 2004) or a linear assignment problem when $\mathcal{G} = \mathcal{P}$ (Burkard et al., 2012). However, when $q = 2$ the stochastic shape distance only depends on the mean and is insensitive to higher-order moments of the neural response (see subsection D.6.1) In the more interesting case where $q \neq 2$, we can use iteratively re-weighted least squares (Kuhn, 1973) to identify the solution.[2] Details of this well-known algorithm are provided in subsection D.6.2.

Now, let $\boldsymbol{T}^* \in \mathcal{G}$ be the solution to Equation D.29. Using this it is straightforward to estimate the desired stochastic shape distance. Our estimate of $d(F_i, F_j)$ is:

$$\frac{1}{m} \sum_{m} \left(\frac{1}{L^2} \sum_{\ell,p} \|\boldsymbol{x}_\ell^{(im)} - \boldsymbol{T}^* \boldsymbol{x}_p^{(jm)}\|^q - \frac{1}{L(L-1)} \sum_{\ell > p} \|\boldsymbol{x}_\ell^{(im)} - \boldsymbol{x}_p^{(im)}\|^q - \frac{1}{L(L-1)} \sum_{\ell > p} \|\boldsymbol{x}_\ell^{(jm)} - \boldsymbol{x}_p^{(jm)}\|^q \right)$$

where the sums over $\ell > p$ are over all $L(L-1)/2$ pairwise combinations between L sampled activations. Each of the three terms in the expression above is a consistent (though not unbiased) estimator of its corresponding term in definition of energy distance (eq. 5.4). However, if the final two terms above are over-estimated in magnitude and the first term is under-estimated, the

[2] One could alternatively consider using manifold optimization methods when $\mathcal{G}$ is a continuous manifold. However, these methods are somewhat cumbersome and aren't easy to extend to the case where $\mathcal{G}$ is a discrete set, such as the set of all permutations.

overall estimate of $d(F_i, F_j)$ may be negative. This violates perhaps the most important property of a metric space that distances should be nonnegative. Triangle inequality violations are also possible.

We propose a simple fix using basic ideas from the literature on *metric repair* (Brickell et al., 2008). Given a collection of K networks, we use the procedure above to compute an estimate of the $K \times K$ distance matrix $\widetilde{\boldsymbol{D}}$ where $\widetilde{\boldsymbol{D}}_{ij} \approx d(F_i, F_j)$. We then find the matrix $\boldsymbol{D}^*$ that is closest to our estimate $\widetilde{\boldsymbol{D}}$ according a quadratic loss, and which satisfies all the axioms of a metric space. This amounts to solving a quadratic program, as detailed in subsection D.6.3.

D.5 Interpolated 2-Wasserstein Metrics

D.5.1 Proof that $\overline{\mathcal{W}}_2^{\alpha}$ is a metric

We start by proving a well-known and basic lemma, which states that the ℓ_p norm of a collection of metrics also defines a metric.

Lemma D.4. *Let $d_1, \ldots, d_n$ be a collection of metrics on a set $\mathcal{S}$. Then, for any $p > 1$,*

$$d(x, y) = \sqrt[p]{d_1(x, y)^p + \ldots + d_n(x, y)^p} \tag{D.30}$$

is a metric on $\mathcal{S}$.

Proof. It is obvious that $d(x, y) = 0$ if and only if $d_1(x, y) = \ldots = d_n(x, y) = 0$ and that $d(x, y) = d(y, x)$. So it is only non-trivial to prove the triangle inequality.

Let $\boldsymbol{d}(x, y)$ denote the vector in $\mathbb{R}^n$ holding each distance. That is:

$$\boldsymbol{d}(x, y) = \begin{bmatrix} d_1(x, y) \\ \vdots \\ d_n(x, y) \end{bmatrix} \tag{D.31}$$

The triangle inequality now follows from:

$$d(x,y) = \sqrt[p]{d_1(x,y)^p + \ldots + d_n(x,y)^p} = \|\boldsymbol{d}(x,y)\|_p \tag{D.32}$$

$$\leq \|\boldsymbol{d}(x,m) + \boldsymbol{d}(m,y)\|_p \tag{D.33}$$

$$\leq \|\boldsymbol{d}(x,m)\|_p + \|\boldsymbol{d}(m,y)\|_p \tag{D.34}$$

$$= d(x,m) + d(m,y) \tag{D.35}$$

for all $x \in \mathcal{S}$, $y \in \mathcal{S}$, and $m \in \mathcal{S}$. The first inequality follows from the triangle inequality on each $d_1, \ldots, d_n$ and then from $\|\cdot\|_p$ being a monotonically increasing function of each vector coordinate. The second inequality follows from the sub-additivity property of all norms. □

This lemma provides further perspective on the closed form expression we gave in subsection D.4.1 for the 2-Wasserstein distance between Gaussian distributions. In particular, if $P_i = \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$ and $P_j = \mathcal{N}(\boldsymbol{\mu}_j, \Sigma_j)$, we saw in Equation D.20 that:

$$\mathcal{W}_2(P_i, P_j) = \left(d^2_{\boldsymbol{\mu}}(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) + d^2_{\Sigma}(\Sigma_i, \Sigma_j)\right)^{1/2} \tag{D.36}$$

where $d^2_{\boldsymbol{\mu}}$ is the squared Euclidean distance between the means and d^2_{Σ} is the squared Bures metric between the covariances. Thus, the 2-Wasserstein distance between Gaussians can be intuitively thought of as the ℓ_2 norm of this pair of metrics.

It is trivial to verify that all the properties of a metric are preserved the distance is multiplied a scalar $\alpha > 0$. That is, if g is a metric, then $d(x,y) = \alpha g(x,y)$ is also a metric. Combining this with Theorem D.4 it is obvious that,

$$\overline{\mathcal{W}}^{\alpha}_2(P_i, P_j) = \left(\alpha \cdot d^2_{\boldsymbol{\mu}}(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) + (2-\alpha) \cdot d^2_{\Sigma}(\Sigma_i, \Sigma_j)\right)^{1/2} \tag{D.37}$$

which is simply a re-statement of Equation 5.7, is a metric for any $0 < \alpha < 2$.

D.5.2 Lower Bound on Interpolated Shape Distances

We now derive a simple lower bound on stochastic shape distances when $\overline{\mathcal{W}}_2^\alpha$ is used as the ground metric. Let d_α^2 denote the squared shape distance of interest, for any chosen $0 \leq \alpha \leq 2$. We have:

$$\begin{aligned}
d_\alpha^2(F_i, F_j) &= \min_{T \in \mathcal{G}} \mathbb{E}_z \left[(\overline{\mathcal{W}}_2^\alpha)^2 \left(F_i^\phi(\cdot \mid z), F_j^\phi(\cdot \mid z) \circ T^{-1} \right) \right] \\
&= \min_{T \in \mathcal{G}} \mathbb{E}_z \left[\alpha \cdot d_{\boldsymbol{\mu}}^2(\boldsymbol{\mu}_i(z), T\boldsymbol{\mu}_j(z)) + (2-\alpha) \cdot d_\Sigma^2(\Sigma_i(z), T\Sigma_j(z)T^\top) \right] \\
&\geq \alpha \cdot \min_{T_\mu \in \mathcal{G}} \mathbb{E}_z \left[d_{\boldsymbol{\mu}}^2(\boldsymbol{\mu}_i(z), T_\mu \boldsymbol{\mu}_j(z)) \right] + (2-\alpha) \cdot \min_{T_\Sigma \in \mathcal{G}} \mathbb{E}_z \left[d_\Sigma^2(\Sigma_i(z), T_\Sigma \Sigma_j(z) T_\Sigma^\top) \right]
\end{aligned}$$

The inequality here follows from the linearity of expectation and then from separately minimizing the two terms. The inequality is tight if the optimal value of $T_{\boldsymbol{\mu}}$ equals the optimal value of T_Σ. Furthermore, the two minimized terms in the final expression are proportional to the squared shape distance when $\alpha = 2$ and $\alpha = 0$, respectively:

$$\min_{T \in \mathcal{G}} \mathbb{E}_z \left[d_{\boldsymbol{\mu}}^2(\boldsymbol{\mu}_i(z), T\boldsymbol{\mu}_j(z)) \right] = \frac{1}{2} \cdot d_{\alpha=2}^2(F_i, F_j) \tag{D.38}$$

$$\min_{T \in \mathcal{G}} \mathbb{E}_z \left[d_\Sigma^2(\Sigma_i(z), T_\Sigma \Sigma_j(z) T_\Sigma^\top) \right] = \frac{1}{2} \cdot d_{\alpha=0}^2(F_i, F_j) \tag{D.39}$$

Thus, in summary we have:

$$d_\alpha^2(F_i, F_j) \geq \frac{\alpha}{2} \cdot d_{\alpha=2}^2(F_i, F_j) + \frac{2-\alpha}{2} \cdot d_{\alpha=0}^2(F_i, F_j) \tag{D.40}$$

$$\Rightarrow \; d_\alpha(F_i, F_j) \geq \sqrt{\frac{\alpha}{2} \cdot d_{\alpha=2}^2(F_i, F_j) + \frac{2-\alpha}{2} \cdot d_{\alpha=0}^2(F_i, F_j)} \tag{D.41}$$

D.5.3 Interpretation of $\overline{\mathcal{W}}_2^{\alpha}$ when Gaussian assumption is violated

When neural responses are Gaussian-distributed, then $\overline{\mathcal{W}}_2^{\alpha}$ can be interpreted as a natural extension of 2-Wasserstein distance (see Figure 5.3). What if neural responses are *not* Gaussian-distributed? Concretely, consider two distributions P_i and P_j, which are not necessarily Gaussian. We can still define the first two moments (mean and covariance) of these distributions:

$$\boldsymbol{\mu}_i = \mathbb{E}_{\boldsymbol{x}\sim P_i}[\boldsymbol{x}] \quad \text{and} \quad \Sigma_i = \mathbb{E}_{\boldsymbol{x}\sim P_i}[(\boldsymbol{x}-\boldsymbol{\mu}_i)(\boldsymbol{x}-\boldsymbol{\mu}_i)^\top]\,. \tag{D.42}$$

Using these, we can still compute $\overline{\mathcal{W}}_2^{\alpha}(P_i, P_j)$ as before.

However, it is no longer the case that this calculation will coincide with the 2-Wasserstein distance between P_i and P_j. Because of this, we can no longer conceptualize $\overline{\mathcal{W}}_2^{\alpha}$ as the amount of energy taken to transport P_i onto P_j with the parameter α differentially weighting the cost of transporting mass due to mismatches in the mean and covariance.

On the other hand, $\overline{\mathcal{W}}_2^{\alpha}$ may still be a reasonable ground metric in many practical circumstances. In particular, it is obvious that $\overline{\mathcal{W}}_2^{\alpha}(P_i, P_j) = 0$ if and only if the mean and covariance of these distributions match. Thus, it is a pseudometric over all probability distributions and a metric on equivalence classes defined by the equivalence relation $P_i \sim P_j$ if and only if $\boldsymbol{\mu}_i = \boldsymbol{\mu}_j$ and $\Sigma_i = \Sigma_j$. From this, it is easy to show that the stochastic shape metric (eq. 5.5) based on this ground metric also satisfies the metric space axioms, including the triangle inequality.

In high-dimensional datasets, it is often challenging to estimate and interpret the higher-order statistical moments of a distribution. In the setting of comparing stochastic neural representations, one could argue that it is reasonable to settle for a ground metric that is insensitive to these higher-order moments. From this perspective, $\overline{\mathcal{W}}_2^{\alpha}$ belongs to a larger family of ground metrics that can be expressed:

$$\mathcal{D}(P_i, P_j) = (d_{\boldsymbol{\mu}}^2(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) + d_{\Sigma}^2(\Sigma_i, \Sigma_j))^{1/2} \tag{D.43}$$

for some chosen metric on the means, $d_{\boldsymbol{\mu}}$, and another chosen metric on the covariances, d_{Σ}. Again, no assumption on whether P_i and P_j being Gaussian is strictly necessary. A more thorough exploration of these alternative ground metrics is a potential direction of future research.

D.6 Miscellaneous Theory and Background

D.6.1 Energy distance as a trial-averaged shape metric when $q = 2$

Performing representational dissimilarity analysis on trial average activity measurements is already common practice in neuroscience. Here, we show that this approach arises as a special case of the stochastic shape distances explored in this manuscript. When $q = 2$, the energy distance is given by:

$$\mathcal{E}_2(P, Q) = (\mathbb{E}\|X - Y\|^2 - \tfrac{1}{2}\mathbb{E}\|X - X'\|^2 - \tfrac{1}{2}\mathbb{E}\|Y - Y'\|^2)^{1/2} \tag{D.44}$$

Since X and X' are independent and identically distributed random variables, we have:

$$\tfrac{1}{2}\mathbb{E}\|X - X'\|^2 = \tfrac{1}{2}\mathbb{E}[X^\top X] + \tfrac{1}{2}\mathbb{E}[X'^\top X'] - \mathbb{E}[X^\top X'] \tag{D.45}$$

$$= \mathbb{E}[X^\top X] - \mathbb{E}[X^\top X'] \tag{D.46}$$

$$= \mathbb{E}[X^\top X] - \mathbb{E}[X]^\top \mathbb{E}[X] \tag{D.47}$$

Likewise,

$$\tfrac{1}{2}\mathbb{E}\|Y - Y'\|^2 = \mathbb{E}[Y^\top Y] - \mathbb{E}[Y]^\top \mathbb{E}[Y]. \tag{D.48}$$

Plugging these expressions into Equation D.44 and simplifying we see that:

$$
\begin{aligned}
\mathcal{E}_2(P, Q) &= (\mathbb{E}\|X - Y\|^2 - \mathbb{E}[X^\top X] + \mathbb{E}[X]^\top \mathbb{E}[X] - \mathbb{E}[Y^\top Y] + \mathbb{E}[Y]^\top \mathbb{E}[Y])^{1/2} \\
&= (\cancel{\mathbb{E}[X^\top X]} + \cancel{\mathbb{E}[Y^\top Y]} - 2\mathbb{E}[X^\top Y] \\
&\quad - \cancel{\mathbb{E}[X^\top X]} + \mathbb{E}[X]^\top \mathbb{E}[X] - \cancel{\mathbb{E}[Y^\top Y]} + \mathbb{E}[Y]^\top \mathbb{E}[Y])^{1/2} \\
&= (\mathbb{E}[X]^\top \mathbb{E}[X] + \mathbb{E}[Y]^\top \mathbb{E}[Y] - 2\mathbb{E}[X^\top Y])^{1/2} \\
&= (\|\mathbb{E}[X] - \mathbb{E}[Y]\|^2)^{1/2} \\
&= \|\mathbb{E}[X] - \mathbb{E}[Y]\|
\end{aligned}
$$

To summarize, we have shown that the $q = 2$ energy distance between a distribution P and Q is equal to the Euclidean distance between the mean of P and the mean of Q. If we use this energy distance as the ground metric, $\mathcal{D}$, in Theorem 5.1 to construct a stochastic shape distance, we are essentially calculating a deterministic shape distance[3] on the mean response patterns.

D.6.2 Iteratively Reweighted Least Squares

Fix q to be a value on the open interval $(0, 2)$ and consider the following optimization problem:

$$
\min_{T \in \mathcal{G}} \left\{ f(\boldsymbol{T}) = \sum_{i=1}^{N} \|\boldsymbol{y}_i - \boldsymbol{T}\boldsymbol{x}_i\|^q \right\} \tag{D.49}
$$

It is easy to see that Equation D.29 is an instance of this problem for a particular choice of vectors $\boldsymbol{x}_i \in \mathbb{R}^n$ and $\boldsymbol{y}_i \in \mathbb{R}^n$.

Our key assumption will be that we can efficiently solve the following weighted least squares problem:

$$
\min_{T \in \mathcal{G}} \sum_{i=1}^{N} w_i \|\boldsymbol{y}_i - \boldsymbol{T}\boldsymbol{x}_i\|^2 \tag{D.50}
$$

[3] Specifically, see the distances covered under Proposition 1 in Williams et al. (2021).

for any choice of weightings, $w_1, \ldots, w_N$. Again, this is possible when $\mathcal{G}$ is the orthogonal group (Procrustes problem) or the permutation group (linear assignment).

Iteratively re-weighted least squares algorithms are a family of methods that are encompassed by the even larger family of majorize-minimization algorithms (Lange, 2016). The specific method we deploy can be viewed as an extension to Weiszfeld's algorithm (Kuhn, 1973). Our starting point is to recognize that the function $s \mapsto s^{q/2}$ is concave for $0 < q < 2$ and $s \geq 0$. Thus, we can derive an upper bound using the first-order Taylor expansion:

$$(s + \delta)^{q/2} \leq s^{q/2} + \delta \left(\frac{\mathrm{d}}{\mathrm{d}s} s^{q/2} \right) = s^{q/2} + \frac{q}{2} \left(\frac{\delta}{s^{(1-q/2)}} \right), \tag{D.51}$$

for any δ such that $s + \delta \geq 0$.

We will now use this fact to derive an upper bound on the objective function in Equation D.49. Let $\boldsymbol{T}^{(t)} \in \mathcal{G}$ represent our estimate of the optimal $\boldsymbol{T} \in \mathcal{G}$ after t iterations of our algorithm, and let $\boldsymbol{T} \in \mathcal{G}$ denote any feasible transformation. Then, for $i \in \{1, \ldots, N\}$, define:

$$s_i^{(t)} = \|\boldsymbol{y}_i - \boldsymbol{T}^{(t)} \boldsymbol{x}_i\|^2 \tag{D.52}$$

$$\delta_i^{(t)} = \|\boldsymbol{y}_i - \boldsymbol{T} \boldsymbol{x}_i\|^2 - s_i^{(t)} \tag{D.53}$$

Notice that these definitions imply $s_i^{(t)} + \delta_i^{(t)} \geq 0$. Now, plugging into Equation D.51, we have:

$$\left(s_i^{(t)} + \delta_i^{(t)} \right)^{q/2} = \|\boldsymbol{y}_i - \boldsymbol{T} \boldsymbol{x}_i\|^q \leq \left(s_i^{(t)} \right)^{q/2} + \frac{q}{2} \left(\frac{\delta_i^{(t)}}{\left(s_i^{(t)} \right)^{(1-q/2)}} \right) \tag{D.54}$$

This is an upper bound for each term in the sum of the original objective function. Therefore,

plugging in the definitions of $s_i^{(t)}$ and $\delta_i^{(t)}$, we have:

$$f(\boldsymbol{T}) = \sum_{i=1}^{N} \|\boldsymbol{y}_i - \boldsymbol{T}\boldsymbol{x}_i\|^q \leq \sum_{i=1}^{N} \left(\|\boldsymbol{y}_i - \boldsymbol{T}^{(t)}\boldsymbol{x}_i\|^2\right)^{q/2} + \frac{q}{2}\left(\frac{\|\boldsymbol{y}_i - \boldsymbol{T}\boldsymbol{x}_i\|^2 - s_i^{(t)}}{\left(\|\boldsymbol{y}_i - \boldsymbol{T}^{(t)}\boldsymbol{x}_i\|^2\right)^{(1-q/2)}}\right) \tag{D.55}$$

$$= \sum_{i=1}^{N} \|\boldsymbol{y}_i - \boldsymbol{T}^{(t)}\boldsymbol{x}_i\|^q + \frac{q}{2}\left(\frac{\|\boldsymbol{y}_i - \boldsymbol{T}\boldsymbol{x}_i\|^2 - \|\boldsymbol{y}_i - \boldsymbol{T}^{(t)}\boldsymbol{x}_i\|^2}{\|\boldsymbol{y}_i - \boldsymbol{T}^{(t)}\boldsymbol{x}_i\|^{2-q}}\right) \tag{D.56}$$

$$\triangleq Q(\boldsymbol{T} \mid \boldsymbol{T}^{(t)}) \tag{D.57}$$

Here, we view $Q(\boldsymbol{T} \mid \boldsymbol{T}^{(t)})$ as a function of $\boldsymbol{T}$—i.e. $\boldsymbol{T}^{(t)}$ is fixed. The calculations above show that $Q(\boldsymbol{T} \mid \boldsymbol{T}^{(t)})$ provides an upper bound on the objective function for any $\boldsymbol{T} \in \mathcal{G}$. Furthermore, it is easy to check that $f(\boldsymbol{T}^{(t)}) = Q(\boldsymbol{T}^{(t)} \mid \boldsymbol{T}^{(t)})$—i.e., the upper bound is tight at $\boldsymbol{T} = \boldsymbol{T}^{(t)}$.

We now have all the necessary ingredients to derive an algorithm. We start by initializing $\boldsymbol{T}^{(1)} \in \mathcal{G}$ by some method. Then we compute $\{\boldsymbol{T}^{(2)}, \boldsymbol{T}^{(3)}, \ldots\}$ iteratively according to:

$$\boldsymbol{T}^{(t+1)} = \operatorname*{argmin}_{\boldsymbol{T} \in \mathcal{G}} Q(\boldsymbol{T} \mid \boldsymbol{T}^{(t)}) = \operatorname*{argmin}_{\boldsymbol{T} \in \mathcal{G}} \sum_{i=1}^{N} \frac{\|\boldsymbol{y}_i - \boldsymbol{T}\boldsymbol{x}_i\|^2}{\|\boldsymbol{y}_i - \boldsymbol{T}^{(t)}\boldsymbol{x}_i\|^{2-q}}. \tag{D.58}$$

The last equality here follows from dropping terms from Equation D.56 that are constant.[4] Intuitively, at each step we are minimizing a surrogate function $Q(\boldsymbol{T} \mid \boldsymbol{T}^{(t)})$ that upper bounds the true objective function. This surrogate function is easy to optimize since the minimization is a special case of Equation D.50 with weightings:

$$w_i = \frac{1}{\|\boldsymbol{y}_i - \boldsymbol{T}^{(t)}\boldsymbol{x}_i\|^{2-q}} \tag{D.59}$$

Furthermore, because we showed that the upper bound is tight at $\boldsymbol{T} = \boldsymbol{T}^{(t)}$, we have:

$$f(\boldsymbol{T}^{(t+1)}) \leq Q(\boldsymbol{T}^{(t+1)} \mid \boldsymbol{T}^{(t)}) = \min_{\boldsymbol{T} \in \mathcal{G}} Q(\boldsymbol{T} \mid \boldsymbol{T}^{(t)}) \leq Q(\boldsymbol{T}^{(t)} \mid \boldsymbol{T}^{(t)}) = f(\boldsymbol{T}^{(t)}) \tag{D.60}$$

[4] It is important to understand that we are treating $\boldsymbol{T}^{(t)}$ as a constant. Only terms that depend on $\boldsymbol{T}$ matter for the minimization. We also further simplified by rescaling $Q(\boldsymbol{T} \mid \boldsymbol{T}^{(t)})$ by a factor of $2/q$, which doesn't affect the value at which the minimum is attained.

which shows that the the objective function never increases as the algorithm progresses.

D.6.3 Quadratic Metric Repair

We are given a symmetric estimate of a distance matrix $\widetilde{\boldsymbol{D}} \in \mathbb{R}^{K \times K}$, which may contain negative entries and triangle inequality violations. Let $\widetilde{\boldsymbol{d}} \in \mathbb{R}^{K(K-1)/2}$ be a vector holding the upper triangular entries of $\widetilde{\boldsymbol{D}}$, excluding the diagonal. Then, consider the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & \|\boldsymbol{x} - \widetilde{\boldsymbol{d}}\|^2 \\ \text{subject to} \quad & x_i \geq 0, \qquad\qquad \forall i \in \{1, \ldots, K(K-1)/2\} \\ & x_i + x_j - x_k \geq 0, \quad \forall (i, j, k) \in \mathcal{T}_K \end{aligned}$$

where $\mathcal{T}_K$ is the set of $3\binom{K}{3}$ directed triples of indices corresponding to a triangle inequality constraint. This is a quadratic program—i.e., a convex optimization problem with a quadratic objective and linear inequality constraints. To solve this problem, we use the open-source and highly optimized OSQP solver (Stellato et al., 2020). The number of inequality constraints grows cubically as K increases, so finding an exact solution may be computationally expensive for analyses of large collections of stochastic neural networks.

D.6.4 Reformulating the Bures metric

Here we will prove that Eqs. D.21,D.22 are equivalent. A similar statement is proved in Theorem 1 of Bhatia et al. (2019). Our proof relies on the following lemma.

Lemma D.5. *Let $\boldsymbol{X} \in \mathbb{R}^{n \times n}$ be a matrix with singular value decomposition $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$. Then $\boldsymbol{V}\boldsymbol{U}^\top = (\boldsymbol{X}^\top \boldsymbol{X})^{-1/2} \boldsymbol{X}^\top$.*

Proof. This follows from the construction of the singular value decomposition. First, recognize that $\boldsymbol{X}$ can be written as the product of an orthogonal $\boldsymbol{Q}$ and symmetric positive semidefinite

matrix, $\boldsymbol{P}$, as follows:

$$\boldsymbol{X} = \underbrace{\boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1/2}}_{=\boldsymbol{Q}}\underbrace{(\boldsymbol{X}^\top\boldsymbol{X})^{1/2}}_{=\boldsymbol{P}} \tag{D.61}$$

It is easy to check that $\boldsymbol{Q}^\top\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^\top = \boldsymbol{I}$. Now, since $\boldsymbol{P}$ is positive semidefinite, we have $\boldsymbol{P} = \boldsymbol{V}\boldsymbol{S}\boldsymbol{V}^\top$ for some orthogonal matrix $\boldsymbol{V}$ and nonnegative diagonal matrix $\boldsymbol{S}$. Defining $\boldsymbol{U} = \boldsymbol{Q}\boldsymbol{V}$, we arrive at the SVD of $\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{P} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$. Now we can see that:

$$\boldsymbol{U} = \boldsymbol{Q}\boldsymbol{V} = \boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1/2}\boldsymbol{V} \quad \Rightarrow \quad \boldsymbol{U}\boldsymbol{V}^\top = \boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1/2} \tag{D.62}$$

Taking the transpose of this we prove the lemma. □

Now we proceed to prove the main result.

Proposition D.6. *Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be two positive definite matrices. Then*

$$\min_{\boldsymbol{Q}\in\boldsymbol{O}} \|\boldsymbol{A}^{1/2} - \boldsymbol{Q}\boldsymbol{B}^{1/2}\|_F^2 = Tr[\boldsymbol{A} + \boldsymbol{B} - 2(\boldsymbol{A}^{1/2}\boldsymbol{B}\boldsymbol{A}^{1/2})^{1/2}] \tag{D.63}$$

Proof. The minimization over $\boldsymbol{Q}$ is an instance of the well-known orthogonal procrustes problem (Gower and Dijksterhuis, 2004). This has a closed form solution. Specifically, denoting the singular value decomposition of $\boldsymbol{B}^{1/2}\boldsymbol{A}^{1/2}$ as $\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$, we have:

$$\boldsymbol{Q}^* = \arg\min_{\boldsymbol{Q}\in\boldsymbol{O}} \|\boldsymbol{A}^{1/2} - \boldsymbol{Q}\boldsymbol{B}^{1/2}\|_F^2 = \boldsymbol{V}\boldsymbol{U}^\top \tag{D.64}$$

Now, by Theorem D.5 above, we have:

$$\boldsymbol{V}\boldsymbol{U}^\top = ((\boldsymbol{B}^{1/2}\boldsymbol{A}^{1/2})^\top(\boldsymbol{B}^{1/2}\boldsymbol{A}^{1/2}))^{-1/2}(\boldsymbol{B}^{1/2}\boldsymbol{A}^{1/2})^\top = (\boldsymbol{A}^{1/2}\boldsymbol{B}\boldsymbol{A}^{1/2})^{-1/2}\boldsymbol{A}^{1/2}\boldsymbol{B}^{1/2} \tag{D.65}$$

Plugging this into the original problem, we have:

$$\|\boldsymbol{A}^{1/2} - \boldsymbol{Q}^*\boldsymbol{B}^{1/2}\|_F^2 = \mathrm{Tr}[\boldsymbol{A} + \boldsymbol{B} - 2\boldsymbol{A}^{1/2}\boldsymbol{Q}^*\boldsymbol{B}^{1/2}] \tag{D.66}$$

$$= \mathrm{Tr}[\boldsymbol{A} + \boldsymbol{B} - 2\boldsymbol{A}^{1/2}(\boldsymbol{A}^{1/2}\boldsymbol{B}\boldsymbol{A}^{1/2})^{-1/2}\boldsymbol{A}^{1/2}\boldsymbol{B}] \tag{D.67}$$

Due to the cyclic trace property, this becomes:

$$\mathrm{Tr}[\boldsymbol{A} + \boldsymbol{B} - 2(\boldsymbol{A}^{1/2}\boldsymbol{B}\boldsymbol{A}^{1/2})^{-1/2}\boldsymbol{A}^{1/2}\boldsymbol{B}\boldsymbol{A}^{1/2}] = \mathrm{Tr}[\boldsymbol{A} + \boldsymbol{B} - 2(\boldsymbol{A}^{1/2}\boldsymbol{B}\boldsymbol{A}^{1/2})^{1/2}] \tag{D.68}$$

as claimed. □

E | Notation

Let non-boldface letters (e.g. N, γ) denote scalar constants. For $N \geqslant 2$, let $K_N := N(N+1)/2$. Let $\mathbb{R}^N$ denote N-dimensional Euclidean space equipped with the Euclidean norm, denoted $\|\cdot\|_2$. Let $\mathbb{R}^N_+$ denote the non-negative orthant in $\mathbb{R}^N$. Given $K \geqslant 2$, let $\mathbb{R}^{N\times K}$ denote the set of $N \times K$ real-valued matrices. Let $\mathbb{S}^N$ denote the set of $N \times N$ symmetric matrices and let $\mathbb{S}^N_{++}$ denote the set of $N \times N$ symmetric positive definite matrices.

Matrices are denoted using bold uppercase letters (e.g., $\mathbf{M}$) and vectors are denoted using bold lowercase letters (e.g., $\mathbf{v}$). Given a matrix $\mathbf{M}$, M_{ij} denotes the entry of $\mathbf{M}$ located at the i^{th} row and j^{th} column. Let $\mathbf{1} = [1, \ldots, 1]^\top$ denote the N-dimensional vector of ones. Let $\mathbf{I}_N$ denote the $N \times N$ identity matrix.

Given vectors $\mathbf{v}, \mathbf{w} \in \mathbb{R}^N$, define their Hadamard product by $\mathbf{v} \circ \mathbf{w} := (v_1 w_1, \ldots, v_N w_N) \in \mathbb{R}^N$. Define $\mathbf{v}^{\circ 2} := (v_1^2, \ldots, v_N^2) \in \mathbb{R}^N$.

Let $\langle \cdot \rangle_t$ denote expectation over $t = 1, 2, \ldots$.

The $\operatorname{diag}(\cdot)$ operator, similar to `numpy.diag()` or MATLAB's `diag()`, can either: 1) map a vector in $\mathbb{R}^K$ to the diagonal of a $K \times K$ matrix of zeros; or 2) map the diagonal entries of a $K \times K$ matrix to a vector in $\mathbb{R}^K$. The specific operation being used should be clear by context. For example, given a vector $\mathbf{v} \in \mathbb{R}^K$, define $\operatorname{diag}(\mathbf{v})$ to be the $K \times K$ diagonal matrix whose $(i,i)^{\text{th}}$ entry is equal to v_i, for $i = 1, \ldots, K$. Alternatively, given a square matrix $\mathbf{M} \in \mathbb{R}^{K\times K}$, define $\operatorname{diag}(\mathbf{M})$ to be the K-dimensional vector whose i^{th} entry is equal to M_{ii}, for $i = 1, \ldots, K$.

Bibliography

Abbott, L. F. and Dayan, P. (1999). The effect of correlated variability on the accuracy of a population code. *Neural Comput.*, 11(1):91–101.

Abbott, L. F., Varela, J. A., Sen, K., and Nelson, S. B. (1997). Synaptic Depression and Cortical Gain Control. *Science*, 275(5297):221–224.

Adrian, E. D. and Matthews, R. (1928a). The action of light on the eye. *The Journal of Physiology*, 65(3):273–298.

Adrian, E. D. and Matthews, R. (1928b). The action of light on the eye: Part III. The interaction of retinal neurones. *The Journal of Physiology*, 65(3):273.

Adrian, E. D. and Zotterman, Y. (1926). The impulses produced by sensory nerve-endings. *The Journal of Physiology*, 61(2):151–171.

Ahmed, N., Natarajan, T., and Rao, K. R. (1974). Discrete cosine transform. *IEEE Transactions on Computers*, 100(1):90–93.

Ainsworth, S. K., Hayase, J., and Srinivasa, S. (2022). Git re-basin: Merging models modulo permutation symmetries.

Aitken, K. and Mihalas, S. (2023). Neural population dynamics of computing with synaptic modulations. *Elife*, 12:e83035.

An, G. (1996). The Effects of Adding Noise During Backpropagation Training on a Generalization Performance. *Neural Computation*, 8(3):643–674.

Anstis, S., Verstraten, F. A., and Mather, G. (1998). The motion aftereffect. *Trends in cognitive sciences*, 2(3):111–117.

Aschner, A., Solomon, S. G., Landy, M. S., Heeger, D. J., and Kohn, A. (2018). Temporal Contingencies Determine Whether Adaptation Strengthens or Weakens Normalization. *Journal of Neuroscience*, 38(47):10129–10142.

Atick, J. J. and Redlich, A. N. (1992). What does the retina know about natural scenes? *Neural Computation*, 4:196–210.

Attneave, F. (1954). Some informational aspects of visual perception. *Psychological Review*, 61(3):183–193.

Averbeck, B. B., Latham, P. E., and Pouget, A. (2006). Neural correlations, population coding and computation. *Nature reviews neuroscience*, 7(5):358–366.

Ballé, J., Chou, P. A., Minnen, D., Singh, S., Johnston, N., Agustsson, E., Hwang, S. J., and Toderici, G. (2020). Nonlinear transform coding. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):339–353.

Bardes, A., Ponce, J., and LeCun, Y. (2022). VICReg: Variance-invariance-covariance regularization for self-supervised learning. *International Conference on Learning Representations*.

Barlow, H. B. (1961). Possible Principles Underlying the Transformations of Sensory Messages. In *Sensory Communication*, pages 216–234. The MIT Press.

Barlow, H. B. and Foldiak, P. (1989). Adaptation and decorrelation in the cortex. In *The Computing Neuron*, pages 54–72. Addison-Wesley.

Batty, E., Whiteway, M., Saxena, S., Biderman, D., Abe, T., Musall, S., Gillis, W., Markowitz, J., Churchland, A., Cunningham, J. P., Datta, S. R., Linderman, S., and Paninski, L. (2019). Behavenet: nonlinear embedding and bayesian neural decoding of behavioral videos. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Bell, A. J. and Sejnowski, T. J. (1996). The "independent components" of natural scenes are edge filters. *Vision Research*, 37:3327–3338.

Benucci, A., Saleem, A. B., and Carandini, M. (2013). Adaptation maintains population homeostasis in primary visual cortex. *Nature Neuroscience*, 16(6):724–729.

Bhatia, R., Jain, T., and Lim, Y. (2019). On the bures–wasserstein distance between positive definite matrices. *Expositiones Mathematicae*, 37(2):165–191.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Bonin, V., Mante, V., and Carandini, M. (2006). The statistical computation underlying contrast gain control. *Journal of Neuroscience*, 26(23):6346–6353.

Bonneel, N., Rabin, J., Peyré, G., and Pfister, H. (2015). Sliced and Radon Wasserstein Barycenters of Measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45.

Borg, I. and Groenen, P. J. (2005). *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge university press.

Brenner, N., Bialek, W., and de Ruyter van Steveninck, R. (2000). Adaptive Rescaling Maximizes Information Transmission. *Neuron*, 26(3):695–702.

Brickell, J., Dhillon, I. S., Sra, S., and Tropp, J. A. (2008). The metric nearness problem. *SIAM J. Matrix Anal. Appl.*, 30(1):375–396.

Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges.

Bull, D. and Zhang, F. (2021). *Intelligent Image and Video Compression: Communicating Pictures*. Academic Press, London.

Burkard, R., Dell'Amico, M., and Martello, S. (2012). *Assignment Problems*. Society for Industrial and Applied Mathematics.

Cai, M., Schuck, N. W., Pillow, J. W., and Niv, Y. (2016). A bayesian method for reducing bias in neural representational similarity analysis. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Carandini, M. and Heeger, D. J. (2012). Normalization as a canonical neural computation. *Nature Reviews Neuroscience*, 13(1):51–62.

Carandini, M. and Ringach, D. L. (1997). Predictions of a recurrent model of orientation selectivity. *Vision Research*, 37(21):3061–3071.

Carlsson, M. (2021). von neumann's trace inequality for Hilbert-Schmidt operators. *Expositiones Mathematicae*, 39(1):149–157.

Casazza, P. G., Kutyniok, G., and Philipp, F. (2013). Introduction to Finite Frame Theory. In Casazza, P. G. and Kutyniok, G., editors, *Finite Frames*, pages 1–53. Birkhäuser Boston, Boston.

Chance, F. S., Abbott, L. F., and Reyes, A. D. (2002). Gain modulation from background synaptic input. *Neuron*, 35(4):773–782.

Chapochnikov, N. M., Pehlevan, C., and Chklovskii, D. B. (2021). Normative and mechanistic model of an adaptive circuit for efficient encoding and feature extraction. *bioRxiv*.

Chung, S. and Abbott, L. F. (2021). Neural population geometry: An approach for understanding biological and artificial neural networks. *Curr. Opin. Neurobiol.*, 70:137–144.

Clifford, C. W., Wenderoth, P., and Spehar, B. (2000). A functional angle on some after-effects in cortical vision. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 267(1454):1705–1710.

Clifford, C. W. G., Webster, M. A., Stanley, G. B., Stocker, A. A., Kohn, A., Sharpee, T. O., and Schwartz, O. (2007). Visual adaptation: Neural, psychological and computational aspects. *Vision Research*, 47(25):3125–3131.

Coates, A., Ng, A., and Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27.

Cover, T. M. (1999). *Elements of information theory*. John Wiley & Sons.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Dapello, J., Feather, J., Le, H., Marques, T., Cox, D., McDermott, J., DiCarlo, J. J., and Chung, S. (2021). Neural population geometry reveals the role of stochasticity in robust perception. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 15595–15607. Curran Associates, Inc.

Dasgupta, S. and Long, P. M. (2005). Performance guarantees for hierarchical clustering. *Journal of Computer and System Sciences*, 70(4):555–569.

Dayan, P. and Abbott, L. F. (2005). *Theoretical neuroscience: computational and mathematical modeling of neural systems*. MIT press.

Degenhart, A. D., Bishop, W. E., Oby, E. R., Tyler-Kabara, E. C., Chase, S. M., Batista, A. P., and Yu, B. M. (2020). Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity. *Nature biomedical engineering*, 4(7):672–685.

Dhruv, N. and Carandini, M. (2014). Cascaded Effects of Spatial Adaptation in the Early Visual System. *Neuron*, 81(3):529–535.

Diedrichsen, J., Berlot, E., Mur, M., Schütt, H. H., Shahbazi, M., and Kriegeskorte, N. (2020). Comparing representational geometries using whitened unbiased-distance-matrix similarity.

Douglas, R. J. and Martin, K. A. (2007). Recurrent neuronal circuits in the neocortex. *Current Biology*, 17(13):R496–R500.

Dragoi, V., Sharma, J., and Sur, M. (2000). Adaptation-induced plasticity of orientation tuning in adult visual cortex. *Neuron*, 28(1):287–298.

Dryden, I. L. and Mardia, K. (2016). *Statistical shape analysis with applications in R*. John Wiley & Sons, Chichester, UK Hoboken, NJ.

Dryden, I. L. and Mardia, K. V. (1993). Multivariate shape analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 460–480.

Duong, L., Leavitt, M., Pieper, F., Sachs, A., and Martinez-Trujillo, J. (2019). A normalization circuit underlying coding of spatial attention in primate lateral prefrontal cortex. *eneuro*, 6(2).

Duong, L. R., Bredenberg, C., Heeger, D. J., and Simoncelli, E. P. (2023a). Adaptive coding efficiency in recurrent cortical circuits via gain control. *arXiv preprint arXiv:2305.19869*.

Duong, L. R., Li, B., Chen, C., and Han, J. (2023b). Multi-rate adaptive transform coding for video compression. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Duong, L. R., Lipshutz, D., Heeger, D. J., Chklovskii, D. B., and Simoncelli, E. P. (2023c). Adaptive whitening in neural populations with gain-modulating interneurons. *Proceedings of the 40th International Conference on Machine Learning, PMLR*, 202:8902–8921.

Duong, L. R., Simoncelli, E. P., Chklovskii, D. B., and Lipshutz, D. (2023d). Adaptive whitening with fast gain modulation and slow synaptic plasticity. *arXiv preprint arXiv:2308.13633*.

Duong, L. R., Zhou, J., Nassar, J., Berman, J., Olieslagers, J., and Williams, A. H. (2023e). Representational dissimilarity metric spaces for stochastic neural networks. In *International Conference on Learning Representations*.

Dwivedi, K. and Roig, G. (2019). Representation similarity analysis for efficient task taxonomy & transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Edelman, S., Grill-Spector, K., Kushnir, T., and Malach, R. (1998). Toward direct visualization of the internal shape representation space by fmri. *Psychobiology*, 26:309–321.

Eldar, Y. C. and Oppenheim, A. V. (2003). MMSE whitening and subspace whitening. *IEEE Transactions on Information Theory*, 49(7):1846–1851.

Ermolov, A., Siarohin, A., Sangineto, E., and Sebe, N. (2021). Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, pages 3015–3024. PMLR.

Fairhall, A. L., Lewen, G. D., and Bialek, W. (2001). Efficiency and ambiguity in an adaptive neural code. *Nature*, 412:787–792.

Ferguson, K. A. and Cardin, J. A. (2020). Mechanisms underlying gain modulation in the cortex. *Nature Reviews Neuroscience*, 21(2):80–92.

Feydy, J., Séjourné, T., Vialard, F.-X., Amari, S.-I., Trouve, A., and Peyré, G. (2019). Interpolating between optimal transport and MMD using sinkhorn divergences. In Chaudhuri, K. and Sugiyama, M., editors, *Proceedings of Machine Learning Research*, volume 89, pages 2681–2690. PMLR.

Finn, C., Rajeswaran, A., Kakade, S., and Levine, S. (2019). Online meta-learning. In *International Conference on Machine Learning*, pages 1920–1930. PMLR.

Friedrich, R. W. (2013). Neuronal computations in the olfactory system of zebrafish. *Annual review of neuroscience*, 36:383–402.

Friedrich, R. W. and Wanner, A. A. (2021). Dense circuit reconstruction to understand neuronal computation: focus on zebrafish. *Annual Review of Neuroscience*, 44:275–293.

Friedrich, R. W. and Wiechert, M. T. (2014). Neuronal circuits and computations: pattern decorrelation in the olfactory bulb. *FEBS letters*, 588(15):2504–2513.

Gallego, J. A., Perich, M. G., Chowdhury, R. H., Solla, S. A., and Miller, L. E. (2020). Long-term stability of cortical population dynamics underlying consistent behavior. *Nature neuroscience*, 23(2):260–270.

Ganguli, D. and Simoncelli, E. P. (2014). Efficient Sensory Encoding and Bayesian Inference with Heterogeneous Neural Populations. *Neural Computation*, 26(10):2103–2134. Publisher: MIT Press.

Gardner, R. J. (1995). *Geometric tomography*, volume 58. Cambridge University Press Cambridge.

Giridhar, S., Doiron, B., and Urban, N. N. (2011). Timescale-dependent shaping of correlation by olfactory bulb lateral inhibition. *Proceedings of the National Academy of Sciences*, 108(14):5843–5848.

Goffinet, J., Brudner, S., Mooney, R., and Pearson, J. (2021). Low-dimensional learned feature spaces quantify individual and group differences in vocal repertoires. *Elife*, 10.

Goris, R. L. T., Movshon, J. A., and Simoncelli, E. P. (2014). Partitioning neuronal variability. *Nat. Neurosci.*, 17(6):858–865.

Gower, J. C. and Dijksterhuis, G. B. (2004). *Procrustes problems*. Oxford University Press, Oxford New York.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773.

Gschwend, O., Abraham, N. M., Lagier, S., Begnaud, F., Rodriguez, I., and Carleton, A. (2015). Neuronal pattern separation in the olfactory bulb improves odor discrimination learning. *Nature Neuroscience*, 18(10):1474–1482.

Gutierrez, G. J. and Denève, S. (2019). Population adaptation in efficient balanced networks. *eLife*, 8.

Gutnisky, D. A. and Dragoi, V. (2008). Adaptive coding of visual information in neural populations. *Nature*, 452(7184):220–224.

Haxby, J. V., Guntupalli, J. S., Nastase, S. A., and Feilong, M. (2020). Hyperalignment: Modeling shared information encoded in idiosyncratic cortical topographies. *Elife*, 9:e56601.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Heeger, D. J. and Mackey, W. E. (2018). ORGaNICs: A Theory of Working Memory in Brains and Machines. *arXiv:1803.06288 [cs, q-bio]*.

Heeger, D. J. and Zemlianova, K. O. (2020). A recurrent circuit implements normalization, simulating the dynamics of V1 activity. *Proceedings of the National Academy of Sciences*, 117(36):22494–22505.

Hendrycks, D. and Dietterich, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*.

Hershenhoren, I., Taaseh, N., Antunes, F. M., and Nelken, I. (2014). Intracellular Correlates of Stimulus-Specific Adaptation. *Journal of Neuroscience*, 34(9):3303–3319.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Higgins, I., Chang, L., Langston, V., Hassabis, D., Summerfield, C., Tsao, D., and Botvinick, M. (2021). Unsupervised deep learning identifies semantic disentanglement in single inferotemporal face patch neurons. *Nat. Commun.*, 12(1):6456.

Howes, N. R. (1995). *Modern Analysis and Topology*. Springer, New York, NY, 1995 edition.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations*.

Hua, T., Wang, W., Xue, Z., Ren, S., Wang, Y., and Zhao, H. (2021). On feature decorrelation in self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9598–9608.

Hyvärinen, A. and Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural Networks*, 13(4-5):411–430.

Karl, W. C., Verghese, G. C., and Willsky, A. S. (1994). Reconstructing Ellipsoids from Projections. *CVGIP: Graphical Models and Image Processing*, 56(2):124–139.

Kepecs, A. and Fishell, G. (2014). Interneuron cell types are fit to function. *Nature*, 505:318–326.

Kessy, A., Lewin, A., and Strimmer, K. (2018). Optimal whitening and decorrelation. *The American Statistician*, 72(4):309–314.

Khemakhem, I., Kingma, D., Monti, R., and Hyvarinen, A. (2020). Variational autoencoders and nonlinear ica: A unifying framework. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2207–2217. PMLR.

Kim, K. J. and Rieke, F. (2003). Slow Na^+ inactivation and variance adaptation in salamander retinal ganglion cells. *Journal of Neuroscience*, 23(4):1506–1516.

King, J. L., Lowe, M. P., Stover, K. R., Wong, A. A., and Crowder, N. A. (2016). Adaptive Processes in Thalamus and Cortex Revealed by Silencing of Primary Visual Cortex during Contrast Adaptation. *Current Biology*, 26(10):1295–1300.

King, P. D., Zylberberg, J., and DeWeese, M. R. (2013). Inhibitory interneurons decorrelate excitatory cells to drive sparse code formation in a spiking model of V1. *Journal of Neuroscience*, 33(13):5475–5485.

Kingma, D. P. and Welling, M. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Ko, H., Hofer, S. B., Pichler, B., Buchanan, K. A., Sjöström, P. J., and Mrsic-Flogel, T. D. (2011). Functional specificity of local synaptic connections in neocortical networks. *Nature*, 473(7345):87–91.

Kohn, A. (2007). Visual Adaptation: Physiology, Mechanisms, and Functional Benefits. *Journal of Neurophysiology*, 97(5):3155–3164.

Kohn, A. and Movshon, J. A. (2003). Neuronal Adaptation to Visual Motion in Area MT of the Macaque. *Neuron*, 39(4):681–691.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. (2019). Similarity of neural network representations revisited. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of Machine Learning Research*, volume 97, pages 3519–3529, Long Beach, California, USA. PMLR.

Kriegeskorte, N. and Diedrichsen, J. (2016). Inferring brain-computational mechanisms with models of activity measurements. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 371(1705):20160278.

Kriegeskorte, N. and Diedrichsen, J. (2019). Peeling the onion of brain representations. *Annual review of neuroscience*, 42:407–432.

Kriegeskorte, N. and Douglas, P. K. (2019). Interpreting encoding and decoding models. *Curr. Opin. Neurobiol.*, 55:167–179.

Kriegeskorte, N., Mur, M., and Bandettini, P. (2008a). Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2:4.

Kriegeskorte, N., Mur, M., Ruff, D. A., Kiani, R., Bodurka, J., Esteky, H., Tanaka, K., and Bandettini, P. A. (2008b). Matching categorical object representations in inferior temporal cortex of man and monkey. *Neuron*, 60(6):1126–1141.

Kriegeskorte, N. and Wei, X.-X. (2021). Neural tuning and representational geometry. *Nature Reviews Neuroscience*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2009). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*.

Kuhn, H. W. (1973). A note on fermat's problem. *Math. Program.*, 4(1):98–107.

Lange, K. (2016). *MM Optimization Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Laughlin, S. (1981). A Simple Coding Procedure Enhances a Neuron's Information Capacity. *Zeitschrift fur Naturforschung C, Journal of Biosciences*, pages 910–2.

Ledoit, O. and Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88:365–411.

Lee, W.-C. A., Bonin, V., Reed, M., Graham, B. J., Hood, G., Glattfelder, K., and Reid, R. C. (2016). Anatomy and function of an excitatory network in the visual cortex. *Nature*, 532(7599):370–374.

Lien, A. D. and Scanziani, M. (2013). Tuned thalamic excitation is amplified by visual cortical circuits. *Nature neuroscience*, 16(9):1315–1323.

Linderman, S., Stock, C. H., and Adams, R. P. (2014). A framework for studying synaptic plasticity with neural spike train data. *Advances in Neural Information Processing Systems*, 27.

Lipshutz, D., Pehlevan, C., and Chklovskii, D. B. (2023). Interneurons accelerate learning dynamics in recurrent neural networks for statistical adaptation. *International Conference on Learning Representations*.

Locatello, F., Abbati, G., Rainforth, T., Bauer, S., Schölkopf, B., and Bachem, O. (2019). On the fairness of disentangled representations. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Lohse, M., Bajo, V. M., King, A. J., and Willmore, B. D. (2020). Neural circuits underlying auditory contrast gain control and their perceptual implications. *Nature Communications*, 11(1):324.

Lopes, R. G., Yin, D., Poole, B., Gilmer, J., and Cubuk, E. D. (2019). Improving robustness without sacrificing accuracy with patch gaussian augmentation.

Maheswaranathan, N., Williams, A., Golub, M., Ganguli, S., and Sussillo, D. (2019). Universality and individuality in neural dynamics across large populations of recurrent networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 15629–15641. Curran Associates, Inc.

Martin, S., Grimwood, P. D., and Morris, R. G. (2000). Synaptic plasticity and memory: an evaluation of the hypothesis. *Annual Review of Neuroscience*, 23(1):649–711.

Masse, N. Y., Yang, G. R., Song, H. F., Wang, X.-J., and Freedman, D. J. (2019). Circuit mechanisms for the maintenance and manipulation of information in working memory. *Nature Neuroscience*, 22(7):1159–1167.

Matthey, L., Higgins, I., Hassabis, D., and Lerchner, A. (2017). dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/.

Mohan, S., Vincent, J. L., Manzorro, R., Crozier, P., Fernandez-Granda, C., and Simoncelli, E. (2021). Adaptive denoising via gaintuning. *Advances in Neural Information Processing Systems*, 34:23727–23740.

Moreno-Bote, R., Beck, J., Kanitscheider, I., Pitkow, X., Latham, P., and Pouget, A. (2014). Information-limiting correlations. *Nature Neuroscience*, 17(10):1410–1417.

Movshon, J. A. and Lennie, P. (1979). Pattern-selective adaptation in visual cortical neurones. *Nature*, 278(5707):850–852.

Muller, J. R., Metha, A. B., Krauskopf, J., and Lennie, P. (1999). Rapid adaptation in visual cortex to the structure of images. *Science*, 285(5432):1405–1408.

Młynarski, W. F. and Hermundstad, A. M. (2021). Efficient and adaptive sensory codes. *Nature Neuroscience*, 24(7):998–1009.

Nagel, K. I. and Doupe, A. J. (2006). Temporal Processing and Adaptation in the Songbird Auditory Forebrain. *Neuron*, 51(6):845–859.

Niles-Weed, J. and Rigollet, P. (2022). Estimation of wasserstein distances in the spiked transport model. *Bernoulli*, to appear.

Olshausen, B. and Field, D. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609.

Patterson, C. A., Wissig, S. C., and Kohn, A. (2013). Distinct Effects of Brief and Prolonged Adaptation on Orientation Tuning in Primary Visual Cortex. *Journal of Neuroscience*, 33(2):532–543.

Pehlevan, C. and Chklovskii, D. B. (2015). A normative theory of adaptive dimensionality reduction in neural networks. *Advances in Neural Information Processing Systems*, 28.

Pehlevan, C. and Chklovskii, D. B. (2019). Neuroscience-Inspired Online Unsupervised Learning Algorithms: Artificial Neural Networks. *IEEE Signal Processing Magazine*, 36(6):88–96.

Pehlevan, C., Hu, T., and Chklovskii, D. B. (2015). A Hebbian/anti-Hebbian neural network for linear subspace learning: A derivation from multidimensional scaling of streaming data. *Neural Computation*, 27(7):1461–1495.

Pehlevan, C., Sengupta, A. M., and Chklovskii, D. B. (2018). Why do similarity matching objectives lead to Hebbian/anti-Hebbian networks? *Neural Computation*, 30(1):84–124.

Peyré, G. and Cuturi, M. (2019). Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

Polack, P.-O., Friedman, J., and Golshani, P. (2013). Cellular mechanisms of brain state–dependent gain modulation in visual cortex. *Nature Neuroscience*, 16(9):1331–1339.

Quiroga, M., Morris, A., and Krekelberg, B. (2016). Adaptation without Plasticity. *Cell Reports*, 17(1):58–68.

Raghu, M., Gilmer, J., Yosinski, J., and Sohl-Dickstein, J. (2017). Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 6076–6085. Curran Associates, Inc.

Rast, L. and Drugowitsch, J. (2020). Adaptation Properties Allow Identification of Optimized Neural Codes. In *Advances in Neural Information Processing Systems*.

Reinhold, K., Lien, A. D., and Scanziani, M. (2015). Distinct recurrent versus afferent dynamics in cortical visual processing. *Nature Neuroscience*, 18(12):1789–1797.

Rossi, L. F., Harris, K. D., and Carandini, M. (2020). Spatial connectivity matches direction selectivity in visual cortex. *Nature*, 588(7839):648–652.

Rubin, D., Van Hooser, S., and Miller, K. (2015). The Stabilized Supralinear Network: A Unifying Circuit Motif Underlying Multi-Input Integration in Sensory Cortex. *Neuron*, 85(2):402–417.

Rumyantsev, O. I., Lecoq, J. A., Hernandez, O., Zhang, Y., Savall, J., Chrapkiewicz, R., Li, J., Zeng, H., Ganguli, S., and Schnitzer, M. J. (2020). Fundamental bounds on the fidelity of sensory cortical coding. *Nature*, 580(7801):100–105.

Salinas, E. and Thier, P. (2000). Gain modulation: a major computational principle of the central nervous system. *Neuron*, 27(1):15–21.

Sanchez-Vives, M. V., Nowak, L. G., and McCormick, D. A. (2000). Membrane mechanisms underlying contrast adaptation in cat area 17in vivo. *Journal of Neuroscience*, 20(11):4267–4285.

Saul, A. B. and Cynader, M. (1989). Adaptation in single units in visual cortex: the tuning of aftereffects in the spatial domain. *Visual neuroscience*, 2(6):593–607.

Schwartz, O. and Simoncelli, E. P. (2001). Natural signal statistics and sensory gain control. *Nature Neuroscience*, 4(8):819–825.

Sejdinovic, D., Sriperumbudur, B., Gretton, A., and Fukumizu, K. (2013). Equivalence of distance-based and rkhs-based statistics in hypothesis testing. *The Annals of Statistics*, 41(5):2263–2291. Full publication date: October 2013.

Seninge, L., Anastopoulos, I., Ding, H., and Stuart, J. (2021). VEGA is an interpretable generative model for inferring biological network activity in single-cell transcriptomics. *Nat. Commun.*, 12(1):5684.

Shadlen, M. N., Britten, K. H., Newsome, W. T., and Movshon, J. A. (1996). A computational analysis of the relationship between neuronal and behavioral responses to visual motion. *J. Neurosci.*, 16(4):1486–1510.

Shahbazi, M., Shirali, A., Aghajan, H., and Nili, H. (2021). Using distance on the riemannian manifold to compare representations in brain and in models. *NeuroImage*, 239:118271.

Shen, L., Zhao, L., and Hong, B. (2015). Frequency-specific adaptation and its underlying circuit model in the auditory midbrain. *Frontiers in Neural Circuits*, 9.

Shi, J., Shea-Brown, E., and Buice, M. (2019). Comparison against task driven artificial neural networks reveals functional properties in mouse visual cortex. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 5764–5774. Curran Associates, Inc.

Sietsma, J. and Dow, R. J. (1991). Creating artificial neural networks that generalize. *Neural Networks*, 4(1):67–79.

Simoncelli, E. P. and Olshausen, B. A. (2001). Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24(1):1193–1216.

Solomon, S. G. and Kohn, A. (2014). Moving sensory adaptation beyond suppressive effects in single neurons. *Current Biology*, 24(20):R1012–R1022.

Srivastava, A. and Klassen, E. P. (2016). *Functional and shape data analysis*. Springer Series in Statistics. Springer, New York, NY, 1 edition.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Stellato, B., Banjac, G., Goulart, P., Bemporad, A., and Boyd, S. (2020). OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672.

Stocker, A. A. and Simoncelli, E. P. (2009). Visual motion aftereffects arise from a cascade of two isomorphic adaptation mechanisms. *Journal of Vision*, 9(9):9–9.

Székely, G. J. and Rizzo, M. L. (2013). Energy statistics: A class of statistics based on distances. *J. Stat. Plan. Inference*, 143(8):1249–1272.

Székely, G. J. and Rizzo, M. L. (2017). The energy of data. *Annual Review of Statistics and Its Application*, 4(1):447–479.

Tatro, N., Chen, P.-Y., Das, P., Melnyk, I., Sattigeri, P., and Lai, R. (2020). Optimizing mode connectivity via neuron alignment. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15300–15311. Curran Associates, Inc.

Teich, A. F. and Qian, N. (2010). V1 orientation plasticity is explained by broadly tuned feedforward inputs and intracortical sharpening. *Visual neuroscience*, 27(1-2):57–73.

Thanwerdas, Y. and Pennec, X. (2022). Bures-wasserstein minimizing geodesics between covariance matrices of different ranks. *arXiv preprint arXiv:2204.09928*.

Thrun, S. and Pratt, L. (2012). *Learning to Learn*. Springer Science & Business Media.

Tong, J., Hu, R., Xi, J., Xiao, Z., Guo, Q., and Yu, Y. (2018). Linear shrinkage estimation of covariance matrices using low-complexity cross-validation. *Signal Processing*, 148:223–233.

Trautmann, E. M., Stavisky, S. D., Lahiri, S., Ames, K. C., Kaufman, M. T., O'Shea, D. J., Vyas, S., Sun, X., Ryu, S. I., Ganguli, S., and Shenoy, K. V. (2019). Accurate estimation of neural population dynamics without spike sorting. *Neuron*, 103(2):292–308.e4.

Tsodyks, M., Pawelzik, K., and Markram, H. (1998). Neural networks with dynamic synapses. *Neural Computation*, 10(4):821–835.

Uffink, J. (1995). Can the maximum entropy principle be explained as a consistency requirement? *Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics*, 26(3):223–261.

Ullman, S. and Schechtman, G. (1982). Adaptation and gain normalization. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 216(1204):299–313.

van Hateren, J. and van der Schaaf, A. (1998). Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings: Biological Sciences*, 265(1394):359–366.

Villani, C. (2009). *Optimal Transport*. Springer Berlin Heidelberg.

Wainwright, M. J., Schwartz, O., and Simoncelli, E. P. (2001). Natural Image Statistics and Divisive Normalization: Modeling Nonlinearities and Adaptation in Cortical Neurons. In *Statistical Theories of the Brain*, page 22. MIT Press.

Walther, A., Nili, H., Ejaz, N., Alink, A., Kriegeskorte, N., and Diedrichsen, J. (2016). Reliability of dissimilarity measures for multi-voxel pattern analysis. *Neuroimage*, 137:188–200.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. (2020). Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*.

Wang, X.-J., Liu, Y., Sanchez-Vives, M. V., and McCormick, D. A. (2003). Adaptation and temporal decorrelation by single neurons in the primary visual cortex. *Journal of Neurophysiology*, 89(6):3279–3293.

Wanner, A. A. and Friedrich, R. W. (2020). Whitening of odor representations by the wiring diagram of the olfactory bulb. *Nature Neuroscience*, 23(3):433–442.

Weber, A. I., Krishnamurthy, K., and Fairhall, A. L. (2019). Coding Principles in Adaptation. *Annual Review of Vision Science*, 5:427–449.

Westerberg, J. A., Cox, M. A., Dougherty, K., and Maier, A. (2019). V1 microcircuit dynamics: altered signal propagation suggests intracortical origins for adaptation in response to visual repetition. *Journal of Neurophysiology*, 121(5):1938–1952.

Westrick, Z. M., Heeger, D. J., and Landy, M. S. (2016). Pattern Adaptation and Normalization Reweighting. *Journal of Neuroscience*, 36(38):9805–9816.

Whitmire, C. and Stanley, G. (2016). Rapid Sensory Adaptation Redux: A Circuit Perspective. *Neuron*, 92(2):298–315.

Wick, S. D., Wiechert, M. T., Friedrich, R. W., and Riecke, H. (2010). Pattern orthogonalization via channel decorrelation by adaptive networks. *Journal of Computational Neuroscience*, 28(1):29–45.

Williams, A. H., Kunz, E., Kornblith, S., and Linderman, S. W. (2021). Generalized shape metrics on neural representations. In *Advances in Neural Information Processing Systems*, volume 34.

Wilson, A. G. (2020). The case for bayesian deep learning.

Wyrick, D. and Mazzucato, L. (2021). State-dependent regulation of cortical processing speed via gain modulation. *Journal of Neuroscience*, 41(18):3988–4005.

Yaron, A., Hershenhoren, I., and Nelken, I. (2012). Sensitivity to Complex Statistical Regularities in Rat Auditory Cortex. *Neuron*, 76(3):603–615.

Yiltiz, H., Heeger, D. J., and Landy, M. S. (2020). Contingent adaptation in masking and surround suppression. *Vision Research*, 166:72–80.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. (2021). Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR.

Zhuang, X., Yang, Z., and Cordes, D. (2020). A technical review of canonical correlation analysis for neuroscience applications. *Human Brain Mapping*, 41(13):3807–3833.

Zucker, R. S. and Regehr, W. G. (2002). Short-term synaptic plasticity. *Annual Review of Physiology*, 64(1):355–405.