ESTIMATING SHAPE DISTANCES ON NEURAL REPRESENTATIONS WITH LIMITED SAMPLES

Dean A. Pospisil[1] **Brett W. Larsen**[2,3,4] **Sarah E. Harvey**[2,3] **Alex H. Williams**[2,3]
[1]Princeton University, Princeton, NJ, 08544; dp4846@princeton.edu
[2]New York University, Center for Neural Science, New York, NY, 10003
[3]Flatiron Institute, Center for Computational Neuroscience, New York, NY, 10010
[4]Flatiron Institute, Center for Computational Mathematics, New York, NY, 10010
{brettlarsen, sharvey, awilliams}@flatironinstitute.org

ABSTRACT

Measuring geometric similarity between high-dimensional network representations is a topic of longstanding interest to neuroscience and deep learning. Although many methods have been proposed, only a few works have rigorously analyzed their statistical efficiency or quantified estimator uncertainty in data-limited regimes. Here, we derive upper and lower bounds on the worst-case convergence of standard estimators of *shape distance*—a measure of representational dissimilarity proposed by Williams et al. (2021). These bounds reveal the challenging nature of the problem in high-dimensional feature spaces. To overcome these challenges, we introduce a new method-of-moments estimator with a tunable bias-variance tradeoff. We show that this estimator achieves substantially lower bias than standard estimators in simulation and on neural data, particularly in high-dimensional settings. Thus, we lay the foundation for a rigorous statistical theory for high-dimensional shape analysis, and we contribute a new estimation method that is well-suited to practical scientific settings.

1 INTRODUCTION

Many approaches have been proposed to quantify similarity in neural network representations. Some popular methods include canonical correlations analysis (Raghu et al., 2017), centered kernel alignment (CKA; Kornblith et al., 2019), representational similarity analysis (RSA; Kriegeskorte et al., 2008a), and shape metrics (Williams et al., 2021). Each of these approaches takes in a set of high-dimensional measurements—e.g., hidden layer activations or neurobiological responses—and outputs a (dis)similarity score. Shape distances additionally satisfy the triangle inequality, thus enabling downstream algorithms for clustering and regression that leverage metric space structure.

Here, we take a closer look at the estimation of shape distance in high-dimensional, noisy, and sample-limited regimes. While shape distances have numerous applications in the physical sciences (Rohlf & Slice, 1990; Goodall, 1991; Andrade et al., 2004; Kendall et al., 2009; Saito et al., 2015) the use of shape metrics and other measures of neural representational similarity has introduced statistical issues that have not been adequately addressed. Specifically, shape metrics are often applied to low-dimensional noiseless measurements (e.g., 3D digital scans of anatomy across animals; Rohlf & Slice 1990) whereas in the study of neural networks the applications have been high-dimensional (e.g., comparing neural activity between brain regions; Kriegeskorte et al. 2008b).

We demonstrate that the noise and high dimensionality of neural representations pose a substantial challenge to estimating representational similarity in sample-limited regimes. Yet, with the noteworthy exceptions of research on RSA (Cai et al., 2016; Walther et al., 2016; Schütt et al., 2023) and CKA (Murphy et al., 2024), there is little work on quantifying accuracy of estimators of representational similarity (e.g. by developing procedures to compute confidence intervals). This poses a serious obstacle to adoption of these methods, particularly in experimental neuroscience where there is a hard limit on the number of conditions that can be feasibly sampled (Shi et al., 2019; Williams & Linderman, 2021).

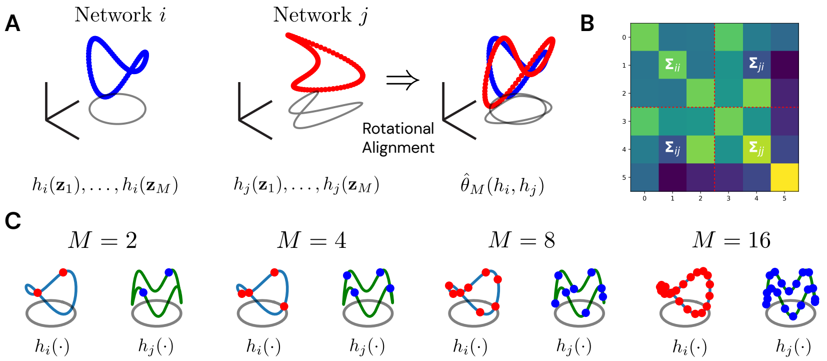

Figure 1: **(A)** Classical shape distances (Kendall et al., 2009) can be used to provide a rotation-invariant distance between neural representations (Williams et al., 2021). Given two labelled points clouds in N-dimensional space (*left* and *middle*), the distance is computed after an optimal orthogonal transformation is chosen to align the point clouds (*right*). In this visual example the point clouds trace out a low-dimensional manifold. **(B)** Heatmap shows the covariances ($\mathbf{\Sigma}_{ii}, \mathbf{\Sigma}_{jj}$) and cross-covariance ($\mathbf{\Sigma}_{ij}$) of the 3D representations in panel A. Shape distances can be re-expressed in terms of these quantities (see eq. 6, 7). **(C)** Our ability to estimate the shape distance is related to M, the number of stimuli. As M increases (*left* to *right*) the number of sampled points along the underlying manifold increases, and we are better able to resolve shape differences between the representations.

To address this challenge, we first obtain high-probability upper and lower bounds on the accuracy of typical "plug-in estimates" of shape distance. These bounds reveal these estimators are biased with low variance. The overall error decays with the number of sampled conditions, M, but the decay rate is inversely related to the number of dimensions, N. To combat these limitations, we propose a new method-of-moments estimator which, while not always strictly optimal compared with the plug-in estimator, provides an explicit and tunable tradeoff between estimator bias and variance.

2 Background and Problem Setting

We begin by considering a simple setting where each neural network is a deterministic map (for the stochastic setting, see section 3.4). A collection of K neural networks can then be viewed as a set of functions, each denoted $h_i : \mathcal{Z} \mapsto \mathbb{R}^N$ for $i \in \{1, \ldots, K\}$. Here, $\mathcal{Z}$ is a feature space and N can be interpreted as the number of neurons in each system (e.g. the size of a hidden layer in an artificial network, or the number of recorded neurons in a biological experiment).[1]

Let h_i and h_j denote neural systems which we assume are mean-centered and bounded:

$$\mathbb{E}[h_i(\boldsymbol{z})] = \mathbb{E}[h_j(\boldsymbol{z})] = \mathbf{0} \qquad \text{and} \qquad \|h_i(\boldsymbol{z})\|_2, \|h_j(\boldsymbol{z})\|_2 < B\sqrt{N} \quad \text{almost surely.} \tag{1}$$

for some constant $B > 0$. Here, the expectations are taken over $\boldsymbol{z} \sim P$, for some distribution P over network inputs. Our assumption that neural population rates are bounded by $B\sqrt{N}$ can result from assuming each neuron has a maximum firing rate equal to B. This assumption is common in the literature and reasonable in both artificial networks (since connection weights are finite) and biological networks (since neurons have a maximal firing rate).

Motivated by the shape theory literature (Goodall, 1991; Kent & Mardia, 1997; Kendall et al., 2009; Williams et al., 2021), we consider estimating the *Procrustes size-and-shape distance*, ρ, and *Riemannian shape distance*, θ, between neural representations. In our setting, these shape distances can

[1] The assumption that each layer has the same number of neurons is not essential. A theoretical connection with the Bures distance pointed out by Harvey et al. (2023) allows one to generalize shape distances to networks of dissimilar sizes. Indeed, we will see in eqs. (6) and (7) how shape distances can be expressed in terms of covariance and cross-covariance matrices that are well-defined in unequal dimensions.

be defined as (see App. D in Williams et al., 2021):

$$\rho(h_i, h_j) = \min_{\boldsymbol{Q} \in \mathcal{O}(N)} \sqrt{\mathbb{E}\|h_i(\boldsymbol{z}) - \boldsymbol{Q} h_j(\boldsymbol{z})\|_2^2} \tag{2}$$

$$\theta(h_i, h_j) = \min_{\boldsymbol{Q} \in \mathcal{O}(N)} \cos^{-1}\left(\frac{\mathbb{E}[h_i(\boldsymbol{z})^\mathsf{T} \boldsymbol{Q} h_j(\boldsymbol{z})]}{\sqrt{\mathbb{E}[h_i(\boldsymbol{z})^\mathsf{T} h_i(\boldsymbol{z})]\mathbb{E}[h_j(\boldsymbol{z})^\mathsf{T} h_j(\boldsymbol{z})]}}\right) \tag{3}$$

where $\mathcal{O}(N)$ denotes the set of $N \times N$ orthogonal matrices. Again, all expectations are taken over $\boldsymbol{z} \sim P$. Note that different notions of distance arise from different choices of input distribution, P.

To simplify our analysis and exposition, we will focus on estimating the *squared Procrustes distance*, ρ^2, and what we call the *cosine shape similarity*, $\cos\theta$. Thus, we ignore the square root term in eq. (2) and the arccosine term in eq. (3), but it should be kept in mind that one must apply these nonlinear functions to achieve a proper metric.

Properties of Shape Distance It is easy to verify that shape distances are invariant to rotations and reflections: that is, if $r : \mathbb{R}^N \mapsto \mathbb{R}^N$ is an orthogonal transformation, then for any function $h : \mathcal{Z} \mapsto \mathbb{R}^N$ representing a neural system we have $\rho(h, r \circ h) = \theta(h, r \circ h) = 0$, where '$\circ$' denotes function composition. Furthermore, ρ and θ are proper metrics, meaning that:

$$\rho(h_i, h_j) = \rho(h_j, h_i) \quad \text{and} \quad \rho(h_i, h_j) \leq \rho(h_i, h_k) + \rho(h_k, h_j) \quad \forall\, i, j, k \in \{1, \dots, K\}, \tag{4}$$

and likewise for θ. These properties are fundamental to rigorously establishing downstream analyses, such as for clustering networks with similar representations (Williams et al., 2021).

It is well-known that the optimal orthogonal alignment appearing in eqs. (2) and (3) can be identified in closed form, allowing us to write the Procrustes and Riemannian shape distances in terms of the covariance and cross-covariance matrices. We define the covariance (Σ_{ii} and Σ_{jj}) and cross-covariance matrices (Σ_{ij}) as

$$\boldsymbol{\Sigma}_{ii} = \mathbb{E}[h_i(\boldsymbol{z})h_i(\boldsymbol{z})^\mathsf{T}] \ , \ \boldsymbol{\Sigma}_{jj} = \mathbb{E}[h_j(\boldsymbol{z})h_j(\boldsymbol{z})^\mathsf{T}] \ , \ \boldsymbol{\Sigma}_{ij} = \mathbb{E}[h_i(\boldsymbol{z})h_j(\boldsymbol{z})^\mathsf{T}], \tag{5}$$

and reformulate the squared Procrustes distance and cosine shape similarity:

$$\rho^2(h_i, h_j) = \mathrm{Tr}[\boldsymbol{\Sigma}_{ii}] + \mathrm{Tr}[\boldsymbol{\Sigma}_{jj}] - 2\|\boldsymbol{\Sigma}_{ij}\|_* \tag{6}$$

$$\cos\theta(h_i, h_j) = \frac{\|\boldsymbol{\Sigma}_{ij}\|_*}{\sqrt{\mathrm{Tr}[\boldsymbol{\Sigma}_{ii}]\,\mathrm{Tr}[\boldsymbol{\Sigma}_{jj}]}} \tag{7}$$

where $\|\boldsymbol{\Sigma}_{ij}\|_*$ denotes the nuclear norm (or Shatten 1-norm) of the cross-covariance matrix:

$$\|\boldsymbol{\Sigma}_{ij}\|_* = \sum_{n=1}^{N} s_n(\boldsymbol{\Sigma}_{ij}) \tag{8}$$

where $s_1(\boldsymbol{M}) \geq \dots \geq s_N(\boldsymbol{M}) \geq 0$ denote the singular values of a matrix $\boldsymbol{M}$. Equations 6 and 7 are derived in Appendix A.1 to provide the reader with a self-contained narrative.

Plug-in Estimators Suppose we are given M independent and identically distributed network inputs $\boldsymbol{z}_1, \dots, \boldsymbol{z}_M \sim P$. How well can we approximate the shape distances between two networks, as a function of M? The standard approach (Williams et al., 2021), is to use a *plug-in estimator* in which one computes eqs. (2) and (3) after identifying the optimal $\boldsymbol{Q} \in \mathcal{O}(N)$. As we show in App. A.2, this is equivalent to estimating the squared Procrustes and cosine Riemannian distances by substituting the empirical covariances:

$$\hat{\boldsymbol{\Sigma}}_{ii} = \tfrac{1}{M}\sum_{m=1}^{M} h_i(\boldsymbol{z}_m)h_i(\boldsymbol{z}_m)^\mathsf{T},\ \hat{\boldsymbol{\Sigma}}_{jj} = \tfrac{1}{M}\sum_{m=1}^{M} h_j(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T},\ \hat{\boldsymbol{\Sigma}}_{ij} = \tfrac{1}{M}\sum_{m=1}^{M} h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T} \tag{9}$$

to approximate the true covariances appearing in eqs. (6) and (7). Thus,

$$\hat{\rho}^2(h_i, h_j) = \mathrm{Tr}[\hat{\boldsymbol{\Sigma}}_{ii}] + \mathrm{Tr}[\hat{\boldsymbol{\Sigma}}_{jj}] - 2\|\hat{\boldsymbol{\Sigma}}_{ij}\|_* \tag{10}$$

$$\cos\hat{\theta}(h_i, h_j) = \frac{\|\hat{\boldsymbol{\Sigma}}_{ij}\|_*}{\sqrt{\mathrm{Tr}[\hat{\boldsymbol{\Sigma}}_{ii}]\,\mathrm{Tr}[\hat{\boldsymbol{\Sigma}}_{jj}]}} \tag{11}$$

define plug-in estimators for the squared Procrustes and cosine Riemannian shape distances. The empirical behavior of these estimators as a function of M was only briefly characterized by Williams et al. (2021) for a pair of artificial networks trained on CIFAR-10.

3 Results

First, we theoretically characterize the accuracy of plug-in estimation as a function of the number of samples, M, and the dimension, N. We show that these estimators are biased and can converge at unfavorably slow rates under certain conditions. To overcome these issues, we introduce a new method-of-moments estimator in section 3.3 which has lower bias at the cost of increased variance.

3.1 Nonasymptotic bounds on the performance of plug-in estimation

First, it is straightforward to estimate $\mathrm{Tr}[\boldsymbol{\Sigma}_{ii}]$ and $\mathrm{Tr}[\boldsymbol{\Sigma}_{jj}]$. Their plug-in estimators are unbiased under our assumptions in eq. (1), and they rapidly converge to the correct answer. This is shown in the following lemma, whose proof relies only on classical concentration inequalities.

Lemma 1 (App. B.1)**.** *Under the assumptions in eq.* (1)*, with probability at least* $1 - \delta$*:*

$$\left|\mathrm{Tr}[\boldsymbol{\Sigma}_{ii}] - \mathrm{Tr}[\hat{\boldsymbol{\Sigma}}_{ii}]\right| \leq B^2 N M^{-1/2} \sqrt{\log(2/\delta)} \tag{12}$$

In contrast, the plug-in estimator for $\|\boldsymbol{\Sigma}_{ij}\|_*$ is biased upwards (see section 3.2) and turns out to converge more slowly. Using the Matrix Bernstein inequality (see Tropp, 2015), we can show:

Lemma 2 (App. B.2)**.** *Under the assumptions in eq.* (1)*, for any* M *and* N*:*

$$\mathbb{E}\left|\|\hat{\boldsymbol{\Sigma}}_{ij}\|_* - \|\boldsymbol{\Sigma}_{ij}\|_*\right| < \frac{2B^2N^2\log(2N)}{3M} + \frac{2B^2N^2\sqrt{\log(2N)}}{M^{1/2}} \tag{13}$$

This only upper bounds the expected error. However, the fluctuations around this expectation turn out to be small (see App. B.3), and so we are able to combine lemmas 1 and 2 into the following:

Theorem 1 (App. B.3)**.** *Under the assumptions in eq.* (1)*, with probability at least* $1 - \delta$

$$\frac{|\hat{\rho}^2 - \rho^2|}{N} \leq \frac{4B^2N\log(2N)}{3M} + \frac{4B^2N\sqrt{\log(2N)}}{M^{1/2}} + \left(\frac{3B^2}{M^{1/2}}\right)\sqrt{2\log\left(\frac{6}{\delta}\right)} \tag{14}$$

Theorem 1 states a non-asymptotic upper bound on the plug-in estimator's error that holds with high probability. We have expressed this bound on the squared size-and-shape Procrustes distance normalized by $1/N$, since the raw error, $|\hat{\rho} - \rho|$, will tend to increase linearly with N for an uninteresting reason—namely, since the the Procrustes shape distance is comprised of terms like $\mathrm{Tr}[\boldsymbol{\Sigma}_{ii}]$ and $\mathrm{Tr}[\boldsymbol{\Sigma}_{jj}]$. The choice of normalization in theorem 1 also makes the result more comparable to the cosine shape similiarity (eq. 7), which is normalized by a factor, $\sqrt{\mathrm{Tr}[\boldsymbol{\Sigma}_{ii}]\,\mathrm{Tr}[\boldsymbol{\Sigma}_{jj}]}$, of order N.

We can gain intuition for theorem 1 by ignoring logarithmic factors and noticing that the second term dominates. Then, roughly speaking, theorem 1 says that we can guarantee the plug-in error decreases as a function of $NM^{-1/2}$. Thus, for any fixed N, we need to increase M by a factor of 4 to decrease estimation error by a factor of 2. Further, when comparing higher-dimensional neural representations (i.e. higher N) we need to sample more landmarks—if N increases by a factor of 2, then M must be increased by a factor of 4 to compensate.

3.2 Failure modes of plug-in estimation and a lower bound on performance

Theorem 1 provides a high probability upper bound on the estimation error. A natural question is whether this upper bound is tight. To investigate, we seek an example where the plug-in estimator performs badly. We intuited that the plug-in estimates will have a large downward bias when two neural representations are very far apart in shape space. This can be understood in two ways. First, from the definitions of ρ and θ in eqs. (2) and (3), we see that both expressions contain a minimization over $\boldsymbol{Q} \in \mathcal{O}(N)$. For large N and small M, this high-dimensional orthogonal matrix can be "overfit" to the M observations resulting in an underestimate of distance. Second, from the alternative formulations in eqs. (6) and (7), we see that the shape distance is large if the true cross-covariance is "small" as quantified by the nuclear norm. In the extreme case where the singular values of $\boldsymbol{\Sigma}_{ij}$ are all zero, the empirical cross-covariance matrix $(1/M)\sum_m h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\top$ will overestimate the nuclear norm, and therefore underestimate the shape distance. This is more severe when M is small, since there are fewer terms in the sum to "average out" spurious correlations, which are particularly problematic in high dimensions (i.e. when N is large).

This intuition led us to construct an example where plug-in estimation error approaches the upper bound in theorem 1. This is summarized in the following result.

Theorem 2 (Lower Bound, App. B.4)**.** *Under the assumptions in eq.* (1)*, there exist neural networks and a distribution over inputs such that in the limit that $N \to \infty$ and $M \gg N$:*

$$\frac{|\hat{\rho}^2 - \rho^2|}{N} = \frac{16B^2}{3\pi} N^{1/2} M^{-1/2} \tag{15}$$

In Appendix B.5 we show the validity of this lower bound on simulated data. Although the bound is asymptotic, it gives a highly accurate approximation to the observed plug-in error for reasonable values of M and N (see Fig. 5). Thus, while future work may seek to improve the upper bound in theorem 1, we cannot hope to improve beyond the lower bound formulated above. If we ignore constant factors and logarithmic terms to gain intuition, we observe there is (roughly) a gap of $N^{1/2}$ between the upper and lower bounds. Thus, it is possible that our analysis in section 3.1 may be conservative in terms of the ambient dimension. That is, to compensate for a two-fold increase in N, theorem 2 only shows a case where M needs to be increased two-fold, in contrast to the four-fold increase suggested by theorem 1. However, in terms of the number of sampled inputs, the lower and upper bounds match: thus, the rate cannot be improved beyond $M^{-1/2}$.

3.3 A NEW ESTIMATOR WITH CONTROLLABLE BIAS

The plug-in estimator of $\|\mathbf{\Sigma}_{ij}\|_*$ has low variance but large and slowly decaying bias (see theorems 1 and 2). Here we develop an alternative estimator that is nearly unbiased.

First, note that the eigenvalues of $\mathbf{\Sigma}_{ij}\mathbf{\Sigma}_{ij}^{\mathsf{T}}$ correspond to the squared singular values of $\mathbf{\Sigma}_{ij}$. Thus, $\mathrm{Tr}[(\mathbf{\Sigma}_{ij}\mathbf{\Sigma}_{ij}^{\mathsf{T}})^{1/2}] = \|\mathbf{\Sigma}_{ij}\|_*$, and so we can reduce our problem to estimating the trace of $(\mathbf{\Sigma}_{ij}\mathbf{\Sigma}_{ij}^{\mathsf{T}})^{1/2}$, which is symmetric. Leveraging ideas from a well-developed literature (Adams et al., 2018), we proceed to define the p^{th} moment of this matrix as:

$$W_p = \mathrm{Tr}[(\mathbf{\Sigma}_{ij}\mathbf{\Sigma}_{ij}^{\mathsf{T}})^p] = \sum_{n=1}^{N} \lambda_n^p \tag{16}$$

where $\lambda_1, \ldots, \lambda_N$ denote the eigenvalues of $\mathbf{\Sigma}_{ij}\mathbf{\Sigma}_{ij}^{\mathsf{T}}$. Now, for any function $f : \mathbb{R} \mapsto \mathbb{R}$ and symmetric matrix $\boldsymbol{S}$ with eigenvalues $\lambda_1, \ldots, \lambda_N$, we define[2] $\mathrm{Tr}[f(\boldsymbol{S})] = \sum_i f(\lambda_i)$. So long as f is reasonably well-behaved, we can approximate it using a truncated power series with P terms. Thus, with $\boldsymbol{S} = \mathbf{\Sigma}_{ij}\mathbf{\Sigma}_{ij}^{\mathsf{T}}$ and $f(x) = \sqrt{x}$:

$$\|\mathbf{\Sigma}_{ij}\|_* = \mathrm{Tr}[(\mathbf{\Sigma}_{ij}\mathbf{\Sigma}_{ij}^{\mathsf{T}})^{1/2}] \approx \sum_{n=1}^{N}\sum_{p=0}^{P} \gamma_p \lambda_n^p = \sum_{p=0}^{P} \gamma_p \sum_{n=1}^{N} \lambda_n^p = \sum_{p=0}^{P} \gamma_p W_p \tag{17}$$

where $\gamma_0, \ldots, \gamma_P$ are scalar coefficients.

In summary, we can estimate $\|\mathbf{\Sigma}_{ij}\|_*$ by (a) specifying an estimator of the top eigenmoments, $W_1, \ldots, W_P$, and (b) specifying a desired set of scalar coefficients $\gamma_0, \ldots, \gamma_P$. To estimate the eigenmoments, we adapt procedures described by Kong & Valiant (2017) to obtain unbiased estimates for each moment, $\hat{W}_1, \ldots, \hat{W}_P$ (see App. C). To select the scalar coefficients, we propose an optimization procedure that trades off between bias and variance in the estimate of $\|\mathbf{\Sigma}_{ij}\|_*$. Our starting point is the usual bias-variance decomposition:

$$\mathbb{E}\left[\left(\|\mathbf{\Sigma}_{ij}\|_* - \textstyle\sum_p \gamma_p \hat{W}_p\right)^2\right] = \left(\mathbb{E}\left[\|\mathbf{\Sigma}_{ij}\|_* - \textstyle\sum_p \gamma_p \hat{W}_p\right]\right)^2 + \mathbb{V}\mathrm{ar}\left[\textstyle\sum_p \gamma_p \hat{W}_p\right]. \tag{18}$$

Since $\mathbb{E}[\hat{W}_p] = W_p = \sum_n \lambda_n^p$, the first term above (i.e. the "bias") simplifies and is upper-bounded:

$$\left(\mathbb{E}\left[\|\mathbf{\Sigma}_{ij}\|_* - \textstyle\sum_p \gamma_p \hat{W}_p\right]\right)^2 = \left(\textstyle\sum_n \left(\lambda_n^{1/2} - \sum_p \gamma_p \lambda_n^p\right)\right)^2 \leq \max_{0 \leq x \leq 1} \left(N\left(x^{1/2} - \textstyle\sum_p \gamma_p x^p\right)\right)^2$$

[2] This is a common convention to extend scalar functions (see e.g. Potters & Bouchaud, 2020, sec. 1.2.6).

The inequality follows from replacing each term in the sum over n with the worst case approximation error of the polynomial expansion (given here as the maximization over x). Thus, we seek to:

$$\underset{\gamma_0,\ldots,\gamma_P}{\text{minimize}} \quad \max_{0 \leq x \leq 1} \left(N \left(x^{1/2} - \textstyle\sum_p \gamma_p x^p \right) \right)^2 + \textstyle\sum_{p,p'} \gamma_p \gamma_{p'} \mathbb{C}\text{ov}(\hat{W}_p, \hat{W}_{p'}). \tag{19}$$

We estimate $\mathbb{C}\text{ov}(\hat{W}_p, \hat{W}_{p'})$ by bootstrapping—i.e. the empirical covariance of these statistics across re-sampled datasets where $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M\}$ are sampled with replacement. Given this estimate of covariance, eq. (19) can be cast as a convex quadratic program and the maximal bias can be bounded to a user defined limit at the expense of variance (see App. C.2). We use the maximal bias (eq. 19, term 1) and variance (eq. 19, term 2) to form approximate confidence intervals (see App. C.3).

3.4 Extension to stochastic networks

Thus far, we have modeled neural networks as deterministic mappings, $h_i : \mathcal{Z} \mapsto \mathbb{R}^N$. This assumption is not satisfied in biological data and in many artificial networks (e.g. VAEs). Here, we briefly explain how to extend the estimators to the stochastic setting. In this setting, the response of network i can be written as $h_i(\boldsymbol{z}) + \epsilon_i(\boldsymbol{z})$. As before, $h_i(\boldsymbol{z})$ is a deterministic mapping conditioned on a random variable $\boldsymbol{z} \sim P$. The "noise" term $\epsilon_i(\boldsymbol{z})$ is a mean-zero random variable that, in addition to inheriting the randomness of $\boldsymbol{z}$, captures the stochastic elements of each forward pass through the network (i.e. trial-to-trial variability even when the stimulus is fixed). Importantly, noise contributions are independent and identically distributed for each pass through the network.

Given a second stochastic network with same structure, $h_j(\boldsymbol{z}) + \epsilon_j(\boldsymbol{z})$, our goal is to estimate the shape distances eqs. (2) and (3) as before, effectively ignoring contributions of the "noise" terms $\epsilon_i(\cdot)$ and $\epsilon_j(\cdot)$. Ignoring these terms is not wholly justified, since it is of great interest to quantify how noise varies across networks (Duong et al., 2023). Nonetheless, it is useful to develop metrics that isolate the "signal" component of neural representations, and a full development of methods to quantify similarity in noise structure is outside the scope of this paper.

Our basic observation is that it suffices to consider two replicates for each network input. That is, let $\boldsymbol{z}' = \boldsymbol{z}$ where $\boldsymbol{z} \sim P$. Then, $\boldsymbol{\Sigma}_{ii} = \mathbb{E}[h_i(\boldsymbol{z}) h_i(\boldsymbol{z}')^\mathsf{T}]$ which can be approximated by the slightly reformulated plug-in estimator: $\hat{\boldsymbol{\Sigma}}_{ii} = (1/M) \sum_m h_i(\boldsymbol{z}_m) h_i(\boldsymbol{z}'_m)^\mathsf{T}$. Further, since noise is independent across networks, i.e. $\epsilon_i(\boldsymbol{z}) \perp\!\!\!\perp \epsilon_j(\boldsymbol{z})$ for all $\boldsymbol{z} \in \mathcal{Z}$, the cross-covariance estimators, including the method-of-moments estimator described in section 3.3, do not require any modification.

4 Applications and Experiments

4.1 Validation on synthetic data

We first validate our method-of-moments estimator (section 3.3) on simulated responses from a multivariate normal distribution. We estimate the cosine shape similarity, $\cos \theta$, defined in eq. 7. Our estimator of $\|\boldsymbol{\Sigma}_{ij}\|_*$ is the principle novelty; thus, it is informative to understand its properties in isolation. To achieve this, we use the ground truth covariance of $\hat{W}_p$ (instead of an estimate from a bootstrap) and use the ground truth values of $\text{Tr}[\boldsymbol{\Sigma}_{ii}]$ and $\text{Tr}[\boldsymbol{\Sigma}_{jj}]$. For details see App. D.1.

We first compared the bias of the plug-in estimator to that of the moment-based estimator across a range of ground truth shape similarity values (Fig. 2A). As expected from our intuition discussed in section 3.1, the plug-in estimator (blue line) tends to inflate estimated similarity when ground truth is low (left side of plot). The moment-based estimator (orange line), in contrast, performs well over the full range, at the cost of increases in estimator variance (blue vs orange error bars).

Next, we fixed the ground truth similarity at 0.2 and studied the effect of sample size, M (Fig. 2B). The moment estimator (constrained to 5% bias) maintains small bias even with small M, at the cost of high variance (orange error bars). Increasing M quickly reduces the variance of the estimator. A similar story emerges when we fix M and vary the dimension N (Fig. 2C). As the dimensionality increases, the plug-in estimator bias quickly explodes. In contrast, the moment estimator (constrained to 10% bias) has roughly constant bias; but it's variance grows with N. Thus our estimator bias outperforms the plug-in when the sample size is low and dimensionality is high.

Finally, an important property of the moment-based estimator is our ability to compute approximate confidence intervals (CI) (see App. C.3). We demonstrate 95% CIs across simulations in Figure 2D. These CIs are conservative, the true shape score is not within the CI's for only 2.3% of simulations.

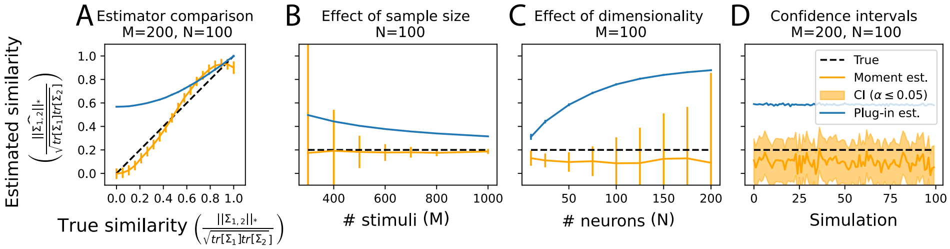

Figure 2: Validation of estimator on synthetic data. **(A)** The moment based estimator (orange) compared to plug-in estimator (blue) in simulation with standard deviation bars calculated across simulations. Estimators are evaluated at 20 linearly spaced ground truth similarity score values. **(B)** Effect of increasing sample size when moment estimator is constrained to have a bias less than 5 %. **(C)** Effect of increasing dimensionality. **(D)** Demonstration of conservative confidence intervals that account for variance and maximal bias of moment estimator. We do not include CIs for the plug in estimator (implied by theorem 1) because for small sample sizes, the theoretical bounds on estimator bias always contain far more than the entire allowable interval ($[0, 1]$).

Control of estimator bias Here we demonstrate the bias-variance tradeoff controlled by the upper-bound on bias defined by the user. The quadratic program in eq. (19) constrains the maximal absolute bias below a chosen constant (Fig. 3A, blue shaded area around true similarity score). The actual maximal bias will then be less than or equal to this user defined bias (cyan shaded area within blue). The expected value of the moment estimator stays within the maximal bias, in this case on its bound (orange trace). The user defined bias bound remains inactive until it is less than the MSE minimizing solution's bias (blue completely overlapped by cyan above 0.1). Variance then begins to increase as higher order W_p terms are weighted more to reduce bias (orange standard deviation bars from simulation increase as cyan region narrows). The mean of the estimator converges to ground truth as it is constrained by the bias bound (dotted orange line converges to dashed black). The plug-in estimator exceeds the maximal bias of the moment estimator (blue trace above cyan area).

Intuition for the moment estimator can be drawn from plots of solutions to the polynomial approximation (eq. 17, Fig. 3B, orange trace approximates black dashed) of the squared singular values of $\Sigma_{1,2}$ (black points all overlapping). Here we have re-scaled the the vertical axis so that the deviation between the square root and polynomial approximation is exactly the bias of the moment estimator. In the case where bias is not constrained (associated with left most estimates in panel B) the approximation is poor (dashed-dot orange trace does not match dashed black trace). For these eigenvalues the the deviation is near the worst possible bias (distance from black point dashed dot orange line is nearly as far as any other vertical deviation between the traces), this is why the estimator in panel B sits at the bound of maximal possible bias. When the upper bound on bias is very small (far right of B) the approximation is very good (dashed orange overlaps dashed black) because higher order terms are used. Yet this results in very high variance (Fig. 3B).

4.2 Application to biological data

Here we investigate noisy non-Gaussian data where the covariance of the $\hat{W}_p$ and the denominator of the similarity score must be estimated from data. We do so by applying our estimator to neural data: calcium recordings from mouse primary visual cortex in responses to a set of 2,800 natural images repeated twice (Stringer et al., 2019). Our estimator became highly variable when applied to this data in part because of its low SNR (average SNR ≈ 0.1). We therefore restricted our analysis to the neurons with the highest SNR in each recording (80 neurons in each recording).

This dataset contains seven recordings from different animals, but the population responses are not directly comparable since each recording targets a different region of primary visual cortex containing neurons with different receptive field. Thus, even though the same images were shown across recording sessions, the recorded neurons were effectively responding to different cropped portions of the image. We therefore only quantified shape similarity between subsampled populations of neurons taken from the same recording session.

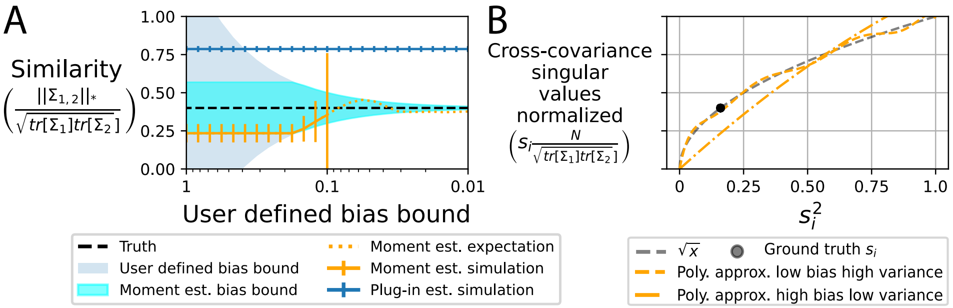

Figure 3: Control of bias-variance tradeoff with user defined bound on bias. **(A)** Moment based estimator expected value is constrained to be within the user defined bias bound (blue region) while minimizing worst case MSE (eq. (19)). Maximal bias can be less than the user defined bias (cyan region within blue). Lower bias leads to increased variance (orange trace converges to black dashed as SD bars widen). Where simulations become unstable we plot the theoretical expected value (dotted orange). Plug-in estimator is well outside bias bounds of moment estimator thus is more biased than moment estimator (blue trace outside cyan line). **(B)** Example plots of solutions to the quadratic program's approximation (orange traces) to square root (black dashed trace) of the eigenvalues of $\Sigma_{1,2}$ (black points). Re-scaling of singular values on vertical axis results in the deviation between the polynomial and the true square root evaluated at the true eigenvalues being exactly the bias of the associated estimates in panel A.

Determining the properties of the bias of our estimator requires comparison to the ground truth value of the similarity score. In the neural data, ground truth is unknown. We thus developed two sampling schemes to set the ground truth similarity in the neural data. To set similarity to 0 we measured similarity between different subpopulations of neurons ($N = 40$ neurons each) shown different stimuli (M=400 stimuli each), thus the two populations responses are independent, thus their cross covariance is 0 so that the similarity score is 0. To set the similarity to 1 we measured similarity between the same subpopulation of neurons ($N = 40$ neurons) shown the same stimuli ($M = 400$) but on different trials, thus the only deviation in their responses is owing to trial-to-trial variability, thus their tuning similarity is 1.

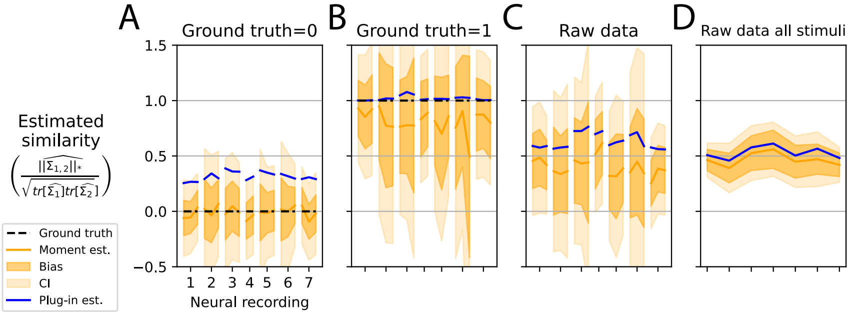

Figure 4: Validation of estimator on neural data (Stringer et al., 2019). **(A)** Comparison of estimators when ground truth similarity of neural data is set to 0. The estimator is applied to three disjoint sets of random stimuli for each recording ($n = 7$). The estimated maximal bias is plotted in dark orange area and the confidence interval, which includes bias, is plotted in light orange. **(B)** Same simulation as (A) except ground truth similarity is 1.**(C)** Same as (B) except estimation of true similarity. **(D)** Estimation of true similarity on all stimuli. ($M \approx 2,800$).

When the ground truth similarity was 0, the moment estimator correctly indicated this outcome (Fig. 4A, orange trace overlaps black dashed) and the confidence intervals always contained the true similarity (light orange contains black dashed). On the other hand the plug-in estimator was upwardly biased (blue above black dashed). Thus the moment based estimator can accurately determine when the similarity is low in noisy neural data whereas the plug-in estimator cannot.

When ground truth similarity was 1, we found the bias of the moment estimator was worse than that of the plug-in (Fig. 4B, blue overlaps black dashed, orange below). This is consistent with our synthetic simulations (see Fig. 2A far right). The CIs always contained the true value but contained nearly the entire possible range of similarity values. Thus while the average estimate is high our confidence intervals are so wide that we do not have much information about the true similarity.

Finally, we aimed to estimate the true shape similarity between these sub-populations of high SNR neurons ($N = 40$ neurons each). In Figure 4C, we show the estimated similarity across three independent folds of the stimulus set ($M = 400$ stimuli each). Across all seven recordings the moment estimator was near 0.5, but confidence intervals were wide so there is little information about similarity even for the highest SNR neurons (light orange extends from 0 to 1 on vertical axis). The plug-in estimator reports a higher degree of similarity, that we heavily discount given its upward bias (documented in Fig. 44A). When we included all stimuli ($M \approx 2800$) we obtained tighter confidence intervals, learning that the true similarity is most likely between 0.25 and 0.75 (Fig. 4D). Thus small populations of well-tuned neurons in the same brain region have only intermediate levels of representational similarity. Overall, we find noisy data is a challenging setting for reducing the bias of shape similarity estimates.

4.3 Application to artificial neural network representations

In Appendix E we apply the plug-in and moment-based estimator to penultimate layer representations between two ResNet-50 architectures (He et al., 2016) trained on ImageNet classification Deng et al. (2009). In this setting, we can accurately determine the ground truth by sampling a very large number of images (large M). However, for simulated analyses with small sample sizes (small M) we find that the plug-in estimator of similarity shows a positive bias, in agreement to our observations in Figure 4. In contrast, the moment-based estimator provides, on average, a better estimate of the shape similarity (albeit with higher variance across simulated analyses). We also observe that the bias of the plug-in estimator depends on how quickly the eigenvalues of the response covariance decays. (“effective dimensionality”; see e.g. Elmoznino & Bonner 2022). Thus, analyses of shape distance across large collections of networks risk contamination from confounding variables, such as effective dimensionality, in under-powered regimes. Overall, our observations on artificial network representations qualitatively agree with our simulated results (sec. 4.1) and analysis of biological data (sec. 4.2).

5 Discussion

There is a vast literature on measuring representational similarity between neural networks (see Klabunde et al., 2023, for review). Recent works have leveraged distance metrics that satisfy the triangle inequality (Williams et al., 2021; Lange et al., 2022; Duong et al., 2023), yet the statistical properties of these shape distances are understudied in high-dimensional settings. Here, we theoretically characterized “plug-in” estimates of shape distance in high-dimensional, noisy, and sample-limited regimes. We found that these estimates tend to over-estimate representational similarity when the true similarity is small. Further, they require a large number of samples, M, to overcome this bias in high-dimensional regimes. Theorems 1 and 2 provide precise guarantees on the worst-case performance of plug-in estimators. These bounds can guide the design of biological experiments, including pre-registered statistical power analysis. We strongly suspect that the dependence on N in these bounds is improvable (e.g. using techniques in Tropp 2015, the bounds can be refined to depend on the intrinsic dimension). However, in terms of the number of samples, the bounds definitively establish that the plug-in error decays at a rate proportional to $M^{-1/2}$.

An equally important contribution of our work is to provide a practical method to *(a)* reduce the bias of plug-in estimators of shape distance, *(b)* quantify uncertainty in shape distance estimates, and *(c)* enable practicioners to explicitly trade off estimator bias and variance. When employed on a biological dataset published by Stringer et al. (2019), we find that shape similarity estimates are highly uncertain, revealing the challenging nature of the problem in high dimensions and with noisy data. Importantly, this degree of uncertainty is not obvious from the procedures and plug-in estimates advertised by existing work on this subject.

In summary, our work characterizes the challenges of estimating shape distances in high-dimensional spaces. While shape distances can be well-behaved in certain settings (e.g. in noiseless artificial networks with many sampled conditions), our results suggest the need for carefully designed experiments and estimation procedures in sample-limited regimes.

ACKNOWLEDGEMENTS

This work was supported by the Center for Computational Neuroscience at the Flatiron Institute of the Simons Foundation. DAP was supported Simons Collaboration on the Global Brain (SCGB AWD543027). We thank Jonathan Pillow for useful feedback. The code for our project is available at `https://github.com/dp4846/eigmom_shape_stats/`.

REFERENCES

Ryan P. Adams, Jeffrey Pennington, Matthew J. Johnson, Jamie Smith, Yaniv Ovadia, Brian Patton, and James Saunderson. Estimating the spectral density of large implicit matrices, 2018.

Jose Manuel Andrade, María P. Gómez-Carracedo, Wojtek Krzanowski, and Mikael Kubista. Procrustes rotation in analytical chemistry, a tutorial. *Chemometrics and Intelligent Laboratory Systems*, 72(2):123–132, July 2004. ISSN 0169-7439. doi: 10.1016/j.chemolab.2004.01.007. URL `https://www.sciencedirect.com/science/article/pii/S0169743904000152`.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

Mingbo Cai, Nicolas W Schuck, Jonathan W Pillow, and Yael Niv. A bayesian method for reducing bias in neural representational similarity analysis. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL `https://proceedings.neurips.cc/paper/2016/file/b06f50d1f89bd8b2a0fb771c1a69c2b0-Paper.pdf`.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Lyndon R. Duong, Jingyang Zhou, Josue Nassar, Jules Berman, Jeroen Olieslagers, and Alex H. Williams. Representational dissimilarity metric spaces for stochastic neural networks. In *International Conference on Learning Representations*, 2023.

Eric Elmoznino and Michael F Bonner. High-performing neural network models of visual cortex benefit from high latent dimensionality. *bioRxiv*, pp. 2022–07, 2022.

Colin R. Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the royal statistical society series b-methodological*, 53:285–321, 1991. URL `https://api.semanticscholar.org/CorpusID:53315995`.

John C Gower and Garmt B Dijksterhuis. *Procrustes problems*, volume 30. OUP Oxford, 2004.

Sarah E. Harvey, Brett W. Larsen, and Alex H. Williams. Duality of Bures and Shape Distances with Implications for Comparing Neural Representations. November 2023. doi: 10.48550/arXiv.2311.11436. URL `http://arxiv.org/abs/2311.11436`. arXiv:2311.11436 [cs, stat].

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

D G Kendall, D Barden, T K Carne, and H Le. *Shape and Shape Theory*. John Wiley & Sons, September 2009.

John T. Kent and Kanti V. Mardia. Consistency of procrustes estimators. *Journal of the Royal Statistical Society. Series B (Methodological)*, 59(1):281–290, 1997. ISSN 00359246. URL `http://www.jstor.org/stable/2345930`.

Max Klabunde, Tobias Schumacher, Markus Strohmaier, and Florian Lemmerich. Similarity of neural network models: A survey of functional and representational measures, 2023.

Weihao Kong and Gregory Valiant. Spectrum estimation from samples. *The Annals of Statistics*, 45(5):2218 – 2247, 2017. doi: 10.1214/16-AOS1525. URL https://doi.org/10.1214/16-AOS1525.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. Representational similarity analysis - connecting the branches of systems neuroscience. *Front. Syst. Neurosci.*, 2:4, November 2008a.

Nikolaus Kriegeskorte, Marieke Mur, Douglas A Ruff, Roozbeh Kiani, Jerzy Bodurka, Hossein Esteky, Keiji Tanaka, and Peter A Bandettini. Matching categorical object representations in inferior temporal cortex of man and monkey. *Neuron*, 60(6):1126–1141, December 2008b.

Richard D Lange, Devin Kwok, Jordan Matelsky, Xinyue Wang, David S Rolnick, and Konrad P Kording. Neural networks as paths through the space of representations. *arXiv preprint arXiv:2206.10999*, 2022.

Alex Graeme Murphy, Joel Zylberberg, and Alona Fyshe. Correcting Biased Centered Kernel Alignment Measures in Biological and Artificial Neural Networks. March 2024. URL https://openreview.net/forum?id=E1NRrGtIHG.

Marc Potters and Jean-Philippe Bouchaud. *A First Course in Random Matrix Theory: for Physicists, Engineers and Data Scientists*. Cambridge University Press, 2020. doi: 10.1017/9781108768900.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

F. James Rohlf and Dennis Slice. Extensions of the Procrustes Method for the Optimal Superimposition of Landmarks. *Systematic Zoology*, 39(1):40–59, 1990. ISSN 0039-7989. doi: 10.2307/2992207. URL https://www.jstor.org/stable/2992207. Publisher: [Oxford University Press, Society of Systematic Biologists, Taylor & Francis, Ltd.].

Victor S. Saito, Alaide A. Fonseca-Gessner, and Tadeu Siqueira. How Should Ecologists Define Sampling Effort? The Potential of Procrustes Analysis for Studying Variation in Community Composition. *Biotropica*, 47(4):399–402, 2015. ISSN 0006-3606. URL https://www.jstor.org/stable/48574958. Publisher: [Association for Tropical Biology and Conservation, Wiley].

Heiko H Schütt, Alexander D Kipnis, Jörn Diedrichsen, and Nikolaus Kriegeskorte. Statistical inference on representational geometries. *eLife*, 12:e82566, aug 2023. ISSN 2050-084X. doi: 10.7554/eLife.82566. URL https://doi.org/10.7554/eLife.82566.

Jianghong Shi, Eric Shea-Brown, and Michael Buice. Comparison against task driven artificial neural networks reveals functional properties in mouse visual cortex. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/748d6b6ed8e13f857ceaa6cfbdca14b8-Paper.pdf.

Le Song, Alex Smola, Arthur Gretton, Karsten Borgwardt, and Justin Bedo. Supervised Feature Selection via Dependence Estimation, April 2007. URL http://arxiv.org/abs/0704.2668. arXiv:0704.2668 [cs].

Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Matteo Carandini, and Kenneth D Harris. High-dimensional geometry of population responses in visual cortex. *Nature*, 571(7765):361–365, 2019.

Joel A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015. ISSN 1935-8237. doi: 10.1561/2200000048. URL http://dx.doi.org/10.1561/2200000048.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

Alexander Walther, Hamed Nili, Naveed Ejaz, Arjen Alink, Nikolaus Kriegeskorte, and Jörn Diedrichsen. Reliability of dissimilarity measures for multi-voxel pattern analysis. *NeuroImage*, 137:188–200, 2016.

Alex H. Williams and Scott W. Linderman. Statistical neuroscience in the single trial limit. *Current Opinion in Neurobiology*, 70:193–205, 2021. ISSN 0959-4388. doi: https://doi.org/10.1016/j.conb.2021.10.008. URL https://www.sciencedirect.com/science/article/pii/S0959438821001203. Computational Neuroscience.

Alex H Williams, Erin Kunz, Simon Kornblith, and Scott Linderman. Generalized shape metrics on neural representations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 4738–4750. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/252a3dbaeb32e7690242ad3b556e626b-Paper.pdf.

A Appendix: Background on Generalized Shape Metrics

Here we provide several relevant derivations for generalized shape metrics. For a more thorough review, we direct the reader to Williams et al. (2021) for the foundational results on generalized shape metrics and Duong et al. (2023) for the extension to stochastic neural networks.

We can intuitively think of the Procrustes distance as the Euclidean distance between two vectors remaining when the rotations and reflections have been "removed". Similarly, the Riemannian shape distance can be thought of as the angle between two vectors after these rotations and reflections are removed. These definitions in eq. (2) and eq. (3) also make clear that Procrustes distance, like Euclidean distance, is sensitive to the overall scaling of h_i or h_j, while the Riemannian shape distance, like the angle between vectors, is scale-invariant.

A.1 Equivalence of eqs. (2) and (6); eqs. (3) and (7)

The squared Procrustes can be reformulated in terms of the covariance and cross-covariance matrices as follows:

$$\begin{aligned}
\rho^2(h_i, h_j) &= \min_{\boldsymbol{Q} \in \mathcal{O}(N)} \mathbb{E}\|h_i(\boldsymbol{z}) - \boldsymbol{Q}h_j(\boldsymbol{z})\|_2^2 \\
&= \min_{\boldsymbol{Q} \in \mathcal{O}(N)} \mathbb{E}\left[h_i(\boldsymbol{z})^\mathsf{T} h_i(\boldsymbol{z}) + h_j(\boldsymbol{z})^\mathsf{T} h_j(\boldsymbol{z}) - 2h_i(\boldsymbol{z})^\mathsf{T} \boldsymbol{Q} h_j(\boldsymbol{z})\right] \\
&= \mathbb{E}\left[h_i(\boldsymbol{z})^\mathsf{T} h_i(\boldsymbol{z})\right] + \mathbb{E}\left[h_j(\boldsymbol{z})^\mathsf{T} h_j(\boldsymbol{z})\right] - 2 \max_{\boldsymbol{Q} \in \mathcal{O}(N)} \mathbb{E}\left[h_i(\boldsymbol{z})^\mathsf{T} \boldsymbol{Q} h_j(\boldsymbol{z})\right] \\
&= \mathbb{E}\left[\mathrm{Tr}\left[h_i(\boldsymbol{z}) h_i(\boldsymbol{z})^\mathsf{T}\right]\right] + \mathbb{E}\left[\mathrm{Tr}\left[h_j(\boldsymbol{z}) h_j(\boldsymbol{z})^\mathsf{T}\right]\right] - 2 \max_{\boldsymbol{Q} \in \mathcal{O}(N)} \mathbb{E}\left[\mathrm{Tr}\left[\boldsymbol{Q} h_j(\boldsymbol{z}) h_i(\boldsymbol{z})^\mathsf{T}\right]\right] \\
&= \mathrm{Tr}\left[\mathbb{E}\left[h_i(\boldsymbol{z}) h_i(\boldsymbol{z})^\mathsf{T}\right]\right] + \mathrm{Tr}\left[\mathbb{E}\left[h_j(\boldsymbol{z}) h_j(\boldsymbol{z})^\mathsf{T}\right]\right] - 2 \max_{\boldsymbol{Q} \in \mathcal{O}(N)} \mathrm{Tr}\left[\boldsymbol{Q}\mathbb{E}\left[h_j(\boldsymbol{z}) h_i(\boldsymbol{z})^\mathsf{T}\right]\right] \\
&= \mathrm{Tr}\left[\boldsymbol{\Sigma}_{ii}\right] + \mathrm{Tr}\left[\boldsymbol{\Sigma}_{jj}\right] - 2 \max_{\boldsymbol{Q} \in \mathcal{O}(N)} \mathrm{Tr}\left[\boldsymbol{Q}\boldsymbol{\Sigma}_{ij}\right] \\
&= \mathrm{Tr}\left[\boldsymbol{\Sigma}_{ii}\right] + \mathrm{Tr}\left[\boldsymbol{\Sigma}_{jj}\right] - 2\|\boldsymbol{\Sigma}_{ij}\|_*
\end{aligned}$$

Similarly for the cosine Riemannian distance:

$$\begin{aligned}
\cos\theta(h_i, h_j) &= \max_{\boldsymbol{Q} \in \mathcal{O}(N)} \left(\frac{\mathbb{E}[h_i(\boldsymbol{z})^\mathsf{T} \boldsymbol{Q} h_j(\boldsymbol{z})]}{\sqrt{\mathbb{E}[h_i(\boldsymbol{z})^\mathsf{T} h_i(\boldsymbol{z})]\mathbb{E}[h_j(\boldsymbol{z})^\mathsf{T} h_j(\boldsymbol{z})]}} \right) \\
&= \frac{\max_{\boldsymbol{Q} \in \mathcal{O}(N)} \mathbb{E}\left[\mathrm{Tr}[\boldsymbol{Q} h_j(\boldsymbol{z}) h_i(\boldsymbol{z})^\mathsf{T}]\right]}{\sqrt{\mathbb{E}\left[\mathrm{Tr}[h_i(\boldsymbol{z}) h_i(\boldsymbol{z})^\mathsf{T}]\right]\mathbb{E}\left[\mathrm{Tr}[h_j(\boldsymbol{z}) h_j(\boldsymbol{z})^\mathsf{T}]\right]}} \\
&= \frac{\max_{\boldsymbol{Q} \in \mathcal{O}(N)} \mathrm{Tr}\left[\boldsymbol{Q}\mathbb{E}[h_j(\boldsymbol{z}) h_i(\boldsymbol{z})^\mathsf{T}]\right]}{\sqrt{\mathrm{Tr}\left[\mathbb{E}[h_i(\boldsymbol{z}) h_i(\boldsymbol{z})^\mathsf{T}]\right]\mathrm{Tr}\left[\mathbb{E}[h_j(\boldsymbol{z}) h_j(\boldsymbol{z})^\mathsf{T}]\right]}} \\
&= \frac{\max_{\boldsymbol{Q} \in \mathcal{O}(N)} \mathrm{Tr}\left[\boldsymbol{Q}\boldsymbol{\Sigma}_{ij}\right]}{\sqrt{\mathrm{Tr}\left[\boldsymbol{\Sigma}_{ii}\right]\mathrm{Tr}\left[\boldsymbol{\Sigma}_{jj}\right]}} = \frac{\|\boldsymbol{\Sigma}_{ij}\|_*}{\sqrt{\mathrm{Tr}\left[\boldsymbol{\Sigma}_{ii}\right]\mathrm{Tr}\left[\boldsymbol{\Sigma}_{jj}\right]}}
\end{aligned}$$

A.2 Reformulations of the Plug-in Estimator of Procrustes distance

Let $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M$ denote a set of independently and identically distributed samples in the network input space. Then, stack the responses of network i row-wise into a matrix $\boldsymbol{X}_i \in \mathbb{R}^{M \times N}$. Given this set up, a common definition of Procrustes distance is (Gower & Dijksterhuis, 2004):

$$\min_{\boldsymbol{Q} \in \mathcal{O}(N)} \frac{1}{\sqrt{M}} \|\boldsymbol{X}_i - \boldsymbol{X}_j \boldsymbol{Q}\|_F \tag{20}$$

Here, we have included a multiplying factor of $1/\sqrt{M}$ for reasons that will become clear shortly. Aside from this factor, the quantity above is how Williams et al. (2021) define the Procrustes distance. Below, we show that the square of this quantity is indeed the plug-in estimator we defined in eq. (10) in terms of the empirical covariance matrices:

$$\begin{aligned}\min_{\boldsymbol{Q}\in\mathcal{O}(N)} \frac{1}{M}\|\boldsymbol{X}_i - \boldsymbol{X}_j\boldsymbol{Q}\|_F^2 &= \min_{\boldsymbol{Q}\in\mathcal{O}(N)} \frac{1}{M}\left(\mathrm{Tr}[\boldsymbol{X}_i^\mathsf{T}\boldsymbol{X}_i] + \mathrm{Tr}[\boldsymbol{X}_j^\mathsf{T}\boldsymbol{X}_j] - 2\,\mathrm{Tr}[\boldsymbol{X}_i\boldsymbol{X}_j^\mathsf{T}\boldsymbol{Q}]\right)\\ &= \mathrm{Tr}\left[\tfrac{1}{M}\boldsymbol{X}_i^\mathsf{T}\boldsymbol{X}_i\right] + \mathrm{Tr}\left[\tfrac{1}{M}\boldsymbol{X}_j^\mathsf{T}\boldsymbol{X}_j\right] - 2\max_{\boldsymbol{Q}\in\mathcal{O}(N)}\mathrm{Tr}\left[\tfrac{1}{M}\boldsymbol{X}_i\boldsymbol{X}_j^\mathsf{T}\boldsymbol{Q}\right]\\ &= \mathrm{Tr}\left[\hat{\boldsymbol{\Sigma}}_{ii}\right] + \mathrm{Tr}\left[\hat{\boldsymbol{\Sigma}}_{jj}\right] - 2\max_{\boldsymbol{Q}\in\mathcal{O}(N)}\mathrm{Tr}\left[\hat{\boldsymbol{\Sigma}}_{ij}\boldsymbol{Q}\right]\\ &= \mathrm{Tr}\left[\hat{\boldsymbol{\Sigma}}_{ii}\right] + \mathrm{Tr}\left[\hat{\boldsymbol{\Sigma}}_{jj}\right] - 2\|\hat{\boldsymbol{\Sigma}}_{ij}\|_*\\ &= \hat{\rho}^2(h_i, h_j)\end{aligned}$$

B Appendix: Plug-in Estimator Theory

B.1 Proof of lemma 1

Here we show that the plug-in estimate of the total variance $\mathrm{Tr}[\hat{\boldsymbol{\Sigma}}_{ii}]$ converges to the true variance $\mathrm{Tr}[\boldsymbol{\Sigma}_{ii}]$ exponentially fast as M increases. We begin with some algebraic manipulations:

$$\begin{aligned}\left|\mathrm{Tr}[\boldsymbol{\Sigma}_{ii} - \hat{\boldsymbol{\Sigma}}_{ii}]\right| &= \left|\mathrm{Tr}\left[\mathbb{E}_{\boldsymbol{z}\sim P}[h_i(\boldsymbol{z})h_i(\boldsymbol{z})^\mathsf{T}] - \tfrac{1}{M}\sum_{m=1}^{M} h_i(\boldsymbol{z}_m)h_i(\boldsymbol{z}_m)^\mathsf{T}\right]\right|\\ &= \left|\mathbb{E}_{\boldsymbol{z}\sim P}\left[\mathrm{Tr}[h_i(\boldsymbol{z})h_i(\boldsymbol{z})^\mathsf{T}]\right] - \tfrac{1}{M}\sum_{m=1}^{M}\mathrm{Tr}[h_i(\boldsymbol{z}_m)h_i(\boldsymbol{z}_m)^\mathsf{T}]\right|\\ &= \left|\mathbb{E}_{\boldsymbol{z}\sim P}\left[\mathrm{Tr}[h_i(\boldsymbol{z})^\mathsf{T}h_i(\boldsymbol{z})]\right] - \tfrac{1}{M}\sum_{m=1}^{M}\mathrm{Tr}[h_i(\boldsymbol{z}_m)^\mathsf{T}h_i(\boldsymbol{z}_m)]\right|\\ &= \left|\mathbb{E}_{\boldsymbol{z}\sim P}\left[h_i(\boldsymbol{z})^\mathsf{T}h_i(\boldsymbol{z})\right] - \tfrac{1}{M}\sum_{m=1}^{M} h_i(\boldsymbol{z}_m)^\mathsf{T}h_i(\boldsymbol{z}_m)\right|\end{aligned}$$

where we have used the property $\mathrm{Tr}[\mathbf{x}\mathbf{x}^\mathsf{T}] = \mathbf{x}^\mathsf{T}\mathbf{x}$ for any column vector $\mathbf{x}$ in the last two lines.

The main assumption we are going to make is that the neural responses are constrained to an ℓ_2 ball of radius $B\sqrt{N}$ or equivalently $h_i(\boldsymbol{z})^\mathsf{T}h_i(\boldsymbol{z}) \leq B^2N$ for all stimuli $\boldsymbol{z}$ in the support of P. Note that this is a reasonable assumption in both biological (energy constraints) and artificial neural networks (weight decay common).

Lemma 3 (Bounded Random Variables are Sub-Gaussian, Wainwright (2019) Example 2.4)**.** *We say that a random variable X with mean μ is sub-Gaussian with parameter σ if:*

$$\mathbb{E}\left[e^{\lambda(X-\mu)}\right] \leq e^{\sigma^2\lambda^2/2} \quad \textit{for all } \lambda\in\mathbb{R}$$

Intuitively, this means that the tails of X fall off faster than a Gaussian. Furthermore, if X is mean zero and supported on the interval $[a, b]$, the X is sub-Gaussian with parameter $\sigma = (b - a)/2$.

Thus our assumption implies that each term with $\frac{1}{M}h_i(\boldsymbol{z}_m)^\mathsf{T}h_i(\boldsymbol{z}_m)$ is sub-Gaussian with parameter $\sigma = B^2N/2M$. We can then immediately apply the Hoeffding bound (Wainwright, 2019, Proposition 2.5) to obtain:

$$\mathbb{P}\left[\left|\mathrm{Tr}[\boldsymbol{\Sigma}_{ii} - \hat{\boldsymbol{\Sigma}}_{ii}]\right| \geq t\right] \leq 2\exp\left[-\frac{2Mt^2}{B^4N^2}\right]. \tag{21}$$

Analogously for term (B) we obtain:

$$\mathbb{P}\left[\left|\mathrm{Tr}[\boldsymbol{\Sigma}_{jj} - \hat{\boldsymbol{\Sigma}}_{jj}]\right| \geq t\right] \leq 2\exp\left[-\frac{2Mt^2}{B^4N^2}\right]. \tag{22}$$

B.2 Proof of lemma 2

Our main tool is the matrix Bernstein inequality, given as theorem 6.1.1 in Tropp (2015). We paraphrase a version of the theorem here to keep our narrative self-contained.

Theorem 3 (Matrix Bernstein). *Consider a finite sequence $\{\boldsymbol{S}_1, \ldots, \boldsymbol{S}_M\}$ of independent, random $N \times N$ matrices. Assume that:*

$$\mathbb{E}\big[\boldsymbol{S}_m\big] = \boldsymbol{0} \quad \textit{and} \quad \|\boldsymbol{S}_m\|_\infty \le L \quad \textit{for each index } m \tag{23}$$

where $\|\boldsymbol{S}_m\|_\infty = \sup\{\|\boldsymbol{S}_m \boldsymbol{v}\|_2 \,:\, \|\boldsymbol{v}\|_2 \le 1\}$ is the matrix operator norm.

Further, define the variance of the sum $\sum_m \boldsymbol{S}_m$ as:

$$V = \max\left\{\left\|\textstyle\sum_m \mathbb{E}\boldsymbol{S}_m^\mathsf{T}\boldsymbol{S}_m\right\|_\infty, \left\|\textstyle\sum_m \mathbb{E}\boldsymbol{S}_m\boldsymbol{S}_m^\mathsf{T}\right\|_\infty\right\} \tag{24}$$

Then:

$$\mathbb{E}\big[\, \|\textstyle\sum_m \boldsymbol{S}_m\|_\infty \,\big] \le \sqrt{2V\log(2N)} + \dfrac{L}{3}\log(2N) \tag{25}$$

We now turn to the proof of lemma 2. Define:

$$\boldsymbol{S}_m = \frac{1}{M}\left(h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T} - \boldsymbol{\Sigma}_{ij}\right) \tag{26}$$

for the sequence of network inputs $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M\}$. Notice that:

$$\mathbb{E}\big[\boldsymbol{S}_m\big] = \frac{1}{M}\left(\mathbb{E}\left[h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T}\right] - \boldsymbol{\Sigma}_{ij}\right) = \frac{1}{M}\left(\boldsymbol{\Sigma}_{ij} - \boldsymbol{\Sigma}_{ij}\right) = \boldsymbol{0} \tag{27}$$

Next, due to triangle inequality, we have:

$$\|\boldsymbol{S}_m\|_\infty = \frac{1}{M}\left\|h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T} - \boldsymbol{\Sigma}_{ij}\right\|_\infty \le \frac{1}{M}\underbrace{\left\|h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T}\right\|_\infty}_{(1)} + \frac{1}{M}\underbrace{\|\boldsymbol{\Sigma}_{ij}\|_\infty}_{(2)} \tag{28}$$

Terms (1) and (2) are each upper bounded by B^2N, since for term (1):

$$\begin{aligned}
\left\|h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T}\right\|_\infty &\le \left\|h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T}\boldsymbol{v}\right\|_2 && \text{(for any vector } \|\boldsymbol{v}\|_2 \le 1\text{)} && (29)\\
&= h_j(\boldsymbol{z}_m)^\mathsf{T}\boldsymbol{v}\,\|h_i(\boldsymbol{z}_m)\|_2 && && (30)\\
&\le \|h_j(\boldsymbol{z}_m)\|_2\,\|\boldsymbol{v}\|_2\,\|h_i(\boldsymbol{z}_m)\|_2 && \text{(Cauchy-Schwarz inequality)} && (31)\\
&\le B\sqrt{N}\cdot 1\cdot B\sqrt{N} = B^2N && \text{(From assumptions in eq. 1)} && (32)
\end{aligned}$$

And for term (2):

$$\begin{aligned}
\|\boldsymbol{\Sigma}_{ij}\|_\infty &= \left\|\mathbb{E}\,h_i(\boldsymbol{z})h_j(\boldsymbol{z})^\mathsf{T}\right\|_\infty && && (33)\\
&\le \left\|\mathbb{E}\,h_i(\boldsymbol{z})h_j(\boldsymbol{z})^\mathsf{T}\boldsymbol{v}\right\|_2 && \text{(for any vector } \|\boldsymbol{v}\|_2 \le 1\text{)} && (34)\\
&\le \mathbb{E}\left\|h_i(\boldsymbol{z})h_j(\boldsymbol{z})^\mathsf{T}\boldsymbol{v}\right\|_2 && \text{(Jensen's inequality)} && (35)\\
&\le B^2N && \text{(Repeat the upper bound on term 1)} && (36)
\end{aligned}$$

To summarize, we have:

$$\|\boldsymbol{S}_m\|_\infty \le \frac{1}{M}\left\|h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T}\right\|_\infty + \frac{1}{M}\|\boldsymbol{\Sigma}_{ij}\|_\infty \le \frac{2B^2N}{M} \tag{37}$$

That is, we have shown that the assumptions of eq. (23) are satisfied with $L = 2B^2N/M$.

Our next task is to determine an expression for the variance V defined in eq. (24). First, we have:

$$\begin{aligned}
\mathbb{E}\,\boldsymbol{S}_m^\mathsf{T}\boldsymbol{S}_m &= \frac{1}{M^2}\mathbb{E}[h_j(\boldsymbol{z}_m)h_i(\boldsymbol{z}_m)^\mathsf{T}h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T} + \boldsymbol{\Sigma}_{ij}^\mathsf{T}\boldsymbol{\Sigma}_{ij} - \boldsymbol{\Sigma}_{ij}^\mathsf{T}h_j(\boldsymbol{z}_m)h_i(\boldsymbol{z}_m)^\mathsf{T} - h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T}\boldsymbol{\Sigma}_{ij}]\\
&= \frac{1}{M^2}\mathbb{E}[h_j(\boldsymbol{z}_m)h_i(\boldsymbol{z}_m)^\mathsf{T}h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T}] + \boldsymbol{\Sigma}_{ij}^\mathsf{T}\boldsymbol{\Sigma}_{ij} - \boldsymbol{\Sigma}_{ij}^\mathsf{T}\mathbb{E}[h_j(\boldsymbol{z}_m)h_i(\boldsymbol{z}_m)^\mathsf{T}] - \mathbb{E}[h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T}]\boldsymbol{\Sigma}_{ij}\\
&= \frac{1}{M^2}\mathbb{E}[h_j(\boldsymbol{z}_m)h_i(\boldsymbol{z}_m)^\mathsf{T}h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T}] + \boldsymbol{\Sigma}_{ij}^\mathsf{T}\boldsymbol{\Sigma}_{ij} - \boldsymbol{\Sigma}_{ij}^\mathsf{T}\boldsymbol{\Sigma}_{ij} - \boldsymbol{\Sigma}_{ij}^\mathsf{T}\boldsymbol{\Sigma}_{ij}\\
&= \frac{1}{M^2}\mathbb{E}[h_j(\boldsymbol{z}_m)h_i(\boldsymbol{z}_m)^\mathsf{T}h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T}] - \boldsymbol{\Sigma}_{ij}^\mathsf{T}\boldsymbol{\Sigma}_{ij}
\end{aligned}$$

Then, by triangle inequality:

$$\begin{aligned}\|\mathbb{E}\,\boldsymbol{S}_m^\mathsf{T}\boldsymbol{S}_m\|_\infty &= \frac{1}{M^2}\|\mathbb{E}[h_j(\boldsymbol{z}_m)h_i(\boldsymbol{z}_m)^\mathsf{T}h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T}] - \boldsymbol{\Sigma}_{ij}^\mathsf{T}\boldsymbol{\Sigma}_{ij}\|_\infty \\ &\le \frac{1}{M^2}\underbrace{\|\mathbb{E}[h_j(\boldsymbol{z}_m)h_i(\boldsymbol{z}_m)^\mathsf{T}h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T}]\|_\infty}_{\text{(A)}} + \frac{1}{M^2}\underbrace{\|\boldsymbol{\Sigma}_{ij}^\mathsf{T}\boldsymbol{\Sigma}_{ij}\|_\infty}_{\text{(B)}}\end{aligned}$$

Terms (A) and (B) are each upper bounded by N^2. First, taking term (A):

$$\begin{aligned}\left\|\mathbb{E}\left[h_j(\boldsymbol{z}_m)h_i(\boldsymbol{z}_m)^\mathsf{T}h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T}\right]\right\|_\infty &\le \left\|\mathbb{E}\left[h_j(\boldsymbol{z}_m)h_i(\boldsymbol{z}_m)^\mathsf{T}h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T}\boldsymbol{v}\right]\right\|_2 && \text{(for } \|\boldsymbol{v}\| \le 1) \\ &\le \mathbb{E}\left\|h_j(\boldsymbol{z}_m)h_i(\boldsymbol{z}_m)^\mathsf{T}h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\mathsf{T}\boldsymbol{v}\right\|_2 && \text{(Jensen's)} \\ &\le \mathbb{E}\left[h_j(\boldsymbol{z}_m)^\mathsf{T}\boldsymbol{v}\|h_i(\boldsymbol{z}_m)\|_2^2\,\|h_j(\boldsymbol{z}_m)\|_2\right] \\ &\le \mathbb{E}\left[\|\boldsymbol{v}\|_2\|h_i(\boldsymbol{z}_m)\|_2^2\,\|h_j(\boldsymbol{z}_m)\|_2^2\right] && \text{(Cauchy-Schwarz)} \\ &\le 1\cdot B^2N\cdot B^2N = B^4N^2 && \text{(from eq. 1)}\end{aligned}$$

For term (B), we first note that $\|\boldsymbol{\Sigma}_{ij}^\mathsf{T}\boldsymbol{\Sigma}_{ij}\|_\infty \le \|\boldsymbol{\Sigma}_{ij}\|_\infty^2$ due to the fact that the operator norm is submultiplicative. Then, term (B) is upper bounded by B^4N^2 follows readily from:

$$\begin{aligned}\|\boldsymbol{\Sigma}_{ij}\|_\infty &= \|\mathbb{E}\,h_i(\boldsymbol{z})h_j(\boldsymbol{z})^\mathsf{T}\|_\infty \\ &\le \|\mathbb{E}\,h_i(\boldsymbol{z})h_j(\boldsymbol{z})^\mathsf{T}\boldsymbol{v}\|_2 && \text{(for } \|\boldsymbol{v}\| \le 1) \\ &\le \mathbb{E}\,\|h_i(\boldsymbol{z})h_j(\boldsymbol{z})^\mathsf{T}\boldsymbol{v}\|_2 && \text{(Jensen's)} \\ &\le \mathbb{E}\,\|h_i(\boldsymbol{z})\|_2\|h_j(\boldsymbol{z})\|_2\|\boldsymbol{v}\|_2 && \text{(Cauchy-Schwarz)} \\ &\le B\sqrt{N}\cdot B\sqrt{N}\cdot 1 = B^2N && \text{(from eq. 1)}\end{aligned}$$

Taking these two bounds together, we have shown $\|\mathbb{E}\,\boldsymbol{S}_m^\mathsf{T}\boldsymbol{S}_m\|_\infty \le 2B^4N^2/M^2$. We are now ready to upper bound the variance term, V, appearing in theorem 3. Specifically, by the triangle inequality and the bounds above, we have:

$$\|\textstyle\sum_m\mathbb{E}\,\boldsymbol{S}_m^\mathsf{T}\boldsymbol{S}_m\|_\infty \le \displaystyle\sum_{m=1}^{M}\|\mathbb{E}\,\boldsymbol{S}_m^\mathsf{T}\boldsymbol{S}_m\|_\infty \le \frac{2B^4N^2}{M}. \tag{38}$$

And it is easy to verify, using the same form of argument, that

$$\|\textstyle\sum_m\mathbb{E}\,\boldsymbol{S}_m\boldsymbol{S}_m^\mathsf{T}\|_\infty \le \displaystyle\sum_{m=1}^{M}\|\mathbb{E}\,\boldsymbol{S}_m\boldsymbol{S}_m^\mathsf{T}\|_\infty \le \frac{2B^4N^2}{M}. \tag{39}$$

Thus, $V \le 2B^4N^2/M$.

With this, we are equipped to apply the matrix Bernstein inequality to obtain an upper bound on the estimation error of the plug-in estimator. Specifically, we have:

$$\begin{aligned}\mathbb{E}\left|\|\hat{\boldsymbol{\Sigma}}_{ij}\|_* - \|\boldsymbol{\Sigma}_{ij}\|_*\right| &\le \mathbb{E}\|\hat{\boldsymbol{\Sigma}}_{ij} - \boldsymbol{\Sigma}_{ij}\|_* && \text{(reverse triangle inequality)} \\ &= \mathbb{E}\big[\|\sum_m \boldsymbol{S}_m\|_*\big] \\ &\le N\mathbb{E}\big[\|\sum_m \boldsymbol{S}_m\|_\infty\big] \\ &\le N\sqrt{2V\log(2N)} + \frac{NL}{3}\log(2N) && \text{(theorem 3)} \\ &\le 2B^2N^2M^{-1/2}\sqrt{\log(2N)} + \frac{2B^2N^2}{3M}\log(2N)\end{aligned}$$

Where we have substituted the derived quantities $L = 2B^2N/M$ and $V \le 2B^4N^2/M$ in the final line.

B.3 Proof of theorem 1

Lemma 2 provides an upper bound on the expected value of $\left| \|\boldsymbol{\Sigma}_{ij}\|_* - \|\hat{\boldsymbol{\Sigma}}_{ij}\|_* \right|$, which is the error of our plug-in estimate of cross-covariance nuclear norm. This bound holds for any true cross-covariance matrix $\boldsymbol{\Sigma}_{ij}$, provided that the constraints in eq. (1) are satisfied. However, this tells us nothing about how the estimation error deviates around its expectation.

Here, we use the bounded differences inequality (Wainwright, 2019, Corollary 2.21), also called McDiarmid's inequality, to show that deviations around this expectation decrease exponentially fast. Thus, the upper bound on the expected error (lemma 2) provides accurate intuition.

Lemma 4 (Bounded Differences Inequality, Wainwright (2019) Corollary 2.21)**.** *Consider a function $f : \mathbb{R}^n \to \mathbb{R}$. The function is said to have the bounded difference property for the kth coordinate if there exists an L_k for which the following holds:*

$$\max_{X_{1:n} \in \mathbb{R}^n, X'_k \in \mathbb{R}} \left| f(X_{1:n}) - f(X_{1:k-1}, X'_k, X_{k+1:n}) \right| \leq L_k$$

Suppose f satisfies this property with $L_1, \ldots, L_n$ for each coordinate respectively. Then the following inequality holds:

$$\mathbb{P}\left[\left| f(X_{1:n}) - \mathbb{E}[f(X_{1:n})] \right| \geq t \right] \leq \exp\left[-\frac{2t^2}{\sum_{i=1}^n L_i^2} \right] \tag{40}$$

We start by applying the reverse triangle inequality:

$$\left| \|\boldsymbol{\Sigma}_{ij}\|_* - \|\hat{\boldsymbol{\Sigma}}_{ij}\|_* \right| \leq \|\boldsymbol{\Sigma}_{ij} - \hat{\boldsymbol{\Sigma}}_{ij}\|_* = \left\| \boldsymbol{\Sigma}_{ij} - \tfrac{1}{M} \sum_{m=1}^M h_i(\boldsymbol{z}_m) h_j(\boldsymbol{z}_m)^{\mathsf{T}} \right\|_*$$

We can bound how much this changes if we change one coordinate of the function, i.e. if $h_i(\boldsymbol{z}_1)^{\mathsf{T}} h_j(\boldsymbol{z}_1)$ is replaced by $h_i(\tilde{\mathbf{z}}_1)^{\mathsf{T}} h_j(\tilde{\mathbf{z}}_1)$. The difference is then bounded by:

$$\begin{aligned}
&\left\| \boldsymbol{\Sigma}_{ij} - \tfrac{1}{M} \sum_{m=1}^M h_i(\boldsymbol{z}_m) h_j(\boldsymbol{z}_m)^{\mathsf{T}} \right\|_* - \left\| \boldsymbol{\Sigma}_{ij} - \left(\tfrac{1}{M} \sum_{m=1}^M h_i(\boldsymbol{z}_m) h_j(\boldsymbol{z}_m)^{\mathsf{T}} - \frac{1}{M} h_i(\boldsymbol{z}_1) h_j(\boldsymbol{z}_1)^{\mathsf{T}} + \frac{1}{M} h_i(\tilde{\mathbf{z}}_1) h_j(\tilde{\mathbf{z}}_1)^{\mathsf{T}} \right) \right\|_* \\
&\leq \left\| \frac{1}{M} \left(h_i(\boldsymbol{z}_1) h_j(\boldsymbol{z}_1)^{\mathsf{T}} - h_i(\tilde{\mathbf{z}}_1) h_j(\tilde{\mathbf{z}}_1)^{\mathsf{T}} \right) \right\|_* = \frac{1}{M} \left\| h_i(\boldsymbol{z}_1) h_j(\boldsymbol{z}_1)^{\mathsf{T}} - h_i(\tilde{\mathbf{z}}_1) h_j(\tilde{\mathbf{z}}_1)^{\mathsf{T}} \right\|_* \\
&\leq \frac{1}{M} \left(\left\| h_i(\boldsymbol{z}_1) h_j(\boldsymbol{z}_1)^{\mathsf{T}} \right\|_* + \left\| h_i(\tilde{\mathbf{z}}_1) h_j(\tilde{\mathbf{z}}_1)^{\mathsf{T}} \right\|_* \right) = \frac{1}{M} \left(\left| h_i(\boldsymbol{z}_1)^{\mathsf{T}} h_j(\boldsymbol{z}_1) \right| + \left| h_i(\tilde{\mathbf{z}}_1)^{\mathsf{T}} h_j(\tilde{\mathbf{z}}_1) \right| \right)
\end{aligned}$$

Finally, we can apply Cauchy-Schwartz and our assumption about the neural activations being bounded to obtain:

$$\begin{aligned}
\frac{1}{M} \left(\left| h_i(\boldsymbol{z}_1)^{\mathsf{T}} h_j(\boldsymbol{z}_1) \right| + \left| h_i(\tilde{\mathbf{z}}_1)^{\mathsf{T}} h_j(\tilde{\mathbf{z}}_1) \right| \right) &\leq \frac{1}{M} \left(\|h_i(\boldsymbol{z}_1)\|_2 \|h_j(\boldsymbol{z}_1)\|_2 + \|h_i(\tilde{\mathbf{z}}_1)\|_2 \|h_j(\tilde{\mathbf{z}}_1)\|_2 \right) \\
&\leq \frac{2B^2 N}{M}
\end{aligned}$$

Thus we have $\sum_{i=1}^M L_i^2 = \sum_{i=1}^M 4B^4 N^2 / M^2 = 4B^4 N^2 / M$, and we can apply the bounded differences inequality to obtain for all $t \geq 0$:

$$\mathbb{P}\left[\left| \left| \|\boldsymbol{\Sigma}_{ij}\|_* - \|\hat{\boldsymbol{\Sigma}}_{ij}\|_* \right| - \mathbb{E} \left| \|\boldsymbol{\Sigma}_{ij}\|_* - \|\hat{\boldsymbol{\Sigma}}_{ij}\|_* \right| \right| \geq t \right] \leq 2 \exp\left[-\frac{Mt^2}{2B^4 N^2} \right] \tag{41}$$

For the deviation from the expectation to be in the range $[-t, t]$ with probability $1 - \delta$ we require:

$$2 \exp\left[-\frac{Mt^2}{2B^4 N^2} \right] \leq \delta$$

Solving for t gives $t \geq B^2 N M^{-1/2}\sqrt{2\log(2/\delta)}$, and thus with probability $1-\delta$ the following holds:

$$\left| \left| \|\mathbf{\Sigma}_{ij}\|_* - \|\hat{\mathbf{\Sigma}}_{ij}\|_* \right| - \mathbb{E}\left| \|\mathbf{\Sigma}_{ij}\|_* - \|\hat{\mathbf{\Sigma}}_{ij}\|_* \right| \right| \leq B^2 N M^{-1/2}\sqrt{2\log(2/\delta)}$$

This also implies that the following holds with probability at least $1-\delta$:

$$\left| \|\mathbf{\Sigma}_{ij}\|_* - \|\hat{\mathbf{\Sigma}}_{ij}\|_* \right| \leq \frac{2B^2N^2\log(2N)}{3M} + \frac{2B^2N^2\sqrt{\log(2N)}}{M^{1/2}} + \frac{B^2N}{M^{1/2}}\sqrt{2\log\left(\frac{2}{\delta}\right)} \tag{42}$$

where we have plugged in our expectation bound from lemma 2.

To complete the proof we need to combine the above tail bound with lemma 1. By the triangle inequality we have

$$\begin{aligned} |\hat{\rho}^2 - \rho^2| &= \left| \mathrm{Tr}[\hat{\mathbf{\Sigma}}_{ii}] + \mathrm{Tr}[\hat{\mathbf{\Sigma}}_{jj}] - 2\|\hat{\mathbf{\Sigma}}_{ij}\|_* - \mathrm{Tr}[\hat{\mathbf{\Sigma}}_{ii}] - \mathrm{Tr}[\hat{\mathbf{\Sigma}}_{jj}] + 2\|\hat{\mathbf{\Sigma}}_{ij}\|_* \right| \\ &= \left| \mathrm{Tr}[\hat{\mathbf{\Sigma}}_{ii}] - \mathrm{Tr}[\mathbf{\Sigma}_{ii}] + \mathrm{Tr}[\hat{\mathbf{\Sigma}}_{jj}] - \mathrm{Tr}[\mathbf{\Sigma}_{jj}] + 2\|\mathbf{\Sigma}_{ij}\|_* - 2\|\hat{\mathbf{\Sigma}}_{ij}\|_* \right| \\ &\leq \left| \mathrm{Tr}[\hat{\mathbf{\Sigma}}_{ii}] - \mathrm{Tr}[\mathbf{\Sigma}_{ii}] \right| + \left| \mathrm{Tr}[\hat{\mathbf{\Sigma}}_{jj}] - \mathrm{Tr}[\mathbf{\Sigma}_{jj}] \right| + 2\left| \|\mathbf{\Sigma}_{ij}\|_* - \|\hat{\mathbf{\Sigma}}_{ij}\|_* \right| \end{aligned}$$

For convenience, we replace δ with $\delta/3$ in our results for these three terms, which yields that the following three inequalities independently hold with probability $\delta/3$:

$$\begin{aligned} &\left| \mathrm{Tr}[\mathbf{\Sigma}_{ii}] - \mathrm{Tr}[\hat{\mathbf{\Sigma}}_{ii}] \right| \geq B^2NM^{-1/2}\sqrt{\frac{1}{2}\log\left(\frac{6}{\delta}\right)} \\ &\left| \mathrm{Tr}[\mathbf{\Sigma}_{jj}] - \mathrm{Tr}[\hat{\mathbf{\Sigma}}_{jj}] \right| \geq B^2NM^{-1/2}\sqrt{\frac{1}{2}\log\left(\frac{6}{\delta}\right)} \\ &\left| \|\mathbf{\Sigma}_{ij}\|_* - \|\hat{\mathbf{\Sigma}}_{ij}\|_* \right| \geq \frac{2B^2N^2\log(2N)}{3M} + \frac{2B^2N^2\sqrt{\log(2N)}}{M^{1/2}} + \frac{B^2N}{M^{1/2}}\sqrt{2\log\left(\frac{6}{\delta}\right)} \end{aligned}$$

By applying the union bound, we obtain that all three inequalities hold simultaneously with probability $\leq \delta/3 + \delta/3 + \delta/3 = \delta$. The three reverse inequalities then hold simultaneously with probability greater than or equal to $1-\delta$. Thus with probability at least $1-\delta$, the following holds:

$$|\hat{\rho}^2 - \rho^2| \leq \frac{4B^2N^2\log(2N)}{3M} + \frac{4B^2N^2\sqrt{\log(2N)}}{M^{1/2}} + \frac{3B^2N}{M^{1/2}}\sqrt{2\log\left(\frac{6}{\delta}\right)}$$

as claimed in theorem 1.

B.4 PROOF OF THEOREM 2 (LOWER BOUND ON PLUG-IN ESTIMATOR ERROR)

We derive a lower bound by constructing an explicit example where the plug-in estimator performs badly. Specifically, we consider a scenario where two networks have entirely decorrelated, high-variance representations. To do this, we use *Rademacher random variables*. A random variable R is called a Rademacher variable if it behaves as follows:

$$R = \begin{cases} +1 & \text{with probability } 1/2 \\ -1 & \text{with probability } 1/2 \end{cases} \tag{43}$$

Now, suppose we sample M network inputs, $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M \sim P$, independently. Further, let $B > 0$ be the constant appearing in eq. (1). For $m \in \{1, \ldots, M\}$ define

$$X_m = \frac{1}{B}h_i(\boldsymbol{z}_m) \quad \text{and} \quad Y_m = \frac{1}{B}h_j(\boldsymbol{z}_m) \tag{44}$$

Note that X_m and Y_m are N-dimensional random vectors. Due to eq. (1), we have $\|h_i(\boldsymbol{z})\|_2 \leq B\sqrt{N}$ and $\|h_j(\boldsymbol{z})\|_2 \leq B\sqrt{N}$ almost surely. Thus, $\|X_m\| \leq \sqrt{N}$ and $\|Y_m\| \leq \sqrt{N}$ almost surely.

Define $X = (1/B)h_i(\boldsymbol{z})$ and $Y = (1/B)h_j(\boldsymbol{z})$ for randomly sampled $\boldsymbol{z} \sim P$. The case we will consider is that X and Y are each composed of N independent Rademacher variables. One trivial way to construct this is to suppose each $\boldsymbol{z} \sim P$ is a random vector with $2N$ elements, all of which are independent Rademacher variables scaled by a factor $B > 0$. Then, let $h_i : \mathbb{R}^{2N} \mapsto \mathbb{R}^N$ be the function which extracts the first N elements of $\boldsymbol{z}$ and let $h_j : \mathbb{R}^{2N} \mapsto \mathbb{R}^N$ be the function which extracts the final N elements.

Thus, we have constructed a setting where $X_1, \dots, X_M, Y_1, \dots, Y_M$ are all composed of independent Rademacher variables. In this setting, the squared Procrustes distance is given by:

$$\rho^2 = \mathrm{Tr}[\boldsymbol{\Sigma}_{ii}] + \mathrm{Tr}[\boldsymbol{\Sigma}_{jj}] - 2\|\boldsymbol{\Sigma}_{ij}\|_* \tag{45}$$

$$= \mathrm{Tr}[\mathbb{E}[h_i(\boldsymbol{z})h_i(\boldsymbol{z})^\mathsf{T}]] + \mathrm{Tr}[\mathbb{E}[h_j(\boldsymbol{z})h_j(\boldsymbol{z})^\mathsf{T}]] - 2\|\mathbb{E}[h_i(\boldsymbol{z})h_j(\boldsymbol{z})^\mathsf{T}]\|_* \tag{46}$$

$$= B^2 \cdot \left(\mathrm{Tr}[\mathbb{E}[XX^\mathsf{T}]] + \mathrm{Tr}[\mathbb{E}[YY^\mathsf{T}]] - 2\|\mathbb{E}[XY^\mathsf{T}]\|_*\right) \tag{47}$$

$$= B^2 \cdot \left(\mathbb{E}[X^\mathsf{T}X] + \mathbb{E}[Y^\mathsf{T}Y] - 2\|\mathbb{E}[X]\mathbb{E}[Y^\mathsf{T}]\|_*\right) \tag{48}$$

$$= B^2 \cdot (N + N - 0) \tag{49}$$

$$= 2B^2N \tag{50}$$

where we have used the fact that X and Y are independent, mean zero, random vectors to conclude that the cross covariance is an $N \times N$ matrix filled with zeros. Furthermore, note that $X_m^\mathsf{T}X_m = N$ and $Y_m^\mathsf{T}Y_m = N$ almost surely for all $m \in 1, \dots, M$ since they are comprised of N Rademacher variables. Thus, the plug-in estimate of the squared Procrustes distance takes the form:

$$\hat{\rho}^2 = B^2 \cdot \left(\mathrm{Tr}[\tfrac{1}{M}\textstyle\sum_m X_mX_m^\mathsf{T}] + \mathrm{Tr}[\tfrac{1}{M}\sum_m Y_mY_m^\mathsf{T}] - 2\|\tfrac{1}{M}\sum_m X_mY_m^\mathsf{T}\|_*\right) \tag{51}$$

$$= B^2 \cdot \left(\tfrac{1}{M}\textstyle\sum_m X_m^\mathsf{T}X_m + \tfrac{1}{M}\sum_m Y_m^\mathsf{T}Y_m - 2\|\tfrac{1}{M}\sum_m X_mY_m^\mathsf{T}\|_*\right) \tag{52}$$

$$= B^2 \cdot \left(N + N - 2\|\tfrac{1}{M}\textstyle\sum_m X_mY_m^\mathsf{T}\|_*\right) \tag{53}$$

$$= 2B^2N - 2B^2\|\tfrac{1}{M}\textstyle\sum_m X_mY_m^\mathsf{T}\|_* \tag{54}$$

Putting these two results together, we conclude that the absolute error of the plug-in estimator is:

$$|\rho^2 - \hat{\rho}^2| = 2B^2\|\tfrac{1}{M}\textstyle\sum_m X_mY_m^\mathsf{T}\|_* \tag{55}$$

Now, the product of two indepedent Rademacher variables is also a standard Rademacher variable. Thus, each element inside the matrix $(1/M)\sum_m X_mY_m^\mathsf{T}$, is the empirical average of M independent Rademacher variables. These matrix elements are asymptotically independent in the limit that $M \to \infty$. Further, the central limit theorem applies in this limit, and thus the distribution of each matrix element approaches a Gaussian distribution $\mathcal{N}(0, 1/M)$.

Such random matrices are well-studied under the name of Ginibre ensembles. In the limit that $N \to \infty$ and the variance of each matrix element is taken to be σ^2/N, the density of the singular values takes the following form (see e.g. Potters & Bouchaud, 2020, sec. 3.1.3):

$$\rho(s) = \frac{\sqrt{4\sigma^2 - s^2}}{\pi\sigma^2} \quad s \in (0, 2\sigma) \tag{56}$$

This is called the quarter circle law since if we look at the density of s it forms a quarter circle. The nuclear norm of the matrix is N times the expected value of s with with respect to the density $\rho(s)$. Integrating this density, we obtain:

$$\lim_{\substack{N\to\infty \\ M\gg N}} \left\|\tfrac{1}{M}\textstyle\sum_m X_mY_m^\mathsf{T}\right\|_* = \frac{N}{\pi\sigma^2}\int_0^{2\sigma} s\sqrt{4\sigma^2 - s^2}\,ds \tag{57}$$

$$= \frac{N}{4\pi\sigma^2}\left[-\frac{1}{3}(4\sigma^2 - s^2)^{3/2}\right]_0^{2\sigma} \tag{58}$$

$$= \frac{N}{\pi\sigma^2}\left[\frac{1}{3}(4\sigma^2)^{3/2}\right] = \frac{N}{\pi\sigma^2}\left[\frac{8}{3}\sigma^3\right] \tag{59}$$

$$= \frac{8\sigma}{3\pi}N = \frac{8}{3\pi}N^{3/2}M^{-1/2} \tag{60}$$

Where in the last line we have substituted $\sigma = \sqrt{N/M}$, which comes from equating σ^2/N (the variance in of each matrix element in eq. 56) with $1/M$ (the variance given by the average of M Rademacher variables under the central limit theorem). Note that the analysis above holds asymptotically as $M, N \to \infty$ and we keep $M \gg N$ so that the central limit theorem continues to hold.

Plugging eq. (60) into eq. (55) and dividing both sides by N we arrive at the expression appearing in theorem 2.

B.5 Numerical verification of lower bound

Figure 5 below shows in simulation that the plug-in estimator closely agrees with the lower bound established by theorem 2.

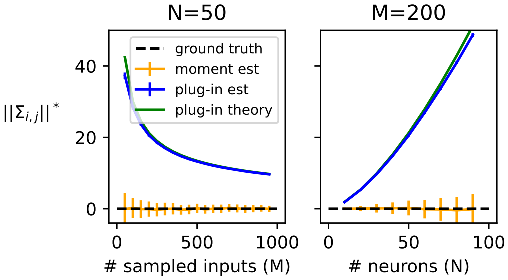

Figure 5: Simulation of lower bound on plug-in error. Simulated responses from N neurons were sampled from $\mathcal{N}(0, \boldsymbol{I})$ independently for pairs of networks over M stimuli. We plot the observed nuclear norm of the empirical cross-covariance matrix in blue, and show close agreement with equation 60 in green. The moment-based estimator is shown in yellow for comparison; note that the variance of the moment-based estimator decreases with M and increases with N. ***Left***, simulations from $N = 50$ neurons as M was varied. ***Right***, simulations from $M = 200$ stimuli as N was varied.

C Appendix: Method-of-Moments Estimator

C.1 Derivation of method-of-moment estimator

We now turn to constructing our method-of-moments estimator of $\|\boldsymbol{\Sigma}_{ij}\|_* = \sum_{n=1}^{N} s_n(\boldsymbol{\Sigma}_{ij})$, which is required for our novel estimator of the Riemannian shape distance. We can form an unbiased estimator of the matrix $\boldsymbol{\Sigma}_{ij}$ by observing a single random stimuli in the two networks:

$$\hat{\boldsymbol{\Sigma}}_{ijm} := h_i(\boldsymbol{z}_m)h_j(\boldsymbol{z}_m)^\top \in \mathbb{R}^{N \times N}, \quad \mathbb{E}[\hat{\boldsymbol{\Sigma}}_{ijm}] = \boldsymbol{\Sigma}_{ij}$$

Note that here the randomness comes from the selection of the stimuli, i.e. $\boldsymbol{z}_m \sim P$; the output of the network is deterministic. Assuming m, m' are distinct stimuli drawn independently from the distribution P, we then have:

$$\mathbb{E}\left[\hat{\boldsymbol{\Sigma}}_{ijm}\hat{\boldsymbol{\Sigma}}_{ijm'}\right] = \boldsymbol{\Sigma}_{ij}\boldsymbol{\Sigma}_{ij}^\top$$

This means we can estimate $\boldsymbol{\Sigma}_{ij}\boldsymbol{\Sigma}_{ij}^\top$ by observing a pair of stimuli in both networks.

$$
\begin{aligned}
\mathrm{Tr}\left[f(\mathbf{\Sigma}_{ij}\mathbf{\Sigma}_{ij}^{\mathsf{T}})\right] &= \sum_{n=1}^{N} f\left(s_n^2(\mathbf{\Sigma}_{ij})\right) = \sum_{n=1}^{N}\sum_{p=0}^{\infty}\gamma_p s_n^{2p}(\mathbf{\Sigma}_{ij}) && \text{Taylor expansion of } f(\cdot)\\
&= \sum_{p=0}^{\infty}\gamma_p \sum_{n=1}^{N} s_n^{2p}(\mathbf{\Sigma}_{ij}) = \sum_{p=0}^{\infty}\gamma_p\,\mathrm{Tr}\left[\left(\mathbf{\Sigma}_{ij}\mathbf{\Sigma}_{ij}^{\mathsf{T}}\right)^{p}\right] && \mathrm{Tr}\left[\left(\mathbf{\Sigma}_{ij}\mathbf{\Sigma}_{ij}^{\mathsf{T}}\right)^{p}\right] = \sum_{n=1}^{N} s_n^{2p}(\mathbf{\Sigma}_{ij})\\
&= \sum_{p=0}^{\infty}\gamma_p \mathbb{E}\left[\mathrm{Tr}\left[\prod_{\sigma=1}^{p}\hat{\mathbf{\Sigma}}_{ij(2\sigma-1)}\hat{\mathbf{\Sigma}}_{ij(2\sigma)}^{\mathsf{T}}\right]\right] && \text{Substitute unbiased estimator for } \left(\mathbf{\Sigma}_{ij}\mathbf{\Sigma}_{ij}^{\mathsf{T}}\right)^{p}\\
&\approx \sum_{p=0}^{P}\gamma_p \mathbb{E}\left[\mathrm{Tr}\left[\prod_{\sigma=1}^{p}\hat{\mathbf{\Sigma}}_{ij(2\sigma-1)}\hat{\mathbf{\Sigma}}_{ij(2\sigma)}^{\mathsf{T}}\right]\right] && \text{Approximate with truncated power series}
\end{aligned}
$$

Our estimator for the nuclear norm of $\mathbf{\Sigma}_{ij}$ is thus:

$$
\widehat{\|\mathbf{\Sigma}_{ij}\|_*} = \sum_{p=0}^{P}\gamma_p\,\mathrm{Tr}\left[\prod_{\sigma=1}^{p}\hat{\mathbf{\Sigma}}_{ij(2\sigma-1)}\hat{\mathbf{\Sigma}}_{ij(2\sigma)}^{\mathsf{T}}\right] \tag{61}
$$

Note that for each element of the product we are considering the estimator based on stimuli $(2\sigma - 1)$ and (2σ); in total this estimator will use $2P$ unique stimuli.

We note that an unbiased estimate of the numerator of CKA has been developed and evaluated (Song et al., 2007; Kornblith et al., 2019; Murphy et al., 2024)—but has not been extended to stochastic networks. The numerator of CKA is W_1 (eq. 16) thus this extension could be made using the strategy described in Section 3.4.

C.2 Deriving the Quadratic Program

The optimization problem in eq. (19) takes the form:

$$
\underset{\boldsymbol{\gamma}}{\text{minimize}} \quad \boldsymbol{\gamma}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{\gamma} + N^2\left(\max_x f^2(\boldsymbol{\gamma}, x)\right) \tag{62}
$$

where $f(\boldsymbol{\gamma}, x) = x^{1/2} - \sum_p \gamma_p x^p$,

$$
\boldsymbol{\gamma} = \begin{bmatrix}\gamma_1\\ \vdots\\ \gamma_P\end{bmatrix} \in \mathbb{R}^P, \quad \boldsymbol{A} = \begin{bmatrix}\mathbb{C}\mathrm{ov}(\hat{W}_1, \hat{W}_1) & \dots & \mathbb{C}\mathrm{ov}(\hat{W}_1, \hat{W}_P)\\ \vdots & & \vdots\\ \mathbb{C}\mathrm{ov}(\hat{W}_P, \hat{W}_1) & \dots & \mathbb{C}\mathrm{ov}(\hat{W}_P, \hat{W}_P)\end{bmatrix} \in \mathbb{R}^{P\times P}, \tag{63}
$$

Notice that f is linear in $\boldsymbol{\gamma}$, and that $\boldsymbol{A}$ is symmetric, positive-definite.

We will reformulate eq. (62) in several steps, and ultimately obtain a quadratic program that can be efficiently solved. First, we introduce a new optimization variable $u \in \mathbb{R}$ whose square is an upper bound on $f^2(\boldsymbol{\gamma}, x)$ for all $x \in [0, 1]$. Thus, the optimal $\boldsymbol{\gamma}$ for the problem:

$$
\begin{aligned}
&\underset{\boldsymbol{\gamma}, u}{\text{minimize}} && \boldsymbol{\gamma}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{\gamma} + N^2 u^2\\
&\text{subject to} && u^2 \geq f^2(\boldsymbol{\gamma}, x) \quad \text{for all } x \in [0, 1]
\end{aligned} \tag{64}
$$

coincides to the optimal $\boldsymbol{\gamma}$ solving eq. (62). This is essentially an *epigraph reformulation* of the original problem (see Boyd & Vandenberghe, 2004, equation 4.11). Notice that the objective function is quadratic in this reformulation.

Next, we lay down a fine grid of linearly spaced test points $x_1, \dots, x_T \in [0, 1]$. We can then obtain a good approximation to the solution in eq. (64) by solving:

$$
\begin{aligned}
&\underset{\boldsymbol{\gamma}, u}{\text{minimize}} && \boldsymbol{\gamma}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{\gamma} + N^2 u^2\\
&\text{subject to} && u^2 \geq f^2(\boldsymbol{\gamma}, x_t) \quad \text{for all } t \in 1, \dots, T
\end{aligned} \tag{65}
$$

Of course, increasing T (the number of test points) improves the approximation arbitrarily well.

Finally, the constraints of the problem can be put into a form that is jointly linear in $\boldsymbol{\gamma}$ and u. First, constraining $u^2 \geq f^2(\boldsymbol{\gamma}, x_t)$ is equivalent to simultaneously constraining $u \geq f(\boldsymbol{\gamma}, x_t)$ and $u \geq -f(\boldsymbol{\gamma}, x_t)$. Then, plugging in the definition of $f(\boldsymbol{\gamma}, x_t)$, and rearranging we have:

$$\begin{aligned} \underset{\boldsymbol{\gamma}, u}{\text{minimize}} \quad & \boldsymbol{\gamma}^\mathsf{T} \boldsymbol{A} \boldsymbol{\gamma} + N^2 u^2 \\ \text{subject to} \quad & u + \sum_p \gamma_p x_t^p \geq x_t^{1/2} \quad \text{for all } t \in 1, \dots, T \\ & u - \sum_p \gamma_p x_t^p \geq -x_t^{1/2} \quad \text{for all } t \in 1, \dots, T \end{aligned} \tag{66}$$

This objective is quadratic and the constraints are linear with respect to the optimized quantities. Thus, a solution (approximated to high accuracy) can be achieved efficiently using off-the-shelf quadratic programming solvers. To enforce the user defined bound on the bias a final two constraints are be appended to eq. (66): $-Nu \geq -c$ and $Nu \geq -c$, where c is the upper bound on the absolute bias.

C.3 Confidence intervals

To form approximate α level confidence intervals around $\widehat{\|\boldsymbol{\Sigma}_{ij}\|_*}$ we use the maximal bias (eq. 19, term 1) and variance (eq. 19, term 2) from the quadratic program's solution:

$$\left[\widehat{\|\boldsymbol{\Sigma}_{ij}\|_*} - z^* \sqrt{\boldsymbol{\gamma}^\mathsf{T} \boldsymbol{A} \boldsymbol{\gamma}} - Nu, \quad \widehat{\|\boldsymbol{\Sigma}_{ij}\|_*} + z^* \sqrt{\boldsymbol{\gamma}^\mathsf{T} \boldsymbol{A} \boldsymbol{\gamma}} + Nu\right],$$

where z^* is the critical value of the standard normal. For confidence intervals of the similarity score we scale this interval by the denominator of the similarity score.

D Appendix: Experiment Details

D.1 Simulated Experiments

To draw data for our simulations, we set the eigenvalues of the $\boldsymbol{\Sigma}_{ii}$ and the singular values of $\boldsymbol{\Sigma}_{ij}$ to a ground truth nuclear norm and similarity score. To demonstrate the estimators accuracy across the space of orthogonal transformations we apply a random orthogonal rotation matrix to each population's covariance in each new parameter setting.

D.2 Experimental data from Stringer et al. (2019)

Neural activity in mouse primary visual cortex was recorded using a two-photon microscope while mice were free to run on an air-floating ball. Recordings were collected across multiple depth planes at a frequency of 2.5 or 3 Hz, with planes 30-35 μm apart. The field of view of the microscope was selected such that 10,000 neurons could be observed within a retinotopic location on the stimulus display.

All stimuli were presented for 0.5s with a random inter-stimulus interval between 0.3 and 1.1s consisting of a grey-screen. The images used in the experiment were taken from the ImageNet database, which includes categories such as birds, cats, and insects. The researchers manually selected images that had a mix of low and high spatial frequencies and that did not consist of more than 50 % uniform background. All images were uniformly contrast-normalized by subtracting the local mean brightness and dividing by the local mean contrast. Each stimulus consisted of a different normalized image from the ImageNet database, with 2,800 different images used in total. The same image was displayed on all three screens, but each screen showed the image at a different rotation. Each of the 2,800 natural image stimuli were displayed twice in a recording in two blocks of the same randomized order.

Calcium movie data was processed using the Suite2p toolbox to estimate spike rates of neurons. Underlying neural activity was estimated using non-negative spike deconvolution (Frierich et. al., 2017). These deconvolved traces were normalized to the mean and standard deviation of their activity during a 30-minute period of grey-screen spontaneous activity. For further detail please see the original study Stringer et al. (2019). All analyses done in this paper were performed on the pre-processed data available on figshare (`https://figshare.com/articles/Recordings_of_ten_thousand_neurons_in_visual_cortex_in_response_to_2_800_natural_images/6845348`).

E Appendix: Applications to Deep Learning

Here we apply the plug-in and moment based estimator to neural network responses to demonstrate impacts of the differences between these estimators and relevance to neural networks. We make the point that the bias of the plug-in estimator, but not the moment estimator, is substantial for small samples. Furthermore, plug-in bias depends on the effective dimensionality of the two populations. Thus, naively using the plug-in can lead to erroneous scientific conclusions because the estimate bias can correlate with irrelevant nuisance variables. Concretely, we find the plug-in estimator bias tends to decrease with the effective dimensionality of neural populations. Thus if similarity between two populations appears to be explained by some manipulation of interest (e.g., training regime) it can be confounded through variation in effective dimensionality.

A common question in the study of neural network representations is how two networks with the same architecture trained on the same task but with different initializations and training procedures are similar/different from each other. To show how the estimators considered in this paper can be used as a tool to study this question, we considered two ResNet-50 (He et al., 2016) architectures trained to categorize ImageNet (Deng et al., 2009), specifically two sets of pretrained weights available in Pytorch (`ResNet50_Weights.IMAGENET1K_V1` and `ResNet50_Weights.IMAGENET1K_V2` as described here). We then compared a randomly chosen subset of the neurons (100 in each network) in the penultimate layer (before the final fully connected layer mapping to the logits) across the two networks using the plug-in estimator and the moment estimator. To compute a ground truth similarity metric we applied the plug-in estimator to the responses of these units across 432,064 images randomly chosen as a subset of the ImageNet dataset. To compare finite sample bias for each number of observed stimuli M, we randomly re-sampled across images and calculated the mean and SE of the two estimators as a function of number of images (Fig. 6). We found that the bias of the plug-in-estimator was at worst 3-fold and this bias decreased slowly, whereas the moment estimator showed a small amount of bias even with the smallest numbers of samples.

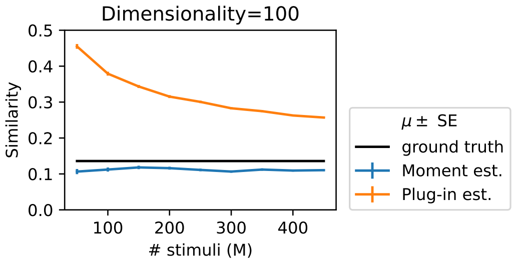

Figure 6: Comparison of plug-in and moment based estimators bias estimating similarity in the hidden layer of a deep neural network as a function of number of samples. Here we specifically study a subset of the neurons ($N = 100$) in the penultimate layer of two ResNet-50 architectures trained on ImageNet with different initializations and training procedures.

Finally, we considered how the bias of the plug-in estimator would vary with respect to irrelevant properties of the neural populations chosen. We reasoned that such a dependence could confound results on similarity between neural populations. It is known that the effective dimensionality, $(\sum_{i=1}^{N} \lambda_i)^2 / \sum_{i=1}^{N} \lambda_i^2$, of a response distribution determines the rate at which its sample covariance and thus singular values can be estimated. To determine if this in turn biased the plug-in estimator in a real application we randomly re-sampled with out replacement 100 units of the 2048 from the two neural networks 1000 times. We measured the ground truth similarity for each subset, the geometric mean of the effective dimensionality of the 100 units from the two networks (calculated across all images), and the plug-in average estimate across 50 random samplings of images. We found that the bias (difference of average plug-in estimate and ground truth), across re-sampling of units had a moderate negative correlation (r=-0.31) with the effective dimensionality of those populations. Thus observed differences in the similarity of neural network units may be confounded by their dimensionality and its effects on the plug-in estimator.

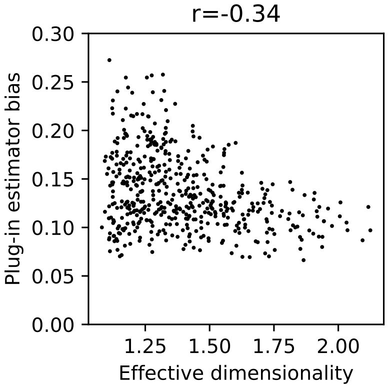

Figure 7: Each dot corresponds to a random selection of 100 neurons from the 2048 neurons in the penultimate layer of a ResNet-50. Across subsets of $N = 100$ neurons, we observe a negative correlation between the bias of the plug-in estimator with $M = 50$ stimuli and the effective dimensionality of the neural activations.