Generalized Shape Metrics on Neural Representations

Alex H. Williams
Statistics Department
Stanford University
ahwillia@stanford.edu

Erin Kunz
Electrical Engineering Department
Stanford University
ekunz@stanford.edu

Simon Kornblith
Google Research, Toronto
skornblith@google.com

Scott W. Linderman
Statistics Department
Stanford University
scott.linderman@stanford.edu

Abstract

Understanding the operation of biological and artificial networks remains a difficult and important challenge. To identify general principles, researchers are increasingly interested in surveying large collections of networks that are trained on, or biologically adapted to, similar tasks. A standardized set of analysis tools is now needed to identify how network-level covariates—such as architecture, anatomical brain region, and model organism—impact neural representations (hidden layer activations). Here, we provide a rigorous foundation for these analyses by defining a broad family of metric spaces that quantify representational dissimilarity. Using this framework we modify existing representational similarity measures based on canonical correlation analysis to satisfy the triangle inequality, formulate a novel metric that respects the inductive biases in convolutional layers, and identify approximate Euclidean embeddings that enable network representations to be incorporated into essentially any off-the-shelf machine learning method. We demonstrate these methods on large-scale datasets from biology (Allen Institute Brain Observatory) and deep learning (NAS-Bench-101). In doing so, we identify relationships between neural representations that are interpretable in terms of anatomical features and model performance.

1 Introduction

The extent to which different deep networks or neurobiological systems use equivalent representations in support of similar task demands is a topic of persistent interest in machine learning and neuroscience [1–3]. Several methods including linear regression [4, 5], canonical correlation analysis (CCA; [6, 7]), representational similarity analysis (RSA; [8]), and centered kernel alignment (CKA; [9]) have been used to quantify the similarity of hidden layer activation patterns. These measures are often interpreted on an ordinal scale and are employed to compare a limited number of networks—e.g., they can indicate whether networks A and B are more or less similar than networks A and C. While these comparisons have yielded many insights [4–12], the underlying methodologies have not been extended to systematic analyses spanning thousands of networks.

To unify existing approaches and enable more sophisticated analyses, we draw on ideas from *statistical shape analysis* [13–15] to develop dissimilarity measures that are proper metrics—i.e., measures that are symmetric and respect the triangle inequality. This enables several off-the-shelf methods with theoretical guarantees for classification (e.g. k-nearest neighbors, [16]) and clustering (e.g. hierarchical clustering [17]). Existing similarity measures can violate the triangle inequality, which complicates these downstream analyses [18–20]. However, we show that existing dissimilarity

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

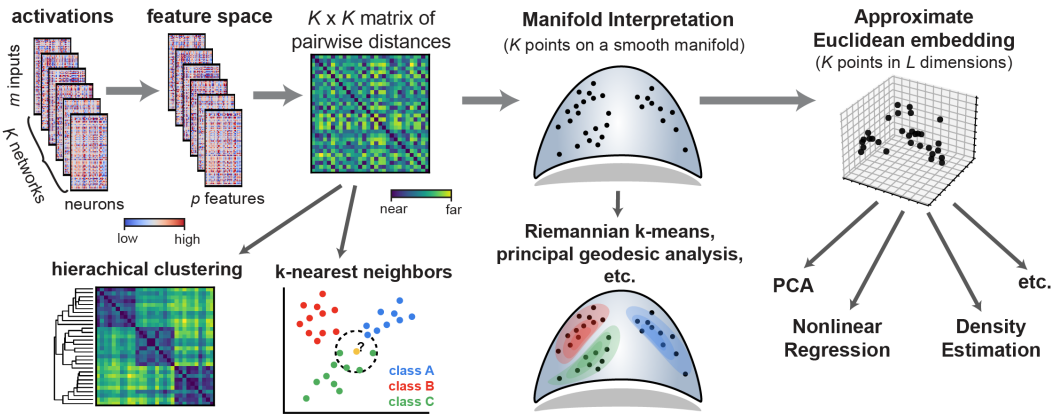

Figure 1: Machine learning workflows enabled by generalized shape metrics.

measures can often be modified to satisfy the triangle inequality and viewed as special cases of the framework we outline. We also describe novel metrics within this broader family that are specialized to convolutional layers and have appealing properties for analyzing artificial networks.

Moreover, we show empirically that these metric spaces on neural representations can be embedded with low distortion into Euclidean spaces, enabling an even broader variety of previously unconsidered supervised and unsupervised analyses. For example, we can use neural representations as the inputs to linear or nonlinear regression models. We demonstrate this approach on neural representations in mouse visual cortex (Allen Brain Observatory; [21]) in order to predict each brain region's anatomical hierarchy from its pattern of visual responses—i.e., predicting a feature of brain structure from function. We demonstrate a similar approach to analyze hidden layer representations in a database of 432K deep artificial networks (NAS-Bench-101; [22]) and find a surprising degree of correlation between early and deep layer representations.

Overall, we provide a theoretical grounding which explains why existing representational similarity measures are useful: they are often close to metric spaces, and can be modified to fulfill metric space axioms precisely. Further, we draw new conceptual connections between analyses of neural representations and established research areas [15, 23], utilize these insights to propose novel metrics, and demonstrate a general-purpose machine learning workflow that scales to datasets with thousands of networks.

2 Methods

This section outlines several workflows (Fig. 1) to analyze representations across large collections of networks. After briefly summarizing prior approaches (sec. 2.1), we cover background material on metric spaces and discuss their theoretical advantages over existing dissimilarity measures (sec. 2.2). We then present a class of metrics that capture these advantages (sec. 2.3) and cover a special case that is suited to convolutional layers (sec. 2.4). We then demonstrate the practical advantages of these methods in Section 3, and demonstrate empirically that Euclidean feature spaces can approximate the metric structure of neural representations, enabling a broad set of novel analyses.

2.1 Prior work and problem setup

Neural network representations are often summarized over a set of m reference inputs (e.g. test set images). Let $\boldsymbol{X}_i \in \mathbb{R}^{m \times n_i}$ and $\boldsymbol{X}_j \in \mathbb{R}^{m \times n_j}$ denote the responses of two networks (with n_i and n_j neurons, respectively) to a collection of these inputs. Quantifying the similarity between $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ is complicated by the fact that, while the m inputs are the same, there is no direct correspondence between the neurons. Even if $n_i = n_j$, the typical Frobenius inner product, $\langle \boldsymbol{X}_i, \boldsymbol{X}_j \rangle = \mathrm{Tr}[\boldsymbol{X}_i^\top \boldsymbol{X}_j]$, and metric, $\|\boldsymbol{X}_i - \boldsymbol{X}_j\| = \langle \boldsymbol{X}_i - \boldsymbol{X}_j, \boldsymbol{X}_i - \boldsymbol{X}_j \rangle^{1/2}$, fail to capture the desired notion of dissimilarity. For instance, let $\boldsymbol{\Pi}$ denote some $n \times n$ permutation matrix and let $\boldsymbol{X}_i = \boldsymbol{X}_j \boldsymbol{\Pi}$. Intuitively, we should consider $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ to be identical in this case since the ordering of neurons is arbitrary. Yet, clearly $\|\boldsymbol{X}_i - \boldsymbol{X}_j\| \neq 0$, except in very special cases.

One way to address this problem is to linearly regress over the neurons to predict $\boldsymbol{X}_i$ from $\boldsymbol{X}_j$. Then, one can use the coefficient of determination (R^2) as a measure of similarity [4, 5]. However, this similarity score is asymmetric—if one instead treats $\boldsymbol{X}_j$ as the dependent variable that is predicted from $\boldsymbol{X}_i$, this will result in a different R^2. Canonical correlation analysis (CCA; [6, 7]) and linear centered kernel alignment (linear CKA; [9, 24]) also search for linear correspondences between neurons, but have the advantage of producing symmetric scores. Representational similarity analysis (RSA; [8]) is yet another approach, which first computes an $m \times m$ matrix holding the dissimilarities between all pairs of representations for each network. These *representational dissimilarity matrices* (RDMs), are very similar to the $m \times m$ kernel matrices computed and compared by CKA. RSA traditionally quantifies the similarity between two neural networks by computing Spearman's rank correlation between their RDMs. A very recent paper by Shahbazi et al. [25], which was published while this manuscript was undergoing review, proposes to use the Riemannian metric between positive definite matrices instead of Spearman correlation. Similar to our results, this establishes a metric space that can be used to compare neural representations. Here, we leverage metric structure over *shape spaces* [13–15] instead of positive definite matrices, leading to complementary insights.

In summary, there are a diversity of methods that one can use to compare neural representations. Without a unifying theoretical framework it is unclear how to choose among them, use their outputs for downstream tasks, or generalize them to new domains.

2.2 Feature space mapping, metrics, and equivalence relations

Our first contribution will be to establish formal notions of distance (metrics) between neural representations. To accommodate the common scenario when the number of neurons varies across networks (i.e. when $n_i \neq n_j$), we first map the representations into a common feature space. For each set of representations, $\boldsymbol{X}_i$, we suppose there is a mapping into a p-dimensional feature space, $\boldsymbol{X}_i \mapsto \boldsymbol{X}_i^\phi$, where $\boldsymbol{X}_i^\phi \in \mathbb{R}^{m \times p}$. In the special case where all networks have equal size, $n_1 = n_2 = \ldots = n$, we can express the feature mapping as a single function $\phi : \mathbb{R}^{m \times n} \mapsto \mathbb{R}^{m \times p}$, so that $\boldsymbol{X}_i^\phi = \phi(\boldsymbol{X}_i)$. When networks have dissimilar sizes, we can map the representations into a common dimension using, for example, PCA [6].

Next, we seek to establish *metrics* within the feature space, which are distance functions that satisfy:

$$\text{Equivalence:} \quad d(\boldsymbol{X}_i^\phi, \boldsymbol{X}_j^\phi) = 0 \iff \boldsymbol{X}_i^\phi \sim \boldsymbol{X}_j^\phi \tag{1}$$

$$\text{Symmetry:} \quad d(\boldsymbol{X}_i^\phi, \boldsymbol{X}_j^\phi) = d(\boldsymbol{X}_j^\phi, \boldsymbol{X}_i^\phi) \tag{2}$$

$$\text{Triangle Inequality:} \quad d(\boldsymbol{X}_i^\phi, \boldsymbol{X}_j^\phi) \leq d(\boldsymbol{X}_i^\phi, \boldsymbol{X}_k^\phi) + d(\boldsymbol{X}_k^\phi, \boldsymbol{X}_j^\phi) \tag{3}$$

for all $\boldsymbol{X}_i^\phi$, $\boldsymbol{X}_j^\phi$, and $\boldsymbol{X}_k^\phi$ in the feature space. The symbol '$\sim$' denotes an *equivalence relation* between two elements. That is, the expression $\boldsymbol{X}_i^\phi \sim \boldsymbol{X}_j^\phi$ means that "$\boldsymbol{X}_i^\phi$ is equivalent to $\boldsymbol{X}_j^\phi$." Formally, distance functions satisfying Eqs. (1) to (3) define a metric over a quotient space defined by the equivalence relation and a pseudometric over $\mathbb{R}^{m \times p}$ (see Supplement A). Intuitively, by specifying different equivalence relations we can account for symmetries in network representations, such as permutations over arbitrarily labeled neurons (other options are discussed below in sec. 2.3).

Metrics quantify dissimilarity in a way that agrees with our intuitive notion of distance. For example, Eq. (2) ensures that the distance from $\boldsymbol{X}_i^\phi$ to $\boldsymbol{X}_j^\phi$ is the same as the distance from $\boldsymbol{X}_j^\phi$ to $\boldsymbol{X}_i^\phi$. Linear regression is an approach that violates this condition: the similarity measured by R^2 depends on which network is treated as the dependent variable.

Further, Eq. (3) ensures that distances are self-consistent in the sense that if two elements ($\boldsymbol{X}_i^\phi$ and $\boldsymbol{X}_j^\phi$) are both close to a third ($\boldsymbol{X}_k^\phi$), then they are necessarily close to each other. Many machine learning models and algorithms rely on this triangle inequality condition. For example, in clustering, it ensures that if $\boldsymbol{X}_i^\phi$ and $\boldsymbol{X}_j^\phi$ are put into the same cluster as $\boldsymbol{X}_k^\phi$, then $\boldsymbol{X}_i^\phi$ and $\boldsymbol{X}_j^\phi$ cannot be too far apart, thus implying that they too can be clustered together. Intuitively, this establishes an appealing transitive relation for clustering, which can be violated when the triangle inequality fails to hold. Existing measures based on CCA, RSA, and CKA, are symmetric, but do not satisfy the triangle inequality. By modifying these approaches to satisfy the triangle inequality, we avoid potential pitfalls and can leverage theoretical guarantees on learning in proper metric spaces [16–20].

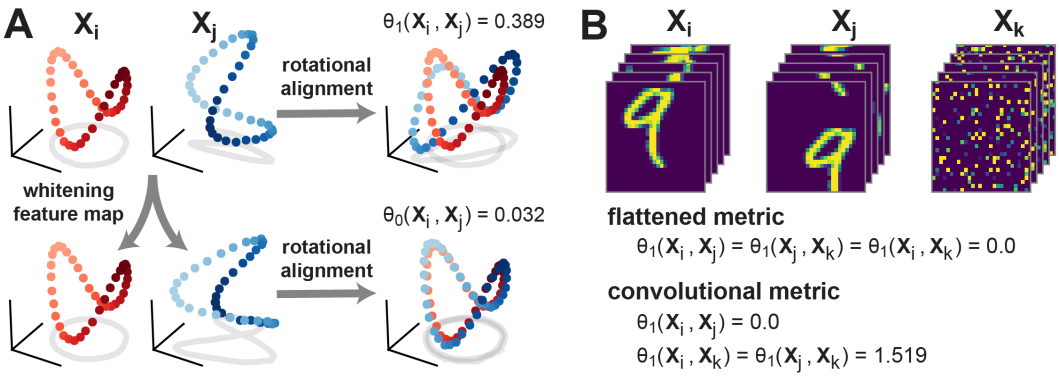

Figure 2: *(A)* Schematic illustration of metrics with rotational invariance (top), and linear invariance (bottom). Red and blue dots represent a pair of network representations $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, which correspond to m points in n-dimensional space. *(B)* Demonstration of convolutional metric on toy data. Flattened metrics (e.g. [6, 9]) that ignore convolutional layer structure treat permuted images ($\boldsymbol{X}_k$, right) as equivalent to images with coherent spatial structure ($\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, left and middle). A convolutional metric, Eq. (11), distinguishes between these cases while still treating $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ as equivalent (obeying translation invariance).

2.3 Generalized shape metrics and group invariance

In this section, we outline a new framework to quantify representational dissimilarity, which leverages a well-developed mathematical literature on *shape spaces* [13–15]. The key idea is to treat $\boldsymbol{X}_i^\phi \sim \boldsymbol{X}_j^\phi$ if and only if there exists a linear transformation $\boldsymbol{T}$ within a set of allowable transformations $\mathcal{G}$, such that $\boldsymbol{X}_i^\phi = \boldsymbol{X}_j^\phi \boldsymbol{T}$. Although $\mathcal{G}$ only contains linear functions, nonlinear alignments between the raw representations can be achieved when the feature mappings $\boldsymbol{X}_i \mapsto \boldsymbol{X}_i^\phi$ are chosen to be nonlinear. Much of shape analysis literature focuses on the special case where $p = n$ and $\mathcal{G}$ is the special orthogonal group $\mathcal{SO}(n) = \{\boldsymbol{R} \in \mathbb{R}^{n\times n} \mid \boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{I}, \det(\boldsymbol{R}) = 1\}$, meaning that $\boldsymbol{X}_i^\phi$ and $\boldsymbol{X}_j^\phi$ are equivalent if there is a n-dimensional rotation (without reflection) that relates them. Standard shape analysis further considers each $\boldsymbol{X}_i^\phi$ to be a mean-centered ($(\boldsymbol{X}_i^\phi)^\top \mathbf{1} = \mathbf{0}$) and normalized ($\|\boldsymbol{X}_i^\phi\| = 1$) version of the raw landmark locations held in $\boldsymbol{X}_i \in \mathbb{R}^{m\times n}$ (an assumption that we will relax). That is, the feature map $\phi : \mathbb{R}^{m\times n} \mapsto \mathbb{S}^{m\times n}$ transforms the raw landmarks onto the hypersphere, denoted $\mathbb{S}^{m\times n}$, of $m \times n$ matrices with unit Frobenius norm. In this context, $\boldsymbol{X}_i^\phi \in \mathbb{S}^{m\times n}$ is called a "pre-shape." By removing rotations from a pre-shape, $[\boldsymbol{X}_i^\phi] = \{\boldsymbol{S} \in \mathbb{S}^{m\times n} \mid \boldsymbol{S} \sim \boldsymbol{X}_i^\phi\}$ for pre-shape $\boldsymbol{X}_i^\phi$, we recover its "shape."

To quantify dissimilarity in neural representations, we generalize this notion of shape to include other feature mappings and alignments. The minimal distance within the feature space, after optimizing over alignments, defines a metric under suitable conditions (Fig. 2A). This results in a broad variety of *generalized shape metrics* (see also, ch. 18 of [15]), which fall into two categories as formalized by the pair of propositions below. Proofs are provided in Supplement B.

Proposition 1. *Let $\boldsymbol{X}_i^\phi \in \mathbb{R}^{m\times p}$, and let $\mathcal{G}$ be a group of linear isometries on $\mathbb{R}^{m\times p}$. Then,*

$$d(\boldsymbol{X}_i^\phi, \boldsymbol{X}_j^\phi) = \min_{\boldsymbol{T}\in\mathcal{G}} \|\boldsymbol{X}_i^\phi - \boldsymbol{X}_j^\phi \boldsymbol{T}\| \tag{4}$$

defines a metric, where $\boldsymbol{X}_i^\phi \sim \boldsymbol{X}_j^\phi$ if and only if there is a $\boldsymbol{T} \in \mathcal{G}$ such that $\boldsymbol{X}_i^\phi = \boldsymbol{X}_j^\phi \boldsymbol{T}$.

Proposition 2. *Let $\boldsymbol{X}_i^\phi \in \mathbb{S}^{m\times p}$, and let $\mathcal{G}$ be a group of linear isometries on $\mathbb{S}^{m\times p}$. Then,*

$$\theta(\boldsymbol{X}_i^\phi, \boldsymbol{X}_j^\phi) = \min_{\boldsymbol{T}\in\mathcal{G}} \arccos\left\langle \boldsymbol{X}_i^\phi, \boldsymbol{X}_j^\phi \boldsymbol{T} \right\rangle \tag{5}$$

defines a metric, where $\boldsymbol{X}_i^\phi \sim \boldsymbol{X}_j^\phi$ if and only if there is a $\boldsymbol{T} \in \mathcal{G}$ such that $\boldsymbol{X}_i^\phi = \boldsymbol{X}_j^\phi \boldsymbol{T}$.

Two key conditions appear in these propositions. First, $\mathcal{G}$ must be a *group* of functions. This means $\mathcal{G}$ is a set that contains the identity function, is closed under composition ($\boldsymbol{T}_1\boldsymbol{T}_2 \in \mathcal{G}$ for any $\boldsymbol{T}_1 \in \mathcal{G}$ and $\boldsymbol{T}_2 \in \mathcal{G}$), and whose elements are invertible by other members of the set (if $\boldsymbol{T} \in \mathcal{G}$ then $\boldsymbol{T}^{-1} \in \mathcal{G}$).

Second, every $\boldsymbol{T} \in \mathcal{G}$ must be an *isometry*, meaning that $\|\boldsymbol{X}_i^\phi - \boldsymbol{X}_j^\phi\| = \|\boldsymbol{X}_i^\phi \boldsymbol{T} - \boldsymbol{X}_j^\phi \boldsymbol{T}\|$ for all $\boldsymbol{T} \in \mathcal{G}$ and all elements of the feature space. On $\mathbb{R}^{m\times p}$ and $\mathbb{S}^{m\times p}$, all linear isometries are orthogonal transformations. Further, the set of orthogonal transformations, $\mathcal{O}(p) = \{\boldsymbol{Q} \in \mathbb{R}^{p\times p} : \boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{I}\}$, defines a well-known group. Thus, the condition that $\mathcal{G}$ is a group of isometries is equivalent to $\mathcal{G}$ being a subgroup of $\mathcal{O}(p)$—i.e., a subset of $\mathcal{O}(p)$ satisfying the group axioms.

Intuitively, by requiring $\mathcal{G}$ to be a group of functions, we ensure that the alignment procedure is symmetric—i.e. it is equivalent to transform $\boldsymbol{X}_i^\phi$ to match $\boldsymbol{X}_j^\phi$, or transform the latter to match the former. Further, by requiring each $\boldsymbol{T} \in \mathcal{G}$ to be an isometry, we ensure that the underlying metric (Euclidean distance for Proposition 1; angular distance for Proposition 2) preserves its key properties.

Together, these propositions define a broad class of metrics as we enumerate below. For simplicity, we assume that $n_i = n_j = n$ in the examples below, with the understanding that a PCA or zero-padding preprocessing step has been performed in the case of dissimilar network sizes. This enables us to express the metrics as functions of the raw activations, i.e. functions $\mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \mapsto \mathbb{R}_+$.

Permutation invariance The most stringent notion of representational similarity is to demand that neurons are one-to-one matched across networks. If we set the feature map to be the identity function, i.e., $\boldsymbol{X}_i^\phi = \boldsymbol{X}_i$ for all i, then:

$$d_{\mathcal{P}}(\boldsymbol{X}_i, \boldsymbol{X}_j) = \min_{\boldsymbol{\Pi}\in\mathcal{P}(n)} \|\boldsymbol{X}_i - \boldsymbol{X}_j\boldsymbol{\Pi}\| \tag{6}$$

defines a metric by Proposition 1 since the set of permutation matrices, $\mathcal{P}(n)$, is a subgroup of $\mathcal{O}(n)$. To evaluate this metric we must optimize over the set of neuron permutations to align the two networks. This can be reformulated (see Supplement C) as a fundamental problem in combinatorial optimization known as the linear assignment problem [26]. Exploiting an algorithm due to Jonker and Volgenant [27, 28] we can solve this problem in $O(n^3)$ time. The overall runtime for evaluating Eq. (6) is $O(mn^2 + n^3)$, since we must evaluate $\boldsymbol{X}_i^\top \boldsymbol{X}_j$ to formulate the assignment problem.

Rotation invariance Let $\boldsymbol{C} = \boldsymbol{I}_m - (1/m)\mathbf{1}\mathbf{1}^\top$ denote an $m \times m$ *centering matrix*, and consider the feature mapping ϕ_1 which mean-centers the columns, $\phi_1(\boldsymbol{X}_i) = \boldsymbol{X}_i^{\phi_1} = \boldsymbol{C}\boldsymbol{X}_i$. Then,

$$d_1(\boldsymbol{X}_i, \boldsymbol{X}_j) = \min_{\boldsymbol{Q}\in\mathcal{O}} \|\boldsymbol{X}_i^{\phi_1} - \boldsymbol{X}_j^{\phi_1}\boldsymbol{Q}\| \tag{7}$$

defines a metric by Proposition 1, and is equivalent to the *Procrustes size-and-shape distance* with reflections [15]. Further, by Proposition 2,

$$\theta_1(\boldsymbol{X}_i, \boldsymbol{X}_j) = \min_{\boldsymbol{Q}\in\mathcal{O}} \arccos \frac{\langle \boldsymbol{X}_i^{\phi_1}, \boldsymbol{X}_j^{\phi_1}\boldsymbol{Q}\rangle}{\|\boldsymbol{X}_i^{\phi_1}\|\|\boldsymbol{X}_j^{\phi_1}\|} \tag{8}$$

defines another metric, and is closely related to the Riemannian distance on Kendall's shape space [15]. To evaluate Eqs. (7) and (8), we must optimize over the set of orthogonal matrices to find the best alignment. This also maps onto a fundamental optimization problem known as the *orthogonal Procrustes problem* [29, 30], which can be solved in closed form in $O(n^3)$ time. As in the permutation-invariant metric described above, the overall runtime is $O(mn^2 + n^3)$.

Linear invariance Consider a partial whitening transformation, parameterized by $0 \le \alpha \le 1$:

$$\boldsymbol{X}^{\phi_\alpha} = \boldsymbol{C}\boldsymbol{X}(\alpha\boldsymbol{I}_n + (1-\alpha)(\boldsymbol{X}^\top\boldsymbol{C}\boldsymbol{X})^{-1/2}) \tag{9}$$

Note that $\boldsymbol{X}^\top\boldsymbol{C}\boldsymbol{X}$ is the empirical covariance matrix of $\boldsymbol{X}$. Thus, when $\alpha = 0$, Eq. (9) corresponds to ZCA whitening [31], which intuitively removes invertible linear transformations from the representations. Thus, when $\alpha = 0$ the metric outlined below treats $\boldsymbol{X}_i \sim \boldsymbol{X}_j$ if there exists an affine transformation that relates them: $\boldsymbol{X}_i = \boldsymbol{X}_j\boldsymbol{W} + \boldsymbol{b}$ for some $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{b} \in \mathbb{R}^n$. When $\alpha = 1$, Eq. (9) reduces to the mean-centering feature map used above.

Using orthogonal alignments within this feature space leads to a metric that is related to CCA. First, let $\rho_1 \ge \ldots \ge \rho_n \ge 0$ denote the singular values of $(\boldsymbol{X}_i^{\phi_\alpha})^\top(\boldsymbol{X}_j^{\phi_\alpha})/\|\boldsymbol{X}_i^{\phi_\alpha}\|\|\boldsymbol{X}_j^{\phi_\alpha}\|$. One can show that

$$\theta_\alpha(\boldsymbol{X}_i, \boldsymbol{X}_j) = \min_{\boldsymbol{Q}\in\mathcal{O}} \arccos \frac{\langle \boldsymbol{X}_i^{\phi_\alpha}, \boldsymbol{X}_j^{\phi_\alpha}\boldsymbol{Q}\rangle}{\|\boldsymbol{X}_i^{\phi_\alpha}\|\|\boldsymbol{X}_j^{\phi_\alpha}\|} = \arccos(\textstyle\sum_\ell \rho_\ell)\,, \tag{10}$$

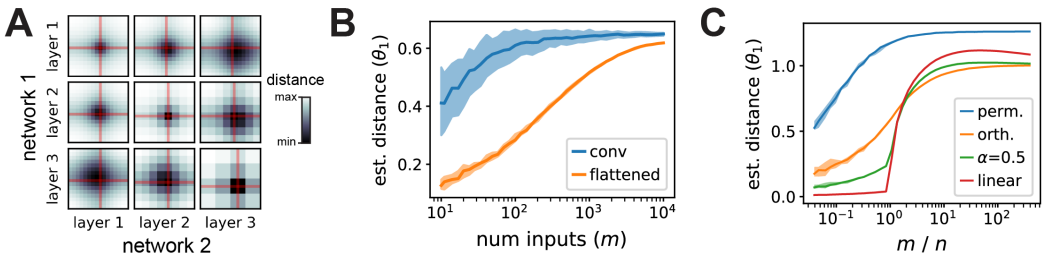

Figure 3: *(A)* Each heatmap shows a brute-force search over the shift parameters along the width and height dimensions of a pair of convolutional layers compared across two networks. The optimal shifts are typically close to zero (red lines). *(B)* Impact of sample size, m, on flattened and convolutional metrics with orthogonal invariance. The convolutional metric approaches its final value faster than the flattened metric, which is still increasing even at the full size of the CIFAR-10 test set ($m = 10^4$). *(C)* Impact of sample density, m/n, on metrics invariant to permutation, orthogonal, regularized linear ($\alpha = 0.5$), and linear transformations. Shaded regions mark the 10$^{\text{th}}$ and 90$^{\text{th}}$ percentiles across shuffled repeats. Further details are provided in Supplement E.

and we can see from Proposition 2 that this defines a metric for any $0 \leq \alpha \leq 1$. When $\alpha = 0$, the values $\rho_1, \ldots, \rho_n$ are proportional to the canonical correlation coefficients, with $1/n$ being the factor of proportionality. When $\alpha > 0$, these values can be viewed as ridge regularized canonical correlation coefficients [32]. See Supplement C for further details. Past works [6, 7] have used the average canonical correlation as a measure of representational similarity. When $\alpha = 0$, the average canonical correlation is given by $\sum_\ell \rho_\ell = \cos\theta_0(\boldsymbol{X}_i, \boldsymbol{X}_j)$. Thus, if we apply $\arccos(\cdot)$ to the average canonical correlation, we modify the calculation to produce a proper metric (see Fig. 4A). Since the covariance is often ill-conditioned or singular in practice, setting $\alpha > 0$ to regularize the calculation is also typically necessary.

Nonlinear invariances We discuss feature maps that enable nonlinear notions of equivalence, and which relate to kernel CCA [33] and CKA [9], in Supplement C.

2.4 Metrics for convolutional layers

In deep networks for image processing, each convolutional layer produces a $h \times w \times c$ array of activations, whose axes respectively correspond to image height, image width, and channels (number of convolutional filters). If stride-1 circular convolutions are used, then applying a circular shift along either spatial dimension produces the same shift in the layer's output. It is natural to reflect this property, known as translation equivariance [23], in the equivalence relation on layer representations. Supposing that the feature map preserves the shape of the activation tensor, we have $\boldsymbol{X}_k^\phi \in \mathbb{R}^{m \times h \times w \times c}$ for neural networks indexed by $k \in 1, \ldots, K$. Letting $\mathcal{S}(n)$ denote the group of n-dimensional circular shifts (a subgroup of the permutation group) and '$\otimes$' denote the Kronecker product, we propose:

$$\boldsymbol{X}_i^\phi \sim \boldsymbol{X}_j^\phi \iff \text{vec}(\boldsymbol{X}_i^\phi) = (\boldsymbol{I} \otimes \boldsymbol{S}_1 \otimes \boldsymbol{S}_2 \otimes \boldsymbol{Q})\text{vec}(\boldsymbol{X}_j^\phi) \tag{11}$$

for some $\boldsymbol{S}_1 \in \mathcal{S}(h)$, $\boldsymbol{S}_2 \in \mathcal{S}(w)$, $\boldsymbol{Q} \in \mathcal{O}(c)$, as the desired equivalence relation. This relation allows for orthogonal invariance across the channel dimension but only shift invariance across the spatial dimensions. The mixed product property of Kronecker products, $(\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{C} \otimes \boldsymbol{D}) = \boldsymbol{AB} \otimes \boldsymbol{CD}$, ensures that the overall transformation maintains the group structure and remains an isometry. Figure 2B uses a toy dataset (stacked MNIST digits) to show that this metric is sensitive to differences in spatial activation patterns, but insensitive to coherent spatial translations across channels. In contrast, metrics that ignore the convolutional structure (as in past work [6, 9]) treat very different spatial patterns as identical representations.

Evaluating Eq. (11) requires optimizing over spatial shifts in conjuction with solving a Procrustes alignment. If we fit the shifts by an exhaustive brute-force search, the overall runtime is $O(mh^2w^2c^2 + hwc^3)$, which is costly if this calculation is repeated across a large collection of networks. In practice, we observe that the optimal shift parameters are typically close to zero (Fig. 3A). This motivates the more stringent equivalence relation:

$$\boldsymbol{X}_i^\phi \sim \boldsymbol{X}_j^\phi \iff \text{vec}(\boldsymbol{X}_i^\phi) = (\boldsymbol{I} \otimes \boldsymbol{I} \otimes \boldsymbol{I} \otimes \boldsymbol{Q})\text{vec}(\boldsymbol{X}_j^\phi) \quad \text{for some } \boldsymbol{Q} \in \mathcal{Q}, \tag{12}$$

which has a more manageable runtime of $O(mhwc^2 + c^3)$. To evaluate the metrics implied by Eq. (12), we can simply reshape each $\boldsymbol{X}_k^\phi$ from a $(m \times h \times w \times c)$ tensor into a $(mhw \times c)$ matrix and apply the Procrustes alignment procedure as done above for previous metrics. In contrast, the "flattened metric" in Fig. 2B reshapes the features into a $(m \times hwc)$ matrix, resulting in a more computationally expensive alignment that runs in $O(mh^2w^2c^2 + h^3w^3c^3)$ time.

2.5 How large of a sample size is needed?

An important issue, particularly in neurobiological applications, is to determine the number of network inputs, m, and neurons, n, that one needs to accurately infer the distance between two network representations [12]. Reasoning about these questions rigorously requires a probabilistic perspective of neural representational similarity, which is missing from current literature and which we outline in Supplement D for generalized shape metrics. Intuitively, looser equivalence relations are achieved by having more flexible alignment operations (e.g. nonlinear instead of linear alignments). Thus, looser equivalence relations require more sampled inputs to prevent overfitting. Figure 3B-C show that this intuition holds in practice for data from deep convolutional networks. Metrics with looser equivalence relations—the "flattened" metric in panel B, or e.g. the linear metric in panel C—converge slower to a stable estimate as m is increased.

2.6 Modeling approaches and conceptual insights

Generalized shape metrics facilitate several new modeling approaches and conceptual perspectives. For example, a collection of representations from K neural networks can, in certain cases, be interpreted and visualized as K points on a smooth manifold (see Fig. 1). This holds rigorously due to the *quotient manifold theorem* [34] so long as $\mathcal{G}$ is not a finite set (e.g. corresponding to permutation) and all matrices are full rank in the feature space. This geometric intuition can be made even stronger when $\mathcal{G}$ corresponds to a connected manifold, such as $\mathcal{SO}(p)$. In this case, it can be shown that the geodesic distance between two neural representations coincides with the metrics we defined in Propositions 1 and 2 (see Supplement C, and [15]). This result extends the well-documented manifold structure of *Kendall's shape space* [35].

Viewing neural representations as points on a manifold is not a purely theoretical exercise—several models can be adapted to manifold-valued data (e.g. principal geodesic analysis [36] provides a generalization of PCA), and additional adaptions are an area of active research [37]. However, there is generally no simple connection between these curved geometries and the flat geometries of Euclidean or Hilbert spaces [38].[1] Unfortunately, the majority of off-the-shelf machine learning tools are incompatible with the former and require the latter. Thus, we can resort to a heuristic approach: the set of K representations can be embedded into a Euclidean space that approximately preserves the pairwise shape distances. One possibility, employed widely in shape analysis, is to embed points in the tangent space of the manifold at a reference point [41, 42]. Another approach, which we demonstrate below with favorable results, is to optimize the vector embedding directly via multi-dimensional scaling [43, 44].

3 Applications and Results

We analyzed two large-scale public datasets spanning neuroscience (Allen Brain Observatory, ABO; Neuropixels - visual coding experiment; [21]) and deep learning (NAS-Bench-101; [22]). We constructed the ABO dataset by pooling recorded neurons from $K = 48$ anatomically defined brain regions across all sessions; each $\boldsymbol{X}_k \in \mathbb{R}^{m \times n}$ was a dimensionally reduced matrix holding the neural responses (summarized by $n = 100$ principal components) to $m = 1600$ movie frames (120 second clip, "natural movie three"). The full NAS-Bench-101 dataset contains $423{,}624$ architectures; however, we analyze a subset of $K = 2000$ networks for simplicity. In this application each $\boldsymbol{X}_k \in \mathbb{R}^{m \times n}$ is a representation from a specific network layer, with $(m, n) \in \{(32^2 \times 10^5, 128), (16^2 \times 10^5, 256), (8^2 \times 10^5, 512), (10^5, 512)\}$. Here, n corresponds to the number of channels and m is the product of the number of test set images (10^5) and the height and width dimensions of the convolutional layer—i.e., we use equivalence relation in Eq. (12) to evaluate dissimilarity.

[1] However, see [39] for a conjectured relationship and [40] for a result in the special case of 2D shapes.

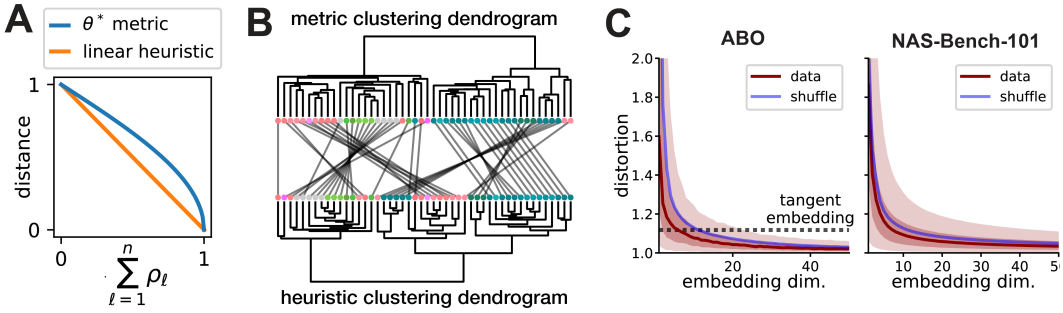

Figure 4: (A) Comparison of metric and linear heuristic. (B) Metric and linear heuristic produce discordant hierarchical clusterings of brain areas in the ABO dataset. Leaves represent brain areas that are clustered by representational similarity (see Fig. 1C), colored by Allen reference atlas, and ordered to maximize dendrogram similarities of adjacent leaves. In the middle, grey lines connect leaves corresponding to the same brain region across the two dendrograms. (C) ABO and NAS-Bench-101 datasets can be accurately embedded into Euclidean spaces. Dark red line shows median distortion. Light red shaded region corresponds to 5th to 95th percentiles of distortion, dark red shaded corresponds to interquartile range. The mean distortion of a null distribution over representations (blue line) was generated by shuffling the m inputs independently in each network.

Triangle inequality violations can occur in practice when using existing methods. As mentioned above, a dissimilarity measure based on the mean canonical correlation, $1 - \sum_\ell \rho_\ell$, has been used in past work [7, 10]. We refer to this as the "linear heuristic." A slight reformulation of this calculation, $\arccos\left(\sum_\ell \rho_\ell\right)$, produces a metric that satisfies the triangle inequality (see Eq. (10)). Figure 4A compares these calculations as a function of the average (regularized) canonical correlation: one can see that $\arccos(\cdot)$ is approximately linear when the mean correlation is near zero, but highly nonlinear when the mean correlation is near one. Thus, we reasoned that triangle inequality violations are more likely to occur when K is large and when many network representations are close to each other. Both ABO and NAS-Bench-101 datasets satisfy these conditions, and in both cases we observed triangle inequality violations by the linear heuristic with full regularization ($\alpha = 1$): 17/1128 network pairs in the ABO dataset had at least one triangle inequality violation, while 10128/100000 randomly sampled network pairs contained violations in the NAS-Bench-101 Stem layer dataset. We also examined a standard version of RSA that quantifies similarity via Spearman's rank correlation coefficient [8]. Similar to the results above, we observed violations in 14/1128 pairs of networks in the ABO dataset.

Overall, these results suggest that generalized shape metrics correct for triangle inequality violations that do occur in practice. Depending on the dataset, these violations may be rare (~1% occurrence in ABO) or relatively common (~10% in the Stem layer of NAS-Bench-101). These differences can produce quantitative discrepancies in downstream analyses. For example, the dendrograms produced by hierarchical clustering differ depending on whether one uses the linear heuristic or the shape distance (~85.1% dendrogram similarity as quantified by the method in [45]; see Fig. 4B).

Neural representation metric spaces can be approximated by Euclidean spaces. Having established that neural representations can be viewed as elements in a metric space, it is natural to ask if this metric space is, loosely speaking, "close to" a Euclidean space. We used standard multidimensional scaling methods (SMACOF, [43]; implementation in [46]) to obtain a set of embedded vectors, $\boldsymbol{y}_i \in \mathbb{R}^L$, for which $\theta_1(\boldsymbol{X}_i^\phi, \boldsymbol{X}_j^\phi) \approx \|\boldsymbol{y}_i - \boldsymbol{y}_j\|$ for $i, j \in 1, \ldots, K$. The embedding dimension L is a user-defined hyperparameter. This problem admits multiple formulations and optimization strategies [44], which could be systematically explored in future work. Our simple approach already yields promising results: we find that moderate embedding dimensions ($L \approx 20$) is sufficient to produce high-quality embeddings. We quantify the embedding distortions multiplicatively [47]:

$$\max\left(\theta_1(\boldsymbol{X}_i^\phi, \boldsymbol{X}_j^\phi)/\|\boldsymbol{y}_i - \boldsymbol{y}_j\|;\ \|\boldsymbol{y}_i - \boldsymbol{y}_j\|/\theta_1(\boldsymbol{X}_i^\phi, \boldsymbol{X}_j^\phi)\right) \tag{13}$$

for each pair of networks $i, j \in 1, \ldots K$. Plotting the distortions as a function of L (Fig. 4C), we see that they rapidly decrease, such that 95% of pairwise distances are distorted by, at most, ~5% (ABO data) or 10% (NAS-Bench-101) for sufficiently large L. Past work [10] has used multidimensional scaling heuristically to visualize collections of network representations in $L = 2$ dimensions. Our results here suggest that such a small value of L, while being amenable to visualization, results in a highly distorted embedding. It is noteworthy that the situation improves dramatically when

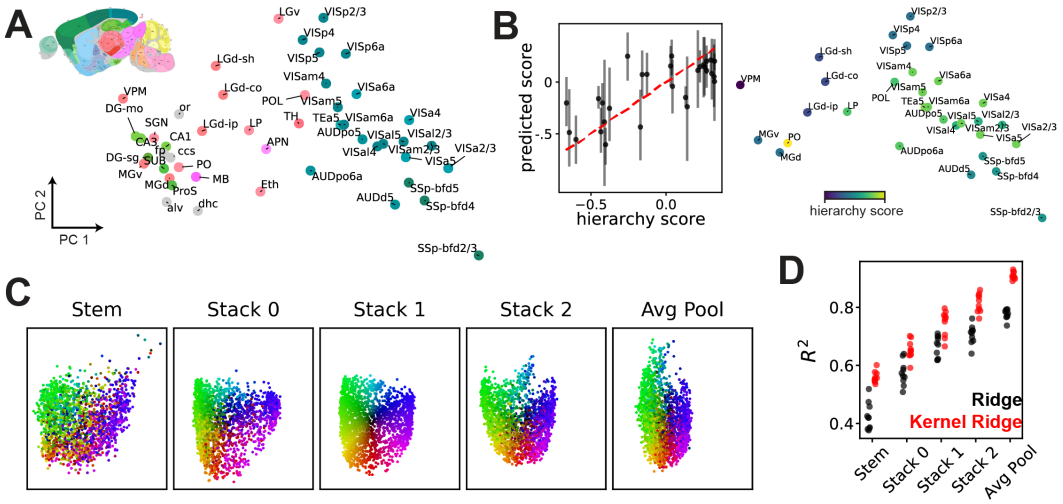

Figure 5: (A) PCA visualization of representations across 48 brain regions in the ABO dataset. Areas are colored by the reference atlas (see inset), illustrating a functional clustering of regions that maps onto anatomy. (B) *Left*, kernel regression predicts anatomical hierarchy [48] from embedded representations (see Supplement E). *Right*, PCA visualization of 31 areas labeled with hierarchy scores. (C) PCA visualization of 2000 network representations (a subset of NAS-Bench-101) across five layers, showing global structure is preserved across layers. Each network is colored by its position in the "Stack 1" layer (the middle of the architecture). (D) Embeddings of NAS-Bench-101 representations are predictive of test set accuracy, *even in very early layers*.

L is even modestly increased. While we cannot easily visualize these higher-dimensional vector embeddings, we can use them as features for downstream modeling tasks. This is well-motivated as an approximation to performing model inference in the true metric space that characterizes neural representations [47].

Anatomical structure and hierarchy is reflected in ABO representations. We can now collect the L-dimensional vector embeddings of K network representations into a matrix $\boldsymbol{Z} \in \mathbb{R}^{K \times L}$. The results in Fig. 4C imply that the distance between any two rows, $\|\boldsymbol{z}_i - \boldsymbol{z}_j\|$, closely reflects the distance between network representations i and j in shape space. We applied PCA to $\boldsymbol{Z}$ to visualize the $K = 48$ brain regions and found that anatomically related brain regions indeed were closer together in the embedded space (Fig. 5A): cortical and sub-cortical regions are separated along PC 1, and different layers of the same region (e.g. layers 2/3, 4, 5, and 6a of VISp) are clustered together. As expected from Fig. 4C, performing multidimensional scaling directly to a low-dimensional space ($L = 2$, as done in [10]) results in a qualitatively different outcome with distorted geometry (see Supplement E). Additionally, we used $\boldsymbol{Z}$ to fit an ensembled kernel regressor to predict an anatomical hierarchy score (defined in [48]) from the embedded vectors (Fig. 5B). Overall, these results demonstrate that the geometry of the learned embedding is scientifically interpretable and can be exploited for novel analyses, such as nonlinear regression. To our knowledge, the fine scale anatomical parcellation used here is novel in the context of representational similarity studies.

NAS-Bench-101 representations show persistent structure across layers. Since we collected representations across five layers in each deep network, the embedded representation vectors form a set of five $K \times L$ matrices, $\{\boldsymbol{Z}_1, \boldsymbol{Z}_2, \boldsymbol{Z}_3, \boldsymbol{Z}_4, \boldsymbol{Z}_5\}$. We aligned these embeddings by rotations in $\mathbb{R}^L$ via Procrustes analysis, and then performed PCA to visualize the $K = 2000$ network representations from each layer in a common low-dimensional space. We observe that many features of the global structure are remarkably well-preserved—two networks that are close together in the `Stack1` layer are assigned similar colors in Fig. 5C, and are likely to be close together in the other four layers. This preservation of representational similarity across layers suggests that even early layers contain signatures of network performance, which we expect to be present in the `AvgPool` layer. Indeed, when we fit ridge and RBF kernel ridge regressors to predict test set accuracy from representation embeddings, we see that even early layers support moderately good predictions (Fig. 5D). This is particularly surprising for the `Stem` layer. This is the first layer in each network, and its architecture is identical for all networks. Thus, the differences that are detected in the `Stem` layer result only from

differences in backpropagated gradients. Again, these results demonstrate the ability of generalized shape metrics to incorporate neural representations into analyses with greater scale (K corresponding to thousands of networks) and complexity (nonlinear kernel regression) than has been previously explored.

4 Conclusion and Limitations

We demonstrated how to ground analyses of neural representations in proper metric spaces. By doing so, we capture a number of theoretical advantages [16–20]. Further, we suggest new practical modeling approaches, such as using Euclidean embeddings to approximate the representational metric spaces. An important limitation of our work, as well as the past works we build upon, is the possibility that representational geometry is only loosely tied to higher-level algorithmic principles of network function [10]. On the other hand, analyses of representational geometry may provide insight into lower-level implementational principles [49]. Further, these analyses are highly scalable, as we demonstrated by analyzing thousands of networks—a much larger scale than is typically considered.

We used simple metrics (extensions of regularized CCA) in these analyses, but metrics that account for nonlinear transformations across neural representations are also possible as we document in Supplement C. The utility of these nonlinear extensions remains under-investigated and it is possible that currently popular linear methods are insufficient to capture structures of interest. For example, the topology of neural representations has received substantial interest in recent years [50–53]. Generalized shape metrics do not directly capture these topological features, and future work could consider developing new metrics that do so. A variety of recent developments in topological data analysis may be useful towards this end [54–56].

Finally, several of the metrics we described can be viewed as geodesic distances on Riemannian manifolds [35]. Future work would ideally exploit methods that are rigorously adapted to such manifolds, which are being actively developed [37]. Nonetheless, we found that optimized Euclidean embeddings, while only approximate, provide a practical off-the-shelf solution for large-scale surveys of neural representations.

Acknowledgments

We thank Ian Dryden (Florida International University), Søren Hauberg (Technical University of Denmark), and Nina Miolane (UC Santa Barbara) for fruitful discussions. A.H.W. was supported by the National Institutes of Health BRAIN initiative (1F32MH122998-01), and the Wu Tsai Stanford Neurosciences Institute Interdisciplinary Scholar Program. E. K. was supported by the Wu Tsai Stanford Neurosciences Institute Interdisciplinary Graduate Fellows Program. S.W.L. was supported by grants from the Simons Collaboration on the Global Brain (SCGB 697092) and the NIH BRAIN Initiative (U19NS113201 and R01NS113119).

References

[1] David GT Barrett, Ari S Morcos, and Jakob H Macke. "Analyzing biological and artificial neural networks: challenges with opportunities for synergy?" *Current Opinion in Neurobiology* 55 (2019). Machine Learning, Big Data, and Neuroscience, pp. 55–64.

[2] Nikolaus Kriegeskorte and Xue-Xin Wei. "Neural tuning and representational geometry". *Nature Reviews Neuroscience* (2021).

[3] Geoffrey Roeder, Luke Metz, and Durk Kingma. "On Linear Identifiability of Learned Representations". *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 9030–9039.

[4] Daniel L. K. Yamins, Ha Hong, Charles F. Cadieu, Ethan A. Solomon, Darren Seibert, and James J. DiCarlo. "Performance-optimized hierarchical models predict neural responses in higher visual cortex". *Proceedings of the National Academy of Sciences* 111.23 (2014), pp. 8619–8624.

[5] Santiago A. Cadena, Fabian H. Sinz, Taliah Muhammad, Emmanouil Froudarakis, Erick Cobos, Edgar Y. Walker, Jake Reimer, Matthias Bethge, Andreas Tolias, and Alexander S. Ecker. "How well do deep neural networks trained on object recognition characterize the mouse visual system?" *NeurIPS Workshop Neuro AI* (2019).

[6] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. "SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and Interpretability". *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 6076–6085.

[7] Ari Morcos, Maithra Raghu, and Samy Bengio. "Insights on representational similarity in neural networks with canonical correlation". *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Curran Associates, Inc., 2018, pp. 5727–5736.

[8] Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. "Representational similarity analysis - connecting the branches of systems neuroscience". *Frontiers in Systems Neuroscience* 2 (2008), p. 4.

[9] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. "Similarity of Neural Network Representations Revisited". Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, 2019, pp. 3519–3529.

[10] Niru Maheswaranathan, Alex Williams, Matthew Golub, Surya Ganguli, and David Sussillo. "Universality and individuality in neural dynamics across large populations of recurrent networks". *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 15629–15641.

[11] Thao Nguyen, Maithra Raghu, and Simon Kornblith. *Do Wide and Deep Networks Learn the Same Things? Uncovering How Neural Network Representations Vary with Width and Depth*. 2020.

[12] Jianghong Shi, Eric Shea-Brown, and Michael Buice. "Comparison Against Task Driven Artificial Neural Networks Reveals Functional Properties in Mouse Visual Cortex". *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 5764–5774.

[13] Christopher G. Small. *The statistical theory of shape*. Springer series in statistics. New York: Springer, 1996.

[14] David George Kendall, Dennis Barden, Thomas K Carne, and Huiling Le. *Shape and shape theory*. New York: Wiley, 1999.

[15] Ian L. Dryden and Kantilal Mardia. *Statistical shape analysis with applications in R*. Chichester, UK Hoboken, NJ: John Wiley & Sons, 2016.

[16] Peter N Yianilos. "Data structures and algorithms for nearest neighbor search in general metric spaces". *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*. 1993, pp. 311–321.

[17] Sanjoy Dasgupta and Philip M Long. "Performance guarantees for hierarchical clustering". *Journal of Computer and System Sciences* 70.4 (2005), pp. 555–569.

[18] Saaid Baraty, Dan A. Simovici, and Catalin Zara. "The Impact of Triangular Inequality Violations on Medoid-Based Clustering". *Foundations of Intelligent Systems*. Ed. by Marzena Kryszkiewicz, Henryk Rybinski, Andrzej Skowron, and Zbigniew W. Raś. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 280–289.

[19] Fei Wang and Jimeng Sun. "Survey on distance metric learning and dimensionality reduction in data mining". *Data Mining and Knowledge Discovery* 29.2 (2015), pp. 534–564.

[20] C. Chang, W. Liao, Y. Chen, and L. Liou. "A Mathematical Theory for Clustering in Metric Spaces". *IEEE Transactions on Network Science and Engineering* 3.1 (2016), pp. 2–16.

[21] Joshua H. Siegle et al. "Survey of spiking in the mouse visual system reveals functional hierarchy". *Nature* 592.7852 (2021), pp. 86–92.

[22] Chris Ying, Aaron Klein, Eric Christiansen, Esteban Real, Kevin Murphy, and Frank Hutter. "NAS-Bench-101: Towards Reproducible Neural Architecture Search". *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 7105–7114.

[23] Taco Cohen and Max Welling. "Group Equivariant Convolutional Networks". *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 2016, pp. 2990–2999.

[24] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. "Algorithms for learning kernels based on centered alignment". *The Journal of Machine Learning Research* 13.1 (2012), pp. 795–828.

[25] Mahdiyar Shahbazi, Ali Shirali, Hamid Aghajan, and Hamed Nili. "Using distance on the Riemannian manifold to compare representations in brain and in models". *NeuroImage* 239 (2021), p. 118271.

[26] Rainer Burkard, Mauro Dell'Amico, and Silvano Martello. *Assignment Problems*. Society for Industrial and Applied Mathematics, 2012.

[27] R Jonker and A Volgenant. "A shortest augmenting path algorithm for dense and sparse linear assignment problems". *Computing* 38.4 (1987), pp. 325–340.

[28] David F. Crouse. "On implementing 2D rectangular assignment algorithms". *IEEE Transactions on Aerospace and Electronic Systems* 52.4 (2016), pp. 1679–1696.

[29] Peter H. Schönemann. "A generalized solution of the orthogonal procrustes problem". *Psychometrika* 31.1 (1966), pp. 1–10.

[30] J. C. Gower and Garmt B. Dijksterhuis. *Procrustes problems*. Oxford New York: Oxford University Press, 2004.

[31] Agnan Kessy, Alex Lewin, and Korbinian Strimmer. "Optimal whitening and decorrelation". *The American Statistician* 72.4 (2018), pp. 309–314.

[32] Hrishikesh D Vinod. "Canonical ridge and econometrics of joint production". *Journal of econometrics* 4.2 (1976), pp. 147–166.

[33] P. L. Lai and C. Fyfe. "Kernel and Nonlinear Canonical Correlation Analysis". *International Journal of Neural Systems* 10.05 (2000). PMID: 11195936, pp. 365–377.

[34] John M. Lee. *Introduction to smooth manifolds*. 2nd ed. Graduate texts in mathematics 218. New York ; London: Springer, 2013.

[35] David G. Kendall. "Shape Manifolds, Procrustean Metrics, and Complex Projective Spaces". *Bulletin of the London Mathematical Society* 16.2 (1984), pp. 81–121.

[36] P. Thomas Fletcher and Sarang Joshi. "Riemannian geometry for the statistical analysis of diffusion tensor data". *Signal Processing* 87.2 (2007). Tensor Signal Processing, pp. 250–262.

[37] Nina Miolane, Nicolas Guigui, Alice Le Brigant, Johan Mathe, Benjamin Hou, Yann Thanwerdas, Stefan Heyder, Olivier Peltre, Niklas Koep, Hadi Zaatiti, Hatem Hajri, Yann Cabanes, Thomas Gerald, Paul Chauchat, Christian Shewmake, Daniel Brooks, Bernhard Kainz, Claire Donnat, Susan Holmes, and Xavier Pennec. "Geomstats: A Python Package for Riemannian Geometry in Machine Learning". *Journal of Machine Learning Research* 21.223 (2020), pp. 1–9.

[38] Aasa Feragen, Francois Lauze, and Soren Hauberg. "Geodesic Exponential Kernels: When Curvature and Linearity Conflict". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[39] Aasa Feragen and Søren Hauberg. "Open Problem: Kernel methods on manifolds and metric spaces. What is the probability of a positive definite geodesic exponential kernel?" *29th Annual Conference on Learning Theory*. Ed. by Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir. Vol. 49. Proceedings of Machine Learning Research. Columbia University, New York, New York, USA: PMLR, 2016, pp. 1647–1650.

[40] Sadeep Jayasumana, Mathieu Salzmann, Hongdong Li, and Mehrtash Harandi. "A Framework for Shape Analysis via Hilbert Space Embedding". *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2013.

[41] Ian L Dryden and Kanti V Mardia. "Multivariate shape analysis". *Sankhyā: The Indian Journal of Statistics, Series A* (1993), pp. 460–480.

[42] F. James Rohlf. “Shape Statistics: Procrustes Superimpositions and Tangent Spaces”. *Journal of Classification* 16.2 (1999), pp. 197–223.

[43] Ingwer Borg and Patrick JF Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

[44] Akshay Agrawal, Alnur Ali, and Stephen Boyd. “Minimum-Distortion Embedding”. *arXiv* (2021).

[45] Alexander J. Gates, Ian B. Wood, William P. Hetrick, and Yong-Yeol Ahn. “Element-centric clustering comparison unifies overlaps and hierarchy”. *Scientific Reports* 9.1 (2019), p. 8574.

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

[47] Leena Chennuru Vankadara and Ulrike von Luxburg. “Measures of distortion for machine learning”. *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., 2018.

[48] Julie A. Harris et al. “Hierarchical organization of cortical and thalamic connectivity”. *Nature* 575.7781 (2019), pp. 195–202.

[49] Jess B. Hamrick and Shakir Mohamed. “Levels of Analysis for Machine Learning”. *Proceedings of the ICLR 2020 Workshop on Bridging AI and Cognitive Science*. 2020.

[50] Erik Rybakken, Nils Baas, and Benjamin Dunn. “Decoding of Neural Data Using Cohomological Feature Extraction”. *Neural Computation* 31.1 (2019), pp. 68–93.

[51] Rishidev Chaudhuri, Berk Gerçek, Biraj Pandey, Adrien Peyrache, and Ila Fiete. “The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep”. *Nature Neuroscience* 22.9 (2019), pp. 1512–1520.

[52] Tevin C. Rouse, Amy M. Ni, Chengcheng Huang, and Marlene R. Cohen. “Topological insights into the neural basis of flexible behavior”. *bioRxiv* (2021).

[53] Richard J. Gardner, Erik Hermansen, Marius Pachitariu, Yoram Burak, Nils A. Baas, Benjamin A. Dunn, May-Britt Moser, and Edvard I. Moser. “Toroidal topology of population activity in grid cells”. *bioRxiv* (2021).

[54] Genki Kusano, Kenji Fukumizu, and Yasuaki Hiraoka. “Kernel Method for Persistence Diagrams via Kernel Embedding and Weight Factor”. *Journal of Machine Learning Research* 18.189 (2018), pp. 1–41.

[55] Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. “Topological Autoencoders”. *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 7045–7054.

[56] Kristopher Jensen, Ta-Chu Kao, Marco Tripodi, and Guillaume Hennequin. “Manifold GPLVMs for discovering non-Euclidean latent structure in neural data”. *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 22580–22592.

[57] Viivi Uurtio, João M. Monteiro, Jaz Kandola, John Shawe-Taylor, Delmiro Fernandez-Reyes, and Juho Rousu. “A Tutorial on Canonical Correlation Methods”. *ACM Comput. Surv.* 50.6 (2017).

[58] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. *Nature Methods* 17 (2020), pp. 261–272.

[59] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. “Canonical correlation analysis: an overview with application to learning methods”. *Neural Comput.* 16.12 (2004), pp. 2639–2664.

[60] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. “Kernel methods in machine learning”. *Ann. Statist.* 36.3 (2008), pp. 1171–1220.

[61] Huiling Le. “On geodesics in Euclidean shape spaces”. *Journal of the London Mathematical Society* 2.2 (1991), pp. 360–372.

[62] Nello Cristianini, John Shawe-Taylor, André Elisseeff, and Jaz Kandola. “On Kernel-Target Alignment”. *Advances in Neural Information Processing Systems*. Ed. by T. Dietterich, S. Becker, and Z. Ghahramani. Vol. 14. MIT Press, 2002.

[63] Anastasios Tsiatis. *Semiparametric theory and missing data*. Springer Science & Business Media, 2007.

[64] Quanxin Wang et al. "The Allen Mouse Brain Common Coordinate Framework: A 3D Reference Atlas". *Cell* 181.4 (2020), 936–953.e20.

Supplemental Information: Generalized Shape Metrics on Neural Representations

This supplement is organized into five sections. First, in Supplement A, we review background material on metric spaces and other relevant mathematical concepts. In Supplement B, we prove the two propositions that appear in the main text. Supplement C collects together several miscellaneous results which demonstrate that generalized shape metrics include similarity measures based on CCA, kernel CCA, and geodesic distance on Kendall's shape space. In Supplement D, we outline an extension and reinterpretation of generalized shape metrics to stochastic random variables. This extension represents a rich opportunity for future research and also provides a better foundation to interpret the results presented in Fig. 3 of the main text, which empirically characterize the number of images needed to estimate the distance between two neural networks. Finally, in Supplement E, we collect additional methodological details about the experiments we present in the main text.

A Background

A.1 Notation

Vectors in real coordinate space are denoted in boldface with lowercase letters, e.g. $\boldsymbol{x} \in \mathbb{R}^n$. Matrices are denoted in boldface with uppercase letters, e.g. $\boldsymbol{X} \in \mathbb{R}^{m \times n}$. We use the same notation to denote linear operators, e.g. $\boldsymbol{T} \in \mathcal{G}$ where $\mathcal{G}$ is a set of linear operators.

Letters in regular type face, e.g. x or X, may denote scalars or elements of some abstract vector space, with the distinction being made clear from context. For example, the space of random variables with outcomes over $\mathbb{R}^n$ defines a vector space that we will see is compatible with the basic framework of generalized shape metrics. This extension of shape metrics to stochastic layers and neural responses is outlined in Supplement D.

If $\boldsymbol{T}$ is a linear operator on some vector space, and X is a vector within this space, we will use $\boldsymbol{T}X$ to denote the transformation $X \mapsto \boldsymbol{T}(X)$. Further, if $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ are linear operators, we write $\boldsymbol{T}_1\boldsymbol{T}_2X$ in place of $\boldsymbol{T}_1(\boldsymbol{T}_2(X))$, and we use $\boldsymbol{T}_1\boldsymbol{T}_2$ to denote the composition of the two linear operators. These notational choices intuitively draw parallels with matrix-vector and matrix-matrix multiplication, respectively.

A.2 Metrics

Here we revisit our definition of a metric given in the main text to provide more rigorous details and clarify the role of the equivalence relation.

Definition 1. *A **metric** on a set $\mathcal{S}$ is a function $\mathcal{S} \times \mathcal{S} \mapsto \mathbb{R}_+$, which satisfies, for all $X, Y, M \in \mathcal{S}$, the following three conditions:*

- *Identity.* $d(X, Y) = 0$ *if and only if* $X = Y$
- *Symmetry.* $d(X, Y) = d(Y, X)$
- *Triangle Inequality.* $d(X, Y) \leq d(X, M) + d(M, Y)$

We have seen that it is useful to relax the first condition (*Identity*) to an equivalence relation. That is, rather than strict equality, we demand that $d(X, Y) = 0$ if and only if $X \sim Y$, for some specified equivalence relation $\sim$. In this scenario, the distance function is *not*, strictly speaking, a metric on $\mathcal{S}$. However, it still does define a metric on the appropriate *quotient set*, which we now define.

Definition 2. *Let $\sim$ denote an equivalence relation defined on some set $\mathcal{S}$. Then given any $M \in \mathcal{S}$, we can define the set of all elements equivalent to M as $\{X \in \mathcal{S} \mid X \sim M\}$, which is called the **equivalence class** of the element M. The set of all equivalence classes, denoted $\mathcal{S}/\sim$, is called the **quotient set** of $\mathcal{S}$ with respect to the specified equivalence relation.*

For example, the Euclidean distance $\|\boldsymbol{x} - \boldsymbol{y}\|$ is a metric on the set of vectors in $\mathbb{R}^n$. The angular distance $\arccos(\boldsymbol{x}^\top \boldsymbol{y} / \sqrt{\boldsymbol{x}^\top \boldsymbol{x} \cdot \boldsymbol{y}^\top \boldsymbol{y}})$ is not a metric on $\mathbb{R}^n$, but it defines a metric between sets of points contained in rays emanating from the origin (i.e. points in $\mathbb{R}^n$ with an equivalence relation given by nonnegative scaling). These technical distinctions above are not central to our story, so we will often refer to a function as a "metric" without explicitly defining what set it acts upon. In all cases, it should be understood as the quotient set defined by the specified equivalence relation.

A.3 Hilbert spaces

A **vector space** $\mathcal{H}$ is a collection of objects (called vectors) that are equipped with two operations: vector addition (given $X \in \mathcal{H}$ and $Y \in \mathcal{H}$ we have $X + Y \in \mathcal{H}$) and scalar multiplication (given $X \in \mathcal{H}$ and $\alpha \in \mathbb{R}$ we have $\alpha X \in \mathcal{H}$). An **inner product space** is a vector space that is additionally equipped with a function $\mathcal{H} \times \mathcal{H} \mapsto \mathbb{R}$, called the *inner product*, which is denoted with angle brackets $\langle \cdot, \cdot \rangle$ and satisfies:

- *Symmetry.* $\langle X, Y\rangle = \langle Y, X\rangle$
- *Linearity.* $\langle Z + \alpha X, Y\rangle = \langle Z, Y\rangle + \alpha\langle X, Y\rangle$
- *Positive Definiteness.* $\langle X, X\rangle \geq 0$ with equality if and only if $X = 0$

A **Hilbert space** is an inner product space that satisfies an additional technical requirement (all Cauchy sequences of vectors in $\mathcal{H}$ converge to a limit in $\mathcal{H}$).

The set of vectors in $\mathbb{R}^n$ defines a Hilbert space, where the inner product corresponds to the usual dot product. Similarly, the set of matrices in $\mathbb{R}^{m\times n}$, equipped with the Frobenius inner product $\langle \boldsymbol{X}, \boldsymbol{Y}\rangle = \mathrm{Tr}[\boldsymbol{X}^\top \boldsymbol{Y}]$ also defines a Hilbert space. In Supplement D, we will exploit the fact that random vectors over $\mathbb{R}^n$ also define a Hilbert space where the inner product is given by the expectation of the dot product. This enables us to extend the framework of generalized shape metrics to stochastic neural layers.

A.4 Euclidean and Angular Distances in Hilbert Spaces

One of the most fundamental properties of a Hilbert space is the **Cauchy-Schwarz inequality**,

$$|\langle X, Y\rangle| \leq \|X\|\|Y\| \quad \text{for all} \quad (X, Y) \in \mathcal{H} \times \mathcal{H}, \tag{14}$$

which can be derived from the properties of the inner product. Using this, we can verify that the norm is sub-additive:

$$\|X + Y\| \leq \|X\| + \|Y\| \quad \text{for all} \quad (X, Y) \in \mathcal{H} \times \mathcal{H}, \tag{15}$$

Defining $d_{\text{euc}}(X, Y) = \|X - Y\|$ to be the generalization of Euclidean distance to Hilbert spaces, we see that triangle inequality follows immediately:

$$d_{\text{euc}}(X, Y) = \|X - Y\| = \|X - M + M - Y\| \leq \|X - M\| + \|M - Y\| = d_{\text{euc}}(X, Y) + d_{\text{euc}}(M, Y) \tag{16}$$

for all choices of X, Y, and M in $\mathcal{H}$. Euclidean distance evidently satisfies the remaining two properties of a metric—symmetry and nonnegativity.

The angular distance is defined as:

$$d_\theta(X, Y) = \arccos\left[\frac{\langle X, Y\rangle}{\|X\|\|Y\|}\right] \tag{17}$$

The Cauchy-Schwarz inequality implies that the argument to $\arccos(\cdot)$ is always within its domain (i.e. on the interval $[-1, 1]$). The angular distance is a metric over equivalence classes defined by nonnegative scaling: formally, $X \sim Y$ if and only if there exists an $s > 0$ such that $X = sY$. Geometrically, one can think of $d_\theta(X, Y)$ as the geodesic path length between points on a sphere. Intuitively, this is nonnegative, symmetric, and obeys the triangle inequality. We provide a short proof that the triangle inequality is indeed satisfied below.

Proof: *Angular distance satisfies the triangle inequality.* Consider three unit-norm vectors: X, Y, and Z. The triangle inequality trivially holds if any pair of X, Y, and Z are equal, so we can assume X, Y, and Z are distinct. Now define two vectors U and V as follows:

$$U = X - Y\langle X, Y\rangle \tag{18}$$
$$V = Z - Y\langle Z, Y\rangle \tag{19}$$

Note that $\langle U, Y\rangle = 0$ and $\langle V, Y\rangle = 0$. Thus, we can interpret U as the part of X that is orthogonal to Y. Likewise, we can interpret V as the part Z that is orthogonal to Y. Further, we have:

$$X = Y\langle X, Y\rangle + U\langle X, U\rangle = Y\cos\theta_{XY} + U\sin\theta_{XY} \tag{20}$$
$$Z = Y\langle Z, Y\rangle + V\langle Z, V\rangle = Y\cos\theta_{ZY} + V\sin\theta_{ZY} \tag{21}$$

where we introduced the shorthand $\theta_{XY} = d_\theta(X, Y)$ for concision. Now,

$$\cos\theta_{XZ} = \langle X, Z\rangle = \langle Y\cos\theta_{XY} + U\sin\theta_{XY}, Y\cos\theta_{ZY} + V\sin\theta_{ZY}\rangle \tag{22}$$
$$= \cos\theta_{XY}\cos\theta_{ZY} + \langle U, V\rangle\sin\theta_{XY}\sin\theta_{ZY} \tag{23}$$
$$\geq \cos\theta_{XY}\cos\theta_{ZY} - \sin\theta_{XY}\sin\theta_{ZY} \tag{24}$$
$$= \cos(\theta_{XY} + \theta_{ZY}) \tag{25}$$

On line (23), many terms simplify since $\langle Y, Y\rangle = 1$, and $\langle U, Y\rangle = \langle V, Y\rangle = 0$. To introduce the inequality on line (24), notice that the Cauchy-Schwarz inequality implies $\langle U, V\rangle \geq -1$. Thus, replacing $\langle U, V\rangle$ with -1 produces a lower bound on $\cos\theta_{XZ}$ since $\sin\theta_{XY}\sin\theta_{ZY} \geq 0$. The final step on line (25) applies an elementary identity from trigonometry. Overall, we have $\cos\theta_{XZ} \geq \cos(\theta_{XY} + \theta_{ZY})$. This directly implies the desired triangle inequality, $\theta_{XZ} \leq \theta_{XY} + \theta_{ZY}$, since $\arccos(\cdot)$ is a monotonically decreasing function. □

A.5 The Orthogonal Group

Another important feature of Hilbert spaces is the notion of an orthogonal transformation. These are linear transformations which preserve the inner product. Below, we also define the familiar transpose operator for a general Hilbert space.

Definition 3. *An **orthogonal transformation** on a Hilbert space $\mathcal{H}$ is any linear transformation $\boldsymbol{Q}$, which satisfies $\langle \boldsymbol{Q}X, \boldsymbol{Q}Y \rangle = \langle X, Y \rangle$ for any choice of $X \in \mathcal{H}$ and $Y \in \mathcal{H}$.*

Definition 4. *Let $\boldsymbol{W} : \mathcal{V} \mapsto \mathcal{V}$ be a linear transformation on a Hilbert space $\mathcal{V}$. For any choice of $\boldsymbol{W}$, there is a unique linear transformation $\boldsymbol{W}^\top$, called the **transpose** (or adjoint) of $\boldsymbol{W}$, which is denoted $\boldsymbol{W}^\top$ and which satisfies $\langle \boldsymbol{W}X, Y \rangle = \langle X, \boldsymbol{W}^\top Y \rangle$ for any choice of $X \in \mathcal{V}$ and $Y \in \mathcal{V}$.*

Let $\boldsymbol{Q}$ be orthogonal. Since $\langle X, Y \rangle = \langle \boldsymbol{Q}X, \boldsymbol{Q}Y \rangle = \langle X, \boldsymbol{Q}^\top \boldsymbol{Q}Y \rangle$, we see that $\boldsymbol{Q}^\top \boldsymbol{Q}$ is the identity transformation and thus $\boldsymbol{Q}^\top$ and $\boldsymbol{Q}$ are inverses. One can show that these inverses commute, and thus $\boldsymbol{Q}^\top$ is also orthogonal since $\langle \boldsymbol{Q}^\top X, \boldsymbol{Q}^\top Y \rangle = \langle \boldsymbol{Q}\boldsymbol{Q}^\top X, \boldsymbol{Q}\boldsymbol{Q}^\top Y \rangle = \langle X, Y \rangle$. Finally, let $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$ be any pair of orthogonal transformations on $\mathcal{V}$. Then, the composition of these transformations $\boldsymbol{Q}_2\boldsymbol{Q}_1$ is evidently orthogonal, since: $\langle X, Y \rangle = \langle \boldsymbol{Q}_1 X, \boldsymbol{Q}_1 Y \rangle = \langle \boldsymbol{Q}_2\boldsymbol{Q}_1 X, \boldsymbol{Q}_2\boldsymbol{Q}_1 Y \rangle$.

In summary, we have just shown that the inverse of every orthogonal matrix is also orthogonal and orthogonal transformations are closed under composition. This shows that the set of orthogonal transformations on a Hilbert space fulfills the axioms of a **group**, as defined below:

Definition 5. *A **group** is a set $\mathcal{G}$ equipped with a binary operation that maps two elements of $\mathcal{G}$ onto another element of $\mathcal{G}$, which satisfies:*

1. *Associativity: For all $\boldsymbol{T}_1, \boldsymbol{T}_2, \boldsymbol{T}_3$ in $\mathcal{G}$, one has $(\boldsymbol{T}_1\boldsymbol{T}_2)\boldsymbol{T}_3 = \boldsymbol{T}_1(\boldsymbol{T}_2\boldsymbol{T}_3)$.*
2. *Identity element: There exists a unique element $\boldsymbol{I} \in \mathcal{G}$ such that $\boldsymbol{I}\boldsymbol{T} = \boldsymbol{T}\boldsymbol{I} = \boldsymbol{T}$ for all $\boldsymbol{T} \in \mathcal{G}$.*
3. *Invertibility: For every $\boldsymbol{T} \in \mathcal{G}$ there exists another element $\boldsymbol{T}^{-1} \in \mathcal{G}$ such that $\boldsymbol{T}\boldsymbol{T}^{-1} = \boldsymbol{T}^{-1}\boldsymbol{T} = \boldsymbol{I}$.*

Here, we are only interested in groups of *linear functions*, so the "binary operation" referred to above is function composition (see Section A.1 for notational conventions regarding linear operators).

We are particularly interested in groups of (linear) transformations that preserve distances. Such transformations are called (linear) **isometries**, which we define below.

Definition 6. *Let $\mathcal{S}$ be a set and let $d : \mathcal{S} \times \mathcal{S} \mapsto \mathbb{R}_+$ be a metric on this set. Then a transformation $\boldsymbol{T} : \mathcal{S} \mapsto \mathcal{S}$, is called an **isometry** on the metric space $(d, \mathcal{S})$ if $d(\boldsymbol{T}X, \boldsymbol{T}Y) = d(X, Y)$ for all $X, Y \in \mathcal{S}$.*

It is easy to see that the orthogonal group is a group of isometries with respect to the (generalized) Euclidean and angular distance metrics. For the Euclidean distance we have:

$$\begin{aligned} d^2_{\text{euc}}(X, Y) = \|X - Y\|^2 &= \langle X, X \rangle + \langle Y, Y \rangle - 2\langle X, Y \rangle \\ &= \langle \boldsymbol{Q}X, \boldsymbol{Q}X \rangle + \langle \boldsymbol{Q}Y, \boldsymbol{Q}Y \rangle - 2\langle \boldsymbol{Q}X, \boldsymbol{Q}Y \rangle \\ &= \|\boldsymbol{Q}X - \boldsymbol{Q}Y\|^2 = d^2_{\text{euc}}(\boldsymbol{Q}X, \boldsymbol{Q}Y) \end{aligned} \tag{26}$$

For the angular distance we have:

$$\cos\left[d_\theta(X, Y)\right] = \frac{\langle X, Y \rangle}{\|X\|\|Y\|} = \frac{\langle \boldsymbol{Q}X, \boldsymbol{Q}Y \rangle}{\|\boldsymbol{Q}X\|\|\boldsymbol{Q}Y\|} = \cos\left[d_\theta(\boldsymbol{Q}X, \boldsymbol{Q}Y)\right] \tag{27}$$

B Proof of Propositions 1 & 2

Both propositions in the main text follow immediately as special cases of the following result, which states that minimizing any metric over a group of isometries results in a metric on the corresponding quotient space. After proving this result we conclude this section by briefly outlining these special cases.

Proposition (A generalization of Propositions 1 & 2)**.** *Let $(g, \mathcal{H})$ be a metric space, where $g : \mathcal{H} \times \mathcal{H} \mapsto \mathbb{R}_+$ denotes the distance function. Let $\mathcal{G}$ be a group of isometries on this metric space. Then the function:*

$$h(X, Y) = \min_{\boldsymbol{T} \in \mathcal{G}} g(X, \boldsymbol{T}Y) \tag{28}$$

defines a metric over the quotient space $\mathcal{H}/\sim$ where the equivalence relation is $X \sim Y$ if and only if $X = \boldsymbol{T}Y$ for some $\boldsymbol{T} \in \mathcal{G}$.

Proof. First, define $\boldsymbol{T}_{XY} = \operatorname{argmin}_{\boldsymbol{T} \in \mathcal{G}} g(X, \boldsymbol{T}Y)$. So, $h(X, Y) = g(X, \boldsymbol{T}_{XY}Y)$. Since g is a metric, $g(X, Y) = 0$ if and only if $X = Y$. Thus, $h(X, Y) = 0$ if and only if $X = \boldsymbol{T}_{XY}Y$, or equivalently if $X \sim Y$ by the stated equivalence relation.

Next, we prove that $h(X,Y) = h(Y,X)$. By the group axioms, every element in $\mathcal{G}$ is invertible by another element in the set, so $\boldsymbol{T}_{XY}^{-1} \in \mathcal{G}$. Further, every element of $\mathcal{G}$ is an isometry with respect to g. Thus,

$$h(X,Y) = g(X, \boldsymbol{T}_{XY}Y) = g(\boldsymbol{T}_{XY}^{-1}X, \boldsymbol{T}_{XY}^{-1}\boldsymbol{T}_{XY}Y) = g(Y, \boldsymbol{T}_{XY}^{-1}X) \geq g(Y, \boldsymbol{T}_{YX}X) = h(Y,X)\,, \tag{29}$$

where the inequality follows from replacing $\boldsymbol{T}_{XY}^{-1}$ with the optimal $\boldsymbol{T}_{YX} = \operatorname{argmin}_{\boldsymbol{T}\in\mathcal{G}} g(Y, \boldsymbol{T}X)$. However, by the same chain of logic, we also have:

$$h(Y,X) = g(Y, \boldsymbol{T}_{YX}X) = g(\boldsymbol{T}_{YX}^{-1}Y, \boldsymbol{T}_{YX}^{-1}\boldsymbol{T}_{YX}X) = g(X, \boldsymbol{T}_{YX}^{-1}Y) \geq g(X, \boldsymbol{T}_{XY}Y) = h(X,Y)\,. \tag{30}$$

Thus, we have $h(X,Y) \geq h(Y,X)$, but also $h(Y,X) \geq h(X,Y)$. We conclude $h(X,Y) = h(Y,X)$ and $\boldsymbol{T}_{XY}^{-1} = \boldsymbol{T}_{YX}$.

It remains to prove the triangle inequality. This is done by the following sequence:

$$\begin{aligned}
h(X,Y) &= g(X, \boldsymbol{T}_{XY}Y) && (31)\\
&\leq g(X, \boldsymbol{T}_{XZ}\boldsymbol{T}_{ZY}Y) && (32)\\
&\leq g(X, \boldsymbol{T}_{XZ}Z) + g(\boldsymbol{T}_{XZ}Z, \boldsymbol{T}_{XZ}\boldsymbol{T}_{ZY}Y) && (33)\\
&= g(X, \boldsymbol{T}_{XZ}Z) + g(Z, \boldsymbol{T}_{ZY}Y) && (34)\\
&= h(X,Z) + h(Z,Y) && (35)
\end{aligned}$$

The first inequality follows from replacing the optimal alignment, $\boldsymbol{T}_{XY}$, with a sub-optimal alignment $\boldsymbol{T}_{XZ}\boldsymbol{T}_{ZY}$. The second inequality follows from the triangle inequality on g, after choosing $\boldsymbol{T}_{XZ}Z$ as the midpoint. The penultimate step follows from $\boldsymbol{T}_{XZ}$ being an isometry on g. □

Relation to Proposition 1 The space $\mathcal{H}$ corresponds to $\mathbb{R}^{m\times p}$, which is equipped with the typical Frobenius inner product. The distance function g is Euclidean distance, see Eq. (16). The group $\mathcal{G}$ corresponds to any group of linear isometries which can be expressed as a matrix multiplication on the right. That is, any transformation from $\mathbb{R}^{m\times p} \mapsto \mathbb{R}^{m\times p}$ that can be expressed as $\boldsymbol{X} \mapsto \boldsymbol{X}\boldsymbol{M}$ for some $\boldsymbol{M} \in \mathbb{R}^{p\times p}$.

Relation to Proposition 2 The space $\mathcal{H}$ corresponds to $\mathbb{S}^{m\times p}$ (the "sphere" of $m \times p$ matrices with unit Frobenius norm). The distance function g is the angular distance, see Eq. (17). The group $\mathcal{G}$ is defined as done directly above in our discussion of Proposition 1.

C Connections to Other Methods

This section describes the connections between generalized shape metrics and existing representational similarity measures in greater detail. For simplicity, we consider quantifying the similarity between two networks with n neurons or hidden layer units. We use $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{Y} \in \mathbb{R}^{m\times n}$ to denote matrices holding the hidden layer activations of two networks over m common test inputs. In many cases, networks have distinct numbers of neurons or hidden units; however, this can be accommodated by applying PCA or zero-padding representations to achieve a common dimension.

For further simplicity, we will assume that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are mean-centered such that $\boldsymbol{X}^\top \mathbf{1}_n = \boldsymbol{Y}^\top \mathbf{1}_n = \mathbf{0}_n$, where $\mathbf{0}_n$ and $\mathbf{1}_n$ respectively denote an n-dimensional vector of zeros and ones. Intuitively, this mean-centering removes the effect of translations in neural activation space when computing distances between neural representations. In the main text, we show this mean-centering step explicitly as a centering matrix $\boldsymbol{C} \in \mathbb{R}^{m\times m}$ that is included in the feature map, ϕ. The mean-centering step is not strictly required, but is a typical preprocessing step in canonical correlations analysis [57] and Procrustes analysis [15].

C.1 Permutation Invariance & Linear Assignment Problems

Consider the problem of finding the best permutation matrix which matches two sets of neural activations in terms of Euclidean distance. That is, we seek to find

$$\boldsymbol{\Pi}^* = \operatorname*{argmin}_{\boldsymbol{\Pi}\in\mathcal{P}} \left\|\boldsymbol{X} - \boldsymbol{Y}\boldsymbol{\Pi}\right\|, \tag{36}$$

where $\mathcal{P}$ is the set of $n \times n$ permutation matrices. Note that this is equivalent to finding the permutation matrix that minimizes squared Euclidean distance, and that:

$$\left\|\boldsymbol{X} - \boldsymbol{Y}\boldsymbol{\Pi}\right\|^2 = \langle \boldsymbol{X}, \boldsymbol{X}\rangle + \langle \boldsymbol{Y}, \boldsymbol{Y}\rangle - 2\langle \boldsymbol{X}, \boldsymbol{Y}\boldsymbol{\Pi}\rangle\,. \tag{37}$$

Since $\langle \boldsymbol{X}, \boldsymbol{X}\rangle$ and $\langle \boldsymbol{Y}, \boldsymbol{Y}\rangle$ are constant terms, the minimization in (36) is equivalent to:

$$\boldsymbol{\Pi}^* = \operatorname*{argmin}_{\boldsymbol{\Pi}\in\mathcal{P}} \; -2\langle \boldsymbol{X}, \boldsymbol{Y}\boldsymbol{\Pi}\rangle = \operatorname*{argmax}_{\boldsymbol{\Pi}\in\mathcal{P}} \; \langle \boldsymbol{X}, \boldsymbol{Y}\boldsymbol{\Pi}\rangle = \operatorname*{argmin}_{\boldsymbol{\Pi}\in\mathcal{P}} \; d_\theta(\boldsymbol{X}, \boldsymbol{Y}\boldsymbol{\Pi}). \tag{38}$$

The final equality holds since the angular distance is given by a monotonically decreasing function (i.e., $\arccos$) of the maximized inner product. Finally, using the definition of the Frobenius inner product, $\langle \boldsymbol{X}, \boldsymbol{Y}\boldsymbol{\Pi}\rangle = \operatorname{Tr}[\boldsymbol{X}^\top \boldsymbol{Y}\boldsymbol{\Pi}]$, and so,

$$\boldsymbol{\Pi}^* = \operatorname*{argmax}_{\boldsymbol{\Pi}\in\mathcal{P}} \; \operatorname{Tr}[\boldsymbol{X}^\top \boldsymbol{Y}\boldsymbol{\Pi}]\,. \tag{39}$$

This final reformulation is the well-known *linear assignment problem* [26]. This can be solved efficiently in $O(n^3)$ time using standard algorithms [28], which are readily available in standard scientific computing environments. For example, the function `scipy.optimize.linear_sum_assignment` provides an implementation in Python [58].

C.2 Orthogonal Procrustes Problems

Instead of optimizing over permutations, we may wish to optimize over orthogonal transformations. Given two matrices $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{Y} \in \mathbb{R}^{m \times n}$, we seek to find

$$\boldsymbol{Q}^* = \operatorname*{argmin}_{\boldsymbol{Q} \in \mathcal{O}} \|\boldsymbol{X} - \boldsymbol{Y}\boldsymbol{Q}\|, \tag{40}$$

where $\mathcal{O}$ is the set of $n \times n$ orthogonal matrices. This is known as the orthogonal Procrustes problem [30]. Following the same steps as above in Section C.1, we can see that $\boldsymbol{Q}^*$ also minimizes the angular distance between two matrices, and maximizes their inner product:

$$\boldsymbol{Q}^* = \operatorname*{argmax}_{\boldsymbol{Q} \in \mathcal{O}} \langle \boldsymbol{X}, \boldsymbol{Y}\boldsymbol{Q} \rangle = \operatorname*{argmin}_{\boldsymbol{Q} \in \mathcal{O}} d_\theta(\boldsymbol{X}, \boldsymbol{Y}\boldsymbol{Q}). \tag{41}$$

The following lemma states the well-known solution to this problem, which is due to Schönemann [29].

Lemma 1 (Schönemann [29])**.** *Let $\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$ denote the singular value decomposition of $\boldsymbol{X}^\top\boldsymbol{Y}$. Then $\boldsymbol{Q}^* = \boldsymbol{U}\boldsymbol{V}^\top$. Furthermore,*

$$\left\langle \boldsymbol{X}, \boldsymbol{Y}\boldsymbol{Q}^* \right\rangle = \|\boldsymbol{X}^\top\boldsymbol{Y}\|_* = \sum_i \sigma_i \tag{42}$$

where $\|\cdot\|_*$ denotes the nuclear matrix norm and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$ are the singular values of $\boldsymbol{X}^\top\boldsymbol{Y}$.

Proof. Let $\boldsymbol{Z} = \boldsymbol{V}^\top\boldsymbol{Q}\boldsymbol{U}$, and note that $\boldsymbol{Z}$ is orthogonal because orthogonal transformations are closed under composition. The cyclic property of the trace operator implies,

$$\max_{\boldsymbol{Q} \in \mathcal{Q}} \left\langle \boldsymbol{X}, \boldsymbol{Y}\boldsymbol{Q} \right\rangle = \max_{\boldsymbol{Q} \in \mathcal{Q}} \operatorname{Tr}[\boldsymbol{X}^\top\boldsymbol{Y}\boldsymbol{Q}] = \max_{\boldsymbol{Q} \in \mathcal{Q}} \operatorname{Tr}[\boldsymbol{S}\boldsymbol{V}^\top\boldsymbol{Q}\boldsymbol{U}] = \max_{\boldsymbol{Z} \in \mathcal{Q}} \operatorname{Tr}[\boldsymbol{S}\boldsymbol{Z}] = \max_{\boldsymbol{Z} \in \mathcal{Q}} \sum_{i=1}^n \sigma_i z_{ii} \tag{43}$$

where $\{z_{ii}\}_{i=1}^n$ are the diagonal elements of $\boldsymbol{Z}$. Since $\boldsymbol{Z}$ is orthogonal, we must have $z_{ii} \leq 1$ for all $i \in \{1, \ldots, n\}$. Since the singular values are nonnegative, the maximum is obtained when each $z_{ii} = 1$. That is, at optimality we have $\boldsymbol{Z} = \boldsymbol{V}^\top\boldsymbol{Q}^*\boldsymbol{U} = \boldsymbol{I}$, which implies $\boldsymbol{Q}^* = \boldsymbol{V}\boldsymbol{U}^\top$. Plugging $z_{ii} = 1$ into the final expression of Eq. (43) shows that the optimal objective is given by the sum of the singular values (i.e. the nuclear norm of $\boldsymbol{X}^\top\boldsymbol{Y}$). □

C.3 Canonical Correlation Analysis (CCA)

CCA identifies matrices $\boldsymbol{W}_x \in \mathbb{R}^{n \times n}$ and $\boldsymbol{W}_y \in \mathbb{R}^{n \times n}$ which maximize the correlation between $\boldsymbol{X}\boldsymbol{W}_x$ and $\boldsymbol{Y}\boldsymbol{W}_y$. Formally, this corresponds to the optimization problem:

$$\begin{aligned} \underset{\boldsymbol{W}_x, \boldsymbol{W}_y}{\text{maximize}} \quad & \operatorname{Tr}[\boldsymbol{W}_x^\top\boldsymbol{X}^\top\boldsymbol{Y}\boldsymbol{W}_y] \\ \text{subject to} \quad & \boldsymbol{W}_x^\top\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{W}_x = \boldsymbol{W}_y^\top\boldsymbol{Y}^\top\boldsymbol{Y}\boldsymbol{W}_y = \boldsymbol{I}. \end{aligned} \tag{44}$$

The maximized objective function, $\langle \boldsymbol{X}\boldsymbol{W}_x, \boldsymbol{Y}\boldsymbol{W}_y \rangle = \operatorname{Tr}[\boldsymbol{W}_x^\top\boldsymbol{X}^\top\boldsymbol{Y}\boldsymbol{W}_y]$, generalizes the dot product between two vectors to the Frobenius inner product between $\boldsymbol{X}\boldsymbol{W}_x$ and $\boldsymbol{Y}\boldsymbol{W}_y$. The constraints of the optimization problem constrain the magnitude of the solution—without these constraints, the objective function could be infinitely large, since multiplying $\boldsymbol{W}_x$ or $\boldsymbol{W}_y$ by a real number larger than one proportionally increases $\langle \boldsymbol{X}\boldsymbol{W}_x, \boldsymbol{Y}\boldsymbol{W}_y \rangle$. Intuitively, the typical (Pearson) correlation is equal to the normalized inner product of two vectors, and CCA generalizes this to matrix-valued datasets.

CCA can be transformed into the Procrustes problem by a change of variables. Assuming that $\boldsymbol{X}^\top\boldsymbol{X}$ and $\boldsymbol{Y}^\top\boldsymbol{Y}$ are full rank, define $\boldsymbol{H}_x = (\boldsymbol{X}^\top\boldsymbol{X})^{1/2}\boldsymbol{W}_x$ and $\boldsymbol{H}_y = (\boldsymbol{Y}^\top\boldsymbol{Y})^{1/2}\boldsymbol{W}_y$. Then, (44) can be reformulated as:

$$\begin{aligned} \underset{\boldsymbol{H}_x, \boldsymbol{H}_y}{\text{maximize}} \quad & \operatorname{Tr}\left[\boldsymbol{H}_x^\top(\boldsymbol{X}^\top\boldsymbol{X})^{-1/2}\boldsymbol{X}^\top\boldsymbol{Y}(\boldsymbol{Y}^\top\boldsymbol{Y})^{-1/2}\boldsymbol{H}_y\right] \\ \text{subject to} \quad & \boldsymbol{H}_x^\top\boldsymbol{H}_x = \boldsymbol{H}_y^\top\boldsymbol{H}_y = \boldsymbol{I}. \end{aligned} \tag{45}$$

By this change of variables, we simplified the constraints of the problem so that $\boldsymbol{H}_x$ and $\boldsymbol{H}_y$ are constrained to be orthogonal matrices. By applying the cyclic property of the trace operator, and defining $\boldsymbol{Q} = \boldsymbol{H}_y\boldsymbol{H}_x^\top$, $\boldsymbol{X}^\phi = \boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1/2}$, $\boldsymbol{Y}^\phi = \boldsymbol{Y}(\boldsymbol{Y}^\top\boldsymbol{Y})^{-1/2}$, we can simplify the problem further:

$$\underset{\boldsymbol{Q} \in \mathcal{O}}{\text{maximize}} \quad \operatorname{Tr}[(\boldsymbol{X}^\phi)^\top\boldsymbol{Y}^\phi\boldsymbol{Q}]. \tag{46}$$

Thus, we see that CCA is equivalent to solving the Procrustes problem on $\boldsymbol{X}^\phi$ and $\boldsymbol{Y}^\phi$. Note that $(\boldsymbol{X}^\phi)^\top \boldsymbol{X}^\phi = (\boldsymbol{Y}^\phi)^\top \boldsymbol{Y}^\phi = \boldsymbol{I}$, and so this change of variables can be interpreted as a whitening operation [31].

From Lemma 1, we see that the optimal objective value to (46) is given by the sum of the singular values of $(\boldsymbol{X}^\phi)^\top \boldsymbol{Y}^\phi$. These singular values, which we denote here as $1 \geq \sigma_1 \geq ... \geq \sigma_n \geq 0$, are called *canonical correlation coefficients*. They are bounded above by one since the singular values of $\boldsymbol{X}^\phi$ and $\boldsymbol{Y}^\phi$ are all equal to one, due to the whitening step, and the operator norm[2] is sub-multiplicative:

$$\|(\boldsymbol{X}^\phi)^\top \boldsymbol{Y}^\phi\|_{\text{op}} \leq \|\boldsymbol{X}^\phi\|_{\text{op}} \|\boldsymbol{Y}^\phi\|_{\text{op}} = 1\,. \tag{47}$$

Putting these pieces together, we see:

$$\min_{\boldsymbol{Q} \in \mathcal{O}} \ \arccos \frac{\langle \boldsymbol{X}^\phi, \boldsymbol{Y}^\phi \boldsymbol{Q} \rangle}{\|\boldsymbol{X}^\phi\| \|\boldsymbol{Y}^\phi\|} = \arccos \ \frac{\|(\boldsymbol{X}^\phi)^\top \boldsymbol{Y}^\phi\|_*}{\sqrt{n} \cdot \sqrt{n}} = \arccos \left(\tfrac{1}{n} \sum_{i=1}^n \sigma_i \right) \tag{48}$$

which coincides with equation 10 in the main text, since $\sigma_i = \rho_i / n$ for the case of CCA. Proposition 2 implies that this defines a metric since $\boldsymbol{X}^\phi / \|\boldsymbol{X}^\phi\|$ and $\boldsymbol{Y}^\phi / \|\boldsymbol{Y}^\phi\|$ are matrices with unit Frobenius norm, and because the set of orthogonal transformations is a group of isometries, as established in Section A.5.

C.4 Ridge CCA

Next, we consider metrics based on regularized CCA, which essentially interpolate between the orthogonally invariant metrics discussed in Section C.2, and the linearly invariant metrics discussed in Section C.3. This interpolation is accomplished by specifying a hyperparameter $0 \leq \alpha \leq 1$, where $\alpha = 0$ corresponds to unregularized CCA and $\alpha = 1$ corresponds to Procrustes alignment (i.e. fully regularized). We formulate this family of optimization problems as:

$$\begin{aligned} \underset{\boldsymbol{W}_x, \boldsymbol{W}_y}{\text{maximize}} \quad & \text{Tr}[\boldsymbol{W}_x^\top \boldsymbol{X}^\top \boldsymbol{Y} \boldsymbol{W}_y] \\ \text{subject to} \quad & \boldsymbol{W}_x^\top ((1-\alpha)\boldsymbol{X}^\top \boldsymbol{X} + \alpha \boldsymbol{I}) \boldsymbol{W}_x = \boldsymbol{W}_y^\top ((1-\alpha)\boldsymbol{Y}^\top \boldsymbol{Y} + \alpha \boldsymbol{I}) \boldsymbol{W}_y = \boldsymbol{I}\,. \end{aligned} \tag{49}$$

Notice that when $\alpha = 1$, the constraints reduce to $\boldsymbol{W}_x$ and $\boldsymbol{W}_y$ being orthogonal, and thus the objective function can be viewed as maximizing $\langle \boldsymbol{X}, \boldsymbol{Y}\boldsymbol{Q} \rangle$ over orthogonal matrices $\boldsymbol{Q} = \boldsymbol{W}_y \boldsymbol{W}_x^\top$. Thus, we recover Procrustes alignment in the limit of $\alpha = 1$. Clearly, when $\alpha = 0$, Eq. (49) reduces to the usual formulation of CCA (see Eq. (44)).

We can solve Eq. (49) by following essentially the same procedure outlined in Section C.3, in which we reduce the problem to Procrustes alignment by a change of variables. In this case, the change of variables corresponds to a partial whitening transformation:

$$\boldsymbol{H}_x = ((1-\alpha)(\boldsymbol{X}^\top \boldsymbol{X}) + \alpha \boldsymbol{I}))^{1/2} \quad \text{and} \quad \boldsymbol{H}_y = ((1-\alpha)(\boldsymbol{Y}^\top \boldsymbol{Y}) + \alpha \boldsymbol{I}))^{1/2}\,. \tag{50}$$

Then, reformulate the optimization problem as:

$$\begin{aligned} \underset{\boldsymbol{H}_x, \boldsymbol{H}_y}{\text{maximize}} \quad & \text{Tr}[\boldsymbol{H}_x^\top ((1-\alpha)(\boldsymbol{X}^\top \boldsymbol{X}) + \alpha \boldsymbol{I}))^{-1/2} \boldsymbol{X}^\top \boldsymbol{Y} ((1-\alpha)(\boldsymbol{Y}^\top \boldsymbol{Y}) + \alpha \boldsymbol{I}))^{-1/2} \boldsymbol{H}_y] \\ \text{subject to} \quad & \boldsymbol{H}_x^\top \boldsymbol{H}_x = \boldsymbol{H}_y^\top \boldsymbol{H}_y = \boldsymbol{I}\,. \end{aligned} \tag{51}$$

Let $\boldsymbol{Q} = \boldsymbol{H}_y \boldsymbol{H}_x$, and let

$$\boldsymbol{X}^\phi = \boldsymbol{X}((1-\alpha)(\boldsymbol{X}^\top \boldsymbol{X}) + \alpha \boldsymbol{I}))^{-1/2} \quad \text{and} \quad \boldsymbol{Y}^\phi = \boldsymbol{Y}((1-\alpha)(\boldsymbol{Y}^\top \boldsymbol{Y}) + \alpha \boldsymbol{I}))^{-1/2}\,. \tag{52}$$

Then, by Proposition 2 and Lemma 1, we have the following metric:

$$\min_{\boldsymbol{Q} \in \mathcal{O}} \ \arccos \frac{\langle \boldsymbol{X}^\phi, \boldsymbol{Y}^\phi \boldsymbol{Q} \rangle}{\|\boldsymbol{X}^\phi\| \|\boldsymbol{Y}^\phi\|} = \arccos \ \left\| \left(\frac{\boldsymbol{X}^\phi}{\|\boldsymbol{X}^\phi\|} \right)^\top \left(\frac{\boldsymbol{Y}^\phi}{\|\boldsymbol{Y}^\phi\|} \right) \right\|_* = \arccos \left(\sum_{i=1}^n \rho_i \right). \tag{53}$$

C.5 Nonlinear Alignments and Kernel CCA

We can also consider metrics based on *kernel CCA* [59], which generalizes CCA to account for nonlinear alignments. As its name suggests, this approach belongs to a more general class of *kernel methods* that operate implicitly in high-dimensional (even infinite-dimensional) feature spaces through inner product evaluations. For a broader review of kernel methods in machine learning, see [60].

First, we recall the inner product between two matrices in a finite dimensional feature space $\mathbb{R}^{m \times p}$:

$$\langle \boldsymbol{X}^\phi, \boldsymbol{Y}^\phi \rangle = \text{Tr}[(\boldsymbol{X}^\phi)^\top \boldsymbol{Y}^\phi] = \sum_{i=1}^m (\boldsymbol{x}_i^\phi)^\top (\boldsymbol{y}_i^\phi)\,. \tag{54}$$

[2]The operator norm of a matrix $\boldsymbol{M}$, denoted $\|\boldsymbol{M}\|_{\text{op}}$, is equal to the largest singular value of $\boldsymbol{M}$.

Here we have introduced notation $\boldsymbol{x}_i^\phi$ and $\boldsymbol{y}_i^\phi$ to denote the p-dimensional vectors holding features to the i^{th} network input. In kernel CCA, we consider more general feature mappings $\boldsymbol{x}_i \mapsto x_i^\phi$ and $\boldsymbol{y}_i \mapsto y_i^\phi$, where each $x_i^\phi \in \mathcal{H}$ and $y_i^\phi \in \mathcal{H}$ are vectors in some Reproducing Kernel Hilbert Space (RKHS). That is, instead of having two matrices $\boldsymbol{X}^\phi$ and $\boldsymbol{Y}^\phi$ to represent the network representations in the feature space, we instead consider the collections of vectors: $X^\phi = \{x_1^\phi, \ldots, x_m^\phi\}$ and $Y^\phi = \{y_1^\phi, \ldots, y_m^\phi\}$.

Given a choice of a positive-definite kernel function k, we begin by computing two $m \times m$ un-centered kernel matrices:

$$[\widetilde{\boldsymbol{K}}_x]_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle x_i^\phi, x_j^\phi \rangle \quad \text{and} \quad [\widetilde{\boldsymbol{K}}_y]_{ij} = k(\boldsymbol{y}_i, \boldsymbol{y}_j) = \langle y_i^\phi, y_j^\phi \rangle \tag{55}$$

for $i, j \in \{1, \ldots, m\}$. Then, we define the centered kernel matrices: $\boldsymbol{K}_x = \boldsymbol{C}\widetilde{\boldsymbol{K}}_x\boldsymbol{C}$ and $\boldsymbol{K}_y = \boldsymbol{C}\widetilde{\boldsymbol{K}}_y\boldsymbol{C}$, where $\boldsymbol{C} = \boldsymbol{I} - \frac{1}{m}\boldsymbol{1}\boldsymbol{1}^\top$ is the centering matrix.

The classic form of CCA (44) can then be reformulated terms of purely kernel operations [57, 59]:

$$\begin{aligned} \underset{\boldsymbol{W}_x, \boldsymbol{W}_y}{\text{maximize}} \quad & \operatorname{Tr}\left[\boldsymbol{W}_x^\top \boldsymbol{K}_x \boldsymbol{K}_y \boldsymbol{W}_y\right] \\ \text{subject to} \quad & \boldsymbol{W}_x^\top \boldsymbol{K}_x^2 \boldsymbol{W}_x = \boldsymbol{W}_y^\top \boldsymbol{K}_y^2 \boldsymbol{W}_y = \boldsymbol{I}\,. \end{aligned} \tag{56}$$

One can show that this optimization problem is equivalent (up to a change of variables) from the classic CCA problem when a linear kernel function, $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \boldsymbol{x}_j$, is used. Furthermore, one can generalize the regularization scheme for CCA (see Section C.4),

$$\begin{aligned} \underset{\boldsymbol{W}_x, \boldsymbol{W}_y}{\text{maximize}} \quad & \operatorname{Tr}\left[\boldsymbol{W}_x^\top \boldsymbol{K}_x \boldsymbol{K}_y \boldsymbol{W}_y\right] \\ \text{subject to} \quad & \boldsymbol{W}_x^\top((1-\alpha)\boldsymbol{K}_x^2 + \alpha\boldsymbol{K}_x)\boldsymbol{W}_x = \boldsymbol{W}_y^\top((1-\alpha)\boldsymbol{K}_y^2 + \alpha\boldsymbol{K}_y)\boldsymbol{W}_y = \boldsymbol{I}\,. \end{aligned} \tag{57}$$

C.6 Geodesic Distances on Kendall's Shape Space

We now consider a modification of the Procrustes alignment problem, where we optimize over the special orthogonal group (i.e. the set of orthogonal matrices with $\det(\boldsymbol{Q}) = +1$)

$$\boldsymbol{R}^* = \underset{\boldsymbol{R} \in \mathcal{SO}}{\operatorname{argmin}} \ \|\boldsymbol{X} - \boldsymbol{Y}\boldsymbol{R}\| = \underset{\boldsymbol{R} \in \mathcal{SO}}{\operatorname{argmax}} \ \operatorname{Tr}[\boldsymbol{X}^\top \boldsymbol{Y}\boldsymbol{R}]\,. \tag{58}$$

We can obtain the solution by a minor modification of Lemma 1. We let $\boldsymbol{X}^\top\boldsymbol{Y} = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{S}}\tilde{\boldsymbol{V}}^\top$ denote the "optimally signed" singular value decomposition of $\boldsymbol{X}^\top\boldsymbol{Y}$ in which $\tilde{\boldsymbol{U}} \in \mathcal{SO}$, $\tilde{\boldsymbol{V}} \in \mathcal{SO}$, and $\tilde{\boldsymbol{S}}$ is a diagonal matrix of signed singular values: $\tilde{\sigma}_1 \geq \ldots \geq \tilde{\sigma}_{n-1} \geq |\tilde{\sigma}_n| \geq 0$. Thus, all optimally signed singular values are positive except if $\det(\boldsymbol{X}^\top\boldsymbol{Y}) < 0$, in which case the final singular value is negated, $\tilde{\sigma}_n = -\sigma_n$, so that $\det(\tilde{\boldsymbol{U}}) = \det(\tilde{\boldsymbol{V}}) = +1$. Then the optimal rotation is given by $\boldsymbol{R}^* = \tilde{\boldsymbol{V}}\tilde{\boldsymbol{U}}^\top$. See Le [61] for a proof.

We refer the reader to Chapters 4 and 5 of Dryden et al. [15] for further details. When $\mathcal{G} = \mathcal{SO}$, our Proposition 1 corresponds to Riemannian distance in size-and-shape space (sec. 5.3, [15]). Likewise, $\mathcal{G} = \mathcal{SO}$, our Proposition 2 corresponds to Riemannian distance Kendall's shape space (sec. 4.1.4, [15]).

C.7 Centered Kernel Alignment (CKA) and Representational Similarity Analysis (RSA)

Linear CKA [9] and RSA [8] are two closely related methods that, in essence, evaluate the similarity between $\boldsymbol{X}\boldsymbol{X}^\top$ and $\boldsymbol{Y}\boldsymbol{Y}^\top$ to capture the similarity of neural representations. When the data are mean-centered as a preprocessing step, these are $m \times m$ covariance matrices capturing the correlations in neural activations over the m test images. Several variants of RSA exist. For example, one can compute the pairwise Euclidean distances between all m hidden layer activation patterns, resulting in representational distance matrices (RDMs) instead of the covariance matrices mentioned above. Likewise, nonlinear extensions of CKA use nonlinear kernel functions to compute centered kernel matrices $\boldsymbol{K}_x$ and $\boldsymbol{K}_y$, as defined above in Section C.5. When a linear kernel function is used (i.e. in linear CKA), the centered kernel matrices reduce to the usual covariance matrices $\boldsymbol{K}_x = \boldsymbol{X}\boldsymbol{X}^\top$ and $\boldsymbol{K}_y = \boldsymbol{Y}\boldsymbol{Y}^\top$.

In essence, these methods proceed by computing the similarity between $\boldsymbol{K}_x$ and $\boldsymbol{K}_y$. Kriegeskorte et al. [8] proposed taking the Spearman correlation between the upper-triangular entries of these matrices. This measure of similarity does not produce a metric, as we verified empirically in the main text. Kornblith et al. [9] proposed to use the following quantity (assuming centered kernels):

$$\text{CKA}(\boldsymbol{K}_x, \boldsymbol{K}_y) = \frac{\operatorname{Tr}[\boldsymbol{K}_x\boldsymbol{K}_y]}{\sqrt{\operatorname{Tr}[\boldsymbol{K}_x^2] \cdot \operatorname{Tr}[\boldsymbol{K}_y^2]}} \tag{59}$$

which is known as centered kernel alignment (originally defined in [24, 62]).

While CKA as originally formulated does not produce a metric, we can modify it to satisfy the requirements of a metric space. First, note that:

$$\text{CKA}(\boldsymbol{K}_x, \boldsymbol{K}_y) = \cos\left[d_\theta(\boldsymbol{K}_x, \boldsymbol{K}_y)\right] \tag{60}$$

where d_θ is the angular distance (see Eq. (17)) over $\mathbb{R}^{m\times m}$ matrices. Thus, one can apply $\arccos(\cdot)$ to CKA achieve a proper metric. For example, a metric based on linear CKA can be calculated as follows:

$$d_\theta(\boldsymbol{X}\boldsymbol{X}^\top, \boldsymbol{Y}\boldsymbol{Y}^\top) = \arccos\left[\frac{\|\boldsymbol{X}^\top\boldsymbol{Y}\|^2}{\|\boldsymbol{X}\boldsymbol{X}^\top\|\|\boldsymbol{Y}\boldsymbol{Y}^\top\|}\right] \tag{61}$$

where, as before, all norms denote the Frobenius matrix norm. Note that this calculation bears some similarity to the fully regularized CCA distance:

$$\theta_1(\boldsymbol{X}, \boldsymbol{Y}) = \min_{\boldsymbol{Q}\in\mathcal{O}} \arccos\left[\frac{\langle \boldsymbol{X}, \boldsymbol{Y}\boldsymbol{Q}\rangle}{\|\boldsymbol{X}\| \cdot \|\boldsymbol{Y}\|}\right] = \arccos\left[\frac{\|\boldsymbol{X}^\top\boldsymbol{Y}\|_*}{\|\boldsymbol{X}\|\|\boldsymbol{Y}\|}\right] \tag{62}$$

The two differences between these metrics are that (a) CKA uses the squared Frobenius norm instead of the nuclear norm to measure the scale of $\boldsymbol{X}^\top\boldsymbol{Y}$ in the numerator, and (b) CKA normalizes by the norms of the covariances, $\boldsymbol{X}\boldsymbol{X}^\top$ and $\boldsymbol{Y}\boldsymbol{Y}^\top$, rather than the norms of the matrices themselves.

While this manuscript was undergoing review, Shahbazi et al. [25] published a different modification of CKA and RSA to satisfy the properties of a metric space. They advocate using the Riemannian metric over positive-definite matrices:

$$d(\boldsymbol{K}_x, \boldsymbol{K}_y) = \sqrt{\sum_{i=1}^{m} \log^2(\lambda_i)}\,, \tag{63}$$

where $\lambda_1, \ldots, \lambda_m$ are the eigenvalues of $\boldsymbol{K}_x^{-1}\boldsymbol{K}_y$. This calculation is appealing because it exploits the fact that $\boldsymbol{K}_x$ and $\boldsymbol{K}_y$ are positive-definite matrices by construction. The extension of CKA discussed above utilizes the generic angular distance between $m \times m$ matrices, which are not necessarily positive-definite.

D Probabilistic interpretations of generalized shape metrics

To extend generalized shape metrics to stochastic neural representations, we must introduce some additional notation and formalize network representations as random variables (rather than $m \times n$ matrices). We can model neural representations as independent random variables when conditioned on the input. That is, let X and Y denote random variables on $\mathbb{R}^n$, which correspond to n-dimensional neural responses to a stochastic input.[3] Further, let Z be some random variable corresponding to process of sampling an input to the network (e.g. choosing one of m input images at random). Then, the joint distribution over representations and inputs decomposes as $P(X, Y, Z) = P(X \mid Z)P(Y \mid Z)P(Z)$ for any pair of networks X and Y.

The goal of this section is to define functions $d(X, Y)$ that are metrics over the set of random variables with outcomes on $\mathbb{R}^n$, and which are natural extensions of Proposition 1 and 2 in the main text. The key step towards achieving this goal is to establish a Hilbert space for random vectors. We provide a short and informal demonstration of this below, but refer the reader to Chapter 2 of Tsiatis [63] for a more complete treatment.

First, we establish that the set of random vectors is a vector space. The zero vector corresponds to a random vector that is equal to the zero vector on $\mathbb{R}^n$ almost surely. Vector addition $X + Y$ creates a new random vector from two inputs X and Y. Intuitively, we can draw samples from $X + Y$ by first sampling X and Y and then adding their outcomes. Scalar multiplication αX creates a new random vector given the input X and a scalar $\alpha \in \mathbb{R}$. Intuitively, we can sample αX by first drawing a sample from X and multiplying this outcome by α. We can then define the inner product between two random vectors in the following lemma.

Lemma. *Let X and Y be random vectors associated with some joint probability density function $p(\boldsymbol{x}, \boldsymbol{y})$ for all $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^n$. Then,*

$$\langle X, Y\rangle = \mathbb{E}[\boldsymbol{x}^\top\boldsymbol{y}]\,, \tag{64}$$

is an inner product over the set of random vectors, where the expectation is taken over joint samples of X and Y.

Proof. Using the linearity of expectation and the inner product on $\mathbb{R}^n$, it is easy to prove that the inner product is symmetric,

$$\langle X, Y\rangle = \mathbb{E}[\boldsymbol{x}^\top\boldsymbol{y}] = \mathbb{E}[\boldsymbol{y}^\top\boldsymbol{x}] = \langle Y, X\rangle\,, \tag{65}$$

and linear,

$$\langle M + \alpha X, Y\rangle = \mathbb{E}[(\boldsymbol{z} + \alpha\boldsymbol{x})^\top\boldsymbol{y}] = \mathbb{E}[\boldsymbol{z}^\top\boldsymbol{y}] + \alpha\mathbb{E}[\boldsymbol{x}^\top\boldsymbol{y}] = \langle M, Y\rangle + \alpha\langle X, Y\rangle \tag{66}$$

[3] As in the main text, we can define feature maps $X \mapsto X^\phi$ and $Y \mapsto Y^\phi$ which establish a common dimensionality between networks of dissimilar sizes.

for any random vector M and $\alpha \in \mathbb{R}$. All that remains is to prove is that $\langle \cdot, \cdot \rangle$ is positive definite, we first note that the mapping $\boldsymbol{x} \mapsto \boldsymbol{x}^\top \boldsymbol{x}$ is a convex function of $\boldsymbol{x}$. Then, we apply Jensen's inequality and the positive definiteness of the inner product on $\mathbb{R}^n$ to show:

$$\langle X, X \rangle = \mathbb{E}[\boldsymbol{x}^\top \boldsymbol{x}] \geq (\mathbb{E}\boldsymbol{x})^\top (\mathbb{E}\boldsymbol{x}) \geq 0\,. \tag{67}$$

Further $\mathbb{E}[\boldsymbol{x}^\top \boldsymbol{x}] = 0$ only when $\boldsymbol{x} = \mathbf{0}$, almost surely. Thus, $\langle X, X \rangle = 0$ if and only if $X = 0$. □

To begin, we consider a special case where the neural responses are deterministic, but the inputs are randomly chosen. That is, to draw a sample of (X, Y), we first sample an input $\boldsymbol{z} \sim P(Z)$ and then calculate $\boldsymbol{x} = f_x(\boldsymbol{z})$ and $\boldsymbol{y} = f_y(\boldsymbol{z})$, where f_x and f_y are functions mapping the input space to $\mathbb{R}^n$.

In the simplest case, $P(Z)$ is a uniform distribution over a discrete set of m network inputs. In this case, we can compute the required inner products exactly. Let $\boldsymbol{z}_i$ denote the i^{th} input to the networks, and let $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{Y} \in \mathbb{R}^{m \times n}$ denote matrices that stack the neural responses, $f_x(\boldsymbol{z}_i)$ and $f_y(\boldsymbol{z}_i)$ row-wise. Then we have

$$\langle X, Y \rangle = \mathbb{E}[\boldsymbol{x}^\top \boldsymbol{y}] = \frac{1}{m} \sum_{i=1}^{m} f_x(\boldsymbol{z}_i)^\top f_y(\boldsymbol{z}_i) = \frac{1}{m} \langle \boldsymbol{X}, \boldsymbol{Y} \rangle\,, \tag{68}$$

where the final inner product $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = \mathrm{Tr}[\boldsymbol{X}^\top \boldsymbol{Y}]$ is the typical Frobenius inner product between matrices that we have used throughout. Because these inner products coincide up to a uniform scaling factor, we can reinterpret the metrics defined in the main text (Propositions 1 & 2) as providing a notion of distance between deterministic neural responses that are drawn uniformly from a set of m inputs.

In many cases, the number of possible inputs to a network is effectively infinite, so we can consider $P(Z)$ to be a continuous distribution. In this scenario, the inner product becomes:

$$\langle X, Y \rangle = \int p(\boldsymbol{z}) f_x(\boldsymbol{z})^\top f_y(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z} \tag{69}$$

which is generally intractable to compute. For example, we typically do not know how to evaluate the density $p(\boldsymbol{z})$. This is the case, for example, when $P(Z)$ corresponds to the distribution over all "natural images." If we are given independent samples $\boldsymbol{z}_i \sim P(Z)$, for $i = 1, \ldots, m$, then the integral can be approximated as

$$\int p(\boldsymbol{z}) f_x(\boldsymbol{z})^\top f_y(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z} \approx \frac{1}{m} \sum_{i=1}^{m} f_x(\boldsymbol{z}_i)^\top f_y(\boldsymbol{z}_i) = \frac{1}{m} \langle \boldsymbol{X}, \boldsymbol{Y} \rangle\,, \tag{70}$$

which coincides with (68). Thus, we can also interpret generalized shape metrics (Propositions 1 & 2) as being approximations to metrics that capture representational dissimilarity over a continuous distribution of input patterns. This final interpretation is appealing from both scientific and engineering perspectives. In neuroscience, we expect animals to encounter sensory input patterns probabilistically from an effectively infinite range of possibilities. Likewise, in machine learning, we are interested in how deep artificial networks generalize to "real-world" applications. In short, the space of possible future inputs is generally more numerous than the space of inputs used for training and validation. Nonetheless, if the statistics of the test set match the "real world," then (70) tells us that we can approximate the "true" distance between network representations appropriately.

The results shown in Figures 3B and 3C in the main text can now be properly interpreted as varying the choice of m (sample size) in the approximation of the integral appearing in equation (70).

The framework above can also be readily extended to define metrics between stochastic neural representations, which are ubiquitous in both biology (due to "noise") and machine learning (e.g. dropout layers). We view this as an intriguing direction for future research that is enabled by our theoretical framing of neural representations.

E Experimental Methods

Code accompanying this paper can be found at — `https://github.com/ahwillia/netrep`

E.1 Experiments on sample size (Fig. 3)

We ran all experiments on a pair of convolutional neural networks trained on CIFAR-10. The architecture is shown in Table 1. In Figure 3A, we sampled activations from the three layers following the stride-2 convolutions. We did a brute-force search over circular shifts along the width and height dimensions. When comparing two layers with unequal dimensions, we upsampled the layer with smaller width and height by linear interpolation. The remaining panels in Figure 3 were computed using activations from the final layer before average pooling.

3×3 conv. 64-BN-ReLU
3×3 conv. 64-BN-ReLU
3×3 conv. 64-BN-ReLU
3×3 conv. 64 stride 2-BN-ReLU
3×3 conv. 128-BN-ReLU
3×3 conv. 128-BN-ReLU
3×3 conv. 128-BN-ReLU
3×3 conv. 128 stride 2-BN-ReLU
3×3 conv. 256-BN-ReLU
3×3 conv. 256-BN-ReLU
3×3 conv. 256-BN-ReLU
3×3 conv. 256 stride 2-BN-ReLU
Global average pooling
Logits

Table 1: The architecture used for experiments in Fig. 3. All convolutions use zero padding to maintain the size of the feature map.

E.2 Allen Brain Observatory

Data were accessed through the Allen Software Development Kit (AllenSDK — `https://allensdk.readthedocs.io/en/latest/`). All isolated single units that met the default quality control standards were loaded and pooled across sessions. The anatomical location of each unit in Common Coordinate Framework (CCF; [64]) was extracted and categorized into anatomical regions according to the reference atlas, using the finest scale anatomical parcellation. Spike counts were calculated over `0.033355` ms timebins (duration of a single movie frame), over 1600 frames. Spikes were then smoothed with a Gaussian filter with a standard deviation of 20 bins (frames), and averaged over 10 trials (repeats of the movie). Then, we projected the data onto the top 100 principal components, resulting in a matrix $\boldsymbol{X}_k \in \mathbb{R}^{1600 \times 100}$ for each brain region $k = \{1, \ldots, K\}$. Regions with fewer than 100 neurons across all sessions were excluded. The following set of 48 regions, listed by their standard abbreviations, contained more than 100 neurons and were then studied for further analysis: `APN, AUDd5, AUDpo5, AUDpo6a, CA1, CA3, DG-mo, DG-sg, Eth, LGd-co, LGd-ip, LGd-sh, LGv, LP, MB, MGd, MGv, PO, POL, ProS, SGN, SSp-bfd2/3, SSp-bfd4, SSp-bfd5, SUB, TEa5, TH, VISa2/3, VISa4, VISa5, VISa6a, VISal2/3, VISal4, VISal5, VISam2/3, VISam4, VISam5, VISam6a, VISp2/3, VISp4, VISp5, VISp6a, VPM, alv, ccs, dhc, fp, or`.

Dendrograms were computed and visualized using tools available in the scipy library [58]. We used Ward's linkage criterion to compute the hierarchical clusterings.

We performed kernel ridge regression to predict anatomical hierarchy scores (defined in [48]) 29 regions: `AUDd5, AUDpo5, AUDpo6a, LGd-co, LGd-ip, LGd-sh, LP, MGd, MGv, POL, SSp-bfd2/3, SSp-bfd4, SSp-bfd5, TEa5, VISa2/3, VISa4, VISa5, VISa6a, VISal2/3, VISal4, VISal5, VISam2/3, VISam4, VISam5, VISam6a, VISp2/3, VISp4, VISp5, VISp6a`. Two regions, `PO` and `VPM`, were excluded from the analysis as they were outliers with exceptionally high and low hierarchy scores. The other regions were excluded because they either had undefined hierarchy scores or had fewer than 100 neurons. We used the scikit-learn implementation of kernel ridge regression, `KernelRidge(alpha=0.01, gamma=1.0, kernel="rbf")`, and fit the model 100 separate times on different approximate Euclidean embeddings found by multi-dimensional scaling (MDS). The error bars in Fig. 5B show range of estimates from different MDS embeddings. An embedding dimension of $L = 20$ was used in all cases.

If our goal is only to visualize the data in 2D we may apply MDS with an embedding dimension of $L = 2$. How does this embedding differ from a larger embedding of $L = 20$? Figure E.2.1 demonstrates that qualitatively distinct structures emerge from these two procedures.

E.3 NAS-Bench-101

We obtained checkpoints for 2000 randomly-selected NAS-Bench-101 architectures trained for 108 epochs following the protocol described in [22] and computed the similarity between activations of every possible pair of these architectures on the CIFAR-10 test set, using an Apache Beam pipeline operating on offsite hardware. In total, the computational cost of these experiments was 260 core-years, including pilot experiments and several experiments not included in the paper.

For ridge regression analyses in Fig. 5D, we train on 80% of the data, use 10% of the data as a validation set to select the optimal ridge hyperparameter and the kernel bandwidth, and compute R^2 on remaining 10% of the data.

In Figure E.3.1, we show the skeleton of the NAS-Bench-101 architecture along with the layers from which we extract representations.

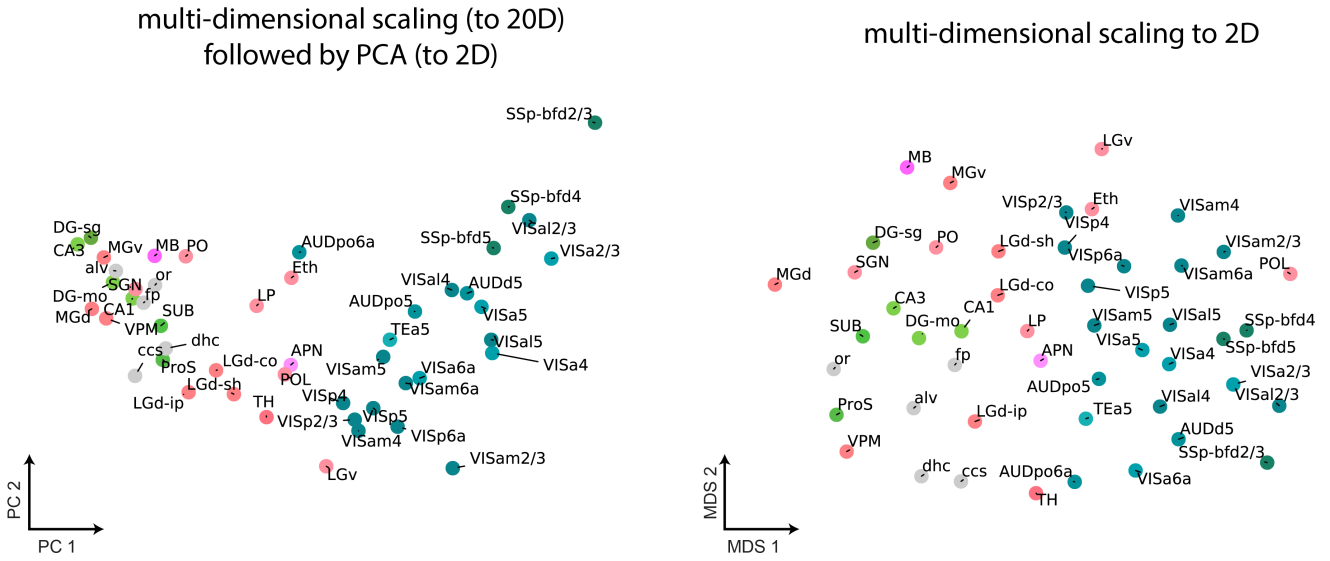

Figure E.2.1: Performing MDS directly to $L = 2$ dimensions (right) produces a distinct low-dimensional visualization of the ABO dataset from multi-dimensional scaling to $L = 20$ dimensions, following by PCA projection down to 2D (left). As shown in the main text (Fig. 4C), the MDS embedding to $L = 20$ dimensions produces a dramatically better approximation of the true metric space than the embedding to $L = 2$ dimensions. Thus, we advocate using the former over the latter for downstream modeling tasks.

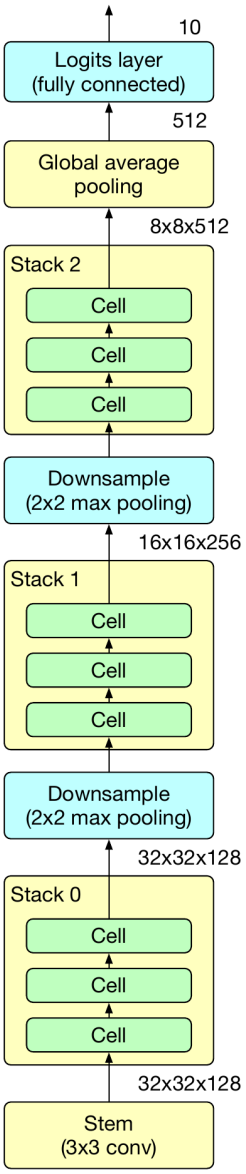

Figure E.3.1: Diagram of the skeleton of the NAS-Bench-101 architecture. The architecture of each cell (shown in green) is selected from a fixed space, described further by Ying et al. [22], and all cells within a single architecture are identical except for the number of channels, which differs by stack. In Fig. 5, we show the results we obtain by analyzing the representations of the outputs of the layers shown in yellow.