Autoencoder networks extract latent variables and encode these variables in their connectomes

Matthew Farrell[1,2] **Stefano Recanatesi**[2] **R. Clay Reid**[3]

Stefan Mihalas[3,†] **Eric Shea-Brown**[1,2,†]

[1]Department of Applied Mathematics, University of Washington; Seattle, WA
[2]Center for Computational Neuroscience and Swartz Center for Theroetical Neuroscience, University of Washington; Seattle, WA
[3]Allen Institute for Brain Science; Seattle, WA
† These authors share senior authorship

Abstract

Spectacular advances in imaging and data processing techniques are revealing a wealth of information about brain connectomes. This raises an exciting scientific opportunity: to infer the underlying circuit function from the structure of its connectivity. A potential roadblock, however, is that – even with well constrained neural dynamics – there are in principle many different connectomes that could support a given computation. Here, we define a tractable setting in which the problem of inferring circuit function from circuit connectivity can be analyzed in detail: the function of input compression and reconstruction, in an autoencoder network with a single hidden layer. Here, in general there is substantial ambiguity in the weights that can produce the same circuit function, because largely arbitrary changes to "input" weights can be undone by applying the inverse modifications to the "output" weights. However, we use mathematical arguments and simulations to show that adding simple, biologically motivated regularization of connectivity resolves this ambiguity in an interesting way: weights are constrained such that the latent variable structure underlying the inputs can be extracted from the weights by using nonlinear dimensionality reduction methods.

1 Introduction

The past years have seen spectacular effort, and spectacular success, in mapping synaptic connections in the brain. For example, the hemi-brain connectome of *Drosophila* was recently released, and imaging of the whole brain connectome is currently underway [1]. The synaptic connections in a cubic millimeter of mouse visual cortex have also recently been imaged [2]. These data build on pioneering efforts to map the connectome of *c. elegans*. From these advances there have emerged stunning new opportunities and new challenges for modelers and theoreticians.

Connectivity data has long been leveraged to shed light on circuit function. This includes the discovery of hierarchical organization in mammalian visual systems [3], which is thought to support the assembly of complex and abstract visual information in higher areas out of combinations of simple, local neural responses in lower areas; and the repeated structure across different neocortical regions [4, 5], indicating that neocortex supports general-purpose learning. As connectivity data become more complete, they can be used to more precisely constrain models [6]. For instance, while a variety of mechanisms have been proposed to explain motion processing in the retina, recent connectivity data allowed the authors of [7] to refine this class of models into one that better fits the observed connectivity. Additional studies link connectivity to the function of sensory circuits through the lens of increasing or decreasing the dimension of their inputs. For example, connections between mossy-fiber and granule cells in the *Drosophila* mushroom body appear to be random and sparse [8], which is thought to support associative learning by expanding dimension [9–11]. Other studies point out physical bottlenecks in sensory circuits that strongly suggest a compression of dimension, particularly in early visual pathways (for a review, see [12]). Such a compression forces circuits to select elements of their inputs which are necessary for downstream computations. Often this operation is modeled as extracting a low-dimensional set of *latent variables* that generate the higher-dimensional input signal.

Our work follows on these observations by probing the following question: in compressive circuits, can the actual structure of the selected-for latent variables be extracted from the connectome? While this question will prove to be a significant challenge to answer in general, here we start with a simple and mathematically tractable model of input compression: a linear autoencoder with a single layer of hidden units. This work sets out to discover if the weights that optimally compress inputs (in an L2 reconstruction sense) contain recoverable information about the input stimulus; namely, the latent variables generating the inputs. Here we focus on using dimensionality reduction methods on the weight matrices to recover this information.

Our main finding is that structure from the latent variables underlying the inputs *can* be extracted from the weights of our network, provided that the model is regularized by biologically inspired costs on weight resources (i.e. penalizing large weights). The structure that can be extracted includes the basic topology of the latent space; in particular, it can be inferred if the latent variables live on a space that wraps around (like a circle) or that doesn't wrap around (like a line). We give both mathematical arguments in the case of linear autoencoders and the simulation results of training autoencoders with hyperbolic tangent nonlinearities. Our results have several important implications. First, they are an important proof of concept that meaningful information about network

function can be extracted from the optimal weights alone. Second, they shed light on *how* the weights can be processed to reveal this latent structure. Specifically, we show that nonlinear (as opposed to linear) dimensionality reduction techniques are important for finding this structure. This can guide efforts to find structure in the weights of more complex models or biological data. Third, we establish that combinations of latent variables are reflected in the weight structure in a predictable way. This can be used as a tool for looking at complicated inputs that may be formed from many latent variables. These results reinforce an emerging narrative in the analysis of neural networks: that regularization is important not only for producing network models that generalize, but also for producing models that are interpretable (see [13, 14]).

2 Network Architecture

The model we consider is an autoencoder network with a single layer of hidden units, as illustrated in Fig. 1. The equation for the hidden unit activations in response to an input $\boldsymbol{x}_s$ is

$$\boldsymbol{h}_s = \phi(\boldsymbol{W}_{\text{in}}\boldsymbol{x}_s) + \boldsymbol{b}_1 \in \mathbb{R}^N$$

where N is the number of hidden units and ϕ is the activation function for the hidden units. The length N vector $\boldsymbol{b}_1$ is a bias term. For our mathematical analysis we take ϕ equal to the identity, but we also show the results of simulations where $\phi = \tanh$. The activation of the output units is

$$\boldsymbol{y}_s = \boldsymbol{W}_{\text{out}}\boldsymbol{h}_s + \boldsymbol{b}_2 \in \mathbb{R}^m. \tag{1}$$

where m is the dimension of the input space. The network is trained via stochastic gradient descent (SGD) with momentum (RMSprop) to minimize the regularized L2 reconstruction error

$$\mathscr{L}(\boldsymbol{W}) = \sum_{s=1}^{T} \|\boldsymbol{x}_s - \boldsymbol{y}_s\|_2^2 + \lambda(\|\boldsymbol{W}_{\text{out}}\|_F^2 + \|\boldsymbol{W}_{\text{in}}\|_F^2) \tag{2}$$

where T is the size of the training dataset. Here the network is trained so that outputs $\boldsymbol{y}_s$ are close to $\boldsymbol{x}_s$, regularized by a cost on weights.

To build some intuition, consider the characteristics of the model for ϕ equal to the identity, $\lambda = 0$, and zero bias terms. The loss function Eq. (2) is then the same as that of principal component analysis, except that here we do not require orthonormality of the input (output) weight rows (columns). Without this constraint, the model is highly *non-identifiable*: it is impossible to infer the value of the parameters (here $\boldsymbol{W}_{\text{in}}$ and $\boldsymbol{W}_{\text{out}}$) by sampling from the model. This is because

$$\boldsymbol{y} = \boldsymbol{W}_{\text{out}}\boldsymbol{A}\boldsymbol{A}^{-1}\boldsymbol{W}_{\text{in}}\boldsymbol{x} = \boldsymbol{W}_{\text{out}}\boldsymbol{W}_{\text{in}}\boldsymbol{x}.$$

for any $N \times N$ invertible matrix $\boldsymbol{A}$, so that the same input-output mapping is satisfied by $\boldsymbol{W}_{\text{out}}\boldsymbol{A}$ and $\boldsymbol{A}^{-1}\boldsymbol{W}_{\text{in}}$ for any invertible $\boldsymbol{A}$. When the orthonormality condition is enforced, the same statement holds excepting that $\boldsymbol{A}$ in this case is an arbitrary orthogonal

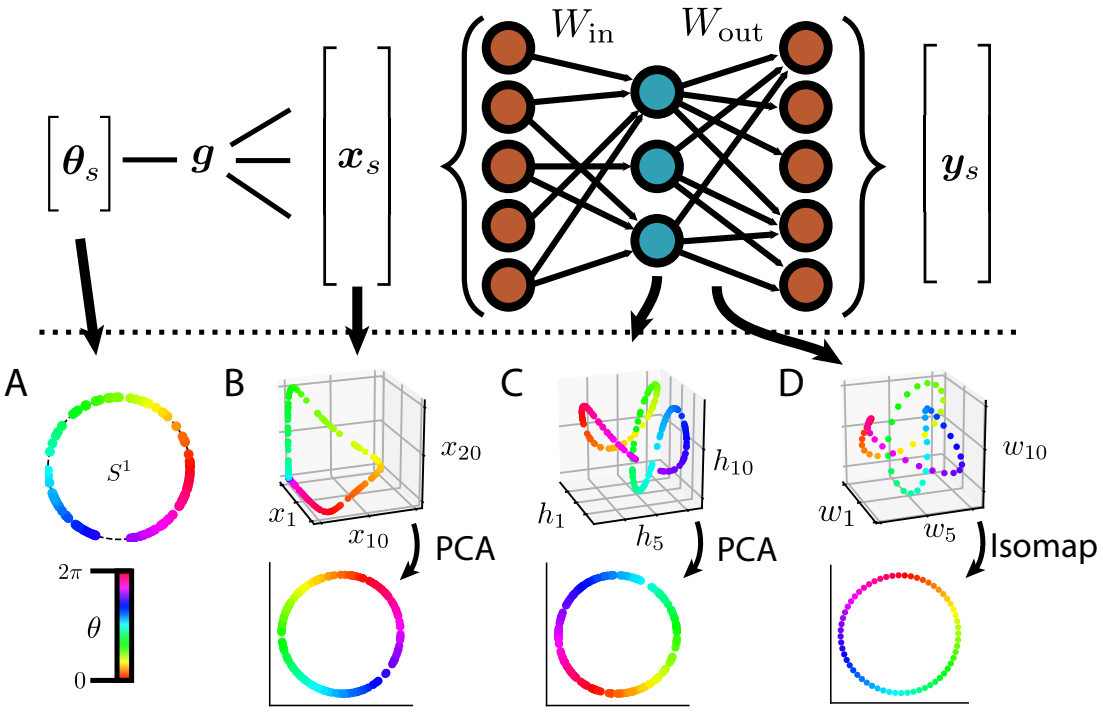

Figure 1: Overview of the characteristics of an autoencoder network trained to reproduce inputs that are generated by a periodic latent variable. Top: Network architecture. A low-dimensional set of latent variables is transformed into a high-dimensional input via a function $\boldsymbol{g}$. The network is then trained to reconstruct this input under the constraint of a bottleneck in the number of hidden unit neurons. A hyperbolic tangent nonlinearity is used for the hidden units. Bottom: Example network measurements for a periodic, scalar latent variable θ_s. Network is trained with $\lambda = 4e^{-6}$ and $m = 60$. (A-C) Color denotes value of θ_s. (A) Latent variable θ_s drawn from S^1. Color denotes value of θ_s. (B) Responses of receptive field neurons 1, 10, and 20 to θ_s. The receptive fields are periodic. Bottom: The response ensemble projected down to a two-dimensional space with PCA. (C) Responses of hidden units 1, 5, and 10 to θ_s, out of 10 hidden units. Bottom: The hidden unit response projected down to a two-dimensional space with PCA. (D) Columns 1, 5, and 10 of the output weight matrix, out of 10 columns. Color corresponds to the receptive field neuron index. While the structure is similar to (C), the relative scaling of the axes is different, as evidenced by the grid lines. Bottom: The output weight matrix projected onto a two-dimensional space with Isomap.

matrix as opposed to an arbitrary invertible matrix. Since we are focused on investigating the structure of the weights, it is important to push the network toward finding a set of solutions that is unique up to orthogonal transformations, as these are geometry preserving. This is accomplished by the requlatizer term $\|\boldsymbol{W}_{\text{out}}\|_F^2 + \|\boldsymbol{W}_{\text{in}}\|_F^2$ in Eq. (2) as we will discuss in more detail in Sec. 4.

Note that the matrix $\boldsymbol{A}$ that is found when training a neural network on the unregularized loss ($\lambda = 0$) may in practice not disturb geometric information very much. Specifically, if the Frobenius norm of $\boldsymbol{A}$ is not far from 1, then the geometric structure of $\boldsymbol{W}_{\text{out}}\boldsymbol{A}$ compared to $\boldsymbol{W}_{\text{out}}$ will be similar, and similarly for $\boldsymbol{W}_{\text{in}}$. Choice of

initialization of the weights before training, adding noise to the inputs while training, or implicit regularization caused by SGD may bias solutions toward regimes where the regularizer term of Eq. (2) is small. We have found experimentally that random weight initializations are often sufficient to lead to learned weights with recoverable structure even without regularization (data not shown).

We describe the training dataset next.

3 Constructing inputs generated by latent variables

3.1 Receptive field encoding of latent variables

Suppose that our inputs have the form $\boldsymbol{x}_s = \boldsymbol{g}(\boldsymbol{\theta}_s) \in \mathbb{R}^m$, where $\boldsymbol{\theta}_s \in \mathcal{S}$ is some low-dimensional underlying process in a space $\mathcal{S}$ and $\boldsymbol{g}$ maps this process into a higher-dimensional space embedded in $\mathbb{R}^m$. Here $\boldsymbol{\theta}_s$ is a latent variable that underlies the inputs $\boldsymbol{x}_s$, and the induced process $\boldsymbol{x}_s$ can be thought of as a high-dimensional encoding of $\boldsymbol{\theta}_s$. The subscript s is the index for samples of the corresponding variables, and is sometimes suppressed when context makes it clear. Throughout, mathematical symbols in bold font denote vectors or vector-valued functions, while scalars are denoted by lowercase symbols with normal font. When indexing the entries of these vectors with an index k, we either use the notation $(x_k)_s$ or suppress the s, simply writing x_k. In our mathematical analysis and simulations we assume that the samples $\boldsymbol{\theta}_s$ are drawn uniformly from $\mathcal{S}$.

The high-dimensional nature of the inputs is important in our framework. While networks that take lower-dimensional inputs and project them into a higher-dimensional hidden representation space are also of general interest, our objective here is to require the network to extract latent variables from a high-dimensional signal. This more constrained scenario affords a greater possibility that the weights will contain information about the latent variables. In particular, when treating weight matrices as geometric objects (as in Fig. 1D), the number of input dimensions m is the number of datapoints that we plot, and these points are embedded in an N-dimensional space. From this perspective, m will need to be large enough for meaningful structure to emerge.

One important class of encodings that increase the dimensionality of the encoded variables are the tuned neural response functions. In this case, each component x_k of $\boldsymbol{x}$ is the response of a neuron with tuning curve g_k to the variable $\boldsymbol{\theta}$. As an example, consider the classic case of direction-selective retinal ganglion cells. We suppose each ganglion cell to be tuned as a Gaussian centered at its preferred orientation, $g_k(\theta) \propto \exp(-d(\theta, z_k)^2/\sigma^2)$, where θ is an angle in the interval from 0 to 2π, z_k is the neuron's preferred orientation, and σ captures the width of the tuning. Here d is a distance function in angle space, which can be written $d(\theta, \theta') = \min\{|\theta - \theta'|, 2\pi - |\theta - \theta'|\}$. A visualization of θ and the response $\boldsymbol{g}(\theta)$ is shown in Fig. 2A. The characteristics of an autoencoder network trained to reproduce this periodic input is shown in Fig. 1. Here we see that the periodicity of the inputs appears in both the hidden representation of the network as well as the weights. This will be elucidated in Sec. 4.

Fig. 2B depicts a latent variable θ that is also scalar-valued, but differs from the previous example in that the latent space $\mathcal{S}$ is the closed interval $[0, 1]$ with the standard

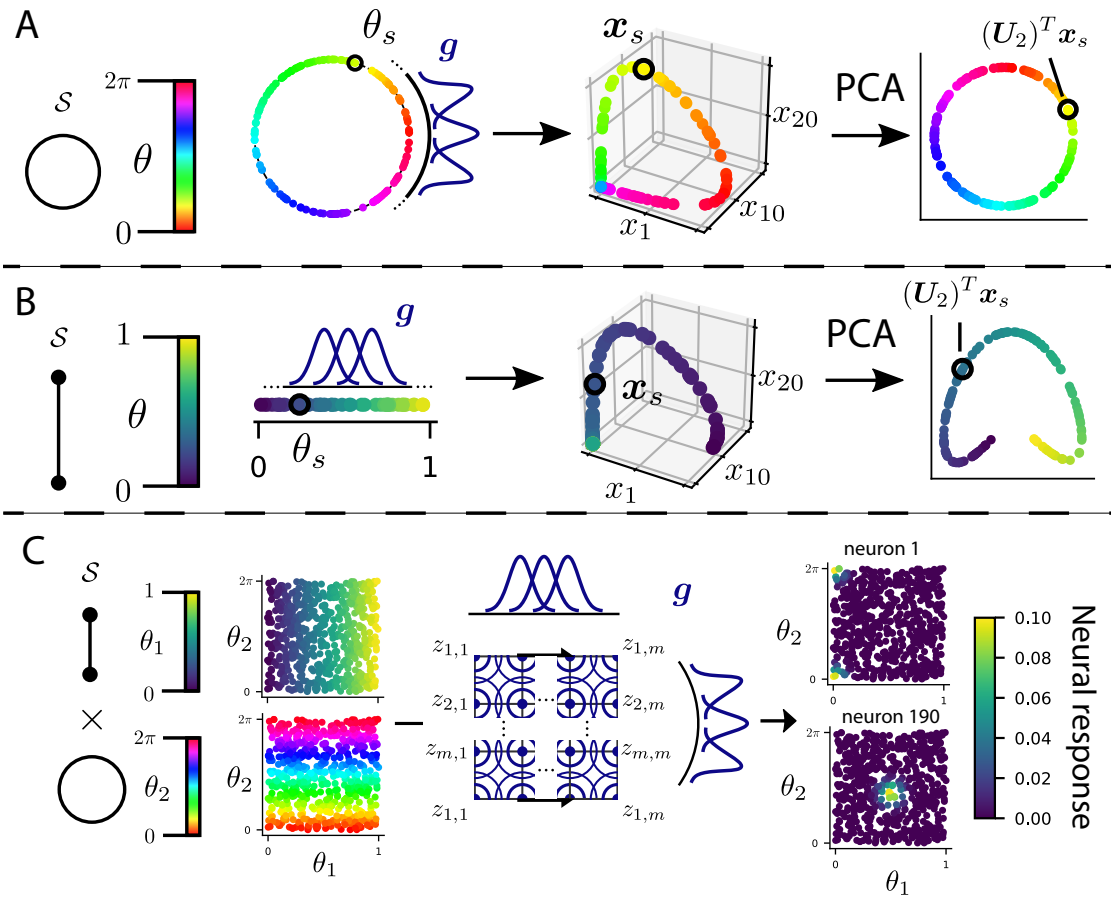

Figure 2: Depiction of latent variables on different spaces $\mathcal{S}$. (A) Example where $\mathcal{S}$ is a circle. A periodic scalar latent variable is transformed into a higher-dimensional encoding via the receptive field neural response function $\boldsymbol{g}$. The periodicity of θ is expressed by a periodic colormap for θ. The periodic structure is revealed by PCA. (B) Same as (A), but for a nonperiodic scalar latent variable, so that $\mathcal{S}$ is a line segment. (C) Example where $\mathcal{S}$ is a cylinder. Tensor product of a nonperiodic (θ_1) and a periodic (θ_2) latent variable is transformed into a higher-dimensional encoding via the receptive field neural response function $\boldsymbol{g}$. The scatter plots to the left depict the samples $\boldsymbol{\theta}_s$ with coloration based on θ_1 (top) and coloration based on θ_2 (bottom). Receptive field centers $\boldsymbol{z}_{k,k'}$ tile the latent space $\mathcal{S}$. Specifically, the top and bottom edges of the space are glued together. The responses of the first and 190th out of 400 receptive field neurons are shown on the right.

topology of an interval, as opposed to a circle. In this case the natural encoding is $g_k(\theta) \propto \exp(-d(\theta, z_k)^2/\sigma^2)$ where $d(\theta, z_k) = |\theta - z_k|$.

Another important case is that of "place cell" neurons tuned to (x, y) position on a grid over a two-dimensional flat surface $\mathcal{S}$. The space $\mathcal{S}$ can be thought of as the product space of the product of two line segments. In this case we suppose each place cell to be tuned as a gaussian centered at its location on the grid, $g_{k,\ell}(\boldsymbol{\theta}) \propto \exp(-d(\boldsymbol{\theta}, \boldsymbol{z}_{k,\ell})^2/\sigma^2)$ where $\boldsymbol{z}_{k,\ell}$ is the neuron's tuning center and $d(\boldsymbol{\theta}, \boldsymbol{z}_{k,\ell}) = \|\boldsymbol{\theta} - \boldsymbol{z}_{k,\ell}\|_2$. Note that in this case the indices (k, ℓ) need to be "unrolled" into a vector to form the vector-valued $\boldsymbol{g}(\boldsymbol{\theta}_s)$. The idea can be further extended to other latent variables, such as the joint spatial and orientation tuning seen also in retinal ganglion cells. To illustrate such a joint encoding, consider "place cell" neurons as before, but instead of tiling a grid, suppose that one edge of the grid wraps around and connects to the opposite edge, so that the neurons tile a cylinder. A realization of a latent variable on a cylinder and the responses of two receptive field neurons is depicted in Fig. 2C.

3.1 The latent variables appear in the weights of the trained autoencoder

Each of these examples illustrates a different topology of the space $\mathcal{S}$ on which the latent variable $\boldsymbol{\theta}_s$ can live. Our main finding is that, in our network trained to autoencode the inputs $\boldsymbol{x}_s$ generated by $\boldsymbol{\theta}_s$, the weights of the network generally reflect the topology of $\mathcal{S}$. This is in addition to the hidden unit activations in response to the inputs reflecting the topology of $\mathcal{S}$. An overview of this phenomenon in the case of a periodic latent variable is given in Fig. 1 as well as in Figs. 3A to 3C. In Fig. 3A, a trained autoencoder's response to the inputs $\boldsymbol{x}_s$ generated by a periodic latent variable θ_s is plotted in the top two-dimensional principal component space. In Fig. 3B, the top two principal components of the columns of the output weights are plotted. The resulting structure suggests periodicity, but isn't always clearly seen. Using the nonlinear dimensionality reduction method Isomap [15] to reduce the output weights to a two-dimensional space reveals the circular structure of the latent variable space clearly (Fig. 3C). A description of Isomap and explanation of its success in recovering the periodic structure is given in Sec. 4.3. This shows that the structure of the latent variable of the trained autoencoder is apparent not only in the hidden unit activities, but also in the learned weights of the network.

A similar phenomenon occurs for an autoencoder trained to reconstruct inputs $\boldsymbol{x}_s$ formed by receptive field responses $\boldsymbol{g}(\theta_s)$ to a non-periodic latent variable θ_s. Here we see that the topology of the latent space $\mathcal{S}$ is again reflected in the network weights (Figs. 3E and 3F).

This phenomenon also occurs in the case of inputs generated by tensored latent variables as in Fig. 2C, resulting in the weights reflecting the cylindrical topology of $\mathcal{S}$ (Figs. 3G to 3I). When both tensored variables are periodic, the structure in the weights is that of a torus (Fig. 3L)

While in the examples above the latent variables are reflected both in the structure of the weights and the structure of the hidden layer activations of the trained network, structure in the activations depends on the choice of inputs given to the network after training. In the example of a periodic random variable, the ring structure in the activations does not appear if white noise inputs are shown to the trained network (data not

shown). This illustrates how the information provided by the weights is in some ways distinct from that provided by the hidden unit activations.

Note also that the structure of the weights is most clearly extracted by the nonlinear dimensionality reduction method Isomap (Figs. 3C, 3F and 3I) as opposed to the linear method of principal component analysis (Figs. 3B, 3E and 3H). We shed light on why this is in Sec. 4.3.

4 Extracting latent variables from network weights

We now explain these observations through mathematical analysis. For ease of analysis, we consider a linear autoencoder model where ϕ is taken to be the identity. Our analysis involves three steps. The first is to relate the minimizers of Eq. (2) to the familiar solutions found by principal component analysis (PCA). The second is to resolve the problem of *non-identifiability* of the model, as introduced in Sec. 2. Once the minimizers of Eq. (2) have been related to the PCA solutions and the degeneracy of the solution space has been resolved, the third step in our analysis is to look more closely at the PCA solutions and to show that these solutions in fact encode the latent variable information.

4.1 Relating autoencoders to PCA

We start by rewriting the loss Eq. (2) in matrix form with $\lambda = 0$ and with ϕ taken to be the identity:

$$L(\boldsymbol{W}_{\text{out}}, \boldsymbol{W}_{\text{in}}, \boldsymbol{b}_1, \boldsymbol{b}_2) = \left\| \boldsymbol{X} - \boldsymbol{W}_{\text{out}}\boldsymbol{W}_{\text{in}}\boldsymbol{X} + \boldsymbol{W}_{\text{out}}\boldsymbol{b}_1\mathbf{1}_T^\top + \boldsymbol{b}_2\mathbf{1}_T^\top \right\|_F^2 \tag{3}$$

where $\boldsymbol{X}$ is an $m \times T$ matrix with column s holding the sample $\boldsymbol{x}_s$ and $\mathbf{1}_T$ is the length T vector of all ones. Let $\boldsymbol{\mu}_{\boldsymbol{x}} = \langle \boldsymbol{x}_s \rangle_s$ and $\boldsymbol{\mu}_{\boldsymbol{h}} = \langle \boldsymbol{h}_s \rangle_s$. The optimal values for the bias terms have the effect of transforming the problem into one that has been mean-centered, i.e. the minimal weights of Eq. (3) coincide with those of

$$\| \boldsymbol{X}' - \boldsymbol{W}_{\text{out}}\boldsymbol{H}' \|_F^2 \tag{4}$$

where $\boldsymbol{X}' = \boldsymbol{X} - \boldsymbol{\mu}_x\mathbf{1}^\top$ and $\boldsymbol{H}' = \boldsymbol{W}_{\text{in}}\boldsymbol{X} - \boldsymbol{\mu}_{\boldsymbol{h}}\mathbf{1}^\top$ [16]. Minimizing this loss while enforcing that $\boldsymbol{W}_{\text{in}}$ and $\boldsymbol{W}_{\text{out}}$ have orthonormal rows and columns, respectively, results in the PCA solution. This solution is naturally expressed in terms of the *singular value decomposition* (SVD) of $\boldsymbol{X}'$: $\boldsymbol{X}' = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ where $\boldsymbol{U}$ is an $m \times m$ orthogonal matrix, $\boldsymbol{V}$ is a $T \times m$ matrix with orthonormal columns, and $\boldsymbol{\Sigma}$ is an $m \times m$ diagonal matrix with nonnegative entries $\sigma_1, \sigma_2, ..., \sigma_m$ called the *singular values* of $\boldsymbol{X}'$. The standard PCA solutions are then $\boldsymbol{W}^*_{\text{out}} = \boldsymbol{U}_N$ and $\boldsymbol{W}^*_{\text{in}} = (\boldsymbol{U}_N)^T$ where $\boldsymbol{U}_N$ is the matrix $\boldsymbol{U}$ truncated to the first N columns. However, as discussed above any solution of the form $\boldsymbol{W}^*_{\text{out}} = \boldsymbol{U}_N\boldsymbol{A}$ and $\boldsymbol{W}^*_{\text{in}} = \boldsymbol{A}^{-1}(\boldsymbol{U}_N)^T$ is also a global minimum, where $\boldsymbol{A}$ is an arbitrary invertible $N \times N$ matrix. This is the most general form of optimal solution [17, 18].

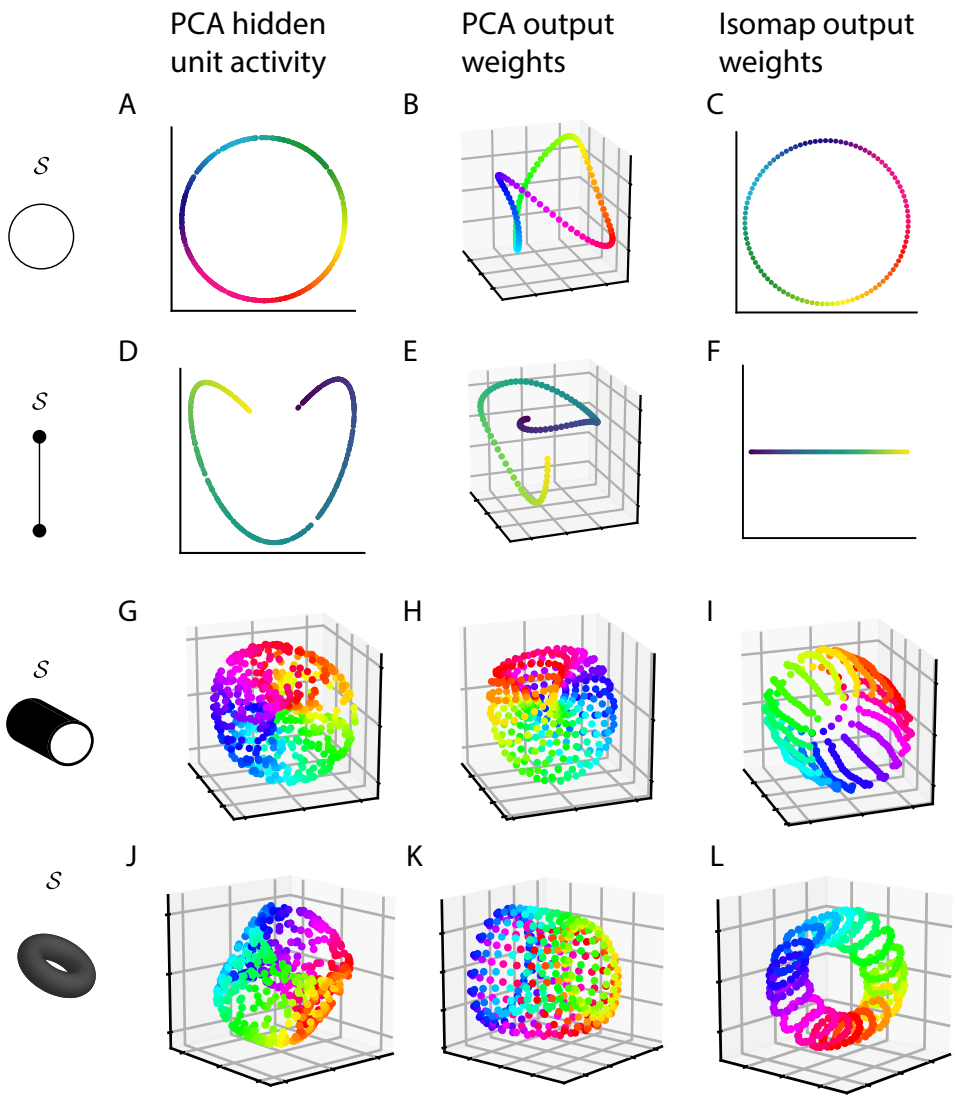

Figure 3: The structure of the latent variables can be recovered from the weights of the trained autoencoder by nonlinear dimensionality reduction methods. (A) Hidden unit activations of the autoencoder trained to reconstruct an encoding $\boldsymbol{g}(\theta_s)$ of a periodic latent variable θ_s as in Fig. 2A and using the same coloration. (B) Principal components of the columns of the output weights of the autoencoder trained on the periodic latent variable. (C) Two-dimensional embedding via Isomap of the columns of the output weights for the network trained on the periodic inputs. (D-F) As in (A-C) but for an encoding $\boldsymbol{g}(\theta_s)$ of a nonperiodic latent variable θ_s as in Fig. 2B. (G-I) As in (A-C) but for a joint encoding $\boldsymbol{g}(\boldsymbol{\theta}_s)$ of a periodic and non-periodic latent variable, such as that illustrated by Fig. 2C. In this case the Isomap embedding in (I) is three-dimensional. In (G) color corresponds with the periodic latent variable, while in (H-I) coloration is by the index of the receptive field centers corresponding to the periodic latent variable. (J-L) Same as in (G-I), but for a joint encoding of two periodic latent variables. Color corresponds with the first latent variable. The latent space $\mathcal{S}$ in this case is a torus.

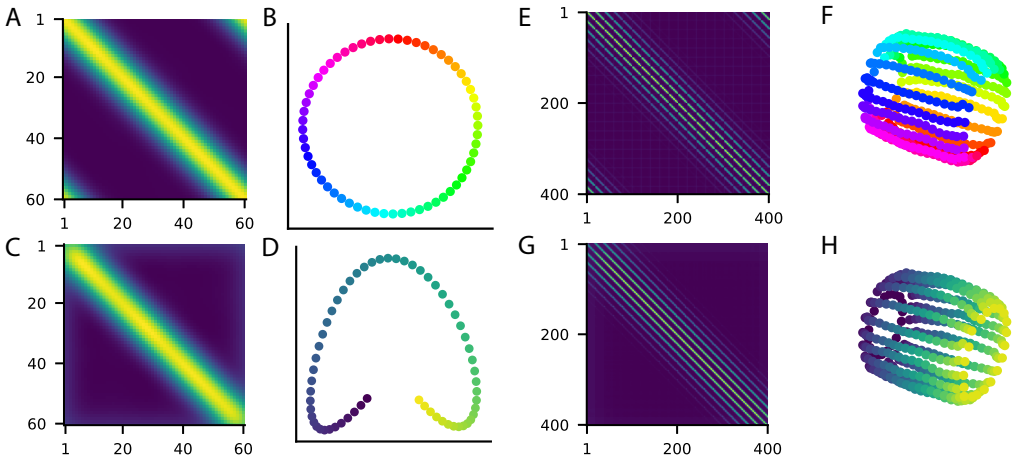

Figure 4: The structure of the latent variables can be recovered from the eigenvectors of the covariance matrix of the inputs. (A) Covariance matrix for the responses of $m = 60$ receptive field neurons to a periodic latent variable as in Fig. 2A. (B) Lowest frequency eigenvectors of the circulant matrix in A, plotted against each other and colored by index. (C) Covariance matrix for the responses of $m = 60$ receptive field neurons to a nonperiodic latent variable as in Fig. 2B. (D) Lowest frequency eigenvectors of the covariance matrix in (C), plotted against each other and colored by index. (E) Covariance matrix resulting from the tensored responses to a periodic and nonperiodic latent variable as in Fig. 2C, where the periodic variable is in the first coordinate and the nonperiodic variable is in the second. (F) Eigenvectors of the covariance matrix in (E), reduced to three dimensions by Isomap. Coloration is by the index of the receptive field centers corresponding to the periodic latent variable. The eigenvectors for the covariance matrix in (G) look similar. (G) Covariance matrix as in (E) but with the position of the nonperiodic and periodic variable switched. (H) Same as (F), but colored by the index of the receptive field centers corresponding to the nonperiodic latent variable.

4.2 Resolving non-uniqueness of the optimal weights

The arbitrary invertible linear transformation $\boldsymbol{A}$ described in the previous section can potentially skew the structure of the weights past the point where the latent variables can be extracted. While $\boldsymbol{A}$ preserves topological information, it doesn't necessarily preserve local distances between points. Nonlinear dimensionality reduction methods like Isomap are generally designed to embed points in a lower-dimensional space while preserving local distances, and losing information about these local distances can be destructive. This can become a problem in practice since the number of columns (rows) of $\boldsymbol{W}_{\text{in}}$ ($\boldsymbol{W}_{\text{out}}$) are finite, and the structure of local distances among finitely many points can be lost as skewing becomes large. Here we describe biologically motivated conditions that address this issue.

If we add the regularizer $\lambda(\|\boldsymbol{W}_{\text{out}}\|_F^2 + \|\boldsymbol{W}_{\text{in}}\|_F^2)$ with $\lambda > 0$ to Eq. (3), then the solution becomes more constrained. Let $\boldsymbol{\Sigma}_\lambda$ be the diagonal matrix $\boldsymbol{\Sigma}_\lambda = \text{diag}(\lfloor 1 - \lambda/\sigma_1^2 \rfloor_+, ..., \lfloor 1 - \lambda/\sigma_N^2 \rfloor_+)$, where the σ_k are again the singular values of $\boldsymbol{X}$, λ is the scaling of the regularizer, and $\lfloor \cdot \rfloor_+$ is the threshold function $\max\{\cdot, 0\}$. [17, 18] recently showed that, under the assumption that $\sigma_1 > \sigma_2 > ... > \sigma_N$, the optimal weights in

this case have the form $\boldsymbol{W}^*_{\text{in}} = \boldsymbol{Q}^T\boldsymbol{\Sigma}_\lambda^{1/2}(\boldsymbol{U}_N)^T$ and $\boldsymbol{W}^*_{\text{out}} = \boldsymbol{U}_N\boldsymbol{\Sigma}_\lambda^{1/2}\boldsymbol{Q}$ where $\boldsymbol{Q}$ is an arbitrary $N \times N$ orthogonal matrix. In particular, these solutions are unique up to arbitrary orthogonal transformations $\boldsymbol{Q}$, rather than arbitrary invertible transformations $\boldsymbol{A}$.

Therefore, this regularizer can help to preserve the geometric information encoded in the weights found by optimization methods. In particular, for positive but small λ, $\boldsymbol{W}^*_{\text{in}} \approx \boldsymbol{Q}^T(\boldsymbol{U}_N)^T$ and $\boldsymbol{W}^*_{\text{out}} \approx \boldsymbol{U}_N\boldsymbol{Q}$. Since $\boldsymbol{Q}$ preserves distances (in other words, $\boldsymbol{Q}$ preserves geometric information), analyzing the geometric structure of optimal weights $\boldsymbol{W}^*_{\text{in}}$ and $\boldsymbol{W}^*_{\text{out}}$ reduces to analyzing the geometric structure of $\boldsymbol{U}_N$.

As we show next, the matrix $\boldsymbol{U}_N$ contains geometric information about the inputs.

4.3 Relating the weights to the latent variables

In the previous section we showed how the optimal weights share the same geometric structure as $\boldsymbol{U}_N$: $\boldsymbol{W}^*_{\text{in}} \approx \boldsymbol{Q}^T(\boldsymbol{U}_N)^T$ and $\boldsymbol{W}^*_{\text{out}} \approx \boldsymbol{U}_N\boldsymbol{Q}$ for λ small. We now show how $\boldsymbol{U}_N$ is related to the latent variables underlying the inputs. To do so, we first note that according to basic properties of the SVD, $\boldsymbol{U}$ is a matrix of normalized eigenvectors of the covariance matrix $\boldsymbol{C} = \left\langle \boldsymbol{x}_s\boldsymbol{x}_s^T \right\rangle_s - \left\langle \boldsymbol{x}_s \right\rangle_s \left\langle \boldsymbol{x}_s \right\rangle_s^T$ of $\boldsymbol{X}$, so $\boldsymbol{C} = \boldsymbol{U}\boldsymbol{\Sigma}^2\boldsymbol{U}^T$ (recall that $\boldsymbol{\Sigma}$ holds the singular values for the mean-centered $\boldsymbol{X}'$). Assuming that $\boldsymbol{x}$ has the form $x_j = g_j(\boldsymbol{\theta}) \propto \exp(-d(\boldsymbol{\theta}, \boldsymbol{z}_j)^2/\sigma^2)$, we can work out the form of the covariance between x_j and x_k. Taking the limit $T \to \infty$ and invoking the law of large numbers, we have that $\langle (x_j)_s \rangle_{s=1,\ldots,\infty} = \langle g_j(\boldsymbol{\theta}) \rangle_{\boldsymbol{\theta}}$ and $\langle (x_j)_s (x_k)_s \rangle_{s=1,\ldots,\infty} = \langle g_j(\boldsymbol{\theta}) g_k(\boldsymbol{\theta}) \rangle_{\boldsymbol{\theta}}$. Letting $\mu_j = \langle g_j(\boldsymbol{\theta}) \rangle_{\boldsymbol{\theta}}$, it follows that in this limit

$$C_{jk} = \langle g_j(\boldsymbol{\theta}) g_k(\boldsymbol{\theta}) \rangle_{\boldsymbol{\theta}} - \mu_j \mu_k.$$

From this form of the covariance matrix, we can now work out the eigenvector structure for different choices of the latent space $\mathcal{S}$ and distance function d.

Periodic latent variables give rise to periodic weight structure

We first consider the case of a periodic latent variable θ. Recall that the inputs are formed by encoding θ via orientation selective receptive fields $g_k(\theta) \propto (\exp(-d(\theta, z_k)^2/\sigma^2)$, with d being a distance function in angle space, $d(\theta, \theta') = \min\{|\theta - \theta'|, 2\pi - |\theta - \theta'|\}$. Assume that the receptive field centers z_k evenly tile the space.

The salient structure of this encoding can be expressed through the idea of *equivariance*. Let $\overline{a+b}$ denote addition of a and b modulo the number, m, of receptive field neurons. In our scenario, equivariance means that shifting the identity of the receptive field neuron is the same as shifting the input to the receptive field neuron: $g_{\overline{k+\ell}}(\theta) = g_k(\theta - z_\ell \mod 1)$. This equivariance implies a special structure of the input covariance matrix $\boldsymbol{C}$: $\boldsymbol{C}$ is a *circulant* matrix. This means that every row in $\boldsymbol{C}$ is a shifted version of the first row, where the shifting operation wraps around at the edges of the matrix. To show this, we show that entry C_{jk} is equal to $C_{\overline{j+\ell},\overline{k+\ell}}$, where ℓ is any integer.

Recall our assumption that θ is uniformly distributed on the circle $\mathcal{S} = S^1$, and suppose without loss of generality that $\mathcal{S}$ has Lebesque measure 2π (so θ varies from 0 to 2π). Then the probability density function of θ is the constant function that returns

$1/(2\pi)$ for all θ. This means in particular that the expected value of $f(\theta)$ is $\langle f(\theta)\rangle = \frac{1}{2\pi}\int_0^{2\pi} f(\theta)d\theta$ for any reasonably well-behaved function f.

To show that $\boldsymbol{C}$ is circulant, we first compute that

$$\begin{aligned}
\langle g_{\overline{j+\ell}}(\theta) g_{\overline{k+\ell}}(\theta)\rangle &= \langle g_j\left(\theta - z_\ell \mod 1\right) g_k\left(\theta - z_\ell \mod 1\right)\rangle \\
&= \frac{1}{2\pi}\int_0^{2\pi} g_j\left(\theta - z_\ell \mod 2\pi\right) g_k\left(\theta - z_\ell \mod 2\pi\right) d\theta \\
&= \frac{1}{2\pi}\int_{0-z_\ell \mod 2\pi}^{2\pi - z_\ell \mod 2\pi} g_j\left(\theta\right) g_k\left(\theta\right) d\theta \\
&= \frac{1}{2\pi}\int_0^{2\pi} g_j\left(\theta\right) g_k\left(\theta\right) d\theta \\
&= \langle g_j(\theta) g_k(\theta)\rangle. \qquad (5)
\end{aligned}$$

Computing shifts of the mean μ_j has a similar flavor:

$$\begin{aligned}
\mu_{\overline{j+\ell}} &= \frac{1}{2\pi}\int_0^{2\pi} g_j\left(\theta - z_\ell \mod 2\pi\right) d\theta \\
&= \frac{1}{2\pi}\int_{0-z_\ell \mod 2\pi}^{2\pi - z_\ell \mod 2\pi} g_j\left(\theta\right) d\theta \\
&= \frac{1}{2\pi}\int_0^{2\pi} g_j\left(\theta\right) \\
&= \mu_j,
\end{aligned}$$

so that μ_j is independent of its index. Hence we can write $\mu_j\mu_k = \mu^2$. Combining this with Eq. (5), we have that $C_{\overline{j+\ell},\overline{k+\ell}} = C_{jk}$. An example of the resulting circulant matrix is shown in Fig. 4A.

In addition to being circulant, the covariance matrix $\boldsymbol{C}$ is by definition symmetric: $\boldsymbol{C} = \boldsymbol{C}^T$. The eigenvectors of circulant matrices are known and together make up the discrete Fourier transform matrix [19]. In the case where the circulant matrix is also symmetric, the real and imaginary parts of the eigenvectors are themselves eigenvectors. This means that an eigenvector basis for $\boldsymbol{C}$ can be taken to be real, which results in eigenvectors that have one of three forms: cosine transforms, sine transforms, and the all-ones vector $\mathbf{1}_m$ (recall that m is the dimension of the inputs $\boldsymbol{x}_s$). More precisely, the jth cosine transform eigenvector has the form $v_k^{(j)} = \cos(2\pi jk/m)$ for $k \in \{0, 1, ..., m-1\}$ and the jth sine transform eigenvector has the form $w_k^{(j)} = \sin(2\pi jk/m)$. In particular, the eigenvectors are periodic, reflecting the periodicity of the latent variable θ. These eigenvectors together form the columns of the matrix $\boldsymbol{U}$. As an illustration, the eigenvectors $\boldsymbol{v}^{(1)}$ and $\boldsymbol{w}^{(1)}$ are plotted against each other in Fig. 4B.

Consider the truncation $\boldsymbol{U}_N$ of $\boldsymbol{U}$ to N columns. We're interested in the properties of $\boldsymbol{U}_N$ embedded as a geometric object, with each row constituting a single data point in N-dimensional space. The sine and cosine structure of the eigenvectors ensures that this structure is periodic. In particular, the rows of $\boldsymbol{U}_N$ are m samples from a loop that nonlinearly curves through N-dimensional space.

Since this loop structure of $\boldsymbol{U}_N$ is nonlinearly embedded, nonlinear dimensionality reduction methods are well suited for recovering this structure. Indeed, since the singular values of $\boldsymbol{U}_N$ are all 1, trying to extract structure from it with the linear method

of PCA will only return a random set of the columns of $\boldsymbol{U}_N$. In general, the ability of nonlinear dimensionality reduction methods to successfully extract the structure of interest from a dataset depends on having enough datapoints, and our situation is no exception. In our case, the number of datapoints is m, and this will need to be a large enough number for the dimensionality reduction method to succeed. The precise number of datapoints needed will depend on the specifics of the dimensionality reduction method used. To proceed, we will assume that m is sufficiently large.

For intuition as to why nonlinear methods work, we focus on the approach of the nonlinear method Isomap. The first step of Isomap involves building a graph on the datapoints where points that are sufficiently close are connected by an edge. Let's consider the strategy of connecting every point to its two nearest neighbors. Then in our case this graph will indeed be a loop through high-dimensional space, and embedding this graph in two dimensions in a way that best preserves distance information reveals a ring.

Recall the minimizer $\boldsymbol{W}^*_{\text{out}} \approx \boldsymbol{U}_N\boldsymbol{Q}$, $\boldsymbol{W}^*_{\text{in}} \approx \boldsymbol{Q}^T(\boldsymbol{U}_N)^T$ of the regularized loss for the linear model. In practice we find that the periodicity of $\boldsymbol{U}_N$ is reflected in the weights $\boldsymbol{W}_{\text{out}}$ in the autoencoder trained with stochastic gradient descent. As illustrated by Fig. 3, this extends to networks with $\tanh$ nonlinearity. Here the network is trained with $\lambda = 4e^{-6}$ and $m = 100$. The latent variable structure can be partially seen in the apparent periodicity of points obtained by using PCA to project the columns of $\boldsymbol{W}_{\text{out}}$ onto a three-dimensional space in Fig. 3B. As discussed above, this periodicity is revealed more clearly by using Isomap to "unravel" the coils caused by the higher frequency modes, as can be seen in Fig. 3C.

Nonperiodic latent variables give rise to nonperiodic weight structure

The above analysis can be repeated in a similar form for the case of a nonperiodic latent variable θ on a line segment, where this time $g_k(\theta) \propto \exp(-|\theta - z_k|^2/\sigma^2)$. Suppose the receptive field centers z_1 through z_m evenly tile the line segment $[0, 1]$, with $z_1 = 0$ and $z_m = 1$. While we are interested in the case where θ is uniformly distributed on $[0, 1]$, this becomes mathematically challenging to work with due to conditions at the boundary being different than conditions in the center of the interval. Instead, we let $\mathcal{S}$ be the interval $[-s, s + 1]$, with the usual interval topology. Taking s sufficiently large will allow us to deal with boundary effects; for instance, this assumption ensures that $\langle g_k(\theta)\rangle$ is approximately independent of k. In this case the covariance matrix, instead of being circulant, is approximately *Toeplitz*, which means that the entries on each descending diagonal from left to right are the same. This can be seen by choosing indices j, k and ℓ constrained such that $j, k \in \{1, ..., m\}$, $j + \ell \in \{1, ..., m\}$, $k + \ell \in$

$\{1, ..., m\}$ and computing

$$\begin{aligned}
\langle g_{j+\ell}(\theta) g_{k+\ell}(\theta)\rangle &= \langle g_j\left(\theta - z_\ell\right) g_k\left(\theta - z_\ell\right)\rangle \\
&= \frac{1}{2s+1}\int_{-s}^{s+1} g_j\left(\theta - z_\ell\right) g_k\left(\theta - z_\ell\right) d\theta \\
&= \frac{1}{2s+1}\int_{-s-z_\ell}^{s+1-z_\ell} g_j\left(\theta\right) g_k\left(\theta\right) d\theta \\
&\approx \frac{1}{2s+1}\int_{-s}^{s+1} g_j\left(\theta\right) g_k\left(\theta\right) d\theta \\
&= \langle g_j(\theta) g_k(\theta)\rangle.
\end{aligned}$$

The approximation is justified when s is much larger than z_m, since the contribution to the integral near the integration limits is vanishingly small. This approximation becomes exact for large enough s if we clip the receptive field functions g_j to have finite support.

Computing the shifted means has a similar flavor:

$$\begin{aligned}
\langle g_{j+\ell}(\theta)\rangle &= \langle g_j\left(\theta - z_\ell\right)\rangle \\
&= \frac{1}{2s+1}\int_{-s}^{s+1} g_j\left(\theta - z_\ell\right) d\theta \\
&= \frac{1}{2s+1}\int_{-s-z_\ell}^{s+1-z_\ell} g_j\left(\theta\right) d\theta \\
&\approx \frac{1}{2s+1}\int_{-s}^{s+1} g_j\left(\theta\right) d\theta \\
&= \langle g_j(\theta)\rangle.
\end{aligned}$$

Taken together, these equations imply that $C_{j+\ell,k+\ell} = C_{j,k}$, so that $\boldsymbol{C}$ is Toeplitz. In our simulations we take $s = 0$ so that $\boldsymbol{C}$ is only approximately Toeplitz, but find that the conclusions below still hold in practice.

While the eigenvectors of Toeplitz matrices are not in general determined as they are for circulant matrices, they are known for tridiagonal Toeplitz matrices. Symmetric tridiagonal Toeplitz matrices all have the same eigenvectors, of the form $u_j^{(k)} = a \sin\left(\frac{j\pi k}{m+1}\right)$ for $k, j = 1, ..., m$, where a is an arbitrary nonzero scalar. The odd eigenvectors $\boldsymbol{u}^{(2k+1)}$ are symmetric (which in particular means that $u_1^{(2k+1)} = u_m^{(2k+1)}$) while the even eigenvectors are antisymmetric (in particular, $u_1^{(2k)} = -u_m^{(2k)}$). Recall that the $\boldsymbol{u}^{(k)}$ make up the columns of $\boldsymbol{U}$.

As before, we consider $\boldsymbol{U}_N$ as a geometric object embedded in N dimensional space, where the rows are datapoints. Under the assumptions of tridiagonal covariance matrix, the eigenvectors $\boldsymbol{u}^{(k)}$ given above reveal a particular structure: in our numerical tests, the rows of $\boldsymbol{U}_N$ lie along a curve with the endpoints disconnected, provided that $N < m$. To show this, we need to show that the distance the first and last row of $\boldsymbol{U}_N$ is larger than the distance between adjacent rows. While we do not prove this here, in practice we have found this to be the case numerically (data not shown). In fact, for

$m - N$ sufficiently large we find that the distance between the first and last row is larger than the distance between rows k and $k+2$, for $k = 1, ..., m-2$ as well.

With this "gap" between the first and last row of $\boldsymbol{U}_N$, we can use nonlinear dimensionality reduction methods to reveal the structure of a line segment. Consider again the strategy of connecting every point to its two nearest neighbors, as is done in using Isomap. In this case the "middle" sections of the curve will look as in the case of the periodic latent variable, but the ends will be different. If $m - N$ is large enough then the two endpoints of the line will not be connected, and the general structure of the graph will be that of a line.

Now we consider Toeplitz matrices with more than three (but still finitely many) nonzero diagonals. For fixed k, the eigenvector $\boldsymbol{u}^{(k)}$ in this case approaches the form $a \sin\left(\frac{j\pi k}{m+1}\right)$ in the asymptotic limit of large m [20]. It follows that, after truncating to $\boldsymbol{U}_N$ for finite N and taking m to be large, we can use the same reasoning as in the tridiagonal case to infer that the rows of $\boldsymbol{U}_N$ lie along a curve with the endpoints disconnected. The structure of the eigenvectors is illustrated by plotting $\boldsymbol{u}^{(1)}$ and $\boldsymbol{u}^{(2)}$ against each other in Fig. 4D for $m = 60$.

This topology appears in the weights of the trained autoencoder, as shown by Isomap in Fig. 3F. Here the network is trained with $\lambda = 4e^{-6}$ and $m = 100$. PCA projections of the weights do not reveal this structure as clearly (Fig. 3E).

Tensored latent variables give rise to tensored weights

In this section we consider combinations of latent variables found by taking tensor products of other latent variables. Consider the case of "place cell" encoding on a torus, where both boundaries of the grid are periodic. This can be thought of as a tensored combination of two periodic latent variables. Suppose that the first and second coordinates of $\boldsymbol{\theta}$ correspond to the periodic latent variables θ_1 and θ_2, respectively, and that each is i.i.d. uniformly distributed on the circle S^1.

Recall our choice of Gaussian curve response function on the circle: $g_k(\theta) = a \exp(-d_{S^1}(\theta, z_k)^2/\sigma^2)$ where a is a positive scalar and $d_{S^1}(\theta, \theta') = \min\{|\theta - \theta'|, 2\pi - |\theta - \theta'|\}$ is distance on the circle. We abuse notation slightly and use the same name for a Gaussian curve response function on the torus: $g_{i,k}(\boldsymbol{\theta}) = a^2 \exp(-d(\boldsymbol{\theta}, \boldsymbol{z}_{i,k})^2/\sigma^2)$ where d is Euclidean distance on the torus, which can be written

$$d(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sqrt{d_{S^1}(\theta_1, \theta_1')^2 + d_{S^1}(\theta_2, \theta_2')^2}.$$

This time the tuning curve centers $\boldsymbol{z}_{i,k}$ have two indices and evenly tile the two-dimensional surface of the torus. Our goal is to decompose $\boldsymbol{C}_{i,j,k,\ell} = \langle g_{i,k}(\boldsymbol{\theta}) g_{j,\ell}(\boldsymbol{\theta})\rangle - \langle g_{i,k}(\boldsymbol{\theta})\rangle\langle g_{j,\ell}(\boldsymbol{\theta})\rangle$ into contributions from tuning curves $g_k(\theta)$ defined on the circle. We start with the observation that

$$g_{ik}(\boldsymbol{\theta}) = g_i(\theta_1)\, g_k(\theta_2)\,.$$

Using this, along with independence of θ_1 and θ_2,

$$\begin{aligned}
\boldsymbol{C}_{i,j,k,\ell} &= \langle g_{i,k}(\boldsymbol{\theta}) g_{j,\ell}(\boldsymbol{\theta})\rangle - \langle g_{i,k}(\boldsymbol{\theta})\rangle\langle g_{j,\ell}(\boldsymbol{\theta})\rangle \\
&= \langle g_i(\theta_1)\, g_j(\theta_1)\, g_k(\theta_2)\, g_\ell(\theta_2)\rangle - \langle g_i(\theta_1)\, g_k(\theta_2)\rangle\langle g_j(\theta_1)\, g_\ell(\theta_2)\rangle \\
&= \langle g_i(\theta_1)\, g_j(\theta_1)\rangle\langle g_k(\theta_2)\, g_\ell(\theta_2)\rangle - \langle g_i(\theta_1)\rangle\langle g_k(\theta_2)\rangle\langle g_j(\theta_1)\rangle\langle g_\ell(\theta_2)\rangle.
\end{aligned}$$

Recall that $\langle g_j(\theta)\rangle = \mu$ is independent of j. If we let $(\boldsymbol{C}_{S^1})_{ij} = \langle g_i(\theta)\, g_j(\theta)\rangle - \mu^2$ be the covariance matrix for inputs on a circle, then we can write

$$\begin{aligned} C_{i,j,k,\ell} &= ((\boldsymbol{C}_{S^1})_{ij} + \mu^2)((\boldsymbol{C}_{S^1})_{k\ell} + \mu^2) - \mu^4 \\ &= (\boldsymbol{C}_{S^1})_{ij}(\boldsymbol{C}_{S^1})_{k\ell} + \mu^2(\boldsymbol{C}_{S^1})_{ij} + \mu^2(\boldsymbol{C}_{S^1})_{k\ell}. \end{aligned}$$

From this equation, we can see that $\boldsymbol{C}$ can be written as sums of Kronecker tensor products (denoted $\otimes$):

$$\boldsymbol{C} = \boldsymbol{C}_{S^1} \otimes \boldsymbol{C}_{S^1} + \mu^2 \boldsymbol{C}_{S^1} \otimes \mathbf{1}_m \mathbf{1}_m^T + \mu^2 \mathbf{1}_m \mathbf{1}_m^T \otimes \boldsymbol{C}_{S^1}.$$

Matrix multiplication of Kronecker products satisfies the *mixed-product property*: $(\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{C} \otimes \boldsymbol{D}) = (\boldsymbol{AC}) \otimes (\boldsymbol{BD})$. Suppose that $\boldsymbol{u}$ and $\boldsymbol{v}$ are eigenvectors of $\boldsymbol{C}_{S^1}$ with eigenvalues λ_u and λ_v, respectively. Then

$$\begin{aligned} \boldsymbol{C}(\boldsymbol{u} \otimes \boldsymbol{v}) &= \lambda_u \lambda_v \boldsymbol{u} \otimes \boldsymbol{v} + \mu^2 \lambda_u \boldsymbol{u} \otimes \mathbf{1}_m \mathbf{1}_m^T \boldsymbol{v} + \mu^2 \lambda_v \mathbf{1}_m \mathbf{1}_m^T \boldsymbol{u} \otimes \boldsymbol{v} \\ &= \lambda_u \lambda_v \boldsymbol{u} \otimes \boldsymbol{v} + \mu^2 \lambda_u \|\boldsymbol{v}\|_1 \boldsymbol{u} \otimes \mathbf{1}_m + \mu^2 \lambda_v \|\boldsymbol{u}\|_1 \mathbf{1}_m \otimes \boldsymbol{v}. \end{aligned}$$

This equation reveals that $\boldsymbol{u} \otimes \boldsymbol{v}$ are eigenvectors of $\boldsymbol{C}$ provided either (1) $\mu = 0$ or (2) $\|\boldsymbol{u}\|_1 = 0$ or $\boldsymbol{u} = \mathbf{1}_m$, and $\|\boldsymbol{v}\|_1 = 0$ or $\boldsymbol{v} = \mathbf{1}_m$. The eigenvectors of $\boldsymbol{C}_{S^1}$ satisfy condition (2), as is easy to verify. Hence in the case of inputs formed from tensoring two periodic latent variables, we can find closed form solutions for the eigenvectors of the covariance matrix. These eigenvectors are tensor products of periodic latent variables, so that their structure reflects that of a torus (twisted nonlinearly through N dimensional space). This structure appears in the weights of the trained autoencoder (Fig. 3L). Here the network is trained with $\lambda = 4e^{-6}$ and $m = 20 \cdot 20 = 400$.

In the case where one or both of the variables being tensored is nonperiodic, we currently lack a general mathematical characterization of the eigenvectors. In the special case when the mean response is zero, $\mu = 0$, the eigenvectors of $\boldsymbol{C}$ are the tensor products of the eigenvectors of the covariance matrices for the two latent variables. This can be shown by similar reasoning as above. Even when μ is nonzero, we see experimentally that the structure of the tensor product of a periodic and nonperiodic latent variable resembles a breaking of the toroidal structure similar to the scalar case. In particular, one end of the torus has a gap, which makes the structure resemble that of a cylinder. In this case the covariance matrix has the form shown in Fig. 4E if the periodic variable is the first coordinate and Fig. 4G if the nonperiodic variable is the first coordinate (this relationship may be reversed depending on how the four indices of $\boldsymbol{C}$ are unrolled into two indices). The cylindrical structure can be seen in Figs. 4F and 4H. This cylindrical structure is also reflected in the weights of the autoencoder (Fig. 3I). Here the network is trained with $\lambda = 4e^{-6}$ and $m = 20 \cdot 20 = 400$.

5 Discussion

It is important to investigate the ways that connectivity data can be used to help us understand neural circuits. Here we focus on using dimensionality reduction techniques to infer elements of the function of a neural circuit from the structure of the weights.

We find that the latent variables that underlie the inputs can be recovered from the weights of an autoencoder with a single layer of hidden units. This is accomplished via nonlinear dimensionality reduction methods, such as Isomap. In particular, periodic inputs give rise to periodic weight structure, and nonperiodic inputs to nonperiodic weight structure. The tensor products of such inputs results in an analogous structure in the weights. The emergence of this structure depends on regularization to penalize large weights. It also depends on the inputs encoding low-dimensional latent variables in a high-dimensional way.

The approach of focusing on connectivity data to deduce information about the function of a neural circuit complements other very fruitful efforts of probing the activity of neurons in the circuit. The latter includes the seminal work of Hubel and Wiesel [21], which provided strong evidence via recordings in cat striate cortex that neural responses in this area are built from simple combinations of the responses of retinal ganglion cell neurons. Another noteworthy example is the analysis of bump-attractor-like dynamics in the *Drosophila* ellipsoid body [22], which demonstrated through two-photon calcium imaging that the circuit tracks orientation information through integrated sensory information. There are, however, difficulties in using neural activations alone to draw inferences. For instance, it isn't always clear how to satisfactorily explore the space of all possible input stimuli. Often multiple competing models arise to reproduce neural circuit function or neural activity, and connectivity data can be used to select among them [6]. Connectivity data may also be useful for choosing parameters in models that are overparametrized. In the *Drosophila* ellipsoid body example, fine-grain analysis of connectivity data will probably be necessary to answer once and for all whether the bump attractor dynamics are implemented by a ring attractor network topology, and how this ring attractor is implemented precisely (see [23] for significant recent steps in this direction).

As analysis of connectivity data has its own set of shortcomings – for instance, information about neural modulation, the precise nonlinear responses of neurons to inputs, and many other factors are left out – hybrid methods that take into account both neural activation as well as connectivity data will be important far into the future. There are also other promising avenues for using connectivity data in ways distinct from the methods considered here. One approach is to develop models that fit neural activity data or that satisfy the believed function of a neural circuit while constraining them with connectivity data (reviewed in [6]). Another fruitful approach is to first generate a network model with a connectivity determined through data-driven means, assume a form of neural unit dynamics in the model, and then analyze the resulting network dynamics (for instance, [24, 25]).

Our approach is limited both by the simplicity of the task and network model, as well as the need to have exact values for the weights of synapses between neurons (a value that is difficult to assign in data). Our mathematical analysis assumes linear neural responses, and while we observe that these results extend in this case to hyperbolic tangent nonlinear responses in simulations, more work needs to be done to see how robust these approaches are to the type of nonlinearity used.

Our analysis opens the door to many interesting future studies. These include extensions to more complicated tasks and models such as deeper autoencoders or more general feedforward networks trained on more sophisticated tasks. It would also be

valuable to determine if the structure can be recovered when exact synapse values are not known, when sparsity constraints on the weights are applied, or when different nonlinearities are used in the model. Brain circuits contain both deep hierarchy and recurrent connections, and it remains to be seen if our methods will be successful in artificial networks that have these complexities. In extending to data from the brain, persistent homology techniques could potentially be combined with nonlinear dimensionality reduction techniques to help deal with inaccuracies in the data.

The observation that using nonlinear – as opposed to linear – dimensionality reduction methods is important for extracting structure in the weights, and that regularization during training also encourages this structure to emerge, can guide efforts to investigate more complex models. In general, network models fail to be identifiable, exemplified by the arbitrary invertible matrix A in our model. In the same way, in deep linear networks arbitrary invertible linear transformations of one layer's weights can be undone by the inverse transformation applied to the next layer's weights. When it comes to analyzing neural networks (be it the connectivity or unit activities), it is important to work out the most natural constraints that result in *meaningful* and *interpretable* network structures. Here we've shown that the solutions enforced by Frobenius norm regularization [17, 18] are sufficiently constrained to yield latent variable information. This regularization can be viewed as a cost on weight resources, a biologically relevant constraining factor. This indicates that biological connectivity data may indeed be constrained such that they yield information about neural circuit function via dimensionality reduction methods like those explored here. In addition, we've found that this regularization is not always necessary for extracting the weights when training with SGD (data not shown). This may be because, with the right initialization, the solutions found by SGD are biased to have relatively low Frobenius norm. It is still an open question as to if L2 regularization or other constraints are sufficient for enforcing interpretability in broader classes of network models (see [14, 26, 27] for works related to these issues).

6 Acknowledgements

MF is funded by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1256082. ESB acknowledges the support of NSF DMS Grant 1514743. We thank the Allen Institute for Brain Science founders, Paul and Jody Allen, for their vision, encouragement, and support.

References

[1] C. Shan Xu, Michal Januszewski, Zhiyuan Lu, Shin-ya Takemura, Kenneth J. Hayworth, Gary Huang, Kazunori Shinomiya, Jeremy Maitin-Shepard, David Ackerman, Stuart Berg, Tim Blakely, John Bogovic, Jody Clements, Tom Dolafi, Philip Hubbard, Dagmar Kainmueller, William Katz, Takashi Kawase, Khaled A. Khairy, Laramie Leavitt, Peter H. Li, Larry Lindsey, Nicole Neubarth, Donald J. Olbris, Hideo Otsuna, Eric T. Troutman, Lowell Umayam, Ting Zhao, Masayoshi Ito, Jens Goldammer, Tanya Wolff, Robert Svirskas, Philipp Schlegel, Erika R. Neace, Christopher J. Knecht, Chelsea X. Alvarado, Dennis A. Bailey, Samantha Ballinger, Jolanta A Borycz, Brandon S. Canino, Natasha Cheatham, Michael Cook, Marisa Dreher, Octave Duclos, Bryon Eubanks, Kelli Fairbanks, Samantha Finley, Nora Forknall, Audrey Francis, Gary Patrick Hopkins, Emily M. Joyce, SungJin Kim, Nicole A. Kirk, Julie Kovalyak, Shirley A. Lauchie, Alanna Lohff, Charli Maldonado, Emily A. Manley, Sari McLin, Caroline Mooney, Miatta Ndama, Omotara Ogundeyi, Nneoma Okeoma, Christopher Ordish, Nicholas Padilla, Christopher Patrick, Tyler Paterson, Elliott E. Phillips, Emily M. Phillips, Neha Rampally, Caitlin Ribeiro, Madelaine K Robertson, Jon Thomson Rymer, Sean M. Ryan, Megan Sammons, Anne K. Scott, Ashley L. Scott, Aya Shinomiya, Claire Smith, Kelsey Smith, Natalie L. Smith, Margaret A. Sobeski, Alia Suleiman, Jackie Swift, Satoko Takemura, Iris Talebi, Dorota Tarnogorska, Emily Tenshaw, Temour Tokhi, John J. Walsh, Tansy Yang, Jane Anne Horne, Feng Li, Ruchi Parekh, Patricia K. Rivlin, Vivek Jayaraman, Kei Ito, Stephan Saalfeld, Reed George, Ian Meinertzhagen, Gerald M. Rubin, Harald F. Hess, Louis K. Scheffer, Viren Jain, and Stephen M. Plaza. A connectome of the adult drosophila central brain. *bioRxiv*, 2020.

[2] The Quest to Unravel The Connectome. https://alleninstitute.org/what-we-do/brain-science/news-press/articles/quest-unravel-connectome, Jan. 25, 2018, 4 p.m.

[3] D. J. Felleman and D. C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex (New York, N.Y.: 1991)*, 1(1):1–47, 1991 Jan-Feb.

[4] Rodney J. Douglas, Kevan A. C. Martin, and David Whitteridge. A canonical microcircuit for neocortex. *Neural Comput.*, 1(4):480–488, December 1989.

[5] Rodney J. Douglas and Kevan A. C. Martin. Neuronal circuits of the neocortex. *Annual Review of Neuroscience*, 27:419–451, 2004.

[6] Ashok Litwin-Kumar and Srinivas C Turaga. Constraining computational models using electron microscopy wiring diagrams. *Current Opinion in Neurobiology*, 58:94–100, October 2019.

[7] Jinseop S. Kim, Matthew J. Greene, Aleksandar Zlateski, Kisuk Lee, Mark Richardson, Srinivas C. Turaga, Michael Purcaro, Matthew Balkam, Amy Robinson, Bardia F. Behabadi, Michael Campos, Winfried Denk, and H. Sebastian Se-

ung. Space–time wiring specificity supports direction selectivity in the retina. *Nature*, 509(7500):331–336, May 2014.

[8] Shin-ya Takemura, Yoshinori Aso, Toshihide Hige, Allan Wong, Zhiyuan Lu, C Shan Xu, Patricia K Rivlin, Harald Hess, Ting Zhao, Toufiq Parag, et al. A connectome of a learning and memory center in the adult drosophila brain. *Elife*, 6:e26975, 2017.

[9] Sophie J. C. Caron, Vanessa Ruta, L. F. Abbott, and Richard Axel. Random convergence of olfactory inputs in the Drosophila mushroom body. *Nature*, 497(7447):113–117, May 2013.

[10] Mala Murthy, Ila Fiete, and Gilles Laurent. Testing Odor Response Stereotypy in the Drosophila Mushroom Body. *Neuron*, 59(6):1009 – 1023, 2008.

[11] Ashok Litwin-Kumar, Kameron Decker Harris, Richard Axel, Haim Sompolinsky, and L. F. Abbott. Optimal Degrees of Synaptic Connectivity. *Neuron*, 93(5):1153–1164.e7, March 2017.

[12] Li Zhaoping. Theoretical understanding of the early visual processes by data compression and data selection. *Network: Computation in Neural Systems*, 17(4):301–334, January 2006.

[13] Arya A. Pourzanjani, Richard M. Jiang, and Linda R. Petzold. Improving the identifiability of neural networks for bayesian inference. 2017.

[14] Andrzej Banburski, Qianli Liao, Brando Miranda, Lorenzo Rosasco, Jack Hidary, and Tomaso Poggio. Theory III: Dynamics and Generalization in Deep Networks – a simple solution. *arXiv:1903.04991 [cs, stat]*, July 2019. arXiv: 1903.04991.

[15] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, December 2000.

[16] H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4):291–294, September 1988.

[17] Daniel Kunin, Jonathan M. Bloom, Aleksandrina Goeva, and Cotton Seed. Loss Landscapes of Regularized Linear Autoencoders. *arXiv:1901.08168 [cs, stat]*, May 2019.

[18] Madeleine Udell, Corinne Horn, Reza Zadeh, and Stephen Boyd. Generalized Low Rank Models. *Foundations and Trends® in Machine Learning*, 9(1):1–118, 2016.

[19] Philip Davis. *Circulant Matrices*. Pure and Applied Mathematics. New York : Wiley, 1979.

[20] Albrecht Böttcher, Sergei M. Grudsky, and Egor A. Maksimenko. *On the Structure of the Eigenvectors of Large Hermitian Toeplitz Band Matrices*, pages 15–36. Springer Basel, Basel, 2010.

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

[22] Johannes D. Seelig and Vivek Jayaraman. Neural dynamics for landmark orientation and angular path integration. *Nature*, 521(7551):186–191, May 2015.

[23] Daniel B. Turner-Evans, Kristopher T. Jensen, Saba Ali, Tyler Paterson, Arlo Sheridan, Robert P. Ray, Tanya Wolff, Scott Lauritzen, Gerald M. Rubin, Davi Bock, and Vivek Jayaraman. The neuroanatomical ultrastructure and function of a biological ring attractor. *bioRxiv*, 2020.

[24] Hannah Choi and Stefan Mihalas. Synchronization dependent on spatial structures of a mesoscopic whole-brain network. *PLOS Computational Biology*, 15(4):e1006978, April 2019.

[25] Stefano Recanatesi, Gabriel Koch Ocker, Michael A. Buice, and Eric Shea-Brown. Dimensionality in recurrent spiking networks: Global trends in activity and local origins in connectivity. *PLOS Computational Biology*, 15(7):e1006446, July 2019.

[26] Arya A Pourzanjani, Richard M Jiang, and Linda R Petzold. Improving the Identifiability of Neural Networks for Bayesian Inference. *NeurIPS*, Second workshop on Bayesian Deep Learning, 2017.

[27] Ilyes Khemakhem, Diederik P. Kingma, Ricardo Pio Monti, and Aapo Hyvärinen. Variational Autoencoders and Nonlinear ICA: A Unifying Framework. *arXiv:1907.04809 [cs, stat]*, October 2019.