Common population codes produce extremely nonlinear neural manifolds

Anandita De[1,2] and Rishidev Chaudhuri[1,3,4]

[1]Center for Neuroscience, University of California, Davis
[2]Department of Physics, University of California, Davis
[3]Department of Neurobiology, Physiology and Behavior, University of California, Davis
[4]Department of Mathematics, University of California, Davis

Abstract

Populations of neurons represent sensory, motor and cognitive variables via patterns of activity distributed across the population. The size of the population used to encode a variable is typically much greater than the dimension of the variable itself, and thus the corresponding neural population activity occupies lower-dimensional subsets of the full set of possible activity states. Given population activity data with such lower-dimensional structure, a fundamental question asks how close the low-dimensional data lies to a linear subspace. The linearity or non-linearity of the low-dimensional structure reflects important computational features of the encoding, such as robustness and generalizability. Moreover, identifying such linear structure underlies common data analysis methods such as Principal Component Analysis. Here we show that for data drawn from many common population codes the resulting point clouds and manifolds are exceedingly nonlinear, with the dimension of the best-fitting linear subspace growing at least exponentially with the true dimension of the data. Consequently, linear methods like Principal Component Analysis fail dramatically at identifying the true underlying structure, even in the limit of arbitrarily many data points and no noise.

1 Introduction

Neural coding is distributed and redundant, with large populations of neurons collectively encoding relevant variables. Geometric frameworks provide a natural setting within which to formulate and test theories of population coding, along with tools that allow population structure to be extracted from data [1–10]. In one particularly fruitful approach, data from a population of N neurons can be embedded in an N-dimensional space, with each axis corresponding to the activity of 1 neuron. The state of the population at each moment in time corresponds to a point in this N-dimensional space. Shared structure in the neural population code then corresponds to lower-dimensional shapes or, in the case of smooth responses, "manifolds" on which the data lies [11–20]. Computation can be understood in terms of trajectories on these low-dimensional manifolds [21–30].

Given some population data with such lower-dimensional structure, a fundamental question asks how close the data lies to a low-dimensional linear subspace or hyperplane (i.e., is the lower-dimensional structure near-linear?). This question is of theoretical interest because the linearity or non-linearity of the population data provides insight into the structure, robustness and generalizability of the encoding [9, 11, 31]. The linearity of data is also of great practical importance because methods that seek to fit a linear subspace to data, such as Principal Component Analysis

(PCA) and Factor Analysis, are extremely widely used, whether to reveal structure in an unsupervised manner or as an initial data processing step before using regression and other supervised methods [1, 32]. Linear dimensionality reduction methods have a number of appealing features, including ease of interpretation, computational tractability, theoretical guarantees, and robustness. Moreover, linear methods are the foundation of a number of more advanced methods. For example, if a manifold is not well-fit by a linear subspace, a natural generalization is to seek a set of linear subspaces that combine to describe the manifold [33, 34]. On the other hand, using linear tools on highly non-linear manifolds will be misleading. Thus, understanding when a manifold is linear or near-linear provides insight both into the coding strategy being used by the corresponding brain region and determines the particular data analysis tools that can be used.

In this study, we examine the linearity of the manifolds generated by common population codes. We show that the resulting manifolds are exceptionally nonlinear. For example, consider a population of neurons with Gaussian tuning to a stimulus with D features—each neuron shows maximum response at some preferred stimulus value and the response decreases as a Gaussian function of the distance between the current stimulus and the maximally preferred stimulus value. Since there are D independent dimensions of variation, the neural population responses at any moment in time can be represented by a D-dimensional vector and the data is contained within a D-dimensional manifold. We prove, however, that a linear subspace that contains 80% (or any other fixed fraction) of the variance in these data must have dimension that grows faster than exponentially with D. This dimension can be in the many thousands even for small values of D. Thus, methods that seek to fit a linear subspace to data will greatly overestimate the dimension of the true manifold, even in the limit of arbitrarily many data points and neurons.

2 Results

2.1 Setup

Low-dimensional population structure and neural manifolds Given activity data from a population of N neurons over time, consider the population activity vector $\boldsymbol{y}(t)$, whose n-th entry $y_n(t)$ is the activity (e.g., number of spikes fired or fluorescence signal) of neuron n in a time window of size δt centered at time t. This population activity vector can be seen as a point in an N-dimensional space, with each dimension of the space corresponding to the activity of one of the recorded neurons (see schematic in Fig. 1). If the population activity is measured at T time points $t_1, \cdots, t_T$, then the recording yields T such population activity vectors. The population activity vectors together form the $N \times T$ data matrix A, where A_{ns} is the activity of neuron n in time bin t_s.

The geometric picture corresponding to this collection of population activity vectors is a cloud of points in N-dimensional space, with each point corresponding to a moment in time (Fig. 1b). If the population of neurons shows structured activity, then the points will cluster around particular locations in the N-dimensional space or trace out particular shapes. These shapes provide ways to discover and to reason about the nature of the underlying representation or computation.

In particular, assume that the responses of the population of N neurons above are driven by some time-varying D-dimensional latent variable $\boldsymbol{x}(t)$, meaning that $\boldsymbol{y}(t) = \boldsymbol{F}(\boldsymbol{x}(t)) + \boldsymbol{\xi}(t)$ for some function $\boldsymbol{F} : \mathbb{R}^D \to \mathbb{R}^N$, and where $\boldsymbol{\xi}(t)$ is residual variance (e.g., noise). Here $\boldsymbol{x}(t)$ could be an external stimulus, attentional or arousal or satiety state, motor plan, decision variable, internal estimate of location, or any combination of such and other variables. In what follows, we will adopt the terminology of Jazayeri & Ostojic [9] and refer to this stimulus or other population variable as the "latent variable".

Note that, ignoring noise, the location of any population activity vector in the N-dimensional space can be specified by at most D coordinates (i.e., the values of the latent variable $\boldsymbol{x}(t)$) and thus the data point cloud lies in a D-dimensional space. We will refer to D as the "intrinsic" dimension of the data, following previous work [9]. Under some mild smoothness conditions on $\boldsymbol{F}$ and $\boldsymbol{x}$, the data lie on a D-dimensional manifold and we will thus refer to data "manifolds", following common practice in the field (but the results do not require continuity and smoothness).

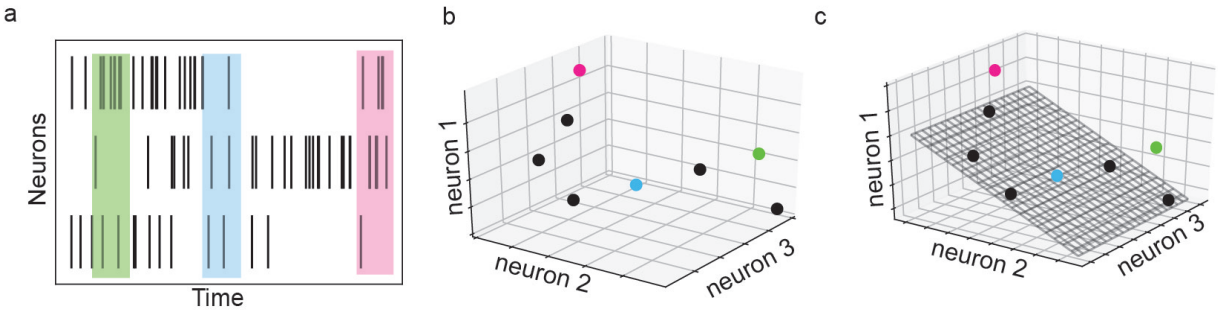

Figure 1: **Schematic of low-dimensional structure in neural population data.** (a) Spiking activity of 3 neurons over time. Shaded regions show three sample time bins, each used to compute an activity vector. (b) Activity represented as a collection of points in 3-dimensional space. Colored points correspond to shaded regions in panel (a). (c) Lower-dimensional linear structure in data, shown as a 2-dimensional plane chosen to capture as much variance in the data as possible. Scatter of points (e.g., pink and green points) off of plane reflects variance that is not captured.

Populations with shared tuning curves In the setting above, the response of the nth neuron is determined by F_n, the nth component of $\boldsymbol{F}$. F_n thus captures the tuning of the nth neuron to the latent variable $\boldsymbol{x}$. In many neural populations, these tuning curves take a similar shape or functional form across neurons but differ in their preferred stimulus, width of selectivity, or other parameters (examples in Figs. 2a, 3a, 4a). Such shared tuning curve structure is common in topographically organized sensory regions [35–40] and in populations that show spatial tuning, such as place, grid and head direction cells [41–43]; it has also been found for more abstract quantities, such as among neurons tuned to numbers [44, 45] and to decision variables such as accumulated evidence [18, 46].

If tuning curves take a similar functional form, the activity of the nth neuron at time t can be modeled by $y_n(t) = f(\boldsymbol{x}(t), \boldsymbol{\alpha}_n)$, with $n = 1, \ldots, N$. Here f is a tuning curve function representing the shared shape of the tuning curve, $\boldsymbol{\alpha}_n$ is a vector of tuning curve parameters (e.g., preferred stimulus or tuning curve center of a sensory neuron, phase and period of a grid cell, preferred value of a decision variable, etc.), and, as before, $\boldsymbol{x}$ is the time-varying D-dimensional latent variable that underlies the population responses. Note that while such tuning curves were historically applied to describe the relationship between neural activity and an external variable, the latent variable can also be an internal cognitive variable [18, 46–51] or even an abstract statistical construct [13, 16, 52] that captures network interactions. For example, in a ring attractor network that encodes heading direction, the latent variable is the network's estimate of heading direction, and the tuning of each neuron emerges from recurrent network interactions [42, 53, 54]. Thus the model is quite flexible and captures a variety of population codes.

Linearity of point clouds and manifolds The most natural structure to seek in population data is linearity, corresponding to finding a lower-dimensional subspace (hyperplane) that contains

the data (see schematic in Fig. 1c). If the dimension of this subspace is L, linear structure corresponds to finding L vectors $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_L$ whose weighted sums account for the data. That is, any data point $\boldsymbol{y}(t) = \sum_{l=1}^{L} a_l(t)\boldsymbol{v}_l$, where $a_l(t)$ is the time-varying contribution of the l-th vector. Equivalently, the rank of the data matrix A is L.

In the presence of noise, data points will not lie exactly in a lower-dimensional linear subspace. Even in the absence of noise, a set of data points may not lie exactly in a linear subspace but might be close enough to be approximated by a linear subspace for practical purposes. Thus, it is typical to look for a linear subspace that captures most of the spread in the data while allowing for some scatter in the data around the subspace (Fig. 1c). Equivalently, one looks for L basis patterns that can approximately sum to any population activity vector $\left(\boldsymbol{y}(t) \approx \sum_{l=1}^{L} a_l \boldsymbol{v}_l(t)\right)$ or for a rank L matrix A_L such that $||A - A_L||$ is small in some appropriate norm (usually 2-norm or Frobenius norm).

More precisely, we define the $(1-\epsilon)$-linear dimension $L_{1-\epsilon}$ of a matrix A to be the smallest R such that there exists a rank R matrix A_R for which $||A - A_R||_F^2 < \epsilon ||A||_F^2$ (this quantity is related to the ϵ-rank of A [55]). This definition of linear dimension corresponds to common practice in neural data analysis, where it is typical to perform PCA and estimate the dimension of data as the number of principal components required to explain some high fraction (i.e., $1-\epsilon$ in our notation) of the variance [1, 9, 56]. Thus, for example, what we call the 0.8-linear dimension is the number of principal components required to account for 80% of the variance in the data.

The best rank R approximation (or best fitting R-dimensional linear subspace) to the data matrix A is given by $\sum_{k=1}^{R} \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^T$, where σ_k is the k-th singular value of A, and $\boldsymbol{u}_k$, $\boldsymbol{v}_k$ are the kth left and right singular vectors respectively. In this case the remaining variance $||A - A_R||_F^2 = \sum_{k=R+1}^{N} \sigma_k^2$. In other words, the matrix A has $(1-\epsilon)$-linear-dimension $L_{1-\epsilon}$ if

$$\sum_{k=1}^{L_\epsilon} \sigma_k^2 / \sum_{k=1}^{N} \sigma_k^2 \geq 1-\epsilon \quad \text{but} \quad \sum_{k=1}^{L_\epsilon - 1} \sigma_k^2 / \sum_{k=1}^{N} \sigma_k^2 < 1-\epsilon \tag{1}$$

The singular values of A can also be calculated from the eigenvalues of the (non-mean-subtracted) covariance matrix AA^T, which is the matrix of covariances between neurons averaged over time, or $A^T A$, which is the matrix of covariances between data points averaged over neurons. The k-th eigenvalue of each of these matrices is $\lambda_k = \sigma_k^2$ (for $k \leq N$, assuming more time points than neurons).

Constructing such a low rank approximation to the data matrix (or, equivalently, fitting a linear subspace to the data point cloud) is the foundation of commonly used dimensionality reduction methods such as PCA and Factor Analysis. Moreover, a number of nonlinear dimensionality reduction techniques rely on approximating the data point cloud or manifold by a family of linear subspaces [33, 34, 57]. Such methods will be expected to perform well when the data point cloud or manifold is near-linear and poorly when the data manifold is highly non-linear.

Overview of approach In this study, we consider a population of N neurons whose activity is driven by a D-dimensional real-valued latent variable $\boldsymbol{x}$ with firing rates given by tuning curve functions $f(\boldsymbol{x}, \boldsymbol{\alpha}_n)$. Thus, the intrinsic dimension of neural activity is D. For several choices of tuning curve function we lower bound the $(1-\epsilon)$-linear dimension of the neural activity (equivalently the number of principal components required to explain a $(1-\epsilon)$ fraction of the data variance) and show that it is very large, growing at least exponentially with the intrinsic dimension D.

We assume that $\boldsymbol{x}$ takes on all possible values in a compact subset of $\mathbb{R}^D$ and that f is continuous and one-to-one, so that in the absence of noise population responses lie on a D-dimensional manifold.

However, the approach can be naturally generalized to finding a linear subspace that contains point cloud data instead instead, and thus extends to cases like non-continuous values of the latent variable.

We consider firing rates and ignore noise so that the response of the nth neuron is given exactly by the mean firing rate, $y_n(t) = f(\boldsymbol{x}, \boldsymbol{\alpha}_n)$, and in this case the time window around t in which the rate is measured does not affect the results (as long as it is small on the timescale at which $\boldsymbol{x}$ changes). In the absence of noise, PCA and Factor Analysis are equivalent and our results thus apply to both methods (as well as to methods like Probabilistic PCA). Given that our results lower bound the linear dimension, including noise would simply strengthen our results by making neural activity more high-dimensional. Thus, our results reflect fundamental lower bounds on the dimensionality of neural activity rather than a lack of data and would not change if neural responses were averaged over multiple stimulus presentations. We consider tuning curve functions $f(\boldsymbol{x}, \boldsymbol{\alpha}_n)$ with certain symmetries and use these symmetries to exactly or approximately calculate the eigenvalues of the neuron-neuron covariance matrix (these eigenvalues are also the squared singular values of the data matrix and, when normalized, are the fractions of variance explained by the different principal components). We then count the number of eigenvalues needed to account for a $(1 - \epsilon)$ fraction of the variance in activity, for some small ϵ. Our results are not sensitive to the choice of ϵ and in general apply when $\epsilon < 0.5$.

To define the covariance matrix and calculate its eigenvalues, it will be convenient to first define the correlation profile function c between neurons with tuning parameters $\boldsymbol{\alpha_m}$ and $\boldsymbol{\alpha_n}$ to be $c(\boldsymbol{\alpha_m}, \boldsymbol{\alpha_n}) = \mathbb{E}_{\boldsymbol{x}}\left[f(\boldsymbol{x}(t), \boldsymbol{\alpha}_m) f(\boldsymbol{x}(t), \boldsymbol{\alpha}_n)\right]$, where the expectation is taken over the values of the latent variable $\boldsymbol{x}$. This is simply the (non-mean-subtracted) covariance between the neurons. For a population of N neurons, the $N \times N$ covariance matrix, C, has (m, n)th entry $C_{mn} = c(\boldsymbol{\alpha_m}, \boldsymbol{\alpha_n})$.

Note that we consider the non-mean-subtracted covariance matrix except where otherwise indicated. Methods such as PCA often first subtract the mean from the data. In Methods, Section 4.2 we show that if $L_{1-\epsilon}$ is the linear dimension of the non-mean-subtracted data, then $L_{1-\epsilon} - 1$ is a lower bound on the linear dimension of the mean-subtracted data (this is a consequence of the Weyl inequalities relating the eigenvalues of perturbed matrices). Thus, our lower bounds on linear dimension for non-mean-subtracted data are easily transformed into lower bounds on mean-subtracted data, and in particular scaling arguments hold.

2.2 Translation-invariant tuning

In many brain regions, responses to the latent variable are given by tuning curves whose shape is the same across neurons but with the tuning curve shifted or centered around a different region of latent variable space for each neuron [35–43]. Examples of such translation-invariant tuning range from early sensory systems, such as orientation tuning in area V1 [37, 58], to cognitive systems, such as spatial tuning in hippocampal place cells and entorhinal grid cells [41, 43, 47, 48] and tuning to abstract variables such as number [44, 45].

In this setting the response of neuron n at time t is determined by the difference between the current value of the latent variable, $\boldsymbol{x}(t)$, and the neuron's preferred value $\boldsymbol{\alpha_n}$ (note that for convenience we refer to the "center" or "preferred stimulus" of the tuning curve but the parameter $\boldsymbol{\alpha_n}$ more generally can simply index the shift of the tuning curve from an arbitrarily chosen reference tuning curve function as, for example, with the phase of a grid cell). If $\boldsymbol{x} \in \mathbb{R}^D$ is the latent D-dimensional variable, then the tuning curve parameter $\boldsymbol{\alpha}$ is also in $\mathbb{R}^D$, and the tuning curve function $f(\boldsymbol{x}, \boldsymbol{\alpha}) = g(\boldsymbol{x} - \boldsymbol{\alpha})$ for some function g (equivalently, given some $\boldsymbol{\delta} \in \mathbb{R}^D$, $f(\boldsymbol{x}, \boldsymbol{\alpha}) = f(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{\alpha} + \boldsymbol{\delta})$). Different neurons have different preferred values, with the preferred values tiling the space of possible values.

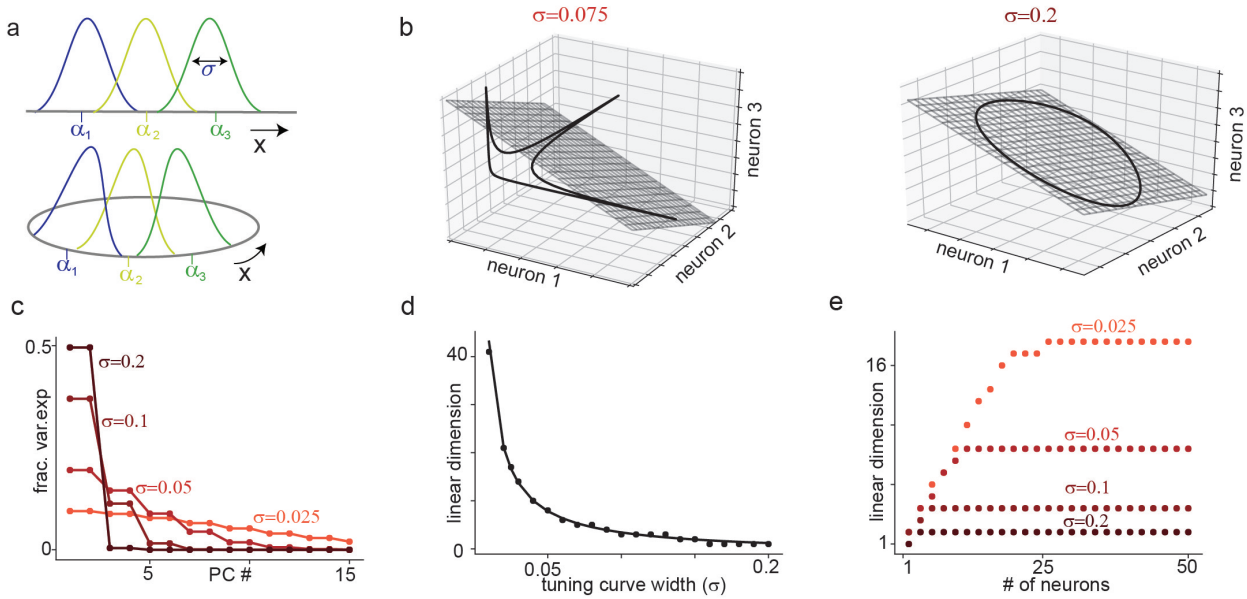

Figure 2: **Translation-invariant tuning to a one-dimensional variable and the inverse relationship between linear dimension and sparsity.** (a) Gaussian tuning curves of 3 neurons encoding a non-circular (top) or circular (bottom) scalar stimulus variable. (b) Black line: Manifold formed by population activity of 3 neurons with Gaussian tuning to a 1-dimensional circular variable. Each axis shows the activity of 1 neuron. Gray: Best fitting 2D linear subspace (i.e., plane spanned by first two principal components). Left and right show an example of narrow ($\sigma = 0.075$) and broad ($\sigma = 0.2$) tuning respectively . For (c)–(e), results shown are for Gaussian tuning to a circular variable, with uniformly spaced tuning curve centers. Circles show numerical simulations and lines show theoretical predictions. (c) Fraction of variance explained by each principal component (equivalently, eigenvalues of covariance matrix) for a population of $N = 50$ neurons. Different curves show different tuning curve widths. (d) Linear dimension of neural data against tuning curve widths, showing that linear dimension grows as $1/\sigma$. (e) Linear dimension against number of neurons in a population for each tuning curve width, showing initial linear growth before saturation at the predicted values shown in (d).

One-dimensional translation invariant populations and the sparsity-linear dimension uncertainty relation Consider a translation-invariant population of N neurons where the encoded variable x and the tuning parameter for each neuron α_n are drawn from a one-dimensional circular space (i.e, $\mathbb{S}^1$), with points in the space parameterized by $[0, 1)$, Fig. 2a. For example, x could be the angle of orientation of a bar, the direction of motion of a stimulus, or the head direction of a moving animal; correspondingly, α_n could be the center of the tuning curve of the nth neuron. If x evenly samples the space, then the correlation profile c for two neurons depends only on the the difference $\alpha_m - \alpha_n$ between tuning parameters. With slight abuse of notation we will thus write the correlation profile as $c(\alpha_m, \alpha_n) = c(\delta)$ where $\delta = \alpha_m - \alpha_n$ and the function c is periodic with period 1.

If the tuning parameters evenly tile the space, then the entries of the covariance matrix are

$$C_{mn} = c(\alpha_m - \alpha_n) = c((m - n)/N) \tag{2}$$

This covariance matrix C is circulant, meaning that each row is a shifted copy of the row above. It is well known (and easy to show, see Section 4.4) that the eigenvalues of C are given by

the Fourier transform of c, the function used to generate each row [4, 31, 59–61]. Thus, the pth eigenvalue of C is

$$\lambda_p = \sum_{l=0}^{N-1} c(l/N)e^{-2\pi ipl/N} \approx N \int_0^1 c(x)e^{-2\pi ipx}dx, \tag{3}$$

where $i = \sqrt{-1}$.[1]

If the tuning curve centers are not evenly spaced, the corresponding matrix is no longer circulant, but the eigenvalues of the matrix C are still approximately given by the Fourier transform of c. If tuning curve centers are randomly sampled, this approximation converges rapidly as the number of neurons increases. Similarly, if the stimulus space is one-dimensional but not circular the corresponding matrix is Toeplitz and the same eigenvalue relationship approximately holds [59, 62–65]. Thus, quite generally, the profile of eigenvalues is given by the Fourier transform of the correlation profile.

As a specific example, consider a population with translation-invariant Gaussian tuning, which is a common model for tuning curves across multiple systems like orientation selective neurons in visual cortex V1 [66] and place cells in the hippocampus [41]. In this case the response of the nth neuron is modeled by

$$y_n(t) = K_1 \exp\left(-\frac{(x(t)-\alpha_n)^2}{2\sigma^2}\right), \tag{4}$$

where K_1 is the maximum firing rate and σ is the width of the tuning curve. The covariance between the mth and nth neurons is

$$C_{mn} = K_2 \exp\left(-\frac{\delta_{mn}^2}{4\sigma^2}\right), \tag{5}$$

where K_2 is a constant and δ_{mn} is the difference between the tuning curve centers (calculated accounting for the circular boundary conditions). The eigenvalues of the covariance matrix are given by

$$\lambda_p = K_3 e^{-4\pi^2\sigma^2 p^2}. \tag{6}$$

where K_3 is a constant, as shown in Fig. 2C (see Section 4.6 for the calculation). Thus, the eigenvalue profile is Gaussian with variance inversely proportional to the width of the tuning curve.

The $(1-\epsilon)$-linear dimension is the smallest $L_{1-\epsilon}$ such that $\sum_{p=0}^{L_{1-\epsilon}} \lambda_p \geq (1-\epsilon)\sum_{p=0}^{N} \lambda_p$. If N is not too small, these sums can be approximated by Gaussian integrals yielding the condition

$$L_{1-\epsilon} = \frac{1}{\sigma}\frac{\operatorname{erf}^{-1}(1-\epsilon)}{\pi}. \tag{7}$$

Thus, to explain a constant fraction of the variance, the linear dimension generically grows as $1/\sigma$, where σ is the tuning curve width. In particular, $L_{0.95}$ corresponds to a 95% confidence interval for a Gaussian distribution and is thus $\frac{1.96}{\sqrt{2}\,\pi\sigma}$.

More generally, so-called uncertainty principles relate the spread, sparsity, entropy or concentration of a function to that of its Fourier transform [67–70]. These principles imply that if the tuning curves (and hence the correlation profile) are sparse, fall off rapidly around their preferred values, or are concentrated on relatively small subsets of latent variable space then λ will decay slowly with increasing p and have many significant non-zero entries. If λ decays slowly, then the

[1] Note that $i = \sqrt{-1}$ throughout this study and is not used as a matrix index.

number of eigenvalues needed to capture most of the variance will be high. Consequently, the linear dimension will be large and the manifold will be highly nonlinear.

Gaussian functions provide a lower bound for several of these uncertainty principles, and thus the $1/\sigma$ scaling relationship for Gaussians will provide a lower bound across a wide class of tuning curve shapes, in particular those with firing localized to some region of latent space (or the manifold). However, highly nonlinear manifolds are to be expected whenever the tuning curves (and hence the covariance profile) are concentrated on comparatively small subsets even if these tuning curves are not localized to a single interval or region.

We highlight two particularly useful uncertainty principles that apply in this more general setting. First, if the covariance profile c has K nonzero entries (i.e., it is K-sparse), then the eigenvalue profile has at least N/K nonzero entries [68]. Consequently, if K is small then there are a large number of nonzero eigenvalues. Second, if a fraction $1-\hat{\epsilon}$ of the covariance profile is concentrated on a set S of size K (meaning that $\sum_{\delta\in S}|c(\delta)| > (1-\hat{\epsilon})\sum|c(\delta)|$), then the smallest set that contains $1-\epsilon$ of the eigenvalue mass has size at least $N(1-\hat{\epsilon})(1-\epsilon)/K$ [68, 70]. Note that the size of this set is just the $(1-\epsilon)$-linear dimension and consequently the linear dimension again grows inversely with K. Sparse coding thus generically implies high linear dimension.

In practice, for systems with relatively broad tuning curves and for which the latent variable is low-dimensional, such as head direction cells [54, 71] or ventral hippocampal place cells [72, 73], the overestimate of intrinsic dimension by linear dimension may not be too large. However, in many systems, both sensory and cognitive, neurons respond to only a small fraction of possible values of the latent variable. For example, foveal V1 cells in the primate cover less than a degree of visual space [37], and rodent hippocampal place cells can cover under 1% of the area of large environments [73, 74]. Similarly, cells that are tuned to complex visual stimuli such as faces or other objects tend to show sparse responses [75, 76], thus covering only a small portion of stimulus space. In these settings, the manifold is likely to be highly nonlinear and linear dimensionality will greatly overestimate intrinsic dimensionality.

Multidimensional translation-invariant tuning and supra-exponential growth of linear dimension As with the one-dimensional case, when a higher-dimensional variable $\boldsymbol{x}$ is encoded with translation-invariant tuning curves (schematic in Fig. 3a), the covariance profile is also translation-invariant and the eigenvalues of the covariance matrix (or singular values of the data matrix) are given by the Fourier transform of the covariance profile. Consequently, as in the 1D case, tuning curves that are sharper or concentrated on smaller sets will yield more slowly decaying eigenvalue profiles and hence higher linear dimension. However, the linear dimension will depend strongly on D, the intrinsic dimension of the latent variable. We first examine this interaction in the Gaussian case, before drawing general conclusions.

Consider a population of N neurons with translation-invariant Gaussian tuning to an underlying D-dimensional latent variable $\boldsymbol{x}$ that takes values within $[0,1]$ along each dimension. The tuning curve for the nth neuron is centered at $\boldsymbol{\alpha}_n$. By an appropriate choice of basis for $\boldsymbol{x}$, the covariance matrix of the Gaussian tuning curve can be assumed diagonal. For simplicity we assume that boundary conditions are circular and that $\boldsymbol{x}$ is scaled so that tuning curves have equal width σ in all directions. Thus, the response of the n-th neuron is

$$y_n(t) = K_1 \exp\left(-\frac{||\boldsymbol{x}(t)-\boldsymbol{\alpha_n}||^{\mathbf{2}}}{2\sigma^2}\right), \tag{8}$$

where K_1 is the maximum firing rate. The corresponding correlation profile is also Gaussian. If

tuning curve centers are equally spaced, then the covariance matrix has (m, n)th entry

$$C_{mn} = K_2 \exp\left(-\frac{||\boldsymbol{\delta_{mn}}||^2}{4\sigma^2}\right), \tag{9}$$

where K_2 is a constant and $\boldsymbol{\delta_{mn}}$ is the difference between the tuning curve centers $\boldsymbol{\alpha_m}$ and $\boldsymbol{\alpha_n}$ (accounting for the circular boundary conditions). The eigenvalues are given by (Section 4.7)

$$\lambda_{\boldsymbol{p}} = K_3 \exp\left(-4\pi^2 \sum_{d=1}^{D} p_d^2 \sigma^2\right) = K_3 \exp(-4\pi^2\sigma^2|\boldsymbol{p}|^2). \tag{10}$$

Here K_3 is a constant. The eigenvalues are indexed by a D-dimensional vector $\boldsymbol{p}$ with dth entry $p_d \in [-N_d/2, \cdots, N_d/2]$, where N_d is the number of tuning curve centers along dimension d (assumed the same for simplicity). Note that these eigenvalues are given by a multivariate Gaussian evaluated at the integer lattice points of a D-dimensional cube with side length N_d, Fig. 3b.

The magnitude of an eigenvalue depends only on the magnitude of $\boldsymbol{p}$ and thus the eigenvalues can be ordered by smallest to largest distance from the origin in $\boldsymbol{p}$-space. There will be multiple eigenvalues with the same magnitude, corresponding to the same value of $|\boldsymbol{p}|$. The number of eigenvalues of a given magnitude will increase with distance from the origin. Thus, moving away from the origin there will be more eigenvalues but of smaller magnitude (schematic in Fig. 3b).

When ordered by their magnitude, the eigenvalue profile thus shows a step-like shape, Fig. 3c. A smoothly interpolating function for the eigenvalue profile can be derived by noting that the eigenvalue profile is spherically symmetric in $\boldsymbol{p}$-space and thus the number of eigenvalues of a given magnitude depends on the number of lattice points at the corresponding radius. The number of lattice points can be interpolated by the volume of a shell in D-dimensional space of radius r and width dr, which is given by

$$V_{shell}(r) = \frac{2\pi^{D/2}}{\Gamma(D/2)} r^{D-1}\, dr \tag{11}$$

Thus, the eigenvalues can be interpolated by the function

$$\lambda_p \sim \exp(-(K_4 \sigma p^{1/D})^2) \tag{12}$$

where K_4 takes into account the constants in Eq. (11) (also see [4] for a similar argument) and the scalar p now indexes the eigenvalues from 1 to N (the total number of neurons). The eigenvalues initially decay slowly and then rapidly, showing a transition between a power-law-like and an exponential regime.

To convert these eigenvalue profiles into an estimate of linear dimension, again note that the covariance profile is radially symmetric. Thus, we first seek the smallest radius r such that a fraction $(1-\epsilon)$ of the total eigenvalue mass lies in a sphere of radius r, and then count the number of eigenvalues in that sphere. The scaling of this radius with D can be derived by observing that the probability mass of a D-dimensional Gaussian concentrates in a shell of radius $\sqrt{D}/\sigma$ around the origin, as shown in Fig. 3d (we provide further details and calculate the radius more exactly using a χ-distribution in Section 4.7). Thus, any sphere that captures a significant fraction of the probability mass must grow as $r \sim \sqrt{D}/\sigma$.[2] The number of eigenvalues in a D-dimensional sphere of radius r grows as r^D. Consequently, the linear dimension grows as $(\sqrt{D}/\sigma)^D$, Fig. 3e,f.

[2] Alternatively, note that this problem is closely related to finding a 95% (or any other percent) confidence interval for a D-dimensional Gaussian). While an interval of width 2σ contains 95% of the probability mass in 1D, in higher dimensions an interval of any fixed width contains a shrinking fraction of the total probability mass. Thus the interval must grow with D.

This scaling of linear dimension with intrinsic dimension is extremely rapid, growing faster than exponentially. Consequently, even low-dimensional Gaussian manifolds will have very high linear dimension. Moreover, this rapid growth means that even quite broad tuning curves will yield high linear dimension unless intrinsic dimension is extremely low. For example, as shown in Fig. 3e, a population of neurons with relatively broad Gaussian tuning to an 8 dimensional encoded variable has linear dimension $L_{0.95} > 6 \times 10^5$.

The structure of the argument presented above is quite general, relying on the interaction between the decay of the eigenvalue magnitude with distance from the origin and the growth of volume with radius—while eigenvalues decay rapidly with distance, the growth of volume means that the radius of a sphere that captures some significant fraction of the total mass of eigenvalues must grow with D. The argument should thus extend to other localized tuning curves even if non-Gaussian, and these are likely to show supraexponential scaling.

Note that these results hold in the limit of arbitrarily many data points and recorded neurons. In practice, the observed linear dimension will be limited by the number of neurons observed and the values of $\boldsymbol{x}$ sampled (or the task complexity [56]) but for these manifolds will continue to grow as more neurons and task variable values are measured.

2.3 Multiplicative tuning

We next consider tuning curve models where tuning to the latent variable can be written as a product over one-dimensional or lower-dimensional factors. That is, the tuning curve function is of the form

$$y_n(t) = f(\boldsymbol{x}(t), \boldsymbol{\alpha}_n) = \prod_{d=1}^{D} f_d(x^d(t), \alpha_n^d). \tag{13}$$

Here the superscript denotes the d-th component of a vector, and the f_d's are scalar functions. Note that in general the multiplicative factors can consist of groups of variables and not just one-dimensional factors, in which case the product would be taken over groups rather than single variables.

As an example of such tuning, if f is a multivariate Gaussian then, after an appropriate choice of basis for $\boldsymbol{x}$, f can be written as a product of one-dimensional Gaussians. As another example, common models of attention involve multiplicative gain modulation [77]. In this case the latent variable includes both the current stimulus value and the value of the attentional signal, and the response of a neuron can be written as a product of stimulus tuning and the response to the attentional signal (Fig. 4a (top)). For a third example, the spatio-temporal receptive fields of early visual cells are often decomposed as a product of the spatial part and the temporal part (Fig. 4a (bottom)) [78, 79]. Finally and more generally, multidimensional tuning curves that are not multiplicative may be able to be approximated by a product of lower-dimensional factors in a so-called "mean field" approximation.

Let m and n be two neurons with parameter vectors $\boldsymbol{\alpha_m}$ and $\boldsymbol{\alpha_n}$. If the sampling of the encoded variable is uncorrelated across dimensions and boundary conditions are rectangular, then the covariance between these neurons can be written as

$$\begin{aligned} c(\boldsymbol{\alpha_m}, \boldsymbol{\alpha_n}) &= \int d\boldsymbol{x}\, p(\boldsymbol{x}) f(\boldsymbol{x}, \boldsymbol{\alpha_m}) f(\boldsymbol{x}, \boldsymbol{\alpha_n}) \\ &= \prod \int dx^d\, p(x^d) f_d(x^d, \alpha_m^d) f_d(x^d, \alpha_n^d) \\ &= \prod c^d(\alpha_m^d, \alpha_n^d), \end{aligned} \tag{14}$$

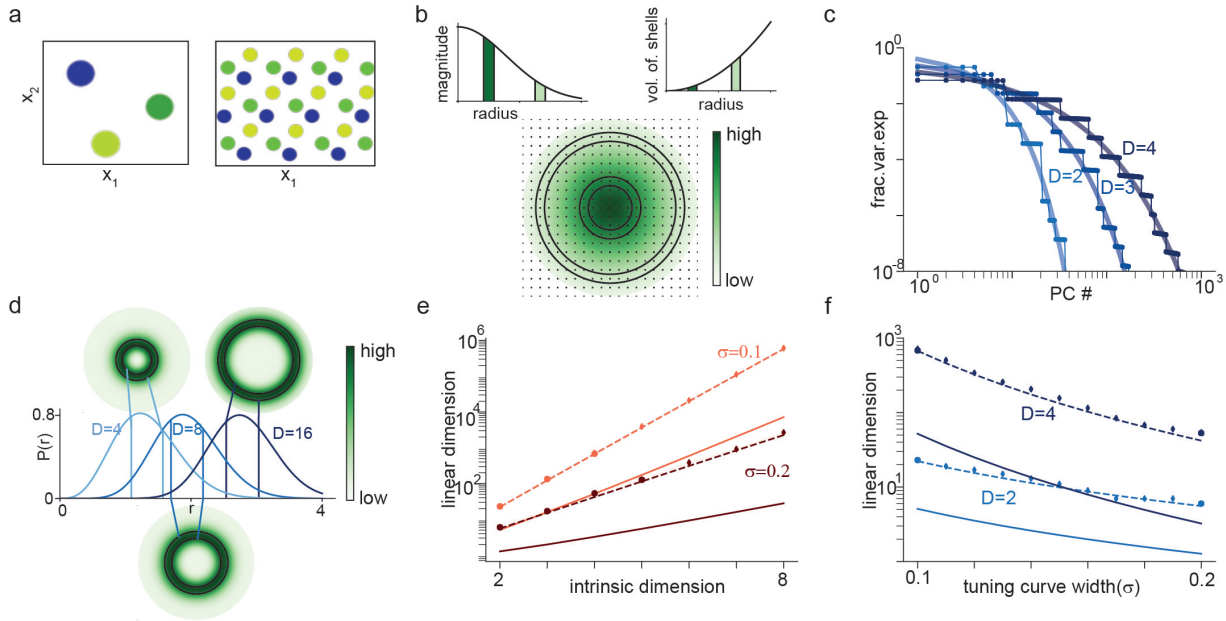

Figure 3: **Translation-invariant tuning to a multi-dimensional variable and supra-exponential growth of linear dimension with intrinsic dimension.** (a) Examples of 2d tuning curves, showing schematics of 3 different place cells with different tuning centers in a square arena (left) and 3 grid cells with the same spacing but different phases (right). (b) For Gaussian tuning curves, eigenvalues of the covariance matrix (variance along each PC) are values of a D-dimensional Gaussian at the lattice points of D-dimensional Fourier space. Each lattice point corresponds to one eigenvalue, and the colormap shows its value. Left inset: Decay of eigenvalues with distance from origin in Fourier space. Right: Number of eigenvalues contained in concentric shells of different radii. Circular shells on plot highlight two sets of eigenvalues, with corresponding magnitude and volume of shell shown as shaded region in insets. For a shell close to the origin the eigenvalues have large magnitude but there are fewer eigenvalues as a consequence of the smaller volume. Away from the origin the value of the eigenvalue is lower but there are more such eigenvalues. This tradeoff between eigenvalue magnitude and the number of eigenvalues of that magnitude explains the shape of the variance explained vs PC number curve. (c) Fraction of variance explained by each PC (or eigenvalues of covariance matrix) for D-dimensional Gaussian tuning curves and periodic boundary conditions along each dimension. Circles show numerical simulations, thin line represents prediction from Fourier transform of covariance matrix rows, and thicker lines represent theoretically-predicted smooth interpolation. (d) Total probability mass at radius r for a D-dimensional Gaussian (i.e., density function of chi distribution), shown for three different values of D. Circular insets show concentric shells colored by total probability mass at that radius. The bulk of the probability mass lies in a shell of radius $\sim \sqrt{D}/\sigma$. Thus, accounting for most of the variance requires considering all eigenvalues within a sphere of radius at least $\sim \sqrt{D}/\sigma$. (e) Semi-log plot of linear dimension ($\epsilon = 0.05$) vs. intrinsic dimension for Gaussian tuning curves with different widths. Circles show numerical results, solid lines show theoretical lower bound (applies whenever $\epsilon \leq 0.5$), and dashed lines show semi-analytic fit using chi distribution. (f) Semi-log plot of linear dimension vs. tuning curve width. Circles and lines as in panel (e).

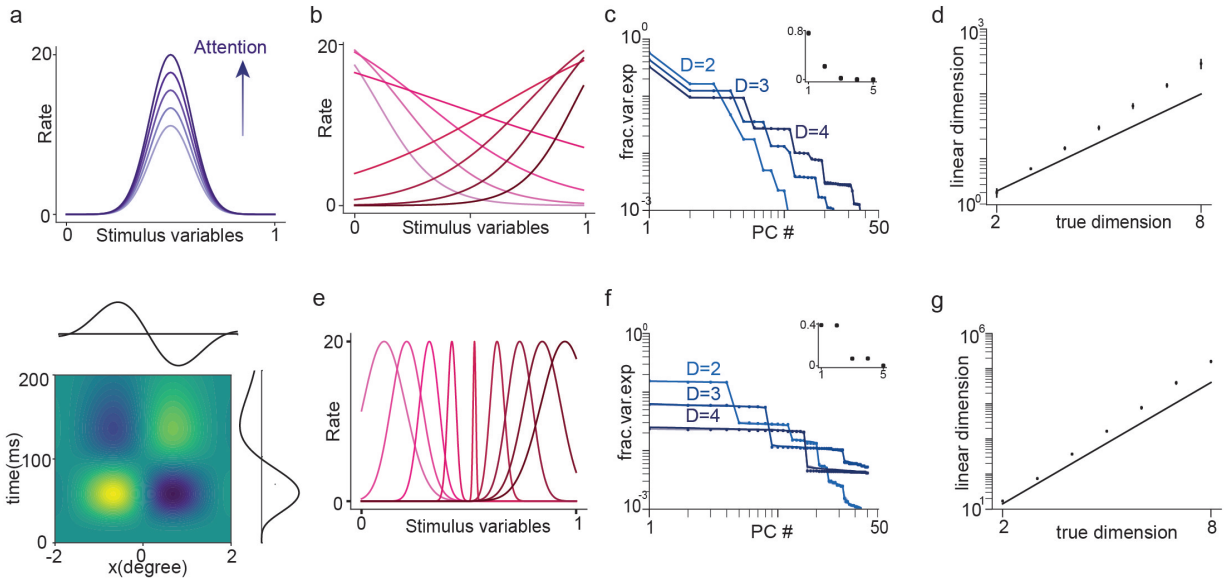

Figure 4: **Multiplicative tuning and exponential growth of linear dimension with intrinsic dimension.** (a) Schematics of common examples of multiplicative tuning. Top: Gain modulation of tuning to a sensory stimulus by attention. Bottom: Separable spatio-temporal receptive field of retinal ganglion cell as product of spatial tuning (horizontal) and temporal tuning (vertical). Panels (b)-(d) show results from a multiplicative tuning model where tuning along each dimension is sigmoidal. (b) Sample tuning along each dimension. Tuning curves are sigmoidal with slopes chosen uniformly in range $[-5, 5]$. (c) Fraction of variance explained vs PC number for the model shown in (b) for different values of intrinsic dimension (D). Circles show numerical simulations and lines show result from tensor product of 1D tuning curves. Inset shows the eigenvalues in the 1D case. (d) Linear dimension against intrinsic dimension for the data in (c). Circles show simulations and solid line shows theoretical lower bound of $2^{D(H-0.05)}$, where H is the entropy of the eigenvalue distribution shown in the inset of panel (c). Panels (e)-(g) show results from a multiplicative tuning model where tuning along each dimension is Gaussian. Gaussians are not translation-invariant and the width of the Gaussian depends on position, with tuning sharpest at the center of the stimulus space (as in visual receptive fields). (e) Sample tuning along each dimension. (f), (g) As in (c), (d) but for the model shown in (e).

where $p(\boldsymbol{x}) = \prod_d p(x^d)$ is the distribution of latent variable values and we have defined $c^d(\alpha^d_m, \alpha^d_n) = \int dx^d p(x^d) f_d(x^d, \alpha^d_m) f_d(x^d, \alpha^d_n)$.

If the tuning curve parameters are chosen to tile the space (note that tuning curves for different parameter values do not need to be shifted copies of each other, unlike in the previous section), then the data covariance matrix can be written as the tensor product of matrices corresponding to individual stimulus dimensions. Thus $C = \otimes_{d=1}^{D} C^d$, where $\otimes$ indicates the tensor product and the matrix C^d has entries $C^d_{mn} = c^d(\alpha^d_m, \alpha^d_n)$ (see Section 4.9 for more details).

Now let $\{\gamma^d_1, \gamma^d_2, ..., \gamma^d_{N_d}\}$ be the eigenvalues of C^d, where N_d is the number of tuning curve parameters sampled along the dth dimension. The eigenvalues of C are all possible products of one eigenvalue from each C^d and so take the form $\prod_d \gamma^d_{p_d}$, where $\gamma^d_{p_d}$ is the p_dth eigenvalue of C^d and each p_d ranges over 0 to $N_d - 1$. In Fig. 4c,f we show examples of eigenvalues from two such architectures.

The linear dimension of multiplicative models grows at least exponentially with intrinsic dimension Consider data from a D-dimensional multiplicative model where the tuning along each dimension has the same shape. Thus each component C^d has the same eigenvalues, which we denote $\{\gamma_1, \gamma_2, ..., \gamma_{N_d}\}$.

To derive the asymptotic scaling of linear dimension with intrinsic dimension, we note that the eigenvalues of each component C^d (when appropriately normalized) can be interpreted as the outcome probabilities of a multinomial random variable. Moreover, the eigenvalues of the covariance matrix itself can be interpreted as the outcome probabilities of D independent draws from this distribution (see Section 4.10 for more details on the argument). Finding the smallest set of eigenvalues that sum to $1-\epsilon$ of the total is thus equivalent to finding the smallest set of outcomes that accounts for $1-\epsilon$ of the total probability mass, sometimes called an ϵ-high-probability set [80]. The $(1-\epsilon)$-linear dimension is the size of this high-probability set.

The asymptotic equipartition property [80] guarantees that asymptotically the high-probability set contains $2^{DH(\boldsymbol{\gamma})}$ outcomes, where $H(\boldsymbol{\gamma}) = -\sum_p \gamma_p \log_2 \gamma_p$ is the Shannon entropy of the eigenvalues of each component (normalized to sum to 1). Moreover, for any $\epsilon' > 0$, $2^{D(H(\boldsymbol{\gamma})-\epsilon')}$ is a lower bound on the size of the high-probability set for large enough D. This scaling is asymptotic, but in practice convergence is rapid and for the example models we consider the asymptotic lower bound captures the scaling even for small intrinsic dimension, as shown in Fig. 4d, g.

For a non-asymptotic lower bound on the linear dimension, assume that only two of the eigenvalues for each factor are nonzero (and thus the overestimate of the intrinsic dimension of 1 by the linear dimension for each multiplicative factor is as small as possible while still being an overestimate). Up to an overall scaling constant (which we can ignore because we are considering eigenvalues as a fraction of the total sum), these eigenvalues can be written as $\{1-\gamma, \gamma\}$, for some $\gamma \leq 0.5$. The eigenvalues of C are again tensor products of $[1-\gamma, \gamma]$ taken D times. In descending order of magnitude, there is 1 eigenvalue of magnitude $(1-\gamma)^D$, $\binom{D}{1}$ eigenvalues of magnitude $(1-\gamma)^{D-1}\gamma$, and so on, with $\binom{D}{k}$ eigenvalues of magnitude $(1-\gamma)^{D-k}\gamma^k$. As a consequence of our normalization, the sum of the eigenvalues $\sum_{k=0}^{D} \binom{D}{k}(1-\gamma)^{D-k}\gamma^k = 1$.

Note that the eigenvalues are the outcome probabilities for a binomial random variable X distributed as $\mathrm{Bin}(D, \gamma)$ (i.e., D trials with success probability γ). Thus, the problem of finding $L_{1-\epsilon}$ is equivalent to first finding the smallest subset of outcomes of a binomial random variable that account for $(1-\epsilon)$ of the probability and then counting the size of this subset.

Using standard bounds on sums of binomial coefficients yields a lower bound on the linear dimension for $\epsilon \geq 0.5$

$$L_{1-\epsilon} \geq \frac{1}{\sqrt{8\rho(1-\rho)}} 2^{(H_b(\rho) - \frac{log_2(D)}{2D})D} \tag{15}$$

where $H_b(\rho) = -\rho \log_2 \rho - (1-\rho)\log_2(1-\rho)$ is the binary entropy function and ρ is lower bounded by $\gamma - (1+\ln(2))/D$ (see Section 4.10 for full argument). Thus, except when D is small enough that $H_b(\rho) < \frac{\log_2(D)}{2D}$ the lower bound grows exponentially with D (with the exponent asymptotically approaching $H_b(\gamma)D$ as desired).

To summarize, for multiplicative tuning the linear dimension grows exponentially with intrinsic dimension, with a scaling constant that approaches the entropy of the eigenvalue distribution for a single factor.

3 Discussion

The relationship between intrinsic and linear (sometimes called "embedding") dimension is emerging as an important means of understanding neural population coding, providing both insight into

fundamental features of information encoding (such as generalizability and the progress of learning) and constraints on the statistical tools that can be used to analyze data [1, 9, 31]. It is widely appreciated that the point clouds and manifolds that emerge from neural population data are often nonlinear [9, 15, 18, 81], and previous work has in particular identified the sparsity of neural population responses as an important factor in this nonlinearity [4, 10, 17, 31]. The key contribution of the present study is to show that the nonlinearity is likely to be much greater than previously believed—for a number of common population codes, the linear dimension grows at least exponentially with the intrinsic dimension of the data. This exponential growth holds even if representations are not sparse, implying that even quite distributed population codes can have extremely high linear dimension. Consequently, dimensionality reduction methods that seek to fit a linear subspace to data, such as Principal Component Analysis and Factor Analysis, will dramatically overestimate the true dimension of data drawn from these population codes. Moreover, the exponential growth of linear dimension means that methods that fit data using a union of linear subspaces will require a large number of subspaces.

We have derived our analytical results for the cases of translation-invariant and multiplicative tuning curves and find exponential or faster growth of linear dimension with intrinsic dimension in both settings. These results are likely to apply more generally to populations of neurons whose responses can be modeled as sparse or localized firing fields on some low-dimensional manifold, even if these firing fields take different shapes across neurons, as well as to populations whose tuning curves can be approximately written as a product of lower-dimensional factors in a so-called "mean-field" or separable approximation.

Our results on the high linear dimension of neural population data reflect the structure of the underlying manifolds or point clouds and do not reflect a lack of data or the presence of noise. In particular, they apply in the limit of large amounts of data and number of sampled neurons and in the absence of noise. For finite amounts of data, the observed linear dimension will be limited by the number of recorded neurons and the complexity of the task or experimental setting [56]. In the presence of noise, the observed linear dimension will be even higher than the noise-free calculations and thus the lower bounds will still hold. Finally, while we choose $90 - 95\%$ variance explained as our criterion to define linear dimension for the figures, none of our results depend on this particular threshold, and exponential growth of linear dimension with intrinsic dimension is required to capture any non-vanishing fraction of variance.

Previous theoretical work measuring linear dimension has focused on the participation ratio [14, 56, 82–85]. If $\{\lambda_1, \ldots, \lambda_N\}$ are the eigenvalues of the data covariance matrix, then the participation ratio (PR) is $\left(\sum_i \lambda_i\right)^2 / \sum_i \lambda_i^2$. Thus, if the eigenvalue mass is concentrated on a few large eigenvalues the participation ratio is high. Notably, Recanatesi et al. show that for 2-dimensional Gaussian tuning, the PR increases inversely proportional to tuning curve width, much as we find [31]. We instead define linear dimension as the number of eigenvalues required to account for a certain fraction (i.e., $1 - \epsilon$) of total variance, as this is more easily interpretable in terms of data variance, and closely matches what is done in practice when using methods such as PCA [1, 9, 32, 81]. Depending on the particular eigenvalue distribution, the PR typically corresponds to the number of eigenvalues required to explain about 80% to 95% of the variance and is thus well-correlated with our definition of linear dimension [56]. Moreover, Wigderson & Wigderson [70] derive an uncertainty principle for localization as measured by the PR. In the context of our results, this principle implies that for translation-invariant tuning, the PR of the eigenvalue distribution must grow as the number of neurons divided by the PR of the covariance profile (similar to the growth of linear dimension with sparsity). Consequently, for translation-invariant tuning our results will extend to linear dimension as measured by the participation ratio. More generally, the participation ratio has a number of nice mathematical properties and for these reasons was introduced as a more

theoretically tractable alternative to fraction of variance explained when measuring dimensionality of neural population datasets [56]. Thus the participation ratio may provide a useful way to extend these results to other population codes.

The analyses presented here suggest that neural data should appear high-dimensional when viewed through linear dimensionality reduction methods, where "high-dimensional" is to be interpreted as large when compared to the number of encoded variables but still low-dimensional with respect to the number of neurons in a brain region. In accordance with our observations, recent data from large neural population data sets show high linear dimension [4, 86], and recordings from a number of low-dimensional systems appear distorted and significantly higher-dimensional than they actually are when viewed through linear methods as compared to nonlinear methods [15, 17, 18, 71].

Despite these observations, in many settings linear methods such as PCA and Factor Analysis have been successful at extracting structure from neural population data. What could explain this good performance?

One possible explanation is that the observed linear dimension is limited by task structure [56]. Gao and Ganguli show that the linear dimension (as measured by participation ratio) of neural data is upper-bounded by a measure of task complexity and argue that in common neuroscience tasks this measure is quite low. In many cases, the measure of task complexity they use grows exponentially with the number of task parameters. Thus, one possible test of the hypothesis that linear dimension is indeed limited by task structure is if observed linear dimension grows very rapidly as more task parameters are added. Another possible test is if linear dimension is higher in the case of naturalistic stimuli or during resting state activity when compared to more controlled task conditions.

A second explanation for when PCA performs well is that the linearity or nonlinearity of neural point clouds and manifolds may differ sharply across different brain regions, reflecting differences in coding strategies, as suggested by recent work [9, 31]. Our results most naturally apply to coding in sensory systems, to circuits in the hippocampal formation that reflect spatial information, and to sparse combinatorial encoding of information in cognitive cortex. By contrast, if linear decodability is a good proxy for generalizability [6] then brain regions that strive to construct easily generalizable representations may show comparatively low linear dimension. Or, if the role of a brain region is to transform an initial condition into a particular dynamical pattern of activity, as suggested for motor cortex [87], then the data will be dominated by the linear dimension of the underlying dynamical system. This dynamical system may occupy a low-dimensional linear subspace because of constraints on learning and connectivity structure or the need for smoothness, controllability, and avoiding chaotic dynamics [3, 9, 12, 23]. As a third possibility, confining neural dynamics to low-dimensional linear subspaces that differ across tasks might enable efficient continual learning without interference [88]. Thus, as recently proposed, the ratio of linear to intrinsic dimension might provide a useful signature of encoding strategies and task demands across brain regions and over the course of learning [9, 31]. Characterizing this ratio is an interesting direction for future work, and is increasingly tractable given advances in both large-scale recordings and nonlinear manifold learning algorithms.

Methods that seek to find the intrinsic dimension (D) of a nonlinear data manifold rather than using the dimension of a linear embedding (L) are an active and promising area of research [15, 17, 31, 34, 57, 61, 81, 85, 86, 89–94]. Our formalism identifies a natural class of low-dimensional manifolds that should exist in neural data. These manifolds could be a useful and theoretically-tractable setting in which to evaluate existing and future methods that estimate intrinsic dimension, and thus to help identify promising candidates for real-world application. For example, a simple way to compare dimensionality estimation algorithms might be to apply them to simulated data

generated from a population of neurons with Gaussian tuning to a D-dimensional latent variable, with added noise. The algorithms could then be compared on whether they successfully extract these highly nonlinear manifolds, how efficient they are at doing so in terms of both computation time and data, and how robust they are to noise in the data.

Finally, while our results suggest caution in the application of PCA and related linear methods, they raise the encouraging possibility that, at least in certain brain regions, low-dimensional structure in population responses may have been missed by linear analyses.

4 Methods

4.1 Figure parameters and simulation details

Figure 2 For Figs. 2c,d, there are $N = 50$ neurons uniformly spaced in $[0, 1]$ and $N_\alpha = 10^4$ values of the latent variable are drawn from a uniform distribution in $[0, 1]$, so that A is a 50×10^4 matrix. All neurons have Gaussian tuning curves with a fixed width σ and periodic boundary conditions. In Fig. 2c, circles are eigenvalues of covariance matrix from simulations and lines are theoretical predictions from Eq. (37). In Fig.2d, circles are $L_{0.95}$ calculated from simulations, and the line is theoretical prediction in Eq. (39).

Figure 3 Circles in Fig 3c,e,f are calculated from simulations with data generated from D dimensional Gaussian tuning curves with periodic boundary conditions along each dimension with the following parameters: $D = 2$, 10 neurons per dimension (100 total); $D = 3$, 10 neurons per dimension (1000 total); $D = 4$, 8 neurons per dimension (4096 total). Simulations used 10^4 uniformly distributed values of the latent variable. In Fig 3c, $\sigma = 0.15$. Thin lines are eigenvalues from theoretical prediction Eq. (10). Thick lines are smooth interpolations from Eq. (12). In Fig 3e,f, diamonds are linear dimensions found numerically from tensor product of 1D eigenvalues (used to speed up computation as D gets larger). Dashed lines are semi-analytic approximation found from the chi-distribution. Solid lines are lower bounds in Eq. (44), and apply to any $L_{1-\epsilon}$ provided $\epsilon < 0.5$.

Figure 4 For Fig 3b-d, neurons along each dimension have sigmoidal response functions $f(x, \alpha) = 1/(1 + e^{-s(x-\alpha)})$. For simulations, there are 8 neurons along each dimension with uniformly spaced α in $[0, 1]$.Slopes s are uniformly spaced in the range $[-5, 5]$. For Fig 3e-g, neurons along each dimension have Gaussian tuning curves with varying widths, starting with the minimum value at the center of the range $[0, 1]$ and increasing towards the ends. There are 8 neurons along each dimension. The minimum width is 0.05 and increases in steps of 0.05 to the maximum of 0.2. In Fig 3c,f, circles are eigenvalues of covariance matrices calculated from D dimensional data and lines are tensor products of eigenvalues of the $1D$ covariance matrix, shown in the inset. In Fig 3d,g, circles are $L_{0.9}$ calculated from simulations and lines are asymptotic lower bounds $2^{D(H-0.05)}$ where H is the entropy of the normalised $1D$ eigenvalues shown in the inset of 3c,f.

4.2 Linear dimension of mean subtracted data is lower bounded by the linear dimension of non-mean-subtracted data minus 1

We consider a population of N neurons with activities at time t given by $\boldsymbol{y}(t) = [y_1(t), y_2(t), \cdots, y_N(t)]$. The mean of the population activity is $\boldsymbol{\mu} = \mathbb{E}_t\left[\boldsymbol{y}(t)\right]$, where the expectation value is taken over time. The non-mean subtracted covariance matrix is given by $C = \mathbb{E}_t\left[\boldsymbol{y}\boldsymbol{y}^T\right]$. And the mean-subtracted covariance matrix is given by $T = \mathbb{E}_t\left[(\boldsymbol{y} - \boldsymbol{\mu})(\boldsymbol{y} - \boldsymbol{\mu})^T\right]$. Note that $C = T + \boldsymbol{\mu}\boldsymbol{\mu}^T$.

The matrices C, T and $\boldsymbol{\mu}\boldsymbol{\mu}^T$ are Hermitian and positive semi-definite. Moreover, $\boldsymbol{\mu}\boldsymbol{\mu}^T$ is rank 1, with 1 non-zero eigenvalue $||\boldsymbol{\mu}||^2$ and remaining eigenvalues 0. Let the eigenvalues of these matrices

be denoted as follows

$$\begin{aligned} C &: \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \geq 0 \\ T &: \rho_1 \geq \rho_2 \geq \cdots \geq \rho_N \geq 0 \\ \mu\mu^T &: \nu_1 \geq \nu_2 = \cdots = \nu_N = 0 \end{aligned} \tag{16}$$

The Weyl inequality for Hermitian matrices yields

$$\rho_a + \nu_b \leq \lambda_p \leq \rho_r + \nu_s \quad a + b - N \geq p \geq r + s - 1 \tag{17}$$

With $a = p, b = N, r = p - 1, s = 2$ we get

$$\rho_p \leq \lambda_p \leq \rho_{p-1} \quad \text{for } p \geq 2. \tag{18}$$

And with $a = 1, b = N, r = 1, s = 1$ we get

$$\rho_1 \leq \lambda_1 \leq \rho_1 + ||\mu||^2 \tag{19}$$

Thus we have the eigenvalue bounds

$$0 \leq \rho_N \leq \lambda_N \leq \rho_{N-1} \leq \cdots \leq \lambda_2 \leq \rho_1 \leq \lambda_1 \leq \rho_1 + ||\mu||^2 \tag{20}$$

Now assume that the $(1 - \epsilon)$ linear dimension of the non-mean-subtracted system is L. Thus

$$\frac{\sum_{k=1}^{L-1} \lambda_k}{\sum_{k=1}^{N} \lambda_k} < (1 - \epsilon) \tag{21}$$

Define $\Delta = \sum_{k=1}^{L-2}(\rho_k - \lambda_k) - \lambda_{L-1}$. Note that $\Delta < 0$ by the above inequalities.

$$\begin{aligned} 0.95 > &\frac{\sum_{k=1}^{L-1} \lambda_k}{\sum_{k=1}^{N} \lambda_k} > \frac{\sum_{k=1}^{L-1} \lambda_k + \Delta}{\sum_{k=1}^{N} \lambda_k + \Delta} = \frac{\sum_{k=1}^{L-2} \rho_k}{\sum_{k=1}^{L-2} \rho_k + \sum_{k=L}^{N} \lambda_k} \\ &\geq \frac{\sum_{k=1}^{L-2} \rho_k}{\sum_{k=1}^{L-2} \rho_k + \sum_{k=L-1}^{N-1} \rho_k} = \frac{\sum_{k=1}^{L-2} \rho_k}{\sum_{k=1}^{N-1} \rho_k} \geq \frac{\sum_{k=1}^{L-2} \rho_k}{\sum_{k=1}^{N} \rho_k} \end{aligned} \tag{22}$$

Thus $\frac{\sum_{k=1}^{L-2} \rho_k}{\sum_{k=1}^{N} \rho_k} < 0.95$.

Consequently if the $(1 - \epsilon)$-linear dimension of C is L, the $(1 - \epsilon)$-linear dimension of T is at least $L - 1$. In general, depending on $||\mu||^2$ the linear dimension of T can be much higher than that of C. Thus, lower bounds on the linear dimension of the non-mean-subtracted case transfer simply to lower bounds on the linear dimension of the mean-subtracted case.

4.3 The covariance matrix of translation invariant data is translation invariant

We assume that the neural response depends on a D-dimensional latent variable, and thus the tuning centers, $\boldsymbol{\alpha_n}$ and the stimulus variables $\boldsymbol{x}_s = \boldsymbol{x}(t_s)$ are in $\mathbb{R}^D$.

The nsth element of the data matrix is

$$A_{ns} = f(\boldsymbol{x_s}, \boldsymbol{\alpha_n}). \tag{23}$$

Here f is a translation invariant tuning curve so that $f(\boldsymbol{x_s} + \boldsymbol{\delta}, \boldsymbol{\alpha_n} + \boldsymbol{\delta}) = f(\boldsymbol{x_s}, \boldsymbol{\alpha_n}) = g(\boldsymbol{x_s} - \boldsymbol{\alpha_n})$ for some function g. We assume periodic boundary condition along each dimension with period 1

and thus $\boldsymbol{x}, \boldsymbol{\alpha} \in [0,1]^D$. The latent variable is assumed to be uniformly distributed so that values are drawn uniformly from $[0,1]^D$.

The neuron-neuron covariance matrix is given by

$$\begin{aligned} C_{mn} &= \frac{1}{N_x} A A^T \\ &= \frac{1}{N_x} \sum_s f(\boldsymbol{x_s}, \boldsymbol{\alpha_m}) f(\boldsymbol{x_s}, \boldsymbol{\alpha_n}) \end{aligned} \tag{24}$$

where N_x is the number of latent variable values. In the limit of large N_x this sample mean will converge to $\mathbb{E}[f(\boldsymbol{x}, \boldsymbol{\alpha_m}) f(\boldsymbol{x}, \boldsymbol{\alpha_n})]_{\boldsymbol{x}}$, where the expectation is taken over the latent variable distribution. Thus the covariance matrix can also be written as

$$C_{mn} = \int_{\mathcal{S}} d\boldsymbol{x}\, p(\boldsymbol{x}) f(\boldsymbol{x}, \boldsymbol{\alpha_m}) f(\boldsymbol{x}, \boldsymbol{\alpha_n}) = \int_{\mathcal{S}} d\boldsymbol{x}\, f(\boldsymbol{x}, \boldsymbol{\alpha_m}) f(\boldsymbol{x}, \boldsymbol{\alpha_n}) \tag{25}$$

where $p(\boldsymbol{x})$ is the the distribution over the latent variable, which we assume to be uniform, and the integral $\int_{\mathcal{S}} d\boldsymbol{x} = \int_{\mathcal{S}} dx_1 \cdots dx_D$, is over all points $\boldsymbol{x}$ in the region $\mathcal{S} = [0,1]^D$. Since the tuning curves are translation invariant we have

$$\begin{aligned} C_{mn} &= \int_{\mathcal{S}} d\boldsymbol{x}\, f(\boldsymbol{x}, \boldsymbol{\alpha_m}) f(\boldsymbol{x}, \boldsymbol{\alpha_n}) \\ &= \int_{\mathcal{S}} d\boldsymbol{x}\, f(\boldsymbol{x} - \boldsymbol{\alpha_n}, \boldsymbol{\alpha_m} - \boldsymbol{\alpha_n}) f(\boldsymbol{x} - \boldsymbol{\alpha_n}, 0) \\ &= \int_{\mathcal{S} - \boldsymbol{\alpha_n}} d\boldsymbol{x}'\, f(\boldsymbol{x}', \boldsymbol{\alpha_m} - \boldsymbol{\alpha_n}) f(\boldsymbol{x}', 0) \\ &= \int_{\mathcal{S}} d\boldsymbol{x}\, f(\boldsymbol{x}, 0) f(\boldsymbol{x}, \boldsymbol{\delta}) \\ &= c(\boldsymbol{\delta}) = c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n}). \end{aligned} \tag{26}$$

In going from the first to the second line, we have used the translation invariance of the tuning curves ($f(\boldsymbol{x}, \boldsymbol{\alpha}) = f(\boldsymbol{x} - \boldsymbol{\gamma}, \boldsymbol{\alpha} - \boldsymbol{\gamma})$); in going from the second line to the third line we have made a change of variable, $\boldsymbol{x}' = \boldsymbol{x} - \boldsymbol{\alpha_n}$ and defined $\boldsymbol{\delta} = \boldsymbol{\alpha_m} - \boldsymbol{\alpha_n}$; and in going from the third line to fourth line we have used the periodic boundary conditions to shift the integral over $\mathcal{S} - \boldsymbol{\alpha_m}$ to $\mathcal{S}$. c is a periodic function of the difference in tuning curve centers with period 1 along each dimension. Thus the covariance between two neurons just depends on the difference between their tuning curve centers.

In the one-dimensional case ($D = 1$), if we take the tuning curve centers to be uniformly spaced (i.e., $\alpha_n = n/N$ for $n = 0, \cdots N-1$), then the covariance matrix is circulant meaning that each row is a shifted version of the row above.

$$\begin{aligned} C_{m+1,n} &= c(\alpha_{m+1} - \alpha_n) \\ &= c((\alpha_{m+1} - 1/N) - (\alpha_n - 1/N)) \\ &= c(\alpha_m - \alpha_{n-1}) \\ &= C_{m,n-1} \end{aligned} \tag{27}$$

Note that this holds for $m, n = 1, \cdots N$, since the periodic boundary conditions allow us to define $C_{N+1,n} = C_{1,n}$ and $C_{m,0} = C_{m,N}$.

Finally, note that the results approximately hold if the boundary conditions are hard rather than periodic, provided that the tuning curves are not too wide. In this case, the deviations from perfect translation-invariance caused by the boundary conditions will be mild and restricted to a small subset of neurons.

4.4 Eigenvalues of translation invariant covariance matrices

It is well-known that matrices with translation-invariant structure have eigenvalues that are given by the Fourier transform of the function that generates the matrix (i.e., c) [59, 63]. We briefly review those results here.

First, consider tuning curve centers that tile the latent space, forming the points of a lattice with N_d tuning curve centers along the d-th dimension. Thus, the spacing of tuning curve centers along the d-th dimension is $1/N_d$, and there are $N = \prod_d N_d$ neurons in total, in a volume of 1^D. The tuning curve centers are given by

$$\boldsymbol{\alpha_n} = \frac{1}{N_d}(a_1^n, \cdots, a_D^n) \tag{28}$$

where each a_d^n belongs to $[0, 1, \cdots, N_d - 1]$.

The covariance matrix has entries $C_{mn} = c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha}_n) = c(\boldsymbol{\delta})$. Note that because of the periodicity of c (as a consequence of the periodic boundary conditions), the vector $\boldsymbol{\delta}$ can be considered to take the same set of possible values as the tuning curve centers.

Next, consider the set of N vectors $\boldsymbol{k_p}$ given by

$$\boldsymbol{k_p} = 2\pi(a_1^p, \cdots, a_D^p), \tag{29}$$

where as before each a_d^p belongs to $[0, 1, \cdots, N_d - 1]$. Corresponding to each $\boldsymbol{k_p}$, define the vector $\boldsymbol{v_p} = (e^{i\boldsymbol{k_p}.\boldsymbol{\alpha_1}}, \cdots, e^{i\boldsymbol{k_p}.\boldsymbol{\alpha_N}})$. Multiplying this vector by the matrix C yields

$$\begin{aligned}
(C\boldsymbol{v_p})_m &= \sum_{n=1}^{N} c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n}) e^{i\boldsymbol{k_p}.\boldsymbol{\alpha_n}} \\
&= \left(\sum_{n=1}^{N} c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n}) e^{-i\boldsymbol{k_p}.(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n})} \right) e^{i\boldsymbol{k_p}.\boldsymbol{\alpha_m}} \\
&= \left(\sum_{n=1}^{N} c(\boldsymbol{\delta_n}) e^{-i\boldsymbol{k_p}.\boldsymbol{\delta_n}} \right) e^{i\boldsymbol{k_p}.\boldsymbol{\alpha_m}}
\end{aligned} \tag{30}$$

Here we have defined $\boldsymbol{\delta_n} = \boldsymbol{\alpha_m} - \boldsymbol{\alpha_n}$. As a consequence of the uniform spacing of tuning curve centers and periodic boundary conditions, the term in parentheses is a sum over all possible values of $\boldsymbol{\delta_n} = \frac{1}{N_d}(a_1^n, \cdots, a_D^n)$, where as before each $a_d^n \in [0, 1, \cdots N_d - 1]$. Thus, it is independent of both m and n and we can define $\lambda_p = \sum_{n=1}^{N} c(\boldsymbol{\delta_n}) e^{-i\boldsymbol{k_p}.\boldsymbol{\delta_n}}$. Consequently, $\boldsymbol{v_p}$ is an eigenvector of C with eigenvalue λ_p. Note that λ_p is the term with frequency k_p of the discrete Fourier transform of c.

In particular, for the one-dimensional case, C is a circulant matrix. In this case, we have tuning curve centers $\alpha_n = \frac{n}{N}$, where $n \in [0, \cdots, N-1]$. For each p with $0 \leq p \leq N-1$, $\boldsymbol{v_p} = (e^{ik_p.\alpha_1}, \cdots, e^{ik_p.\alpha_N})$ is an eigenvector with eigenvalue $\lambda_p = \sum_{n=0}^{N-1} c(\alpha_n) e^{-2\pi i p n/N}$.

Alternatively, if tuning curve centers are uniformly distributed through the latent space, either evenly spaced or randomly chosen, then in the large N limit we can consider the translation invariant kernel $c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n})$, which is a continuous function of $\boldsymbol{\alpha_m}, \boldsymbol{\alpha_n} \in \mathbb{R}^D$. This is the continuous generalisation of the covariance matrix for translation invariant tuning curves derived above. The product of the matrix C with a vector can then be approximated by the convolution of the kernel with a function $f_1(\boldsymbol{\alpha_n})$,

$$f_2(\boldsymbol{\alpha_m}) = \int d\boldsymbol{\alpha_n}\, c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n})\, f_1(\boldsymbol{\alpha_n}) \tag{31}$$

where $\int d\boldsymbol{\alpha_n}$ means an integral over all D dimensional vectors $\boldsymbol{\alpha}_n$. $f(\boldsymbol{\alpha}_n)$ is an eigenfunction if

$$\int d\boldsymbol{\alpha_n}\, c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n})\, f(\boldsymbol{\alpha_n}) = \lambda f(\boldsymbol{\alpha_m}) \tag{32}$$

Consider $f(\boldsymbol{\alpha}_n) = e^{i\boldsymbol{k}.\boldsymbol{\alpha_n}}$ where $\boldsymbol{k}$ is an arbitrary D dimensional vector.

$$\begin{aligned}
&\int d\boldsymbol{\alpha_n}\, c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n})\, e^{i\boldsymbol{k}.\boldsymbol{\alpha_n}} \\
&= \int d\boldsymbol{\alpha_n}\, c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n})\, e^{-i\boldsymbol{k}.(\boldsymbol{\alpha_m}-\boldsymbol{\alpha_n})}\, e^{i\boldsymbol{k}.\boldsymbol{\alpha_m}} \\
&= \left(\int d\boldsymbol{\delta}\, c(\boldsymbol{\delta})\, e^{-i\boldsymbol{k}.\boldsymbol{\delta}}\right) e^{i\boldsymbol{k}.\boldsymbol{\alpha_m}}
\end{aligned} \tag{33}$$

As before in going from line 2 to line 3, we have defined a new variable, $\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n} = \boldsymbol{\delta}$ and used the periodic boundary conditions to replace the integral over $\boldsymbol{\alpha_n}$ with an integral over $\boldsymbol{\delta}$. The expression in parenthesis does not depend on $\boldsymbol{\alpha_m}$ or $\boldsymbol{\alpha_n}$ and is the Fourier transform of $c(\boldsymbol{\delta})$. So $e^{i\boldsymbol{k}.\boldsymbol{\alpha_n}}$ is an eigenfunction of $c(\boldsymbol{\alpha_m} - \boldsymbol{\alpha_n})$ with eigenvalue $\int d\boldsymbol{\delta}\, c(\boldsymbol{\delta})\, e^{-i\boldsymbol{k}.\boldsymbol{\delta}}$.

Finally, while these results are for uniformly distributed data with periodic boundary conditions, note that covariance matrices are symmetric and thus normal. Consequently, the effect of perturbations on the eigenvalue spectrum is as mild as possible, and the results should hold approximately for matrices that only approximately satisfy these conditions.

4.5 Gaussian tuning curves show a Gaussian covariance profile

Consider the case of Gaussian tuning, where the tuning curve of the n-th neuron is given by

$$y_n(t) = f(\boldsymbol{x(t)}, \boldsymbol{\alpha_n}) = K_1 \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\alpha_n})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\alpha_n})\right). \tag{34}$$

Here Σ is the covariance matrix of the tuning curve.

To generate the neuron-neuron covariance matrix, we wish to calculate the covariance profile $c(\boldsymbol{\delta}) = \int_{\mathcal{S}} d\boldsymbol{x}\, f(\boldsymbol{x}, 0) f(\boldsymbol{x}, \boldsymbol{\delta})$, which is the covariance between pairs of neurons whose tuning curve centers are separated by $\boldsymbol{\delta}$. Note that c is periodic, with period 1 in each direction. For notational convenience we shift the range of $\boldsymbol{\delta}$ so that each component lies within $[-1/2, 1/2]$ rather than $[0, 1]$. Thus,

$$\begin{aligned}
c(\boldsymbol{\delta}) &= \int_{\mathcal{S}} d\boldsymbol{x}\, f(\boldsymbol{x}, 0) f(\boldsymbol{x}, \boldsymbol{\delta}) \\
&= K_1^2 \int_{\mathcal{S}} \boldsymbol{dx} \exp\left(-\frac{1}{2}\boldsymbol{x}^T \Sigma^{-1} \boldsymbol{x}\right) \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\delta})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\delta})\right) \\
&= K_1^2 \int_{\mathcal{S}} \boldsymbol{dx} \exp\left[-\left(\boldsymbol{x}^T \Sigma^{-1} \boldsymbol{x} - \frac{\boldsymbol{\delta}^T \Sigma^{-1} \boldsymbol{x}}{2} - \frac{\boldsymbol{x}^T \Sigma^{-1} \boldsymbol{\delta}}{2} + \frac{\boldsymbol{\delta}^T \Sigma^{-1} \boldsymbol{\delta}}{2}\right)\right] \\
&= K_1^2 \int_{\mathcal{S}} \boldsymbol{dx} \exp\left[-\left(\boldsymbol{x} - \frac{\boldsymbol{\delta}}{2}\right)^T \Sigma^{-1} \left(\boldsymbol{x} - \frac{\boldsymbol{\delta}}{2}\right)\right] \exp\left(-\frac{\boldsymbol{\delta^T} \Sigma^{-1} \boldsymbol{\delta}}{4}\right) \\
&= K_1^2 \exp\left(-\frac{\boldsymbol{\delta^T} \Sigma^{-1} \boldsymbol{\delta}}{4}\right) \int_{\mathcal{S}} \boldsymbol{dx} \exp\left[-\left(\boldsymbol{x} - \frac{\boldsymbol{\delta}}{2}\right)^T \Sigma^{-1} \left(\boldsymbol{x} - \frac{\boldsymbol{\delta}}{2}\right)\right] \\
&= K_2 \exp\left(-\frac{\boldsymbol{\delta^T} \Sigma^{-1} \boldsymbol{\delta}}{4}\right).
\end{aligned} \tag{35}$$

Here the constant $K_2 = K_1^2\sqrt{\det(\pi\Sigma)}$. The last equality is approximate but holds when tuning curves are not too wide, meaning that no single neuron responds to the full range of latent variable values, so that integrating over the range of latent variable values is equivalent to integrating over the range of the tuning curve. Thus, neurons with Gaussian tuning curves have a Gaussian covariance profile with covariance twice that of the tuning curve.

4.6 Variance explained and linear dimension for one-dimensional Gaussian tuning curves

In the one-dimensional case, the covariance profile is

$$\begin{aligned} c(\delta) =& K_2 \exp\left(-\frac{\delta^2}{4\sigma^2}\right), \quad \delta \leq \frac{1}{2} \\ =& K_2 \exp\left(-\frac{(1-\delta)^2}{4\sigma^2}\right), \quad \delta > \frac{1}{2} \end{aligned} \tag{36}$$

If tuning curve centers are evenly spaced, this covariance profile is sampled at $\delta_n = n/N$, where n takes values in $[0, \ldots, N-1]$.

As described in Section 4.4, the eigenvalues of the covariance matrix are given by the Fourier transform of this profile. For convenience we index the eigenvalues by p ranging from $\lfloor -N/2 \rfloor$ to $\lfloor (N-1)/2 \rfloor$. Ignoring the overall normalization constant K_2, for each such p we have an eigenvalue

$$\begin{aligned} \lambda_p &= \sum_{n=0}^{N-1} c(n/N) e^{-2\pi i p\, n/N} = \sum_{n=0}^{\lfloor (N-1)/2 \rfloor} c(n/N) e^{-2\pi i p\, n/N} + \sum_{l=1}^{\lfloor (N+1)/2 \rfloor} c(1-l/N) e^{2\pi i p\, l/N} \\ &= \sum_{n=0}^{\lfloor (N-1)/2 \rfloor} \exp\left(\frac{-n^2}{4\sigma^2 N^2}\right) \exp\left(\frac{-2\pi i n p}{N}\right) + \sum_{l=1}^{\lfloor (N+1)/2 \rfloor} \exp\left(\frac{-l^2}{4\sigma^2 N^2}\right) \exp\left(\frac{2\pi i l p}{N}\right) \\ &= \sum_{n=\lfloor -N/2 \rfloor}^{\lfloor (N-1)/2 \rfloor} \exp\left(\frac{-n^2}{4\sigma^2 N^2}\right) \exp\left(\frac{-2\pi i n p}{N}\right) \\ &= K_3 \exp\left(-4\pi^2\sigma^2 p^2\right) \end{aligned} \tag{37}$$

where K_3 is a constant, and in the last line we have completed the square and noted that the sum over n is a constant. Thus the eigenvalues have a Gaussian profile with width $\frac{2\sqrt{2}\pi}{\sigma}$. Also note that the eigenvalues decrease monotonically with the magnitude of p and that, except for λ_0, they occur in pairs with $\lambda_p = \lambda_{-p}$.

Since the eigenvalues occur in pairs, the $(1-\epsilon)$-linear dimension is the smallest $L_{1-\epsilon}$ such that

$$\sum_{p=0}^{\frac{L_{1-\epsilon}}{2}} \lambda_p \geq (1-\epsilon) \sum_{p=0}^{N/2} \lambda_p. \tag{38}$$

For large N, approximating both sides of this equation as an integral and canceling the common

prefactor K_3 yields

$$\begin{aligned}
\int_0^{\frac{L_{1-\epsilon}}{2}} e^{-(2\pi\sigma p)^2} dp \geq \ (1-\epsilon) \int_0^{\infty} e^{-(2\pi\sigma p)^2} dp = \frac{(1-\epsilon)}{4\sqrt{\pi}\sigma} \\
4\sqrt{\pi}\sigma \int_0^{\frac{L_{1-\epsilon}}{2}} e^{-(2\pi\sigma p)^2} dp \geq \ (1-\epsilon) \\
\frac{2}{\sqrt{\pi}} \int_0^{\pi\sigma L_{1-\epsilon}} e^{-z^2} dz \geq (1-\epsilon) \\
\operatorname{erf}(\pi\sigma L_{1-\epsilon}) \geq (1-\epsilon) \\
L_{1-\epsilon} = \frac{1}{\sigma} \frac{\operatorname{erf}^{-1}(1-\epsilon)}{\pi}
\end{aligned} \tag{39}$$

Thus, $L_{1-\epsilon}$ grows as $\frac{1}{\sigma}$, with a proportionality constant that depends on the fraction of variance explained. In particular, for 95% variance explained, we have $L_{0.95} = \frac{1.96}{\sqrt{2}\,\pi\sigma}$.

4.7 Variance explained and supraexponential scaling of linear dimension for multi-dimensional Gaussians

We next consider a population neurons with Gaussian tuning to an underlying D-dimensional latent variable. We assume that the tuning curve centers lie on a D dimensional grid with N_d equally spaced centers along each dimension. Therefore there are $N = N_d^D$ neurons.

For simplicity we also assume that the width of the tuning is the same along each direction so that the covariance matrix of the tuning curve Σ is diagonal with elements σ^2 along the diagonal. Thus, as shown in Sec 4.4, the covariance between two neurons whose tuning curve centers are separated by $\boldsymbol{\delta}$ is $c(\boldsymbol{\delta}) = K_2 \exp\left(-\frac{||\boldsymbol{\delta}||^2}{4\sigma^2}\right)$, for $\boldsymbol{\delta} \in [-\frac{1}{2}, \frac{1}{2}]^D$, where K_2 is an overall scaling constant. $c(\boldsymbol{\delta})$ is extended periodically outside this range.

As for the one-dimensional case and as discussed in Sec. 4.4, the eigenvalues of the neuron-neuron covariance matrix are given by the Fourier transform of $c(\boldsymbol{\delta})$. We can index the eigenvalues by a D-dimensional vector $\boldsymbol{p}$ with dth entry $p_d \in [-N_d/2, \cdots, N_d/2]$ yielding

$$\lambda_{\boldsymbol{p}} = K_3 \exp\left(-4\pi^2 \sum_{d=1}^{D} p_d^2 \sigma^2\right) = K_3 \exp(-4\pi^2\sigma^2 ||\boldsymbol{p}||^2). \tag{40}$$

The magnitude of an eigenvalue is thus a function only of $||\boldsymbol{p}||$, the Euclidean distance from the origin in $\boldsymbol{p}$-space. As there are multiple lattice points with the same distance from the origin, there will be multiple eigenvalues with the same magnitude. Moreover, the number of eigenvalues at a given distance from the origin increases as the distance increases. Consequently, as the distance from the origin increases, there will be more eigenvalues but with smaller magnitude.

Overloading notation to define the D-dimensional continuous Gaussian function $\lambda : \mathbb{R}^D \to \mathbb{R}$ as $\lambda(||\boldsymbol{p}||) = K_3 \exp(-4\pi^2\sigma^2||\boldsymbol{p}||^2)$, observe that the eigenvalues are given by $\lambda(||\boldsymbol{p}||)$ sampled at the integer lattice points of a D-dimensional cube with side length N_d.

To estimate the $(1-\epsilon)$ linear dimension, note that summing up the $L_{1-\epsilon}$ largest eigenvalues is equivalent to summing up all eigenvalues for which $||\boldsymbol{p}|| \leq R$ (i.e., eigenvalues corresponding to all lattice points with a ball of radius R), up to some edge effects resulting from the discreteness of the lattice points that will vanish when we take the continuous limit below. Thus we first compute the smallest radius R such that

$$\sum_{||\boldsymbol{p}|| \leq R} \lambda_{\boldsymbol{p}} \geq (1-\epsilon) \sum_{p} \lambda_{\boldsymbol{p}}. \tag{41}$$

Approximating the sum by the integral of $\lambda(\boldsymbol{p})$ and setting the constant $K_4 = \int_{||\boldsymbol{p}||} d\boldsymbol{p}\lambda(\boldsymbol{p})$, we seek R such that

$$\frac{1}{K_4}\int_{||\boldsymbol{p}||\leq R} dp_1 dp_2 \cdots dp_D \exp(-(2\pi\sigma)^2 \sum_{d=1}^{D} p_d^2) \geq (1-\epsilon). \tag{42}$$

Converting to radial coordinates with $r^2 = \sum_{d=1}^{D} p_d^2$ and then making a second change of coordinates $y = \sqrt{8}\pi\sigma r$ yields

$$\begin{aligned}\frac{1}{K_4}\int_{||\boldsymbol{p}||\leq R} dp_1 \ldots dp_D \exp(-(2\pi\sigma)^2 \sum_{d=1}^{D} p_d^2) &= \frac{1}{K_5}\int_0^R dr\, r^{D-1} \exp(-(2\pi\sigma)^2 r^2) \\ &= \frac{1}{K_6}\int_0^{\sqrt{8}\pi\sigma R} dy\, y^{D-1} \exp(-y^2/2),\end{aligned} \tag{43}$$

where K_5 and K_6 are constants.

Note that λ is the (unnormalized) probability density function of a multivariate Gaussian distribution and the argument of the integral above is the density function of a chi distribution with D degrees of freedom (appropriately normalized as a result of dividing by the integral across the entire range). Thus, defining the random variable Z distributed according to a chi distribution with D degrees of freedom, note that the integral itself corresponds to the CDF of Z and we seek the smallest R such that $P(Z \leq \sqrt{8}\pi\sigma R) \geq 1-\epsilon$.

If $\epsilon < 1/2$ (i.e., we are capturing at least 50% of the variance in the data), then by definition R must be at least the median value of Z. Standard results on chi distributions then show that $\sqrt{8}\pi\sigma R \geq \sqrt{D}\left(1-\frac{2}{9D}\right)^{3/2} > 0.8\sqrt{D}$, where the final inequality holds for $D \geq 2$. Consequently, $R > \frac{0.8}{\sqrt{8}\pi}\frac{\sqrt{D}}{\sigma}$.

Finally, to estimate the linear dimension itself we need to count the number of eigenvalues that lie within a sphere of radius R. Since the eigenvalues correspond to lattice points, the number of eigenvalues is approximately the volume of this D-dimensional sphere yielding

$$L_{1-\epsilon} \approx V_D = \frac{\pi^{D/2}}{\Gamma\left(\frac{D}{2}+1\right)} R^D = \left(\frac{K_7\sqrt{D}}{\sigma}\right)^D, \tag{44}$$

where K_7 is a constant. Consequently the linear dimension grows supra-exponentially with D.

4.8 The number of localized tuning curves required to cover a D-dimensional manifold grows supraexponentially with D

Some further intuition for the $\sqrt{D}^D$ scaling of linear dimension in the translation-invariant case can be gained by considering the number of neurons with localized firing fields required to cover a manifold. Consider a population of neurons that show localized firing fields to a D-dimensional latent variable that takes values in $[0,1]^D$. As a consequence of the localized firing fields, for each neuron the region of high firing rate on the manifold can be approximated as a hypersphere of some radius r with corresponding volumes $V_{single} = \frac{\pi^{D/2} r^D}{\Gamma(D/2+1)}$.

If there are N neurons, the manifold volume that lies within the firing fields is at most

$$\begin{aligned}V_{covered} &\leq N\frac{\pi^{D/2} r^D}{\Gamma(D/2+1)} \\ &\approx N(2e\pi)^{D/2}\left(\frac{r}{\sqrt{D}}\right)^D,\end{aligned} \tag{45}$$

where the second line follows from Stirling's approximation.

Thus to cover some fixed fraction $1-\epsilon$ of the total volume, N must grow at least as fast as $\frac{1}{(2e\pi)^{D/2}}\left(\frac{\sqrt{D}}{r}\right)^D$. This supraexponential scaling reflects the fact that as the number of dimensions increases, an increasing amount of manifold volume lies in the spaces between the spheres.

4.9 Covariance matrix and eigenvalues of covariance matrix for multiplicative tuning curves

As in the main text we consider tuning curves of neurons which are a product of 1-dimensional tuning curves.

$$y_n(t) = f(\boldsymbol{x}(t), \boldsymbol{\alpha}_n) = \prod_{d=1}^{D} f_d(x^d(t), \alpha_n^d). \tag{46}$$

Assuming that the distribution of the latent variable along each dimension is independent (i.e., $p(\boldsymbol{x}) = \prod_d p(x^d)$, the covariance profile between neurons with tuning parameters $\boldsymbol{\alpha_m}$ and $\boldsymbol{\alpha_n}$ is

$$\begin{aligned} c(\boldsymbol{\alpha_m}, \boldsymbol{\alpha_n}) &= \int d\boldsymbol{x}\, p(\boldsymbol{x}) f(\boldsymbol{x}, \boldsymbol{\alpha_m}) f(\boldsymbol{x}, \boldsymbol{\alpha_n}) \\ &= \prod \int dx^d\, p(x^d) f_d(x^d, \alpha_m^d) f_d(x^d, \alpha_n^d) \\ &\equiv \prod c^d(\alpha_m^d, \alpha_n^d), \end{aligned} \tag{47}$$

where the last line serves to define the individual covariance factors c^d.

If the tuning curve centers are chosen to tile the space, then the covariance matrix C can be written as a tensor product

$$C = C^1 \otimes \cdots \otimes C^D. \tag{48}$$

Here the factors $C^d(\alpha_m^d, \alpha_n^d) = \int dx^d\, p(x^d) f(x^d, \alpha_m^d) f(x^d, \alpha_n^d)$, each correspond to a dimension of the latent variable.

Let $\{\gamma_{p_d}^d, \boldsymbol{u_{p_d}^d}\}$ be the p_dth eigenvalue-eigenvector pair for each C^d. Note that

$$\begin{aligned} C\left(\boldsymbol{u_{p_1}^1} \otimes \cdots \otimes \boldsymbol{u_{p_D}^D}\right) &= \left(C^1 \otimes \cdots \otimes C^D\right)\left(\boldsymbol{u_{p_1}^1} \otimes \cdots \otimes \boldsymbol{u_{p_D}^D}\right) \\ &= \left(C^1 \boldsymbol{u_{p_1}^1}\right) \otimes \cdots \otimes \left(C^D \boldsymbol{u_{p_D}^D}\right) \\ &= \left(\gamma_{p_1}^1 \boldsymbol{u_{p_1}^1}\right) \otimes \cdots \otimes \left(\gamma_{p_D}^D \boldsymbol{u_{p_D}^D}\right) \\ &= \left(\gamma_{p_1}^1 \times \cdots \times \gamma_{p_D}^D\right)\left(\boldsymbol{u_{p_1}^1} \otimes \cdots \otimes \boldsymbol{u_{p_D}^D}\right) \end{aligned} \tag{49}$$

Consequently, $\left(\gamma_{p_1}^1 \times \cdots \times \gamma_{p_D}^D\right)$ is an eigenvalue of C. All N eigenvalues of C can be constructed this way, as products of the eigenvalues of the factor C^d. Consequently, the eigenvalues of C are given by all possible products of the eigenvalues of the individual factors.

4.10 The linear dimension of multiplicative models grows exponentially with intrinsic dimension

As described in the main text, for simplicity, we will consider a multiplicative model where the tuning to each dimension has the same shape. Thus, each factor C^d is the same.

Let the eigenvalues of the covariance matrix for each factor be $\{\gamma_1, \ldots, \gamma_{N_D}\}$. The N eigenvalues of the overall covariance matrix C are given by all products of the form $\prod_d \gamma_{p_d}$, where each $p_d \in [1, \ldots, N_D]$. To compute the linear dimension we reframe the problem as a problem in probability theory.

As we are considering fraction of variance explained, we can rescale $\{\gamma_1, \ldots, \gamma_{N_D}\}$ to sum to 1. Having made this normalization, consider a set of D independent random variables Z_d, each distributed according to a multinomial distribution with outcome probabilities $\{\gamma_1, \ldots, \gamma_{N_D}\}$. Each eigenvalue of C corresponds to the probability of some outcome of the string of random variables $Z_1 \ldots Z_D$. Finding the smallest set of eigenvalues whose sum is at least $1 - \epsilon$ (i.e., the $L_{1-\epsilon}$ linear dimension) is equivalent to finding the smallest set of outcomes whose probability is at least $1 - \epsilon$. This subset is often referred to as an ϵ high-probability set [80]. Standard results in information theory show that as D increases, the number of elements in this high-probability set approaches 2^{DH_γ}, where $H_\gamma = -\sum_p \gamma_p \log_2 \gamma_p$ is the Shannon entropy of the distribution $\{\gamma_1, \ldots, \gamma_{N_d}\}$. As each element in the high-probability set corresponds to an eigenvalue of the covariance matrix C, asymptotically $L_{1-\epsilon}$ grows as 2^{DH_γ}, thus increasing exponentially with D.

While this is an asymptotic argument, exponential scaling holds for small D as well. For a non-asymptotic lower bound on the linear dimension of these models, consider the case where only two of the eigenvalues of the matrix C^d are nonzero. By normalizing to sum to 1, we can write these eigenvalues as $1 - \gamma$ and γ, for some $\gamma \leq 0.5$. As for the multinomial case, the eigenvalues of C are the tensor product of $[1 - \gamma, \gamma]$ taken D times, i.e. $[1 - \gamma, \gamma] \underset{D\,times}{\otimes} \cdots \otimes [1 - \gamma, \gamma]$. In descending order of magnitude, there is 1 eigenvalue of magnitude $(1 - \gamma)^D$, $\binom{D}{1}$ eigenvalues of magnitude $(1 - \gamma)^{D-1}\gamma$, and so on, with $\binom{D}{k}$ eigenvalues of magnitude $(1 - \gamma)^{D-k}\gamma^k$.

To lower bound the linear dimension, note that if K is such that

$$\sum_{k=0}^{K} \binom{D}{k} (1 - \gamma)^{D-k} \gamma^k \leq (1 - \epsilon), \tag{50}$$

then the linear dimension $L \geq \sum_{k=0}^{K} \binom{D}{k}$. Thus, we will first find such a K and then use it to lower bound the linear dimension L.

Consider a random variable Z distributed according to the binomial distribution with probability γ. That is, $X \sim Bin(D, \gamma)$. Note that $\sum_{k=0}^{K} \binom{D}{k} (1 - \gamma)^{D-k} \gamma^k = P(X \leq K)$. The median of this binomial distribution is at least $\lfloor \gamma D \rfloor$. Thus if we choose $K = \lfloor \gamma D \rfloor - 1$ we have

$$\sum_{k=0}^{K} \binom{D}{k} (1 - \gamma)^{D-k} \gamma^k < 0.5 < 0.95. \tag{51}$$

For simplicity, define γ' such that $\gamma' D = \lfloor \gamma D \rfloor$ and note that we can rewrite $K = \rho D$, where $\rho > \gamma - 2/D$ (or, using results bounding the distance of the mean to the median of a binomial, this can be improved to $\rho > \gamma - (1 + \ln(2))/D$). We can now lower bound L as $L \geq \sum_{k=0}^{\rho D} \binom{D}{k}$.

A standard bound on the sum of binomial coefficients [80] yields

$$\sum_{k=0}^{\rho D} \binom{D}{k} \geq \frac{1}{\sqrt{8D\rho(1 - \rho)}} 2^{H(\rho)D} = \frac{1}{\sqrt{8\rho(1 - \rho)}} 2^{(H_b(\rho) - \frac{log_2(D)}{2D})D} \tag{52}$$

where $H_b(\rho) = -\rho \log_2 \rho - (1 - \rho) \log_2(1 - \rho)$ is the binary entropy function.

Thus, except when D is small enough that $H(\rho) < \frac{\log_2(D)}{2D}$ the lower bound grows exponentially with D, with the exponent asymptotically approaching $H_b(\gamma)D$.

Acknowledgements

We thank Ila Fiete for helpful discussions and comments on this work, and John Murray for comments on the manuscript. This work was supported by funding from the Alfred P. Sloan Foundation under grant number FG-2021-16304. The authors were benefited by participating in the activities of the UC Davis TETRAPODS Institute of Data Science, which has been funded by NSF TRIPODS grant CCF-1934568.

References

1. Cunningham, J. P. & Byron, M. Y. Dimensionality reduction for large-scale neural recordings. *Nat. Neurosci.* **17,** 1500–1509 (2014).
2. Giusti, C., Pastalkova, E., Curto, C. & Itskov, V. Clique topology reveals intrinsic geometric structure in neural correlations. *Proc. Natl. Acad. Sci.* **112,** 13455–13460 (2015).
3. Russo, A. A. *et al.* Motor cortex embeds muscle-like commands in an untangled population response. *Neuron* **97,** 953–966 (2018).
4. Stringer, C., Pachitariu, M., Steinmetz, N., Carandini, M. & Harris, K. D. High-dimensional geometry of population responses in visual cortex. *Nature* **571,** 361–365 (2019).
5. Semedo, J. D., Zandvakili, A., Machens, C. K., Byron, M. Y. & Kohn, A. Cortical areas interact through a communication subspace. *Neuron* **102,** 249–259 (2019).
6. Bernardi, S. *et al.* The geometry of abstraction in the hippocampus and prefrontal cortex. *Cell* **183,** 954–967 (2020).
7. Chung, S. & Abbott, L. Neural population geometry: An approach for understanding biological and artificial neural networks. *Curr. Opin. Neurobiol.* **70,** 137–144 (2021).
8. Azeredo da Silveira, R. & Rieke, F. The geometry of information coding in correlated neural populations. *Annu. Rev. Neurosci.* **44,** 403–424 (2021).
9. Jazayeri, M. & Ostojic, S. Interpreting neural computations by examining intrinsic and embedding dimensionality of neural activity. *Curr. Opin. Neurobiol.* **70,** 113–120 (2021).
10. Kriegeskorte, N. & Wei, X.-X. Neural tuning and representational geometry. *Nat. Rev. Neurosci.* **22,** 703–718 (2021).
11. DiCarlo, J. J. & Cox, D. D. Untangling invariant object recognition. *Trends Cogn. Sci.* **11,** 333–341 (2007).
12. Sadtler, P. T. *et al.* Neural constraints on learning. *Nature* **512,** 423–426 (2014).
13. Gallego, J. A., Perich, M. G., Miller, L. E. & Solla, S. A. Neural manifolds for the control of movement. *Neuron* **94,** 978–984 (2017).
14. Chung, S., Lee, D. D. & Sompolinsky, H. Classification and geometry of general perceptual manifolds. *Physical Review X* **8,** 031003 (2018).
15. Low, R. J., Lewallen, S., Aronov, D., Nevers, R. & Tank, D. W. Probing variability in a cognitive map using manifold inference from neural dynamics. *bioRxiv,* 418939 (2018).
16. Rubin, A. *et al.* Revealing neural correlates of behavior without behavioral measurements. *Nat. Commun.* **10,** 1–14 (2019).

17. Chaudhuri, R., Gerçek, B., Pandey, B., Peyrache, A. & Fiete, I. The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep. *Nat. Neurosci.* **22,** 1512–1520 (2019).

18. Nieh, E. H. *et al.* Geometry of abstract learned knowledge in the hippocampus. *Nature* **595,** 80–84 (2021).

19. Ehrlich, D. B. & Murray, J. D. Geometry of neural computation unifies working memory and planning. *bioRxiv* (2021).

20. Gardner, R. J. *et al.* Toroidal topology of population activity in grid cells. *Nature,* 1–6 (2022).

21. Stopfer, M., Jayaraman, V. & Laurent, G. Intensity versus identity coding in an olfactory system. *Neuron* **39,** 991–1004 (2003).

22. Mazor, O. & Laurent, G. Transient dynamics versus fixed points in odor representations by locust antennal lobe projection neurons. *Neuron* **48,** 661–673 (2005).

23. Yu, B. *et al.* Gaussian-Process Factor Analysis for Low-Dimensional Single-Trial Analysis of Neural Population Activity. *J. Neurophysiol.* **102,** 614 (2009).

24. Churchland, M. M. *et al.* Neural population dynamics during reaching. *Nature* **487,** 51–56 (2012).

25. Mante, V., Sussillo, D., Shenoy, K. V. & Newsome, W. T. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature* **503,** 78–84 (2013).

26. Kaufman, M. T., Churchland, M. M., Ryu, S. I. & Shenoy, K. V. Cortical activity in the null space: permitting preparation without movement. *Nat. Neurosci.* **17,** 440–448 (2014).

27. Remington, E. D., Narain, D., Hosseini, E. A. & Jazayeri, M. Flexible sensorimotor computations through rapid reconfiguration of cortical dynamics. *Neuron* **98,** 1005–1019 (2018).

28. Sohn, H., Narain, D., Meirhaeghe, N. & Jazayeri, M. Bayesian computation through cortical latent dynamics. *Neuron* **103,** 934–947 (2019).

29. Russo, A. A. *et al.* Neural trajectories in the supplementary motor area and motor cortex exhibit distinct geometries, compatible with different classes of computation. *Neuron* **107,** 745–758 (2020).

30. Libby, A. & Buschman, T. J. Rotational dynamics reduce interference between sensory and memory representations. *Nat. Neurosci.* **24,** 715–726 (2021).

31. Recanatesi, S. *et al.* Predictive learning as a network mechanism for extracting low-dimensional latent space representations. *Nat. Commun.* **12,** 1–13 (2021).

32. Gao, P. & Ganguli, S. On simplicity and complexity in the brave new world of large-scale neuroscience. *Curr. Opin. Neurobiol.* **32,** 148–155 (2015).

33. Roweis, S. T. & Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *Science* **290,** 2323–2326 (2000).

34. Park, M. *et al.* Bayesian manifold learning: the locally linear latent variable model (LL-LVM). *Adv Neural Inf Process Syst.* **28** (2015).

35. Hubel, D. H. & Wiesel, T. N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* **160,** 106–154 (1962).

36. Kiang, N., Sachs, M. B. & Peake, W. Shapes of tuning curves for single auditory-nerve fibers. *J. Acoust. Soc. Am* **42,** 1341–1342 (1967).

37. Hubel, D. H. & Wiesel, T. N. Receptive fields and functional architecture of monkey striate cortex. *J. Physiol.* **195,** 215–243 (1968).

38. Jones, J. P. & Palmer, L. A. The two-dimensional spatial structure of simple receptive fields in cat striate cortex. *J. Neurophysiol.* **58,** 1187–1211 (1987).

39. Ringach, D. L. Mapping receptive fields in primary visual cortex. *J. Physiol.* **558,** 717–728 (2004).

40. Dayan, P. & Abbott, L. F. *Theoretical neuroscience: computational and mathematical modeling of neural systems* (MIT press, 2005).

41. O'Keefe, J. & Burgess, N. Geometric determinants of the place fields of hippocampal neurons. *Nature* **381,** 425–428 (1996).

42. Sharp, P. E., Blair, H. T. & Cho, J. The anatomical and computational basis of the rat head-direction cell signal. *Trends Neurosci.* **24,** 289–294 (2001).

43. Moser, E. I., Kropff, E. & Moser, M.-B. Place cells, grid cells, and the brain's spatial representation system. *Annu. Rev. Neurosci.* **31,** 69–89 (2008).

44. Nieder, A. & Miller, E. K. Coding of cognitive magnitude: Compressed scaling of numerical information in the primate prefrontal cortex. *Neuron* **37,** 149–157 (2003).

45. Nieder, A. & Miller, E. K. A parieto-frontal network for visual numerical information in the monkey. *Proc. Natl. Acad. Sci.* **101,** 7457–7462 (2004).

46. Hanks, T. D. *et al.* Distinct relationships of parietal and prefrontal cortices to evidence accumulation. *Nature* **520,** 220–223 (2015).

47. O'Keefe, J. & Nadel, L. *The hippocampus as a cognitive map* (Oxford university press, 1978).

48. Hafting, T., Fyhn, M., Molden, S., Moser, M.-B. & Moser, E. I. Microstructure of a spatial map in the entorhinal cortex. *Nature* **436,** 801–806 (2005).

49. Harvey, C. D., Coen, P. & Tank, D. W. Choice-specific sequences in parietal cortex during a virtual-navigation decision task. *Nature* **484,** 62–68 (2012).

50. Moser, E. I. *et al.* Grid cells and cortical representation. *Nat. Rev. Neurosci.* **15,** 466–481 (2014).

51. Aronov, D., Nevers, R. & Tank, D. W. Mapping of a non-spatial dimension by the hippocampal–entorhinal circuit. *Nature* **543,** 719–722 (2017).

52. Macke, J. H. *et al.* Empirical models of spiking in neural populations. *Adv Neural Inf Process Syst.* **24** (2011).

53. Skaggs, W., Knierim, J., Kudrimoti, H. & McNaughton, B. A model of the neural basis of the rat's sense of direction. *Adv Neural Inf Process Syst.* **7** (1994).

54. Taube, J. S. The head direction signal: origins and sensory-motor integration. *Annu. Rev. Neurosci.* **30,** 181–207 (2007).

55. Beckermann, B. & Townsend, A. On the singular values of matrices with displacement structure. *SIAM J. Matrix Anal. Appl.* **38,** 1227–1248 (2017).

56. Gao, P. *et al.* A theory of multineuronal dimensionality, dynamics and measurement. *bioRxiv,* 214262 (2017).

57. Tenenbaum, J. B., Silva, V. d. & Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science* **290,** 2319–2323 (2000).

58. Rust, N. C., Schwartz, O., Movshon, J. A. & Simoncelli, E. P. Spatiotemporal elements of macaque v1 receptive fields. *Neuron* **46,** 945–956 (2005).

59. Gray, R. M. Toeplitz and circulant matrices: A review (2006).

60. Chaudhuri, R., Bernacchia, A. & Wang, X.-J. A diversity of localized timescales in network activity. *elife* **3,** e01239 (2014).

61. Farrell, M., Recanatesi, S., Reid, R. C., Mihalas, S. & Shea-Brown, E. Autoencoder networks extract latent variables and encode these variables in their connectomes. *Neural Networks* **141,** 330–343 (2021).

62. Grenander, U. & Szegö, G. *Toeplitz forms and their applications* (Univ of California Press, 1958).

63. Gray, R. On the asymptotic eigenvalue distribution of Toeplitz matrices. *IEEE Trans. Inf. Theory* **18,** 725–730 (1972).

64. Zhu, Z. & Wakin, M. B. On the asymptotic equivalence of circulant and Toeplitz matrices. *IEEE Trans. Inf. Theory* **63,** 2975–2992 (2017).

65. Antognini, J. & Sohl-Dickstein, J. PCA of high dimensional random walks with comparison to neural network training. *Advances in Neural Information Processing Systems* **31** (2018).

66. Henry, G. H., Dreher, B. & Bishop, P. Orientation specificity of cells in cat striate cortex. *J. Neurophysiol.* **37,** 1394–1409 (1974).

67. Folland, G. B. & Sitaram, A. The uncertainty principle: a mathematical survey. *J Fourier Anal Appl* **3,** 207–238 (1997).

68. Donoho, D. L. & Stark, P. B. Uncertainty principles and signal recovery. *SIAM Journal on Applied Mathematics* **49,** 906–931 (1989).

69. Dembo, A., Cover, T. M., Thomas, J. A., *et al.* Information theoretic inequalities. *IEEE Transactions on Information theory* **37,** 1501–1518 (1991).

70. Wigderson, A. & Wigderson, Y. The uncertainty principle: variations on a theme. *Bulletin of the American Mathematical Society* **58,** 225–261 (2021).

71. Peyrache, A., Lacroix, M. M., Petersen, P. C. & Buzsáki, G. Internally organized mechanisms of the head direction sense. *Nat. Neurosci.* **18,** 569–575 (2015).

72. Jung, M. W., Wiener, S. I. & McNaughton, B. L. Comparison of spatial firing characteristics of units in dorsal and ventral hippocampus of the rat. *J. Neurosci.* **14,** 7347–7356 (1994).

73. Kjelstrup, K. B. *et al.* Finite scale of spatial representation in the hippocampus. *Science* **321,** 140–143 (2008).

74. Lee, J. S., Briguglio, J. J., Cohen, J. D., Romani, S. & Lee, A. K. The statistical structure of the hippocampal code for space as a function of time, context, and value. *Cell* **183,** 620–635 (2020).

75. Young, M. P. & Yamane, S. Sparse population coding of faces in the inferotemporal cortex. *Science* **256,** 1327–1331 (1992).

76. Quiroga, R. Q., Kreiman, G., Koch, C. & Fried, I. Sparse but not 'grandmother-cell'coding in the medial temporal lobe. *Trends in cognitive sciences* **12,** 87–91 (2008).

77. Salinas, E. & Abbott, L. F. A model of multiplicative neural responses in parietal cortex. *Proc. Natl. Acad. Sci.* **93,** 11956–11961 (1996).

78. DeAngelis, G. C., Ohzawa, I. & Freeman, R. Spatiotemporal organization of simple-cell receptive fields in the cat's striate cortex. I. General characteristics and postnatal development. *J. Neurophysiol.* **69,** 1091–1117 (1993).

79. Lindeberg, T. A computational theory of visual receptive fields. *Biological cybernetics* **107,** 589–635 (2013).

80. Cover, T. M. & Thomas, J. A. Information theory and statistics. *Elements of information theory* **1,** 279–335 (1991).

81. Altan, E., Solla, S. A., Miller, L. E. & Perreault, E. J. Estimating the dimensionality of the manifold underlying multi-electrode neural recordings. *PLoS Comput. Biol.* **17,** e1008591 (2021).

82. Abbott, L. F., Rajan, K. & Sompolinsky, H. Interactions between intrinsic and stimulus-evoked activity in recurrent neural networks. *The dynamic brain: an exploration of neuronal variability and its functional significance,* 1–16 (2011).

83. Mazzucato, L., Fontanini, A. & La Camera, G. Stimuli reduce the dimensionality of cortical activity. *Front. Syst. Neurosci.* **10,** 11 (2016).

84. Litwin-Kumar, A., Harris, K. D., Axel, R., Sompolinsky, H. & Abbott, L. Optimal degrees of synaptic connectivity. *Neuron* **93,** 1153–1164 (2017).

85. Recanatesi, S., Bradde, S., Balasubramanian, V., Steinmetz, N. A. & Shea-Brown, E. A scale-dependent measure of system dimensionality. *bioRxiv* (2020).

86. Dahmen, D. *et al.* Strong coupling and local control of dimensionality across brain areas. *bioRxiv* (2020).

87. Churchland, M. M. *et al.* Stimulus onset quenches neural variability: a widespread cortical phenomenon. *Nat. Neurosci.* **13,** 369–378 (2010).

88. Duncker, L., Driscoll, L., Shenoy, K. V., Sahani, M. & Sussillo, D. Organizing recurrent network dynamics by task-computation to enable continual learning. *Advances in neural information processing systems* **33,** 14387–14397 (2020).

89. Procacia, I. *et al.* Measuring the strangeness of strange attractors. *Physica. D* **9,** 189–208 (1983).

90. Levina, E. & Bickel, P. Maximum likelihood estimation of intrinsic dimension. *Adv Neural Inf Process Syst.* **17** (2004).

91. Lombardi, G., Rozza, A., Ceruti, C., Casiraghi, E. & Campadelli, P. *Minimum neighbor distance estimators of intrinsic dimension Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (2011), 374–389.

92. Ceruti, C. *et al.* DANCo: dimensionality from angle and norm concentration. *arXiv preprint arXiv:1206.3881* (2012).

93. Wu, A., Roy, N. A., Keeley, S. & Pillow, J. W. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. *Advances in neural information processing systems* **30** (2017).

94. Pandarinath, C. *et al.* Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods* **15,** 805–815 (2018).