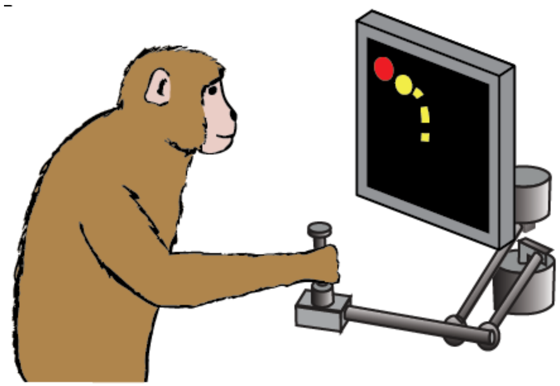
Linear and nonlinear dimensionality reduction

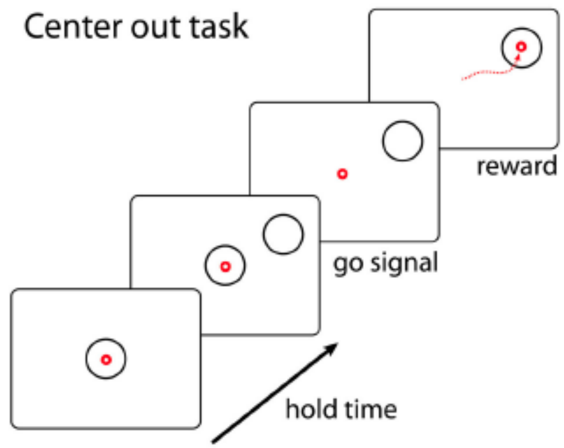
Sara A. Solla
Northwestern University

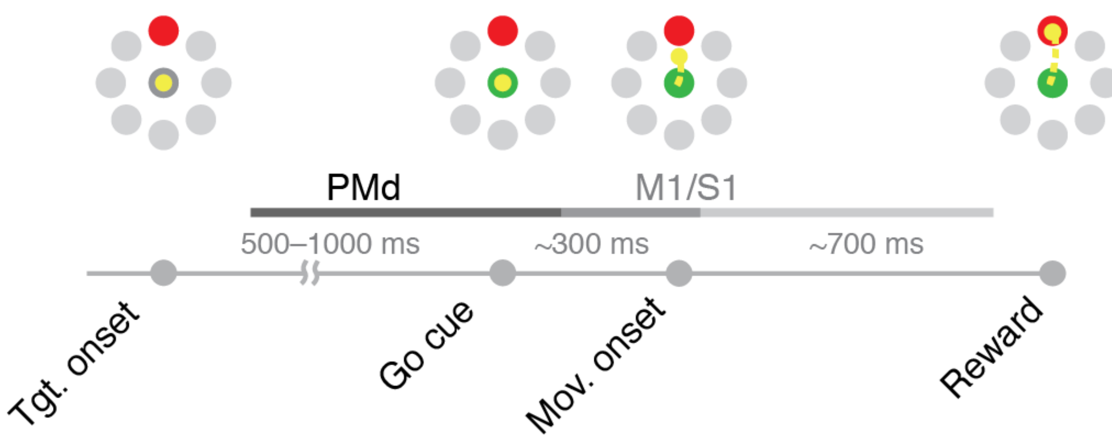
Statistical Physics of Machine Learning
Summer School
Les Houches, July 4-29, 2022

A simple motor task: center-out reaches

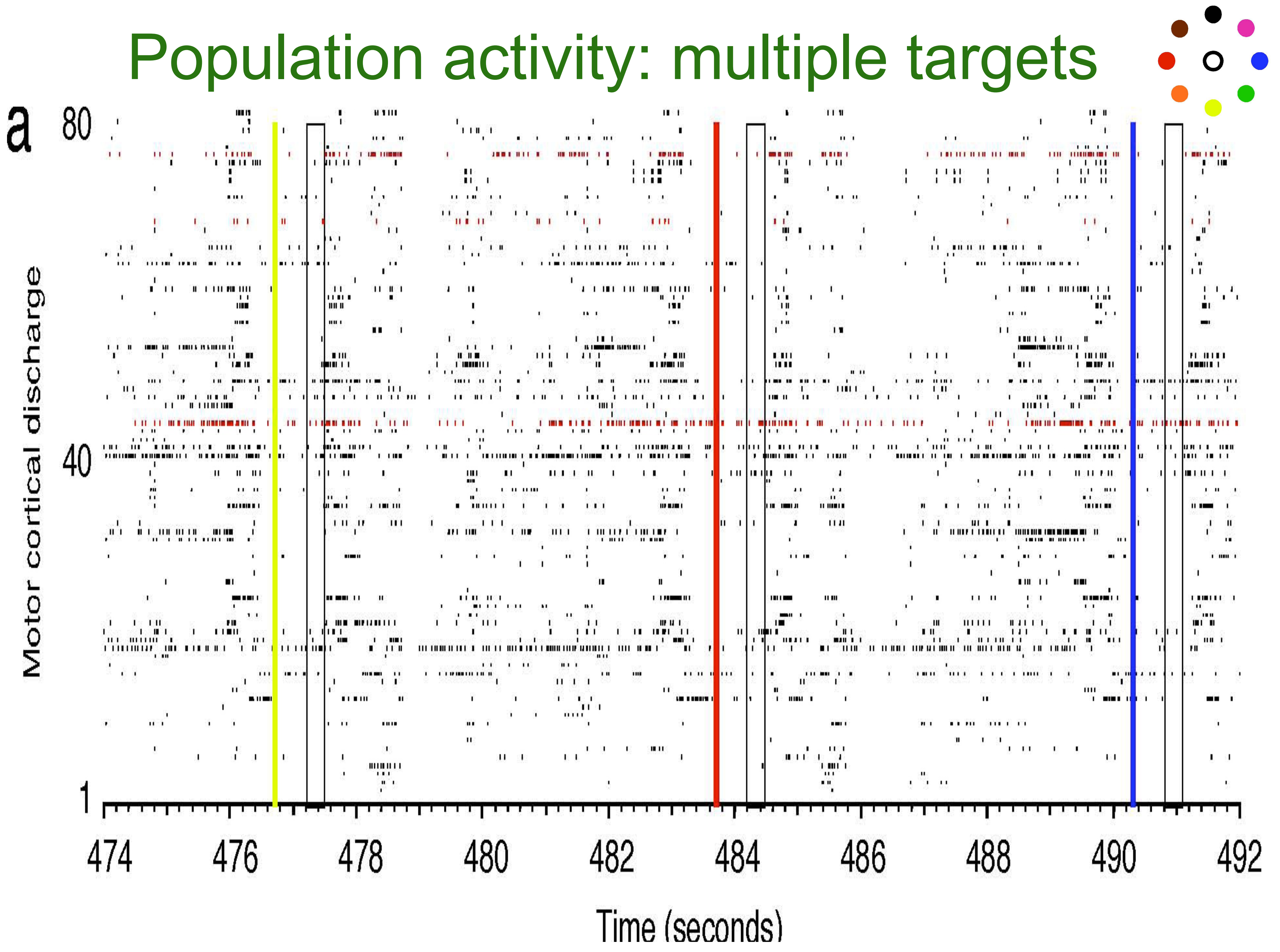
Instructed delay center-out reaching task







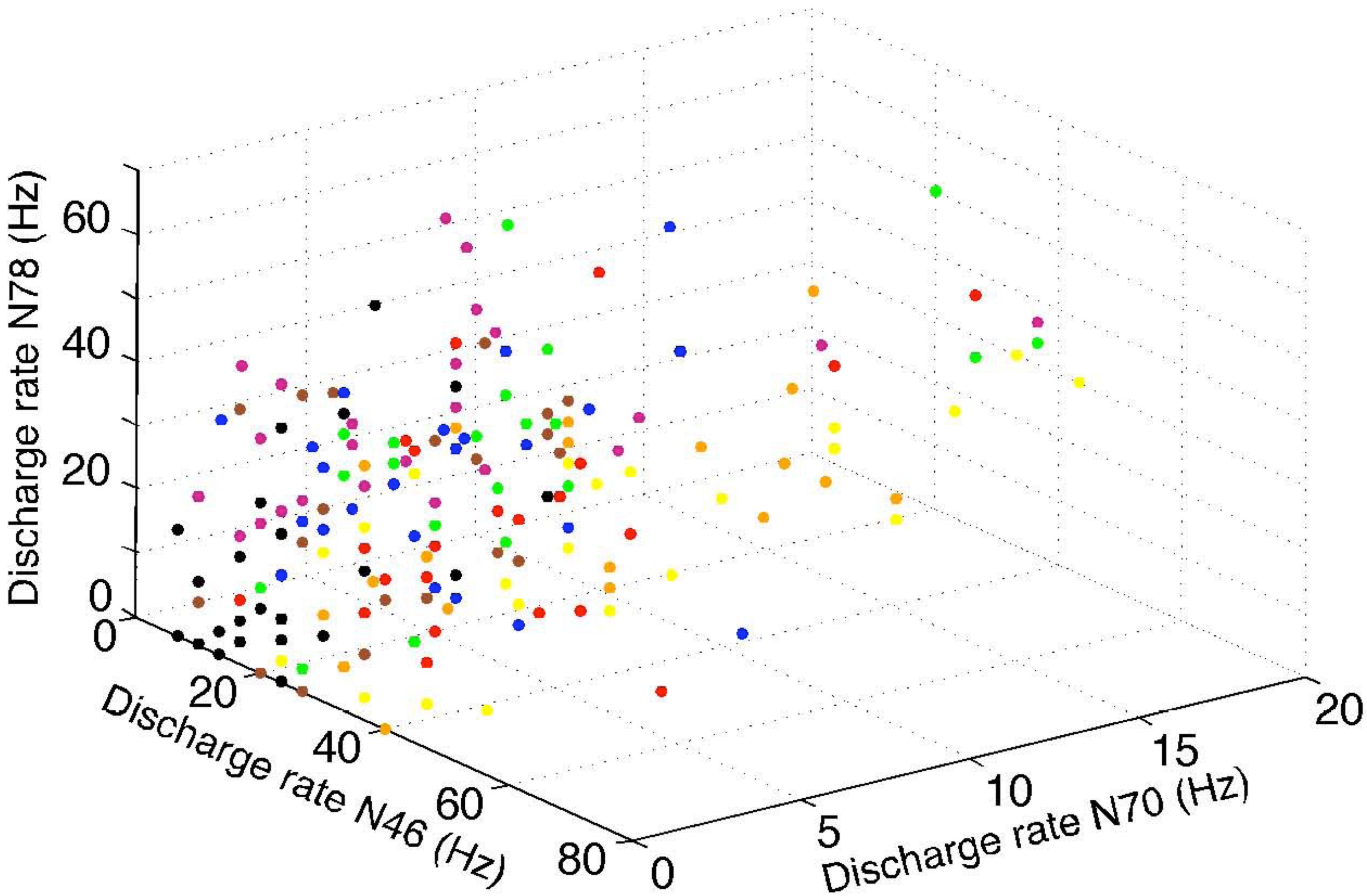
Population activity: multiple targets

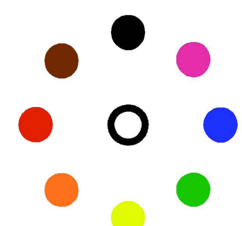


Target-dependent population activity

$$\vec{f} = (f_1, f_2, \cdots, f_N)$$

$$f_i = \frac{n_i}{\Delta}$$





Principal Components Analysis (PCA)

Consider data in the form of N-dimensional vectors. Here, the data is the N-dimensional vector of firing rates associated with each reach.

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_N \end{pmatrix}$$

Data for M reaches result in an N x M matrix:

$$\boldsymbol{X} = (\vec{x_1}|\vec{x_2}|\ldots|\vec{x_M})$$

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1M} \\ x_{21} & x_{22} & \cdots & x_{2M} \\ \vdots & \vdots & & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NM} \end{bmatrix}$$

Principal Components Analysis (PCA)

Consider data in the form on M samples of N-dimensional data

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1M} \\ x_{21} & x_{22} & \cdots & x_{2M} \\ \vdots & \vdots & & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NM} \end{bmatrix}$$

Estimate the mean firing rate of each neuron:

$$\hat{\mu}_i = \frac{1}{M} \sum_{k=1}^{M} x_{ik} \qquad 1 \leq i \leq N$$

and subtract it from the corresponding row.

Mean centered data

Principal Components Analysis (PCA)

Next, estimate the covariance of the data:

$$\hat{C} = \frac{1}{(M-1)} \; X X^T$$

$$\hat{C}_{ij} = \frac{1}{(M-1)} \sum_{k=1}^{M} (x_{ik} - \hat{\mu}_i)(x_{jk} - \hat{\mu}_j)$$

$$1 \leq i, j \leq N$$

The diagonalization of the covariance matrix yields eigenvectors and eigenvalues: the principal components

$$\widehat{C} \, \vec{u}_\nu = \lambda_\nu \, \vec{u}_\nu$$

$$1 \leq \nu \leq N$$

Principal Components Analysis (PCA): relation to Singular Value Decomposition

Consider the singular value decomposition of the data matrix X:

$$X = U\,\Sigma\,V^T$$

SVD

The columns of the N x N orthonormal matrix U provide a basis for the neural space.

The columns of the M x M orthonormal matrix V provide a basis for the space of samples.

Assume $M > N$. The N x M matrix Σ consists of an N x N diagonal block and a rectangular block of zeros of size N x $(M - N)$.

The matrix X has at most rank N; only the leading N columns of V, or N rows of V^T contribute to X.

Principal Components Analysis: relation to Singular Value Decomposition

Given the singular value decomposition of the data matrix X:

$$X = U\,\Sigma\,V^T$$

$$\begin{aligned} X\,X^T &= (U\,\Sigma\,V^T)\,(V\,\Sigma^T U^T\,) \\ &= U\,(\Sigma\,\Sigma^T)\,U^T \end{aligned}$$

$$\hat{C} = \frac{1}{(M-1)}\;X\,X^T = U\Lambda\,U^T$$

$$\Lambda = \frac{1}{(M-1)}\;\Sigma\,\Sigma^T$$

Principal Components Analysis: dimensionality reduction

$$\Lambda = \frac{1}{(M-1)} \; \Sigma \; \Sigma^T$$

$$\Lambda = \begin{bmatrix} \lambda_1 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & \lambda_K & \dots & 0 \\ 0 & \cdots & 0 & \cdots & \lambda_N \end{bmatrix}$$

with $\lambda_1 \geq \cdots \geq \lambda_K \geq \cdots \geq \lambda_N$

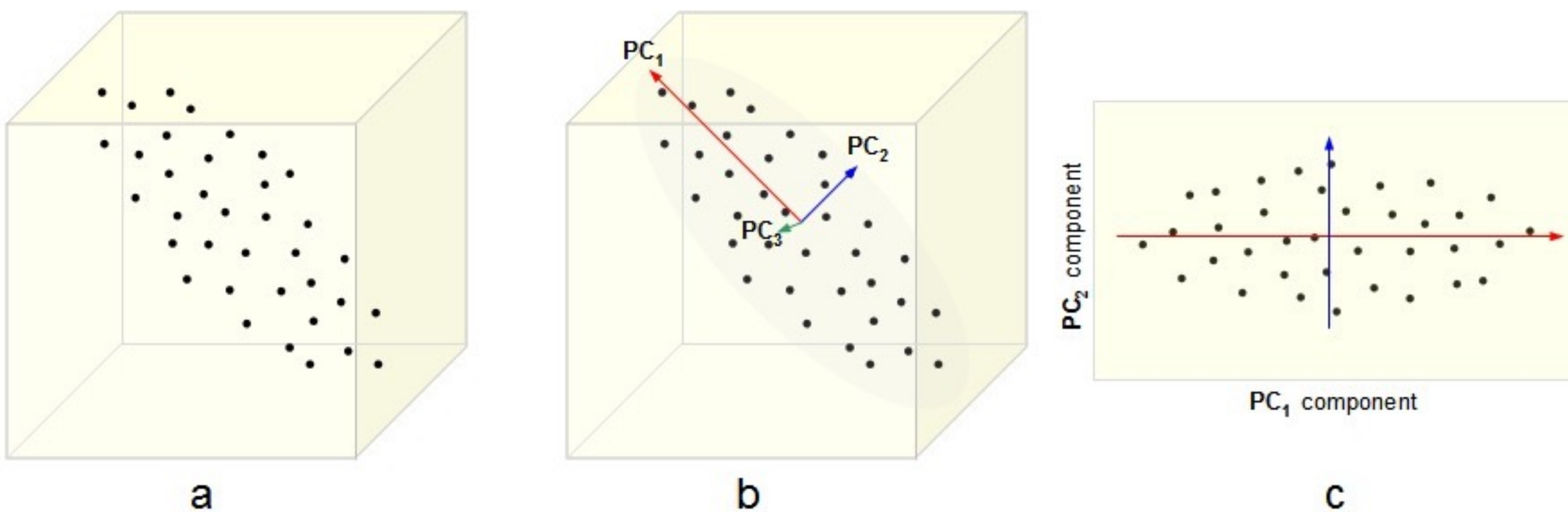
Dimensionality reduction: keep only the K leading eigenvalues

$$\hat{C} = U \Lambda \; U^T$$

$$\hat{C} = \sum_{\mu=1}^{N} \lambda_\mu \; \vec{u}_\mu \; \vec{u}_\mu^T \qquad \Longrightarrow \qquad \hat{C} = \sum_{\mu=1}^{K} \lambda_\mu \; \vec{u}_\mu \; \vec{u}_\mu^T$$

Principal Components Analysis: dimensionality reduction

$$\hat{C} = \sum_{\mu=1}^{N} \lambda_\mu \; \vec{u}_\mu \; \vec{u}_\mu^T \qquad \Longrightarrow \qquad \hat{C} = \sum_{\mu=1}^{K} \lambda_\mu \; \vec{u}_\mu \; \vec{u}_\mu^T$$



Principal Components Analysis

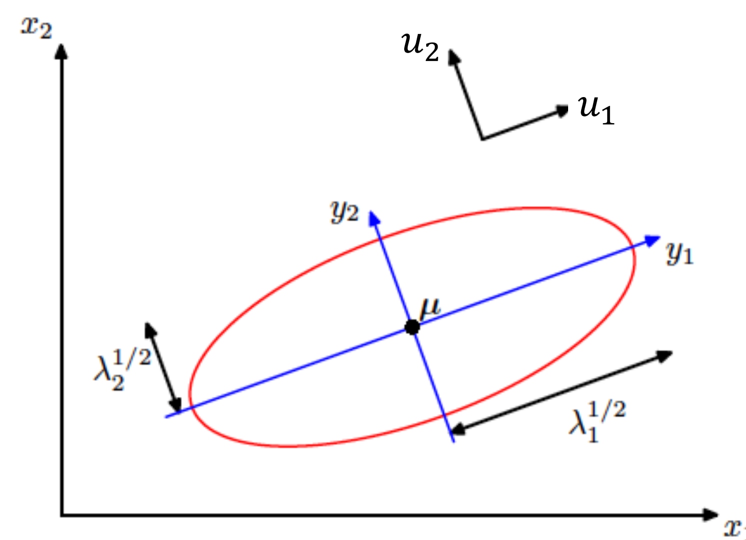
Find the eigenvalues and eigenvectors of the covariance matrix $\hat{C}$

$$\widehat{C}\,\vec{u}_\nu = \lambda_\nu\,\vec{u}_\nu \qquad\qquad 1 \le \nu \le N$$

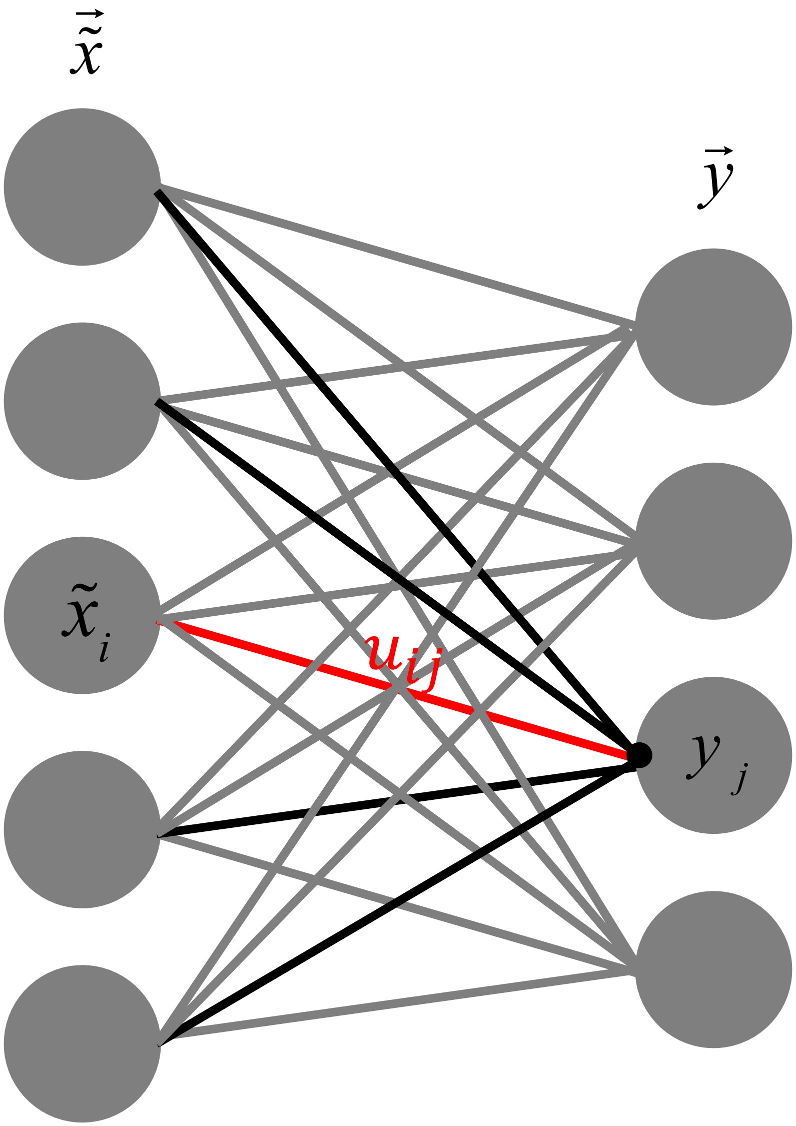
Construct the matrix U, which has the eigenvectors $\vec{u}_\nu$ as columns

Then $\hat{C} = U\Lambda\, U^T$ where Λ is the diagonal matrix of eigenvalues

The coordinates of the data points expressed in the new coordinate system are $Y = U^T\,\tilde{X}$



Principal Components as latent variables



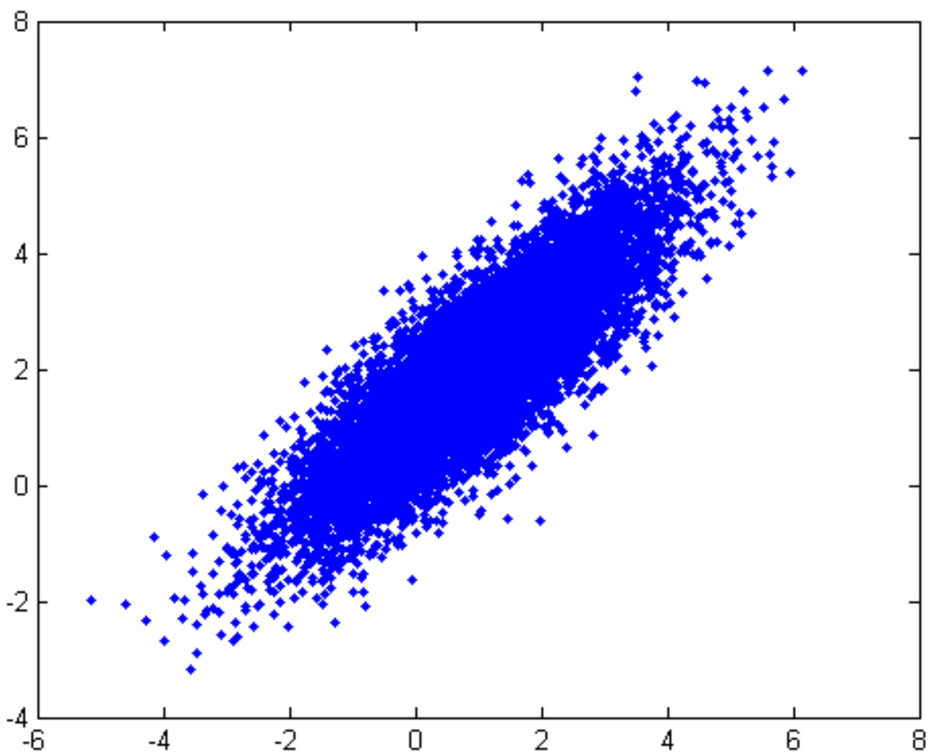
$$\tilde{X} = U\,Y \Longrightarrow \tilde{x}_i = \sum_{j=1}^{d} u_{ij}\, y_j$$

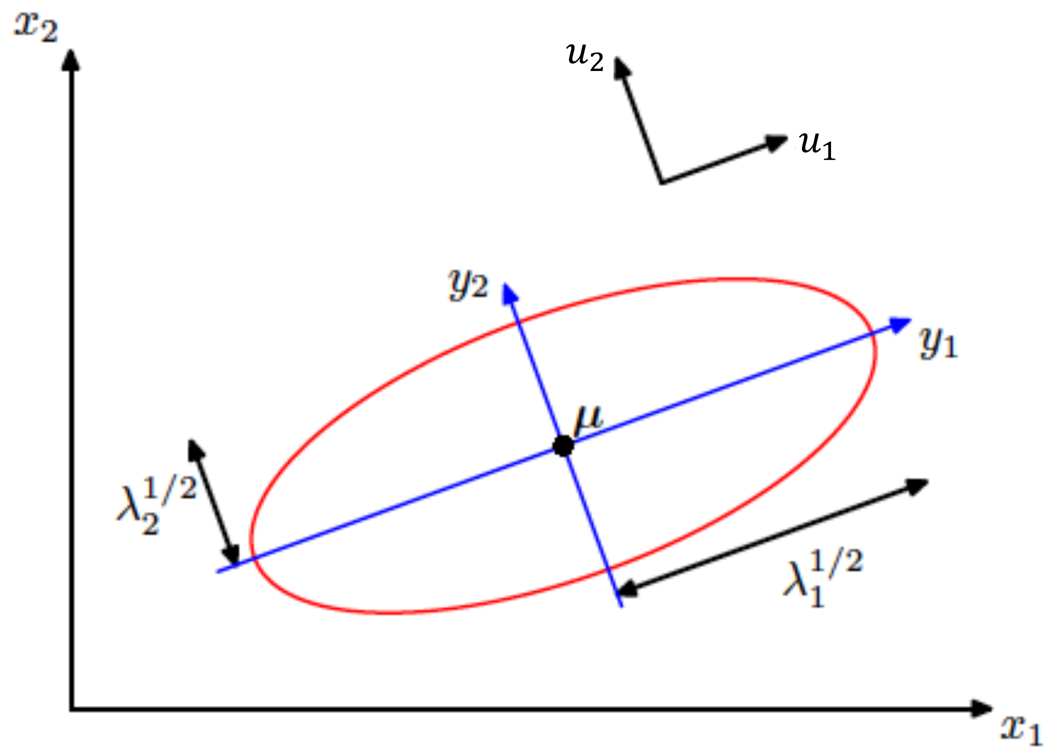
The i-th component u_{ij} of the j-th eigenvector is the "weight" from y_j to $\tilde{x}_i$

Generative model

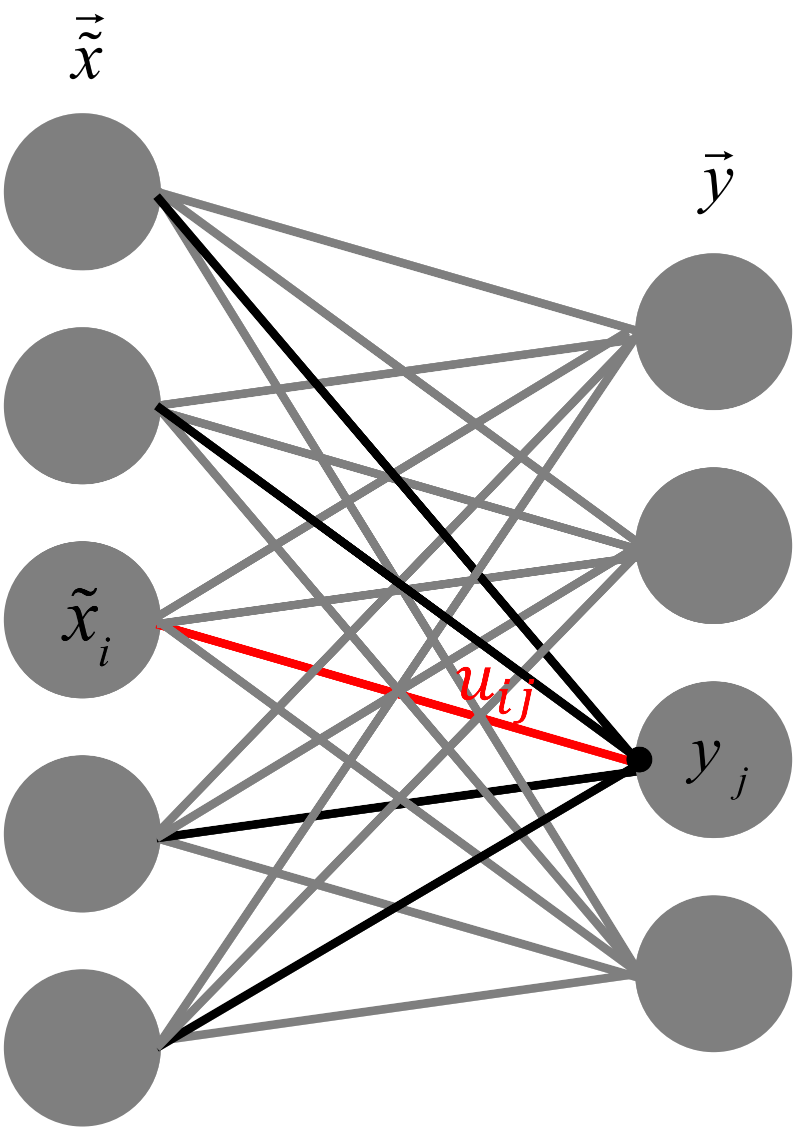
$\tilde{x}_i = \sum_{j=1}^{d} u_{ij}\, y_j$

$P\left(y_j\right) = \mathcal{N}\left(0, \lambda_j\right)$





Probabilistic PCA (PPCA)



$$P\left(y_j\right) = \mathcal{N}\left(0, \lambda_j\right)$$

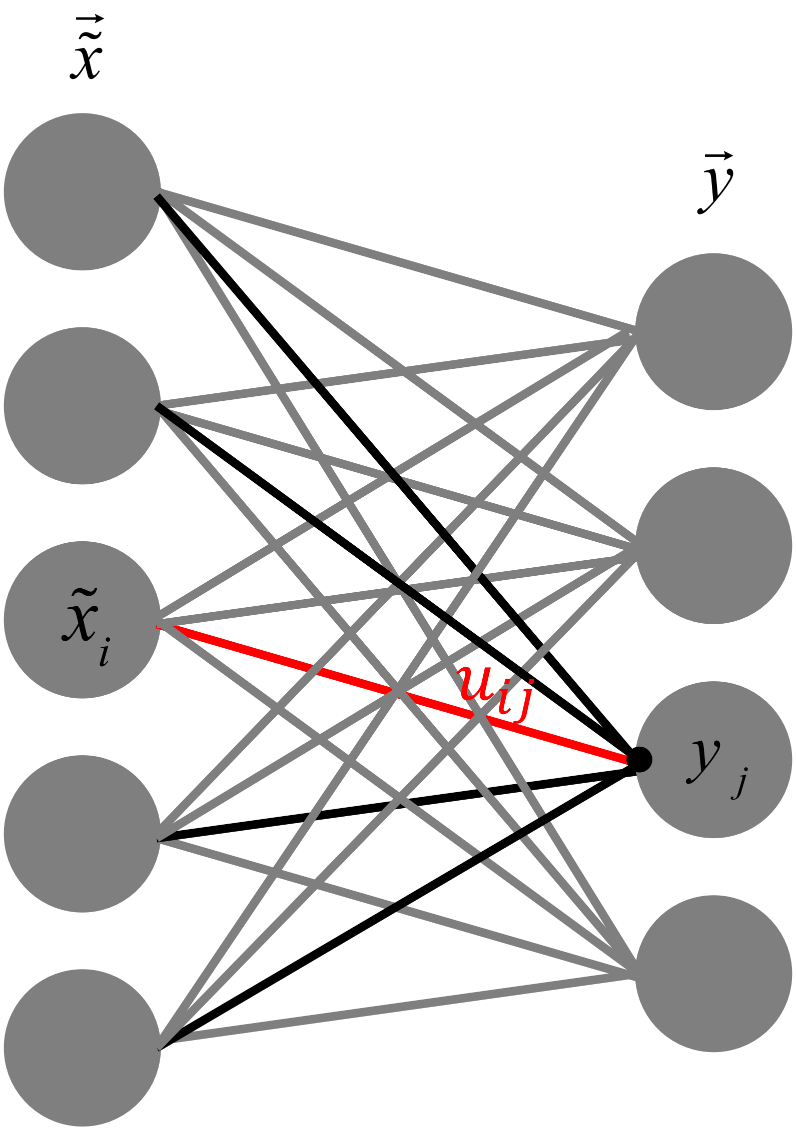
$$\tilde{x}_i = \sum_{j=1}^{d} u_{ij}\, y_j + \eta_i$$

$$P\left(\eta_i\right) = \mathcal{N}\left(0, \sigma^2\right)$$

$$P\left(\tilde{x}_i | y_j\right) = \mathcal{N}\left(u_{ij} y_j, \sigma^2\right)$$

$$P\left(\tilde{x}_i | \vec{y}\right) = \mathcal{N}\left(\sum_j u_{ij} y_j, \sigma^2\right)$$

Factors Analysis (FA)



$$P(y_j) = \mathcal{N}(0, \lambda_j)$$

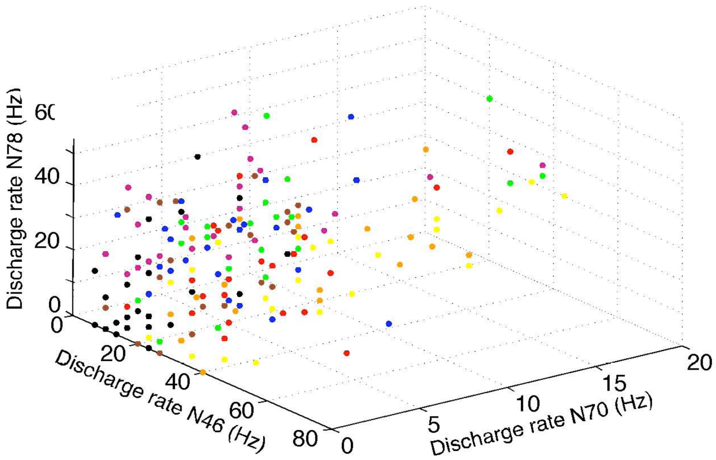
$$\tilde{x}_i = \sum_{j=1}^{d} u_{ij}\, y_j + \eta_i$$

$$P(\eta_i) = \mathcal{N}(0, \sigma_i^2)$$

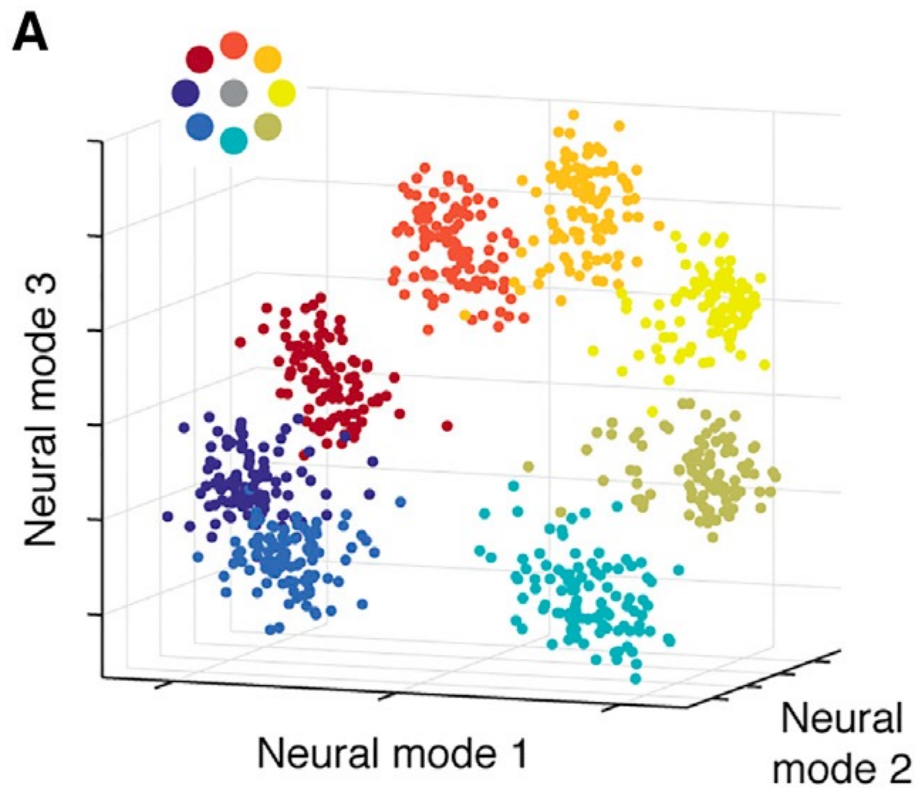
$$P(\tilde{x}_i | y_j) = \mathcal{N}(u_{ij} y_j, \sigma_i^2)$$

$$P(\tilde{x}_i | \vec{y}) = \mathcal{N}\left(\sum_j u_{ij} y_j, \sigma_i^2\right)$$

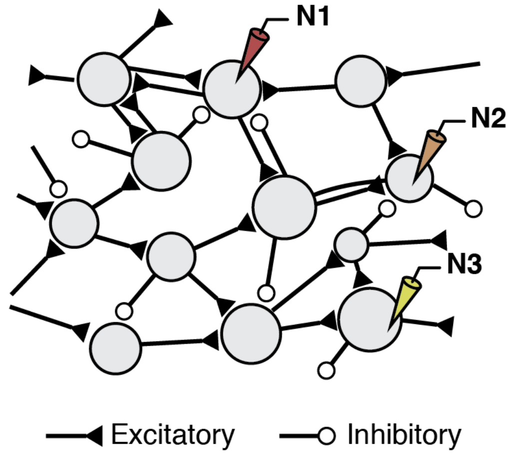
Target-dependent population activity

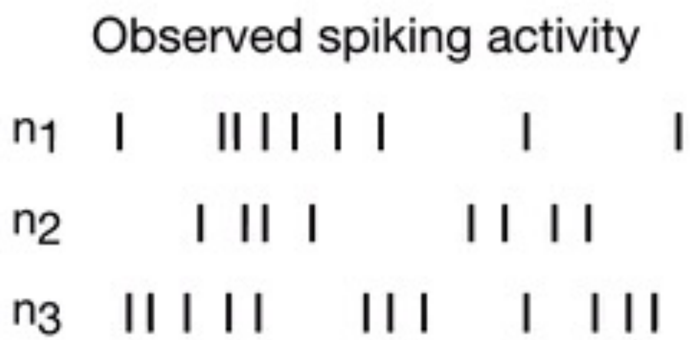


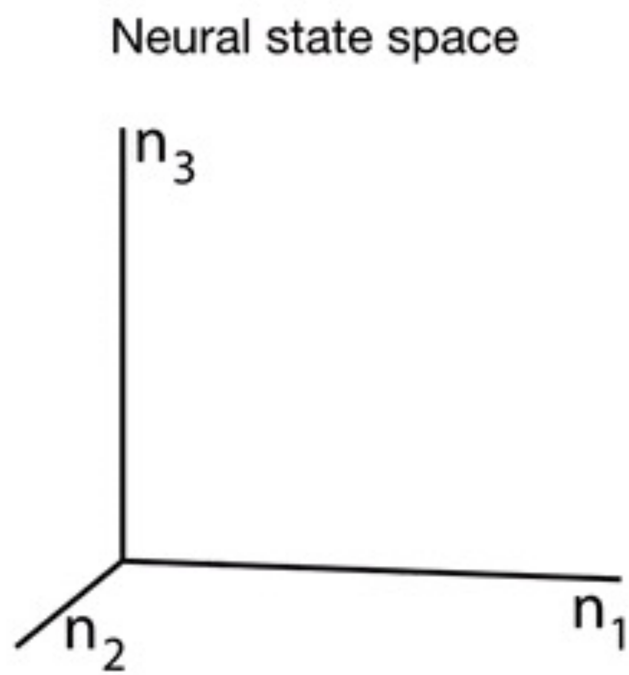
$$\vec{f} = (f_1, f_2, \ldots, f_N)$$



Population dynamics: the empirical neural space

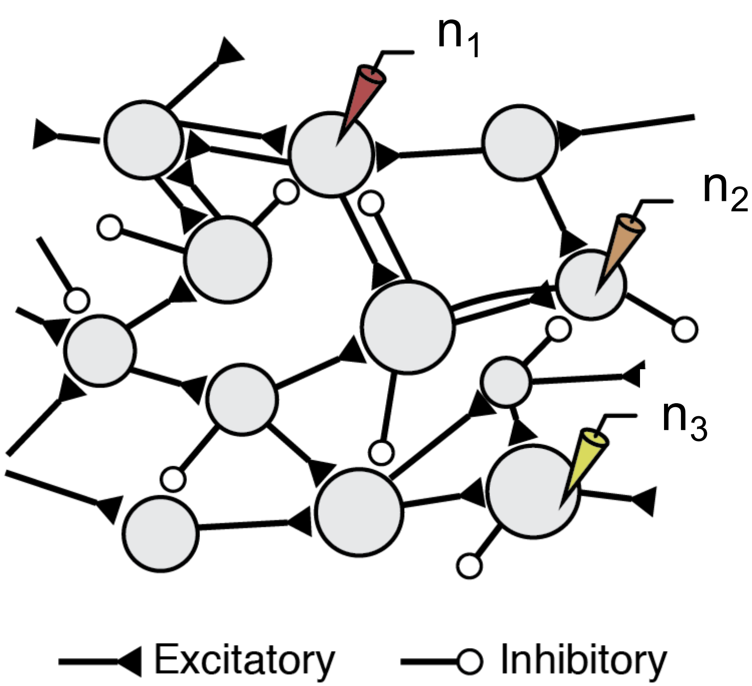


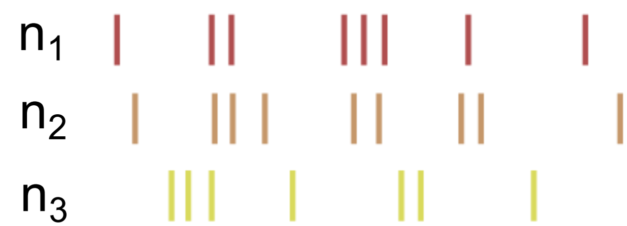


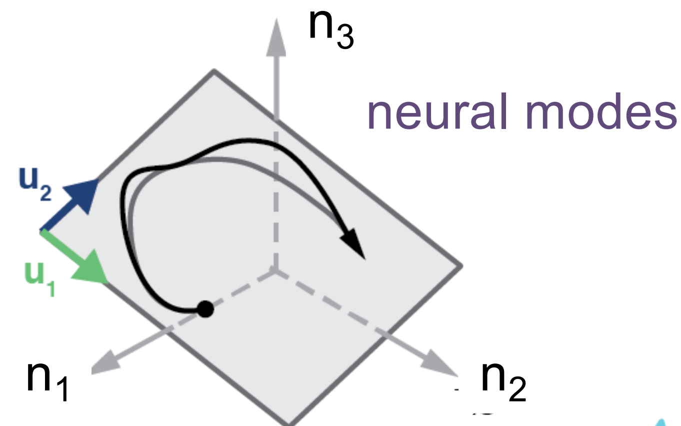


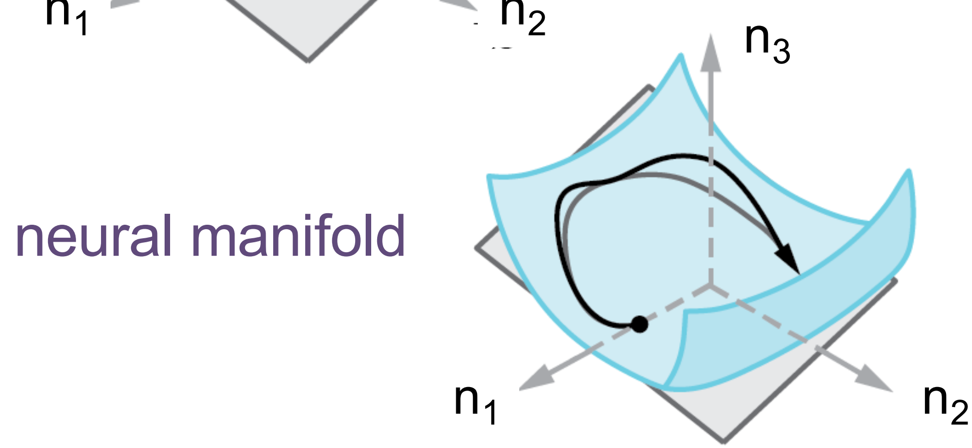
Cunningham & Yu, *Nat Neuro* 2014

Dimensionality reduction: neural modes and latent variables



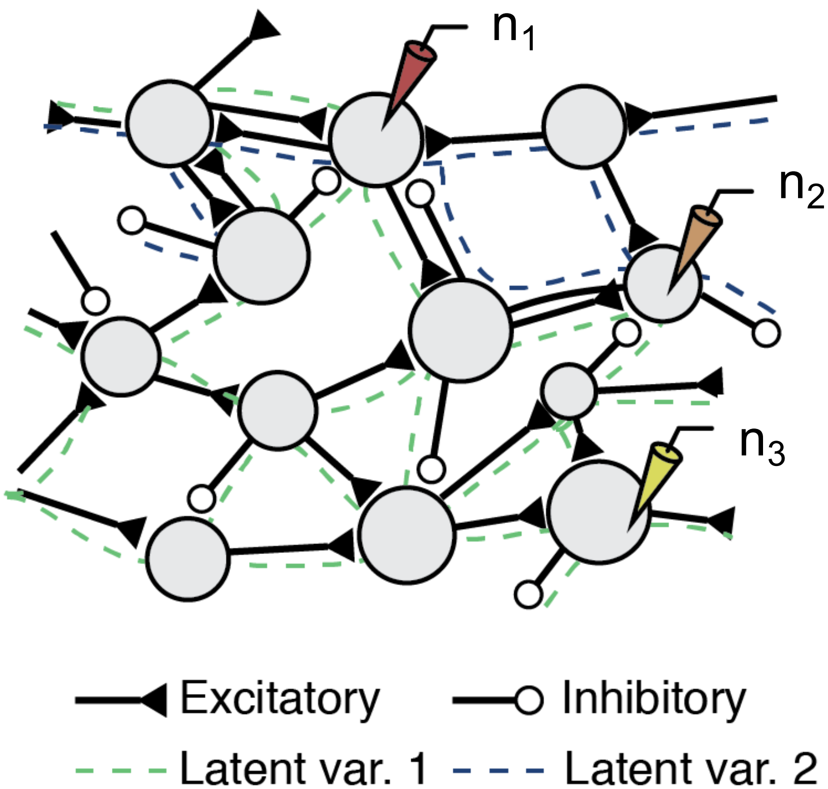


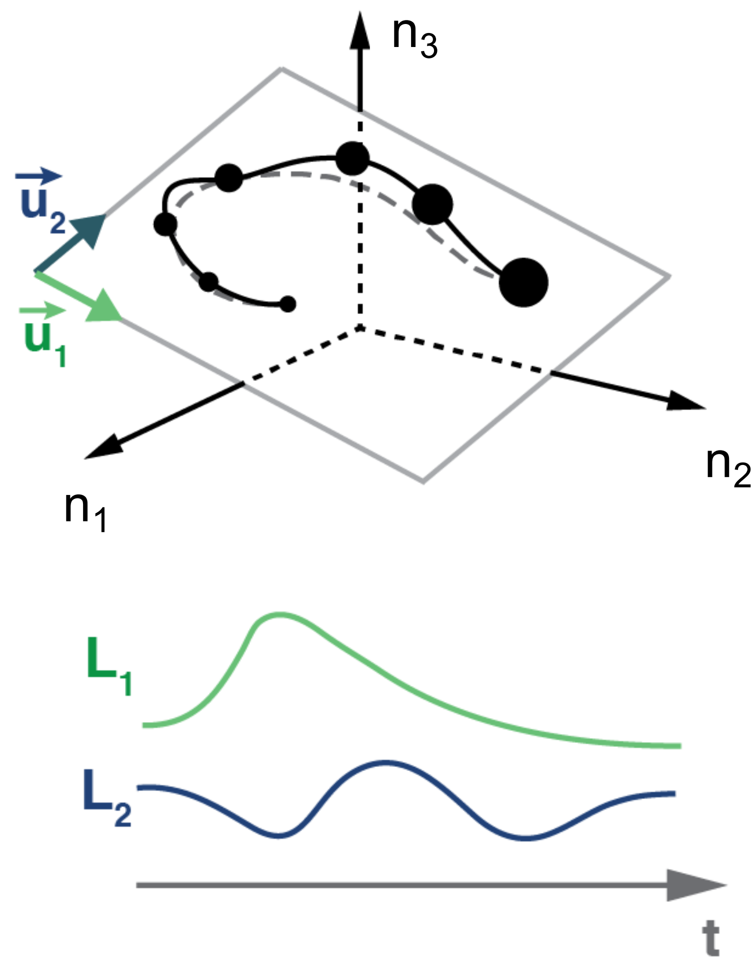




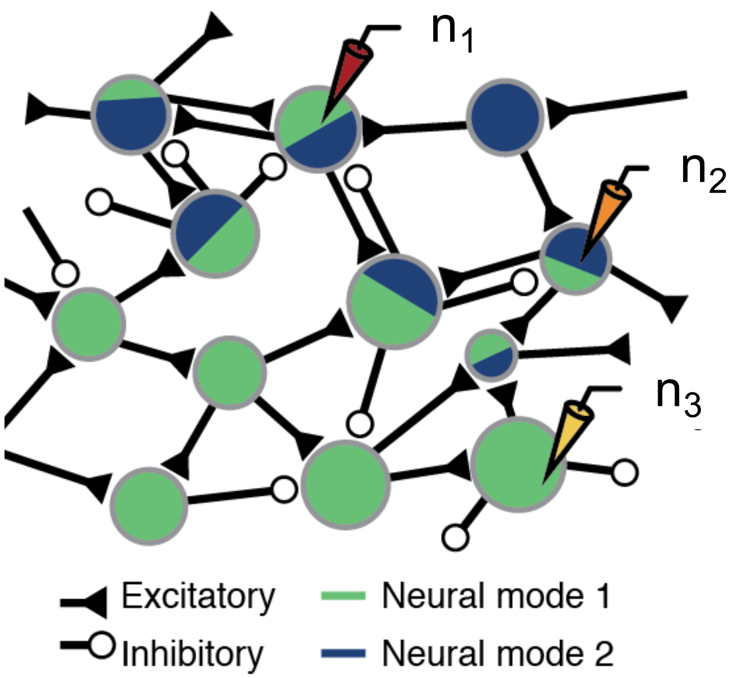
DIMENSIONALITY REDUCTION
linear or nonlinear?

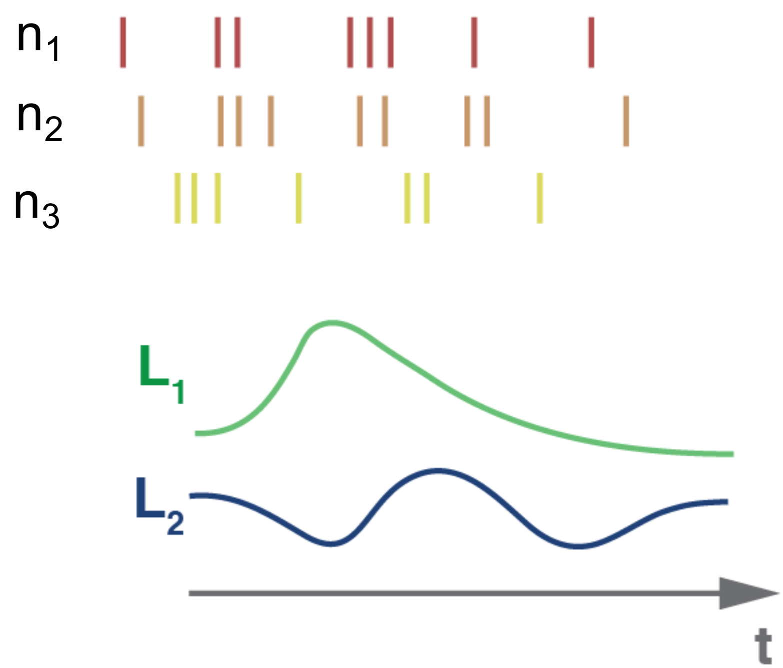
Dimensionality reduction: neural modes and latent variables

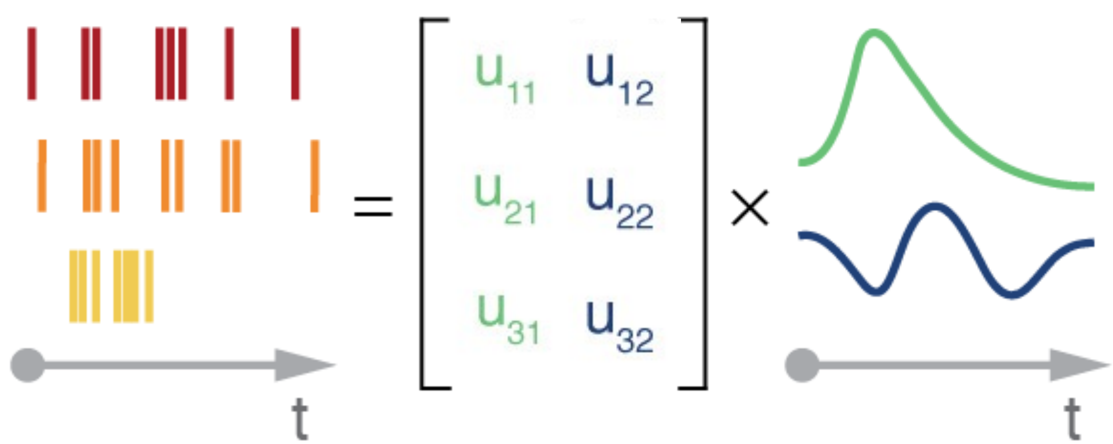




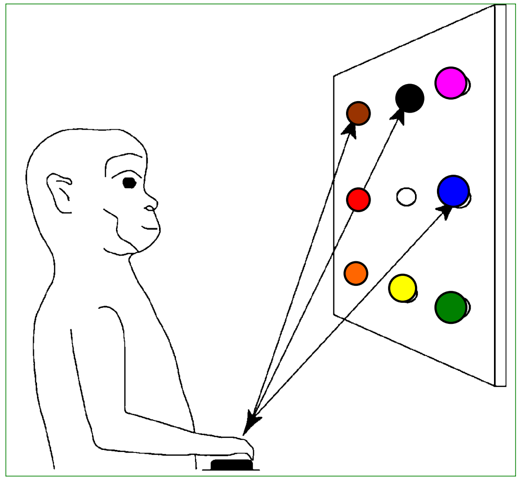
Population dynamics:
latent variables as a generative model

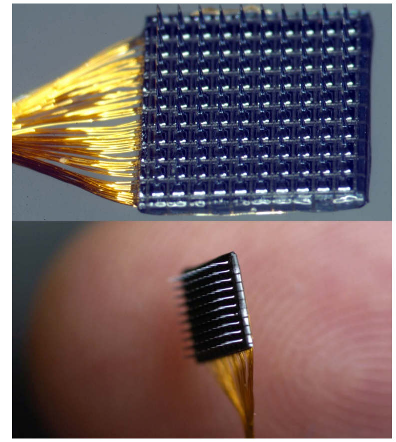


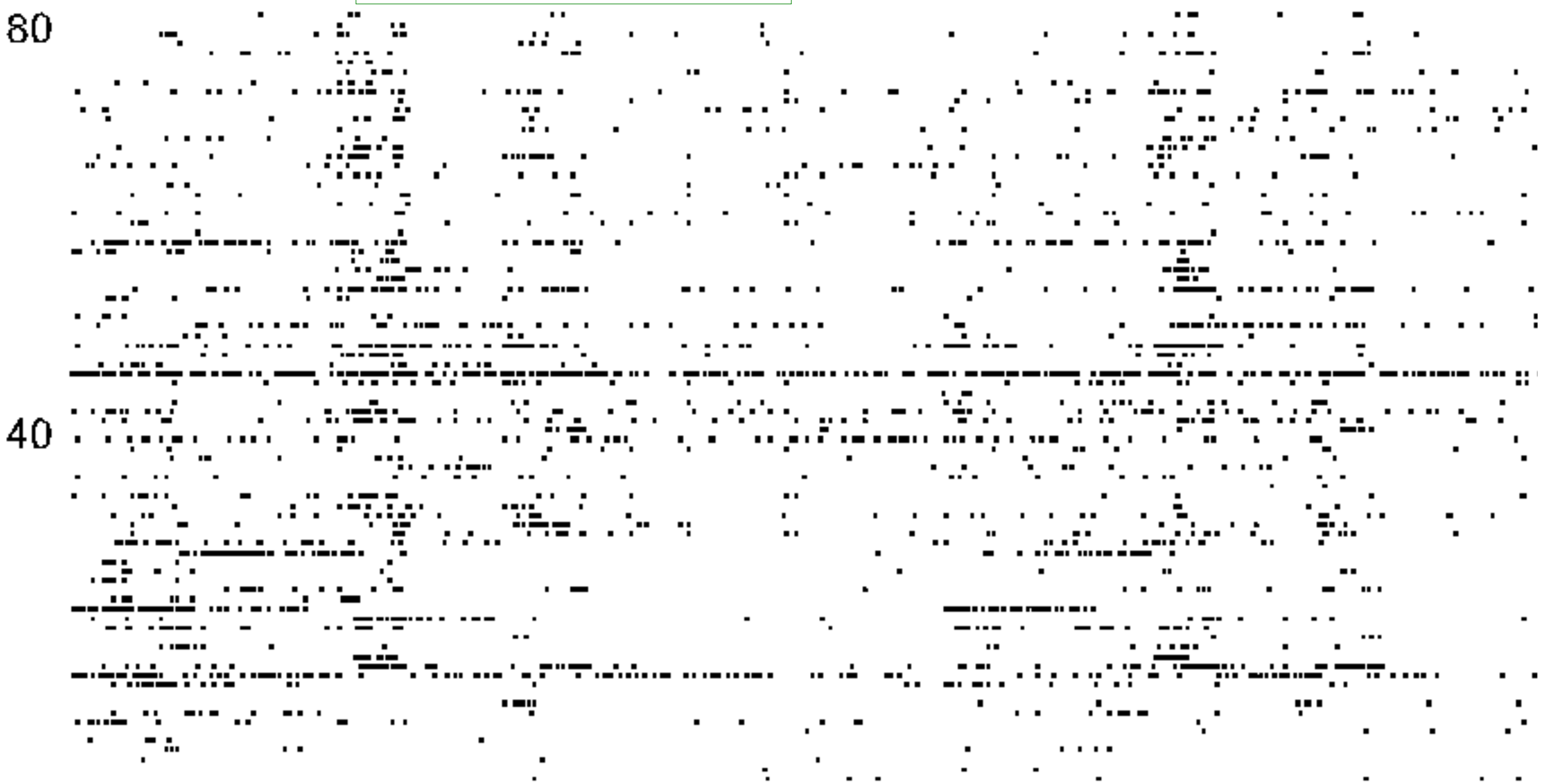




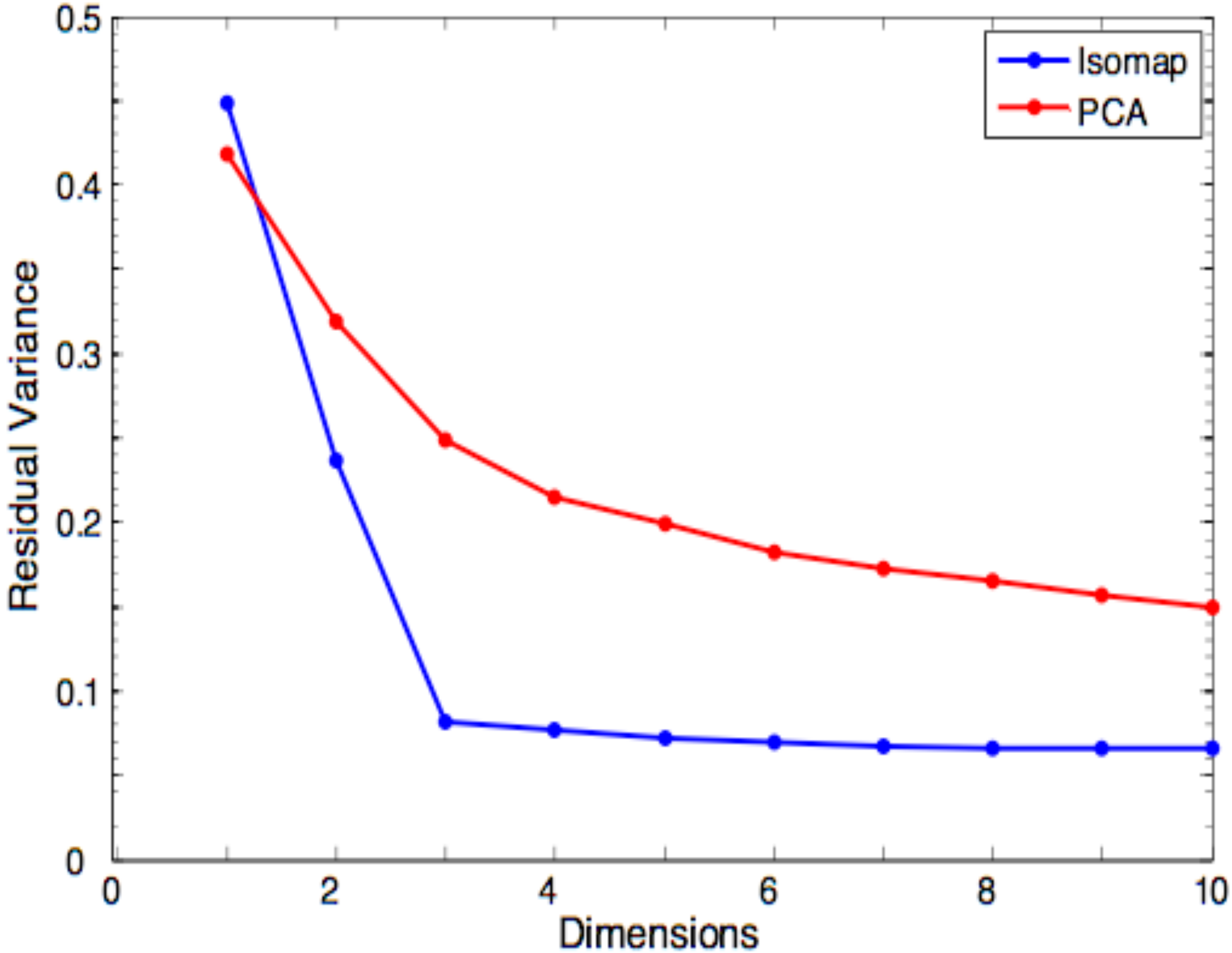
Neural recordings: center-out task



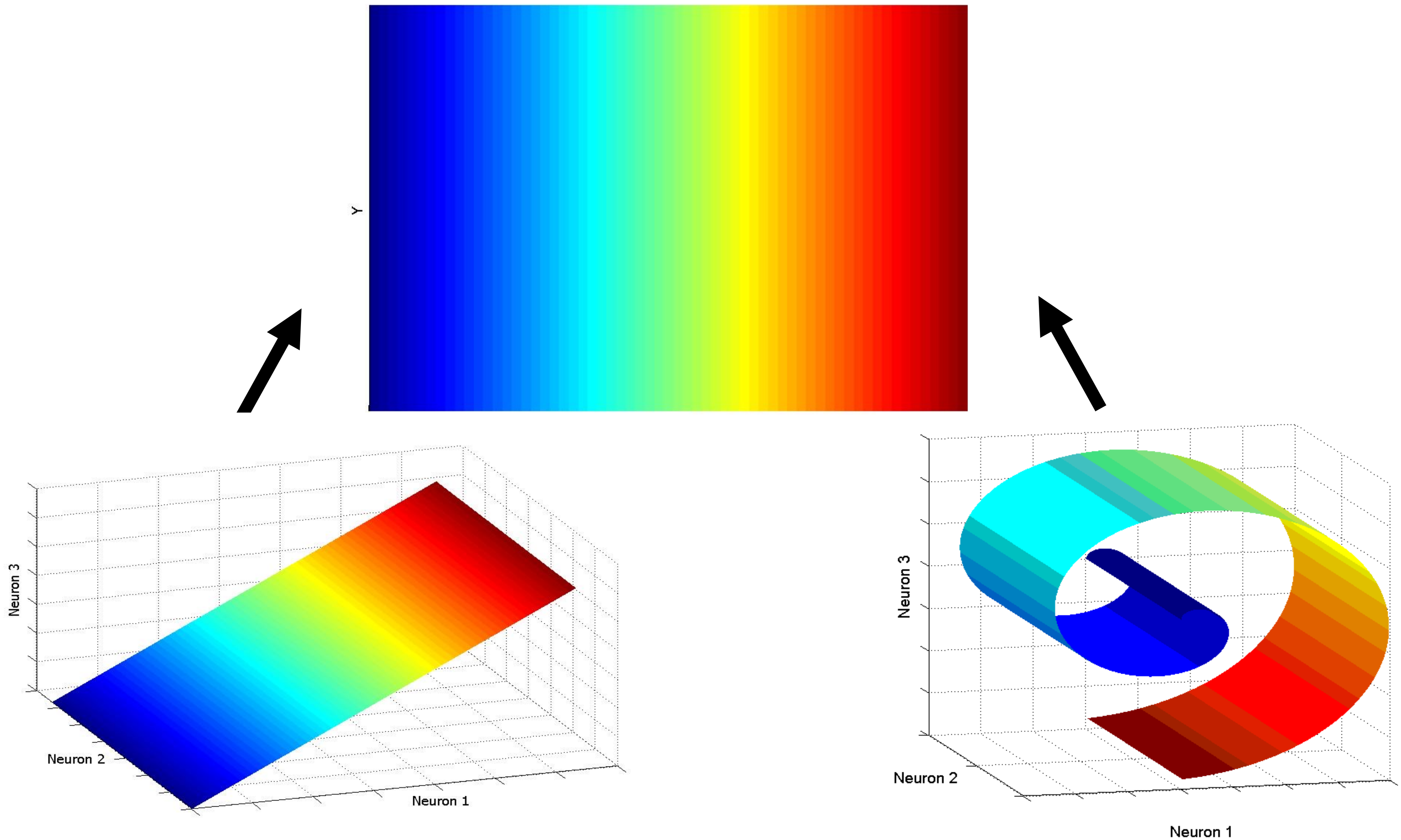




Eigenvalues



ISOMAP
nonlinear dimensionality reduction

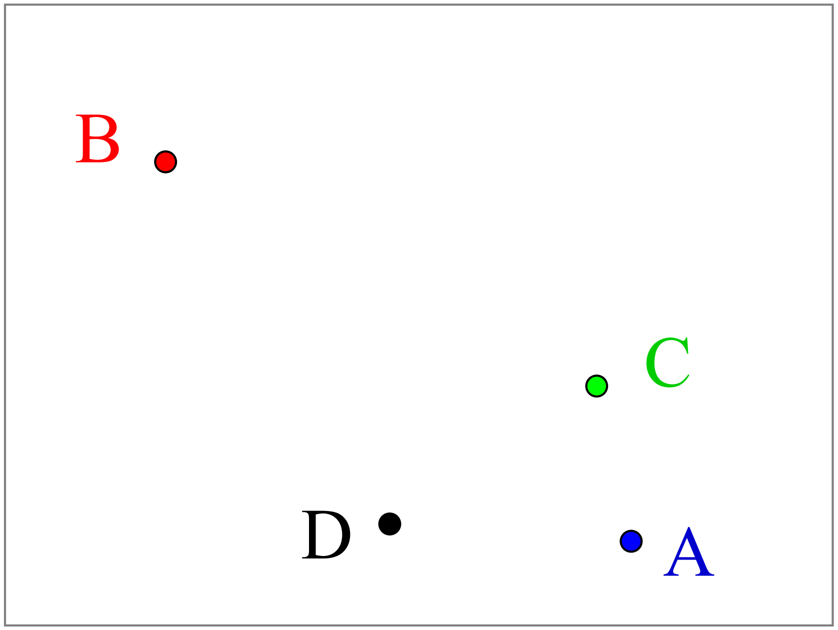


Multidimensional scaling

Represent objects as points in a low dimensional space:
Euclidean distances between the corresponding points reproduce as well as possible an empirical matrix of distances or dissimilarities.

	A	B	C	D
A	0	7	2	3
B	7	0	4.5	6
C	2	4.5	0	5
D	3	6	5	0

→



Multidimensional scaling

Consider data in the form of N-dimensional vectors. Here, the data is the N-dimensional vector of firing rates associated with each reach.

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \ldots \\ x_N \end{pmatrix}$$

Data for M reaches result in an N x M matrix:

$$\boldsymbol{X} = (\vec{x_1} | \vec{x_2} | \ldots | \vec{x_M})$$

If the matrix $\boldsymbol{X}$ is hidden from us, but we are given instead an M x M matrix $\boldsymbol{S}$ of squared distances between the points, can we reconstruct the matrix $\boldsymbol{X}$?

Multidimensional scaling

If the distances are Euclidean:

$$S_{ij} = d_{ij}^2 = (\vec{x}_i - \vec{x}_j)^T(\vec{x}_i - \vec{x}_j)$$

the scalar product between data points can be written as:

$$\vec{x}_i^T \vec{x}_j = -(1/2)(S_{ij} - ||\vec{x}_i||^2 - ||\vec{x}_j||^2)$$

In matrix form, $\vec{x}_i{}^T \vec{x}_j = (X^T X)_{ij}$

and $S_{ij} - ||\vec{x}_i||^2 - ||\vec{x}_j||^2 = (JSJ)_{ij}$

where J is the M x M centering matrix $J = I - (1/M)\, \mathbf{e}\, \mathbf{e}^T$

$$\boxed{X^T X = -(1/2) JSJ}$$

Multidimensional scaling

In matrix form: $$\boldsymbol{X}^T\boldsymbol{X} = -(1/2)\boldsymbol{JSJ}$$

From this equation the data matrix $\boldsymbol{X}$ can be easily obtained:

$$\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{U}\Lambda\boldsymbol{U}^T \Longrightarrow \boldsymbol{X} = \Lambda^{1/2}\boldsymbol{U}^T$$

- If the distance matrix to which this calculation is applied is based on Euclidean distances, this process allows us to recover the data matrix $\boldsymbol{X}$ from the distance matrix $\boldsymbol{S}$.
- A reduction of the dimensionality of the original data space follows from truncation of the number of eigenvalues from M to K, and the corresponding restriction in the number of eigenvectors used to reconstruct $\boldsymbol{X}$.
- It can be proved that this truncation is equivalent to PCA, which is based on the diagonalization of $\boldsymbol{X}\boldsymbol{X}^T$.

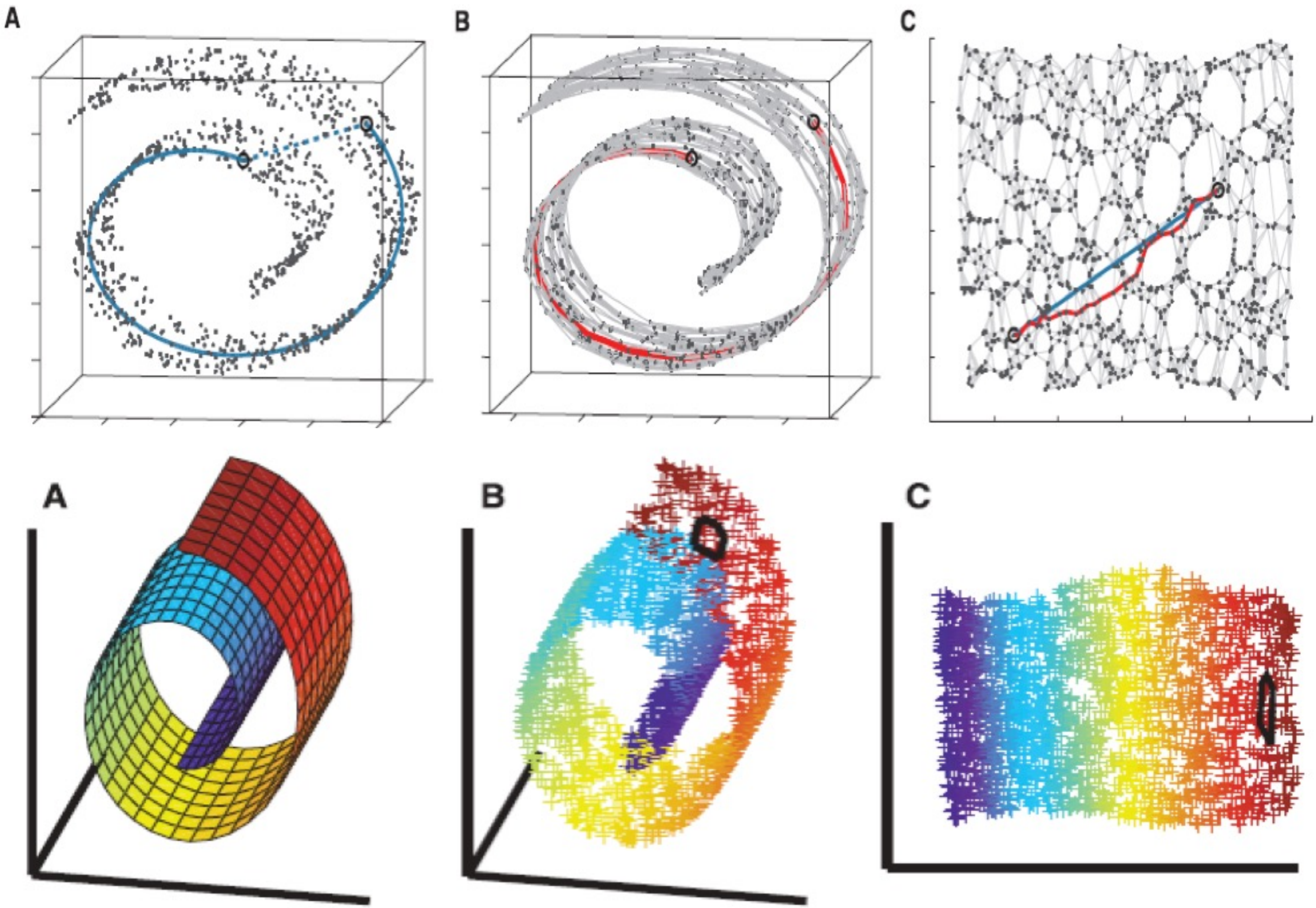
Multidimensional scaling

When applied to an arbitrary matrix ***S*** of 'squared distances', the method still implies defining an 'inner product' matrix Y through the centering operation: $Y = -(1/2)\, JSJ$, followed by the diagonalization of Y: $Y = U \Lambda U^T$ and the identification of the data matrix X as $X = \Lambda^{1/2} U^T$.

This procedure minimizes a cost function E that measures the Frobenius norm of the difference between two matrices: the original matrix Y and the inner product matrix $X^T X$ obtained from the Euclidean representation of the data:

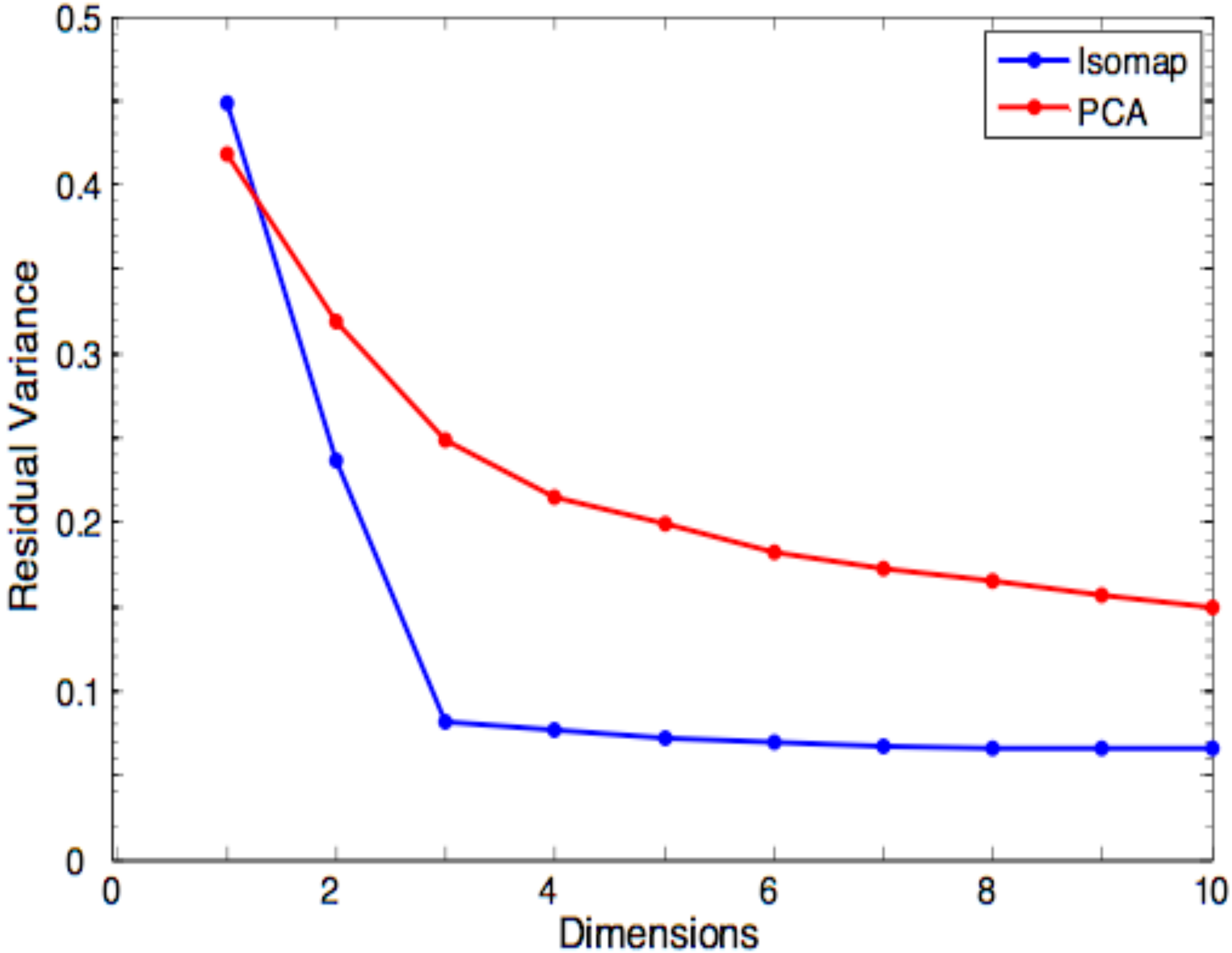
$$E(X) = \| X^T X - Y \|_F$$

ISOMAP: nonlinear embedding

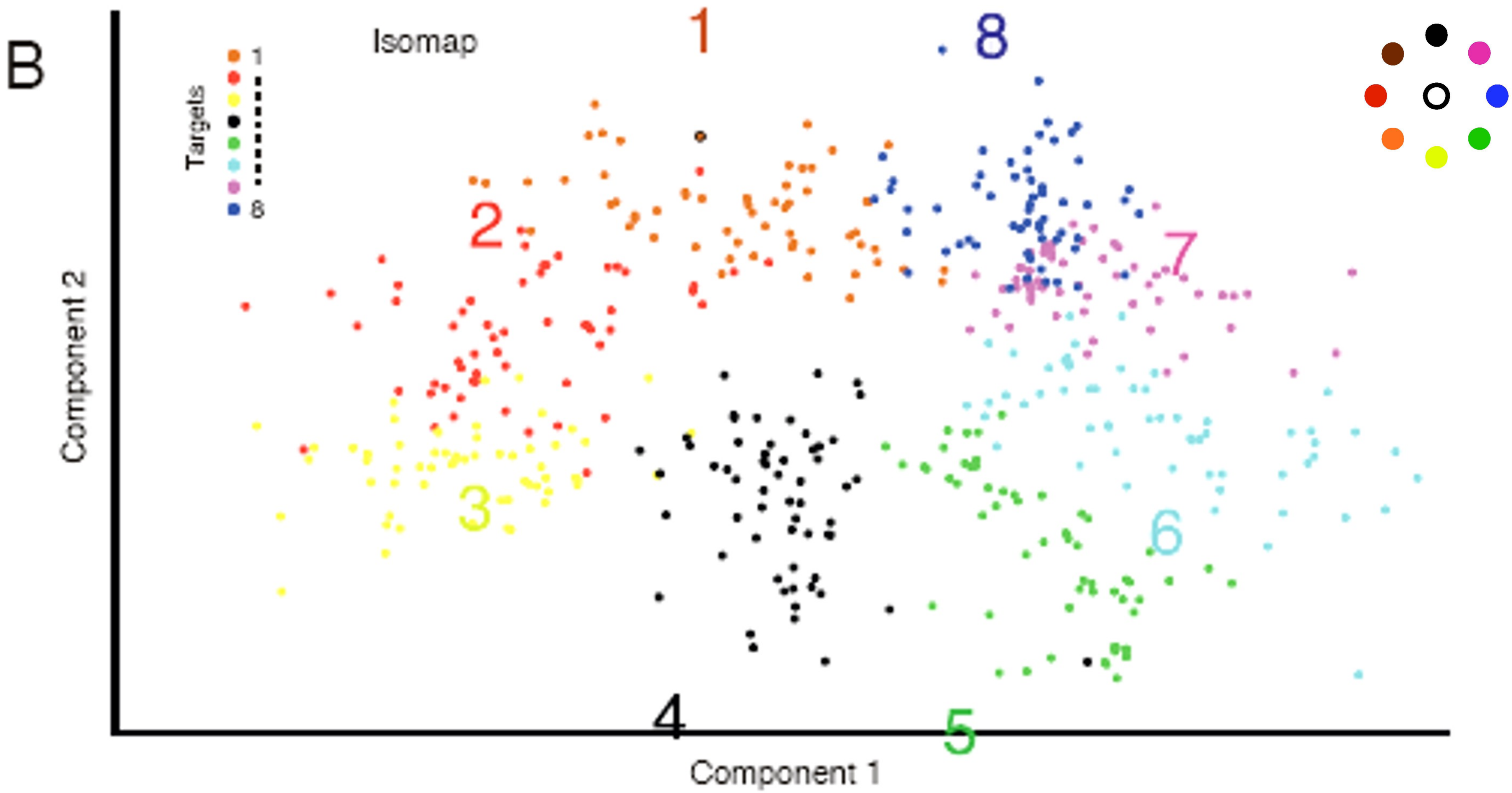


Tenenbaum, de Silva, Langford, *Science* (2000)

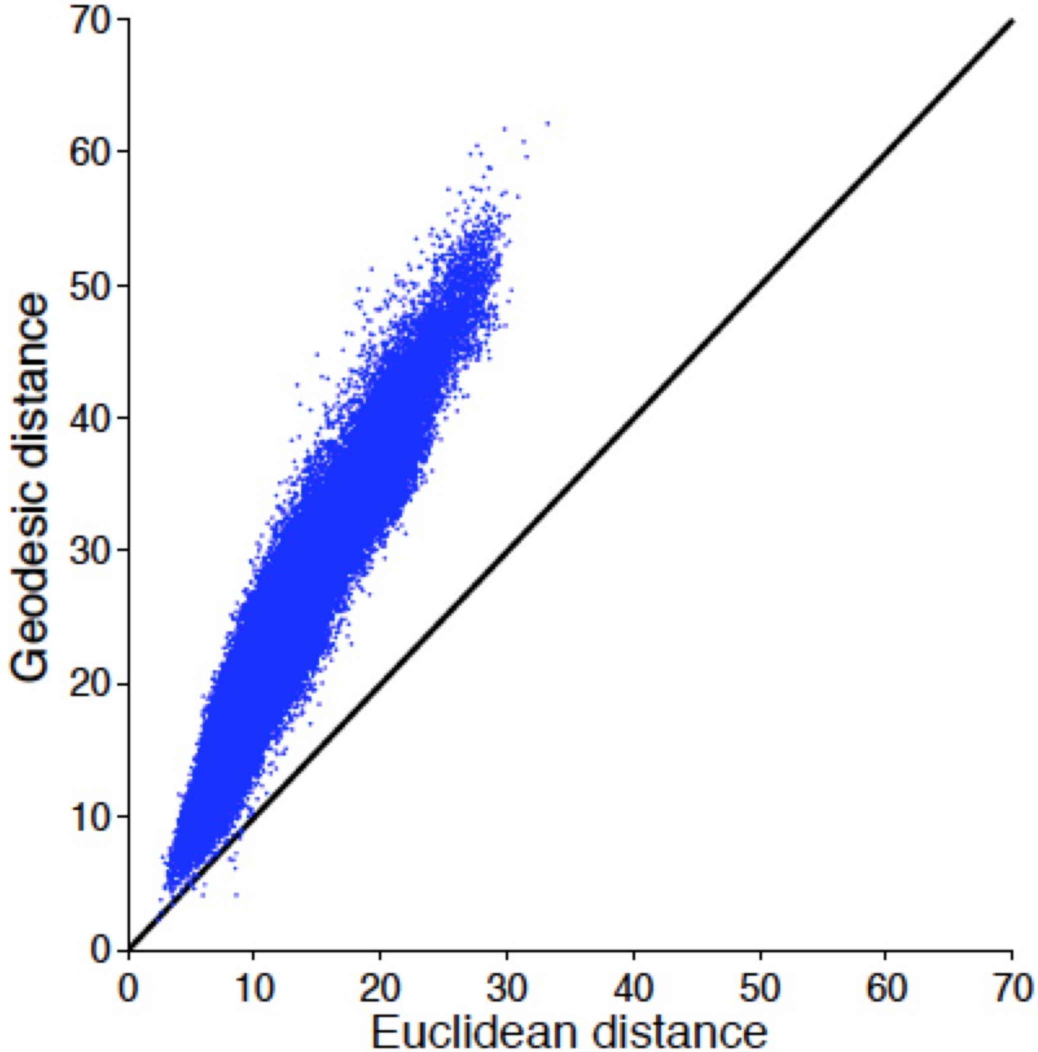
ISOMAP eigenvalues



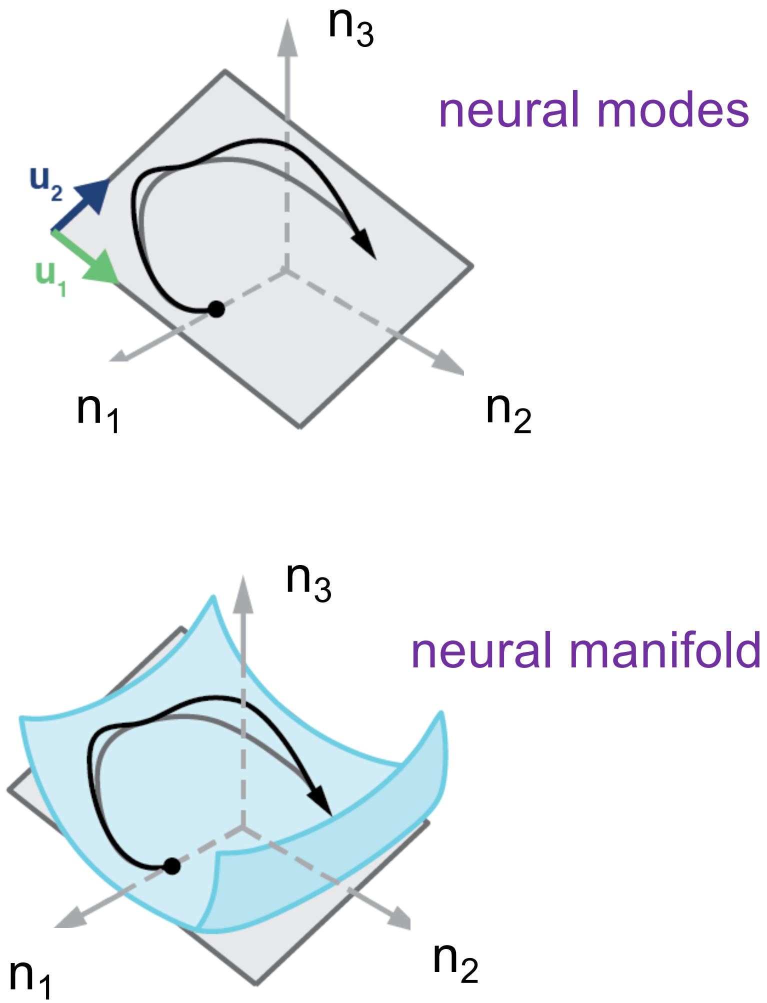
ISOMAP: two-dimensional projection

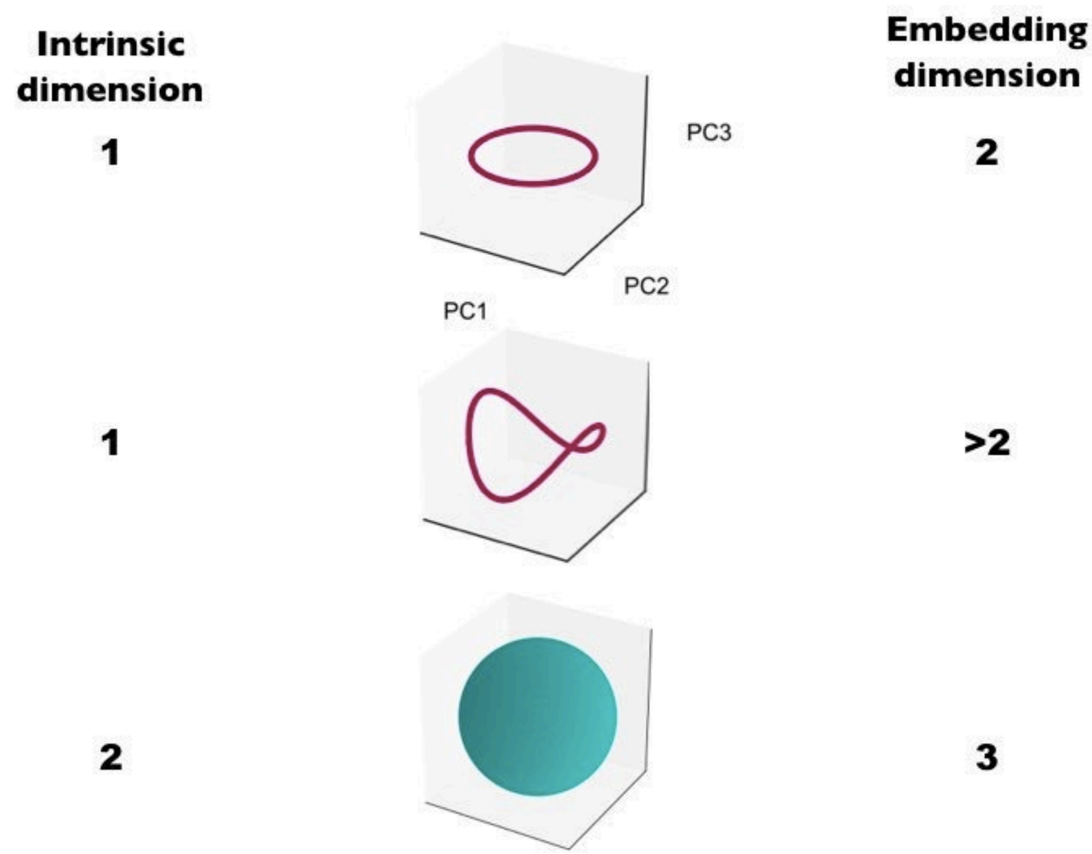


Euclidean vs Geodesic distances



Neural manifolds: linear or nonlinear?



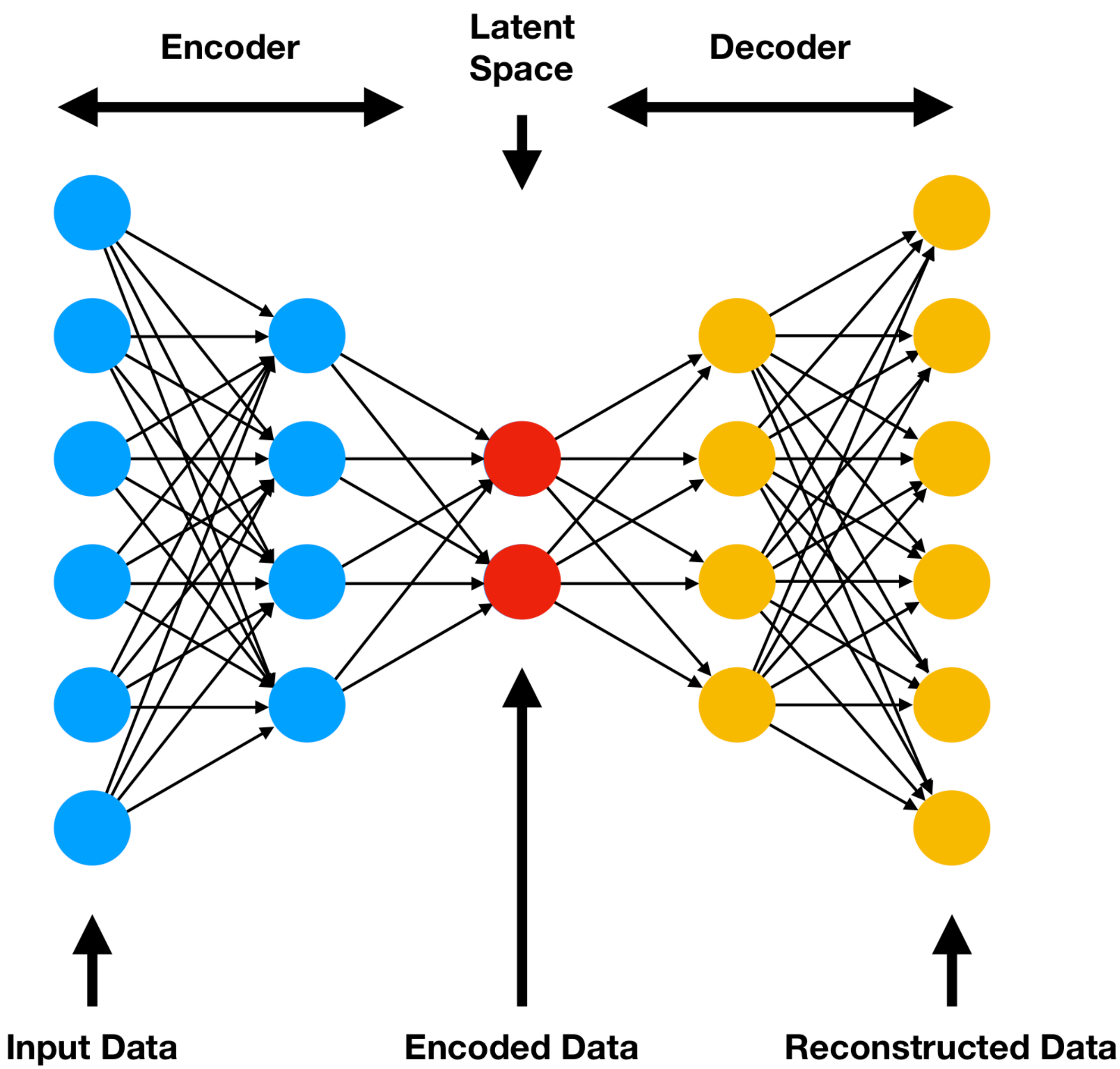


intrinsic

flat

Jazayeri, Ostojic, *Curr Opin Neurobiol* (2021)

Autoencoders



Baldi, Hornik, *Neural Netw* (1989)

Manifolds for images

Published as a conference paper at ICLR 2021

THE INTRINSIC DIMENSION OF IMAGES AND ITS IMPACT ON LEARNING

Phillip Pope[1], **Chen Zhu**[1], **Ahmed Abdelkader**[2], **Micah Goldblum**[1], **Tom Goldstein**[1]
[1]Department of Computer Science, University of Maryland, College Park
[2]Oden Institute for Computational Engineering and Sciences, University of Texas at Austin
{pepope,chenzhu}@umd.edu, akader@utexas.edu, {goldblum,tomg}@umd.edu

ABSTRACT

It is widely believed that natural image data exhibits low-dimensional structure despite the high dimensionality of conventional pixel representations. This idea underlies a common intuition for the remarkable success of deep learning in computer vision. In this work, we apply dimension estimation tools to popular datasets and investigate the role of low-dimensional structure in deep learning. We find that common natural image datasets indeed have very low intrinsic dimension relative to the high number of pixels in the images. Additionally, we find that low dimensional datasets are easier for neural networks to learn, and models solving these tasks generalize better from training to test data. Along the way, we develop a technique for validating our dimension estimation tools on synthetic data generated by GANs allowing us to actively manipulate the intrinsic dimension by controlling the image generation process. Code for our experiments may be found here.

Manifolds for images

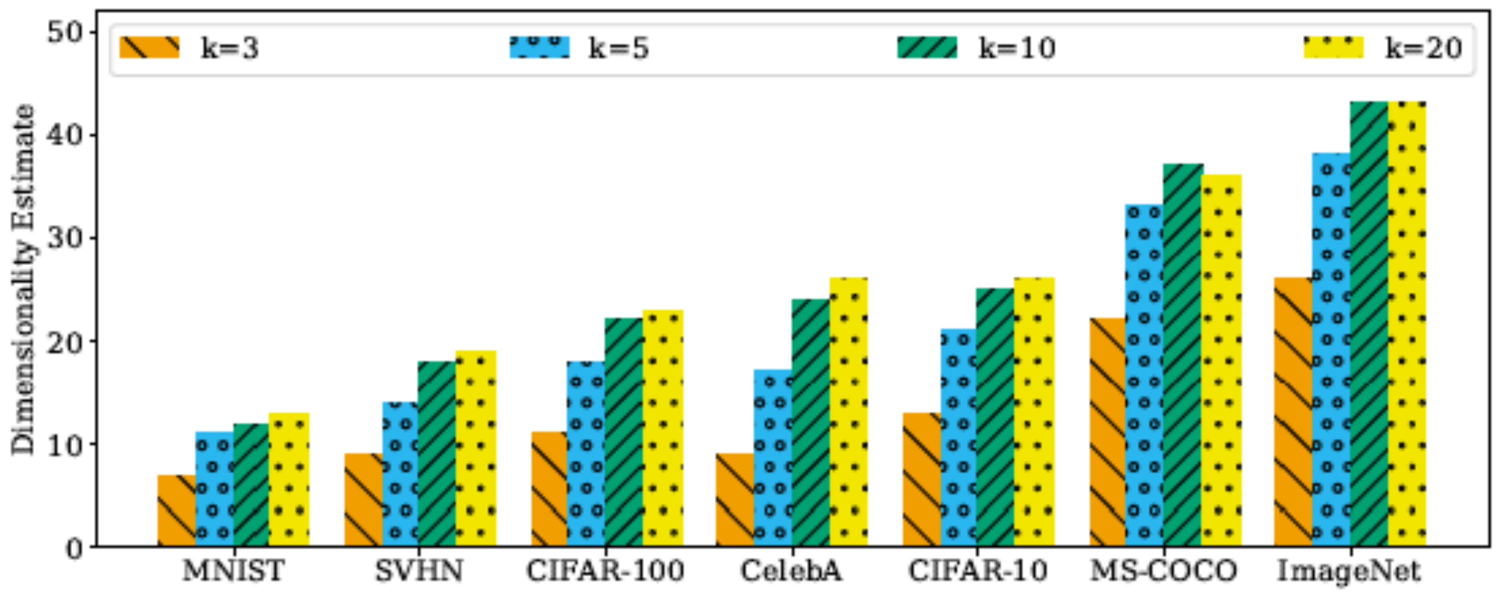


Figure 1: Estimates of the intrinsic dimension of commonly used datasets obtained using the MLE method with $k = 3, 5, 10, 20$ nearest neighbors (left to right). The trends are consistent using different k's.

ImageNet contains 224 x 224 x 3 = 150528 pixels per image,
d between 26 and 43