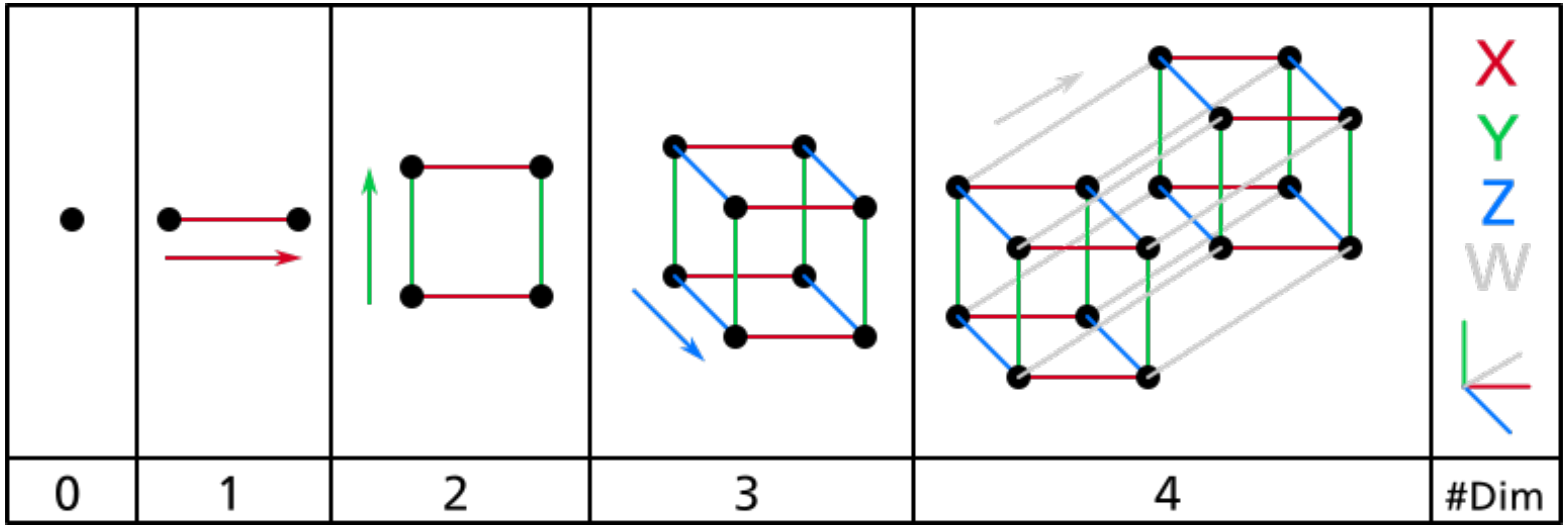
Advanced Python for Neuroscientists

Lecture 2: Dimensionality Reduction

2022/06/30

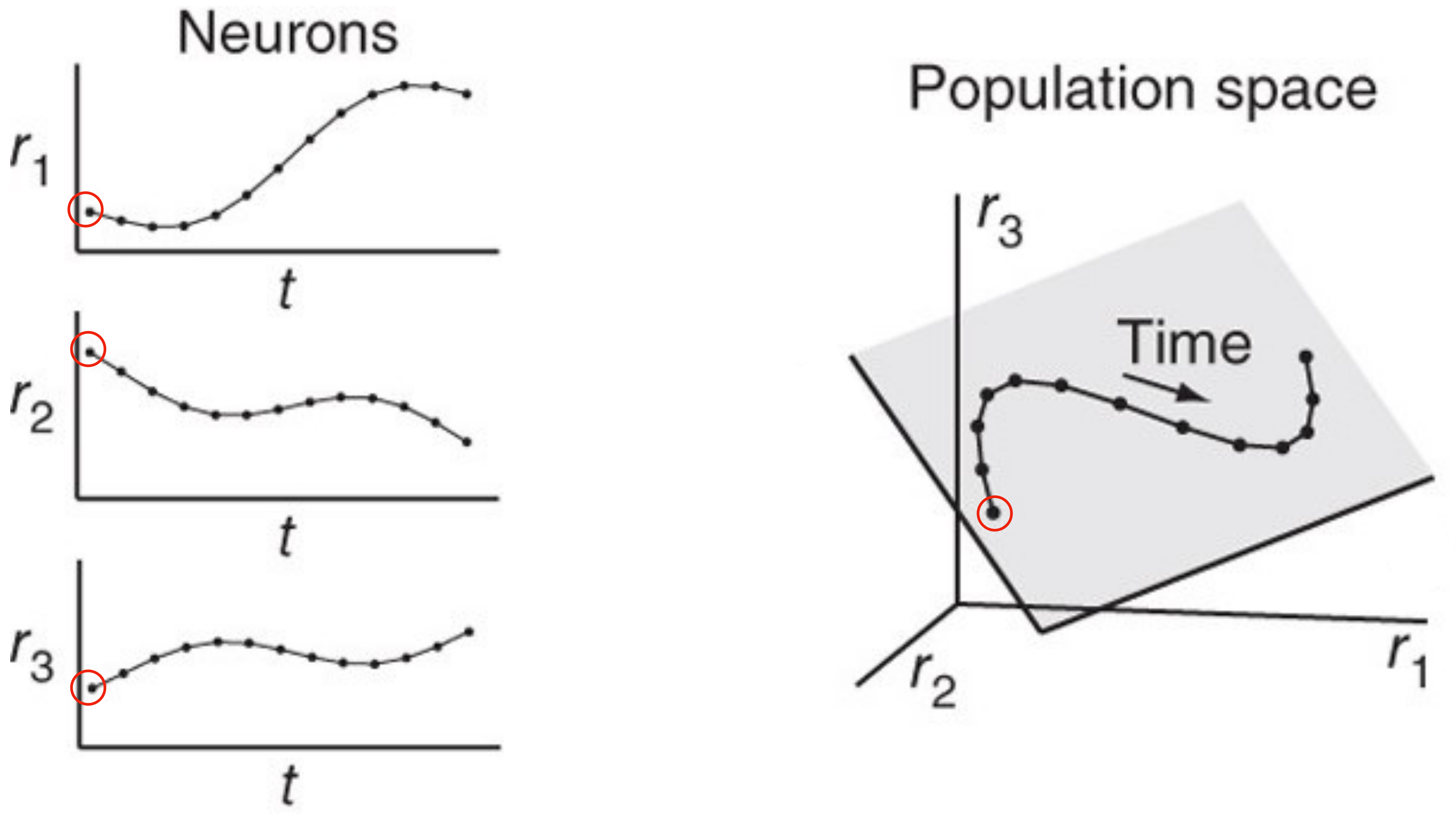
Dimensions in neuroscience

- Dimension in physical space - position coordinate(s)



Dimensions in neuroscience

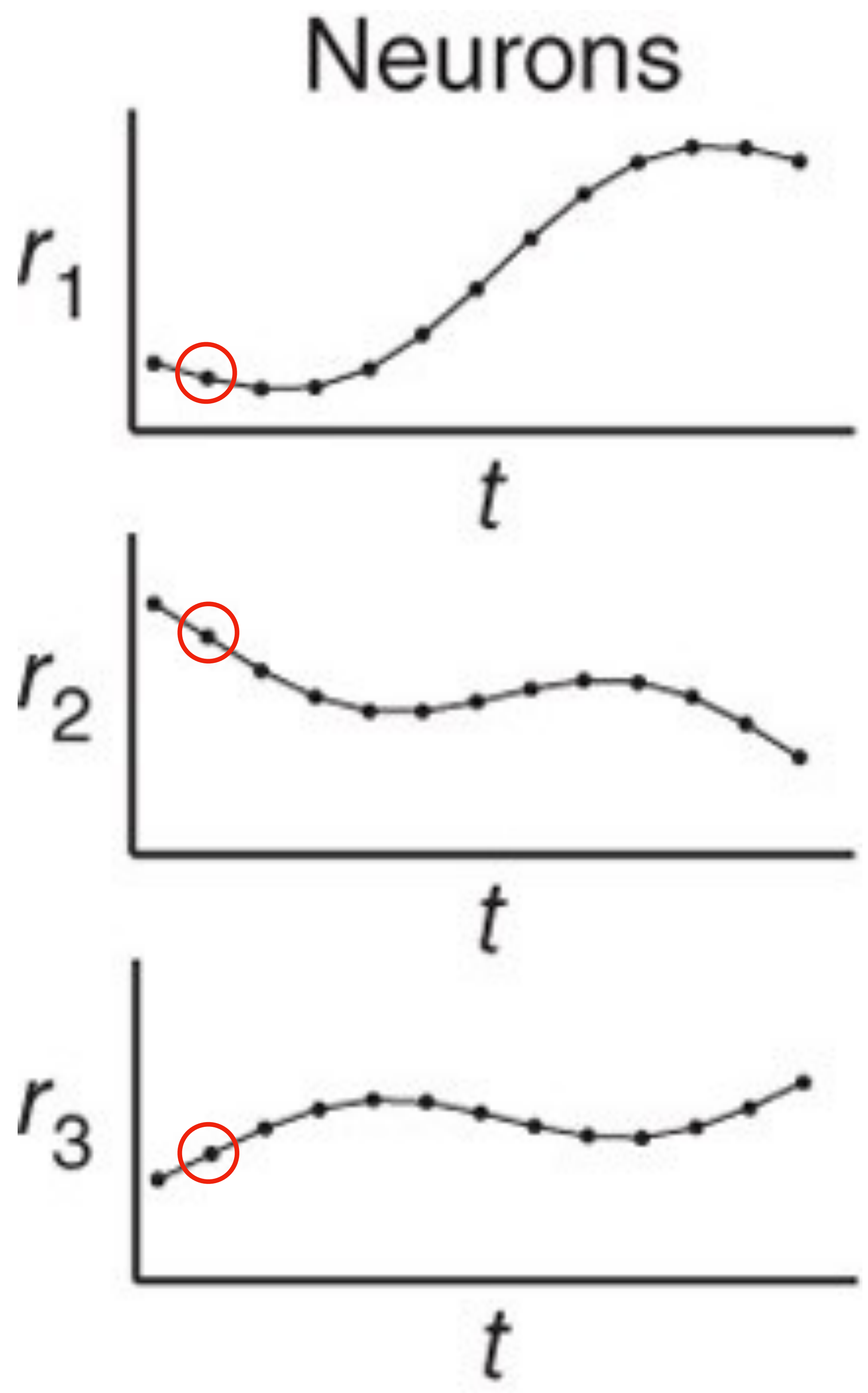
- Dimension in physical space - position coordinate(s)
- Dimension in neuroscience - neural activities

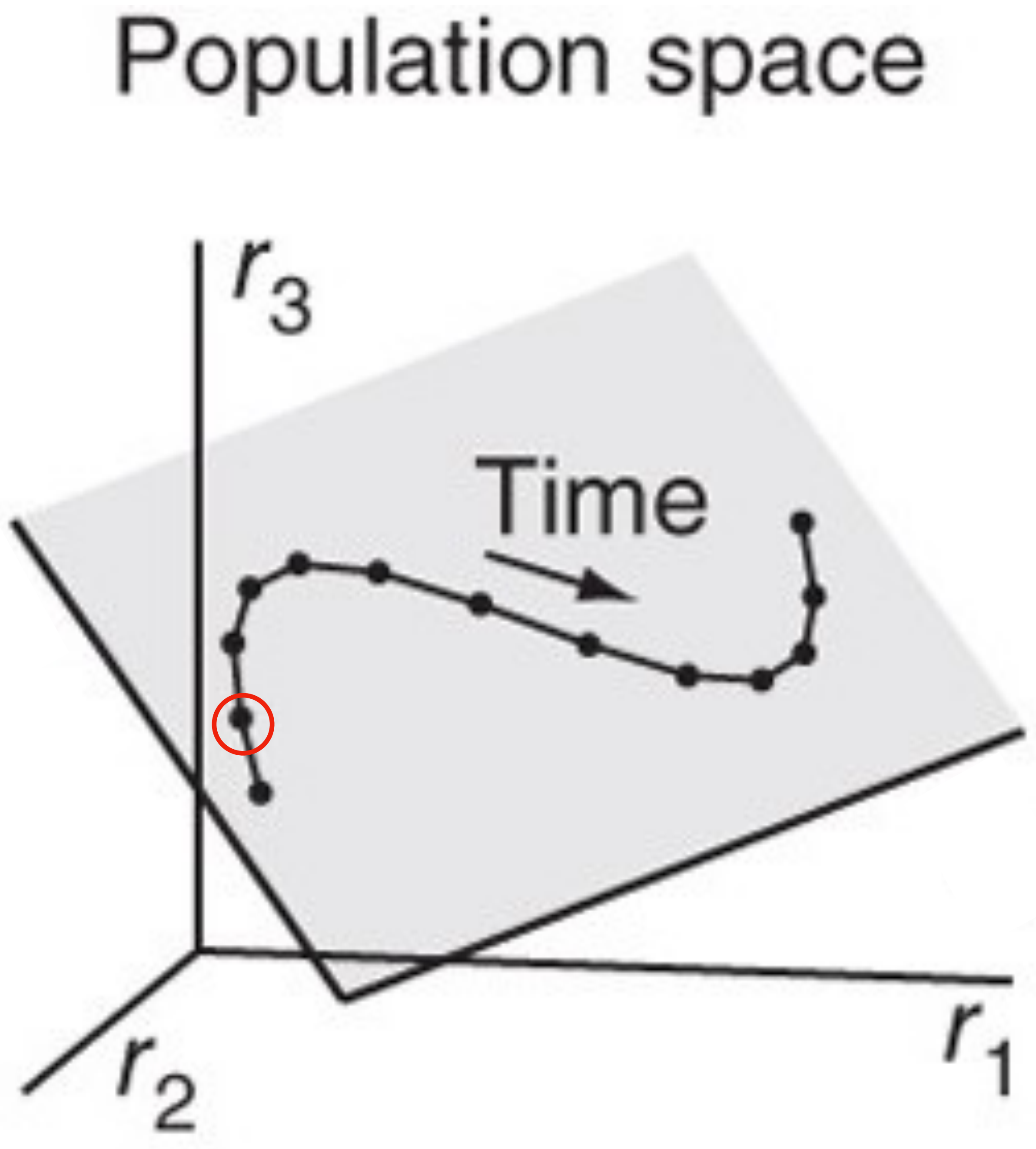


(Cunningham & Yu, 2014)

Dimensions in neuroscience

- Dimension in physical space - position coordinate(s)
- Dimension in neuroscience - neural activities

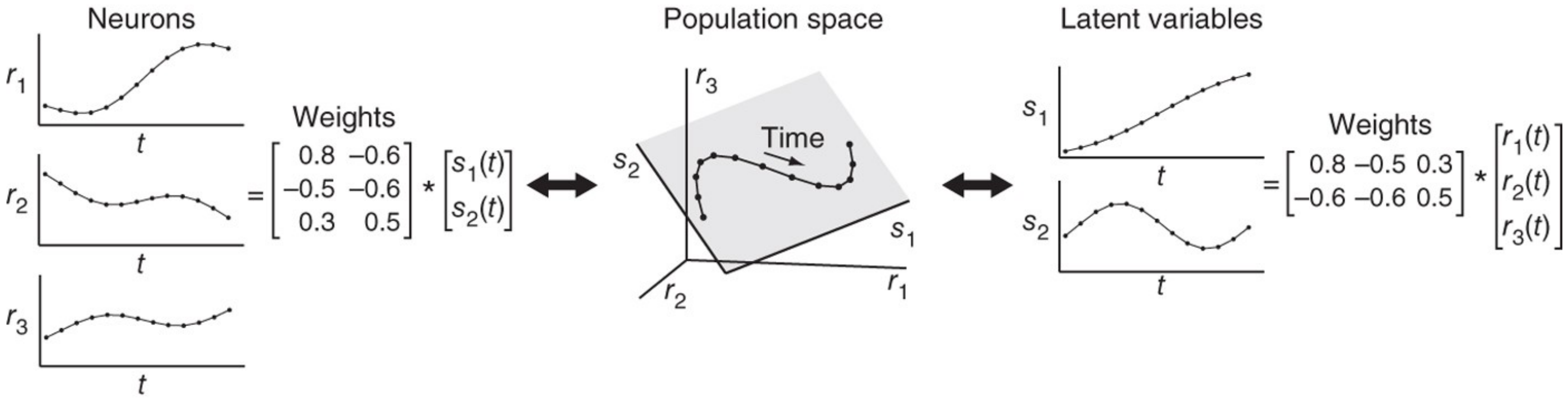




(Cunningham & Yu, 2014)

Dimensionality Reduction

- Dimension of the population space - # neurons / voxels / channels

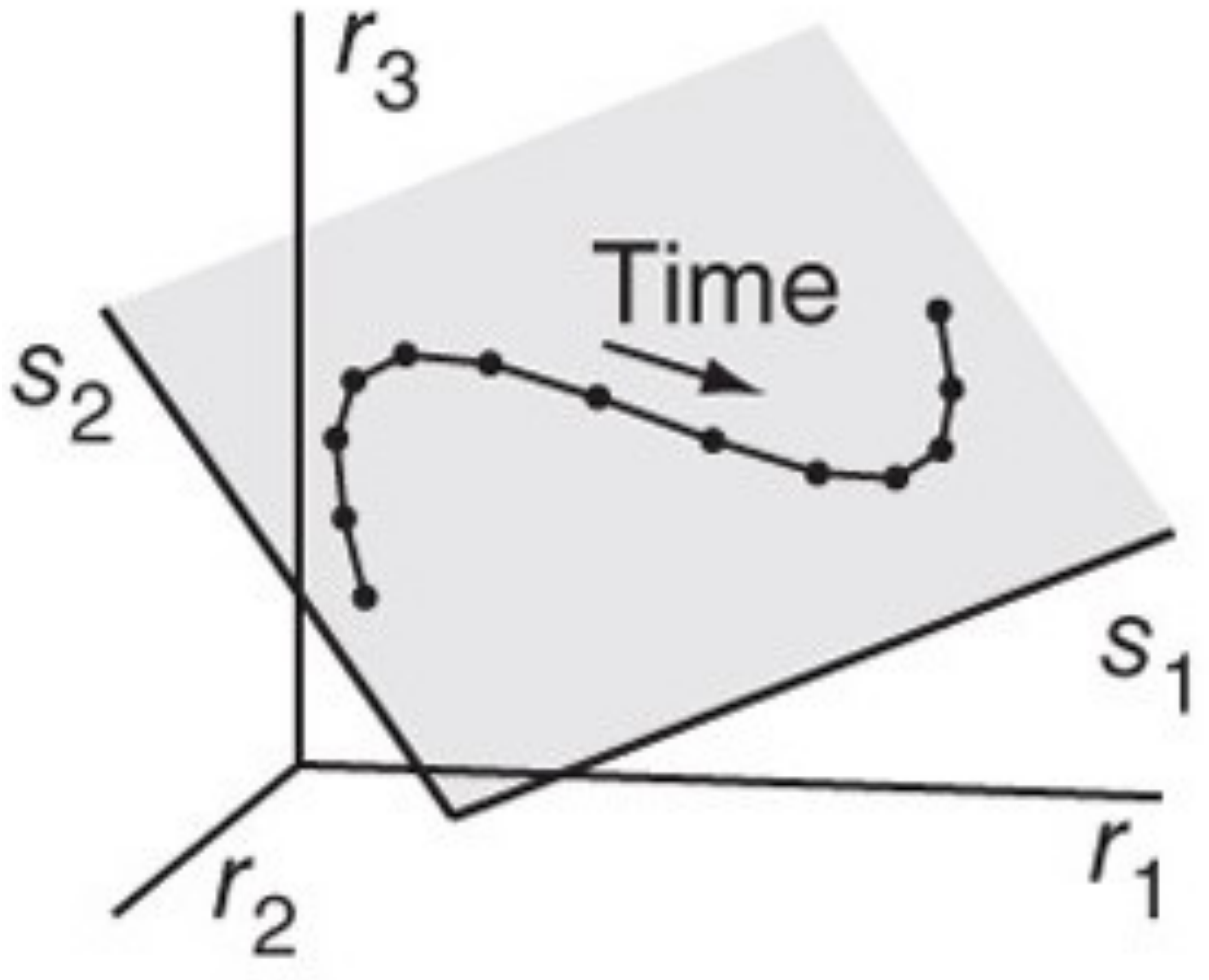


(Cunningham & Yu, 2014)

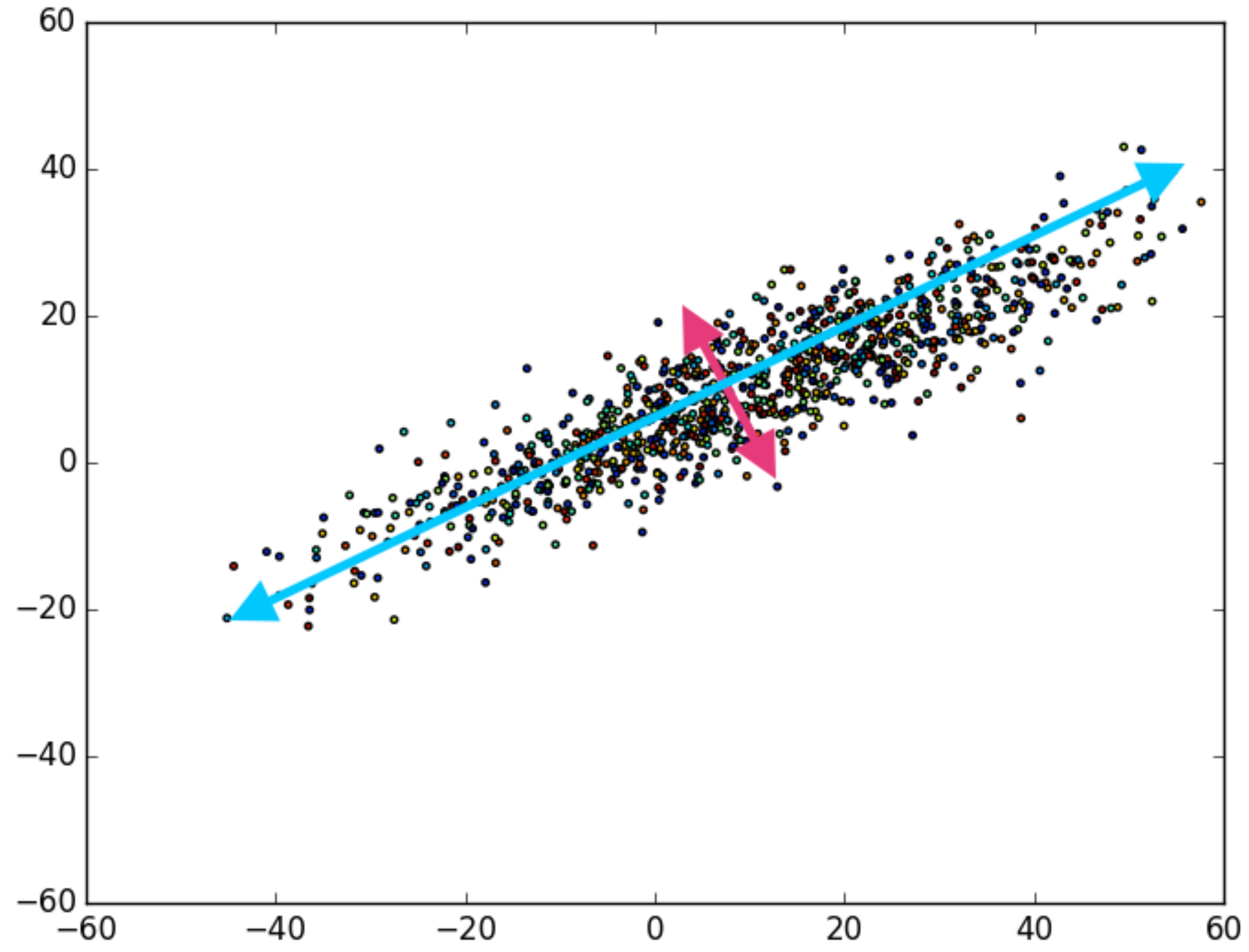
Dimensionality Reduction

- Dimension of the population space - # neurons / voxels / channels

3D to 2D



2D to 1D



(Cunningham & Yu, 2014)

Dimensions in neuroscience

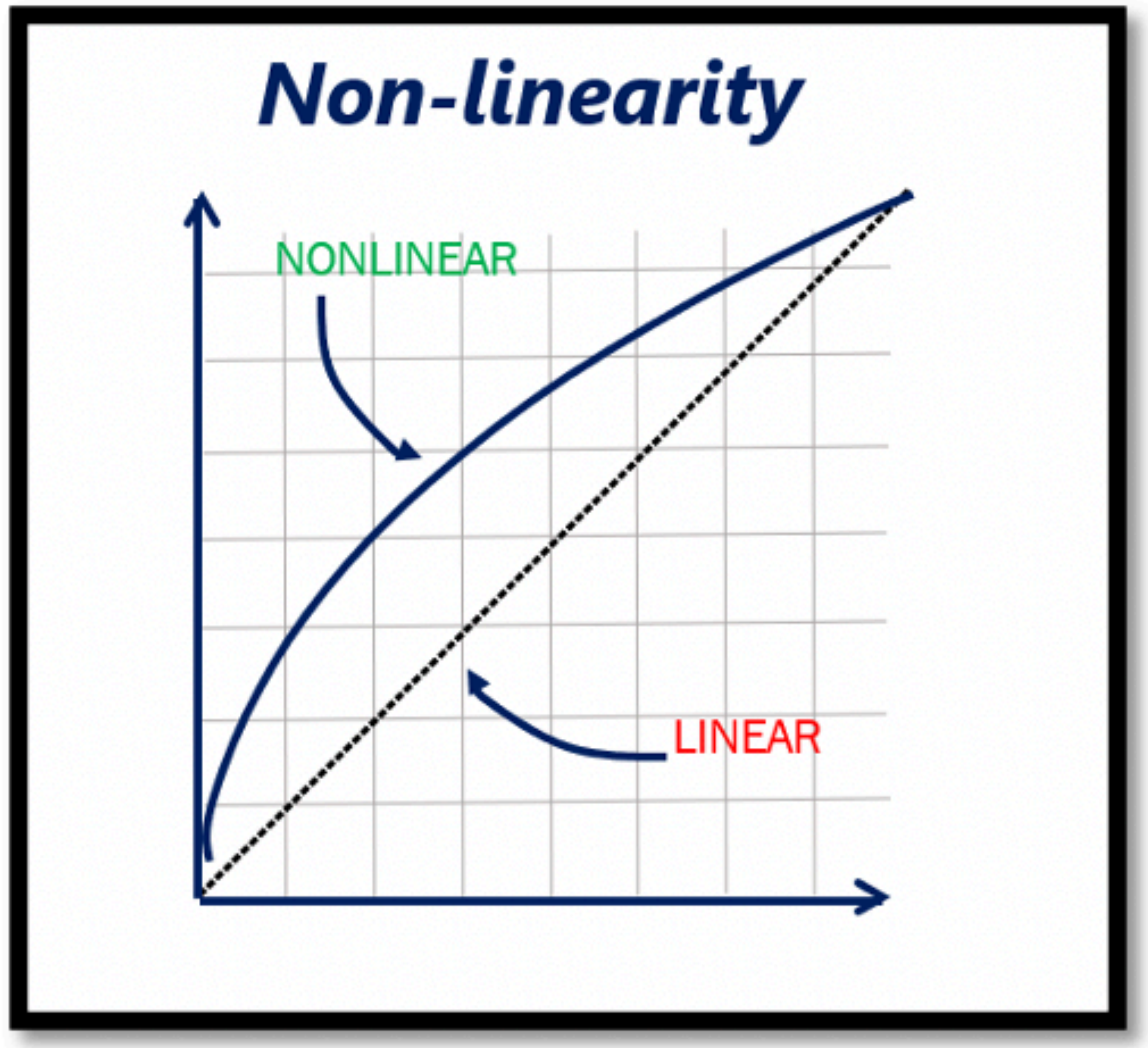
- Dimension of the population space - # neurons / voxels / channels
- Alternative names for “dimensions” - features / predictors / variables
- Reduce the number of variables to look at

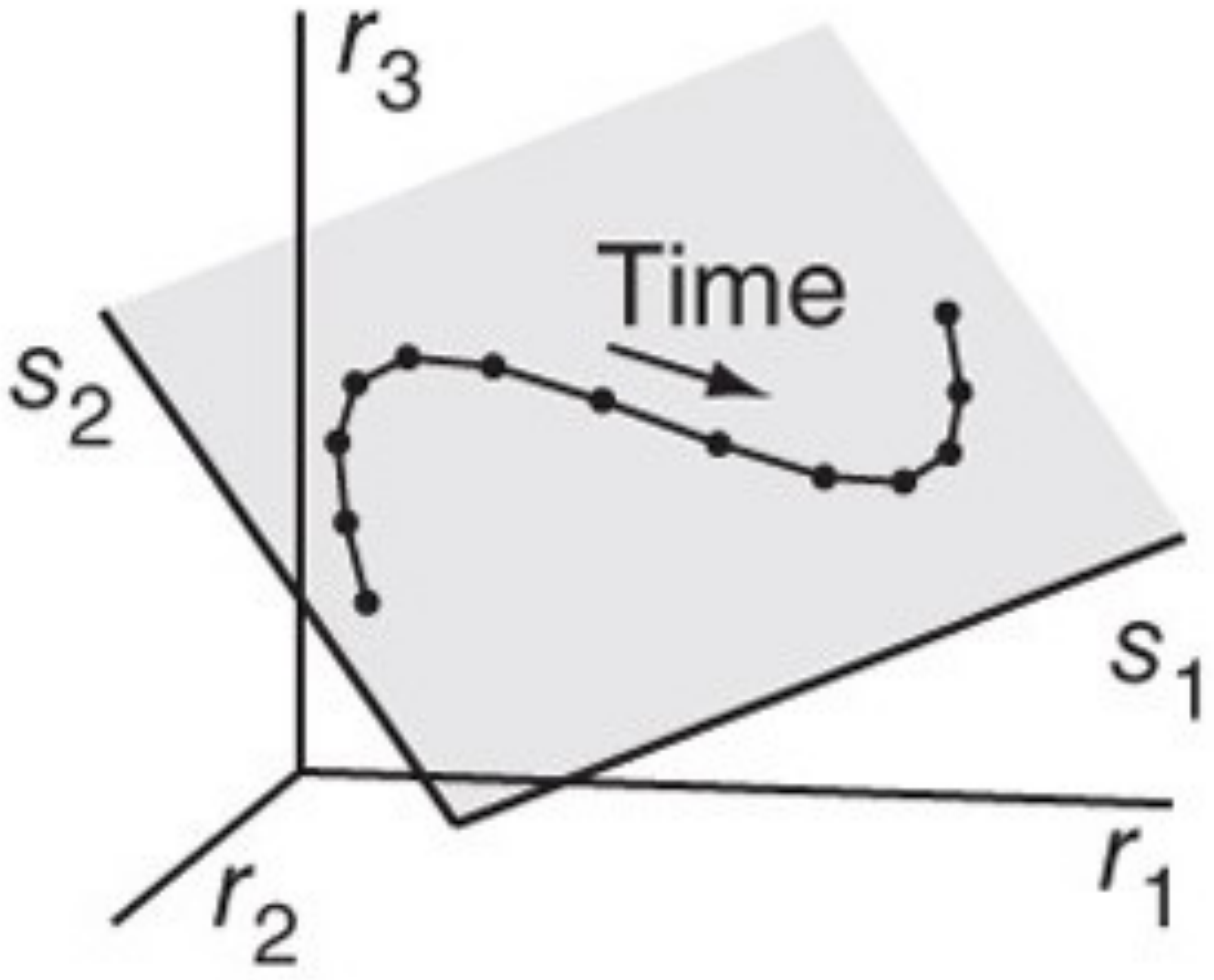
Dimensionality Reduction

- PCA
- Factor analysis
- Manifold learning
  - t-SNE
- Autoencoders

Linearity

- Definition: $f(x+y) = f(x) + f(y), f(\alpha x) = \alpha f(x)$
- Example: $3r_1 + 2.5r_2 + \ldots 0r_n + 100$

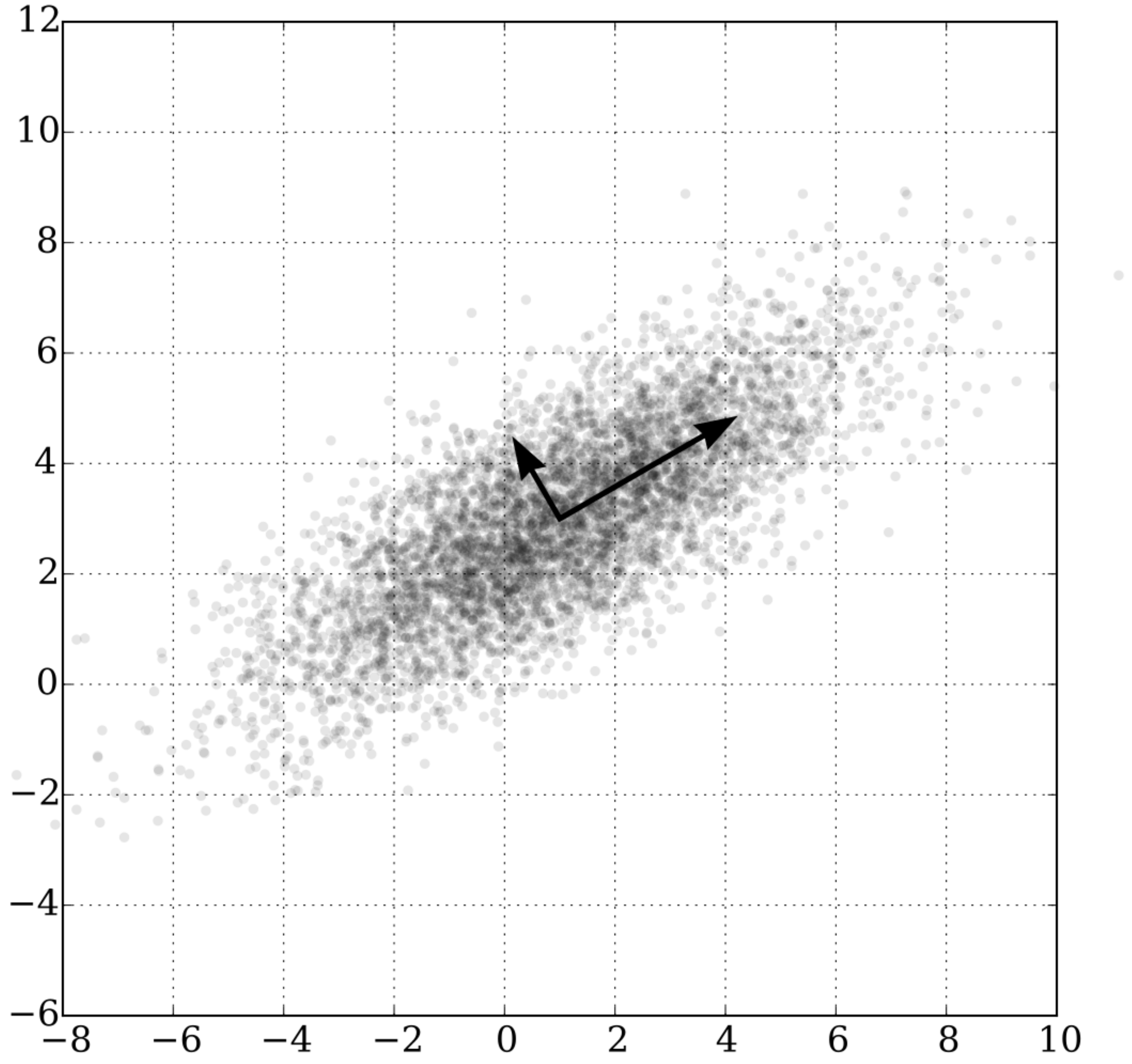




PCA - if a linear model works, go with it first

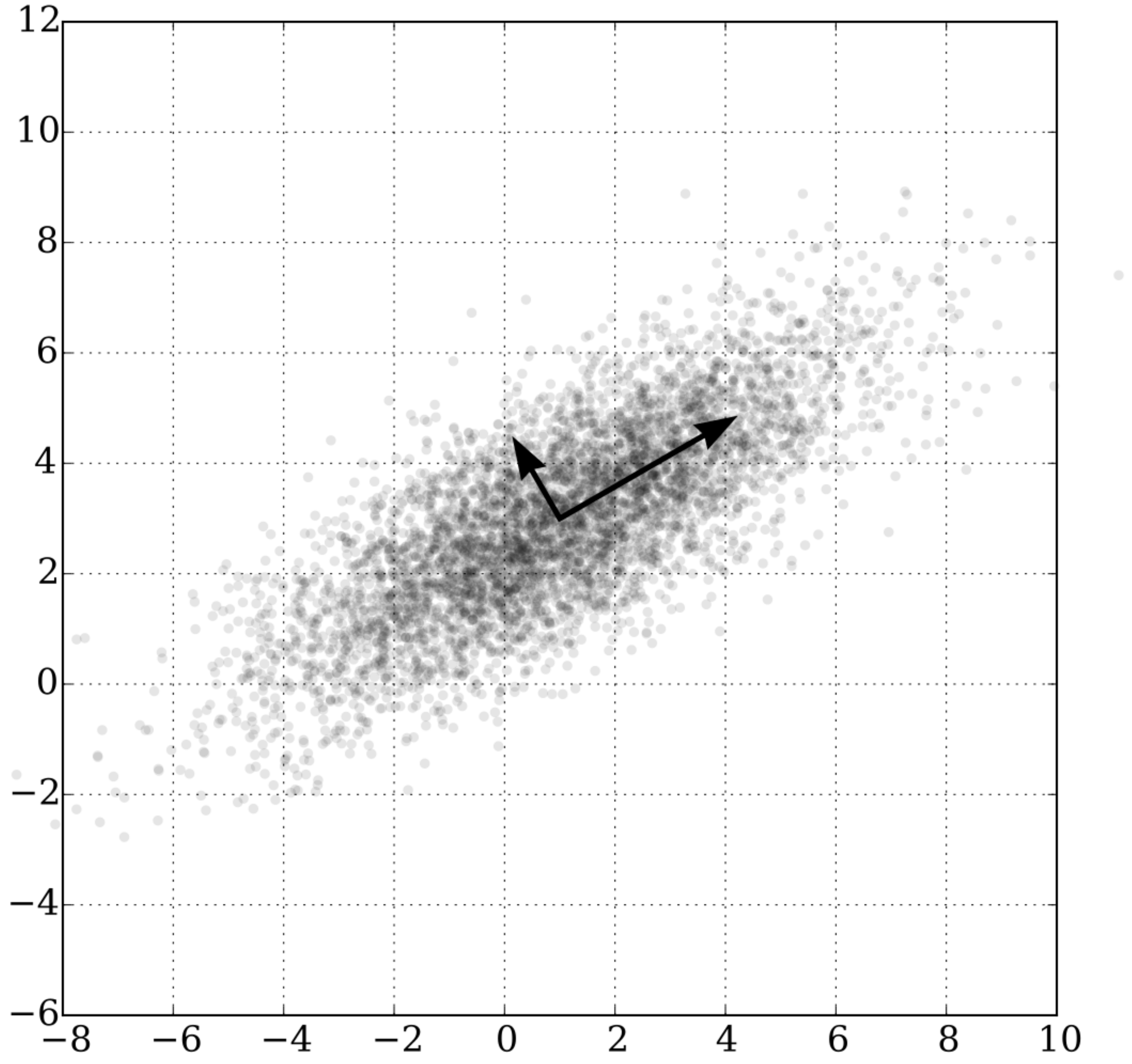
PCA

- **Principal** component analysis
- Linear projection: $w_1 r_1 + w_2 r_2 + \ldots$
- $\arg\max_{w_1, w_2} V(w_1 r_1 + w_2 r_2 + \ldots)$
- Continue to do step 3 after "removing" previous components by $\mathbf{X} - \sum_{s=1}^{k-1} \mathbf{X}\mathbf{w}_{(s)}\mathbf{w}_{(s)}^{\top}$
- Components are orthogonal

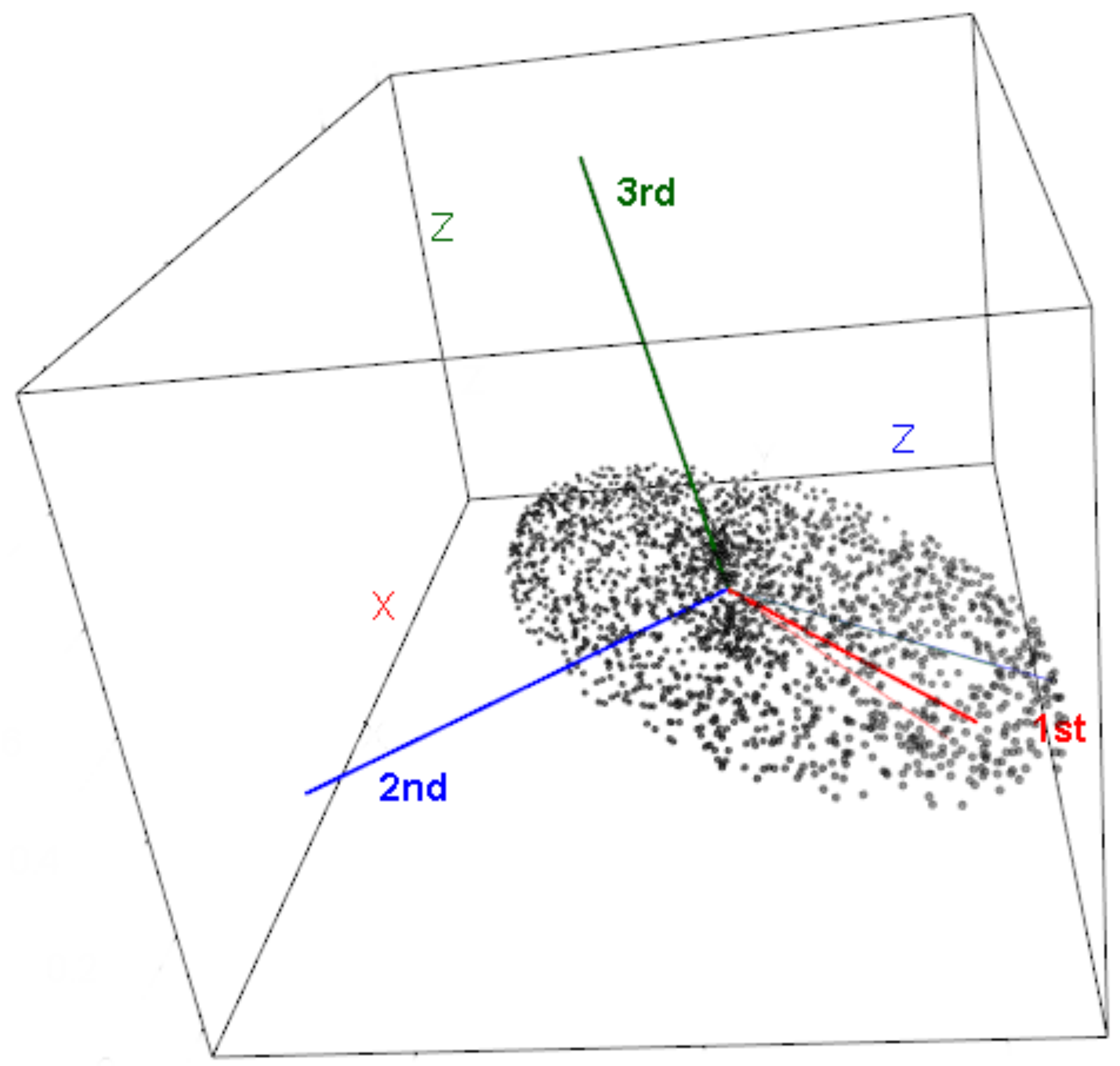


PCA - parameters / nomenclatures

- n_components: number of directions to project onto



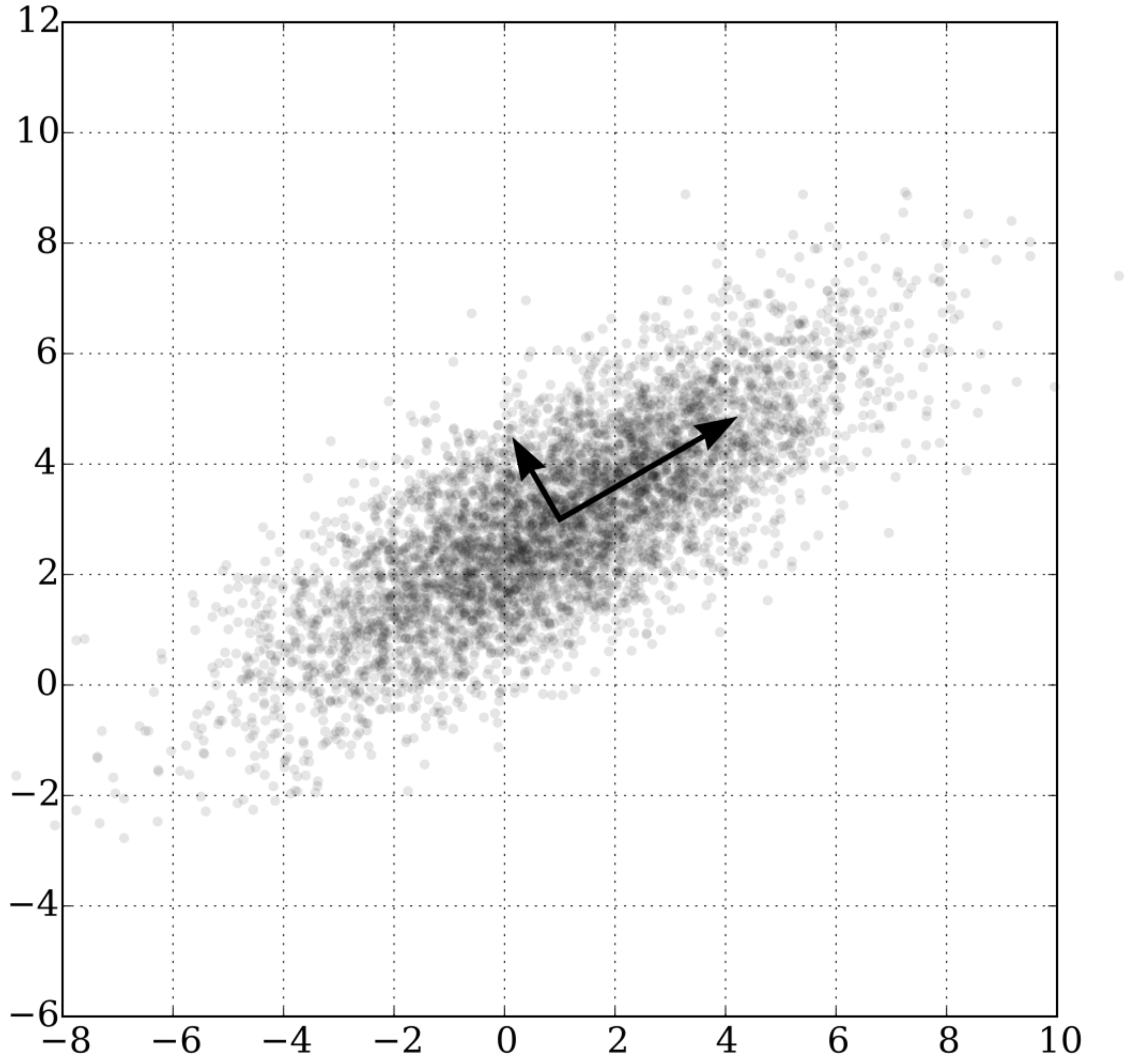
PCA applied to an ellipsoidically shaped point cloud



more information: www.joyofdata.de/blog/illustration-of-principal-component-analysis-pca

PCA - parameters / nomenclatures

- n_components: number of directions to project onto
- explained_variance: the amount of variance retained after projecting onto the PC



PCA - parameters / nomenclatures

- n_components: number of directions to project onto
- explained_variance: the amount of variance retained after projecting onto the PC
- loadings: principal component direction $w_1, w_2, ...$

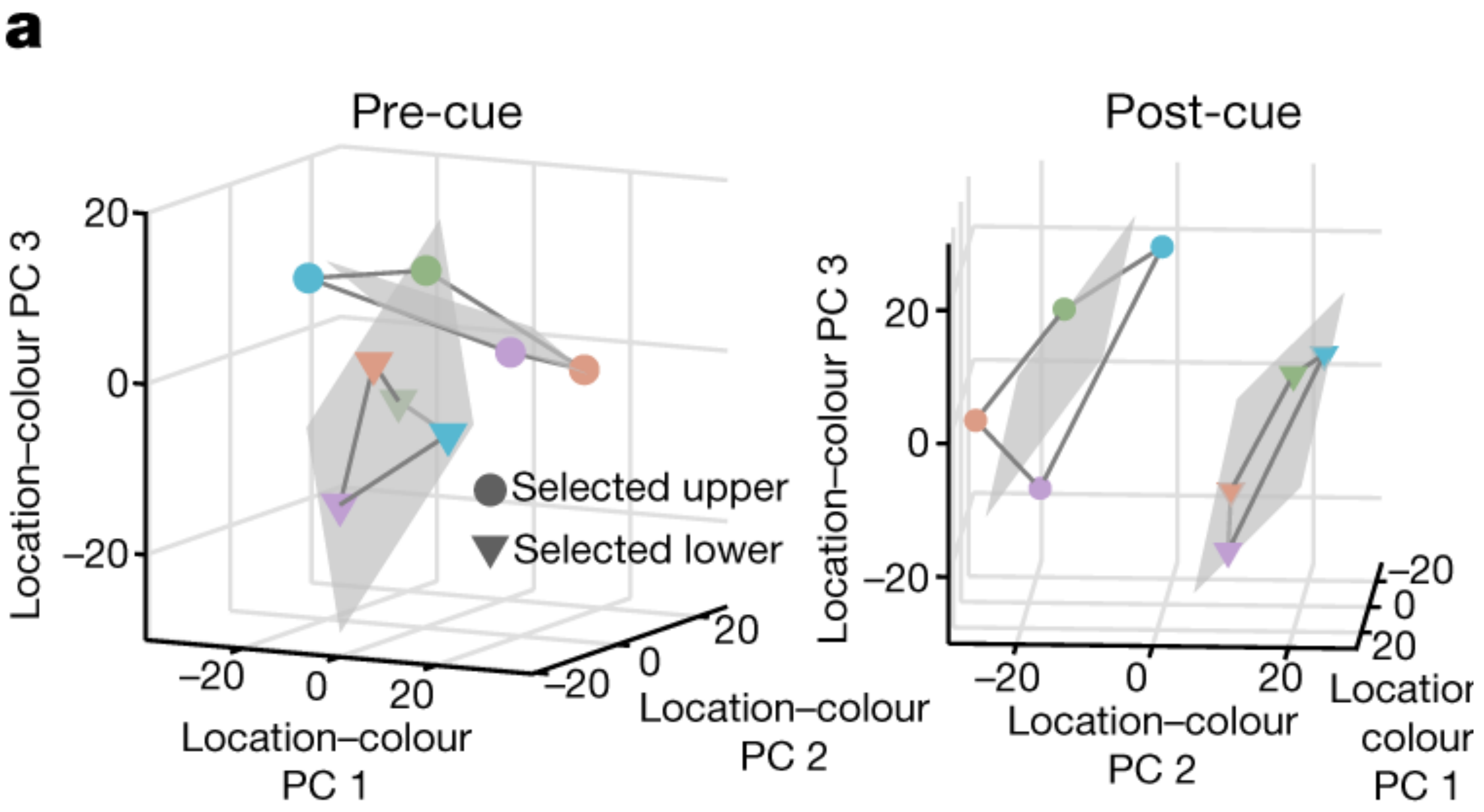
PCA - parameters / nomenclatures

- n_components: number of directions to project onto
- explained_variance: the amount of variance retained after projecting onto the PC
- loadings: principal component direction $w_1, w_2, \ldots$
- components: latent factors, latent variables

PCA

Principal Component Analysis

- Widely used

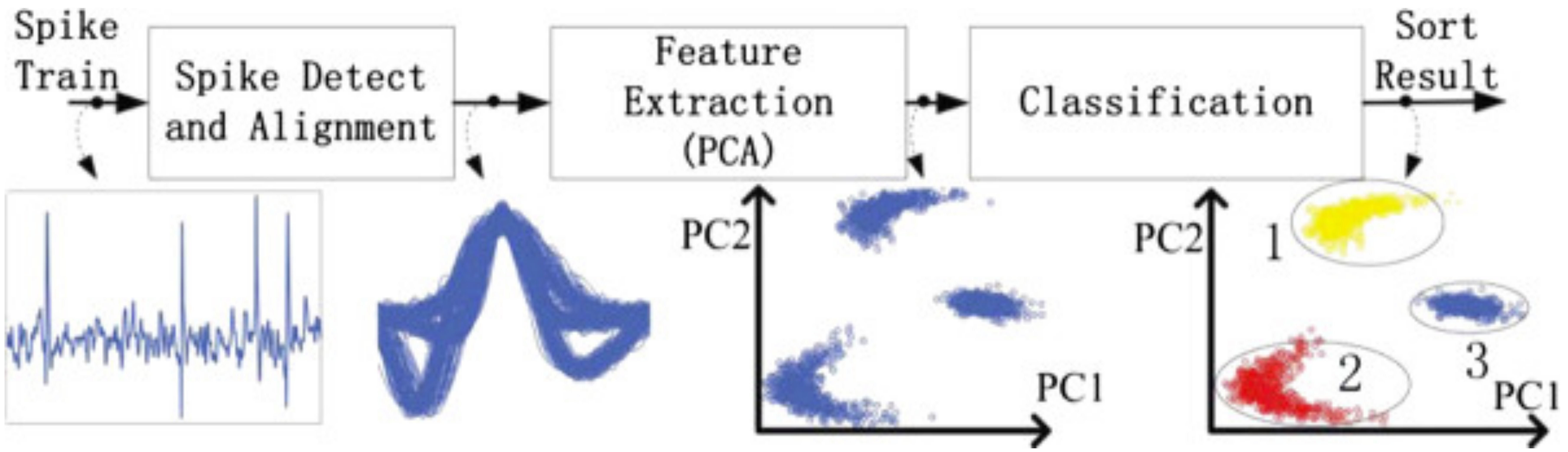


(Buschman group)

PCA

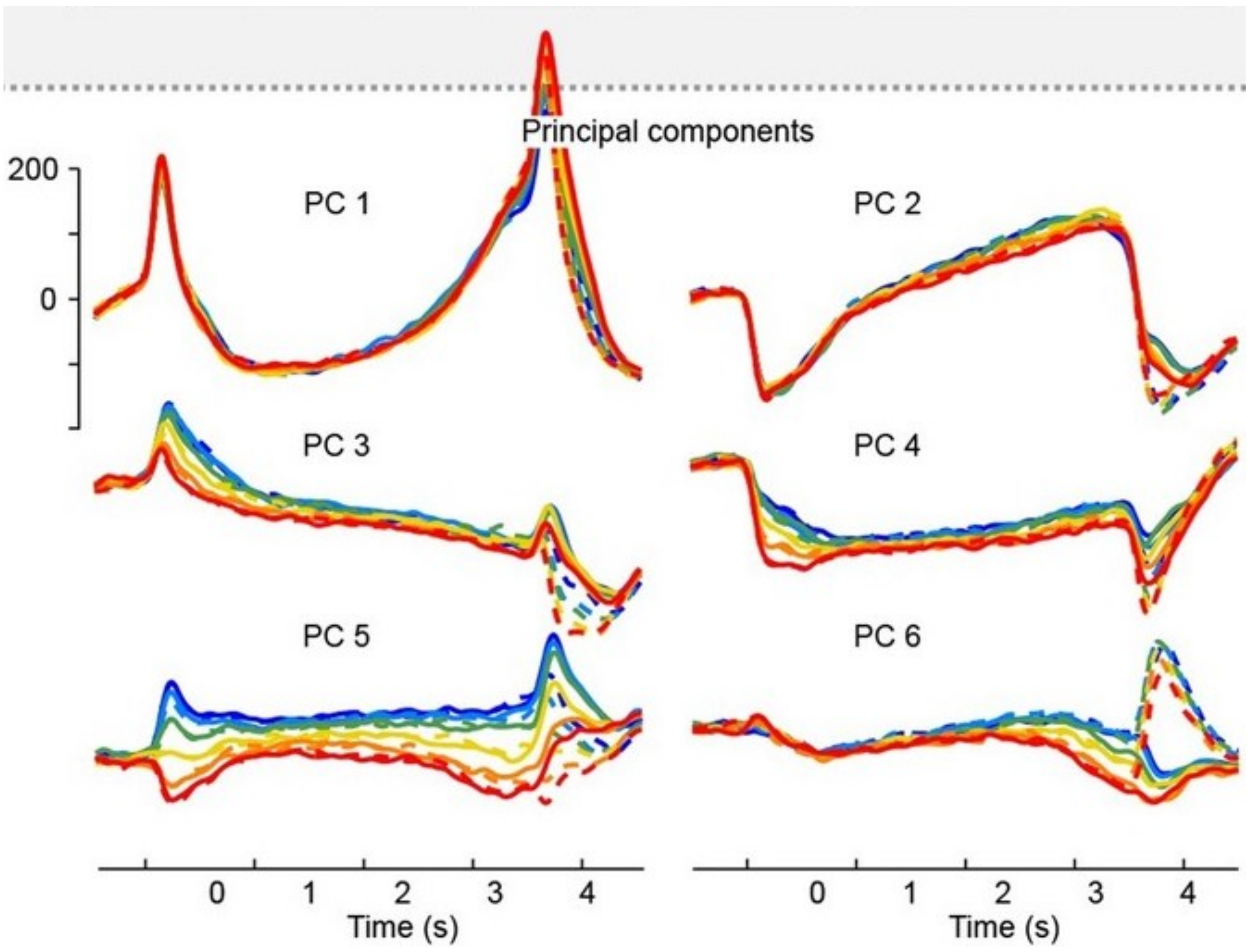
Principal Component Analysis

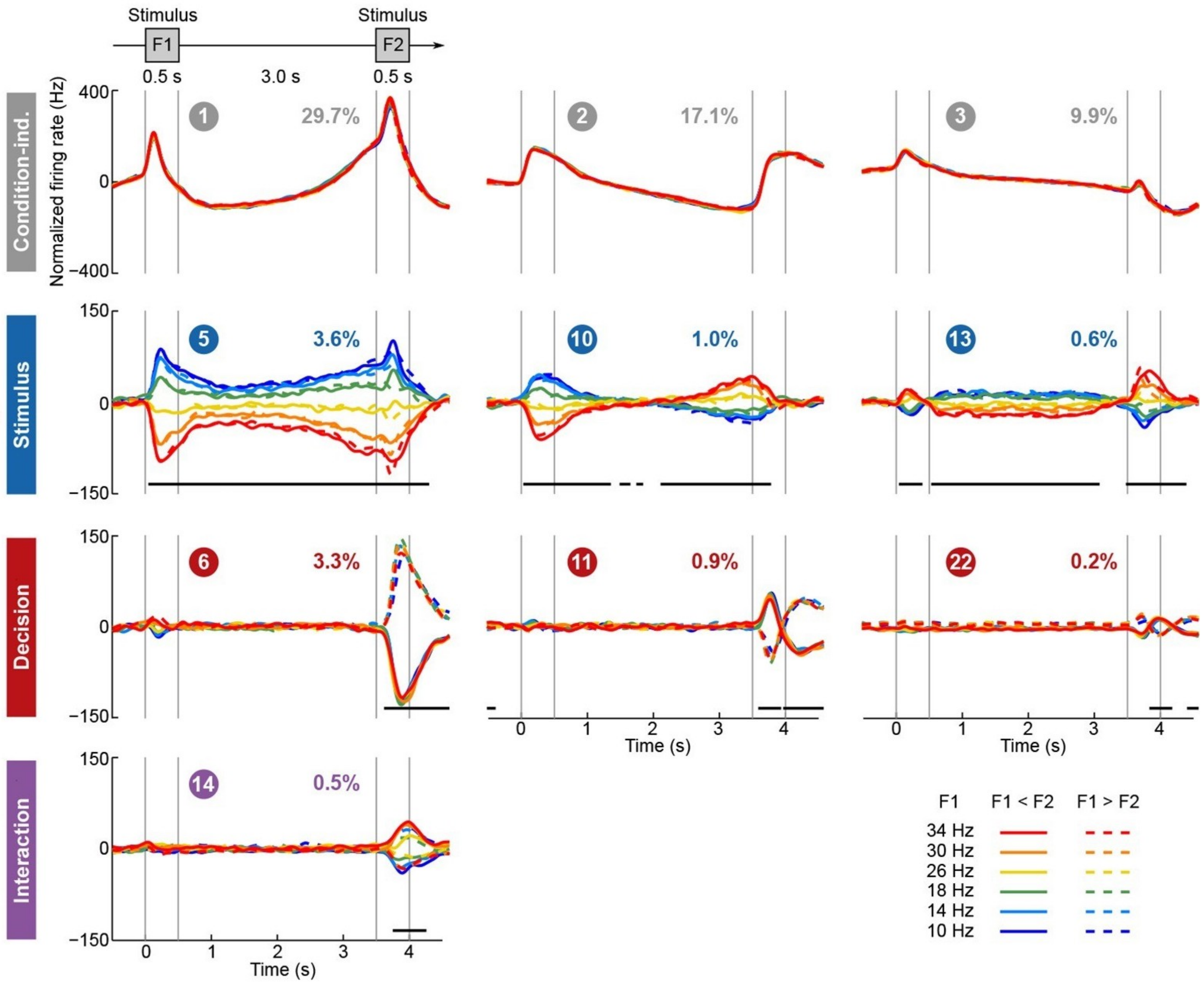
- Widely used



PCA family

- Demix-PCA
  - Do PCA on data averaged by condition of interest





(Kobak et al., 2016)

PCA family

- Demix-PCA
- ICA - independent component analysis
    - For each projected $s_i = w_{i1}r_1 + \ldots$, maximum some measure of independent of $F(s_1, s_2, \ldots)$

PCA family

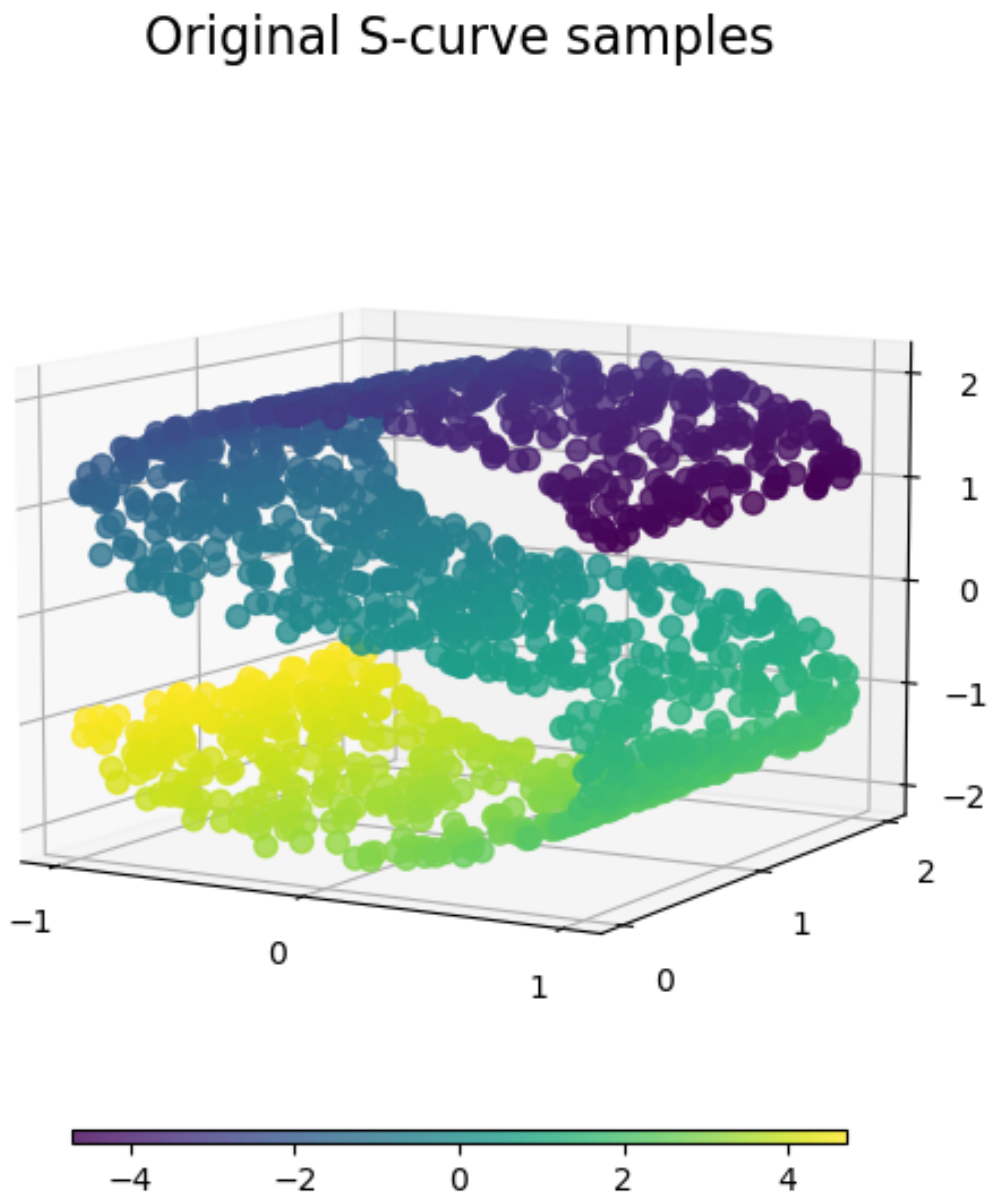
- Demix-PCA
- ICA - independent component analysis
  - For each projected $s_i = w_{i1}r_1 + \ldots$, maximum some measure of independent of $F(s_1, s_2, \ldots)$
  - Face
    - PCA: brightness, average face
    - ICA: nose, eyes,…

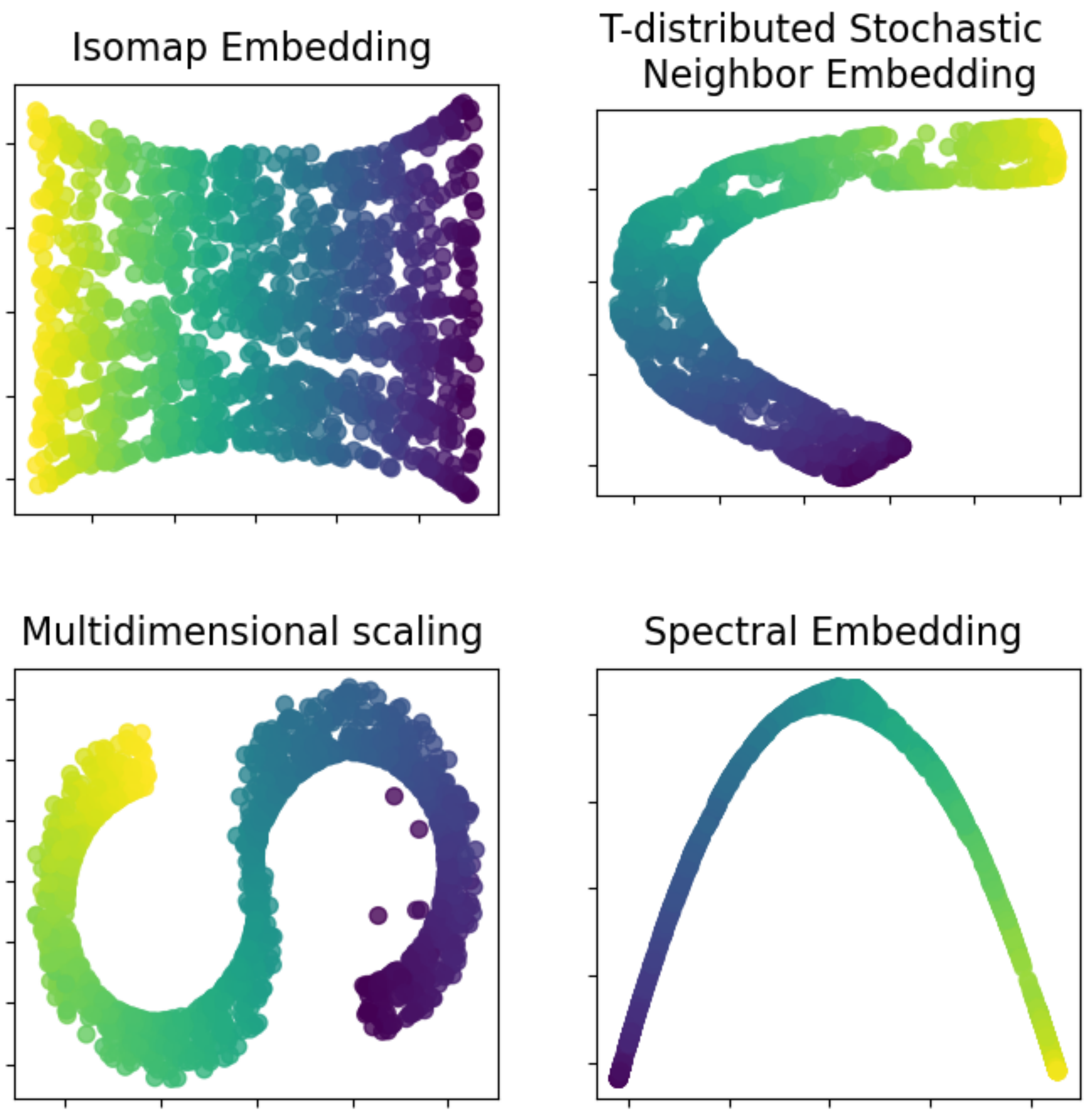
PCA family

- Demix-PCA
- ICA - independent component analysis
- FA - factor analysis
  - r_i of each neuron is made up of series factors $f_1, f_2, ..$
  - Probability based fitting
  - When f_i are orthonormal, probabilistic PCA

Manifold Learning - nonlinear

Manifold learning





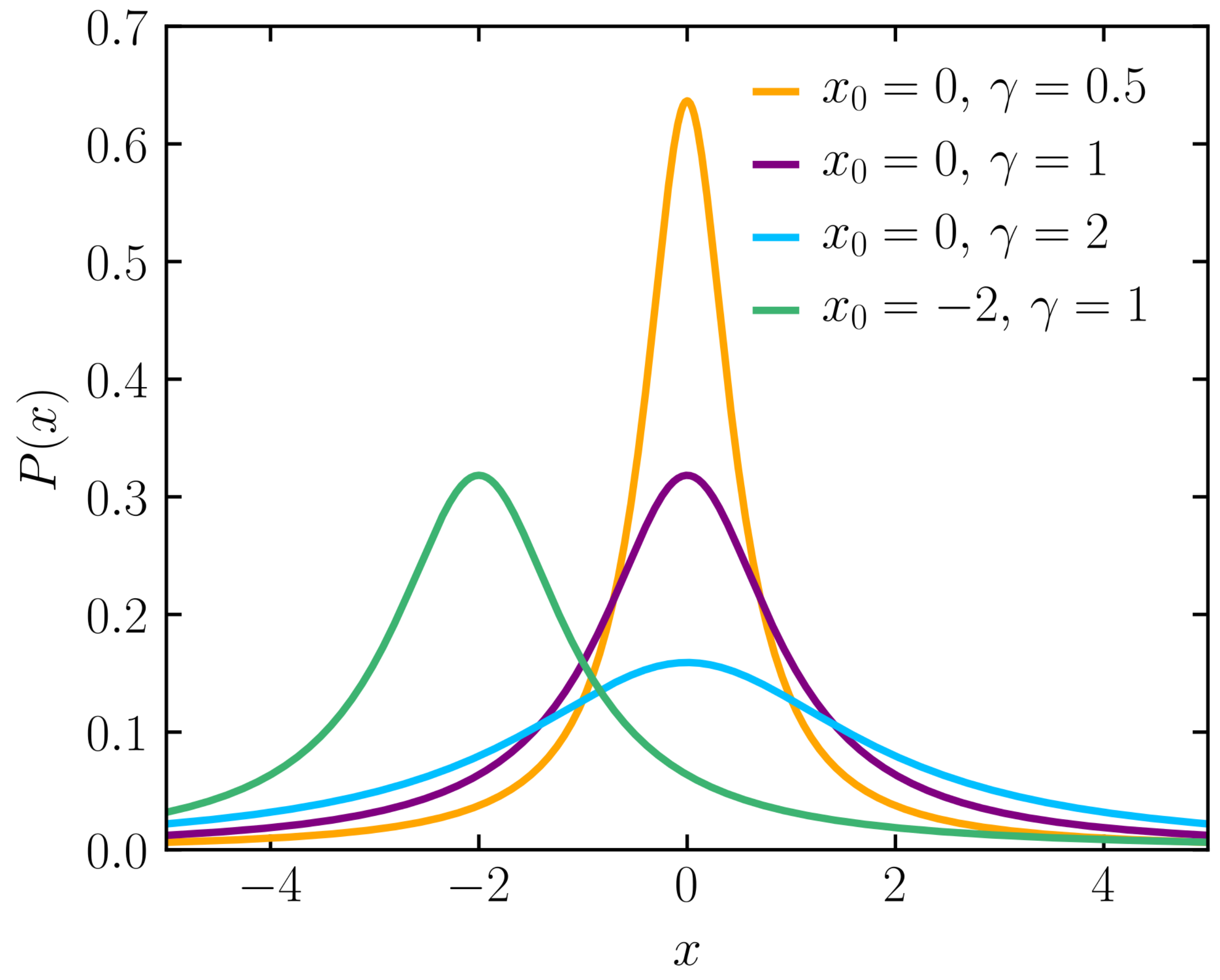
Manifold learning

t-SNE

- SNE - stochastic neighborhood embedding
  - Probability of point i would pick point j as neighbor
  - Make embedded $q_{j|i}$ as similar as possible with $p_{j|i}$

$$p_{j|i} = \frac{\exp\left(-\frac{1}{2\sigma_i^2} \parallel \boldsymbol{x}_i - \boldsymbol{x}_j \parallel^2\right)}{\sum_{k\neq i}\exp\left(-\frac{1}{2\sigma_i^2} \parallel \boldsymbol{x}_i - \boldsymbol{x}_k \parallel^2\right)}$$

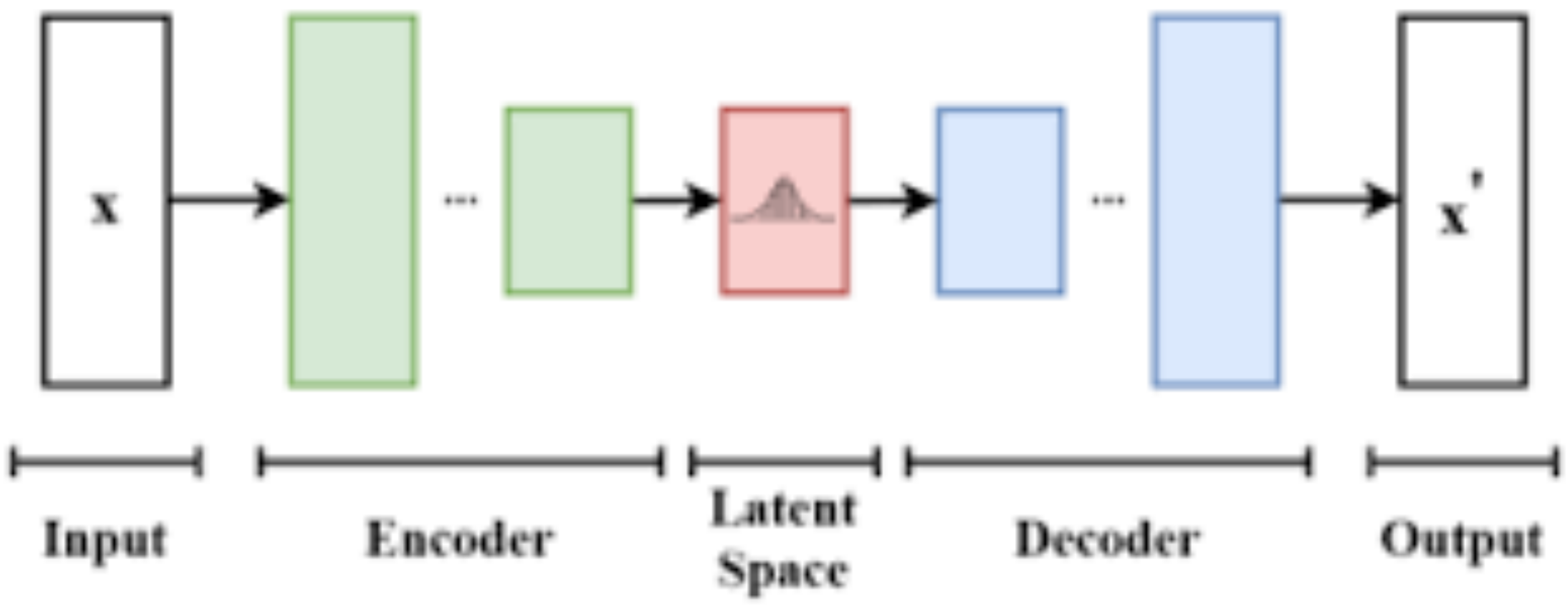
- t - Student's t-distribution
  - Use it to measure distance (heavier tails)



Auto-encoders - NN based

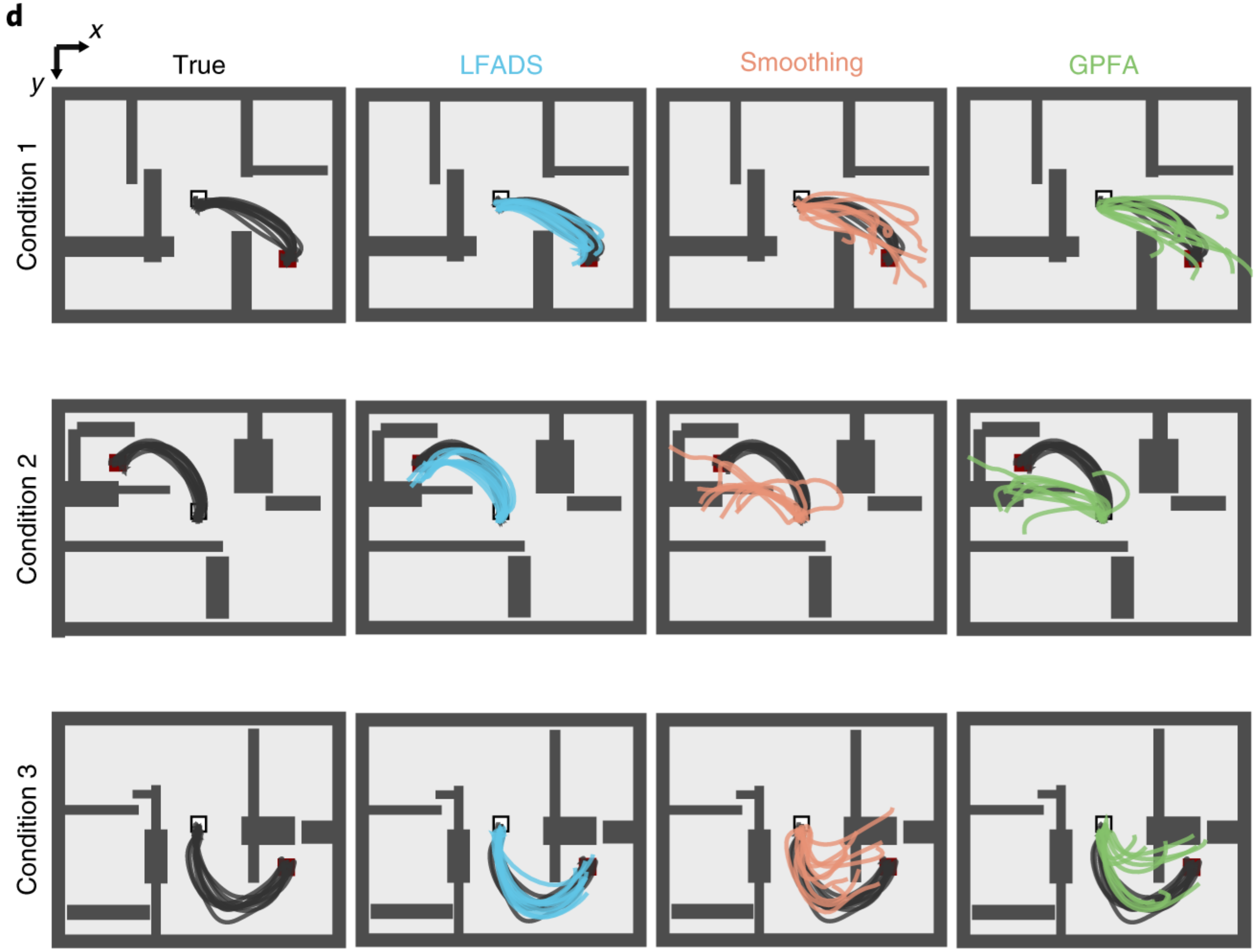
Auto-encoders

- NN to infer latent variable



Auto-encoders

- NN to infer latent variable



(Pandarinath et al., 2018)