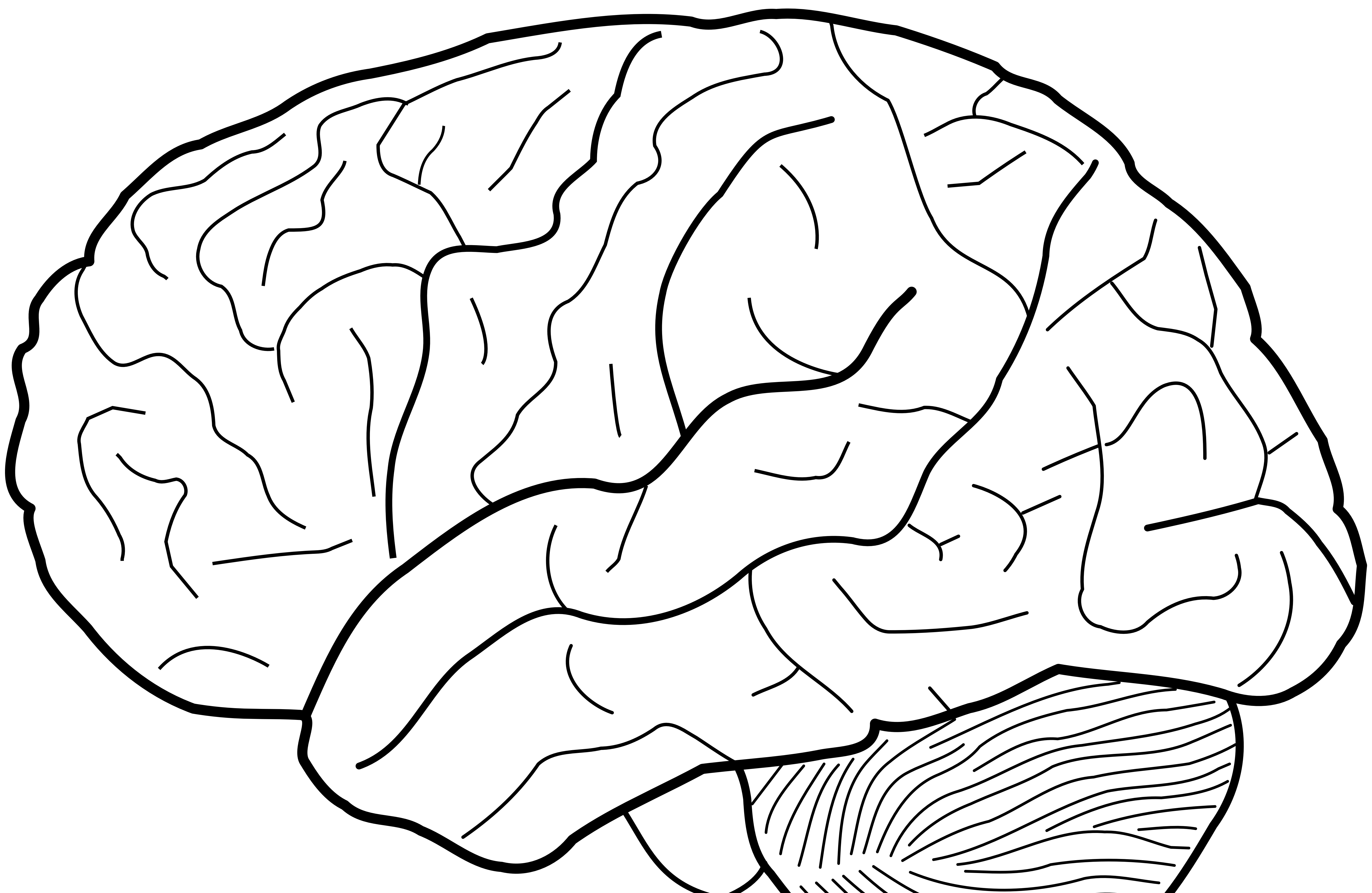
Neuroscience Introduction

hhmi | janelia Research Campus

databricks™

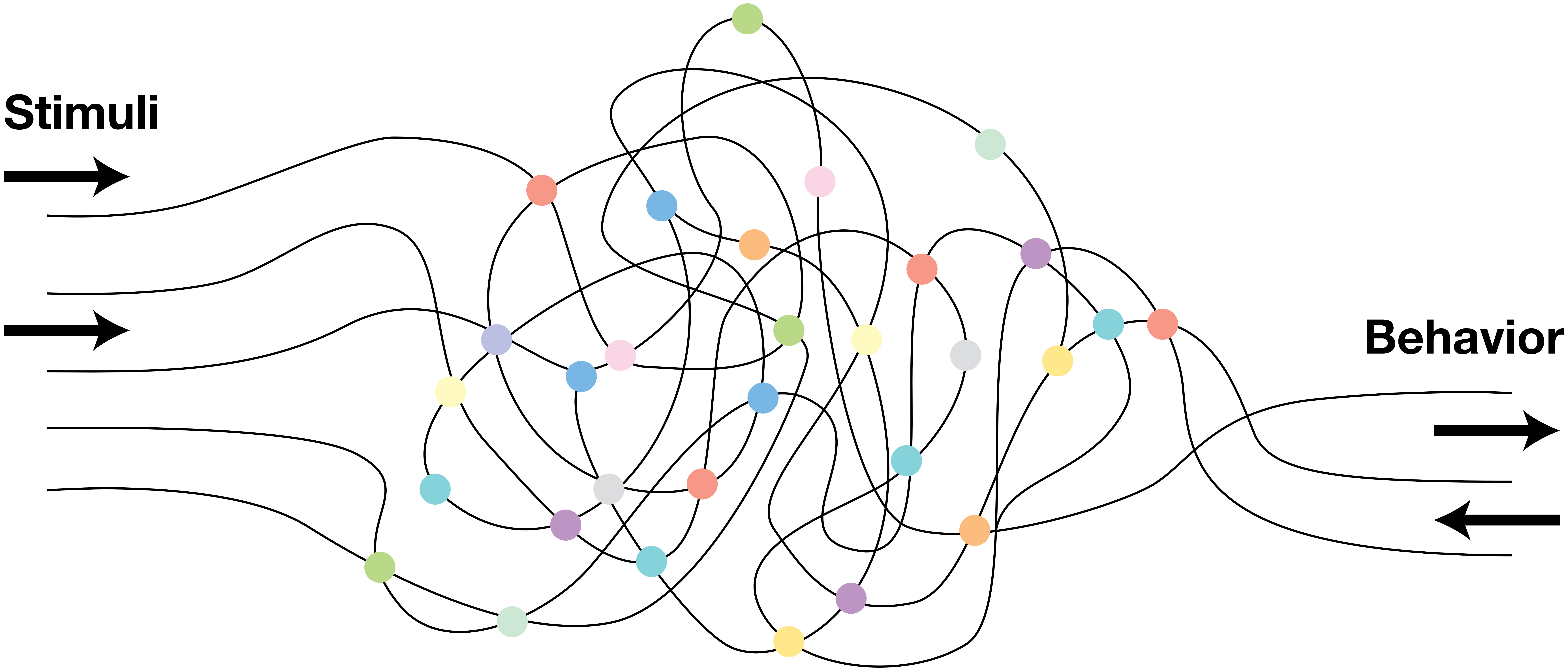
The brain



“As humans, we can identify galaxies light years away, we can study particles smaller than an atom. **But we still haven’t unlocked the mystery of the three pounds of matter that sits between our ears**.”

President Obama

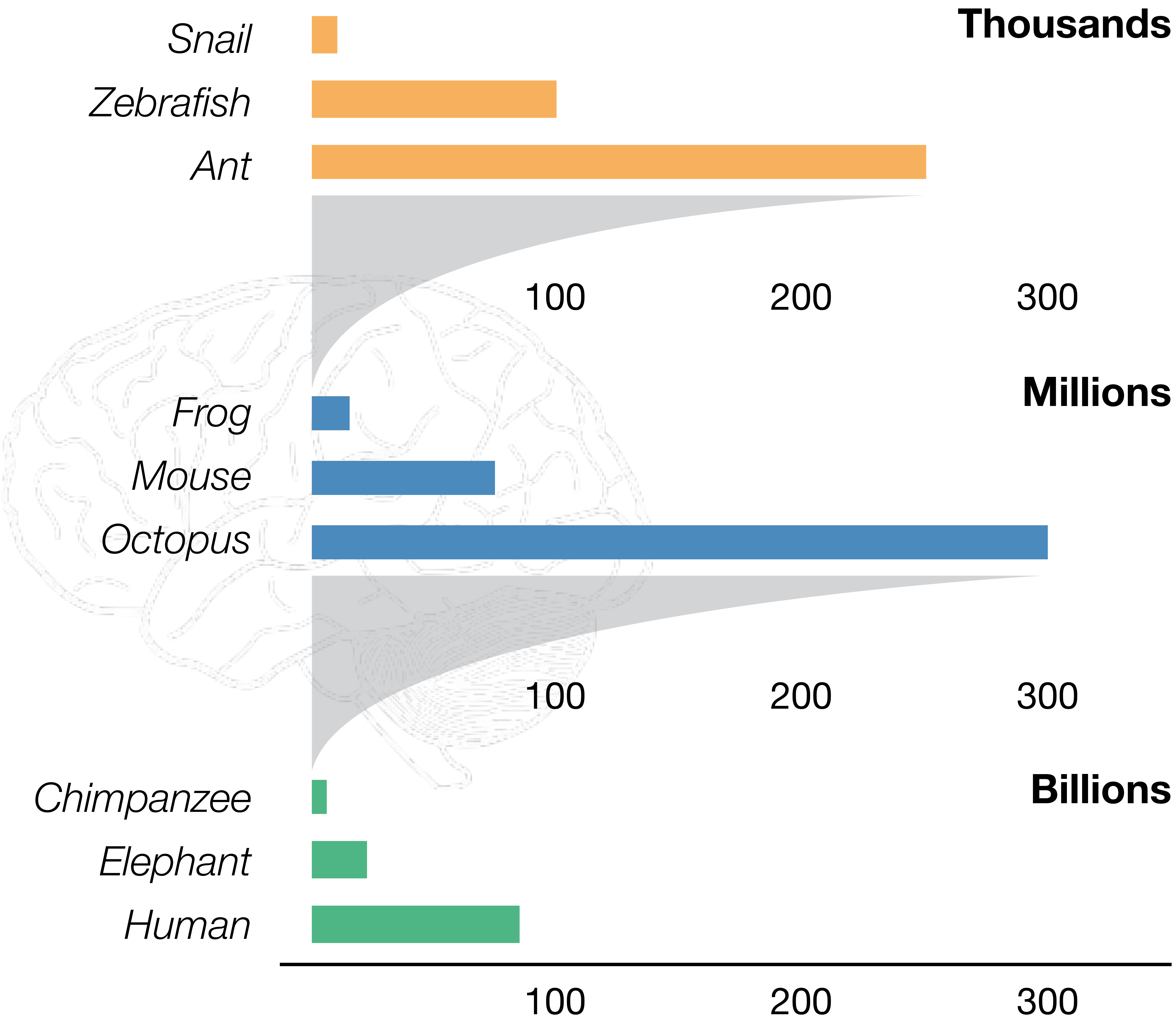
The brain



Stimuli

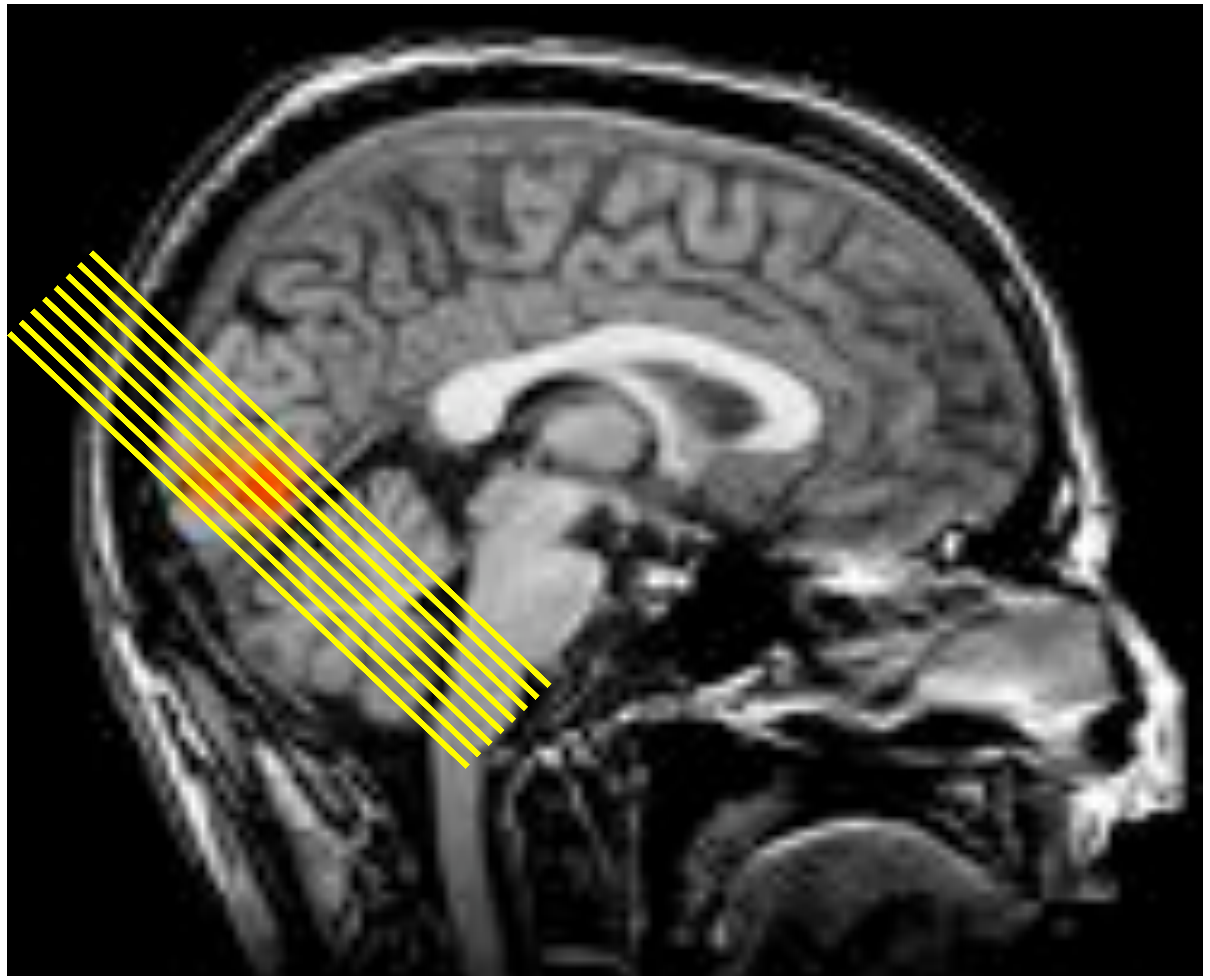
Behavior

Numbers of neurons

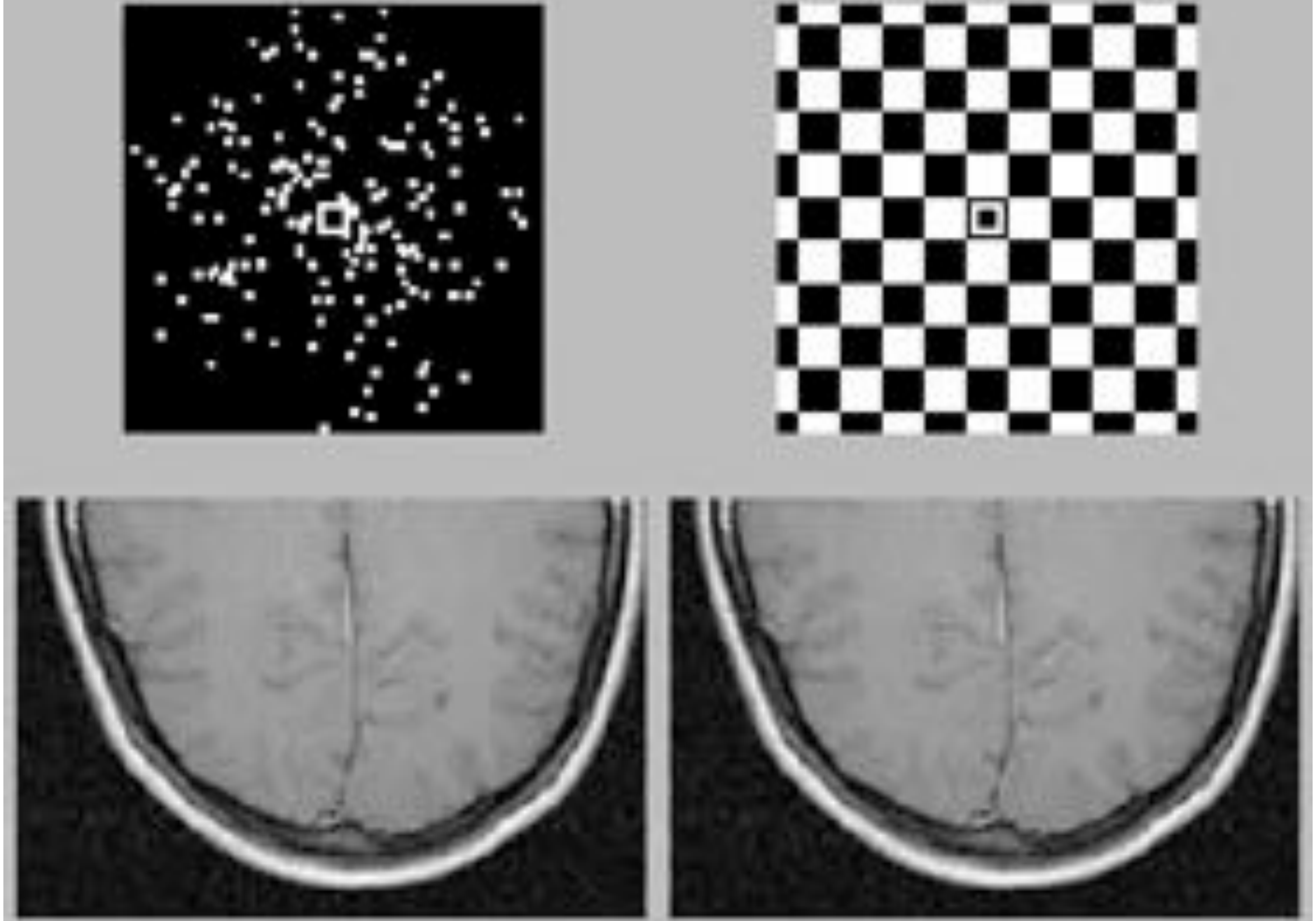


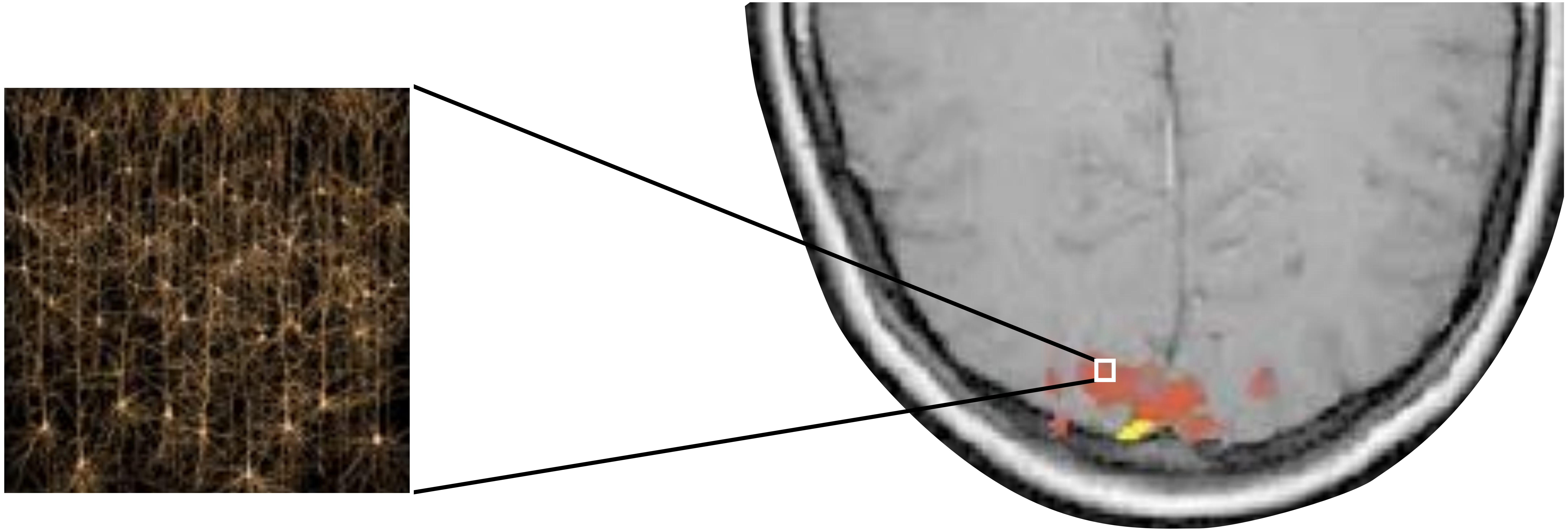
Studying the brain in humans

fMRI scanner



human brain





~50,000 neurons per cubic millimeter

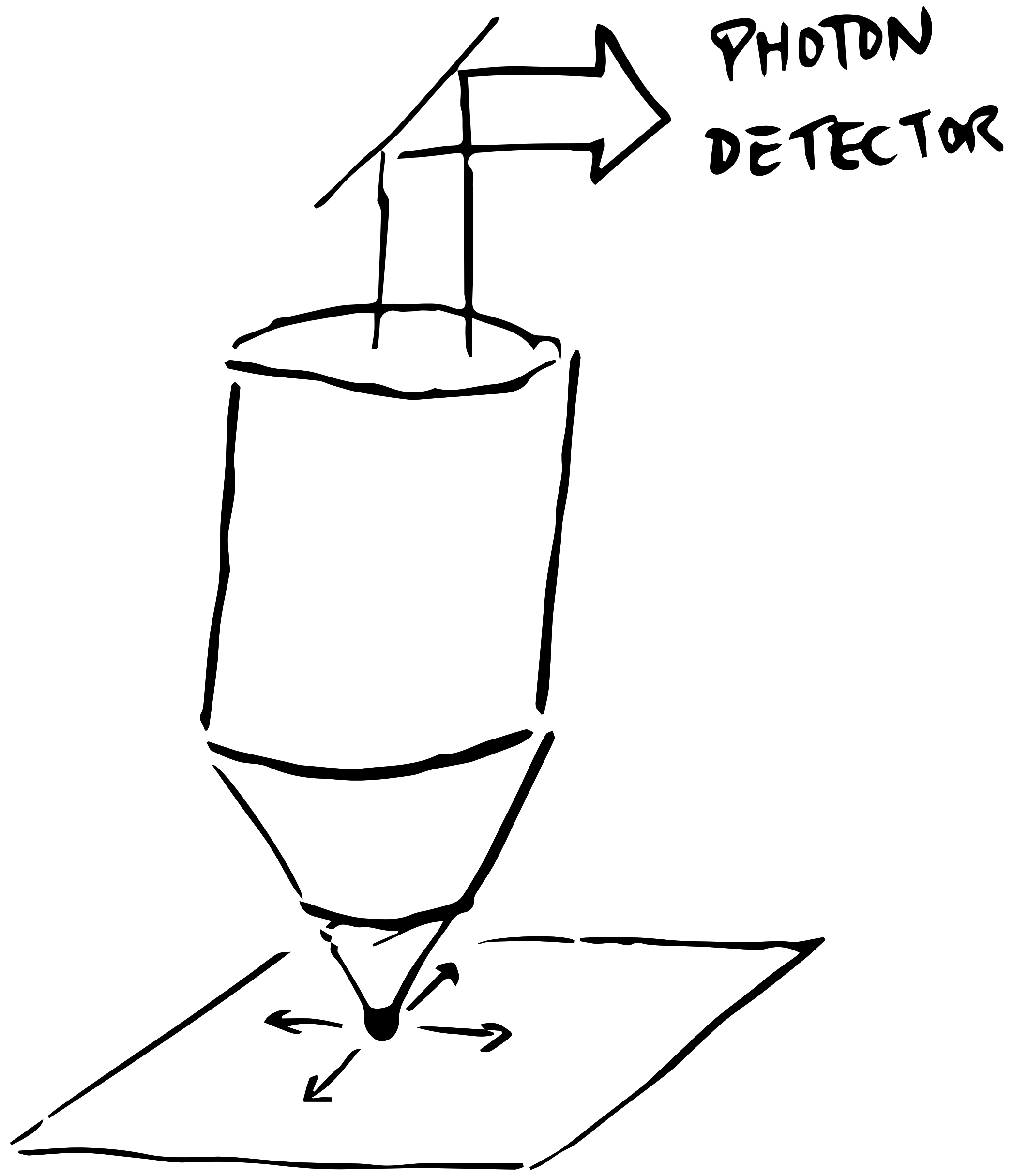
-> need higher resolution!

multielectrode

10-100

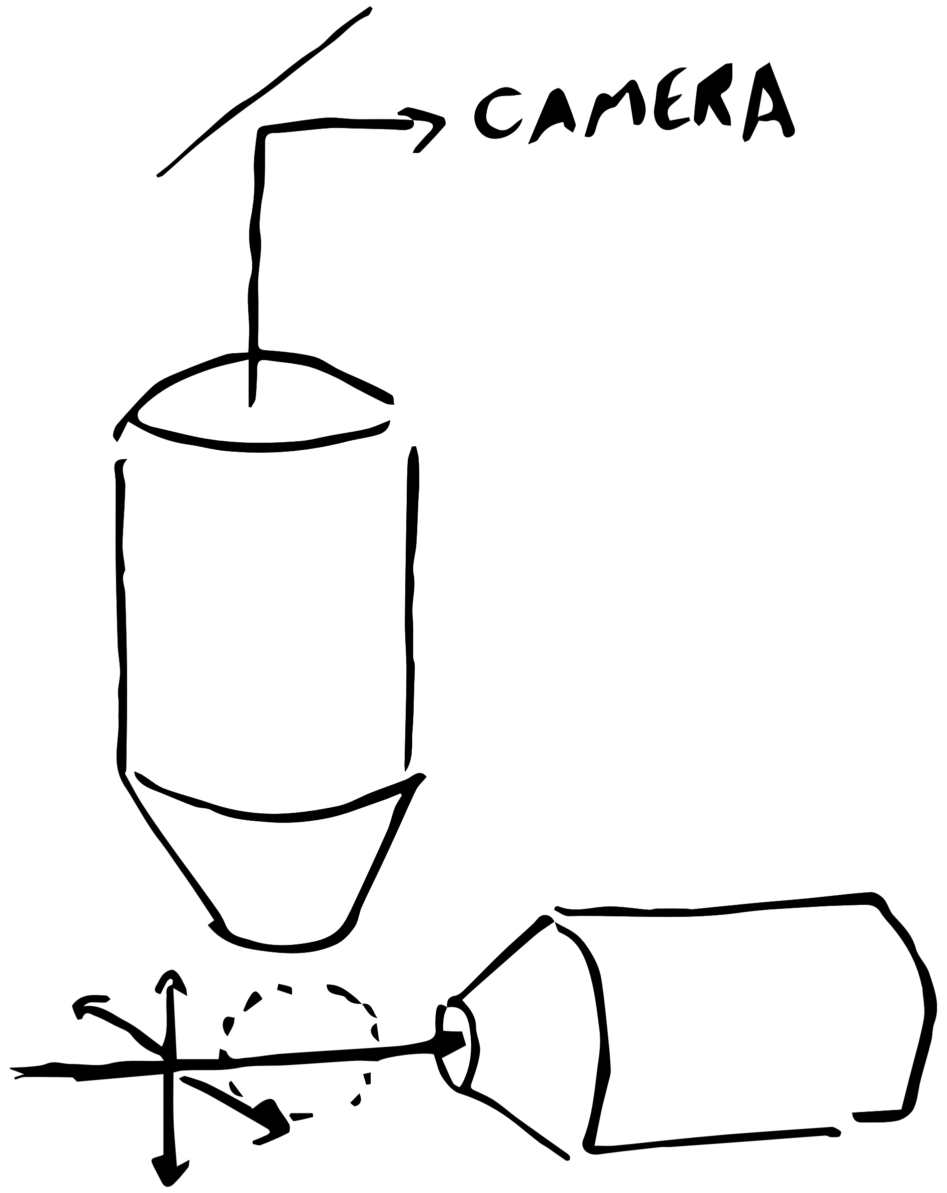
two-photon

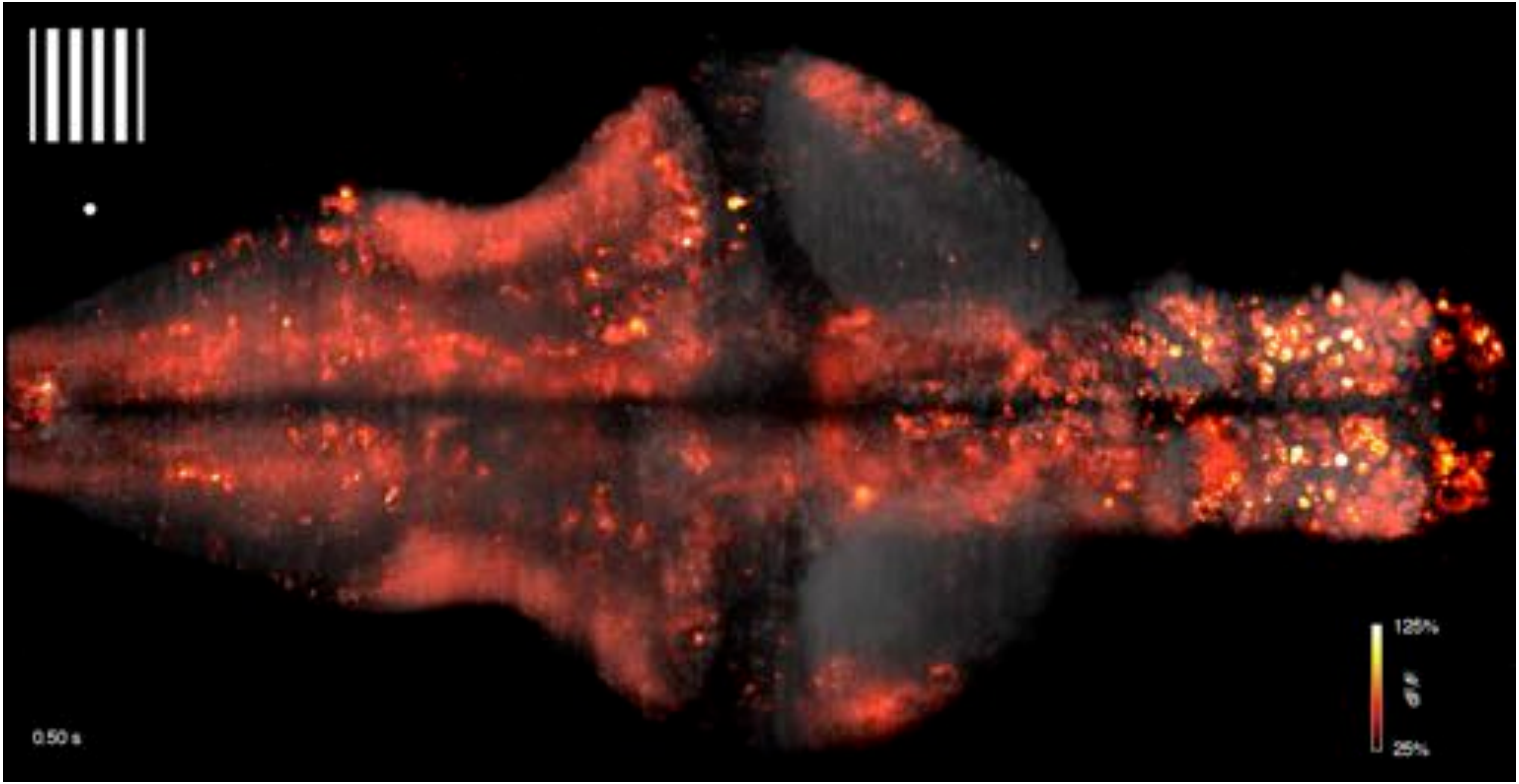
100-1000



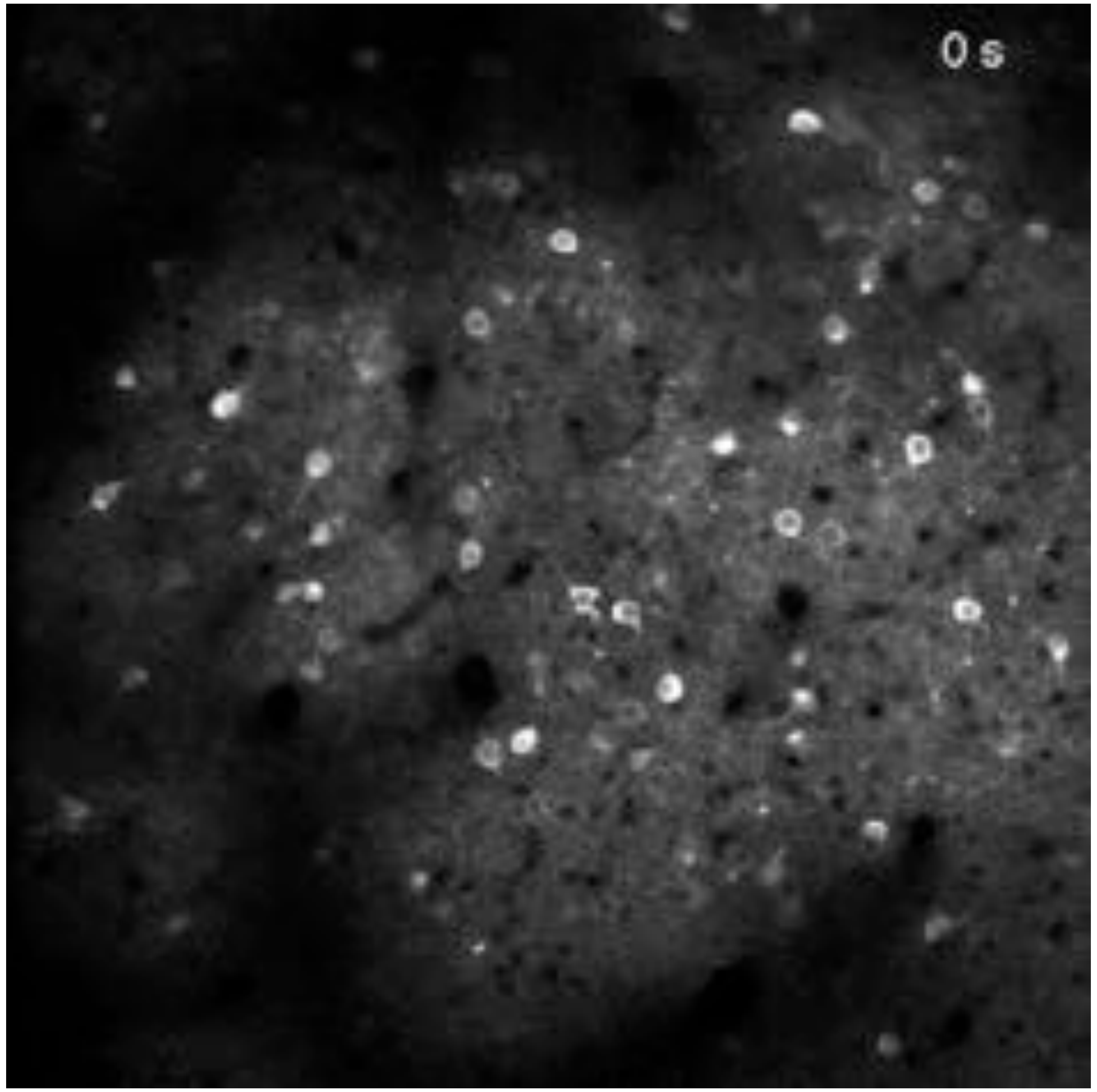
light-sheet

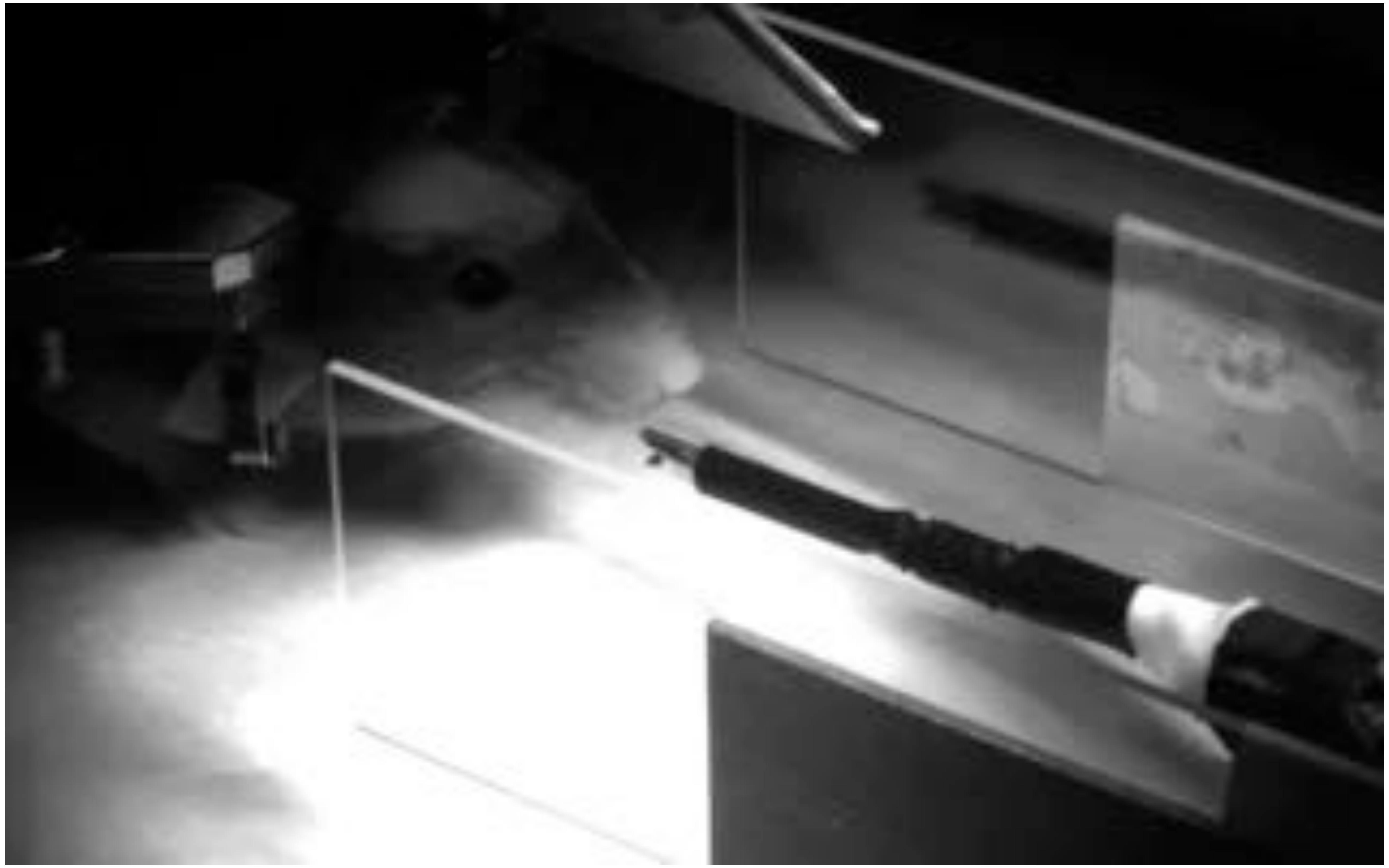
100000





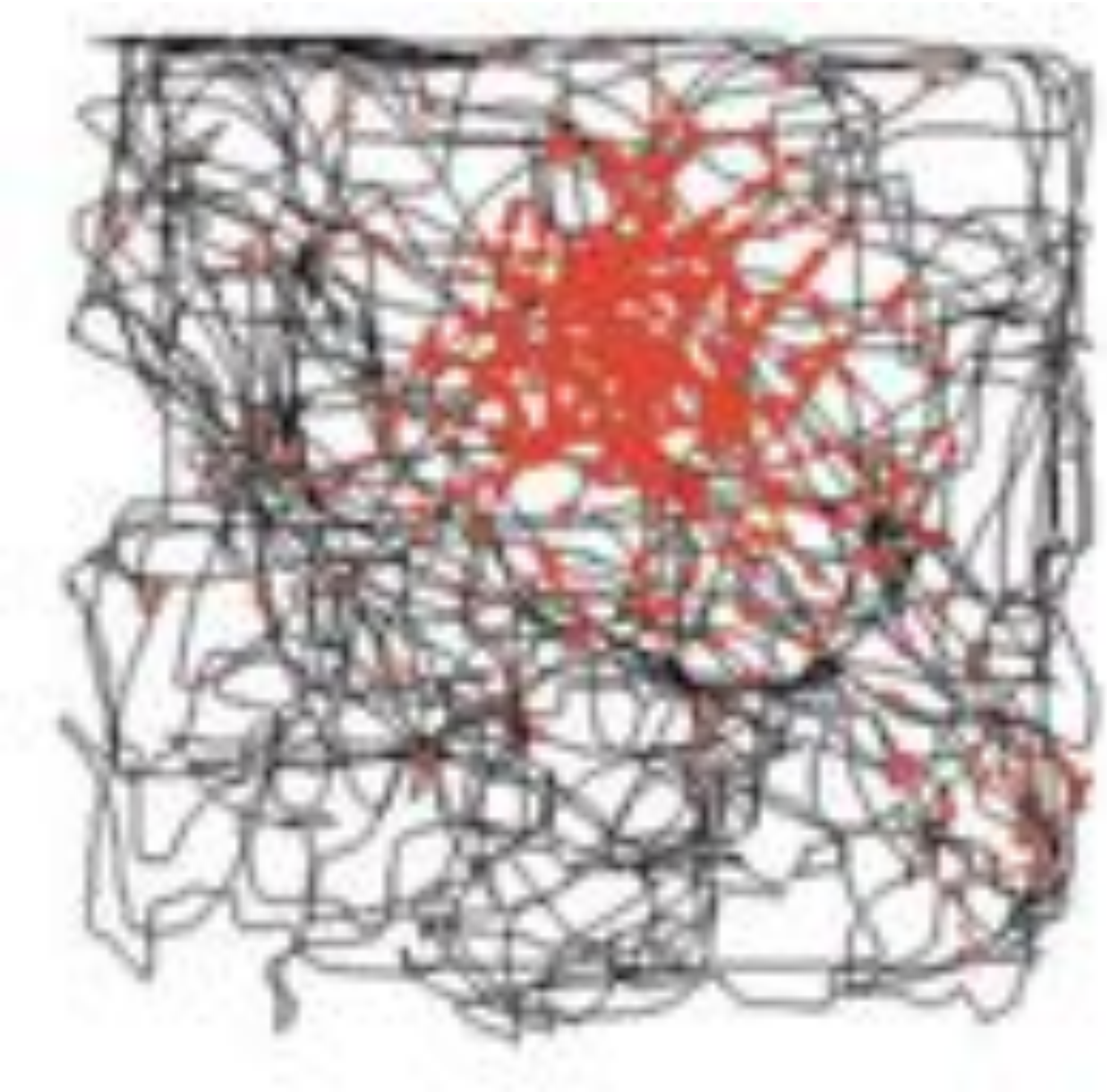
Vladimirov, et al., 2014



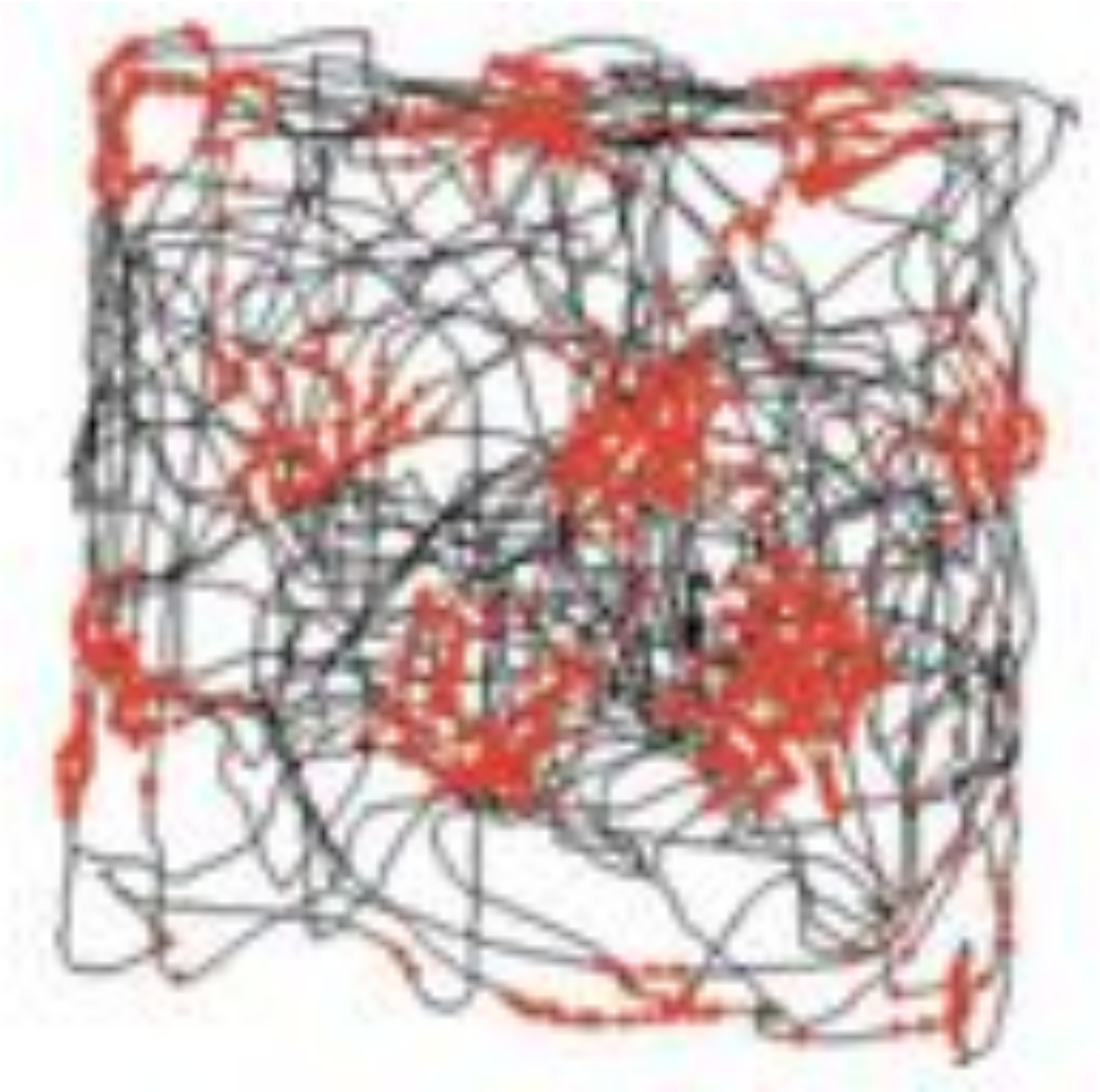


Sofroniew, et al., 2014

relating neuronal responses to properties of an animal and its environment



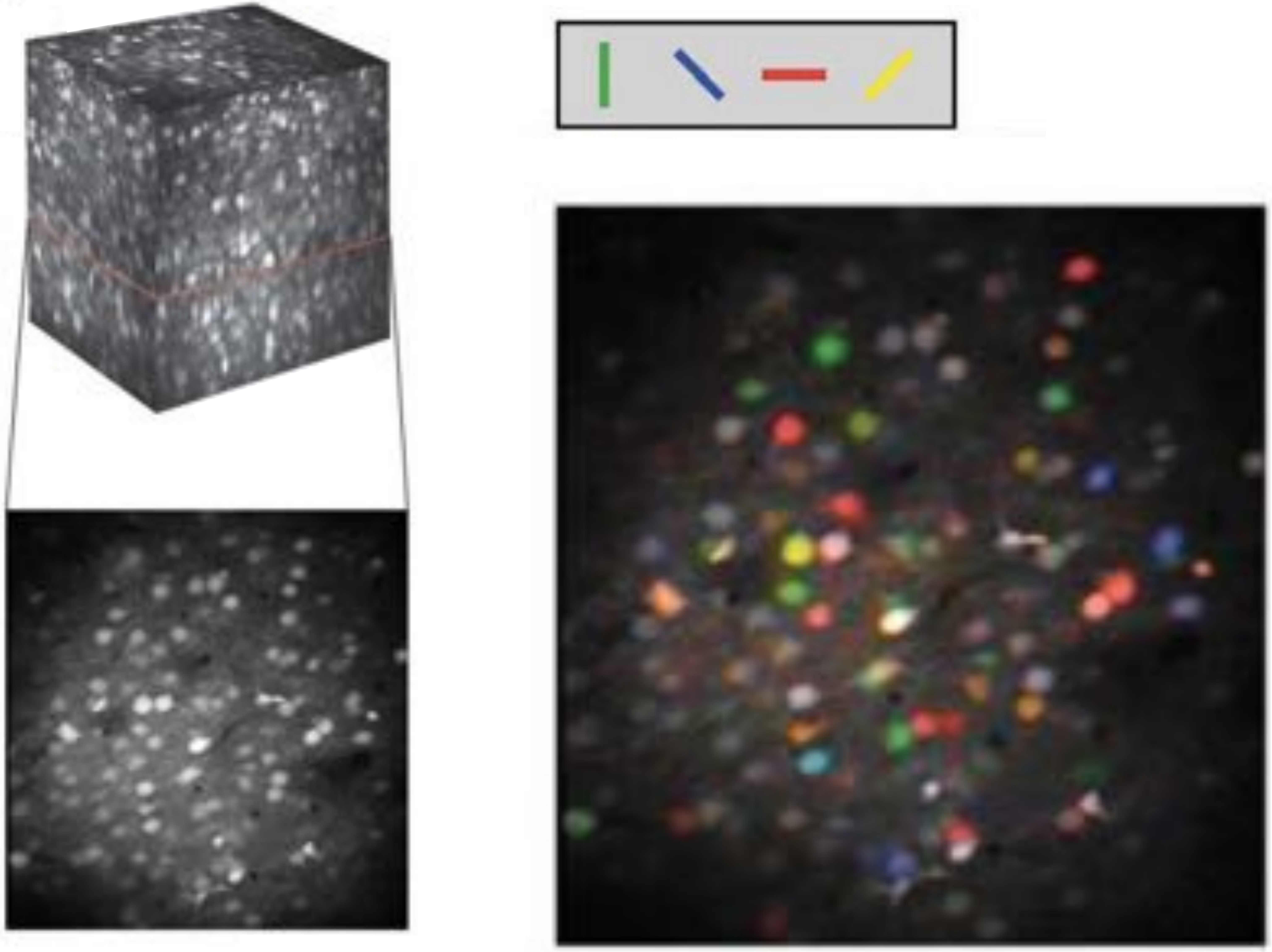
“place cell”



“grid cell”

● position of a
mouse in maze

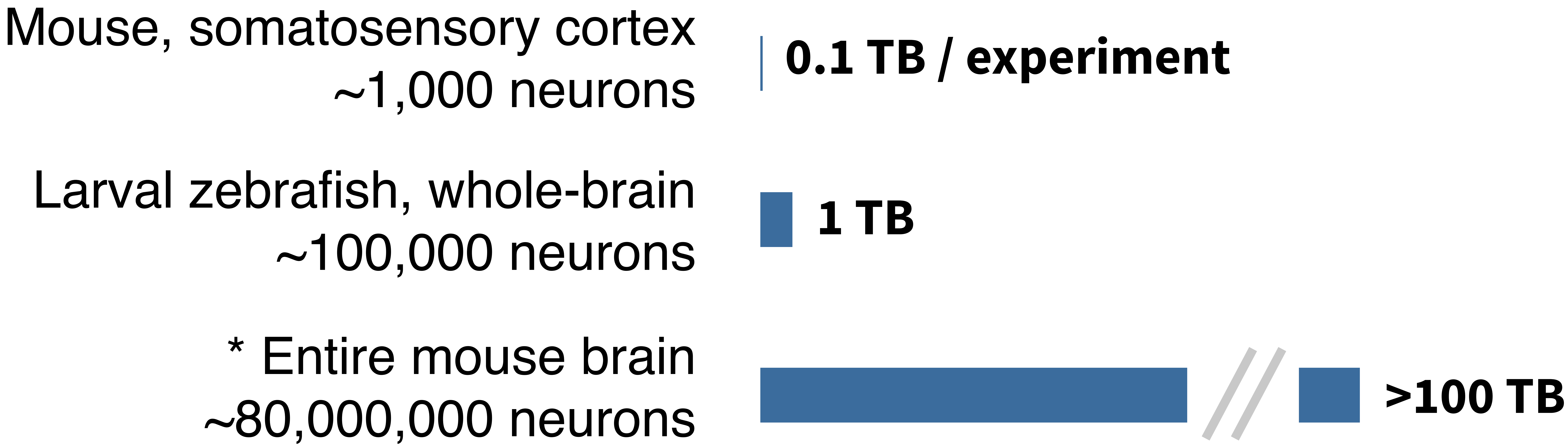
Moser et al., 2008



fine-scale sensory tuning

Hubel & Weisel, 1959

Ohki et al., 2006



Mouse, somatosensory cortex
~1,000 neurons

0.1 TB / experiment

Larval zebrafish, whole-brain
~100,000 neurons

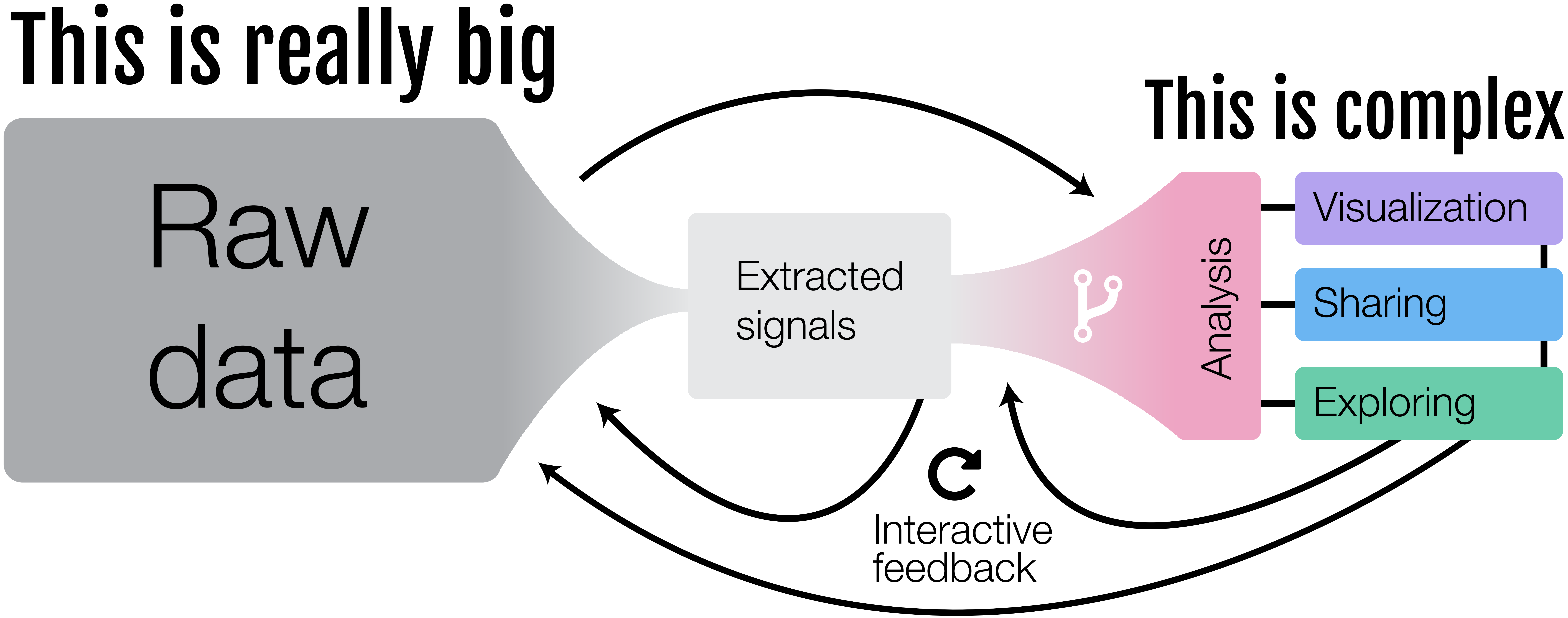
1 TB

* Entire mouse brain
~80,000,000 neurons

>100 TB

* hypothetical

Exploratory
Data Analysis



This is really big

Raw data

Extracted signals

This is complex

Analysis

Visualization

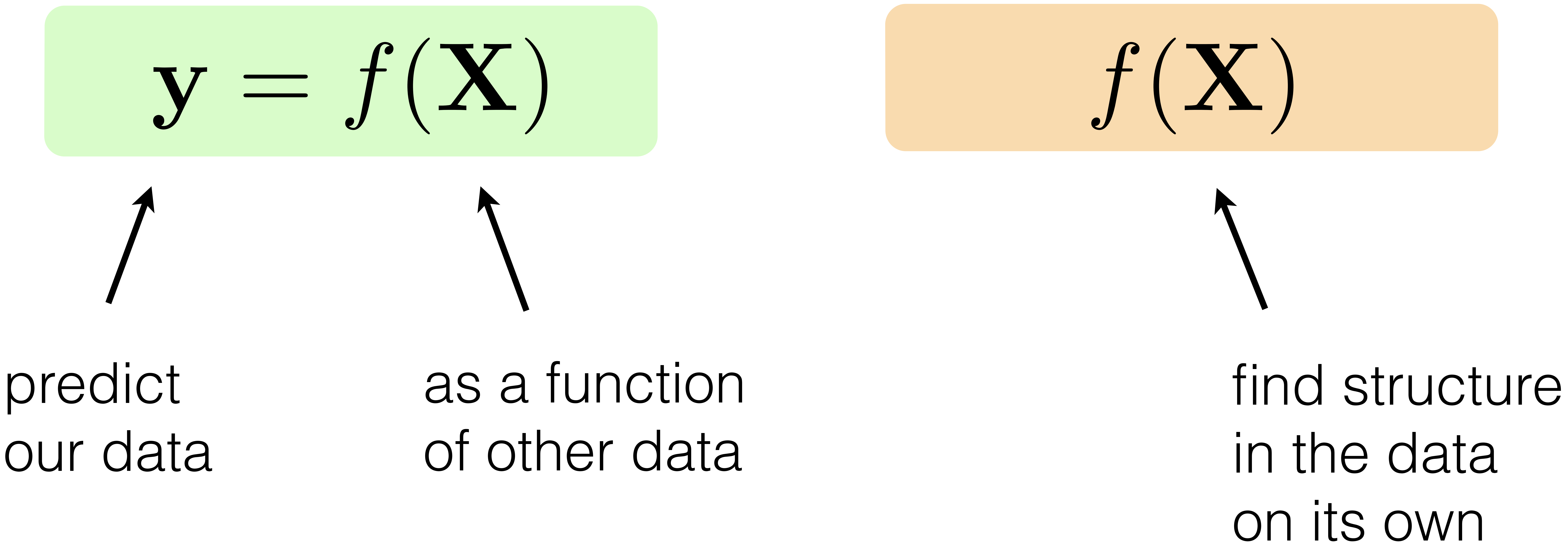
Sharing

Exploring

Interactive feedback

Supervised methods

Unsupervised methods



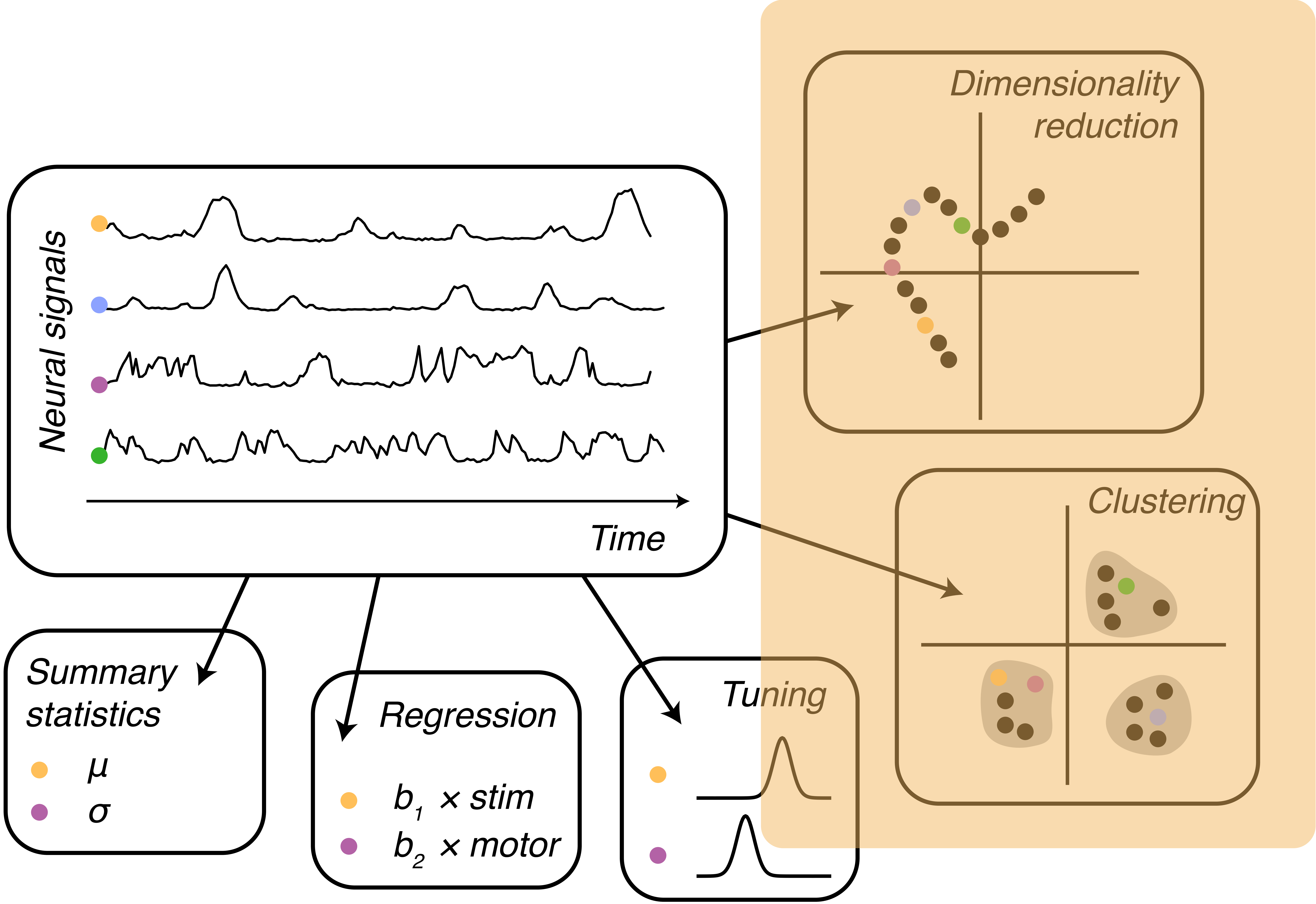
$$\mathbf{y} = f(\mathbf{X})$$

predict
our data

as a function
of other data

$$f(\mathbf{X})$$

find structure
in the data
on its own


Neural signals
Time
Dimensionality reduction
Clustering
Summary statistics
● μ
● σ
Regression
● b_1 × stim
● b_2 × motor
Tuning

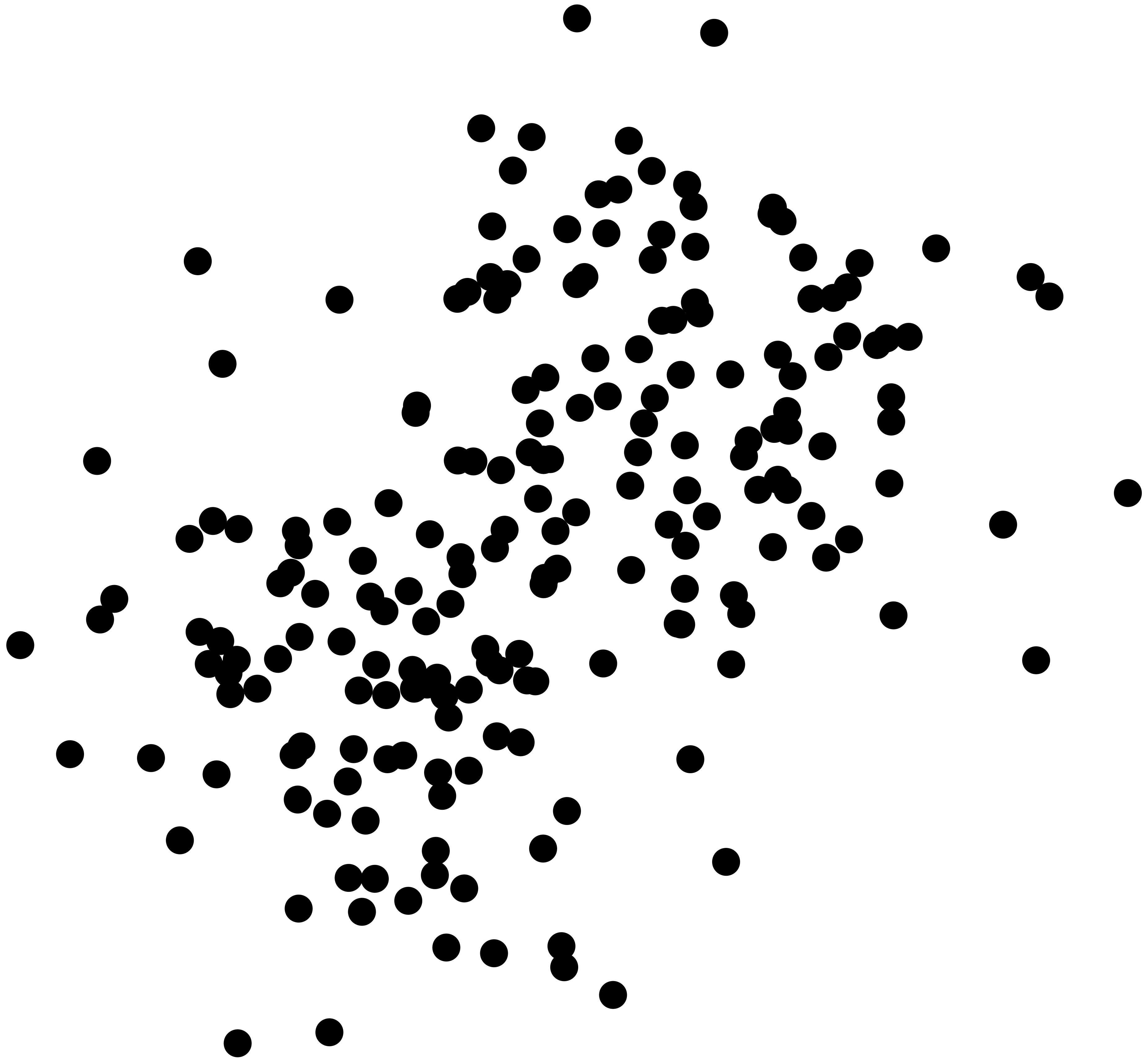
Clustering for preprocessing

- Raw data is complex and high-dimensional

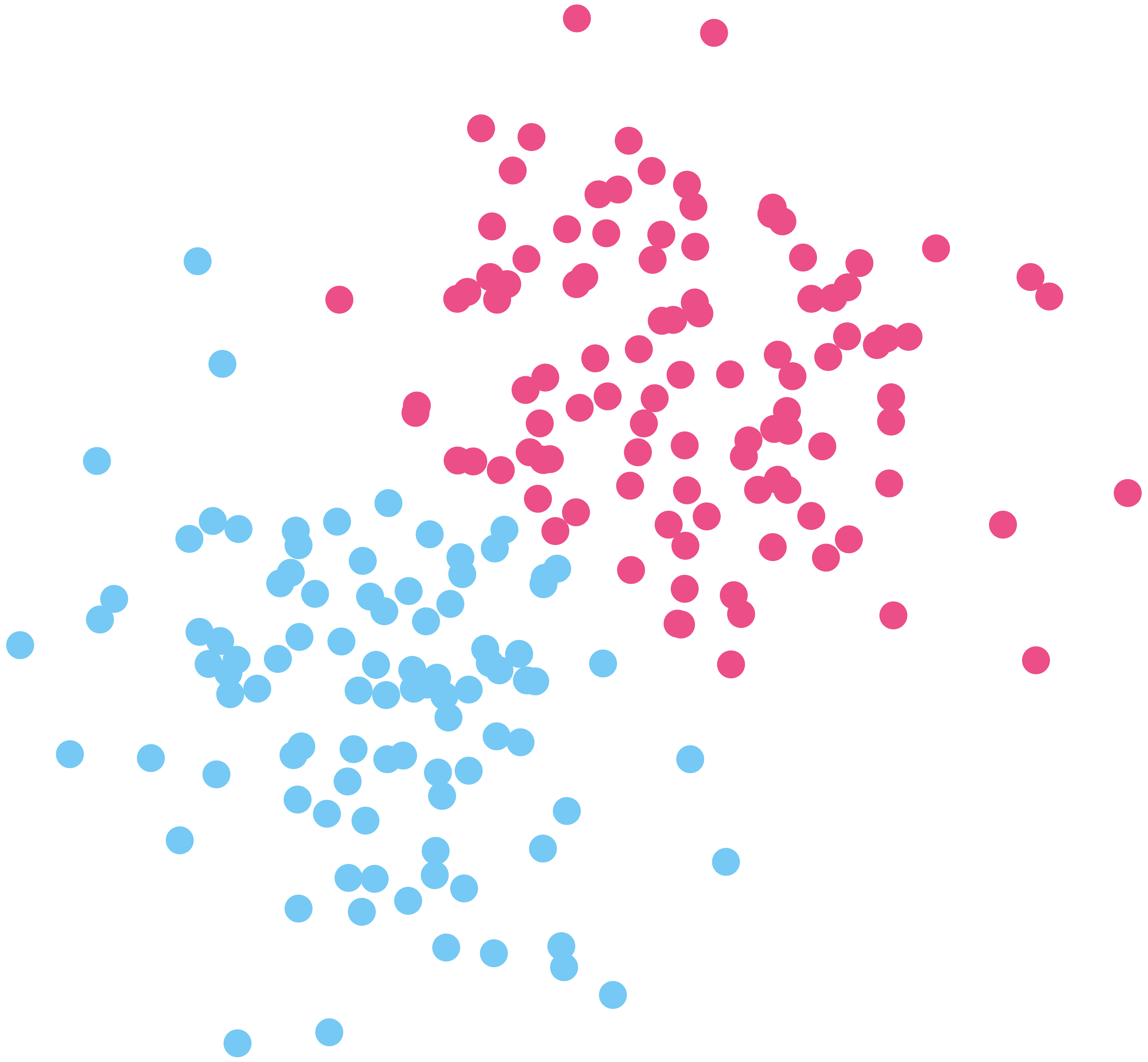
- Clustering finds collections of inputs that are similar to one another

- These groups of clusters may be the more meaningful “unit” of measurement

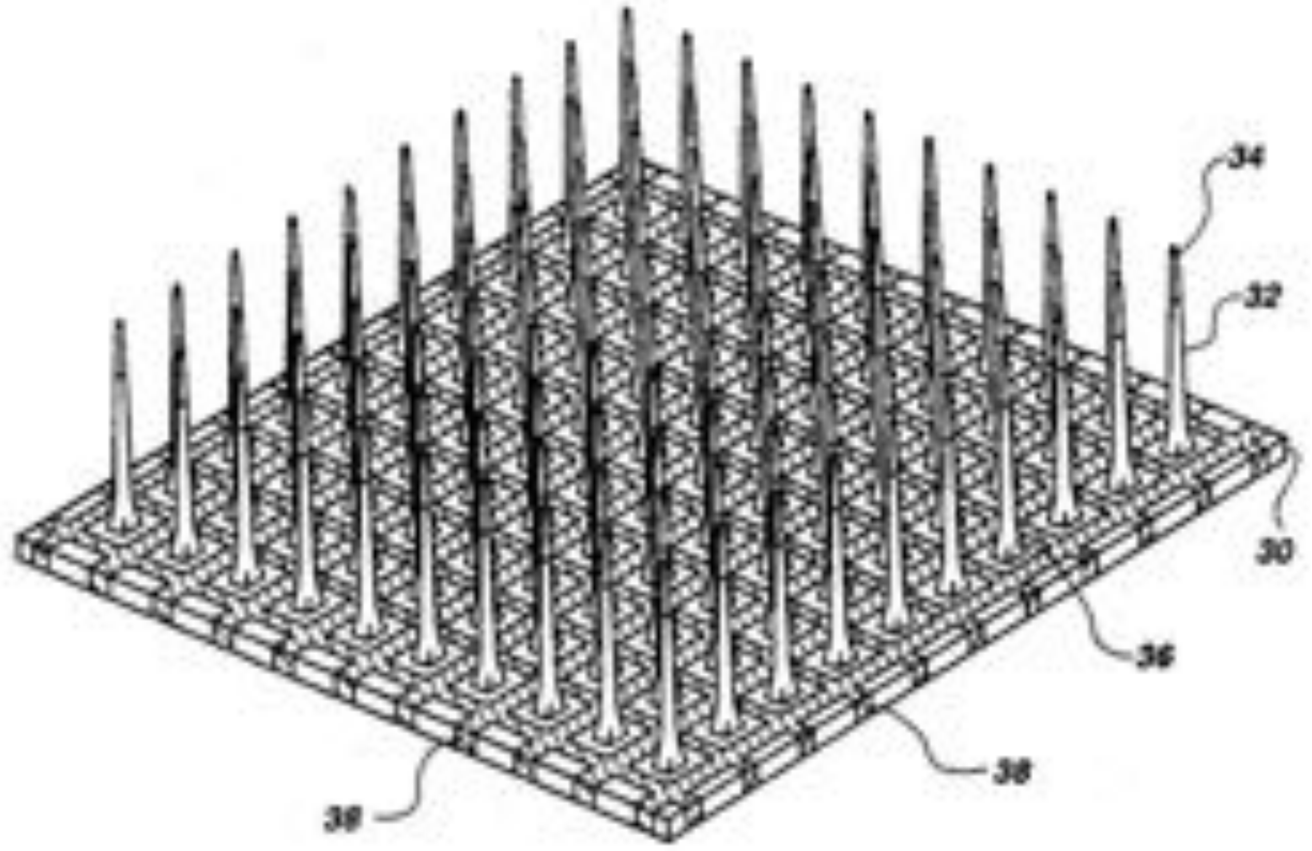
Raw data

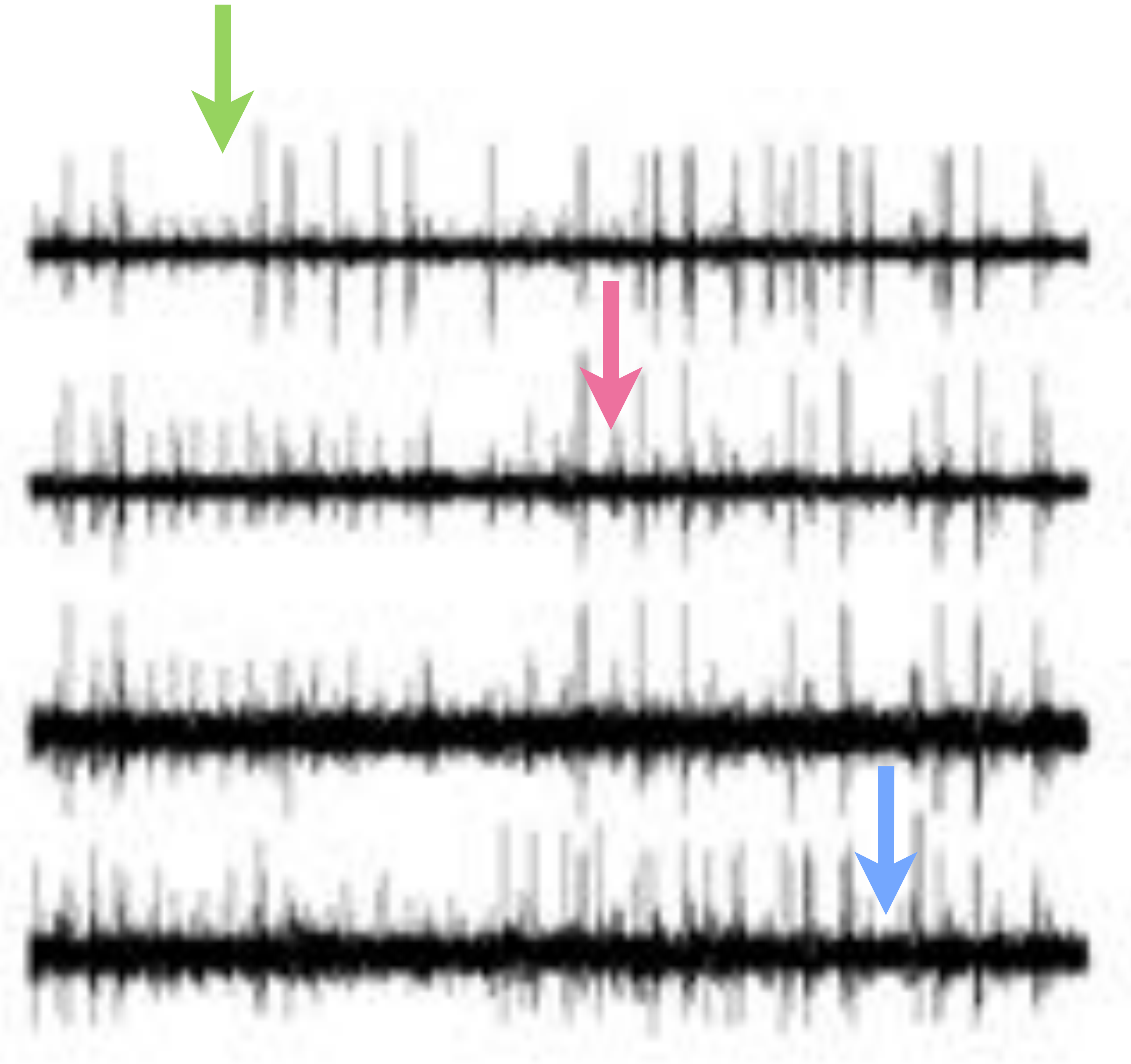


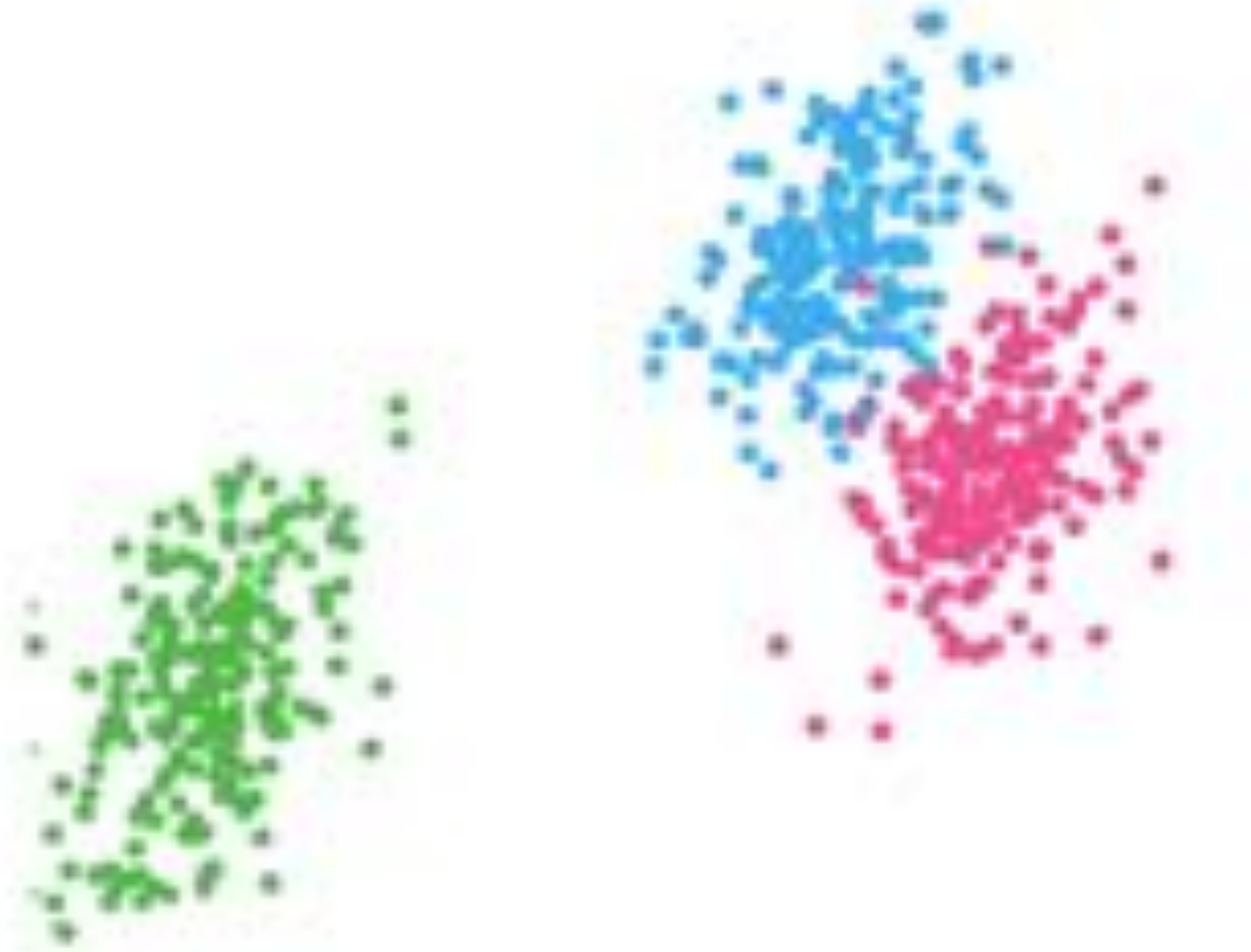
Clustered data



Clustering to find waveforms associated with individual neurons based on their traces across multiple electrodes





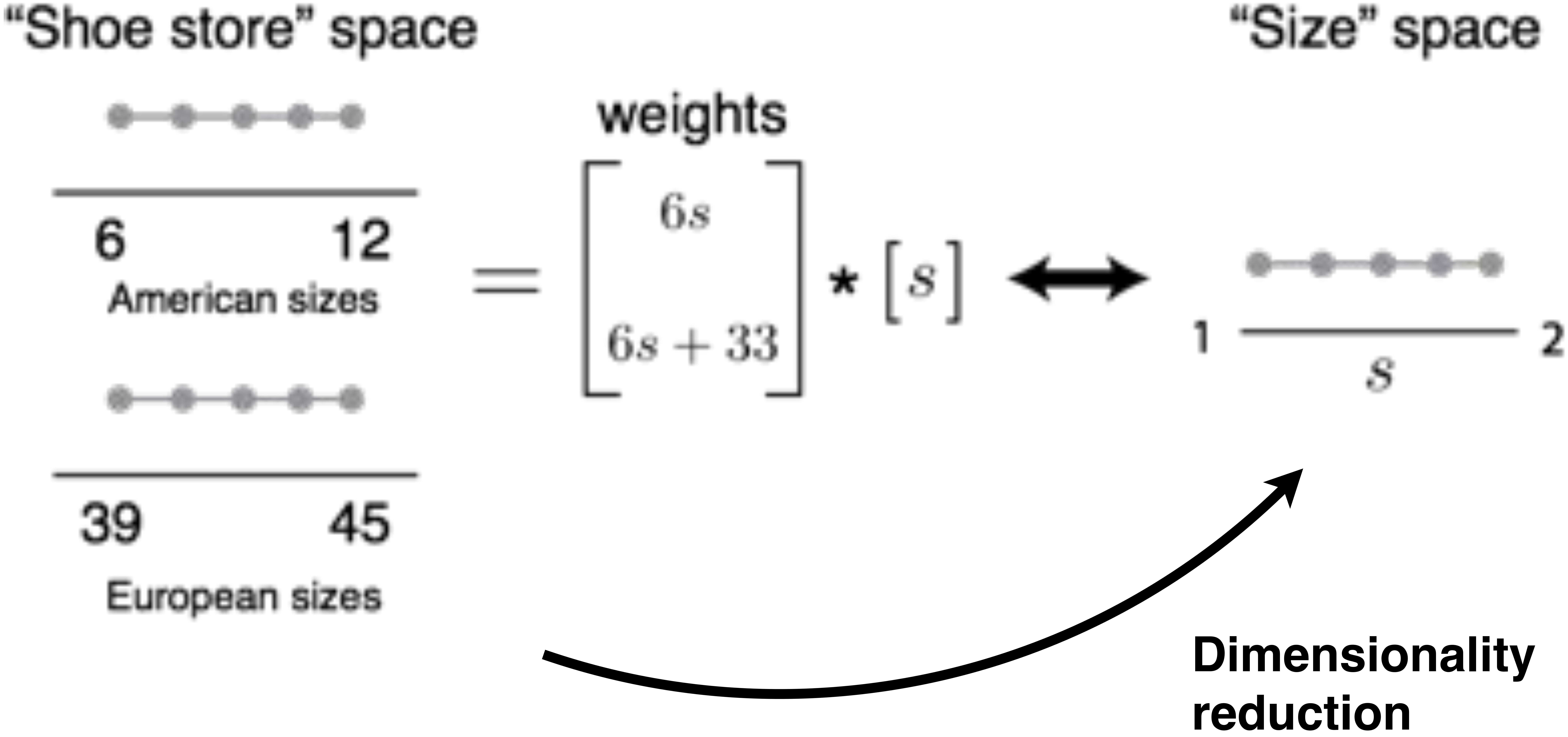


Dimensionality reduction for insight

- Raw data is complex and high-dimensional

- Dimensionality reduction describes the data using a simpler, more compact representation

- This representation may make interesting patterns in the data more clear or easier to see



“Shoe store” space

6 12

American sizes

39 45

European sizes

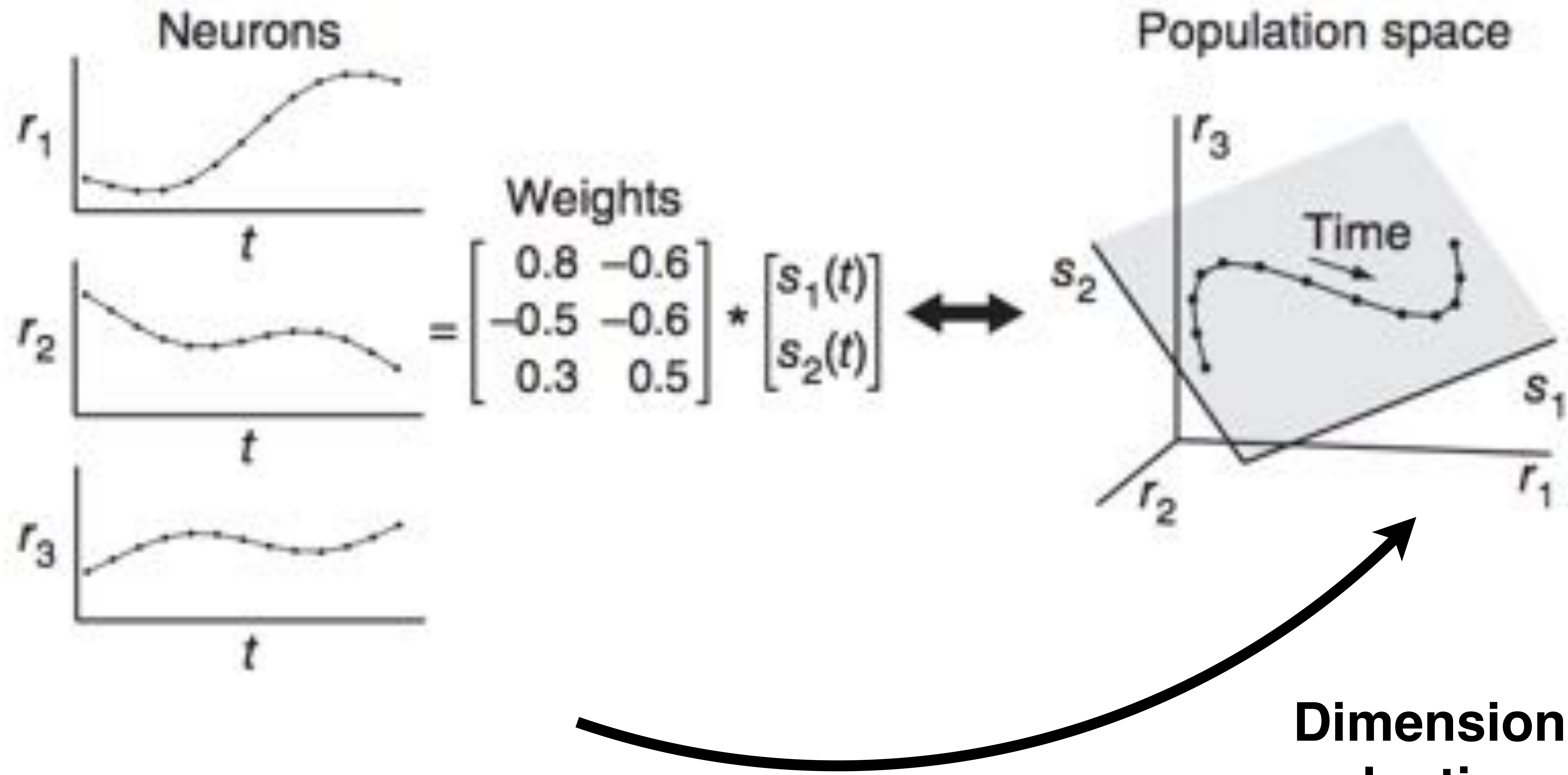
weights

$$= \begin{bmatrix} 6s \\ 6s + 33 \end{bmatrix} \star [s] \longleftrightarrow$$

“Size” space

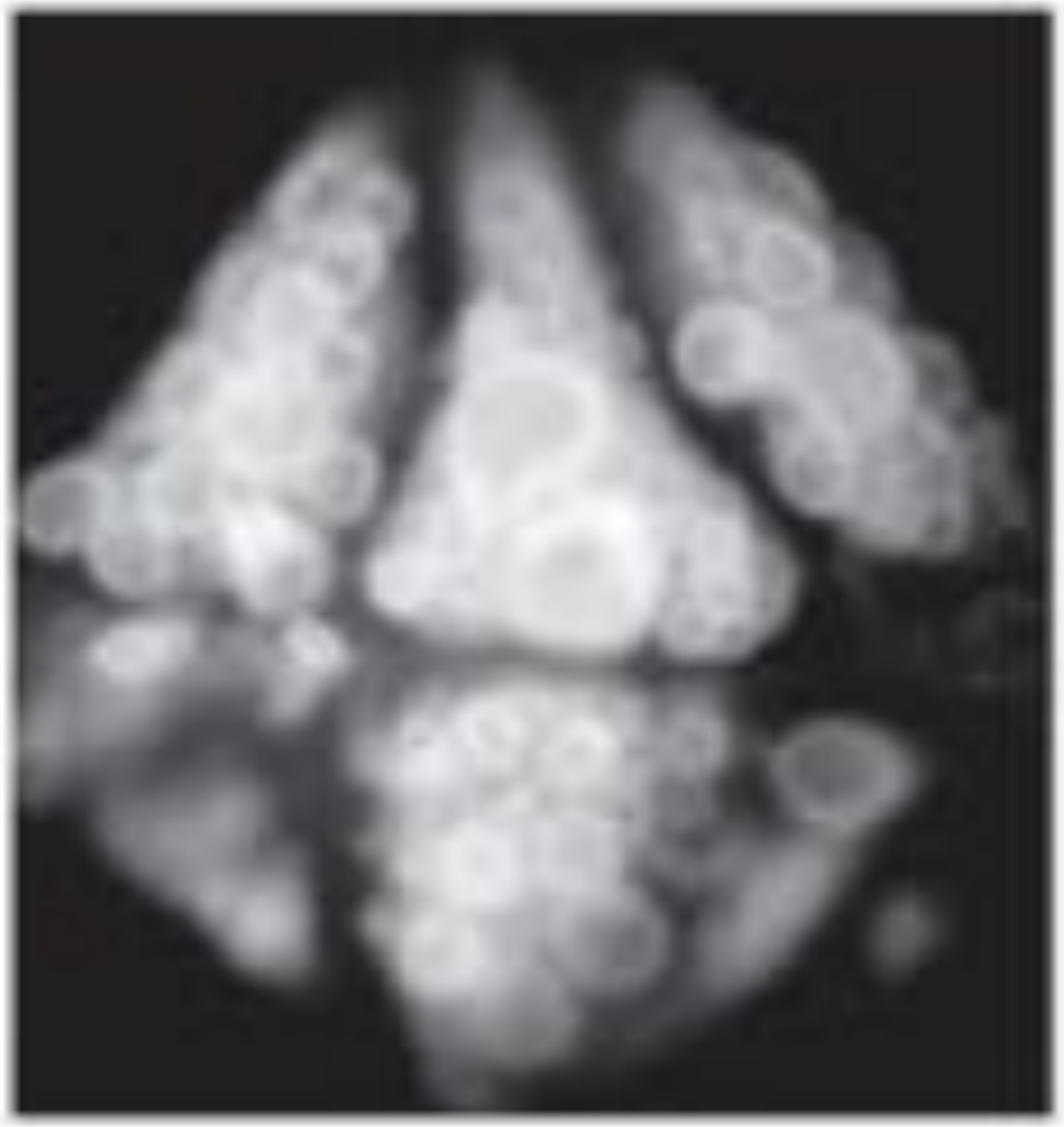
1 s 2

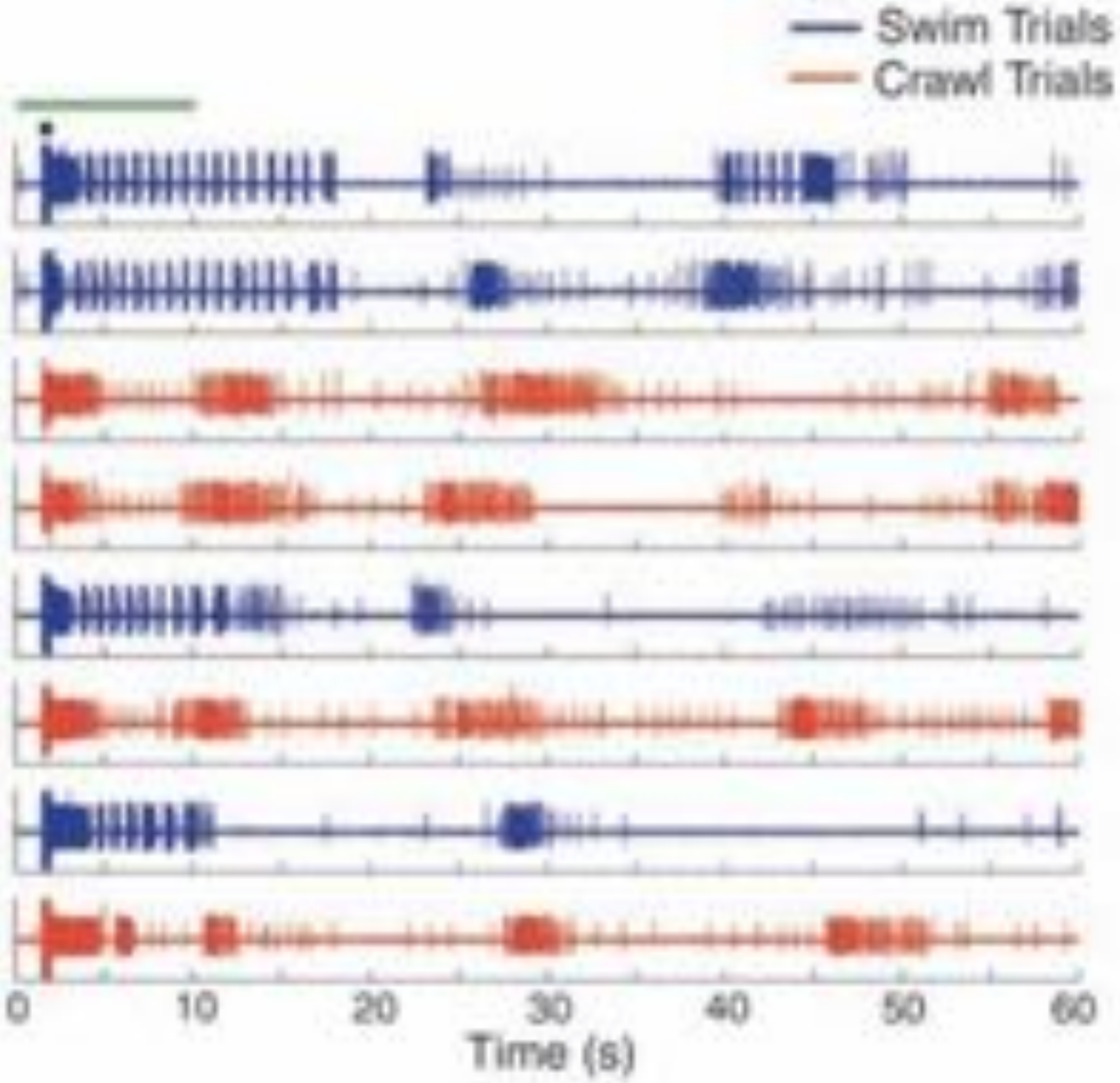
Dimensionality reduction


Neurons
r_1
t
r_2
t
r_3
t
Weights
$= \begin{bmatrix} 0.8 & -0.6 \\ -0.5 & -0.6 \\ 0.3 & 0.5 \end{bmatrix} * \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix}$
Population space
r_3
Time
s_2
s_1
r_2
r_1

Dimensionality reduction

Yu and Cunningham, 2014

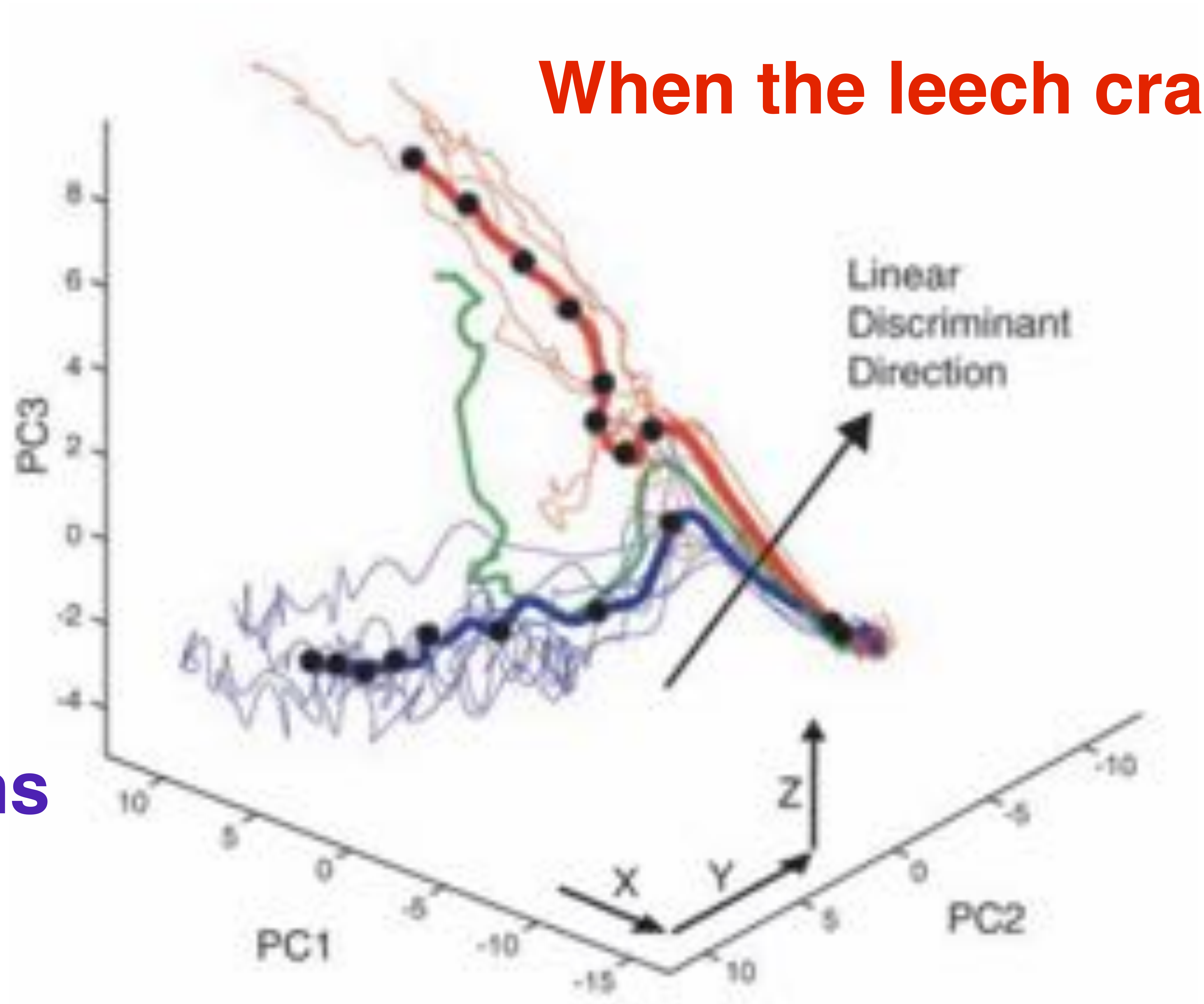




Briggman et al., 2005

When the leech changes its mind!

When the leech crawls



When the leech swims

Briggman et al., 2005

Principal Component Analysis (PCA) Overview

Raw data can be Complex, High-dimensional

To understand a phenomenon we measure various related quantities

If we knew what to measure or how to represent our measurements we might find simple relationships

But in practice we often *measure redundant signals*, e.g., US and European shoe sizes

We also *represent data via the method by which it was gathered*, e.g., pixel representation of brain imaging data

Dimensionality Reduction

Issues

- *Measure redundant signals*
- *Represent data via the method by which it was gathered*

Goal: Find a ‘better’ representation for data

- To visualize and discover hidden patterns
- Preprocessing for supervised task, e.g., feature hashing

How do we define ‘better’?

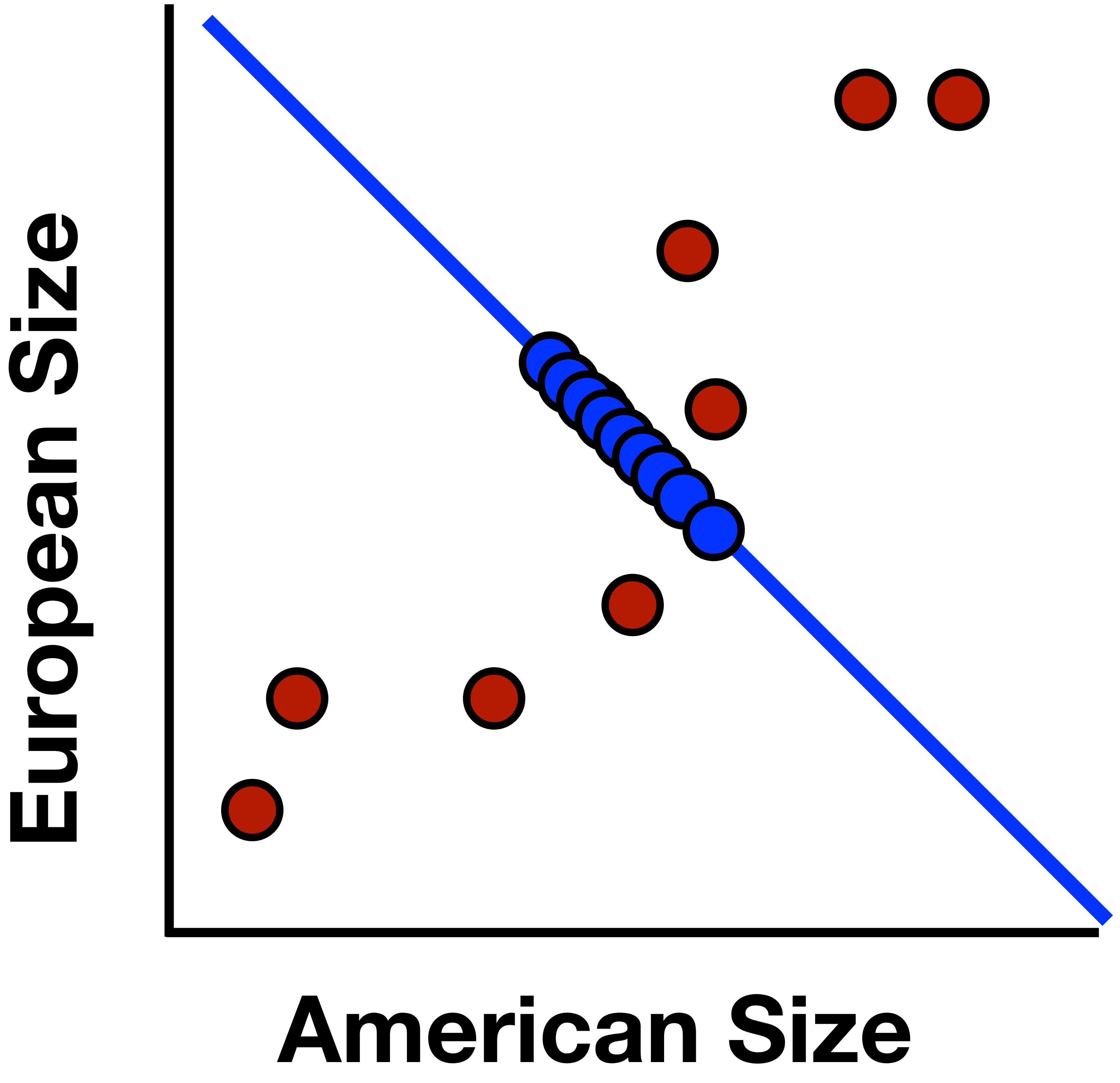
E.g., Shoe Size

We take noisy measurements on European and American scale

- Modulo noise, we expect perfect correlation

How can we do ‘better’, i.e., find a simpler, compact representation?

- Pick a direction and project onto this direction



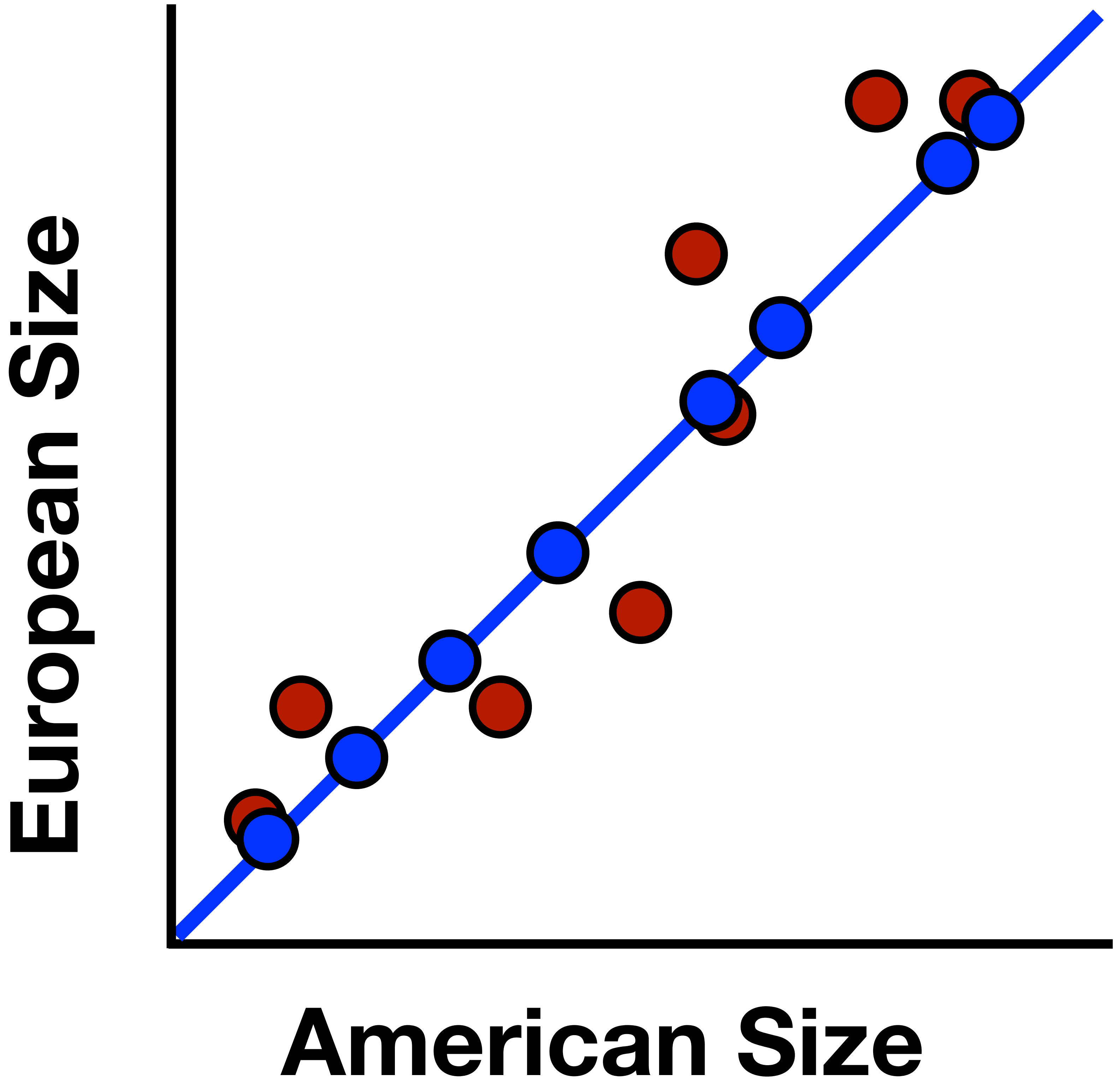
E.g., Shoe Size

We take noisy measurements on European and American scale

- Modulo noise, we expect perfect correlation

How can we do ‘better’, i.e., find a simpler, compact representation?

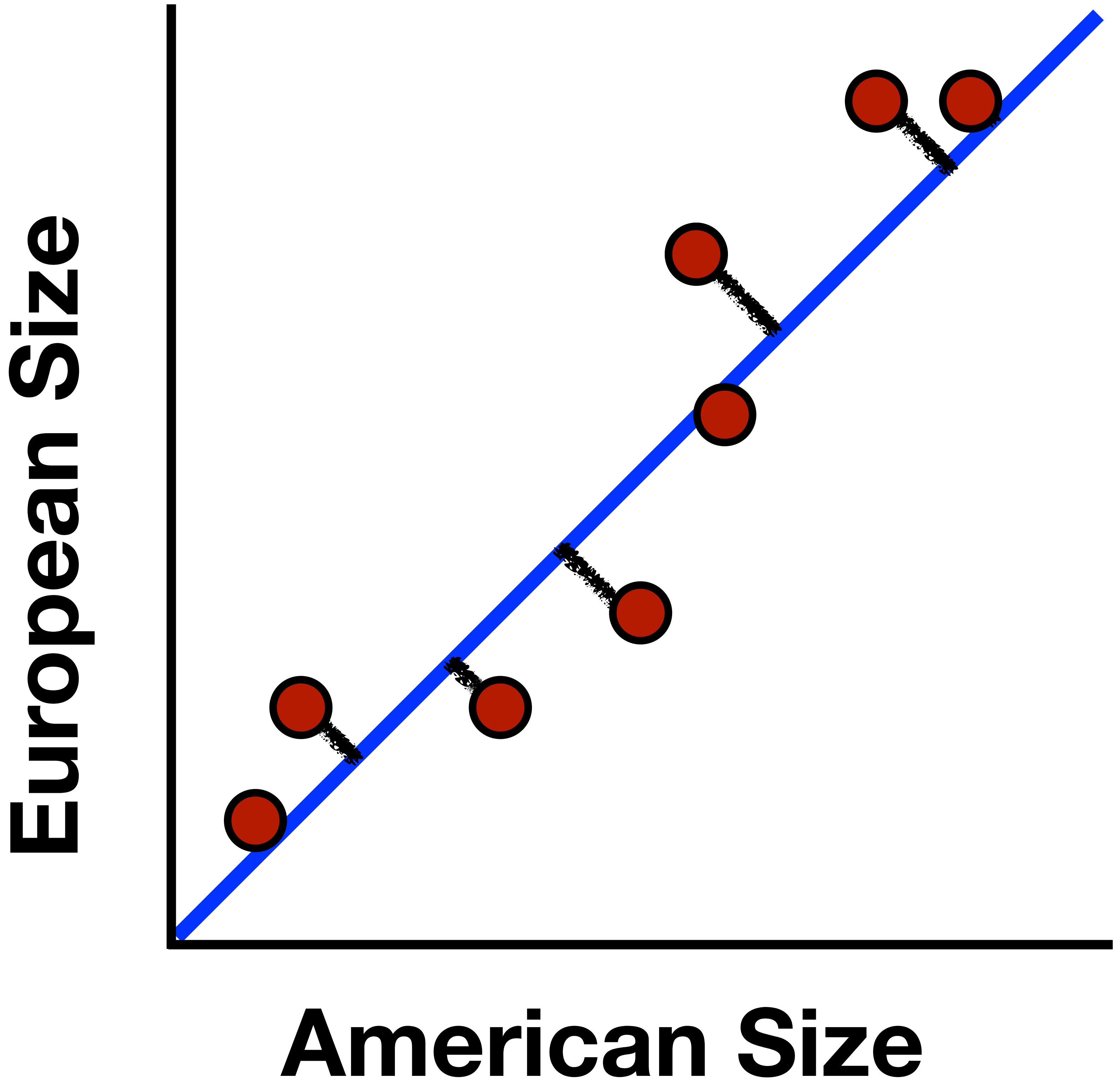
- Pick a direction and project onto this direction



Goal: Minimize Reconstruction Error

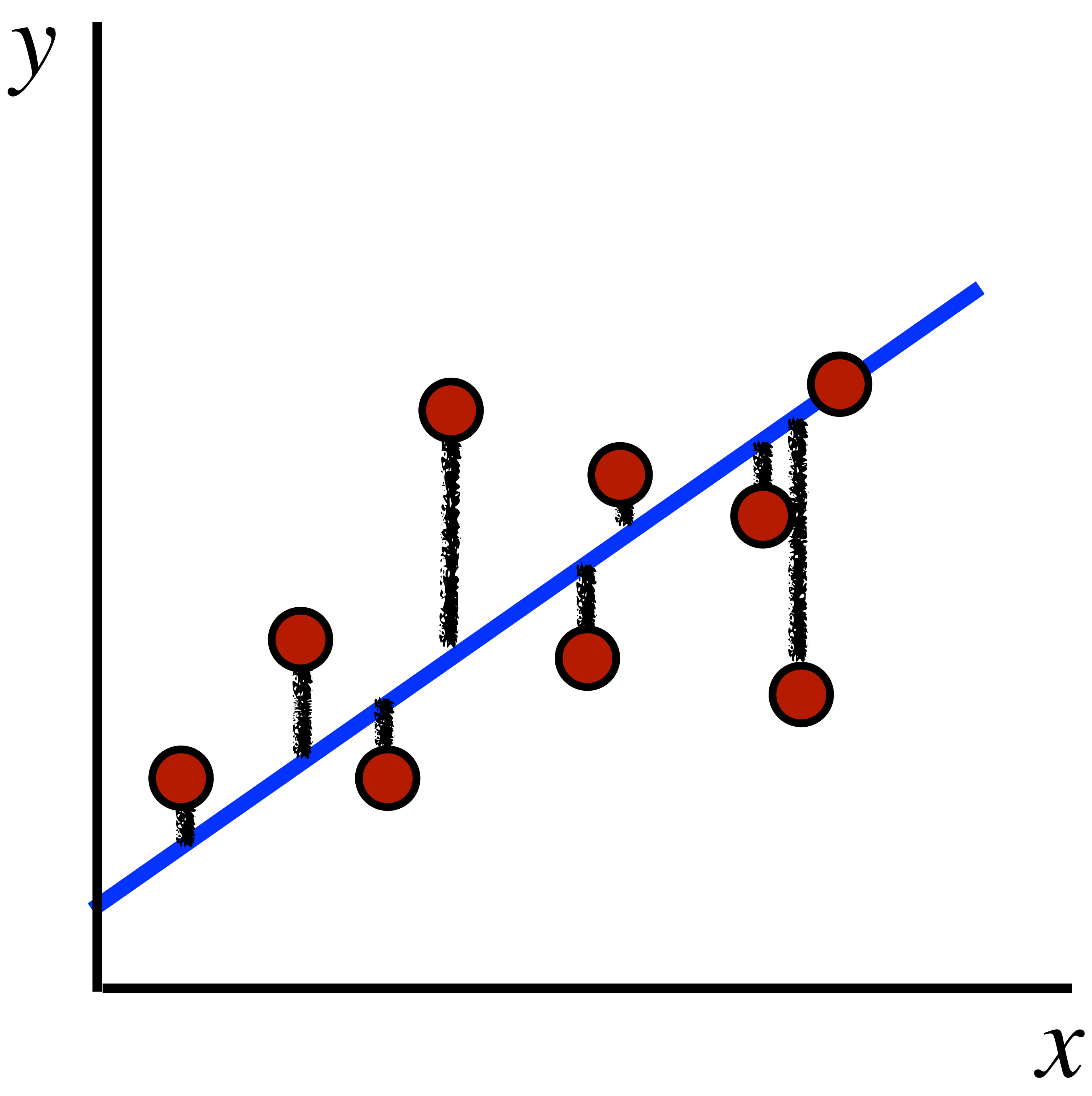
Minimize Euclidean distances between original points and their projections

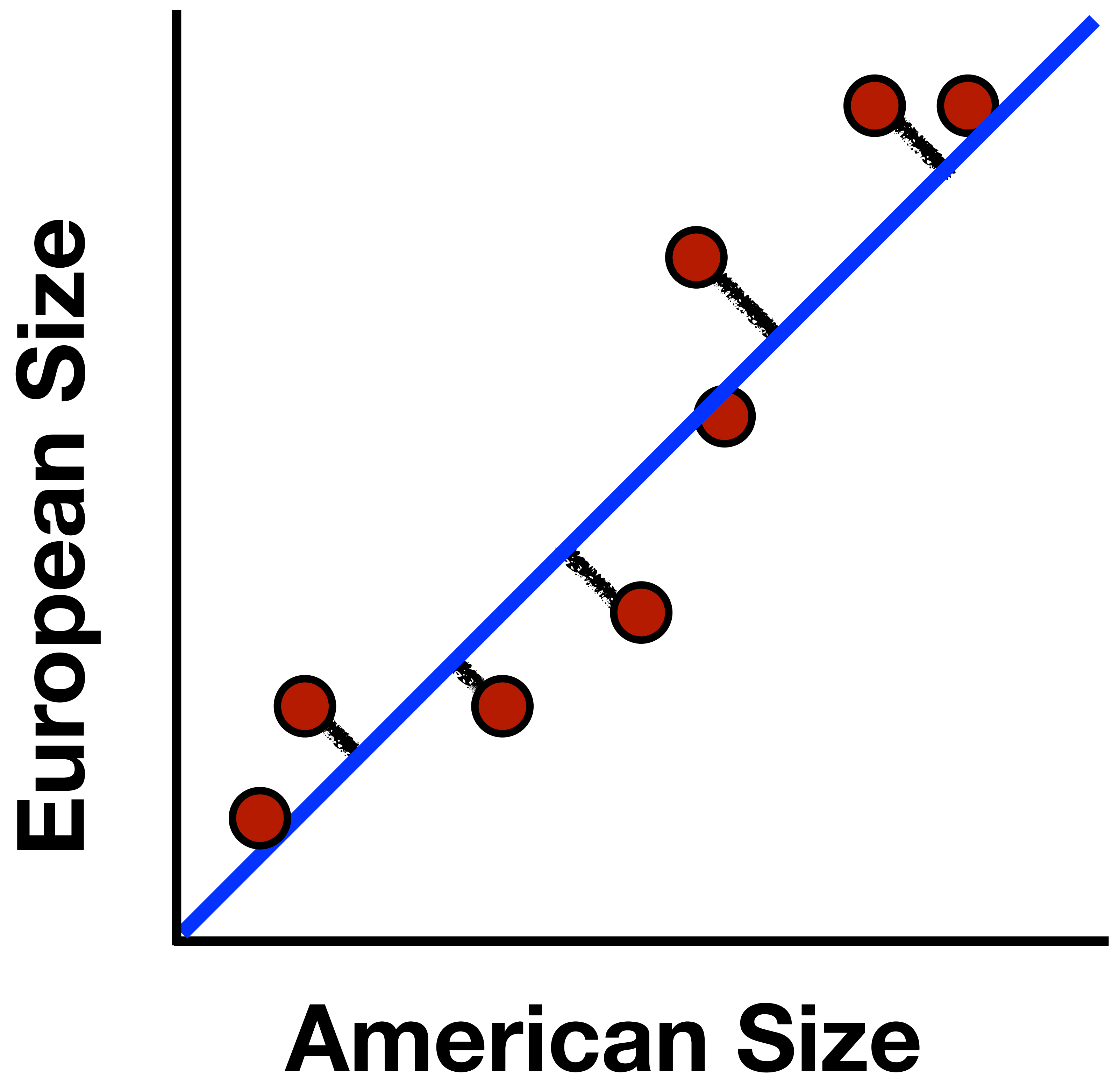
PCA solution solves this problem!



Linear Regression — predict y from x. Evaluate accuracy of predictions (represented by blue line) by **vertical** distances between points and the line

PCA — reconstruct 2D data via 2D data with single degree of freedom. Evaluate reconstructions (represented by blue line) by **Euclidean** distances

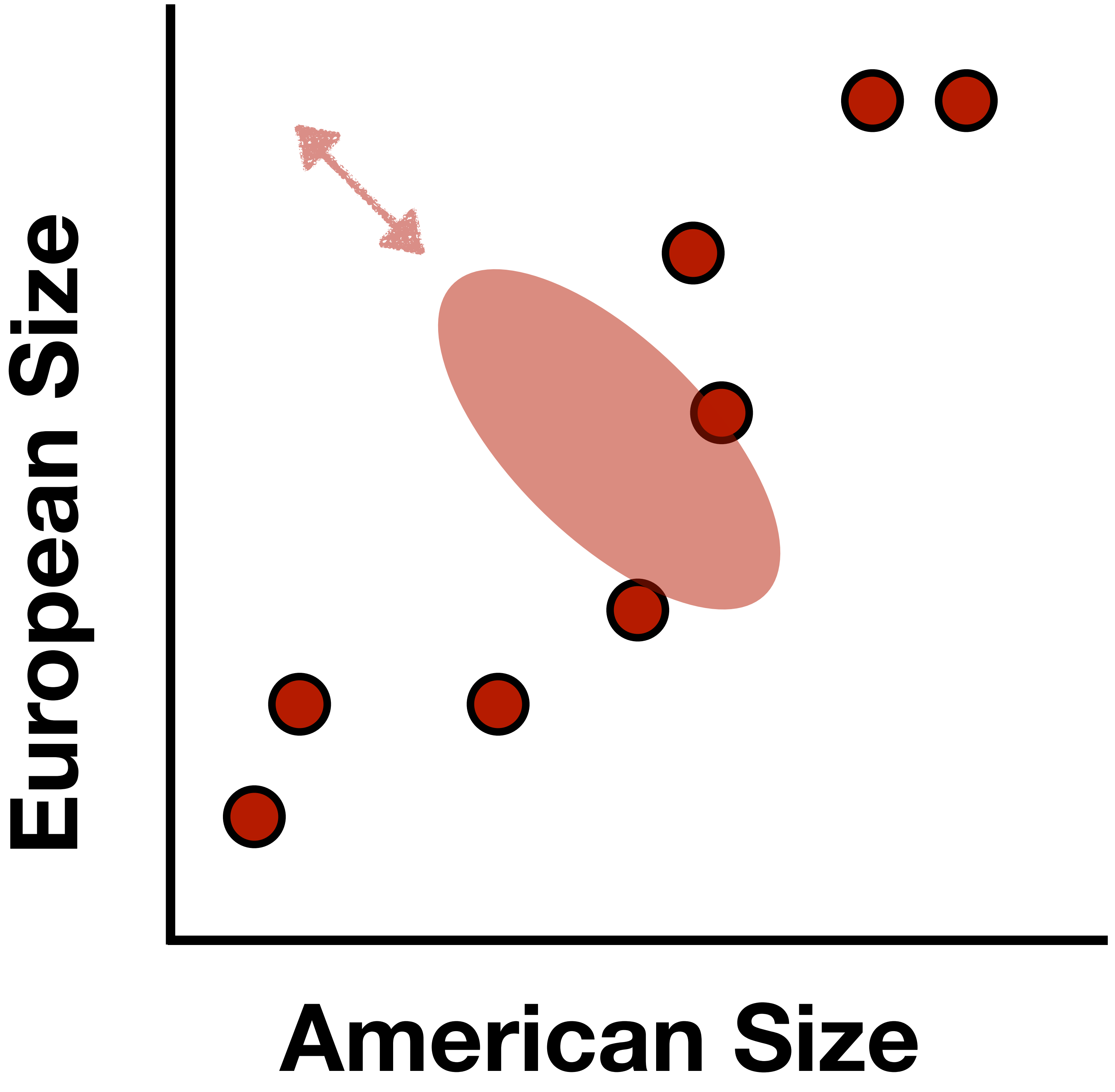




Another Goal: Maximize Variance

To identify patterns we want to study variation across observations

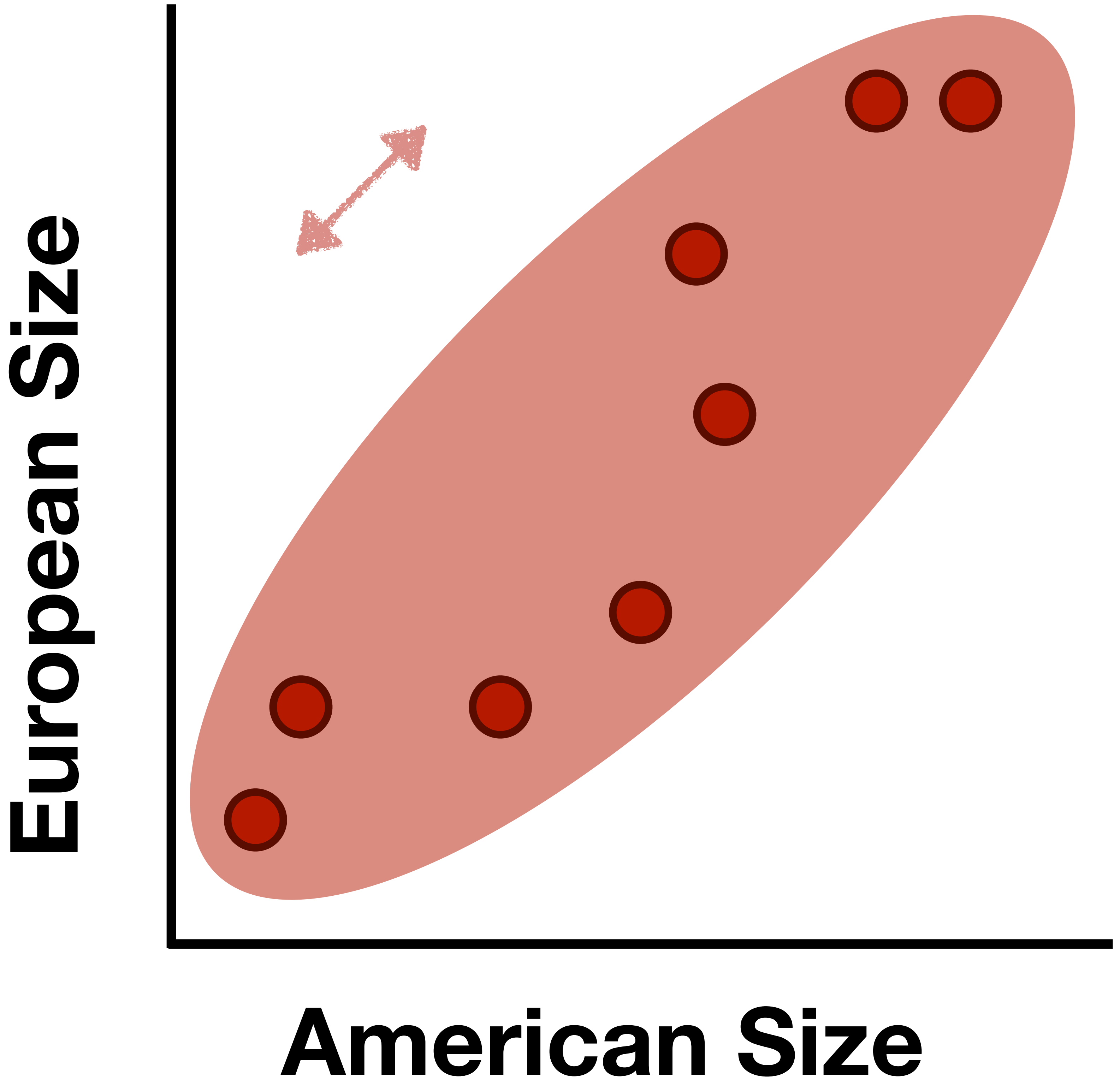
Can we do 'better', i.e., find a compact representation that captures variation?



Another Goal: Maximize Variance

To identify patterns we want to study variation across observations

Can we do ‘better’, i.e., find a compact representation that captures variation?

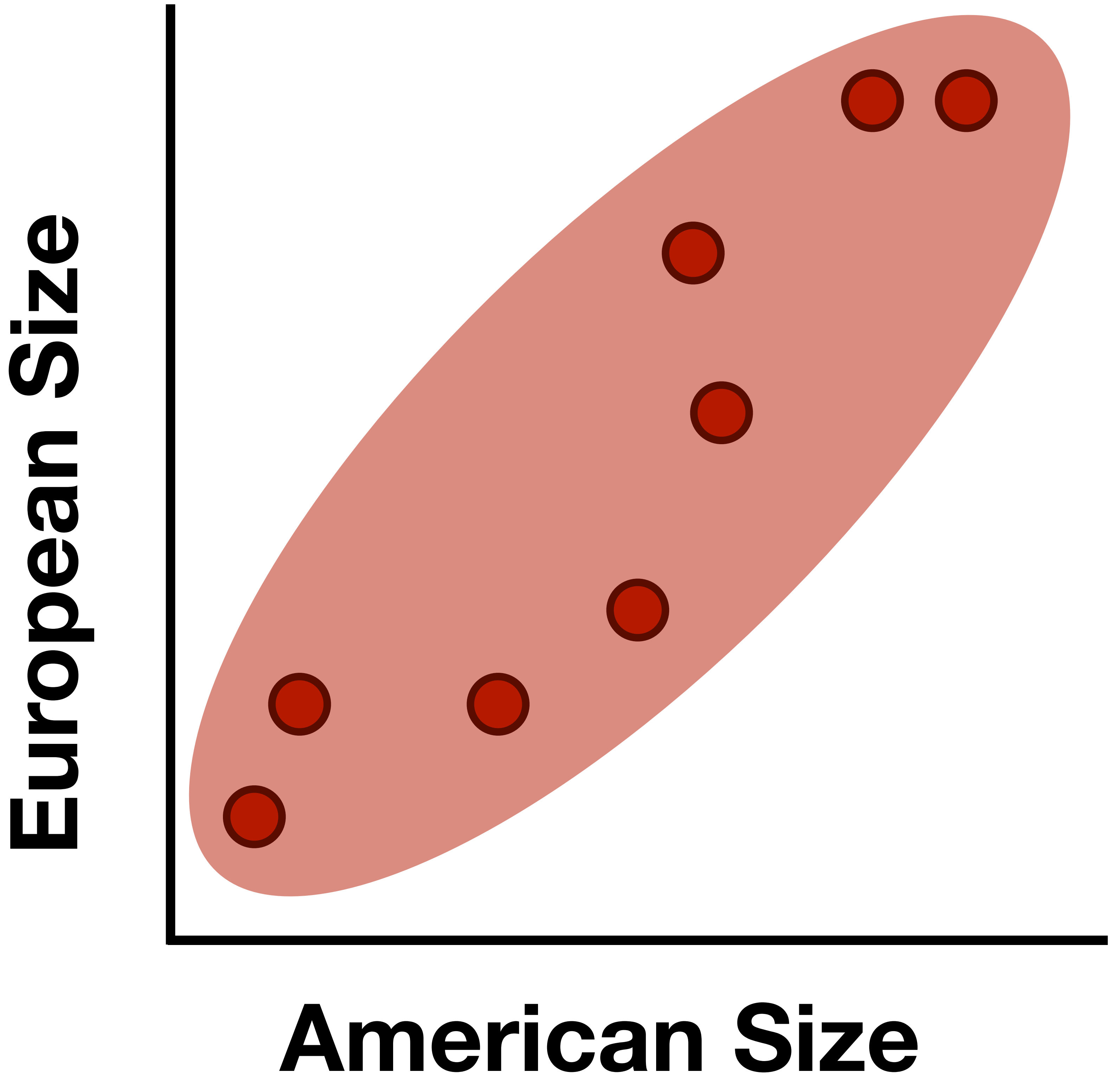


Another Goal: Maximize Variance

To identify patterns we want to study variation across observations

Can we do ‘better’, i.e., find a compact representation that captures variation?

PCA solution finds directions of maximal variance!



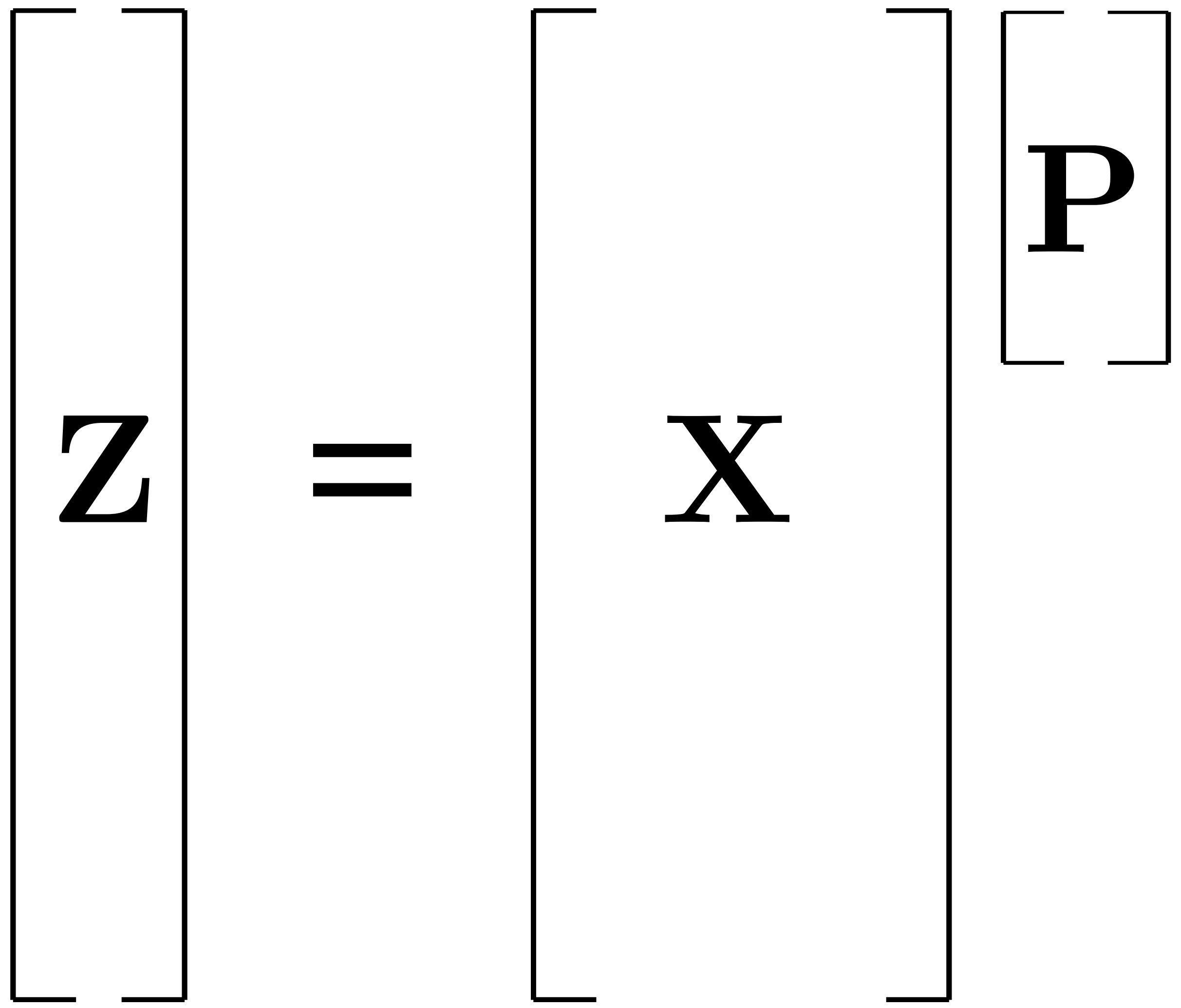
PCA Assumptions and Solution

PCA Formulation

PCA: find lower-dimensional representation of raw data

- $\mathbf{X}$ is $n \times d$ (raw data)
- $\mathbf{Z} = \mathbf{X}\mathbf{P}$ is $n \times k$ (reduced representation, PCA ‘scores’)
- $\mathbf{P}$ is $d \times k$ (columns are k principal components)
- Variance constraints

Linearity assumption ($\mathbf{Z} = \mathbf{X}\mathbf{P}$) simplifies problem



Given n training points with d features:

- $\mathbf{X} \in \mathbb{R}^{n \times d}$: matrix storing points
- $x_j^{(i)}$: jth feature for ith point
- μ_j : mean of jth feature

Variance of 1st feature $$\sigma_1^2 = \frac{1}{n} \sum_{i=1}^{n} \left(x_1^{(i)} - \mu_1 \right)^2$$

Variance of 1st feature
(assuming zero mean) $$\sigma_1^2 = \frac{1}{n} \sum_{i=1}^{n} \left(x_1^{(i)} \right)^2$$

Given n training points with d features:

- $\mathbf{X} \in \mathbb{R}^{n \times d}$: matrix storing points
- $x_j^{(i)}$: *j*th feature for *i*th point
- μ_j : mean of *j*th feature

Covariance of 1st and 2nd features (assuming zero mean) $$\sigma_{12} = \frac{1}{n} \sum_{i=1}^{n} x_1^{(i)} x_2^{(i)}$$

- Symmetric: $\sigma_{12} = \sigma_{21}$
- Zero → uncorrelated
- Large magnitude → (anti) correlated / redundant
- $\sigma_{12} = \sigma_1^2 = \sigma_2^2$ → features are the same

Covariance Matrix

Covariance matrix generalizes this idea for many features

$d \times d$ covariance matrix with zero mean features

$$\mathbf{C_X} = \frac{1}{n}\mathbf{X}^\top\mathbf{X}$$

- ith diagonal entry equals variance of ith feature
- ijth entry is covariance between ith and jth features
- Symmetric (makes sense given definition of covariance)

$$\text{Variance: } \sigma_1^2 = \frac{1}{n}\sum_{i=1}^{n}\left(x_1^{(i)}\right)^2$$

$$\underset{\mathbf{X}^\top}{\begin{bmatrix} 2 & -1 & -1 \\ 3 & 2 & -5 \end{bmatrix}} \underset{\mathbf{X}}{\begin{bmatrix} 2 & 3 \\ -1 & 2 \\ -1 & -5 \end{bmatrix}} = \underset{\mathbf{X}^\top\mathbf{X}}{\begin{bmatrix} 6 & \\ & \end{bmatrix}}$$

Variance: $\sigma_1^2 = \frac{1}{n} \sum_{i=1}^{n} \left(x_1^{(i)} \right)^2$

Covariance: $\sigma_{12} = \frac{1}{n} \sum_{i=1}^{n} x_1^{(i)} x_2^{(i)}$

$$\underset{\mathbf{X}^\top}{\begin{bmatrix} 2 & -1 & -1 \\ 3 & 2 & -5 \end{bmatrix}} \underset{\mathbf{X}}{\begin{bmatrix} 2 & 3 \\ -1 & 2 \\ -1 & -5 \end{bmatrix}} = \underset{\mathbf{X}^\top \mathbf{X}}{\begin{bmatrix} 6 & 9 \\ 9 & 38 \end{bmatrix}}$$

Dividing by n yields covariance matrix

PCA Formulation

PCA: find lower-dimensional representation of raw data

- $\mathbf{X}$ is $n \times d$ (raw data)
- $\mathbf{Z} = \mathbf{XP}$ is $n \times k$ (reduced representation, PCA ‘scores’)
- $\mathbf{P}$ is $d \times k$ (columns are k principal components)
- Variance / Covariance constraints

What constraints make sense in reduced representation?

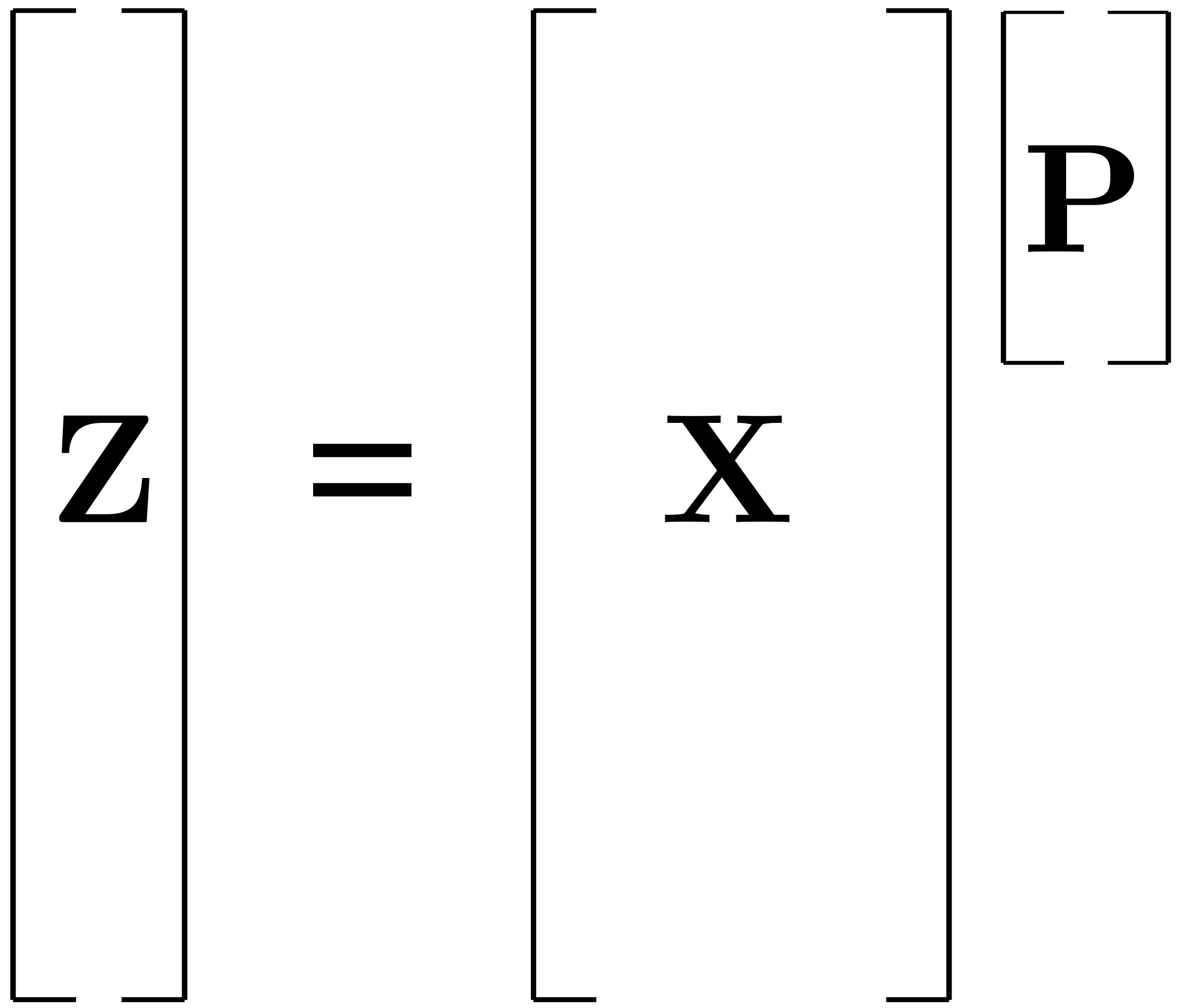
- No feature correlation, i.e., all off-diagonals in $\mathbf{C_Z}$ are zero
- Rank-ordered features by variance, i.e., sorted diagonals of $\mathbf{C_Z}$

PCA Formulation

PCA: find lower-dimensional representation of raw data

- $\mathbf{X}$ is $n \times d$ (raw data)
- $\mathbf{Z} = \mathbf{XP}$ is $n \times k$ (reduced representation, PCA ‘scores’)
- $\mathbf{P}$ is $d \times k$ (columns are k principal components)
- Variance / Covariance constraints

$\mathbf{P}$ equals the top k eigenvectors of $\mathbf{C}_{\mathbf{X}}$



PCA Solution

All covariance matrices have an eigendecomposition

- $\mathbf{C_X} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$(eigendecomposition)
- $\mathbf{U}$ is $d \times d$ (column are eigenvectors, sorted by their eigenvalues)
- $\mathbf{\Lambda}$ is $d \times d$ (diagonals are eigenvalues, off-diagonals are zero)

The d eigenvectors are orthonormal directions of max variance

- Associated eigenvalues equal variance in these directions
- 1st eigenvector is direction of max variance (variance is λ_1)

In lab, we'll use the `eigh` function from `numpy.linalg`

Choosing k

How should we pick the dimension of the new representation?

Visualization: Pick top 2 or 3 dimensions for plotting purposes

Other analyses: Capture 'most' of the variance in the data

- Recall that eigenvalues are variances in the directions specified by eigenvectors, and that eigenvalues are sorted

- Fraction of retained variance: $\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$

Can choose k such that we retain some fraction of the variance, e.g., 95%

Other Practical Tips

PCA assumptions (linearity, orthogonality) not always appropriate

- Various extensions to PCA with different underlying assumptions, e.g., manifold learning, Kernel PCA, ICA

Centering is crucial, i.e., we must preprocess data so that all features have zero mean before applying PCA

PCA results dependent on scaling of data

- Data is sometimes rescaled in practice before applying PCA

PCA Algorithm

Orthogonal and Orthonormal Vectors

Orthogonal vectors are **perpendicular** to each other

- Equivalently, their dot product equals zero
- $\mathbf{a}^\top \mathbf{b} = 0$ and $\mathbf{d}^\top \mathbf{b} = 0$, but $\mathbf{c}$ isn't orthogonal to others

$\mathbf{a} = [1 \quad 0]^\top$ $\mathbf{b} = [0 \quad 1]^\top$ $\mathbf{c} = [1 \quad 1]^\top$ $\mathbf{d} = [2 \quad 0]^\top$

Orthonormal vectors are orthogonal and have unit norm

- $\mathbf{a}$ are $\mathbf{b}$ are orthonormal, but $\mathbf{b}$ are $\mathbf{d}$ are not orthonormal

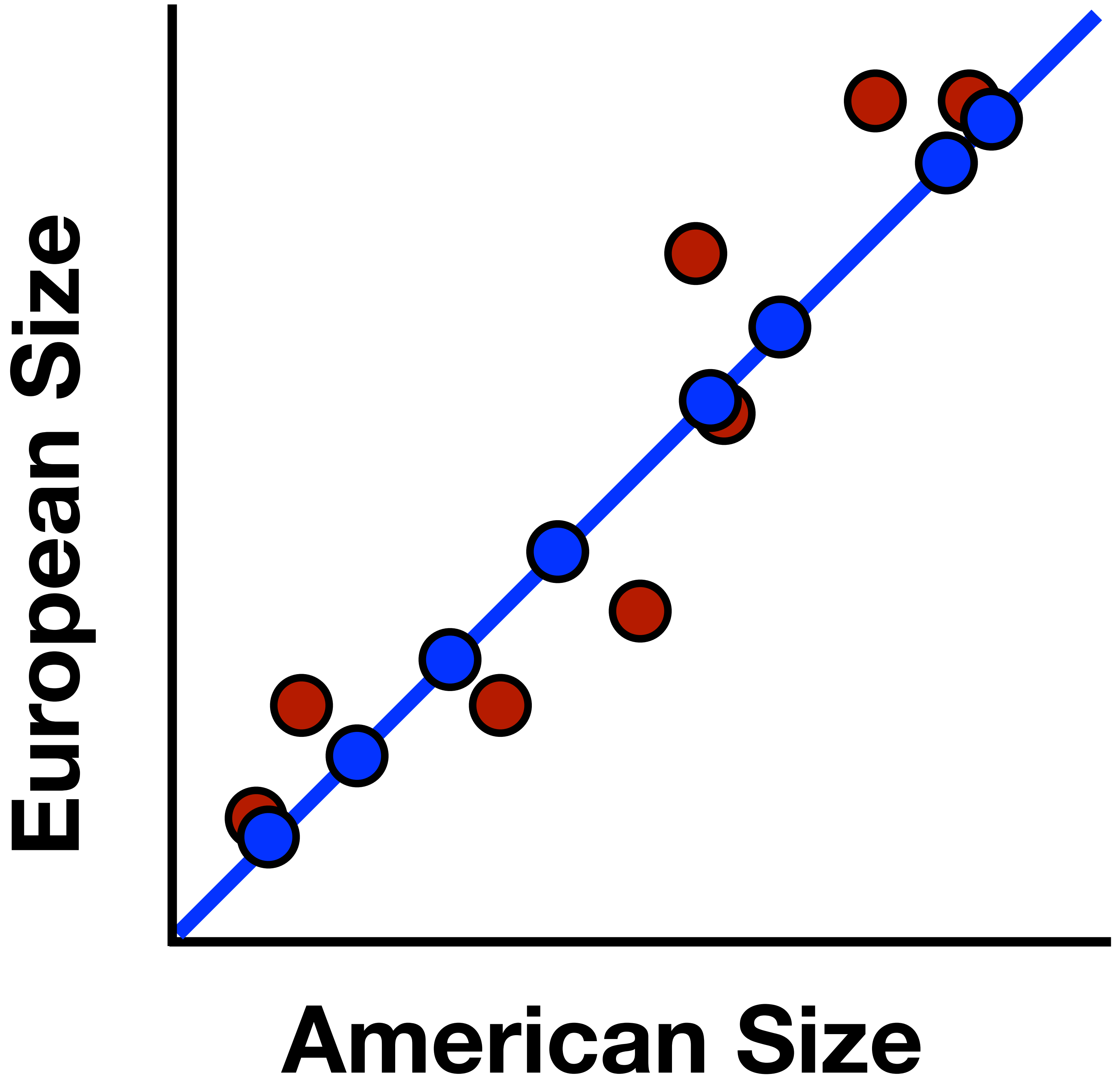
PCA Iterative Algorithm

$k = 1$: Find direction of max variance, project onto this direction

- Locations along this direction are the new 1D representation

More generally, for i in $\{1, \ldots, k\}$:

- Find direction of max variance that is *orthonormal* to previously selected directions, project onto this direction
- Locations along this direction are the ith feature in new representation


European Size
American Size