Christian Machens
Champalimaud Centre for the Unknown

Dimensionality reduction:
PCA and friends

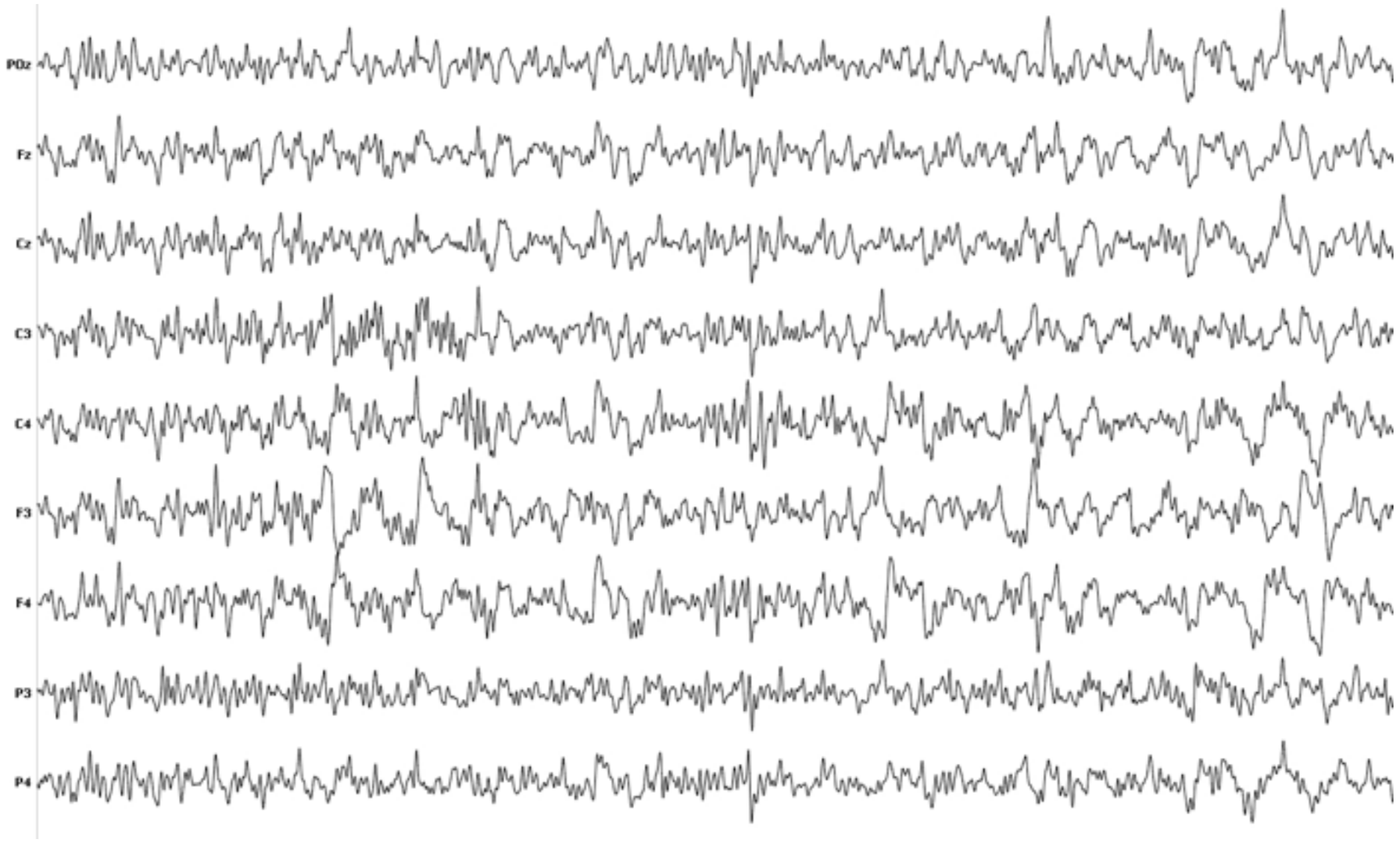
Champalimaud Foundation

What is dimensionality reduction and why do we need it?

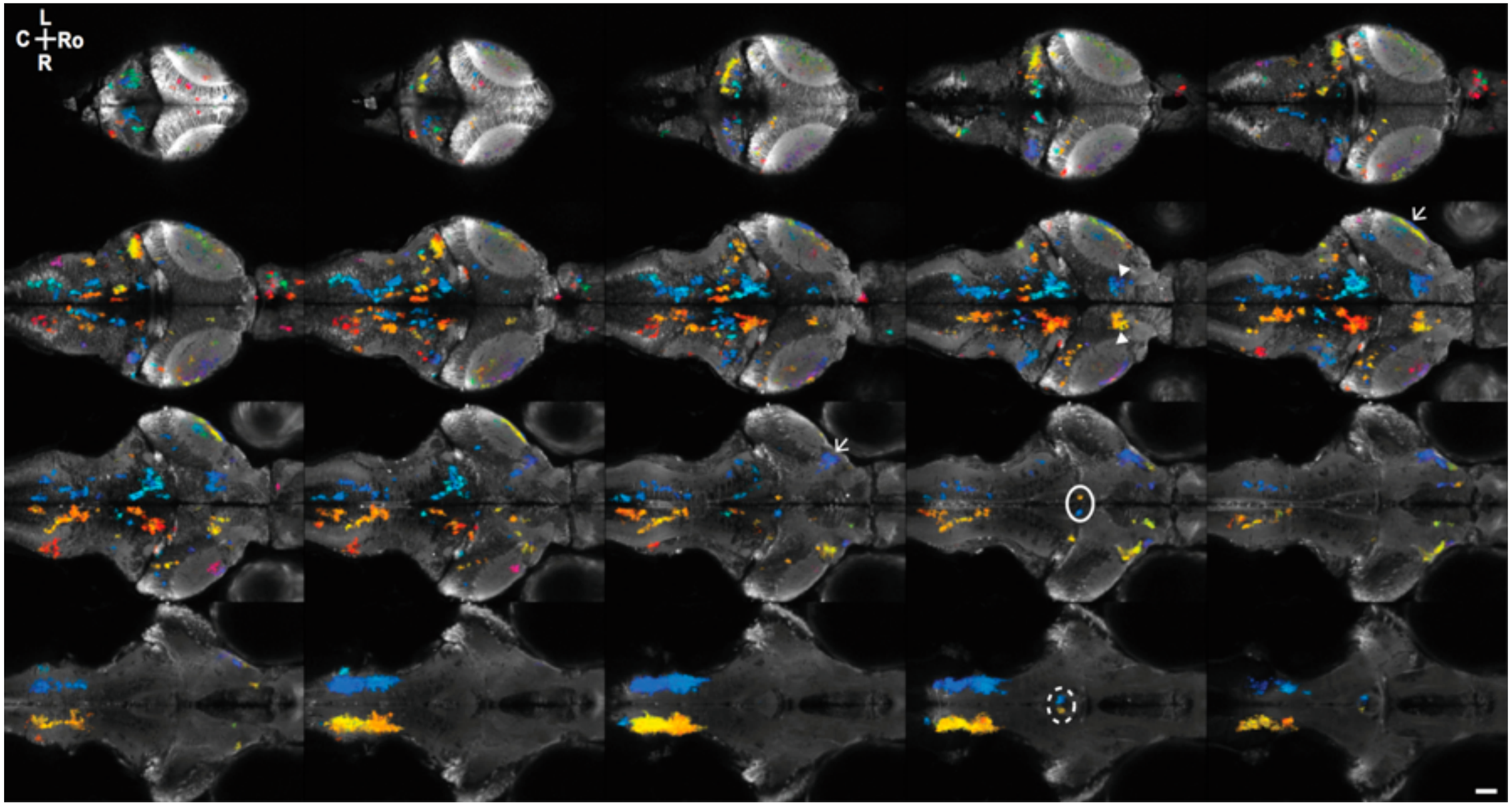
Reduce the data but keep the essential features

- Saves time and storage on computer
- Allows visualization
- Can be used as input to other methods

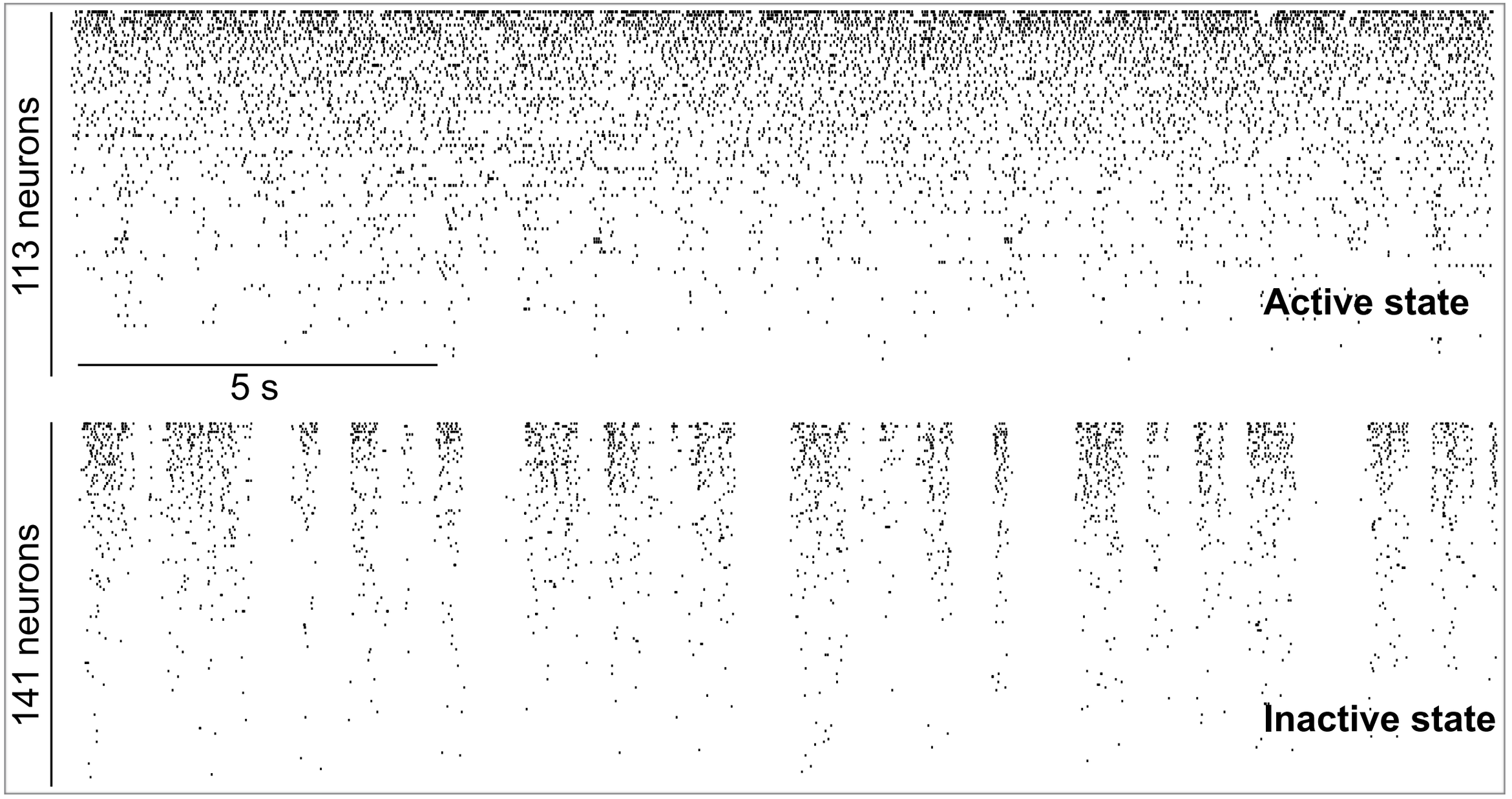
Example 1: Electroencephalography (EEG)



Example 2:
Whole-brain imaging in zebrafish



Example 3:
Large-scale Electrophysiology



Population Data

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots \\ x_{21} & \dots & & \\ \vdots & & \ddots & \\ x_{N1} & \dots & & x_{NT} \end{bmatrix}$$

neuron 1

neuron 2

neuron N

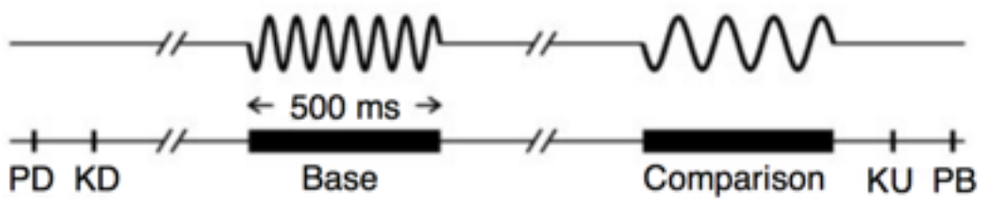
time 1 time 2 time 3 time T

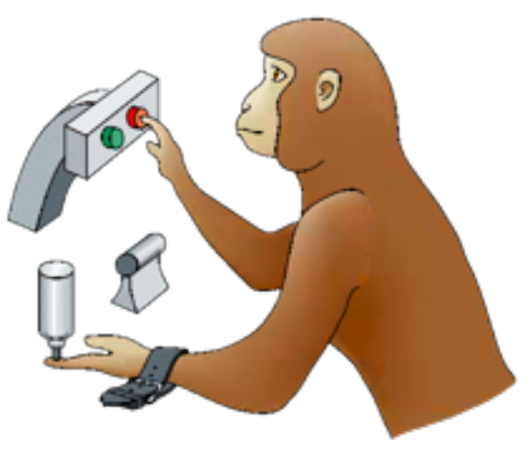
Population Data

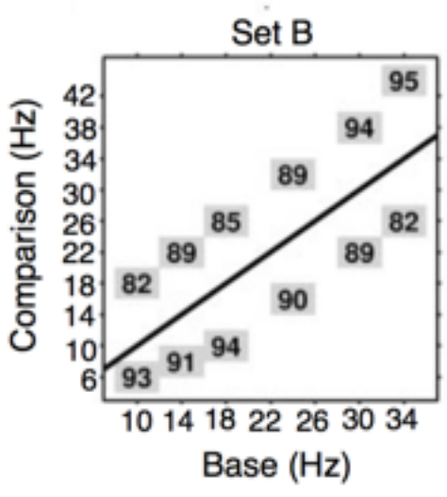
$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots \\ x_{21} & \dots & & \\ \vdots & & \ddots & \\ x_{N1} & \dots & & x_{NT} \end{bmatrix} \quad \begin{matrix} \text{neuron 1} \\ \text{neuron 2} \\ \\ \\ \text{neuron N} \end{matrix}$$

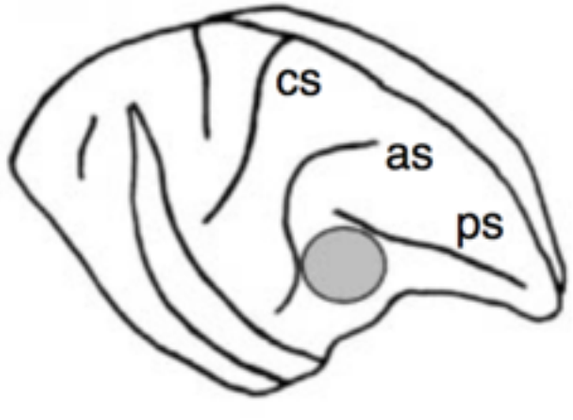
$$\underbrace{\qquad\qquad}_{\text{condition 1}} \quad \underbrace{\qquad\qquad}_{\text{condition 2}} \quad \dots$$

Our example for today:
Pooled population data



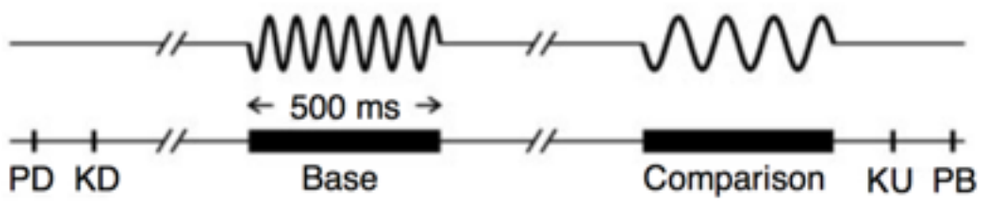


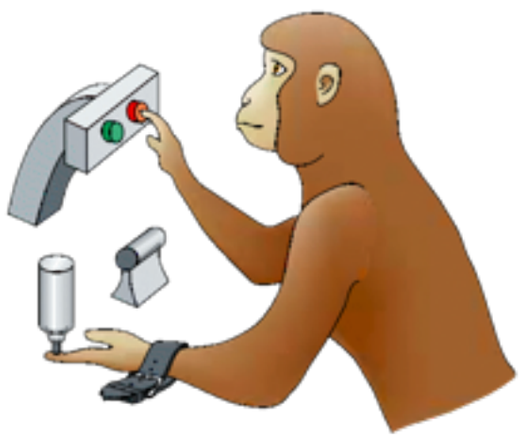


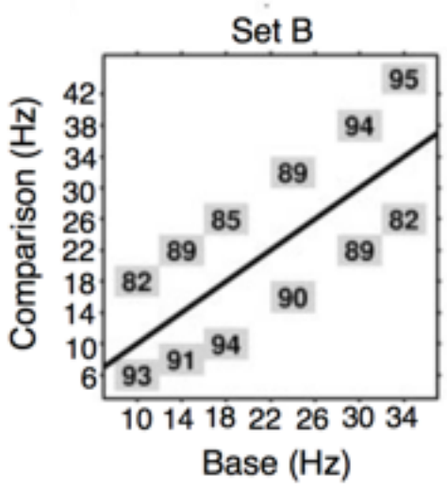


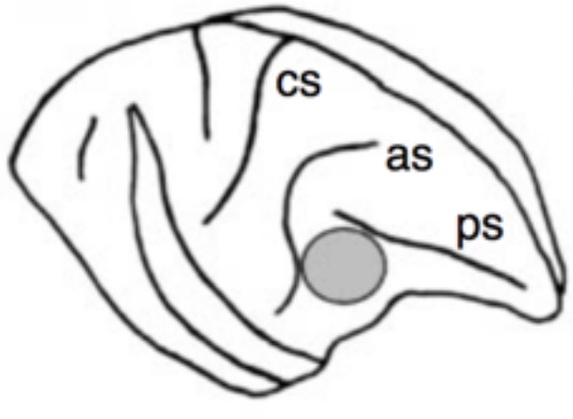
Romo et al, 1999, Nature

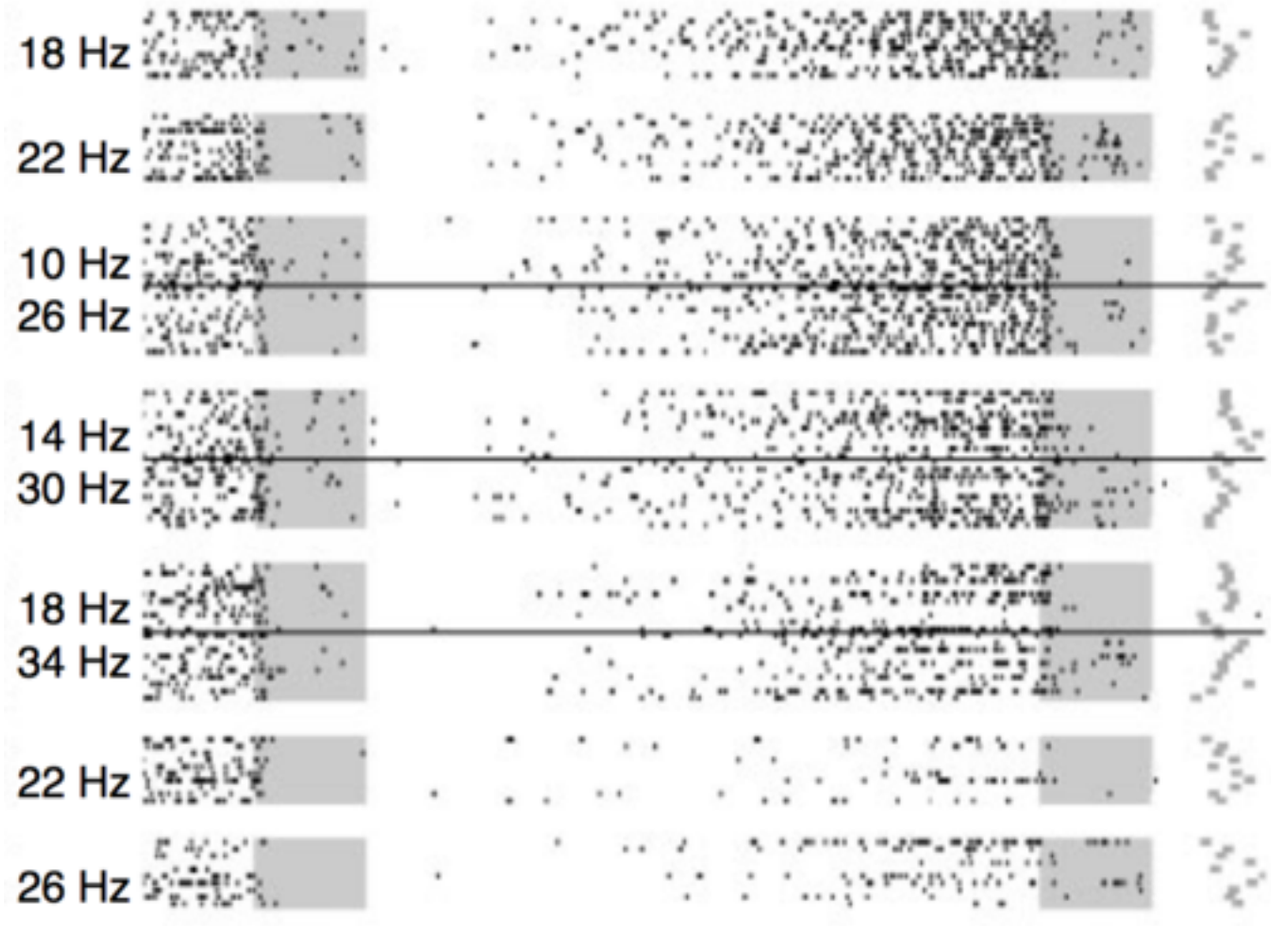
Our example for today: Pooled population data





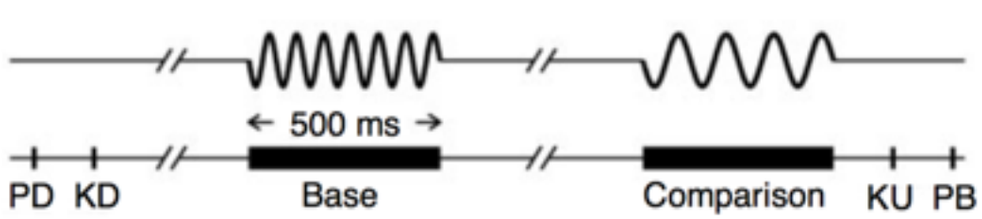


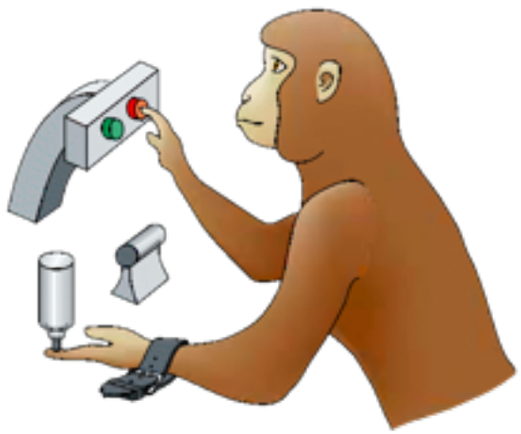


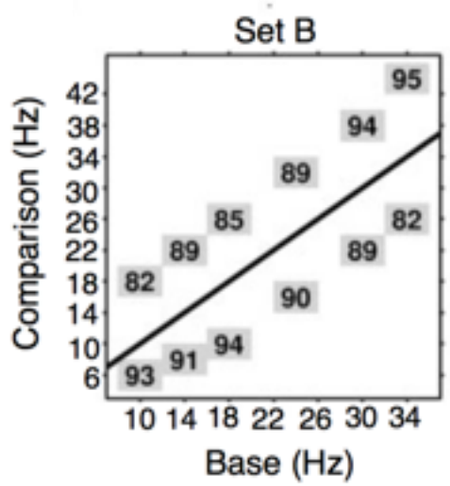


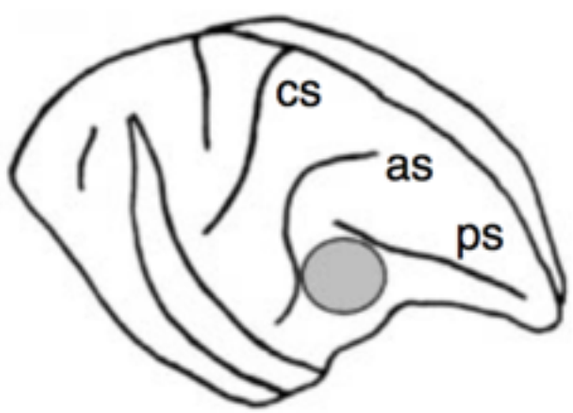
Romo et al, 1999, Nature

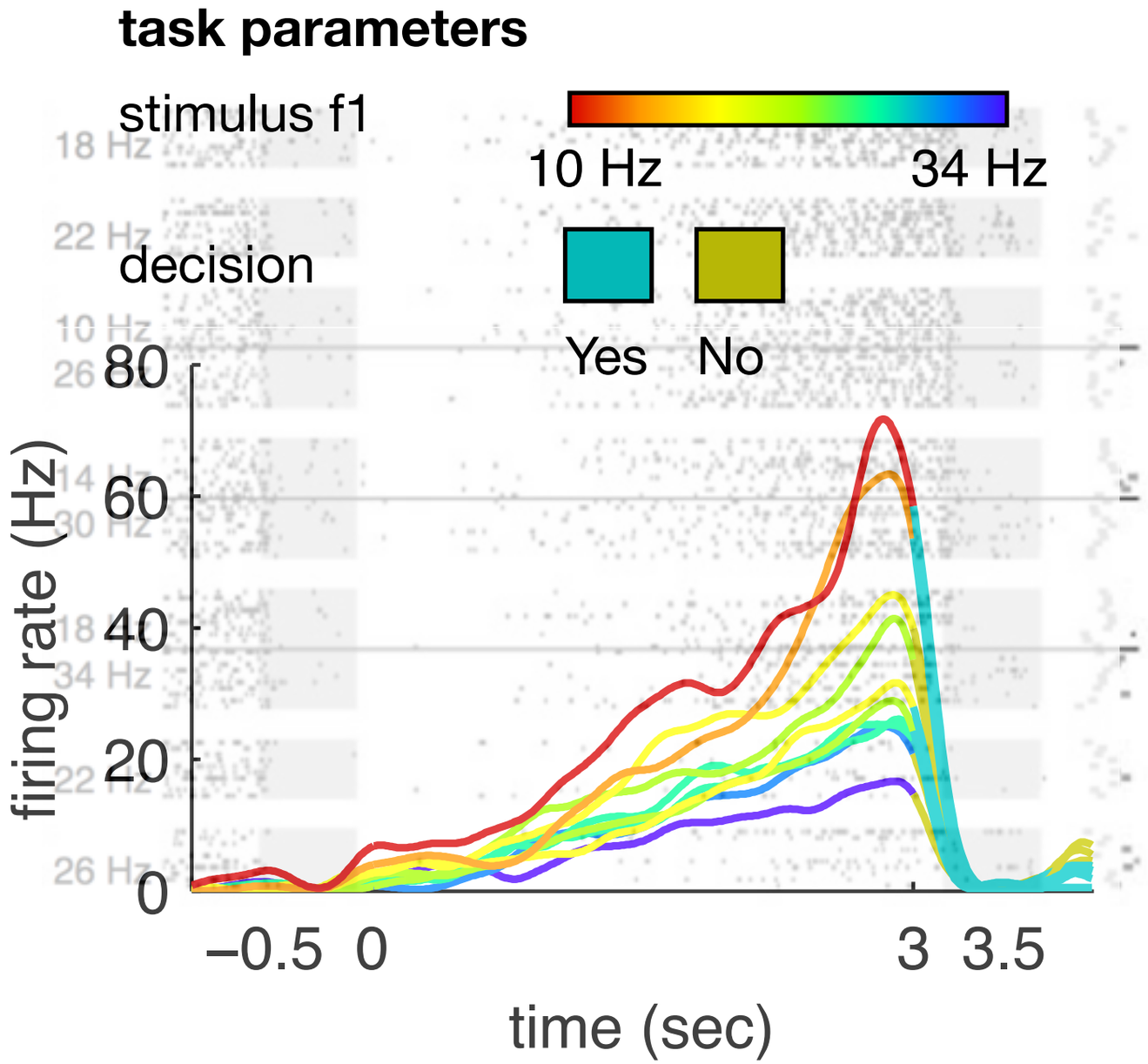
Our example for today:
Pooled population data







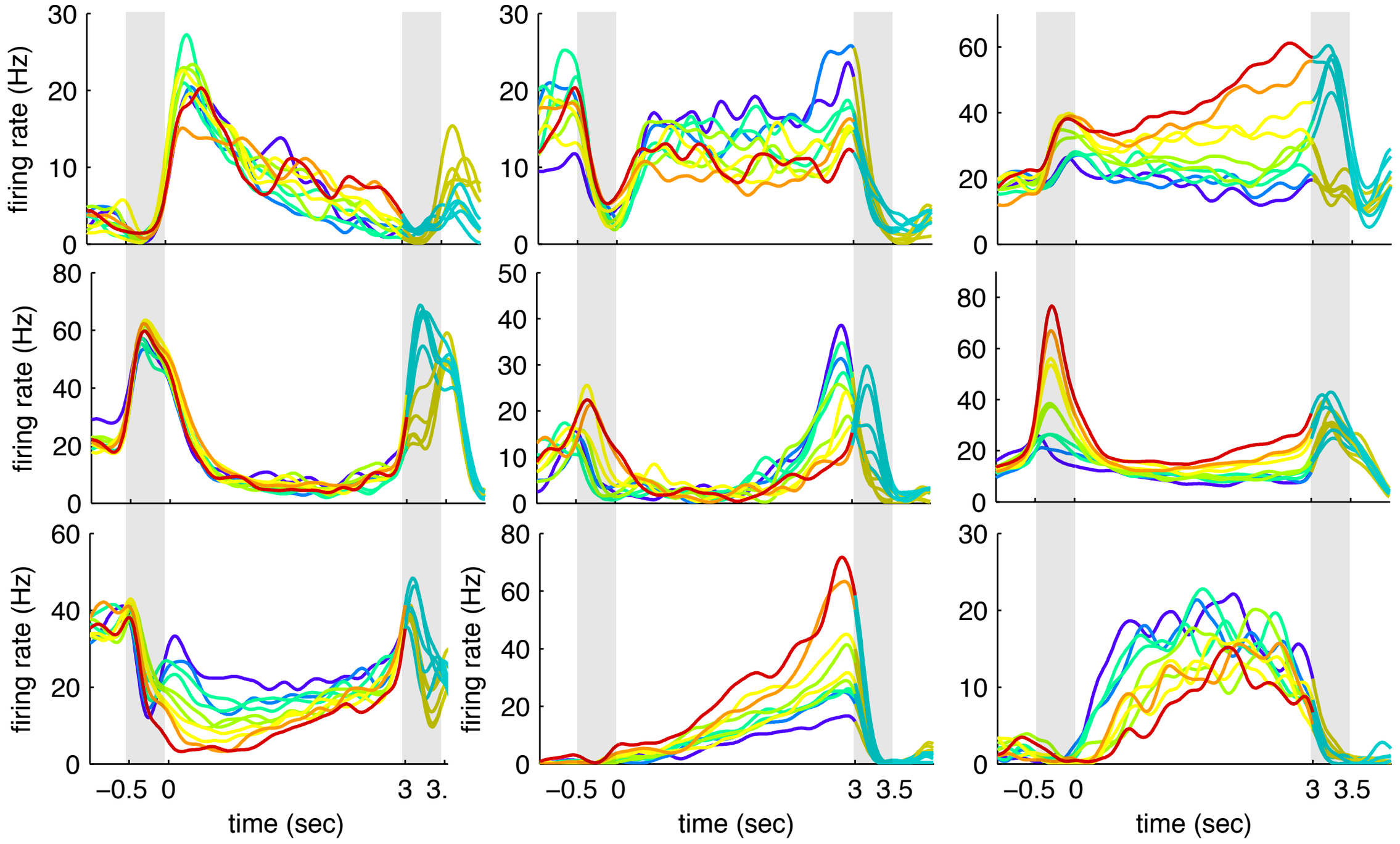




Romo et al, 1999, Nature

Pooled population data
‘Mixed selectivity’ in single cells

PFC example neurons (Romo et al, 1999)



Outline:

1. Exercise: Get the ‘data’ into the right format!
    1. Spike raster
    2. Peristimulus time histogram (PSTH)
    3. Generate Data matrix with PSTHs for all cells

2. Lecture + Exercise: PCA on all cells

3. Lecture: demixed PCA

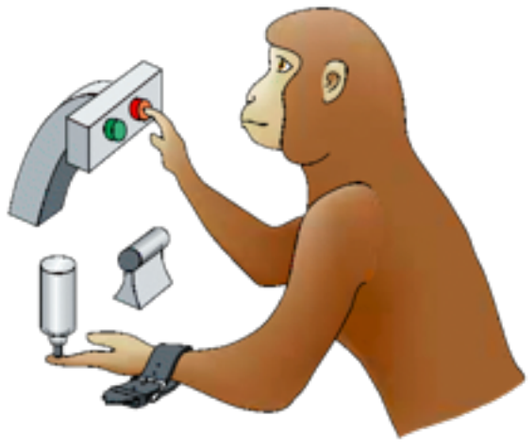
Outline:

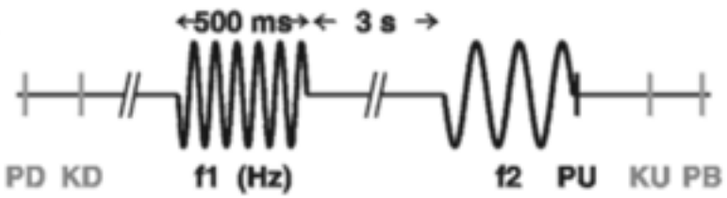
1. Exercise: Get the ‘data’ into the right format!
    1. Spike raster
    2. Peristimulus time histogram (PSTH)
    3. Generate Data matrix with PSTHs for all cells

2. Lecture + Exercise: PCA on all cells

3. Lecture: demixed PCA

Spike Raster of PFC neurons





f1>f2?

Data:

trial 1 trial 2

Recordings over trials:

Stimulus f1 **f1** = (10 18 ...)

Stimulus f2 **f2** = (18 26 ...)

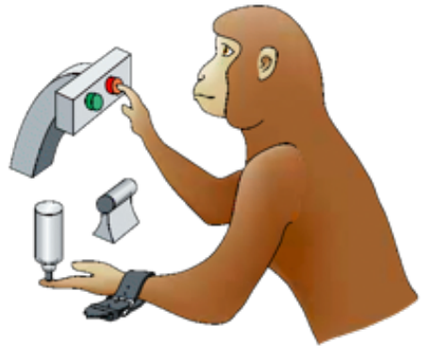
Decisions: **d** = (1 0 ...)

Spike times #1 **s1** = (12 28 53 104 112 256 234 ... 15 29 508 223 145 ...)

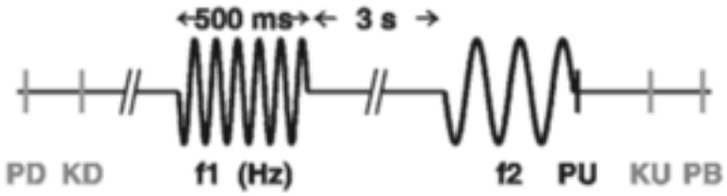
Spike times #2 **s2** = (44 80 178 245 335 399 400 ... 66 94 100 174 205 ...)

...

Spike Raster / PSTH of PFC neurons



f1>f2?



To-do-list

(1) load a file from romo folder
(2) extract all spike times of one neuron
(3) make a raster plot for that neuron
(4) sort spike times according to trial type
(5) compute peri-stimulus time histograms (PSTH) for each trial type
(6) [Advanced:] PSTH standard deviation
(7) loop over all electrodes
(8) loop over all files

MATLAB commands

(1)
(2) cell arrays
(3) ind2sub; sub2ind
(4) unique;
(5)

(6)
(7)
(8)

data courtesy of Ranulfo Romo (UNAM)

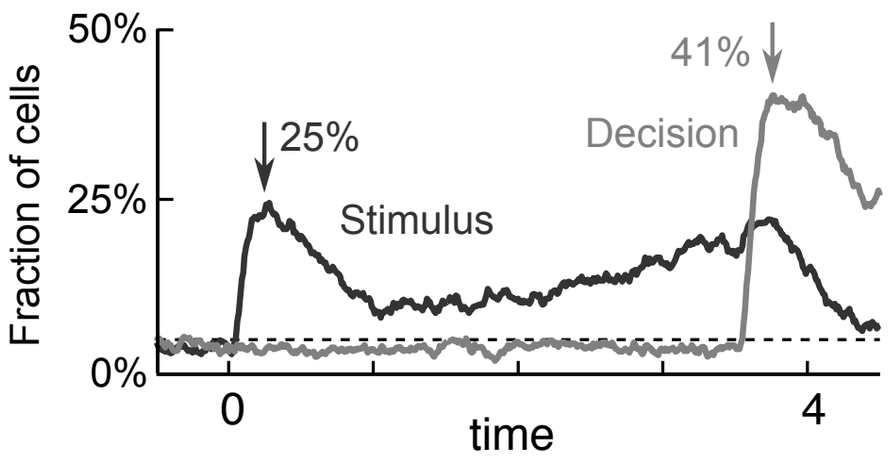
Outline:

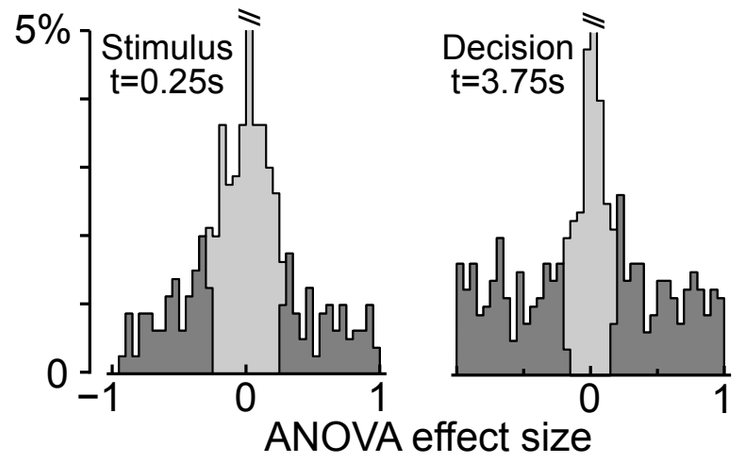
1. Exercise: Get the ‘data’ into the right format!
    1. Spike raster
    2. Peristimulus time histogram (PSTH)
    3. Generate Data matrix with PSTHs for all cells

2. Lecture + Exercise: PCA on all cells

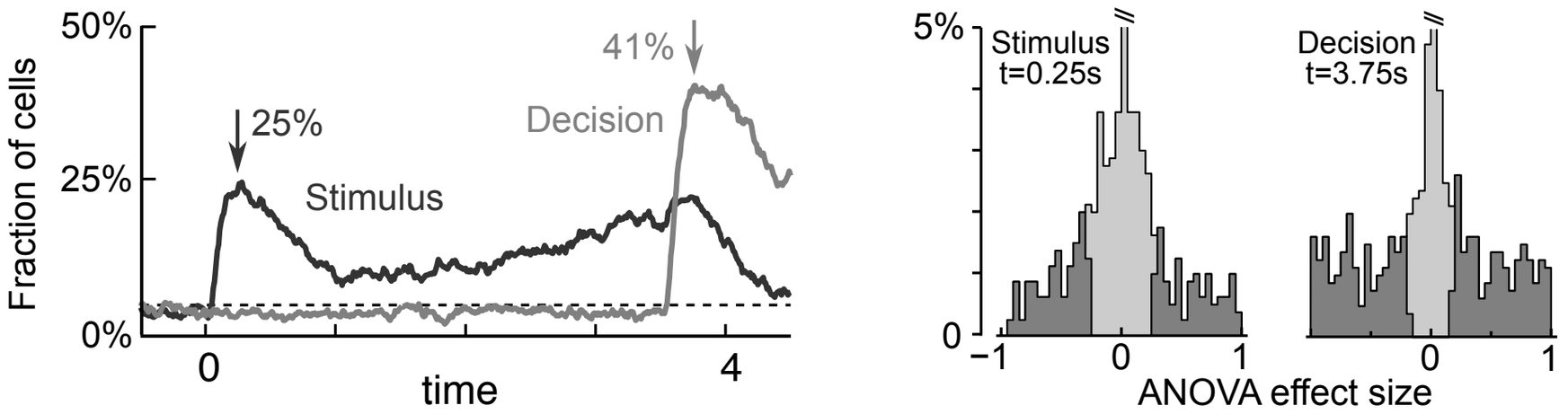
3. Lecture: demixed PCA

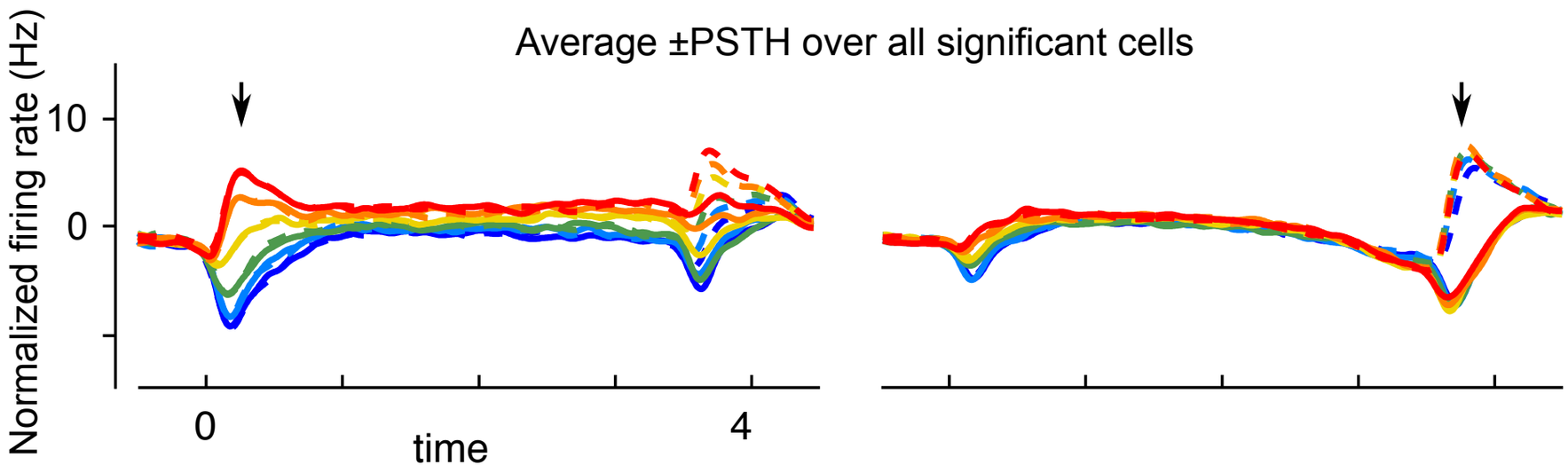
How to summarise the population?
Classical Approaches



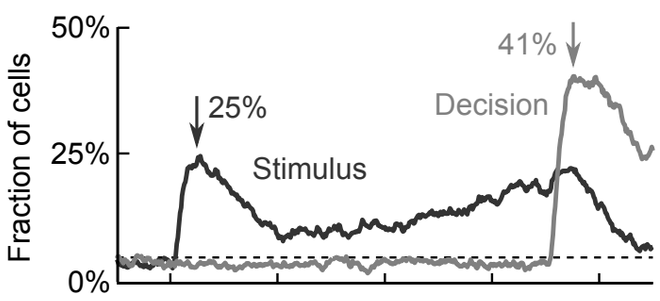


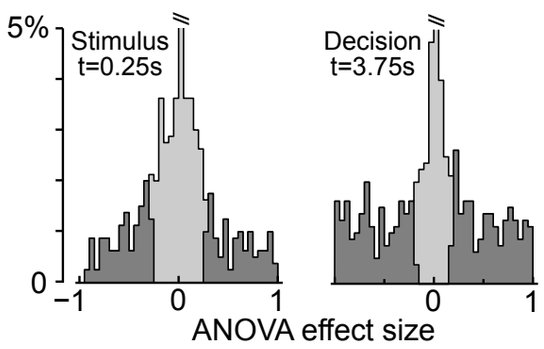
How to summarise the population? Classical Approaches

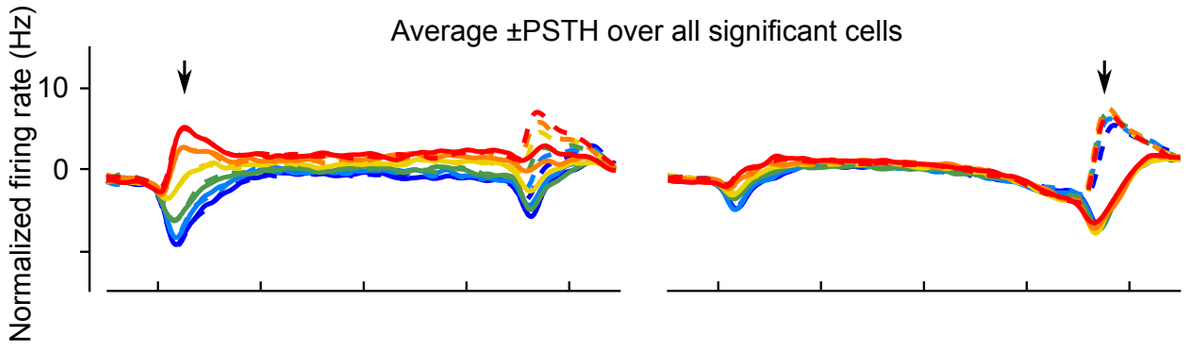




Population average over significantly tuned cells







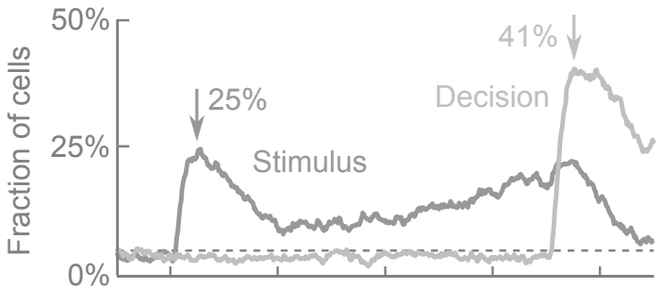
Linear ‘readout’: $x(t, s, d) = \sum_{i=1}^{N} d_i r_i(t, s, d)$

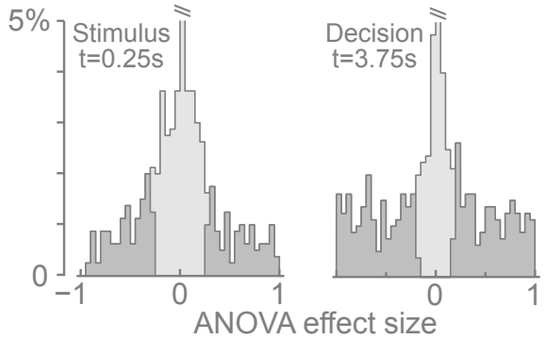
significant neurons: $d_i = 1/N_s$

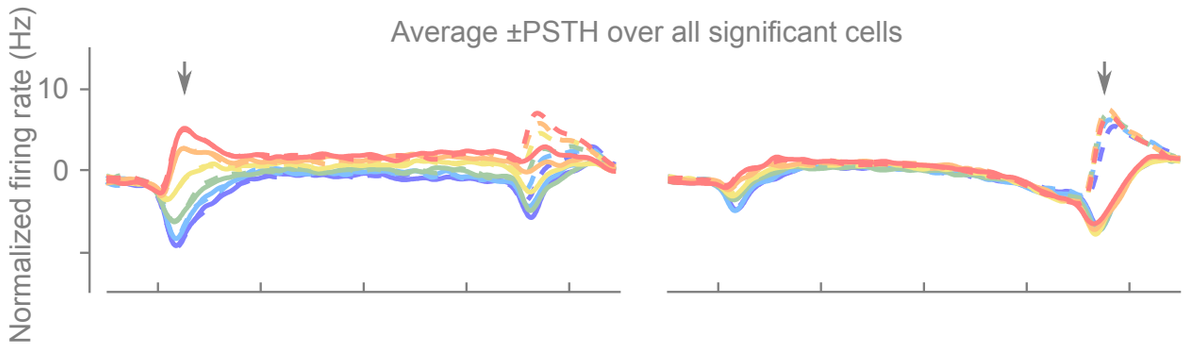
insignificant neurons: $d_i = 0$

Linear Regression over task parameters

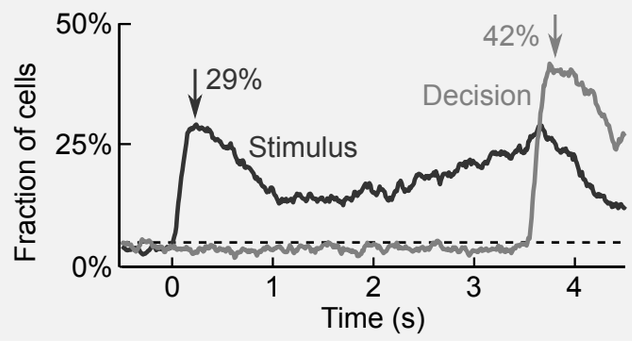
ANOVA

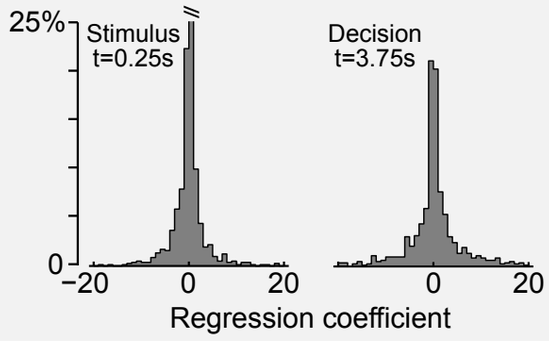


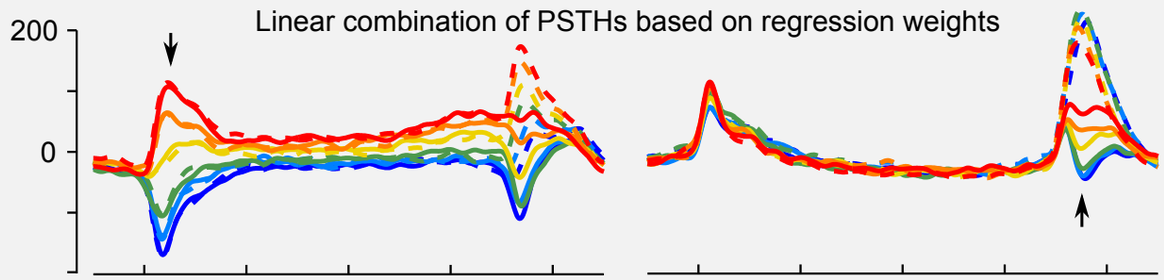




Regression







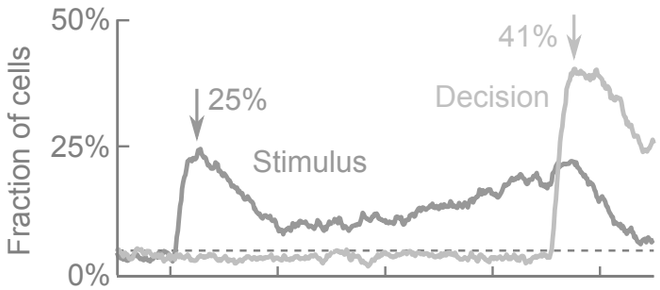
e.g. Brody et al (2003), Mante et al (2013)

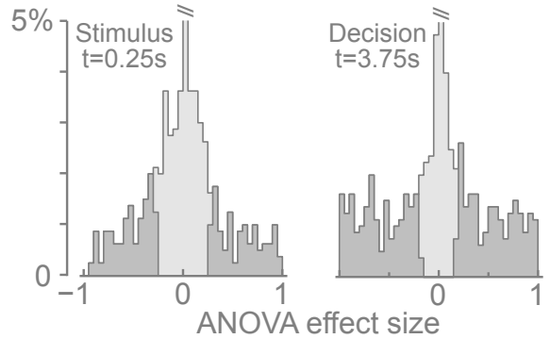
Linear ‘readout’: $$x(t, s, d) = \sum_{i=1}^{N} d_i r_i(t, s, d)$$

regression coefficients
(readout weights)

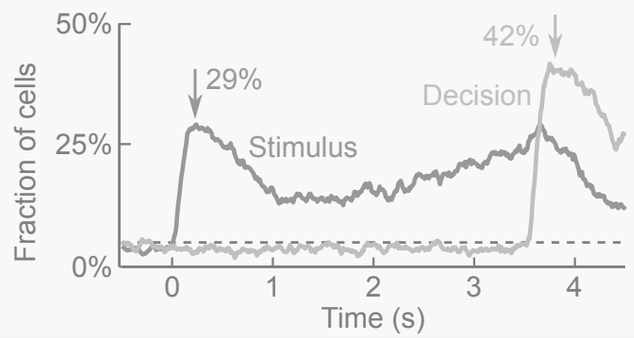
Principal Component Analysis

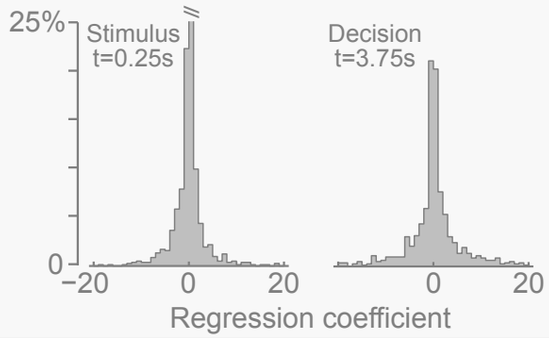
ANOVA

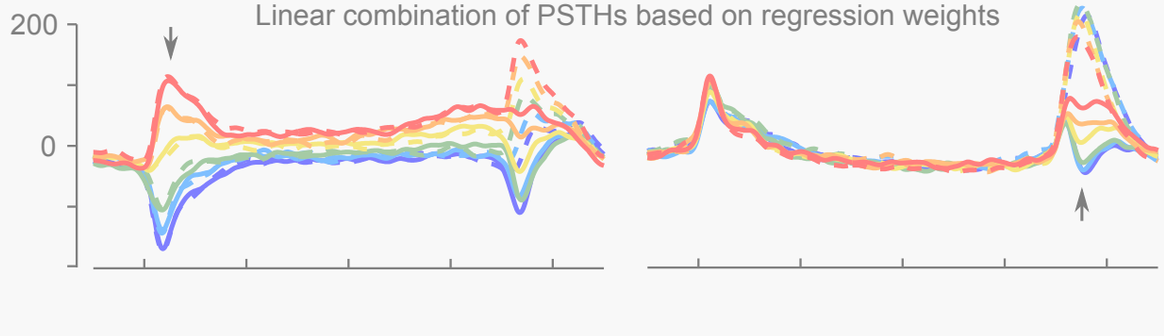




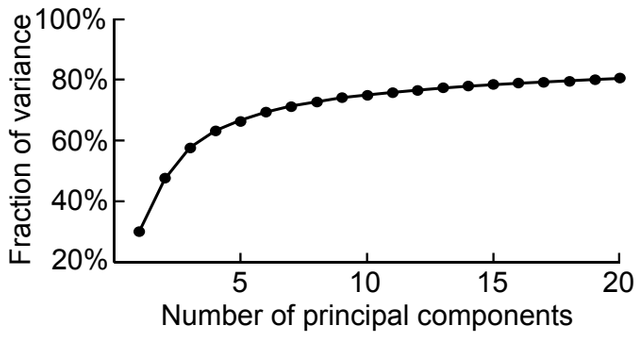
Regression

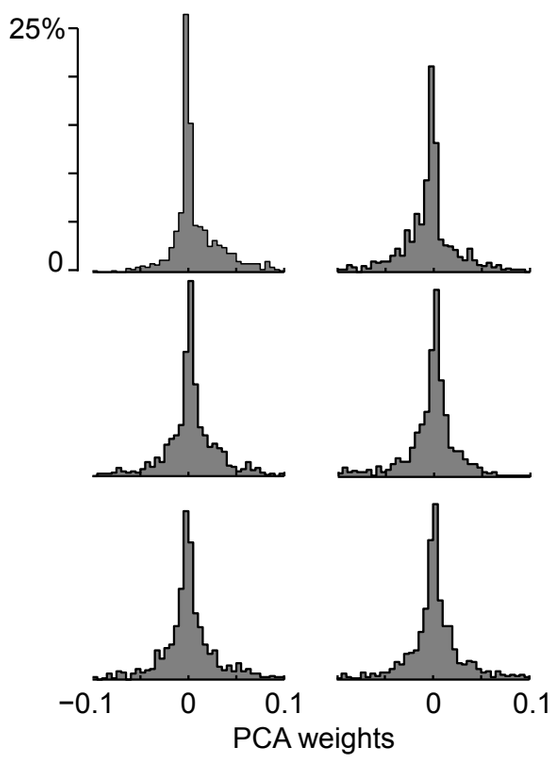


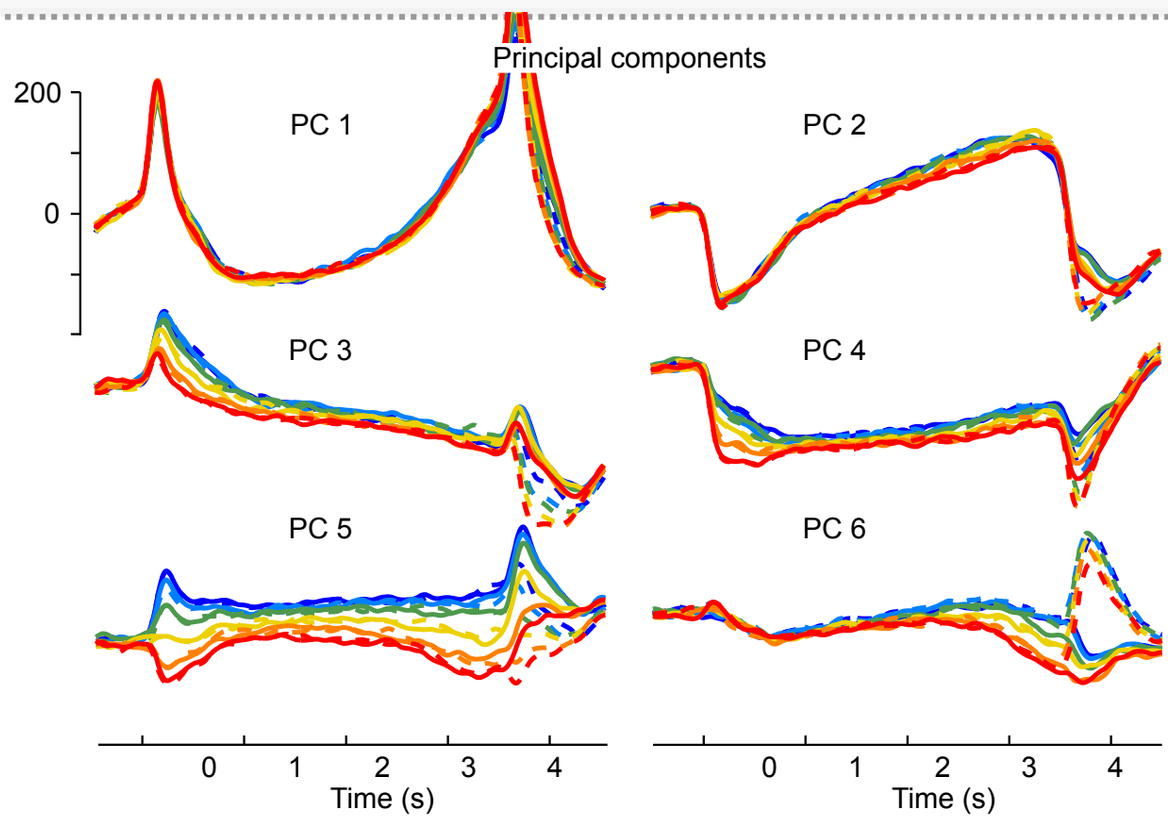




Principal Component Analysis







Common dimensionality reduction methods

- **Principal Component Analysis**
  Classical method of linear dimensionality reduction

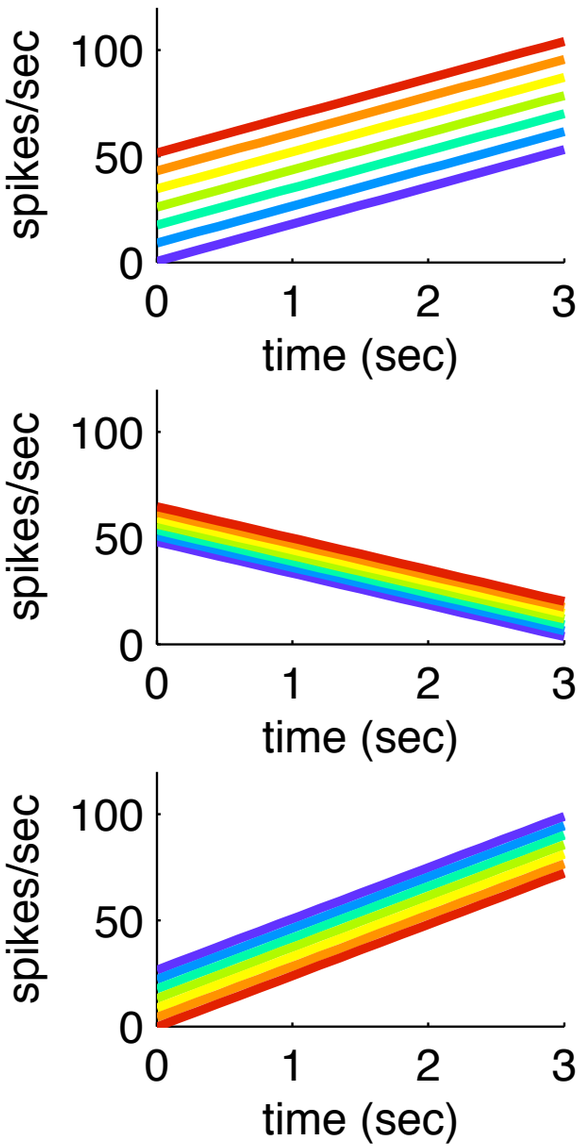
- **Factor Analysis**
  Similar to PCA, but includes a noise model

- **Latent Dynamical Systems**
  Similar to PCA, but includes a noise model and a *dynamical* model!

- **Nonlinear Methods (LLE / Isomap)**
  Cool, but you need to know what you are doing! Careful - they do not deal well with noisy data

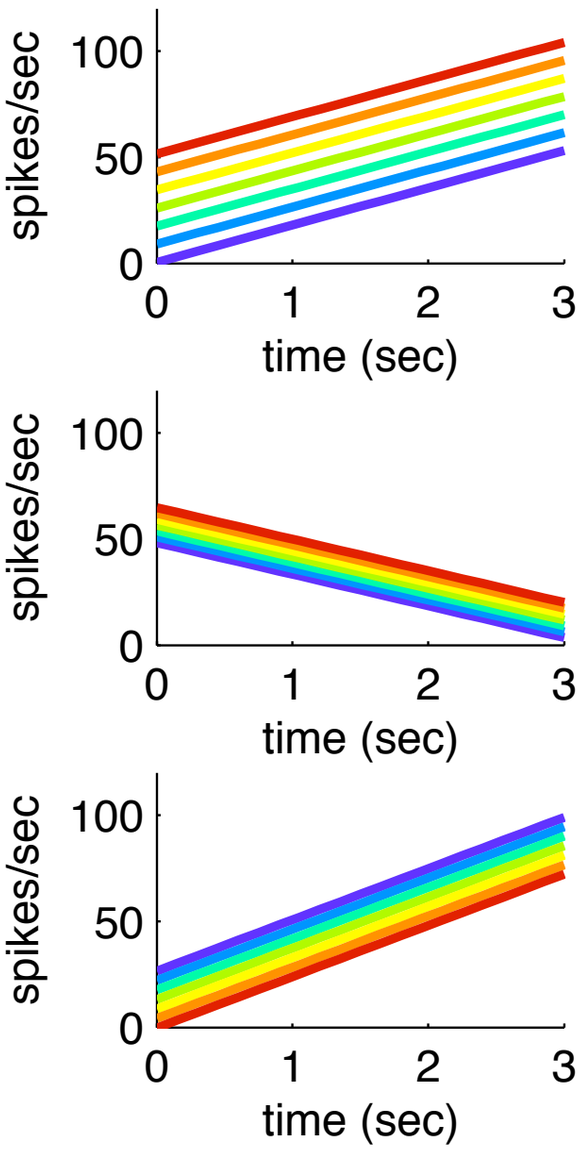
The intuition behind dimensionality reduction

simulated neural data

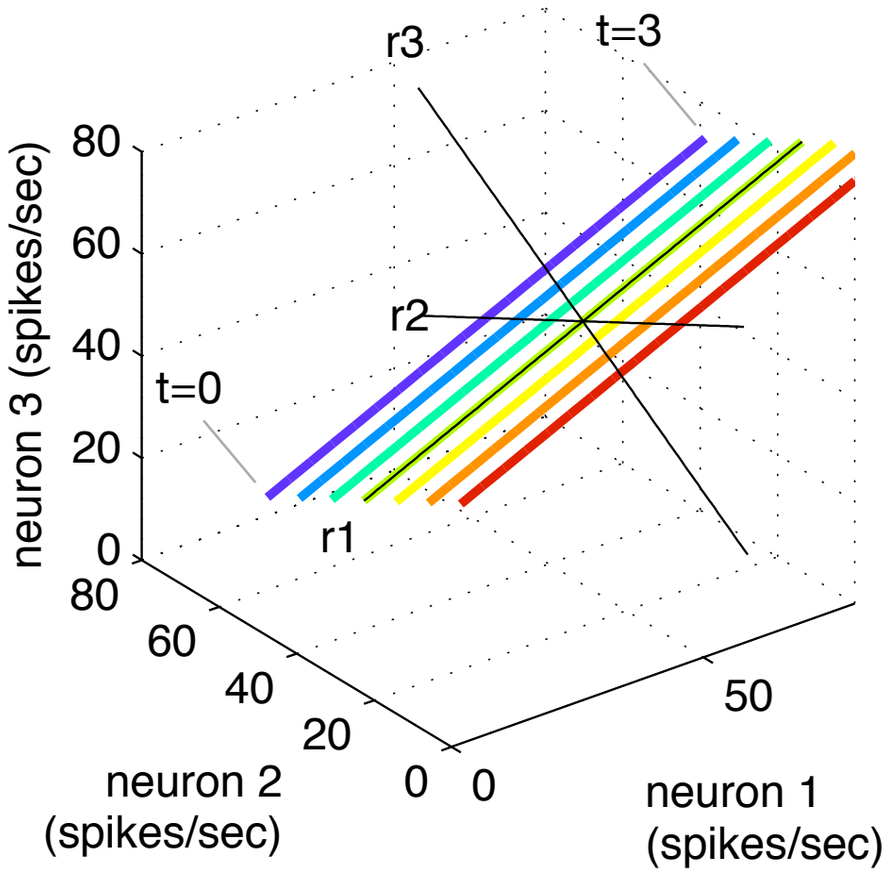


The intuition behind dimensionality reduction

simulated neural data

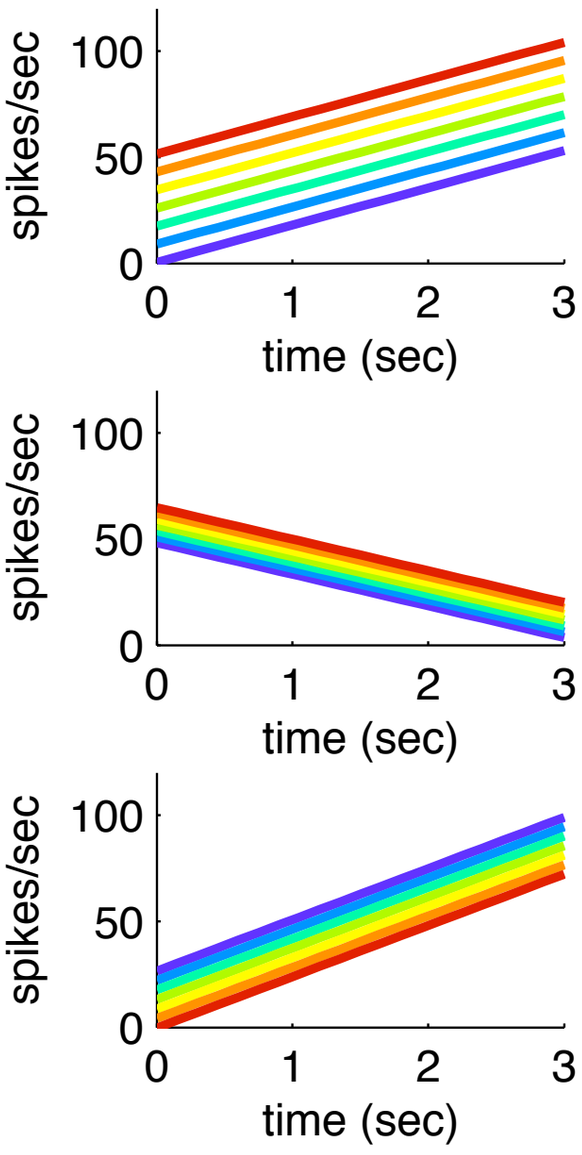


state space embedding

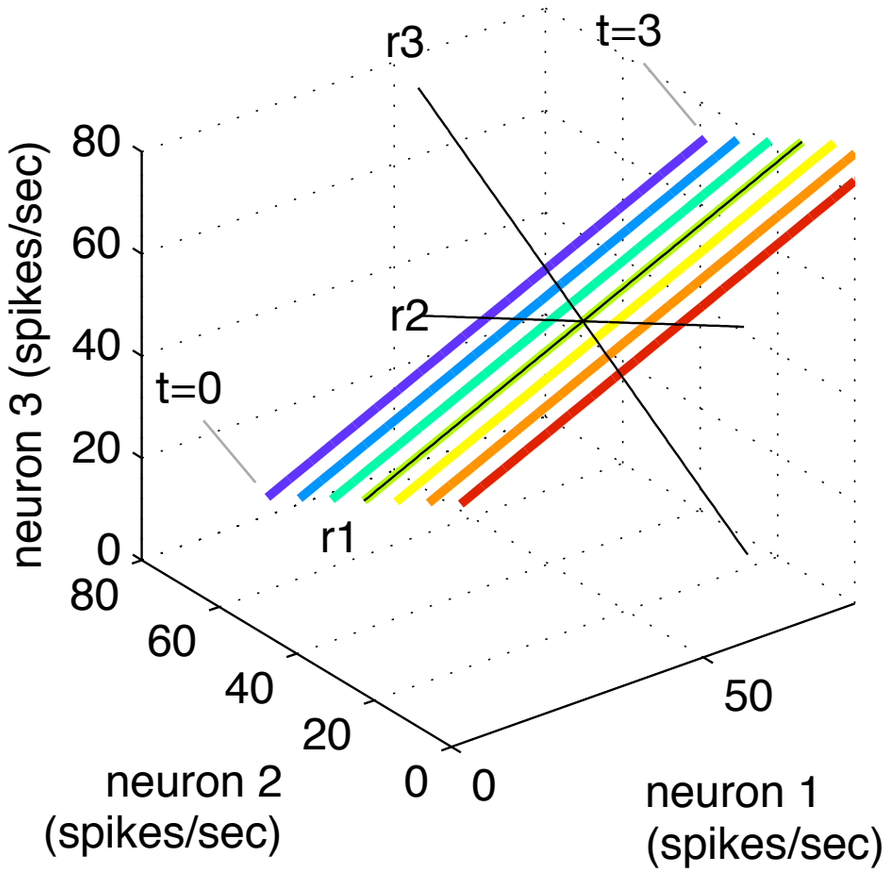


The intuition behind dimensionality reduction

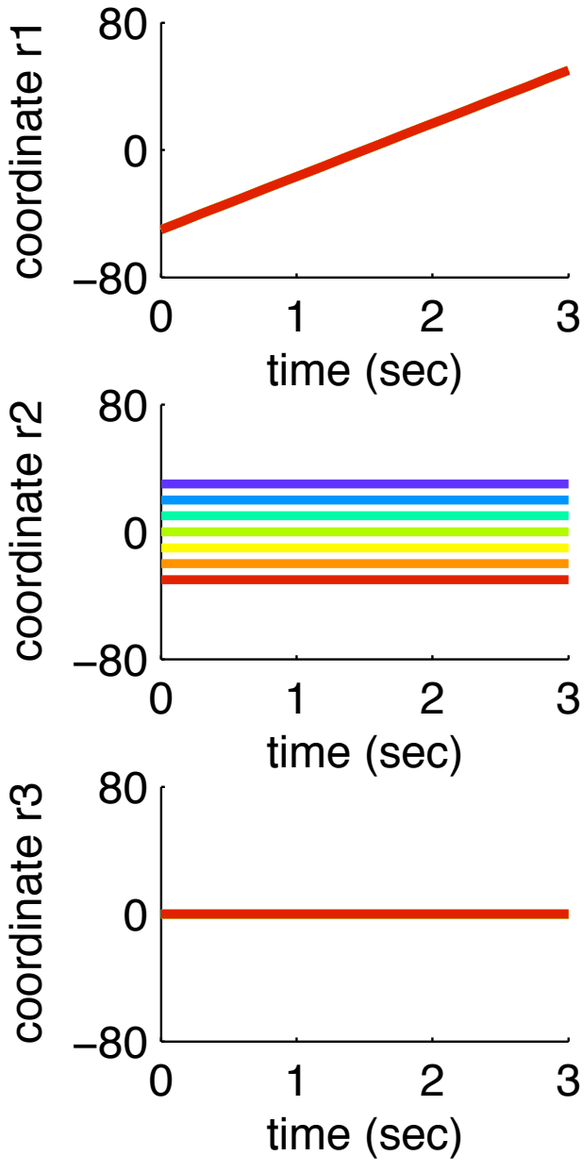
simulated neural data



state space embedding



data in new coordinates

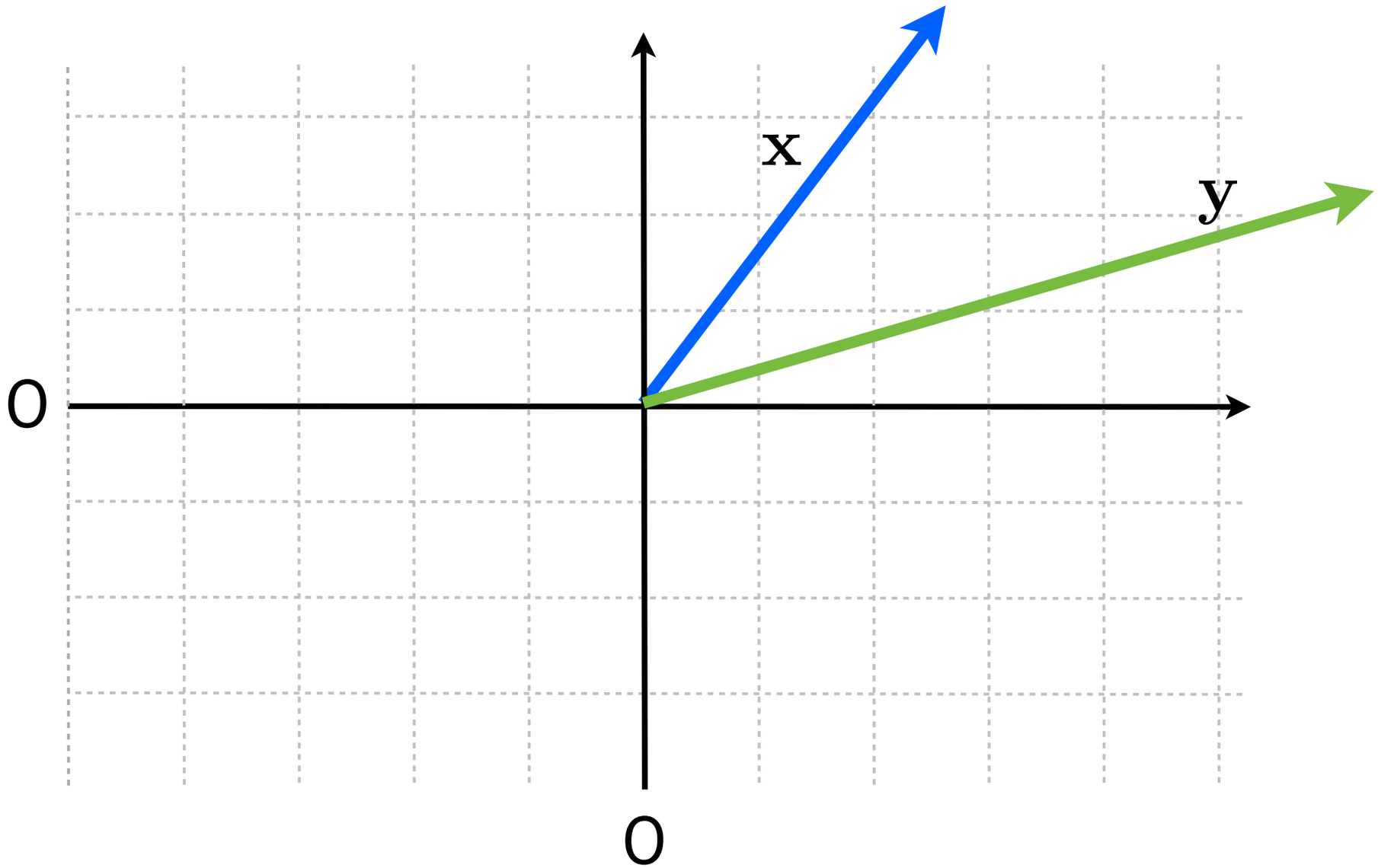


use PCA to find a lower-dimensional subspace if possible

The math behind dimensionality reduction

The inner product

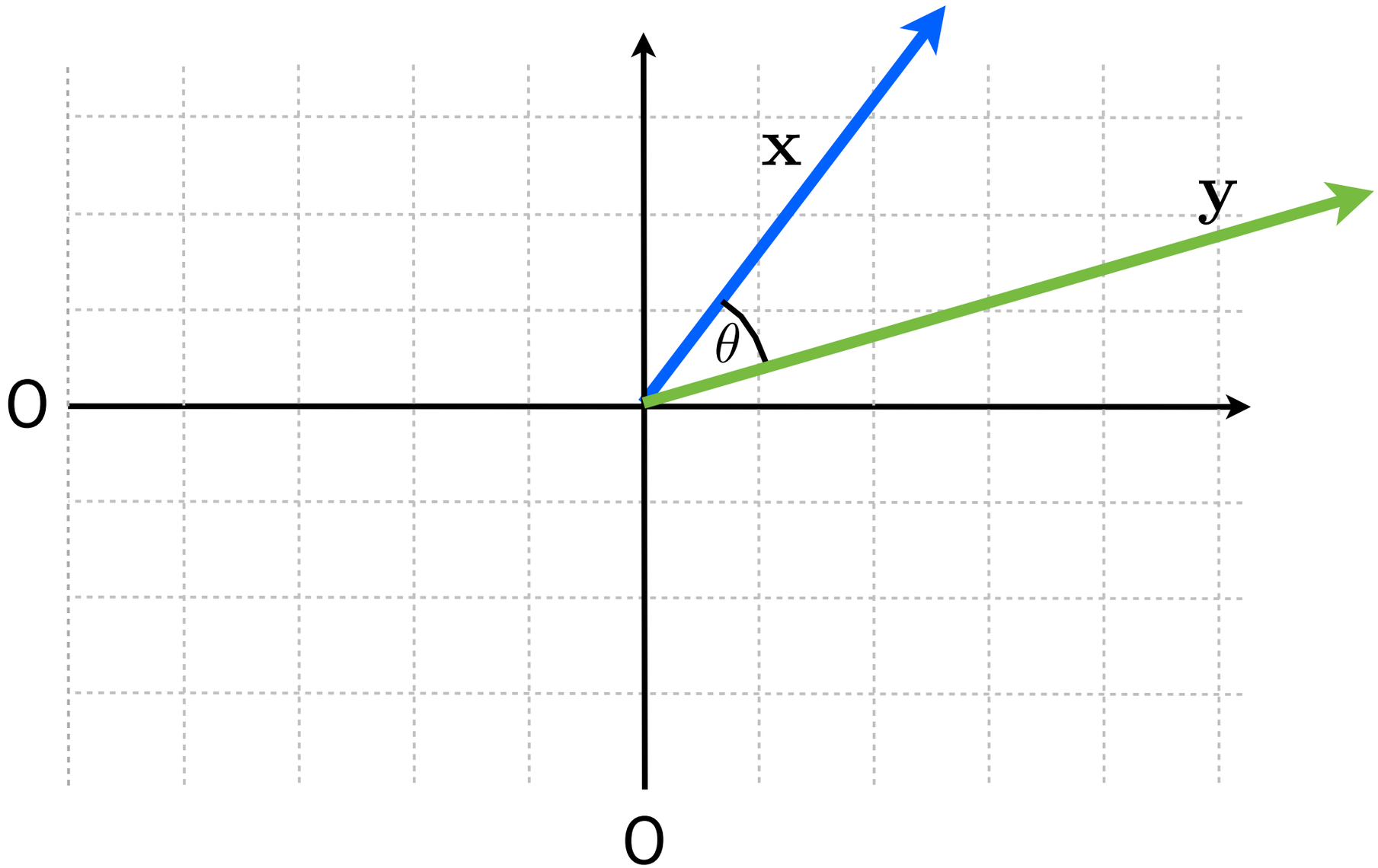
(1) computation: $\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + x_2 y_2 + \ldots = \mathbf{x}^\top \mathbf{y} = \begin{pmatrix} x_1 & x_2 & \ldots \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \ldots \end{pmatrix}$



The inner product

(1) computation: $\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + x_2 y_2 + \dots$

(2) interpretation (geometric): $\cos\theta = \dfrac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{y}\| \|\mathbf{x}\|}$

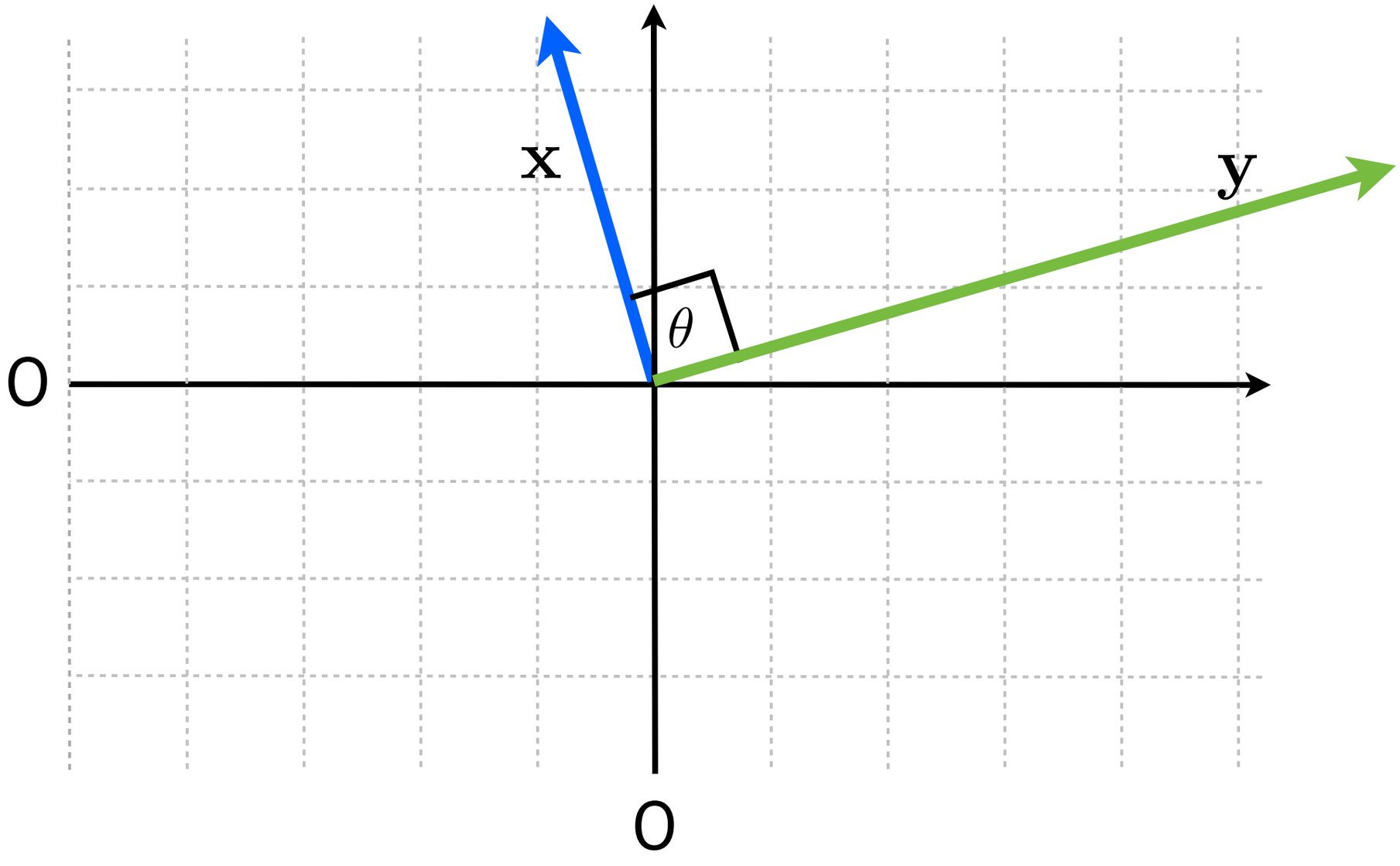


The inner product

(1) computation: $\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + x_2 y_2 + \dots$

(2) interpretation (geometric): $\cos\theta = \dfrac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{y}\| \|\mathbf{x}\|}$

$$\cos\theta = 0 \longrightarrow \theta = 90^\circ \longrightarrow \mathbf{x} \text{ orthogonal to } \mathbf{y}$$

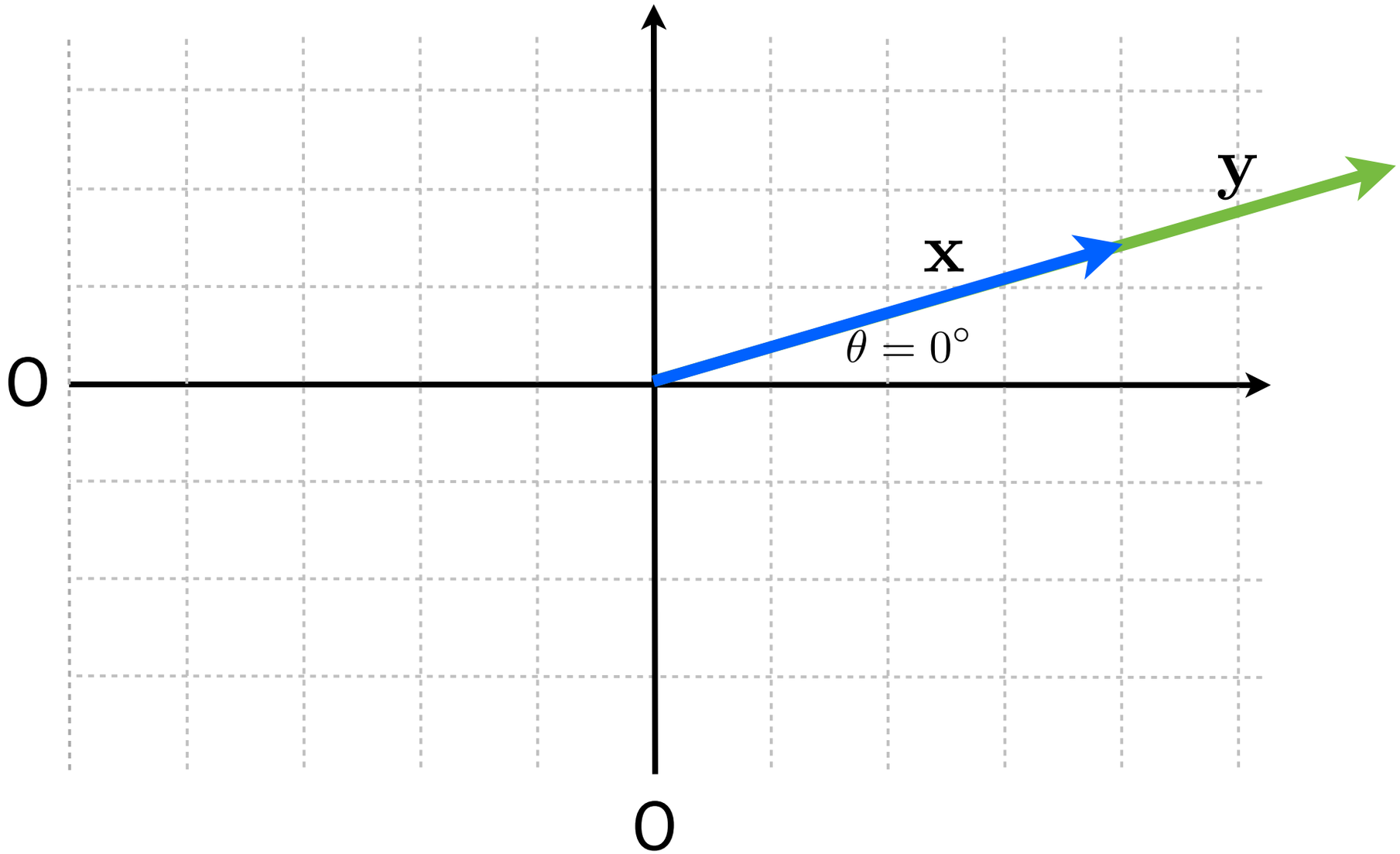


The inner product

(1) computation: $\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + x_2 y_2 + \dots$

(2) interpretation (geometric): $\cos\theta = \dfrac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{y}\| \|\mathbf{x}\|}$

$$\cos\theta = 1 \longrightarrow \theta = 0^\circ \longrightarrow \mathbf{x} \text{ parallel to } \mathbf{y}$$

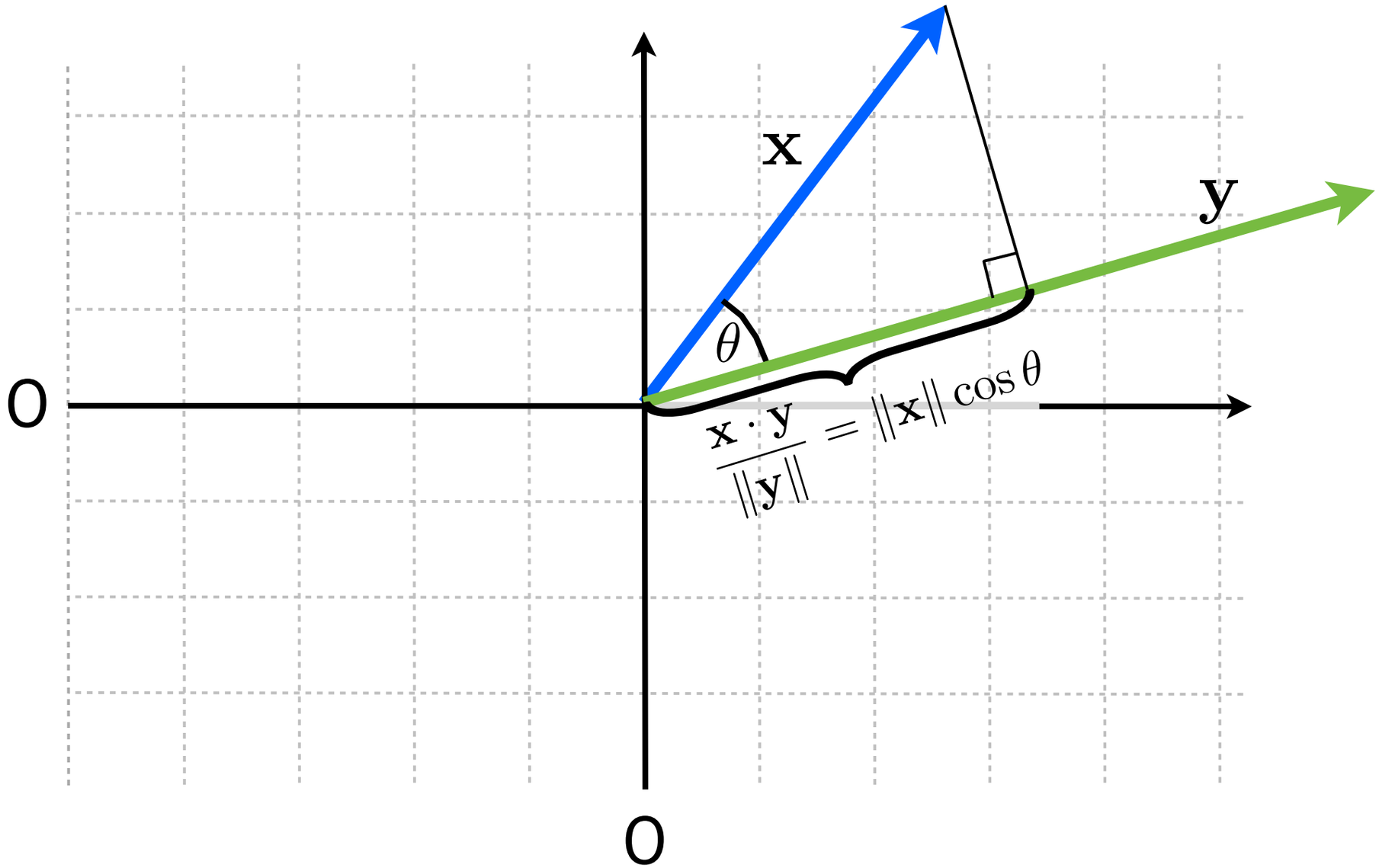


The inner product

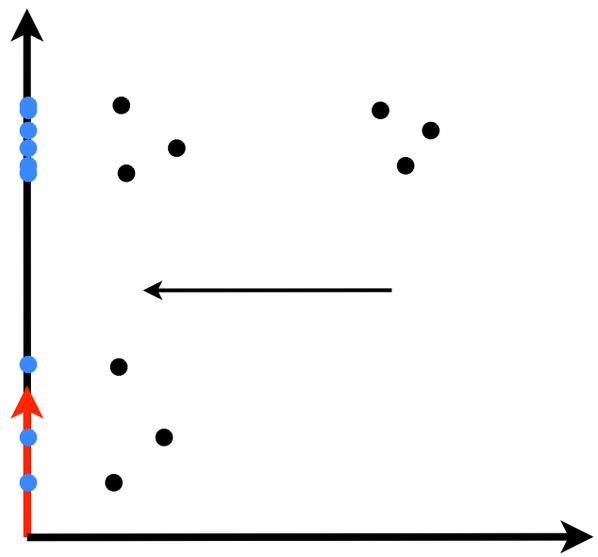
(1) computation: $\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + x_2 y_2 + \dots$

(2) interpretation (geometric): $\cos\theta = \dfrac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{y}\| \|\mathbf{x}\|}$

(3) If $\|\mathbf{y}\| = 1$, then the inner product projects $\mathbf{x}$ on $\mathbf{y}$.

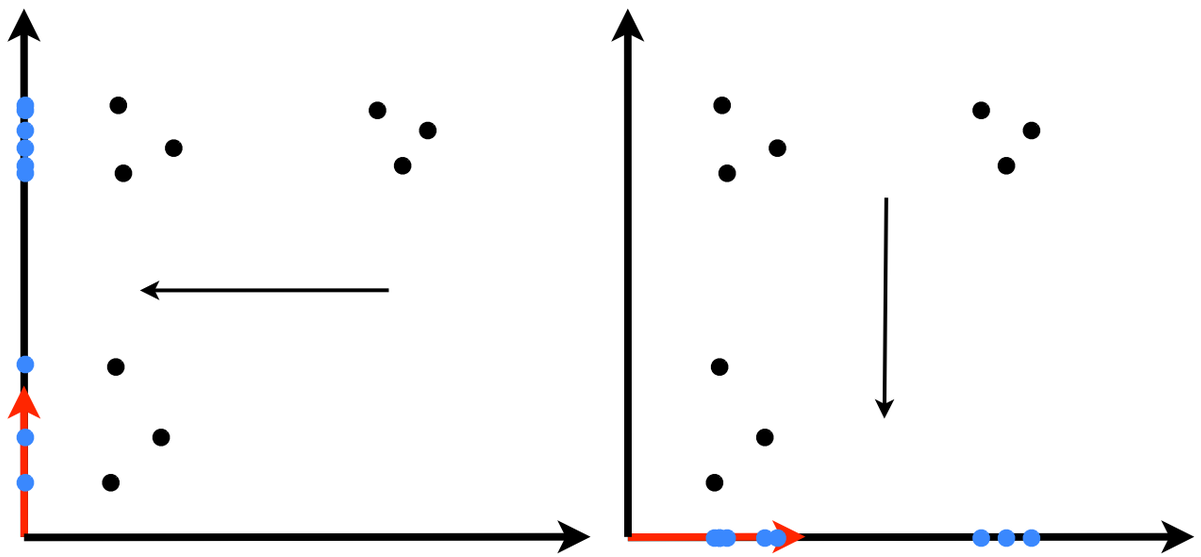


The inner product and projections



$$\mathbf{y} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

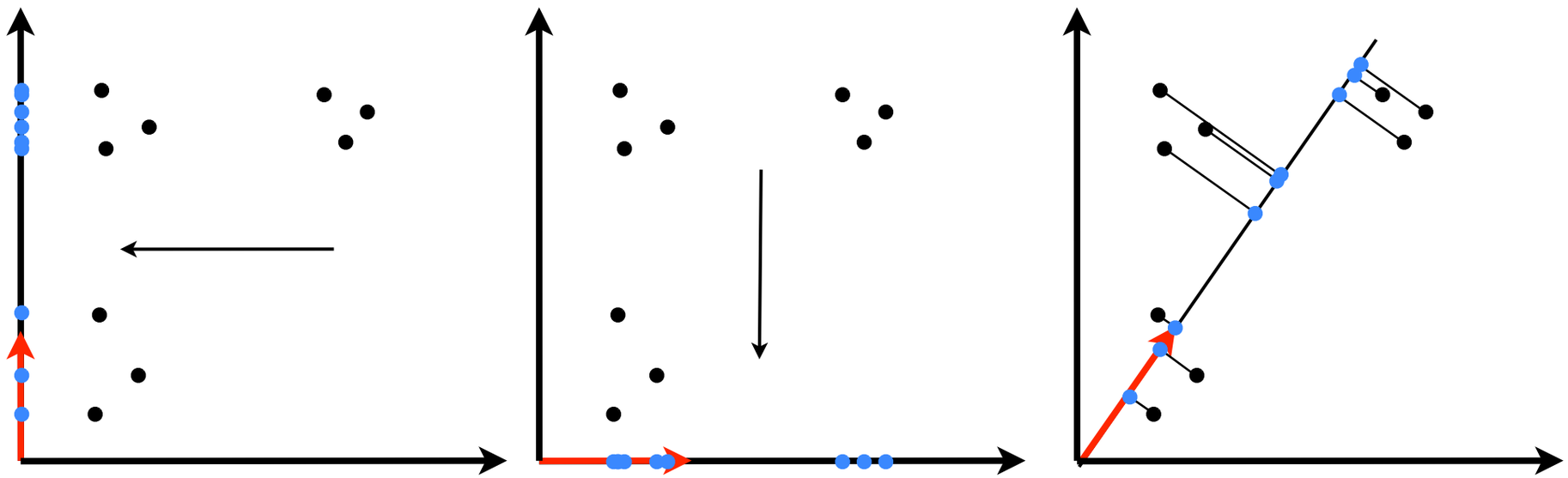
The inner product and projections



$$\mathbf{y} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\mathbf{y} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

The inner product and projections



$$\mathbf{y} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} 0.6 \\ 0.8 \end{pmatrix}$$

Eigenvalues and eigenvectors

Assume a matrix $\mathbf{A}$ and vectors $\mathbf{x}$ for which the following relation holds:

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

then $\mathbf{x}$ is an eigenvector of $\mathbf{A}$, and λ its associated eigenvalue. By convention, $\mathbf{x}$ is assumed to be normalized (unit-length).

Geometric interpretation: $\mathbf{A}$ leaves the direction of $\mathbf{x}$ unchanged.

Eigenvalues and eigenvectors

Assume a matrix $\mathbf{A}$ and vectors $\mathbf{x}$ for which the following relation holds:

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

then $\mathbf{x}$ is an eigenvector of $\mathbf{A}$, and λ its associated eigenvalue. By convention, $\mathbf{x}$ is assumed to be normalized (unit-length).

Geometric interpretation: $\mathbf{A}$ leaves the direction of $\mathbf{x}$ unchanged.

1. Identity Matrix $\mathbf{I}$: every vector is an eigenvector

Eigenvalues and eigenvectors

Assume a matrix **A** and vectors **x** for which the following relation holds:

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

then **x** is an eigenvector of **A**, and λ its associated eigenvalue. By convention, **x** is assumed to be normalized (unit-length).

Geometric interpretation: **A** leaves the direction of **x** unchanged.

1. Identity Matrix **I**: every vector is an eigenvector
2. Diagonal Matrix **D:** eigenvectors are the Eucledian unit vectors

Eigenvalues and eigenvectors

Assume a matrix **A** and vectors **x** for which the following relation holds:

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

then **x** is an eigenvector of **A**, and λ its associated eigenvalue. By convention, **x** is assumed to be normalized (unit-length).

Geometric interpretation: **A** leaves the direction of **x** unchanged.

1. Identity Matrix **I**: every vector is an eigenvector
2. Diagonal Matrix **D:** eigenvectors are the Eucledian unit vectors
3. Orthogonal matrix **U:** eigenvectors / values are often complex (sorry!)

Eigenvalues and eigenvectors

Assume a matrix **A** and vectors **x** for which the following relation holds:

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

then **x** is an eigenvector of **A**, and λ its associated eigenvalue. By convention, **x** is assumed to be normalized (unit-length).

Geometric interpretation: **A** leaves the direction of **x** unchanged.

1. Identity Matrix **I**: every vector is an eigenvector
2. Diagonal Matrix **D:** eigenvectors are the Eucledian unit vectors
3. Orthogonal matrix **U:** eigenvectors / values are often complex (sorry!)
4. Symmetric matrix **S:** all eigenvalues are real, eigenvectors are orthonormal

$$\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$$

Note: the inverse of an orthogonal matrix $\mathbf{U}$ is given by $\mathbf{U}^{-1} = \mathbf{U}^\top$

Eigenvalues and eigenvectors

Assume a matrix **A** and vectors **x** for which the following relation holds:

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

then **x** is an eigenvector of **A**, and λ its associated eigenvalue. By convention, **x** is assumed to be normalized (unit-length).

Geometric interpretation: **A** leaves the direction of **x** unchanged.

1. Identity Matrix **I**: every vector is an eigenvector
2. Diagonal Matrix **D:** eigenvectors are the Eucledian unit vectors
3. Orthogonal matrix **U:** eigenvectors / values are often complex (sorry!)
4. Symmetric matrix **S:** all eigenvalues are real, eigenvectors are orthonormal
5. Projection matrix **P:** eigenvalues are zeros or ones

Orthogonal matrices revisited

$$\mathbf{U} = \begin{pmatrix} U_{11} & U_{12} & U_{13} \\ U_{21} & U_{22} & U_{23} \\ U_{31} & U_{32} & U_{33} \end{pmatrix}$$

$$\mathbf{U}_{\cdot,2} \cdot \mathbf{U}_{\cdot,3} = 0$$

1. All column vectors are orthogonal to each other (for i ≠ j)

$$\mathbf{U}_{\cdot,i} \cdot \mathbf{U}_{\cdot,j} = 0$$

2. And have unit norm (for i=j)

$$\mathbf{U}_{\cdot,i} \cdot \mathbf{U}_{\cdot,i} = 1$$

Orthogonal matrices revisited

$$\mathbf{U} = \begin{pmatrix} U_{11} & U_{12} & U_{13} \\ U_{21} & U_{22} & U_{23} \\ U_{31} & U_{32} & U_{33} \end{pmatrix}$$

$$\|\mathbf{U}_{.,3}\| = 1$$

1. All column vectors are orthogonal to each other (for i ≠ j)

$$\mathbf{U}_{.,i} \cdot \mathbf{U}_{.,j} = 0$$

2. And have unit length (or norm)

$$\mathbf{U}_{.,i} \cdot \mathbf{U}_{.,i} = 1$$

3. We write in all brevity:

$$\mathbf{U}_{.,i} \cdot \mathbf{U}_{.,j} = \delta_{ij}$$

$$\mathbf{U}^\top \mathbf{U} = \mathbf{I}$$

Example:

Principal Component Analysis

Assume you have N data points (vectors) $\mathbf{x}_n$
Their average is given by

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$$

Example:

Principal Component Analysis

Assume you have N data points (vectors) $\mathbf{x}_n$
Their average is given by

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$$

Their covariance matrix is given by

$$\mathbf{C} = \frac{1}{N-1}\sum_{n=1}^{N}(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^\top$$

Example:

Principal Component Analysis

Assume you have N data points (vectors) $\mathbf{x}_n$
Their average is given by

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$$

Their covariance matrix is given by

$$\mathbf{C} = \frac{1}{N-1}\sum_{n=1}^{N}(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^\top$$

The covariance matrix is symmetric and can be decomposed as

$$\mathbf{C} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$$

Example:

Principal Component Analysis

Assume you have N data points (vectors) $\mathbf{x}_n$
Their average is given by

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$$

Their covariance matrix is given by

$$\mathbf{C} = \frac{1}{N-1}\sum_{n=1}^{N}(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^\top$$

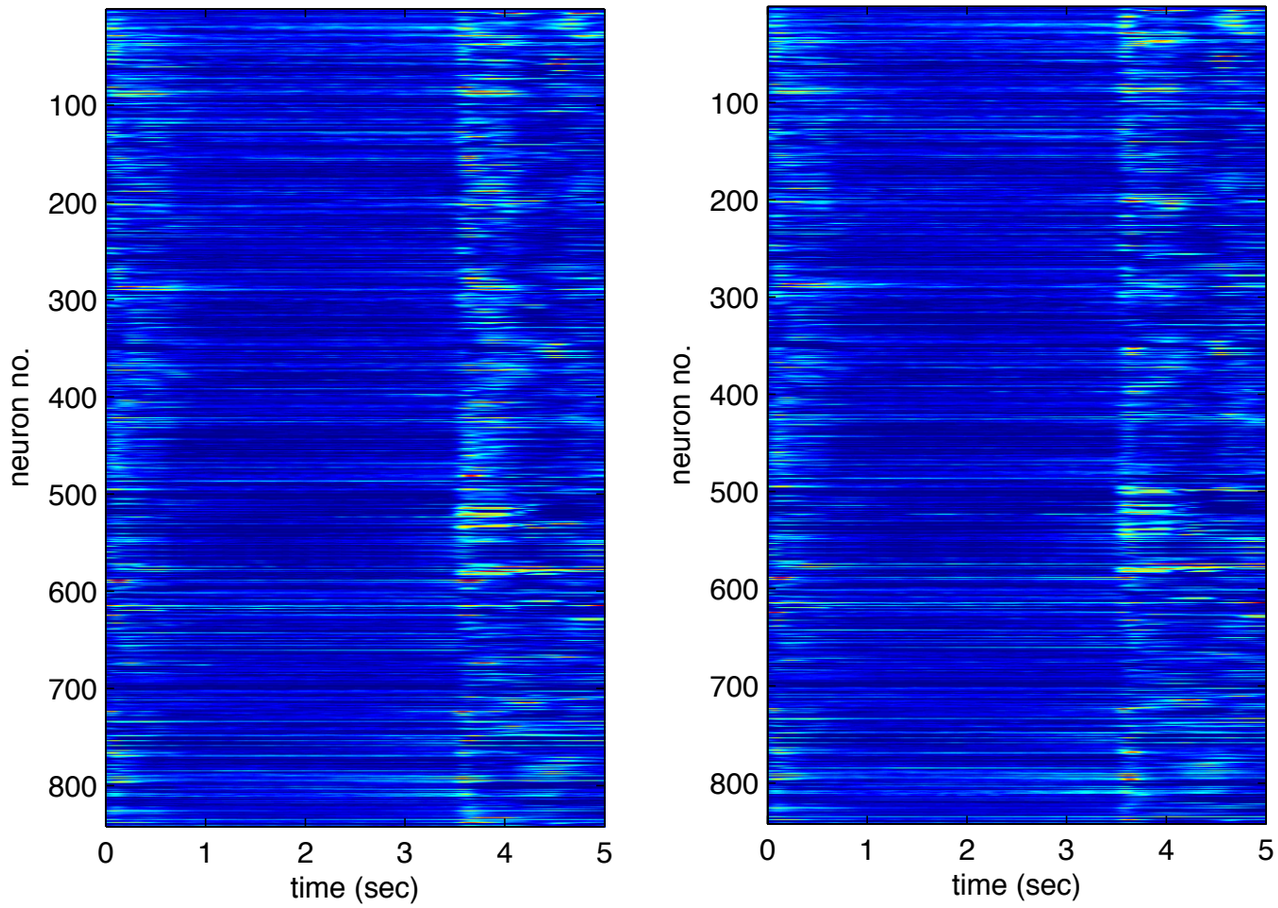
The covariance matrix is symmetric and can be decomposed as

$$\mathbf{C} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$$

The columns of $\mathbf{U}$ are the principal axes of the data, $\mathbf{y} = \mathbf{U}^\top\mathbf{x}$ its principal components

Exercise:

PCA on Romo data



To-do-list

(1) Compute and plot data matrix X
(2) 'center' the data
(3) compute and plot covariance matrix
(4) determine eigenvalues and eigenvectors of this matrix
(5) plot eigenvalues
(6) compute and plot first principal components

data courtesy of Ranulfo Romo (UNAM)

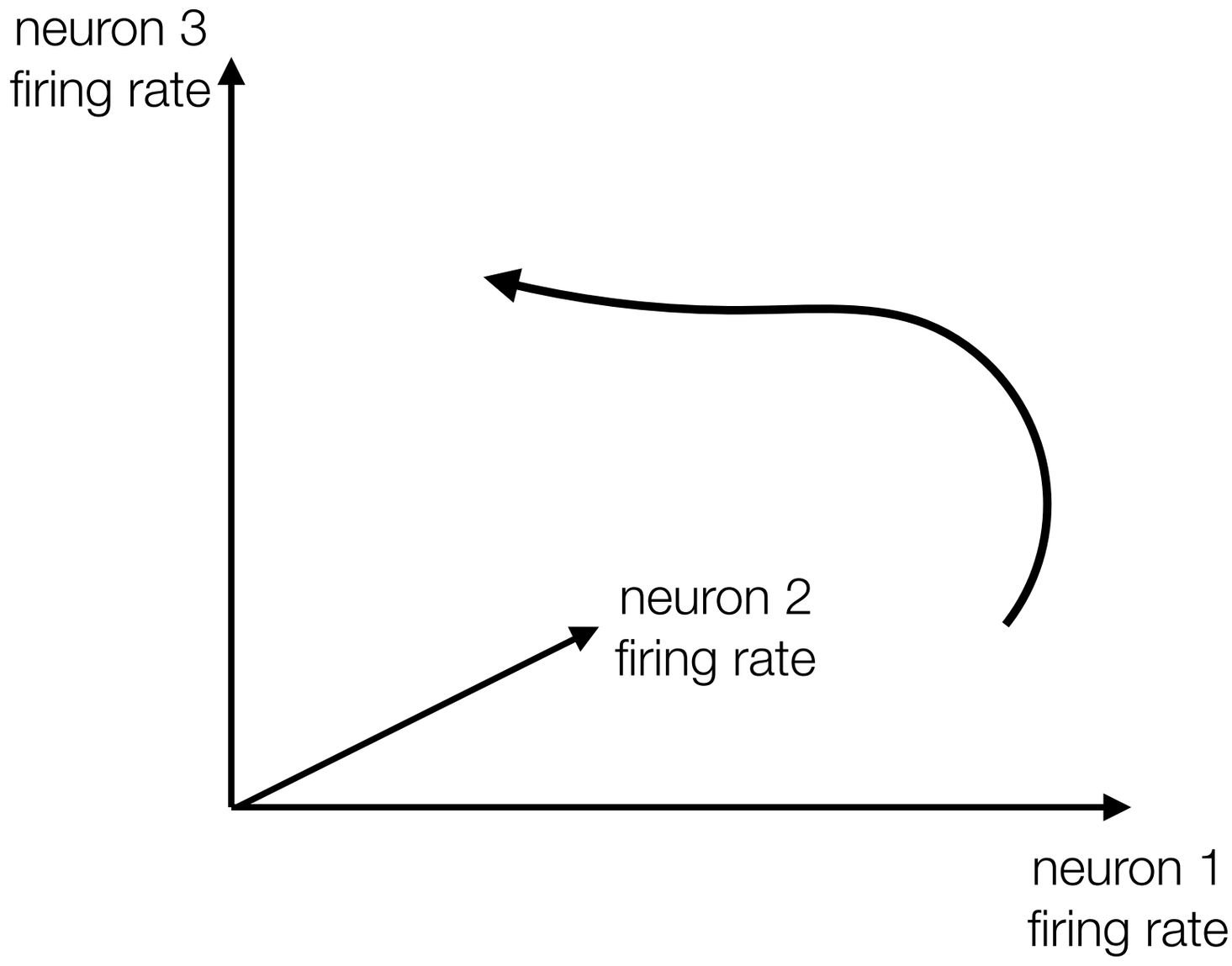
Outline:

1. Exercise: Get the ‘data’ into the right format!
    1. Spike raster
    2. Peristimulus time histogram (PSTH)
    3. Generate Data matrix with PSTHs for all cells

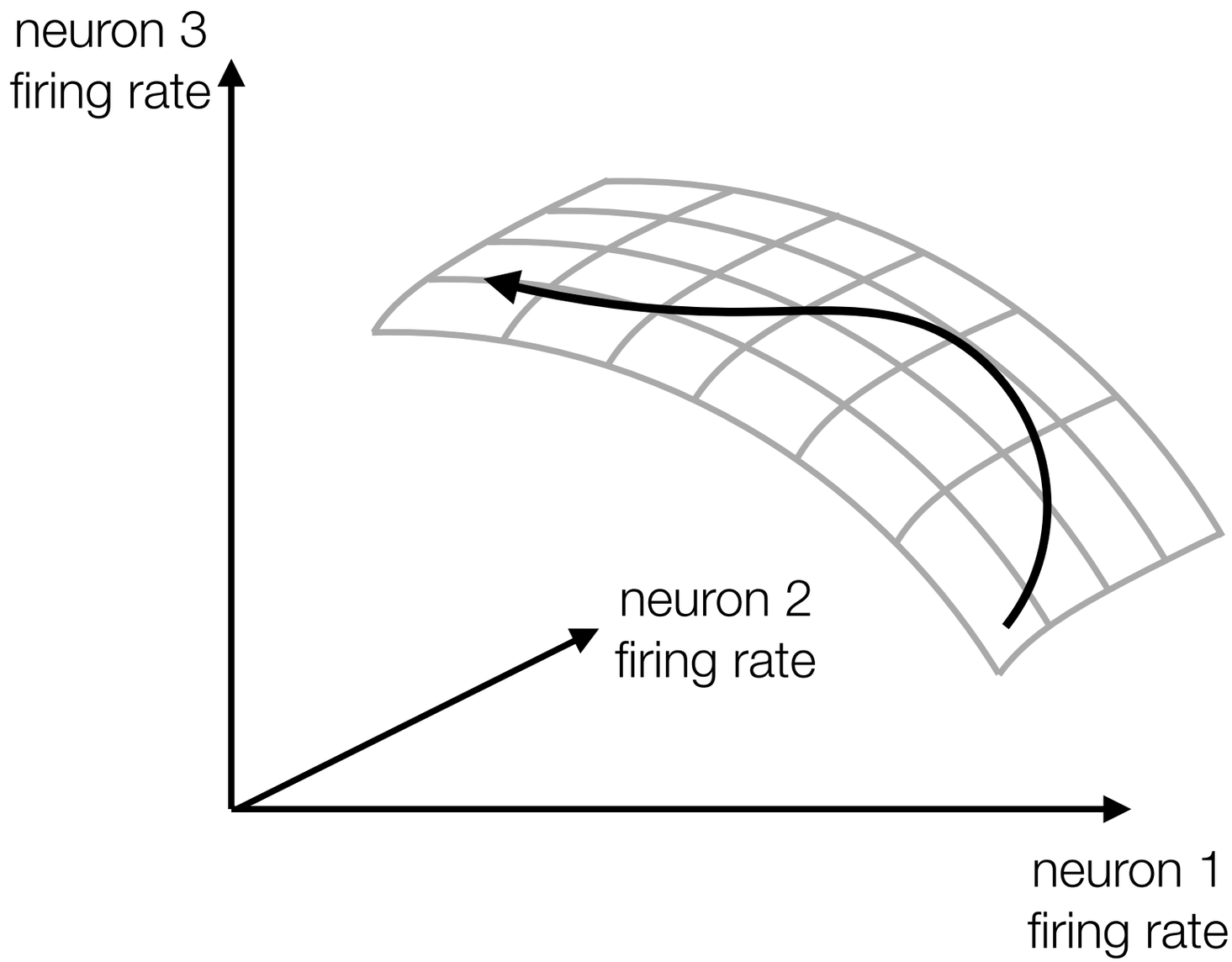
2. Lecture + Exercise: PCA on all cells

3. Lecture: demixed PCA

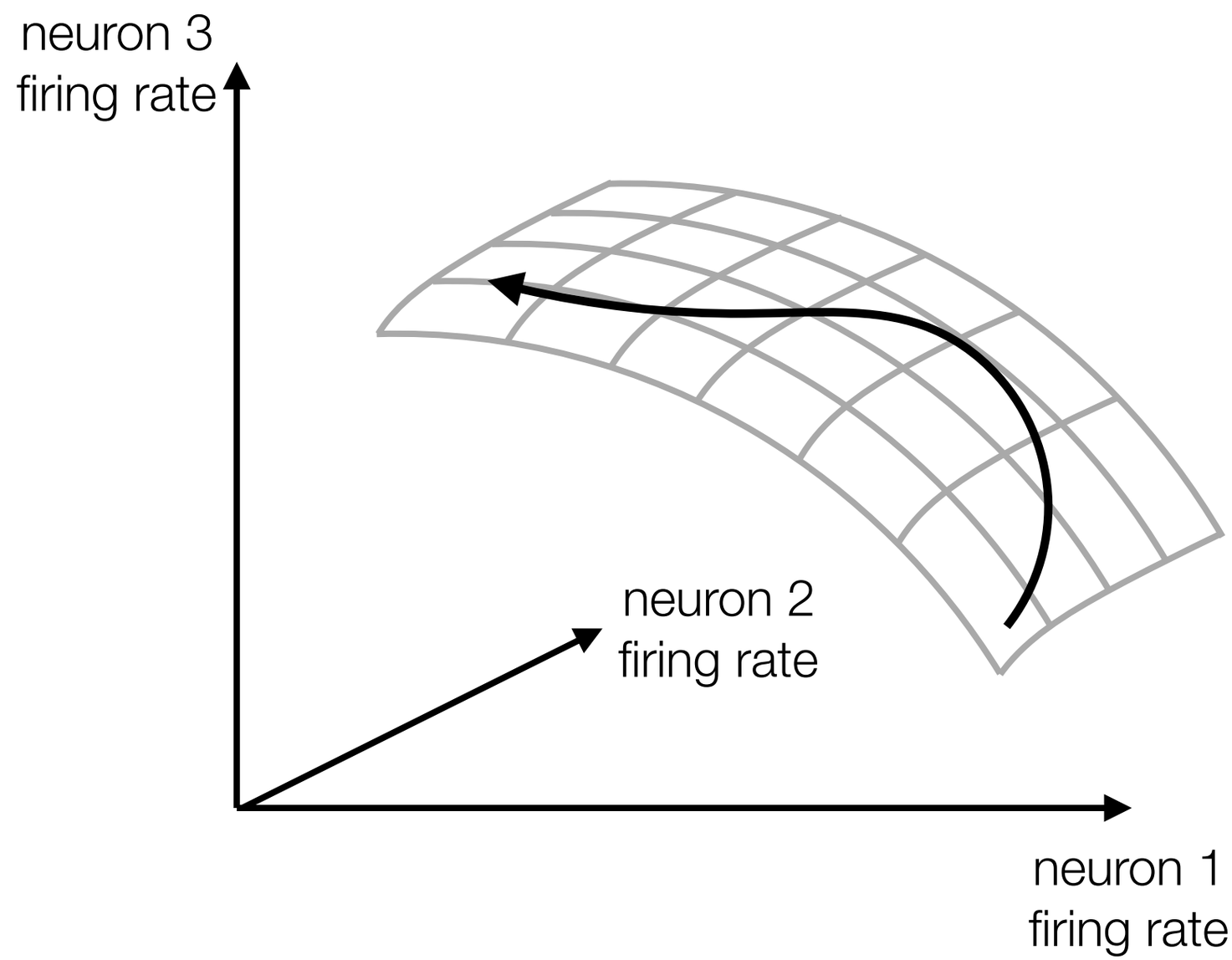
State space embedding and ‘neural manifolds’



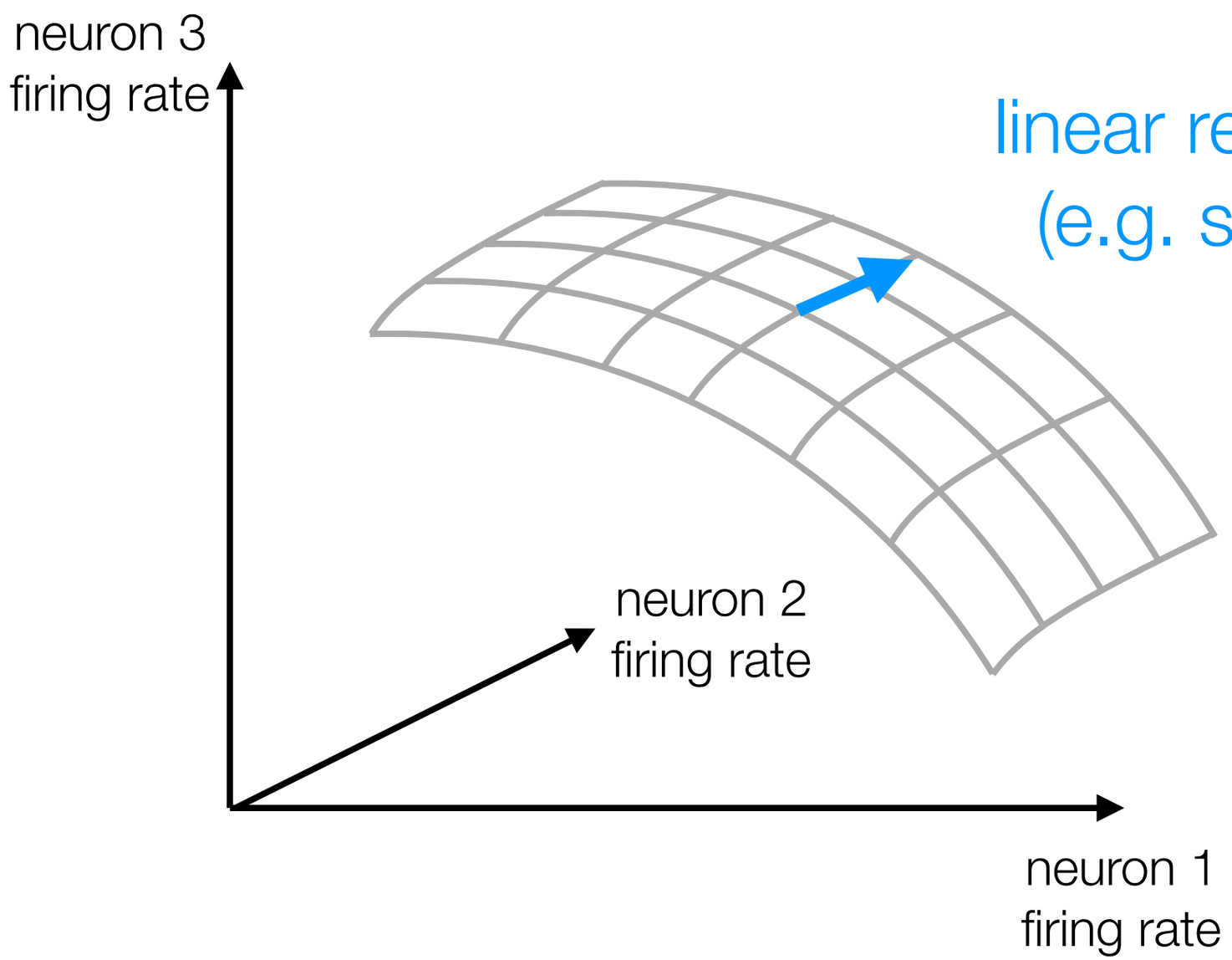
State space embedding and ‘neural manifolds’



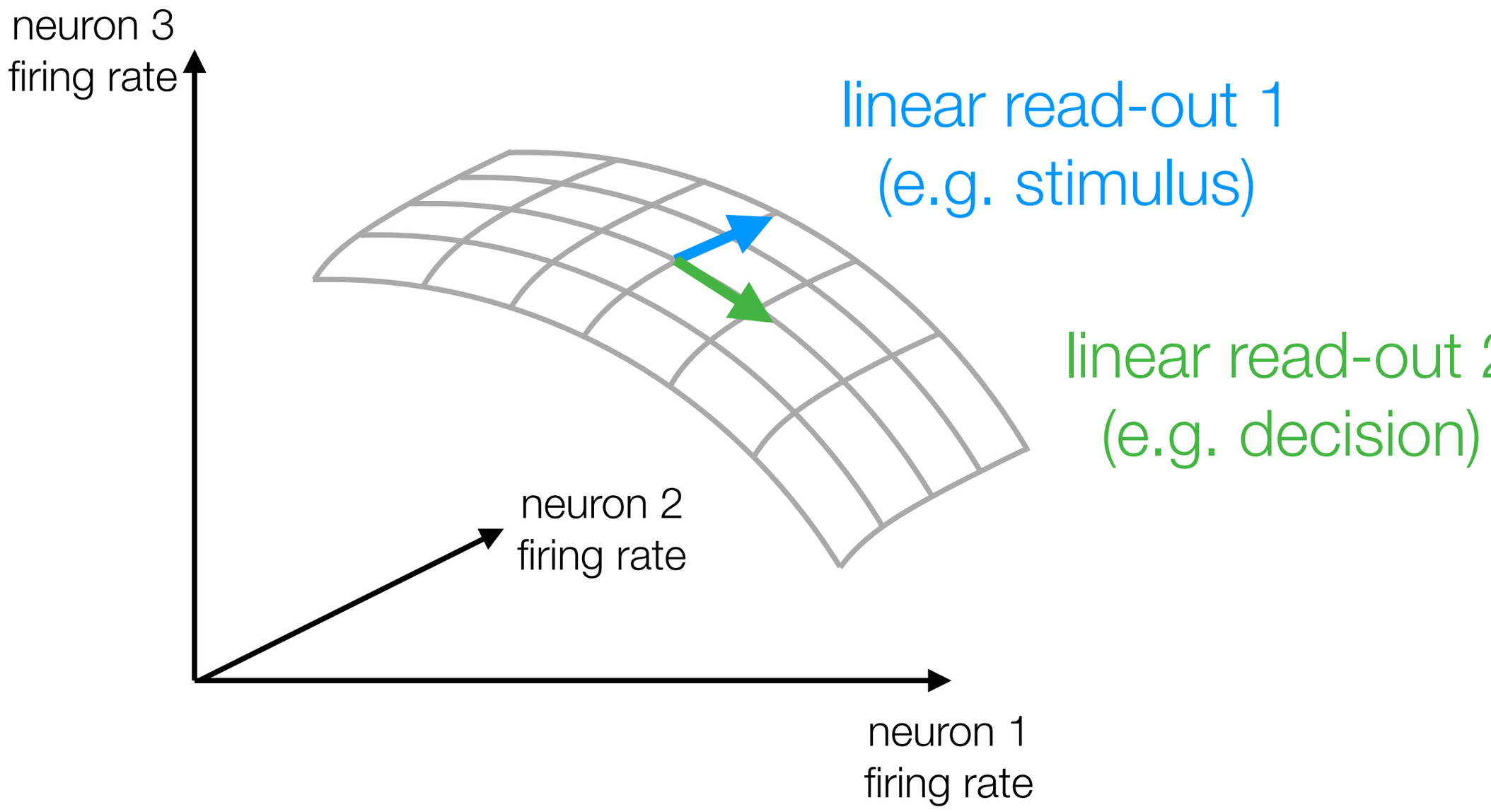
Can we assign meaning to the manifold?



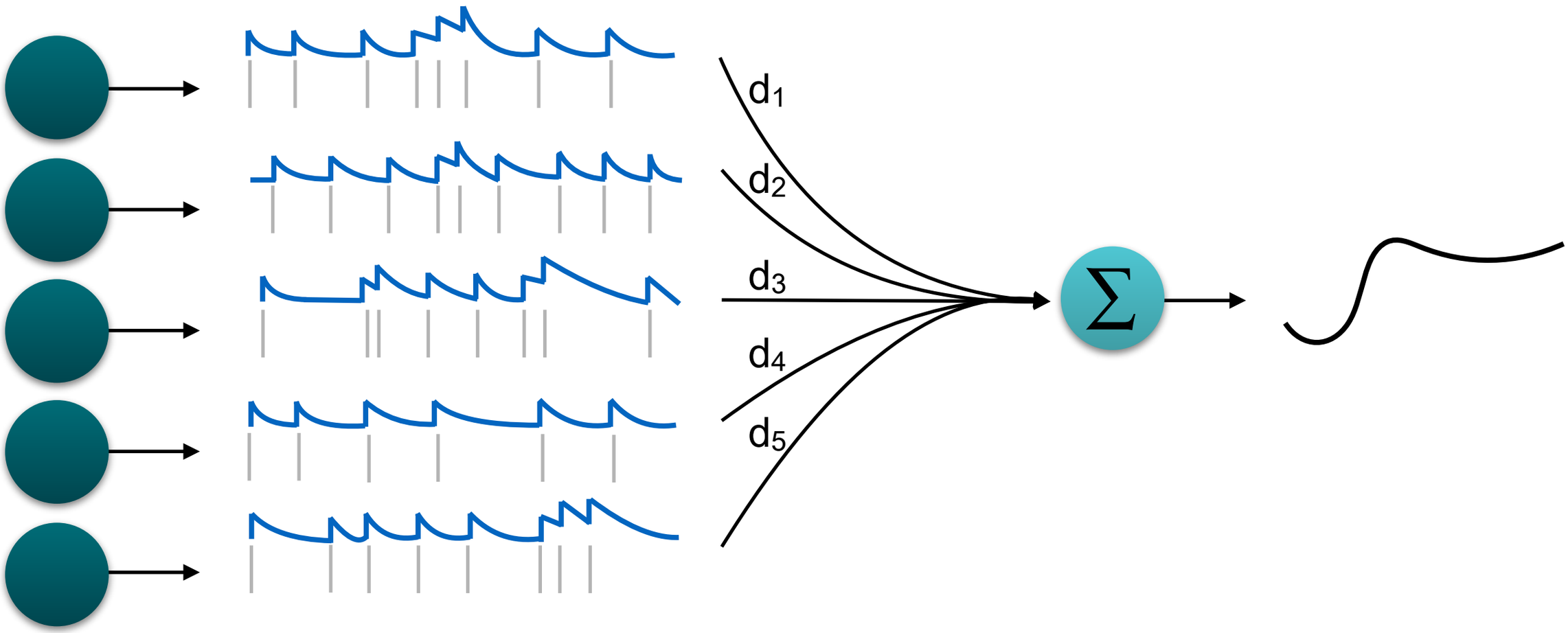
Movement on manifold = change in linear read-out



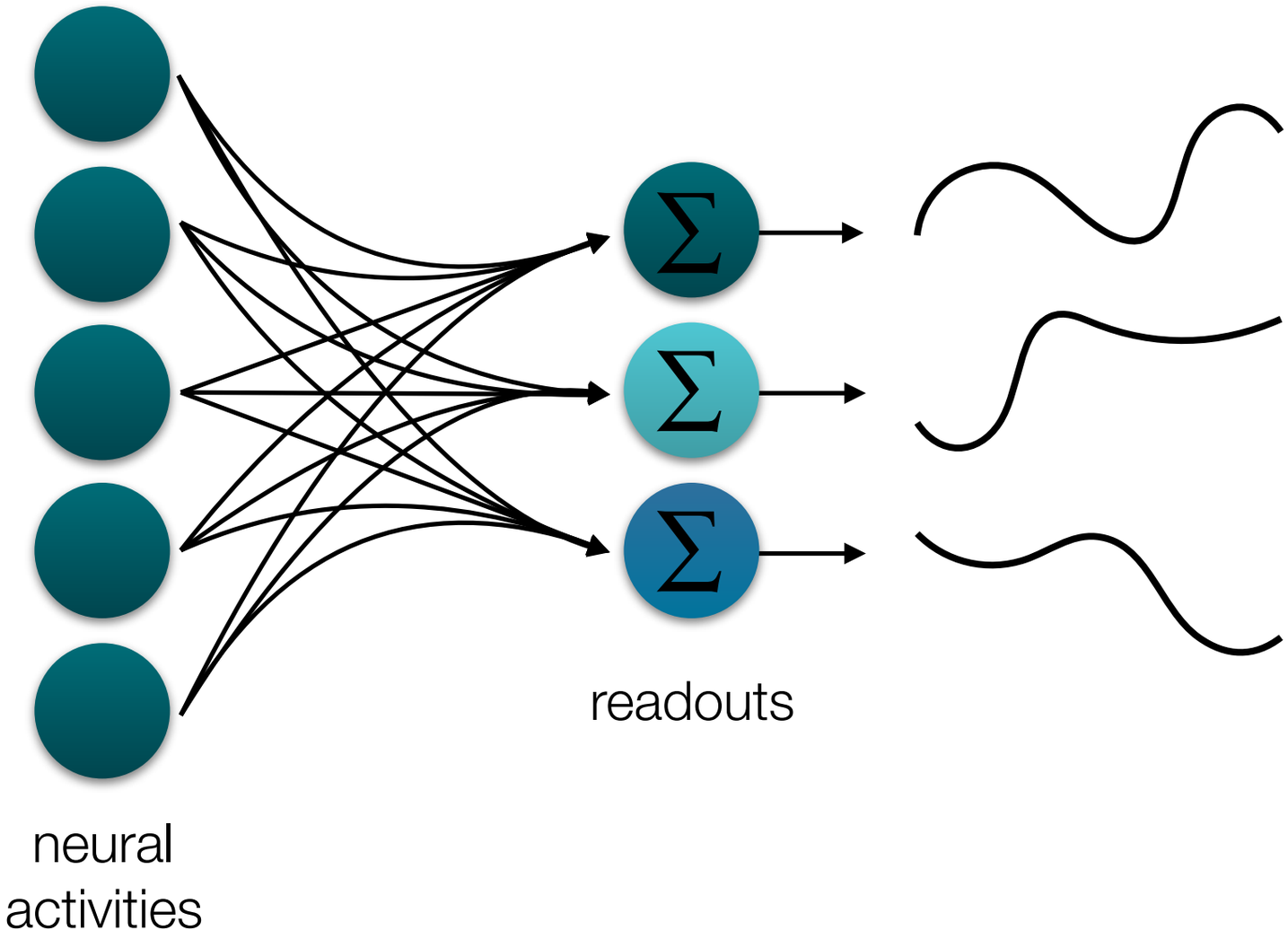
Movement on manifold = change in linear read-out



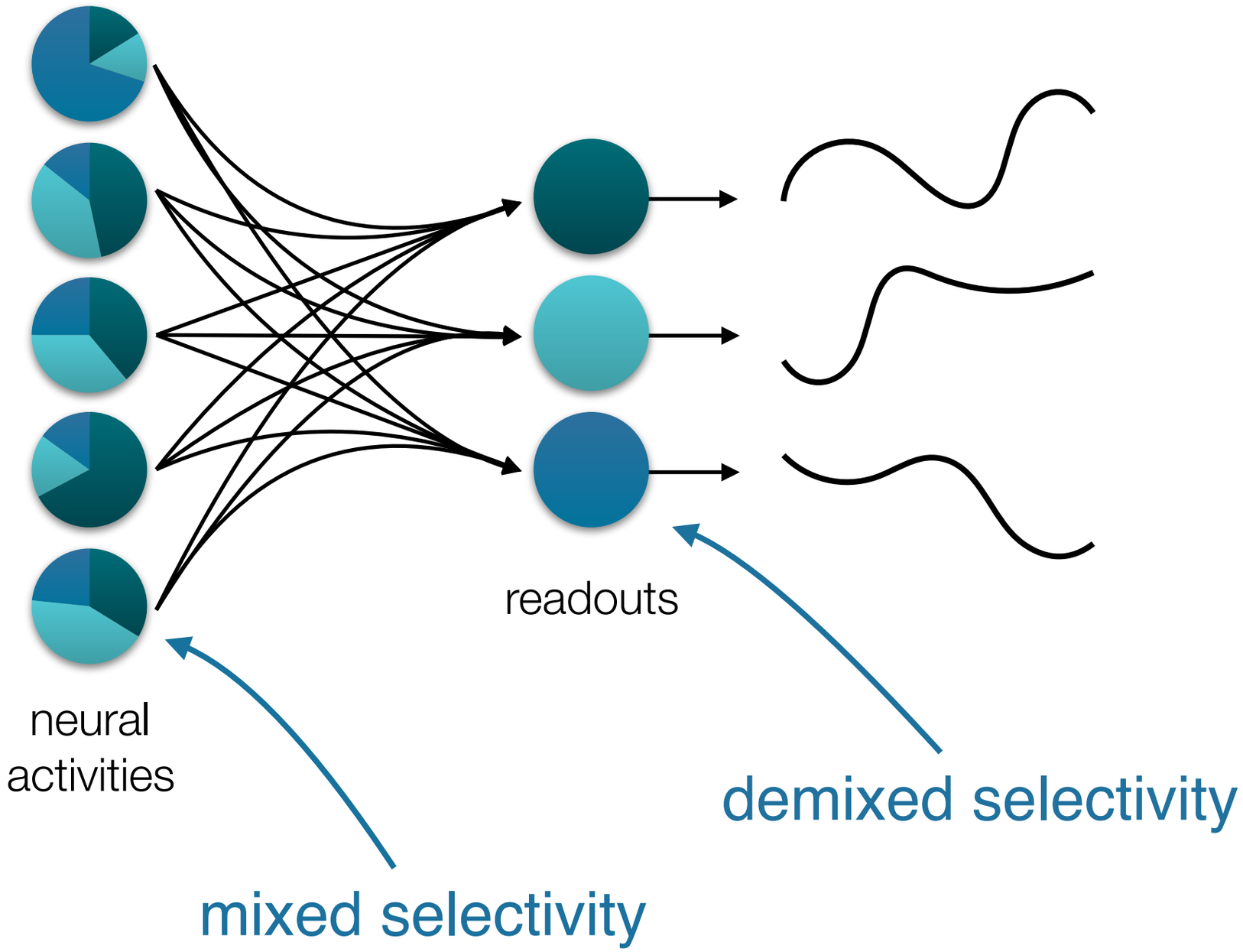
All approaches are based on linear readouts from the population



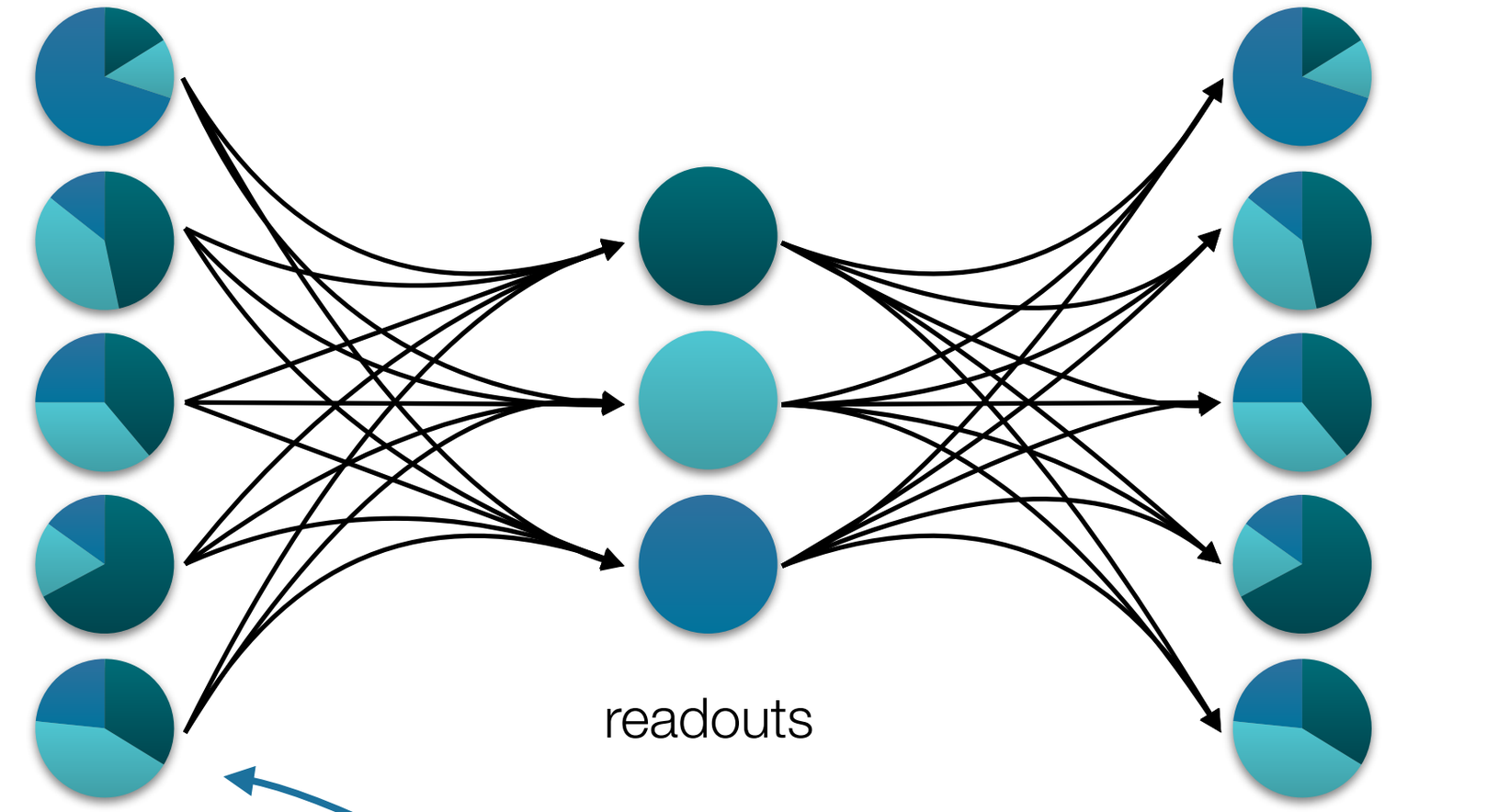
All approaches are based on multiple linear readouts from the population



Our goal: choose readouts that
(1) demix dependencies on task parameters …

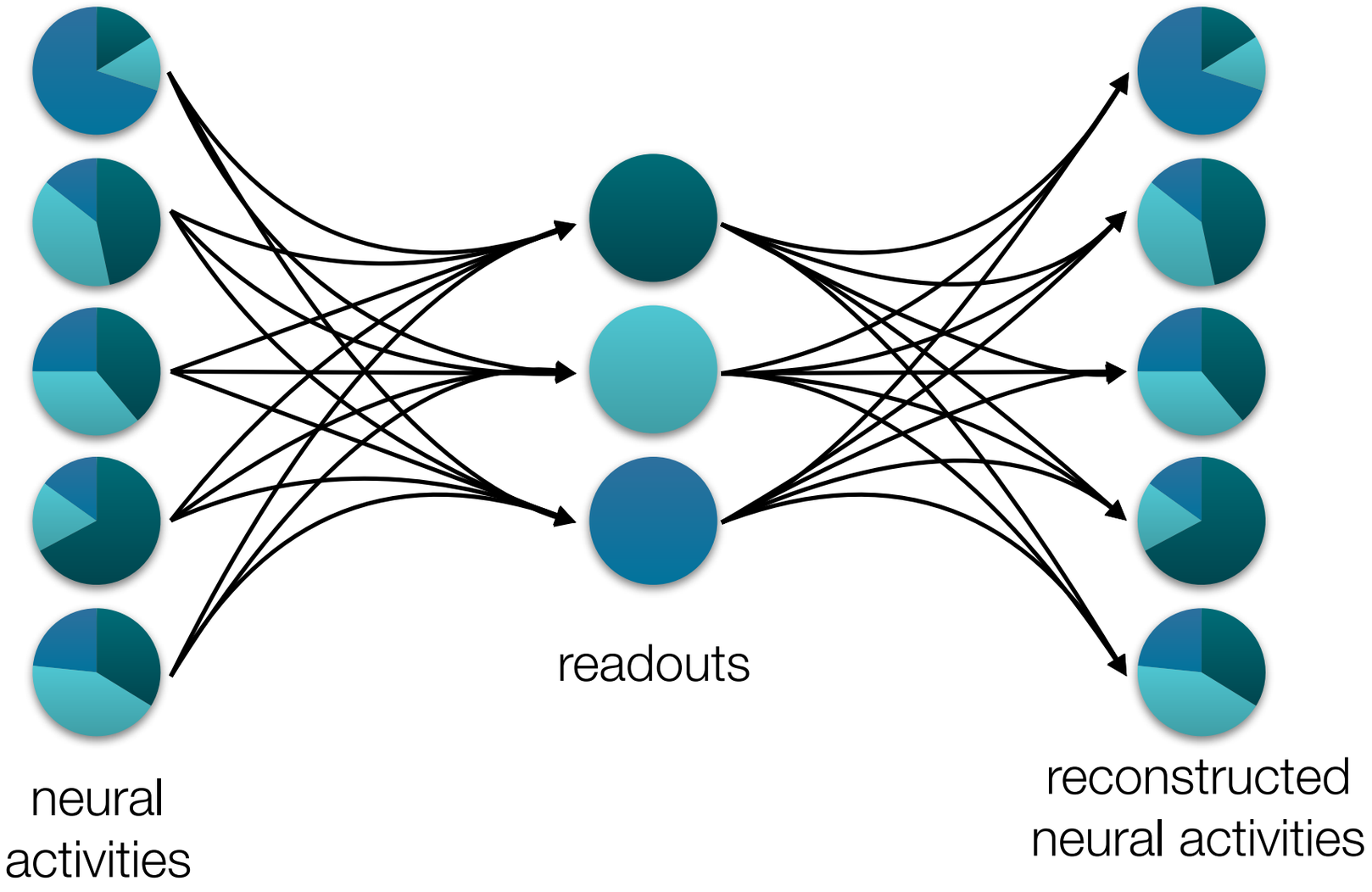


Our goal: choose readouts that (2) and capture all aspects of activity



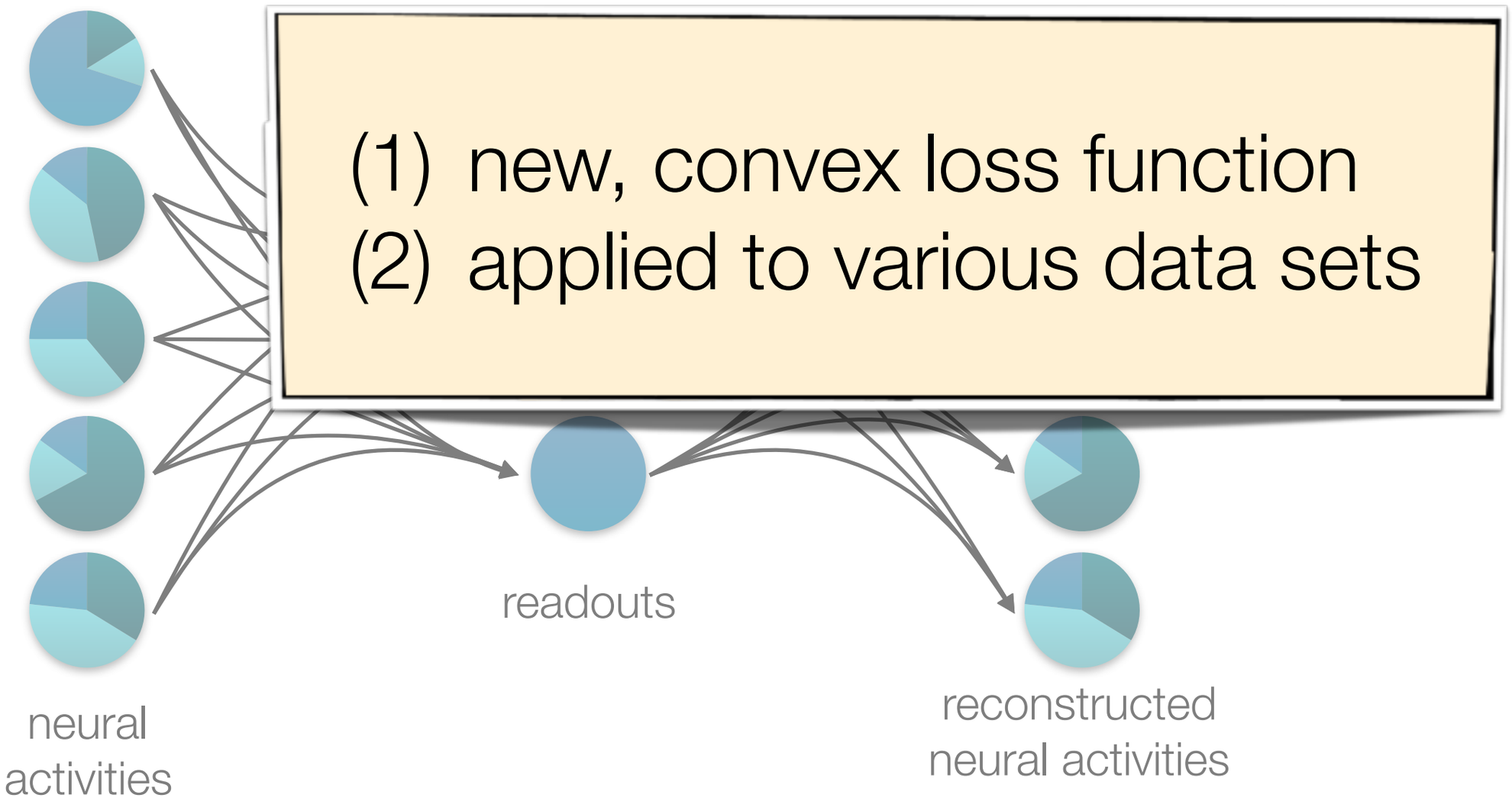
every spike matters

‘Demixed’ principal component analysis
(Brendel et al, NIPS, 2011)

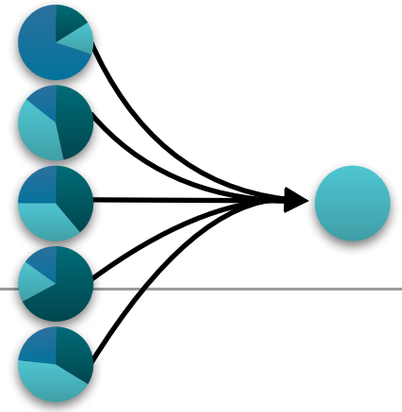


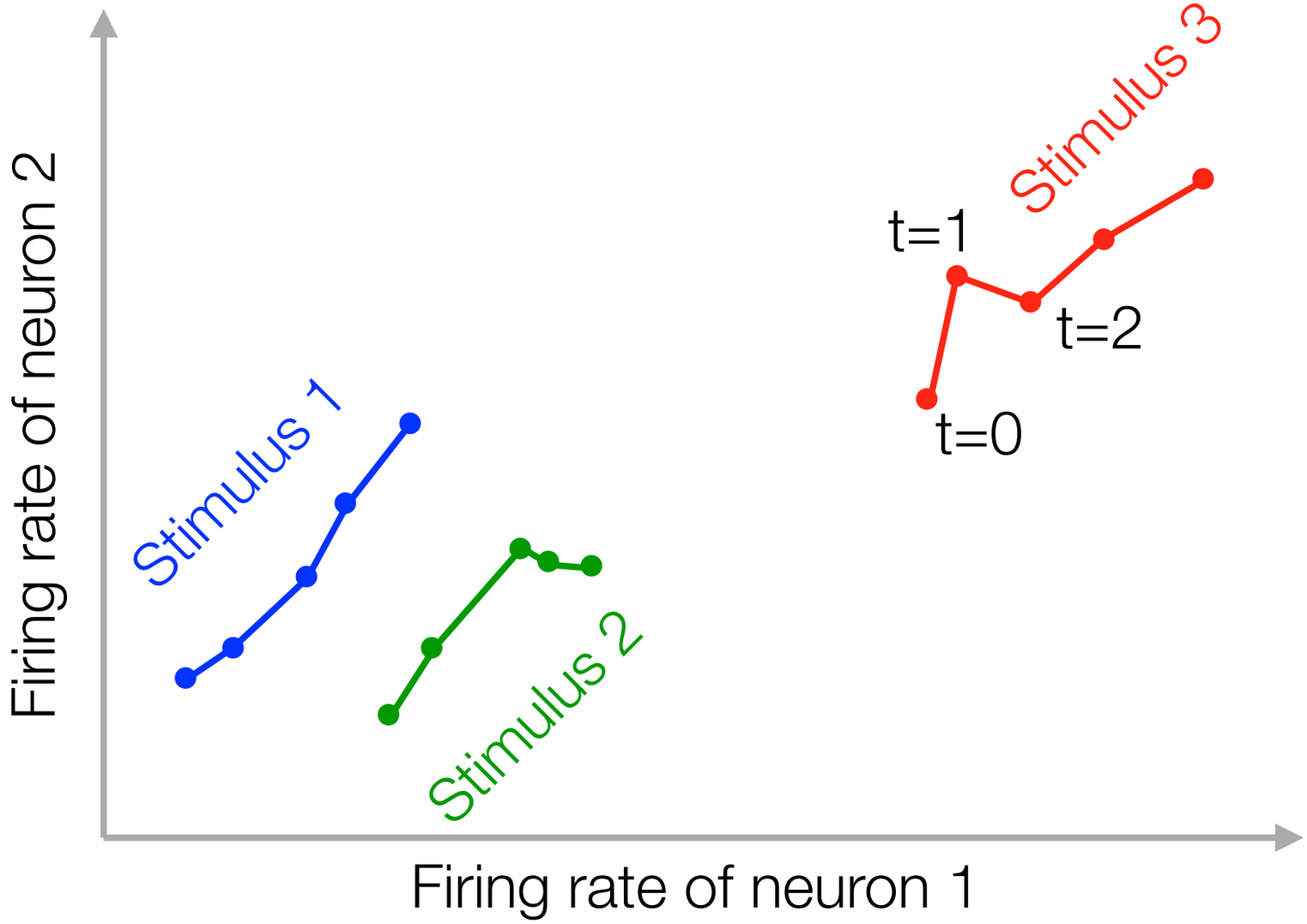
‘Demixed’ principal component analysis 2.0

(Kobak, Brendel et al, 2016, eLife)

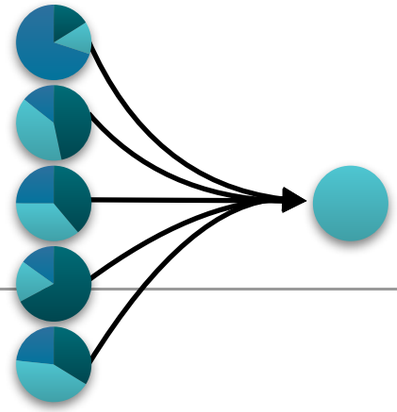


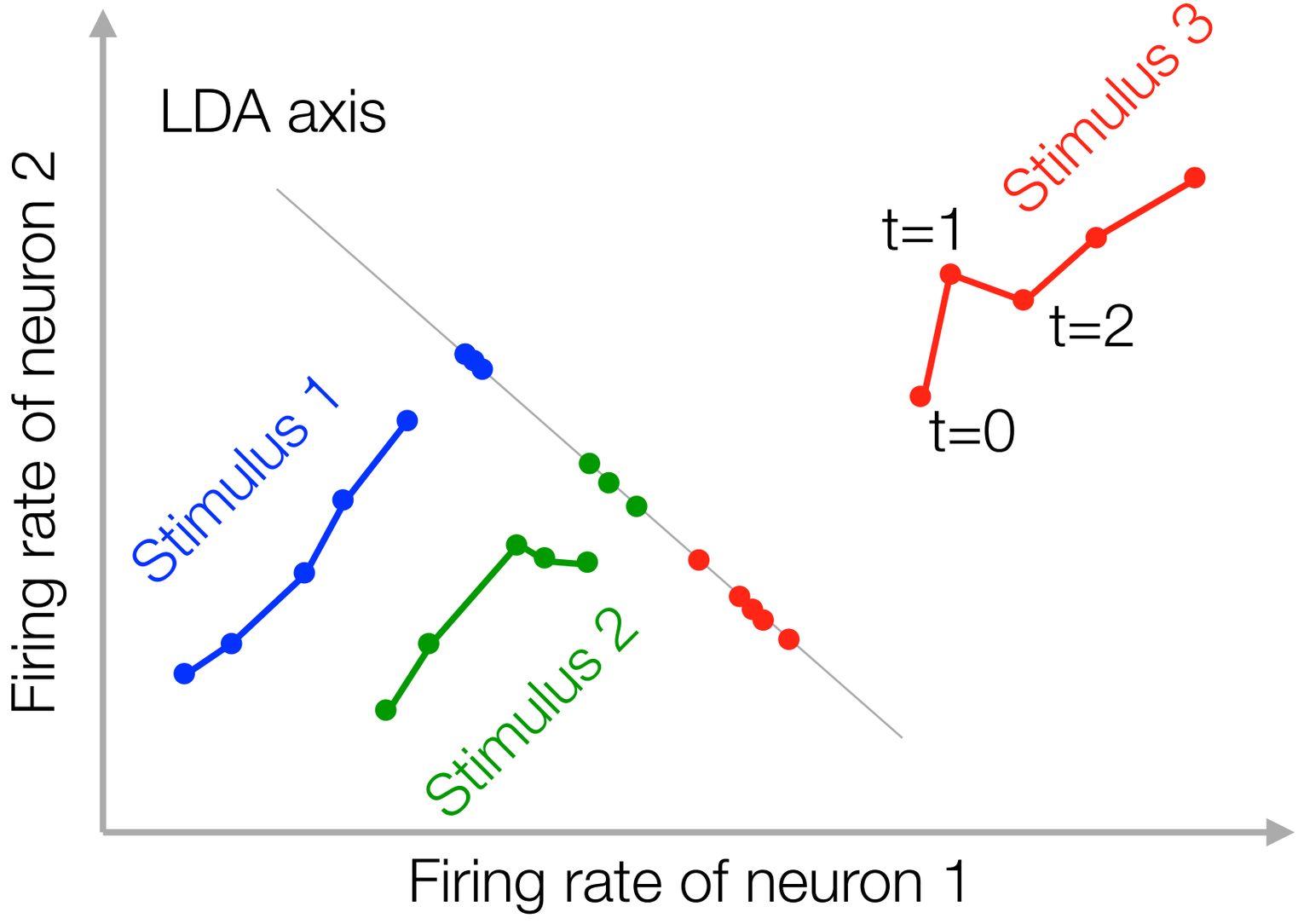
Linear readouts revisited …



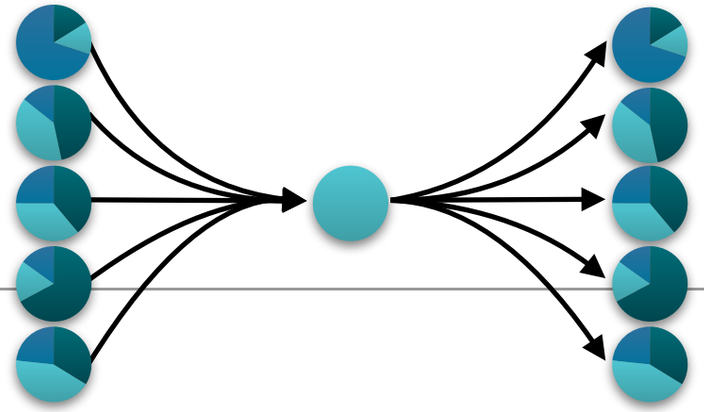


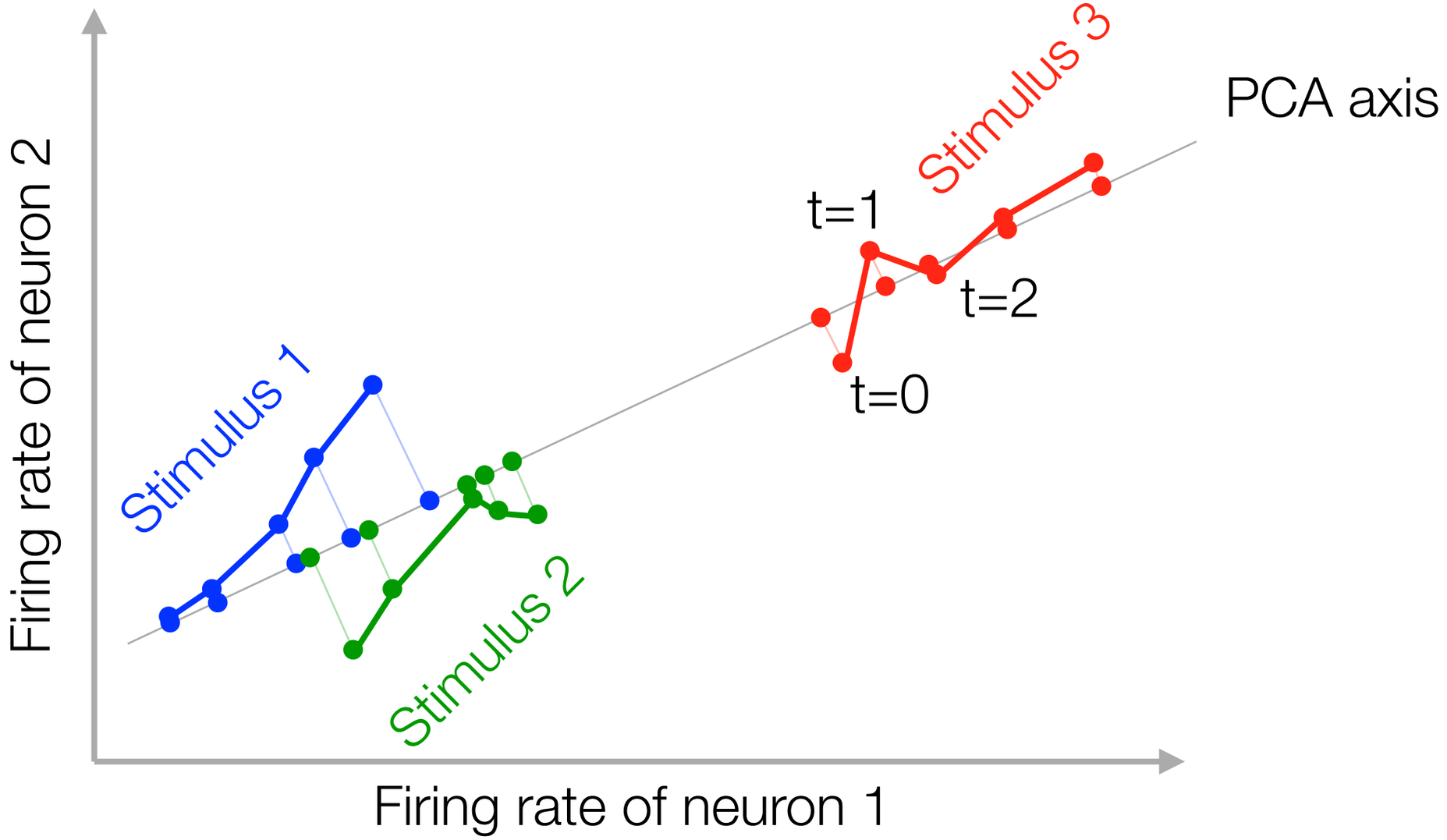
Linear readout with LDA



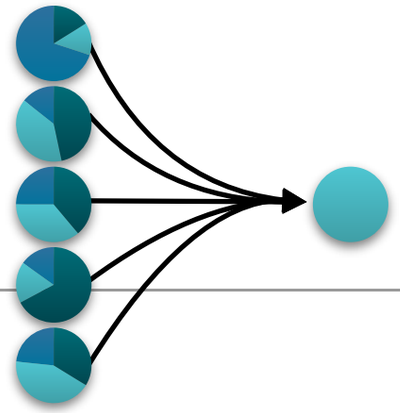


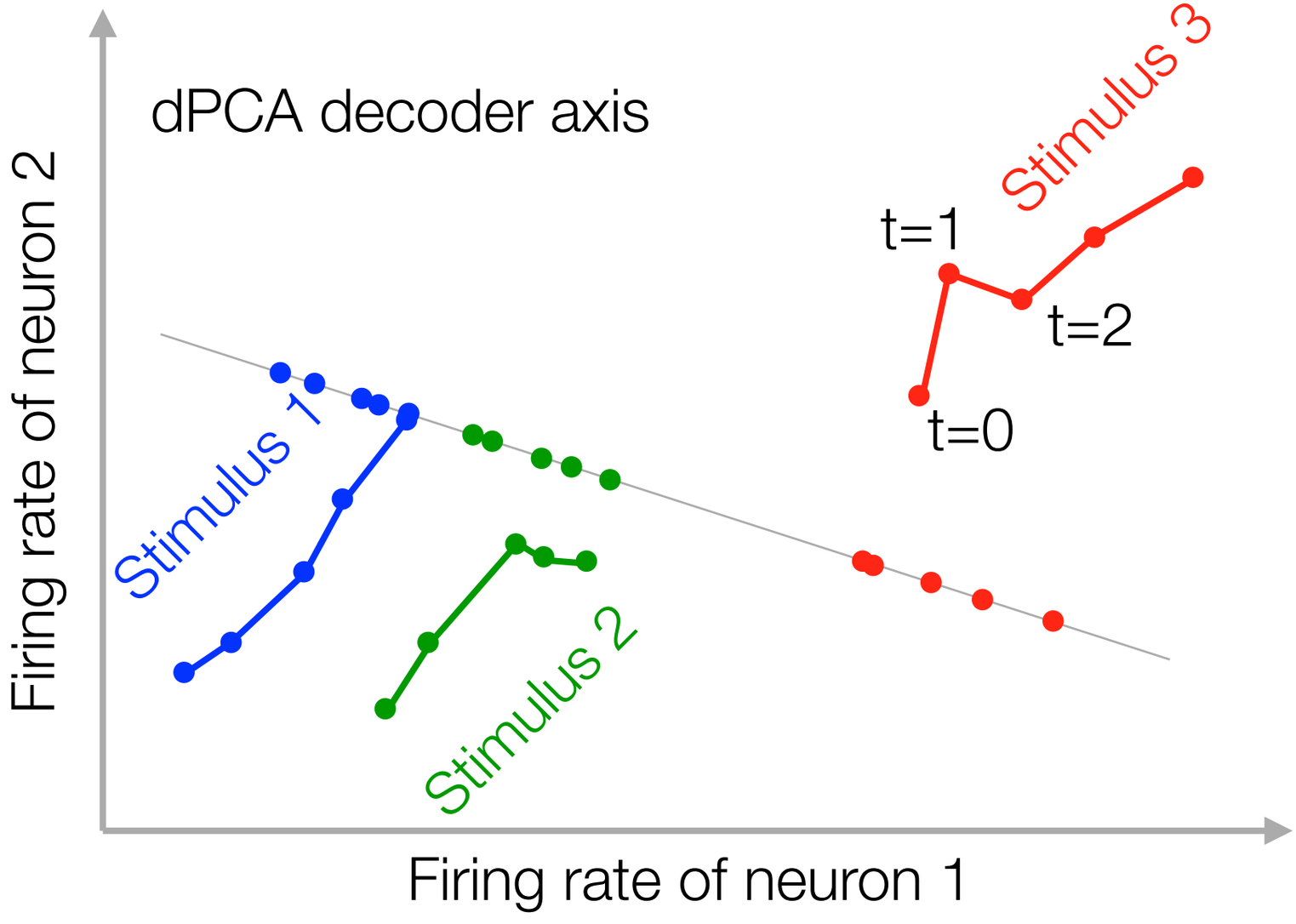
Linear readout with PCA



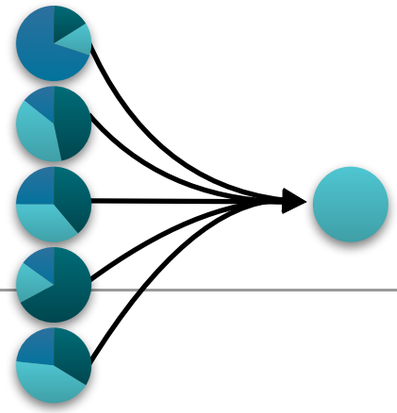


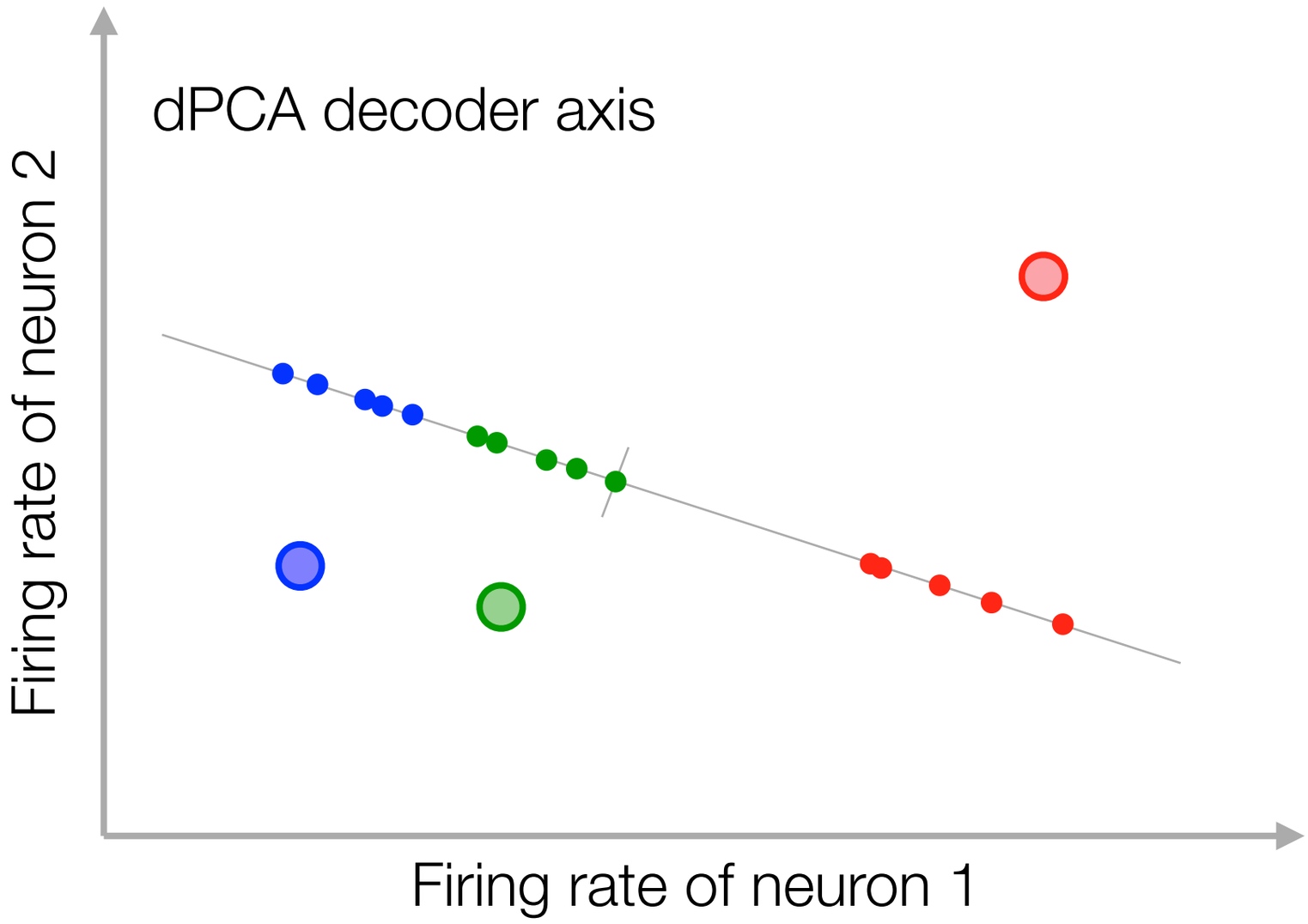
Linear readout with dPCA
‘metric-preserving decoder’



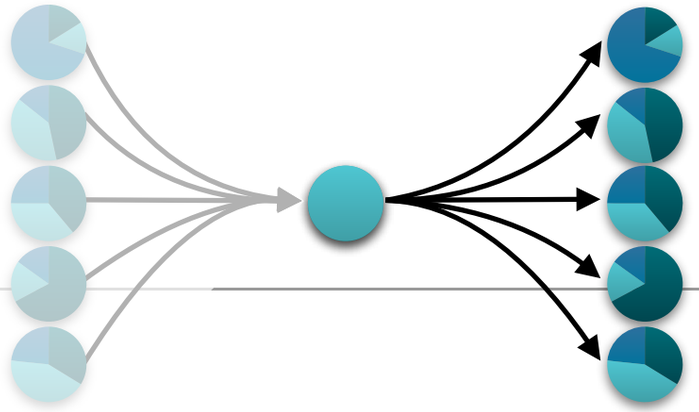


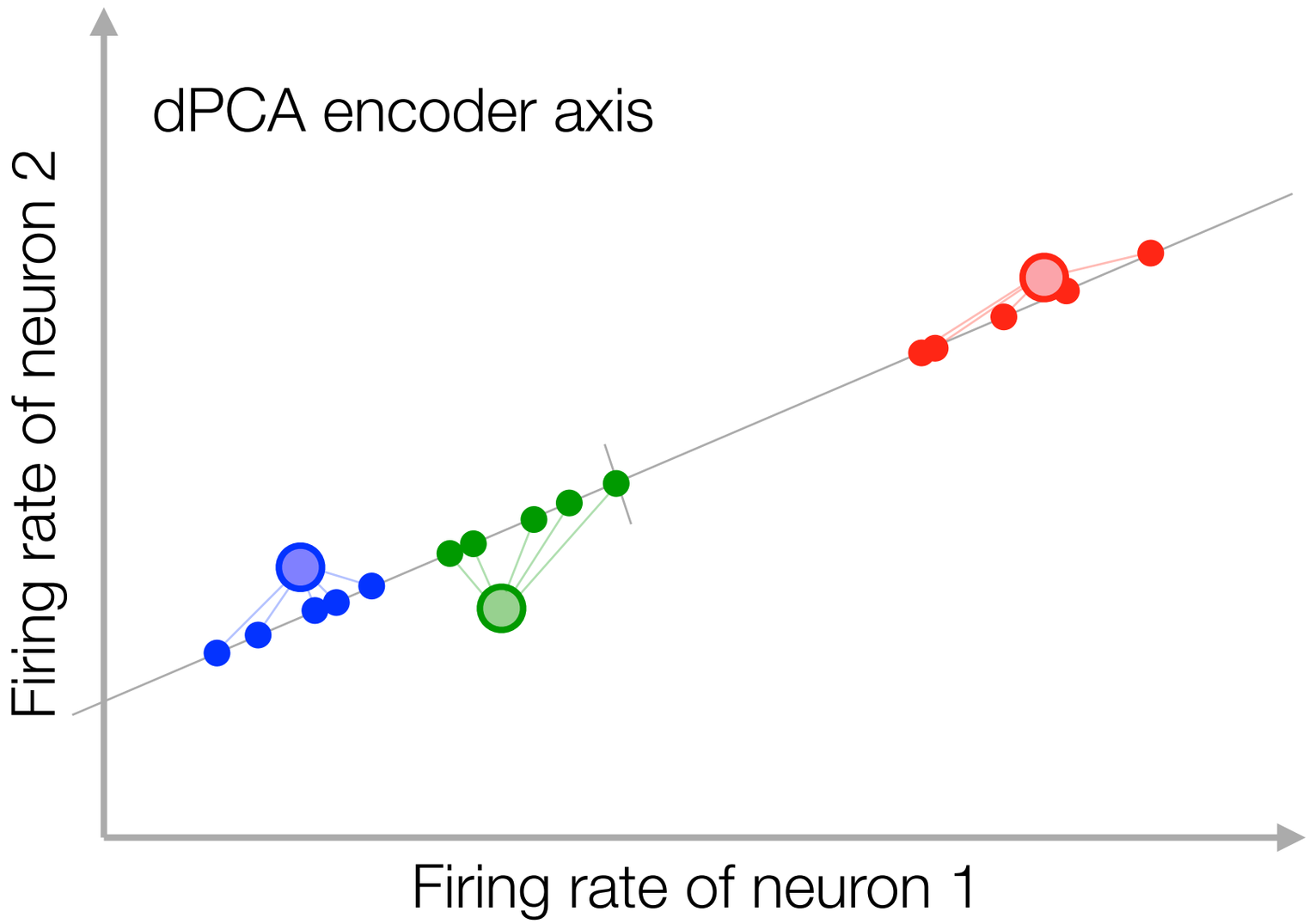
Linear readout with dPCA
‘metric-preserving decoder’



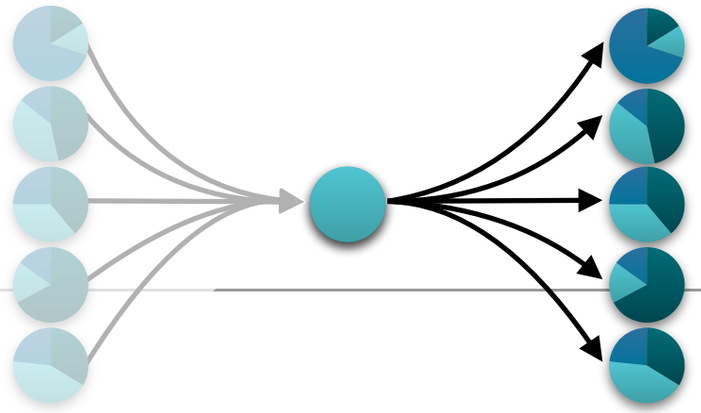


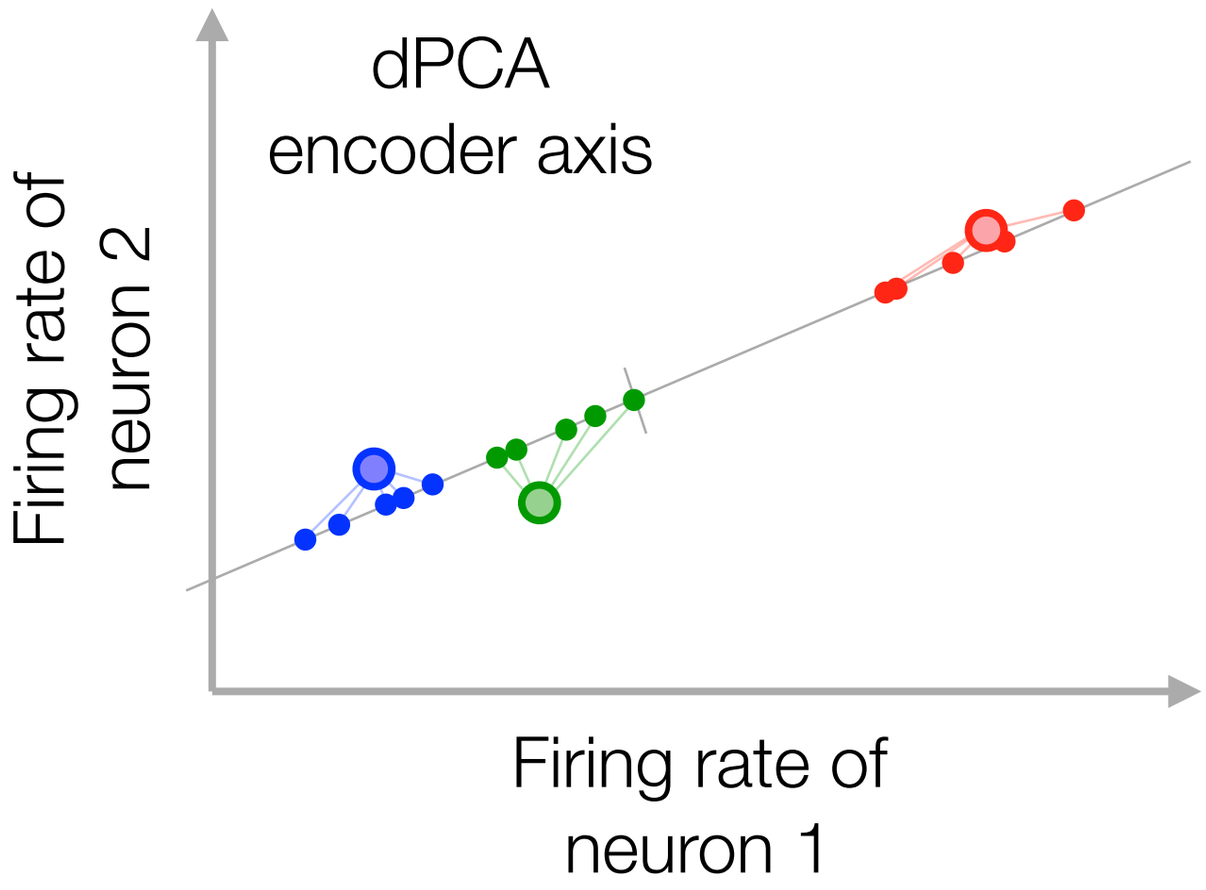
Reconstruction along different ‘encoder’ axis





Reconstruction along different ‘encoder’ axis





Loss function:

$$L = \|\mathbf{X}_s - \mathbf{FDX}\|^2$$

‘marginalized’ averages

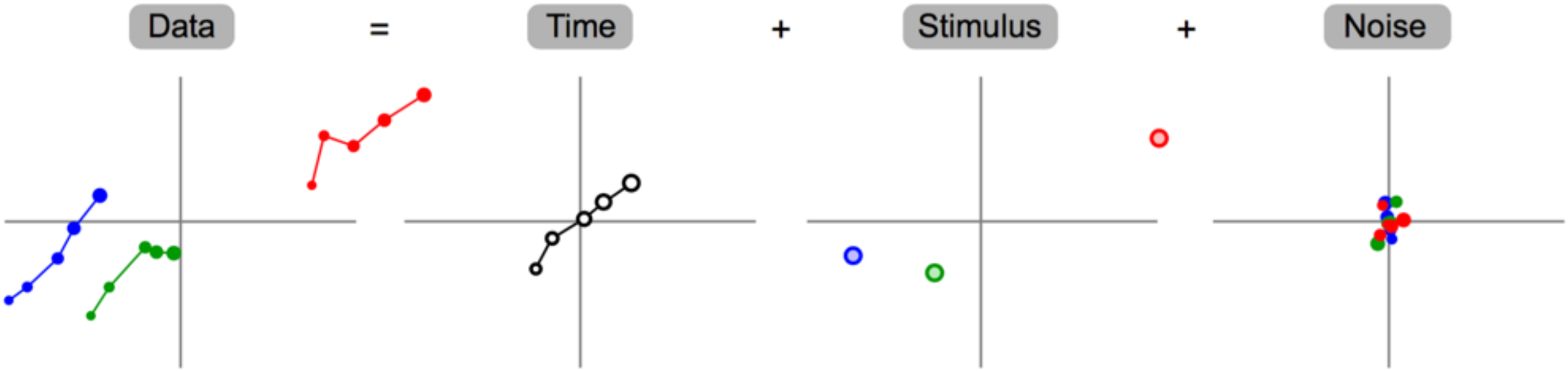
neural activities

encoder

decoder

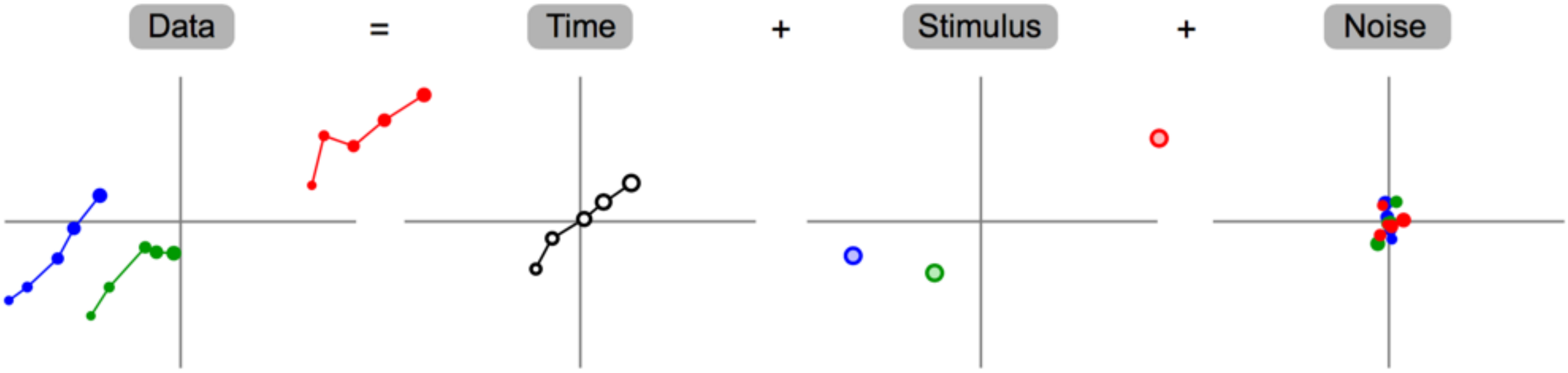
“reduced rank regression”

Data can be decomposed into ‘marginalized averages’



$$\mathbf{X} = \mathbf{X}_t + \mathbf{X}_s + \mathbf{X}_{st}$$

Separate decoder/encoder pairs for each marginalized average allow reconstruction of full data

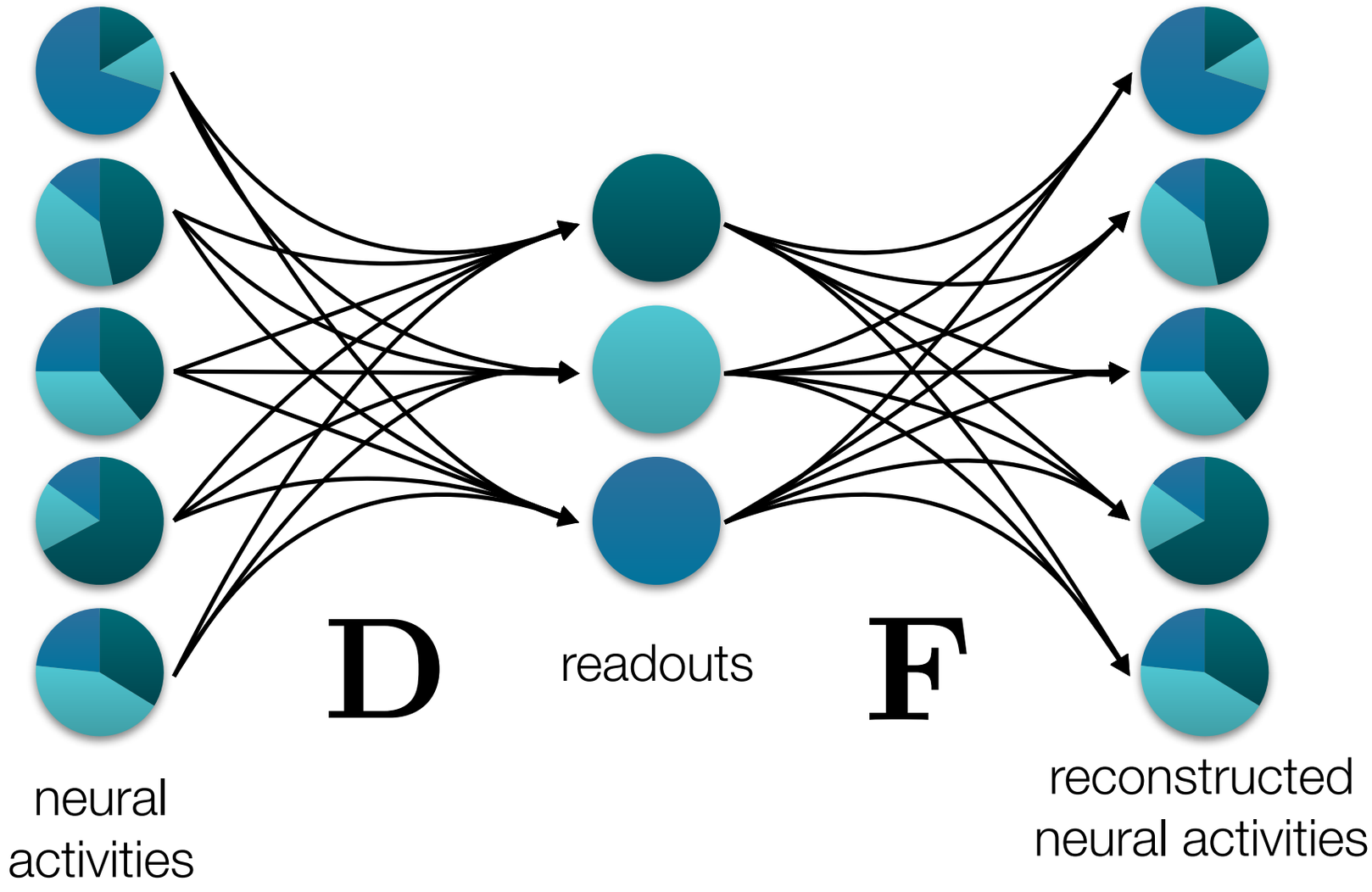


$$\mathbf{X} = \mathbf{X}_t + \mathbf{X}_s + \mathbf{X}_{st}$$

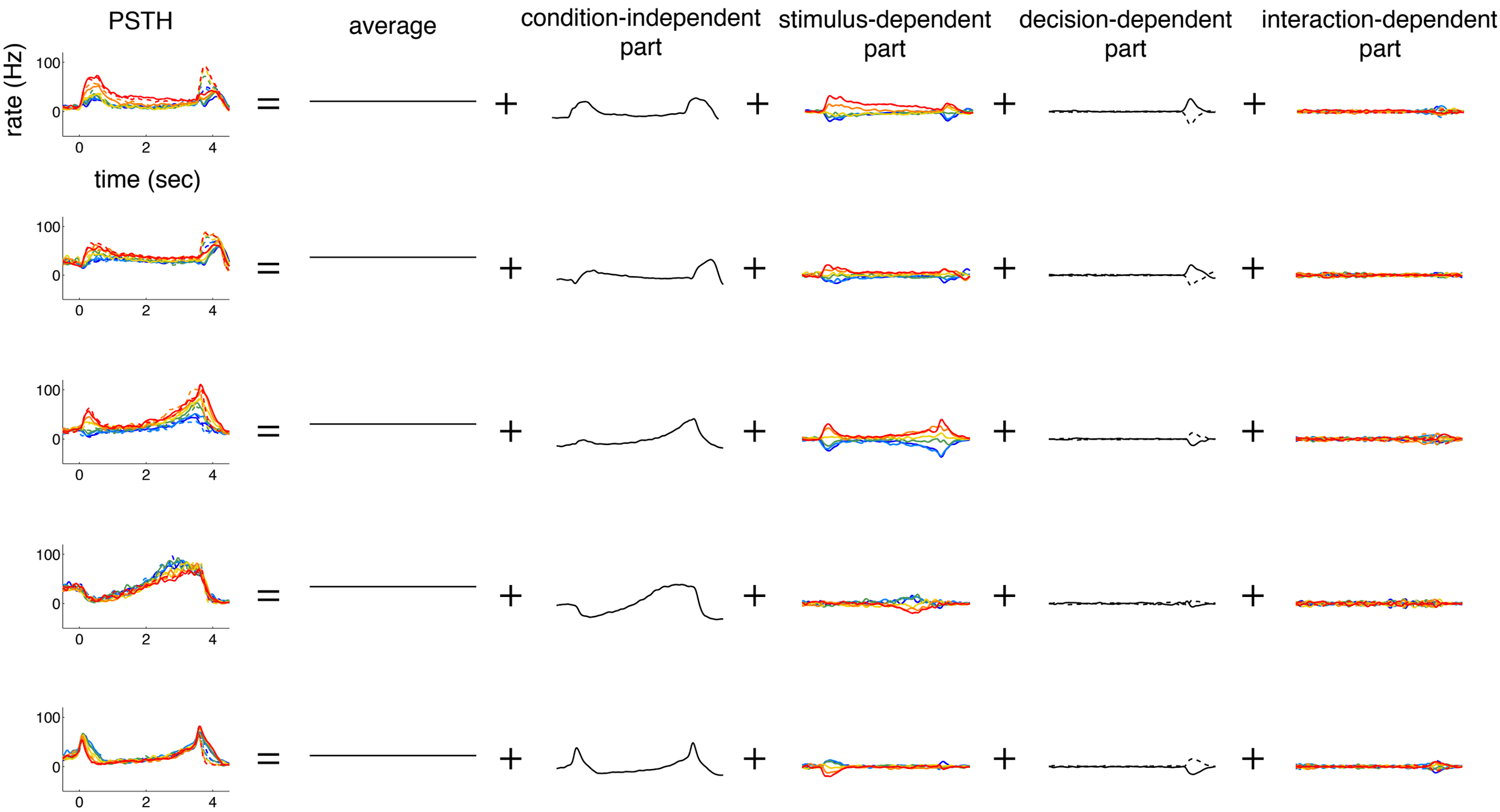
$$L = \|\mathbf{X}_t - \mathbf{F}_t\mathbf{D}_t\mathbf{X}\|^2 + \|\mathbf{X}_s - \mathbf{F}_s\mathbf{D}_s\mathbf{X}\|^2 + \|\mathbf{X}_{st} - \mathbf{F}_{st}\mathbf{D}_{st}\mathbf{X}\|^2$$

‘Demixed’ principal component analysis 2.0

(Kobak, Brendel et al, 2016, eLife)

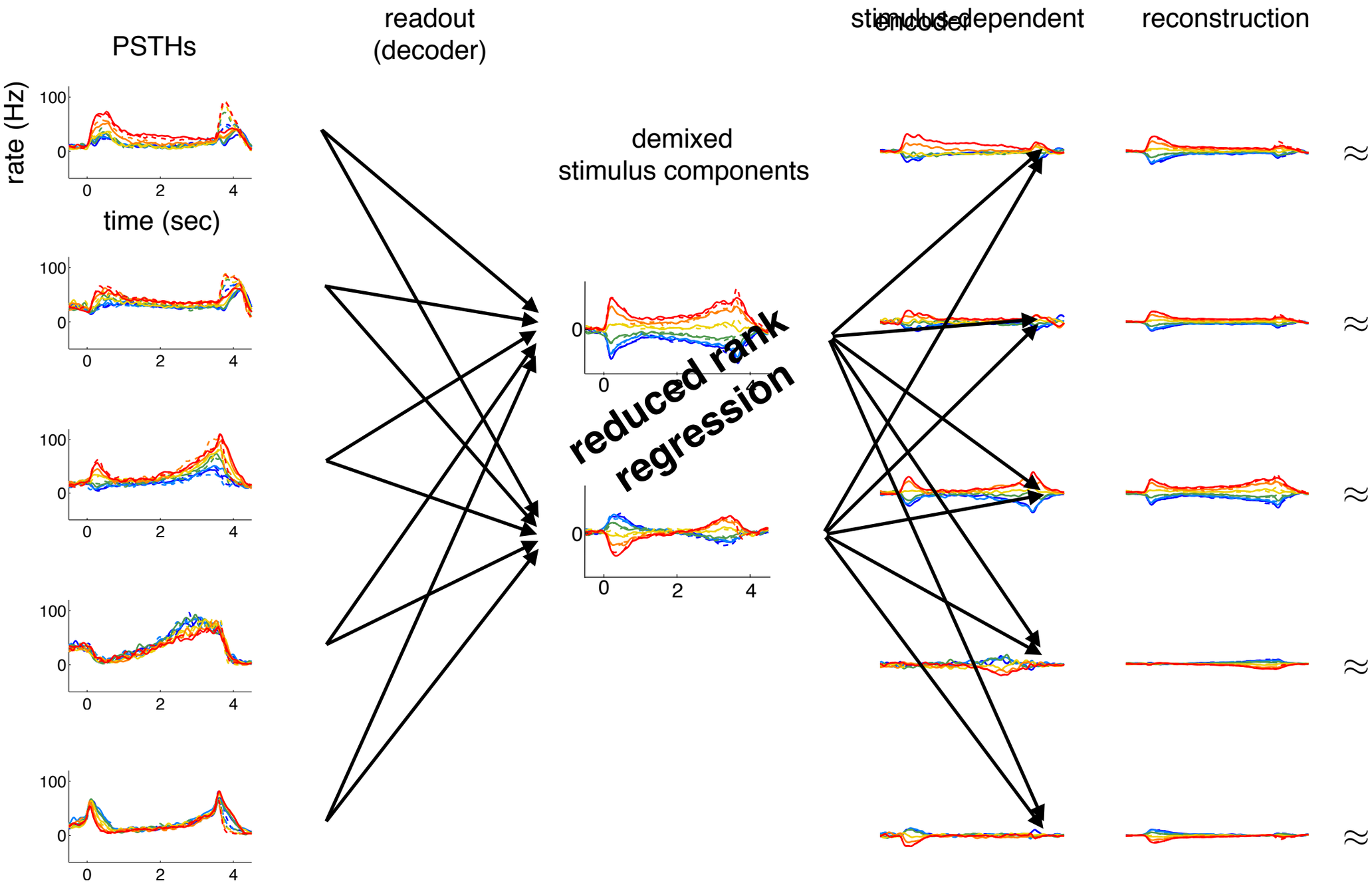


Algorithm
Step 1: decompose PSTHs into marginalized averages



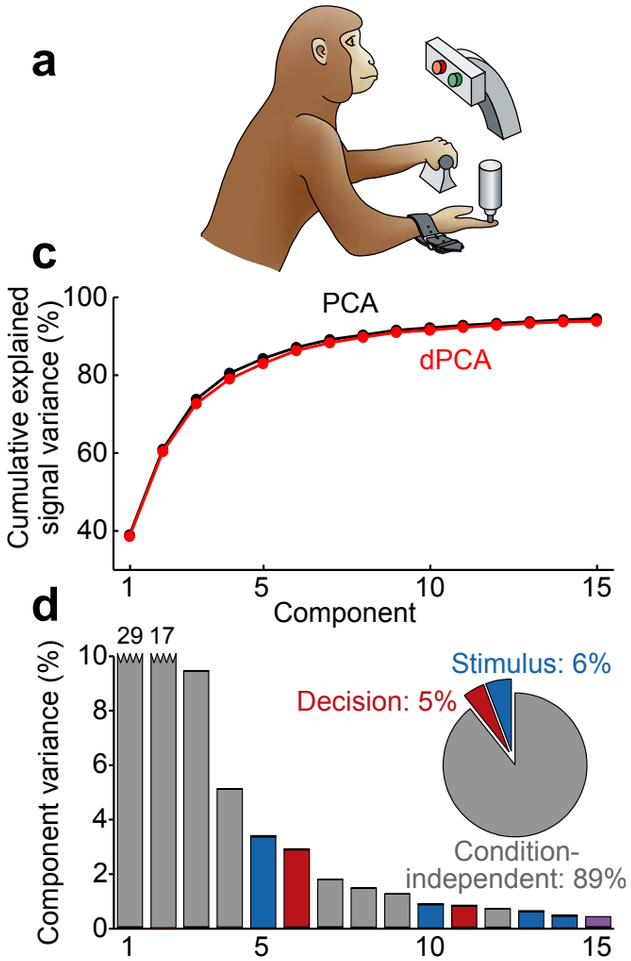
Kobak*, Brendel* et al (2016) eLife 5:e10989

Algorithm
Step 2: reduced-rank regression



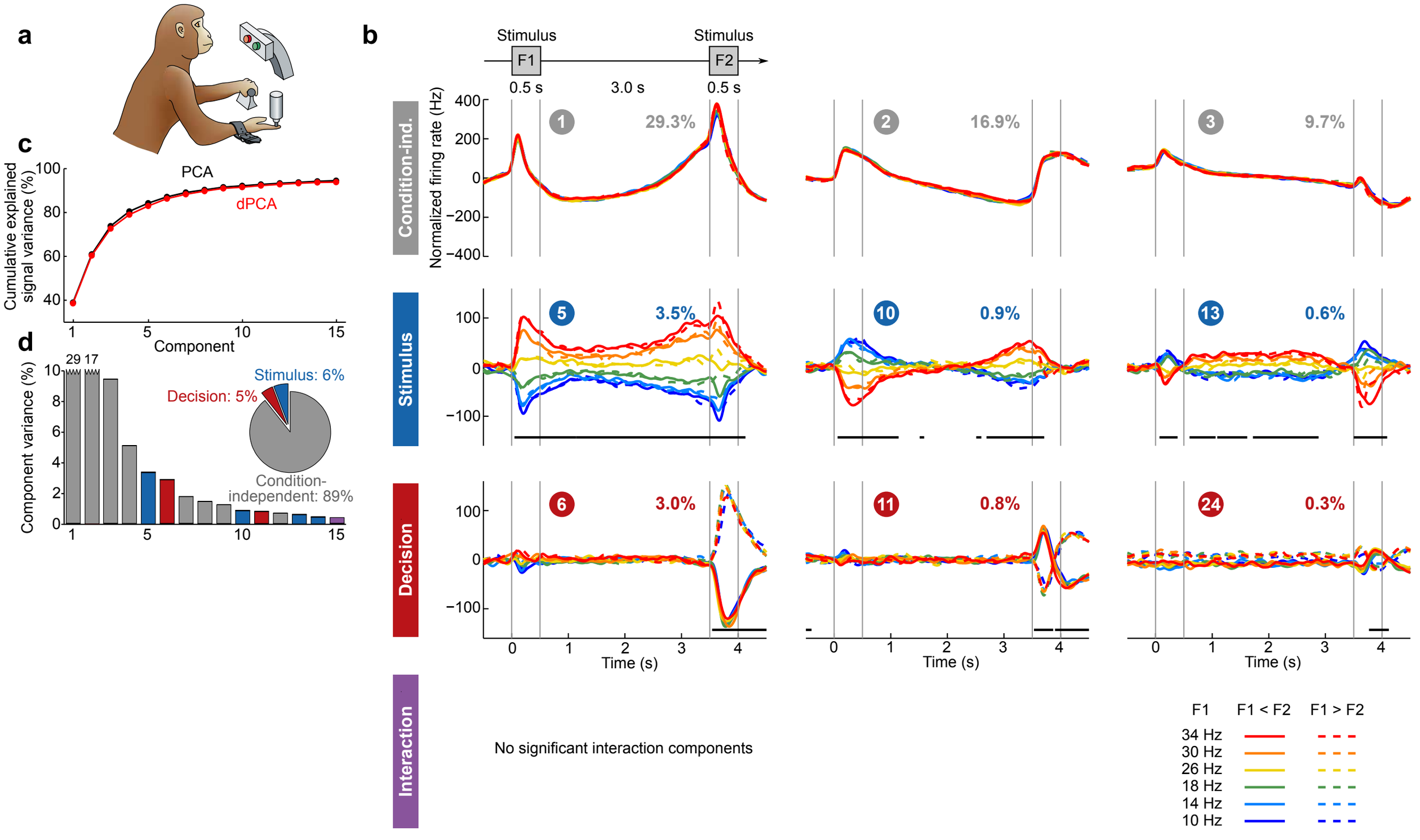
Kobak*, Brendel* et al (2016) eLife 5:e10989

Somatosensory Working Memory Task / PFC (Romo et al, Nature, 1999)



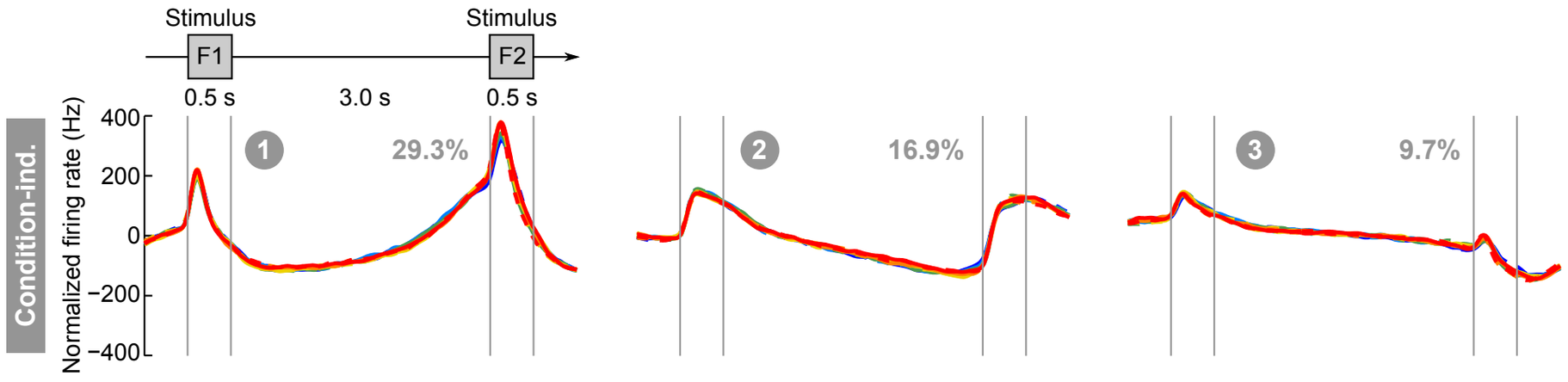
Kobak*, Brendel* et al (2016) eLife 5:e10989

Somatosensory Working Memory Task / PFC (Romo et al, Nature, 1999)

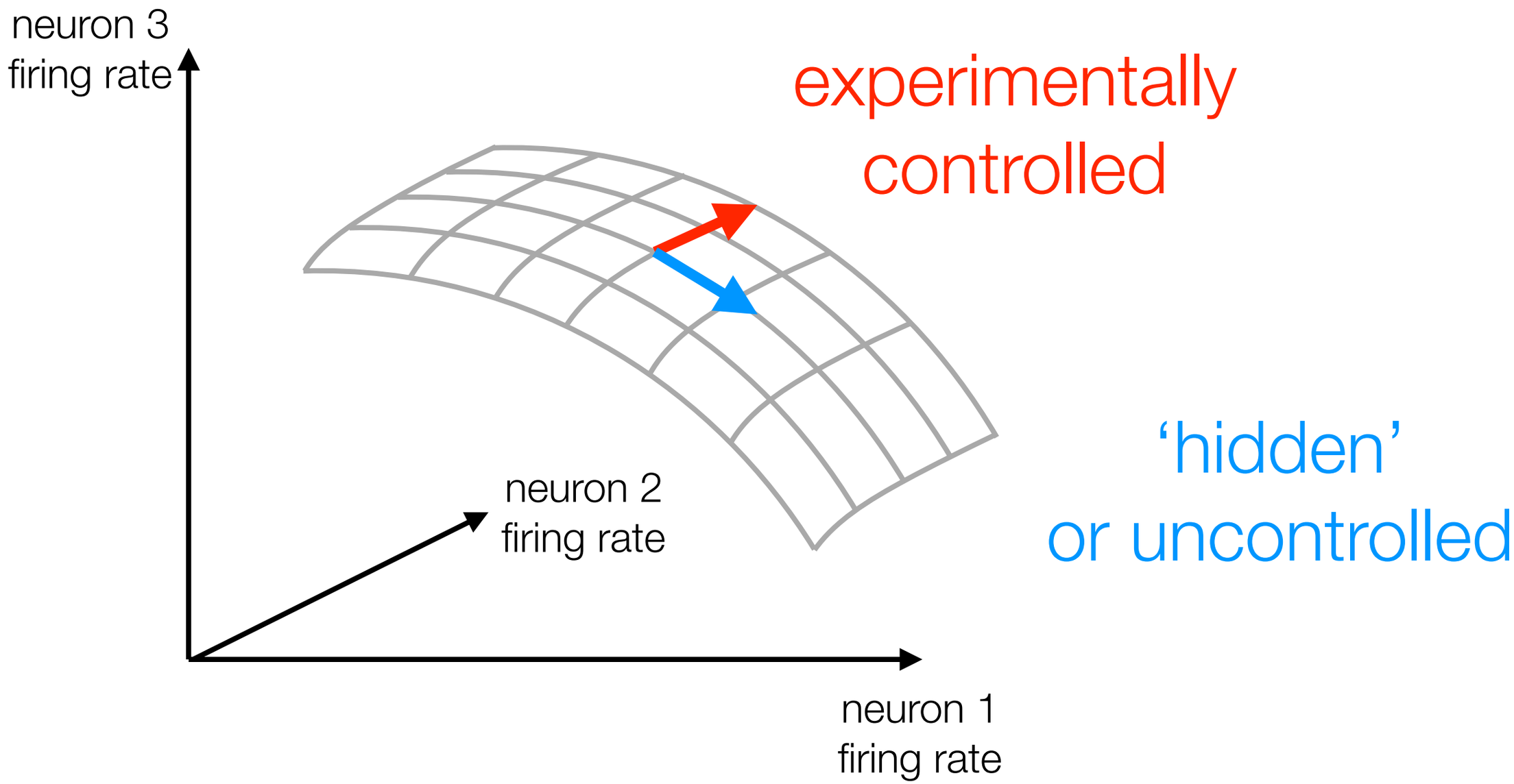


Kobak*, Brendel* et al (2016) eLife 5:e10989

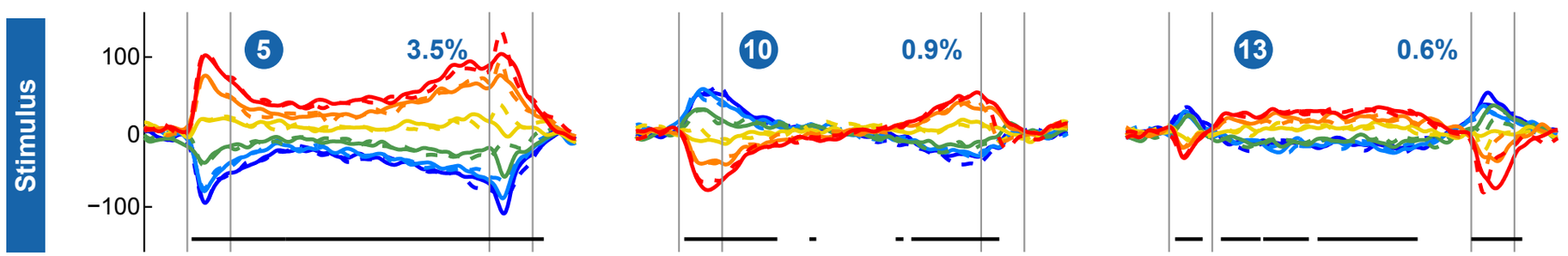
Condition-independent components:



Condition-independent components:
Only part of manifold is experimentally controlled



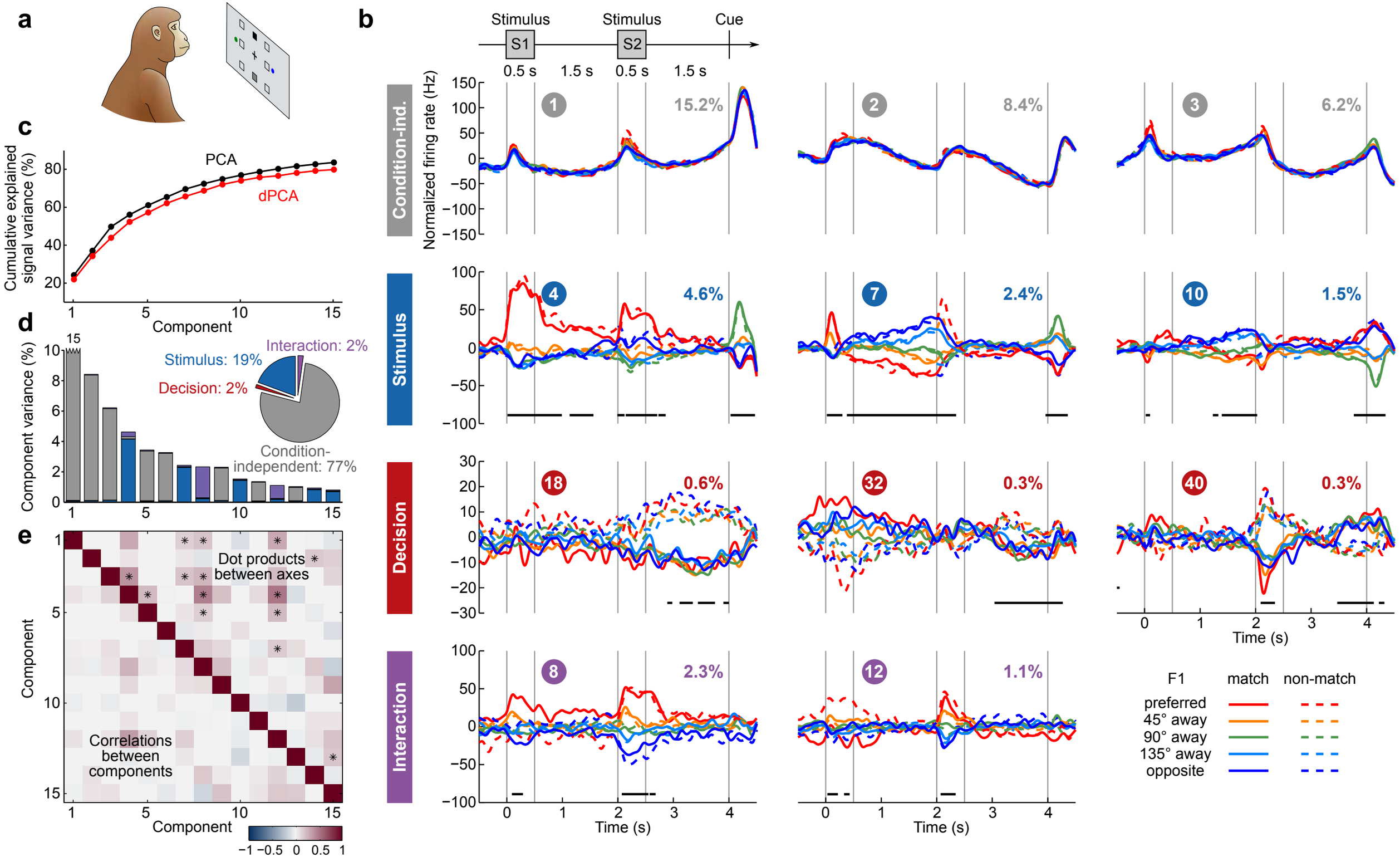
Visualization of stimulus supspace



Kobak, Brendel et al (2014) arxiv:1410.6031

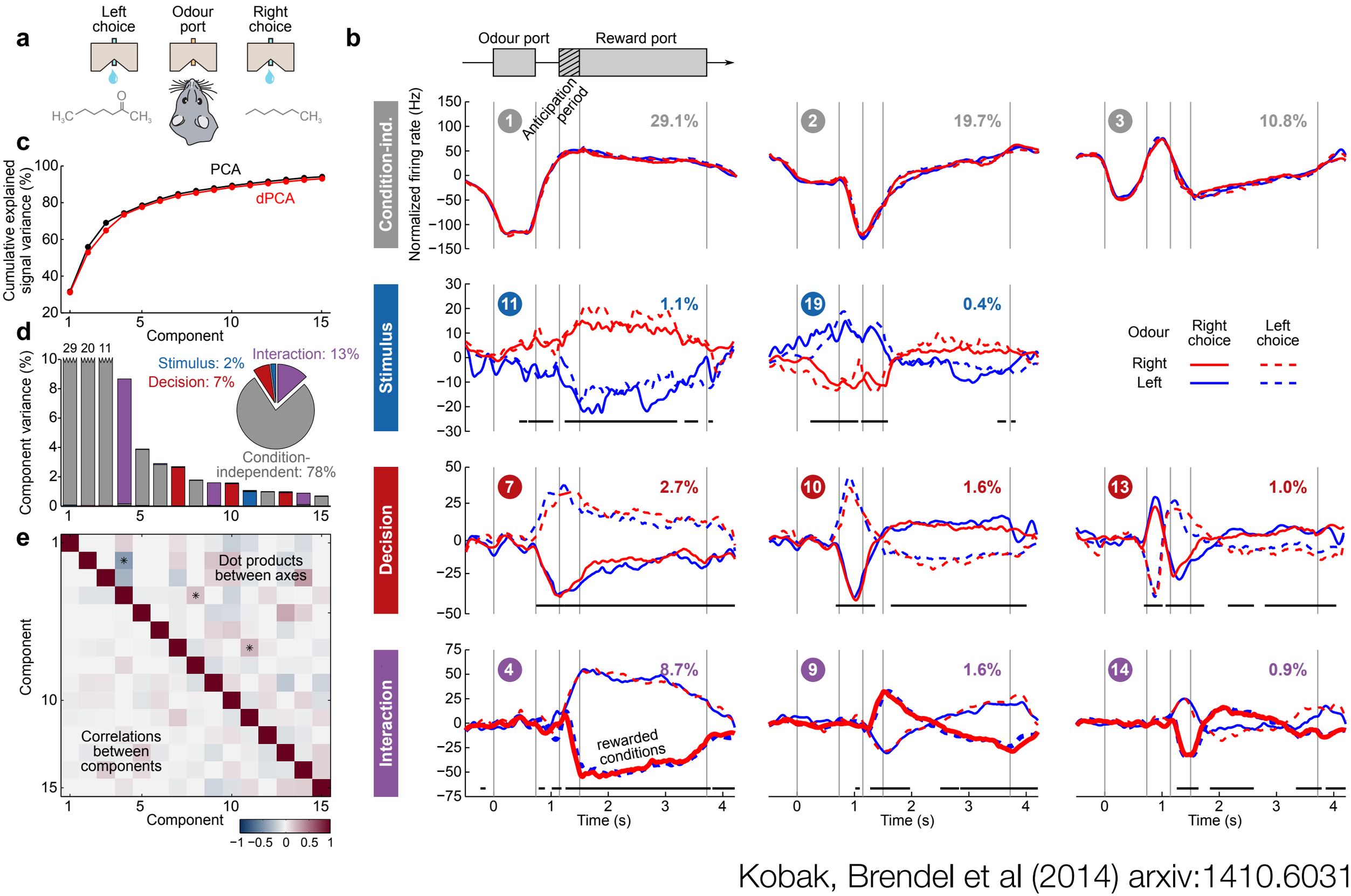
Visualization of stimulus supspace

Visuospatial Working Memory Task
(Qi et al, J Neurosci, 2010)



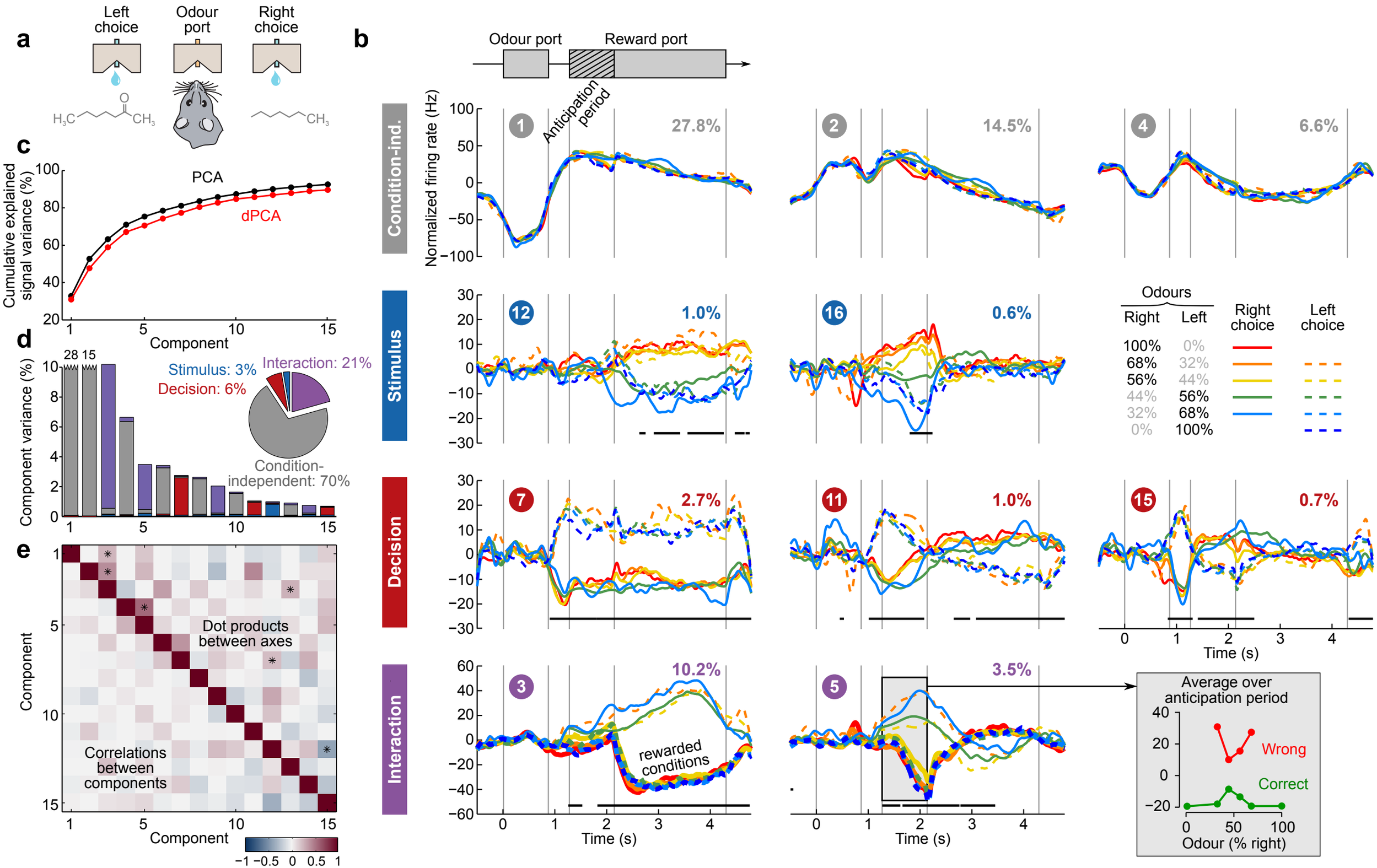
Kobak, Brendel et al (2014) arxiv:1410.6031

Olfactory Discrimination Task (Feierstein et al, Neuron, 2006)



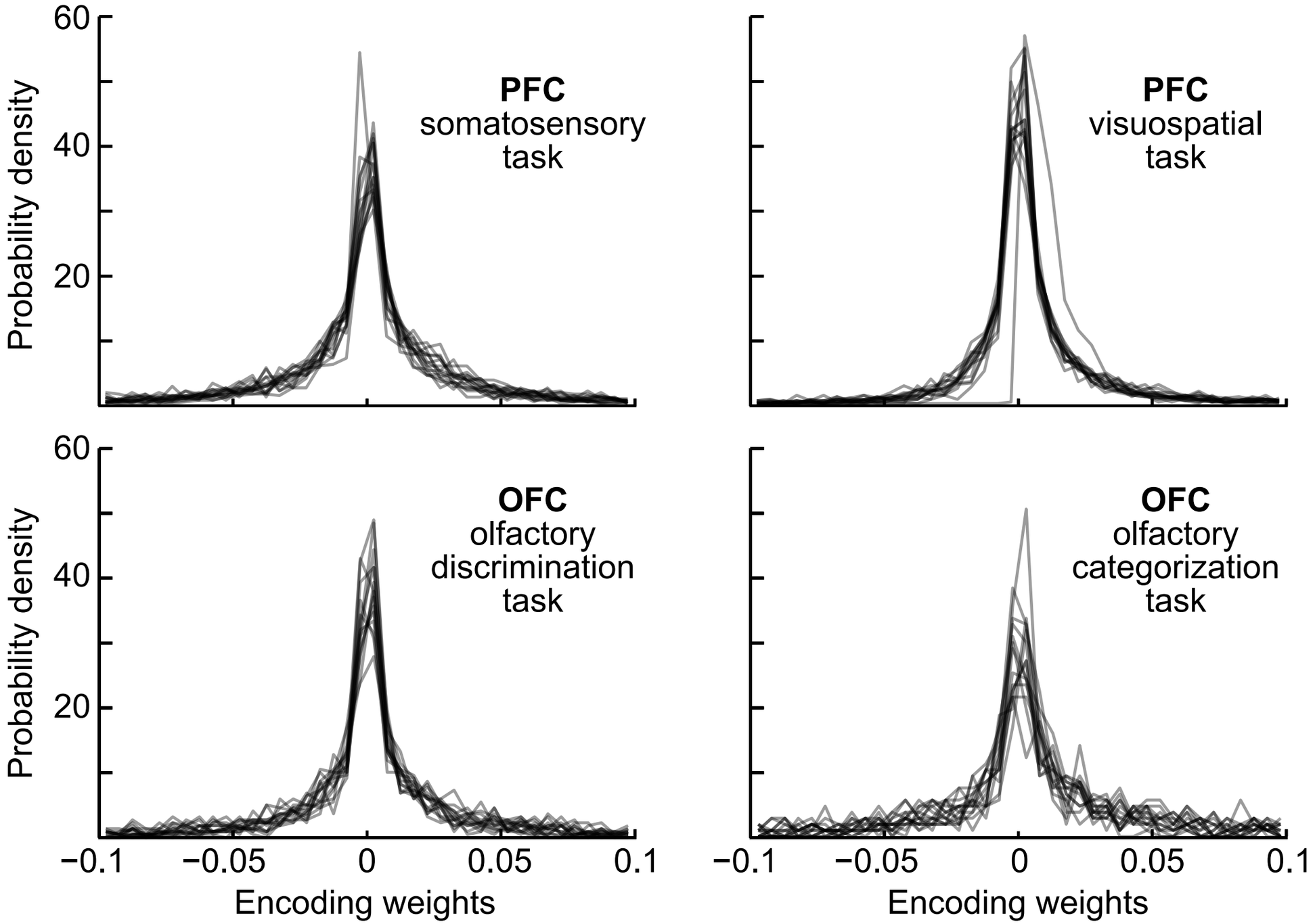
Kobak, Brendel et al (2014) arxiv:1410.6031

Olfactory Categorization Task (Kepecs et al, Nature, 2008)



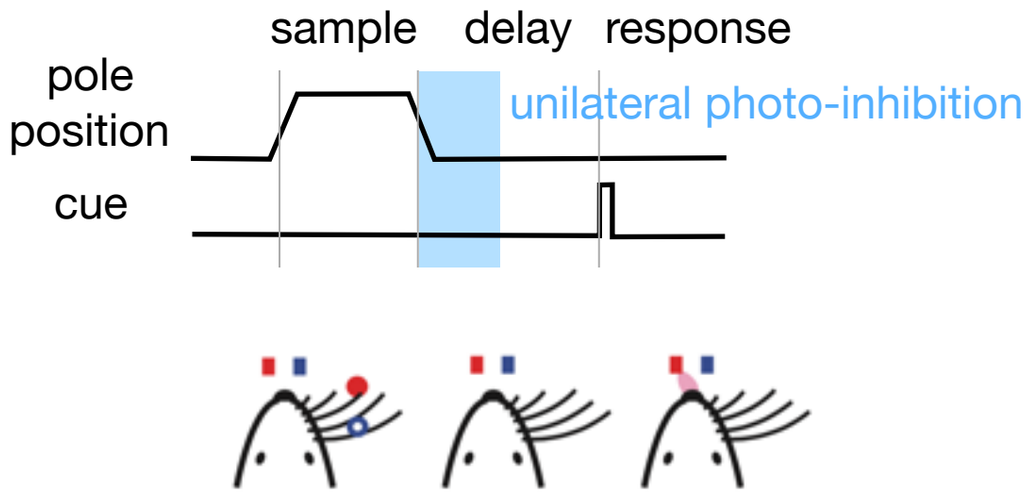
Kobak, Brendel et al (2014) arxiv:1410.6031

Distribution of encoding weights

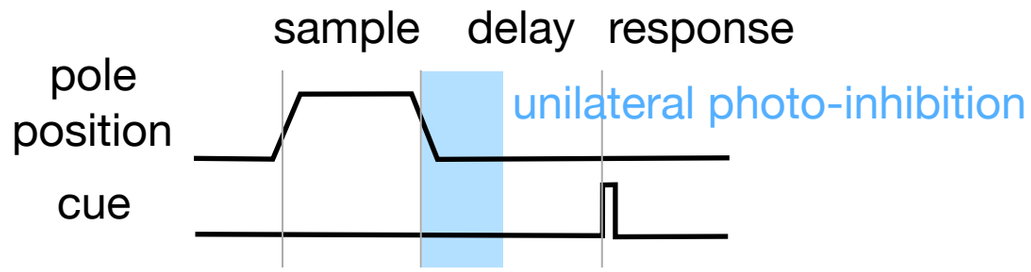


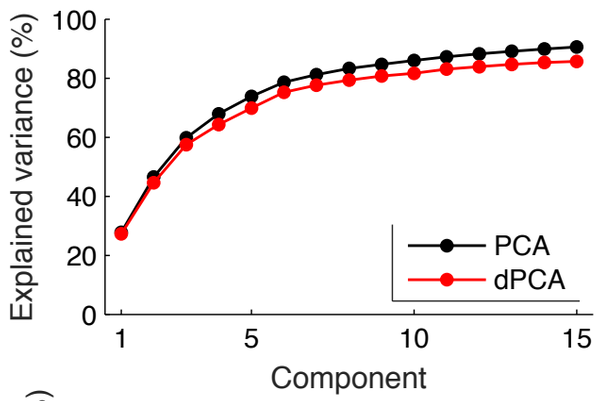
Kobak, Brendel et al (2014) arxiv:1410.6031

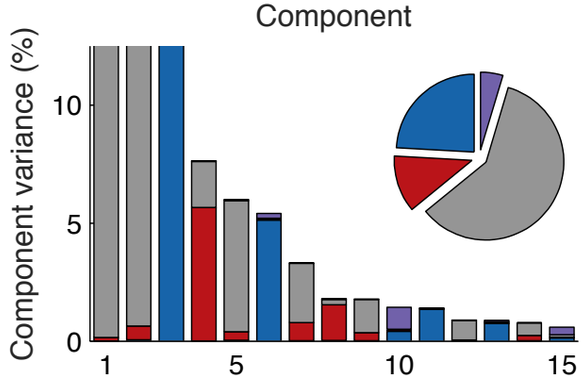
Whisker Detection Task / ALM (Li, Daie et al, Nature, 2016)



Whisker Detection Task / ALM (Li, Daie et al, Nature, 2016)







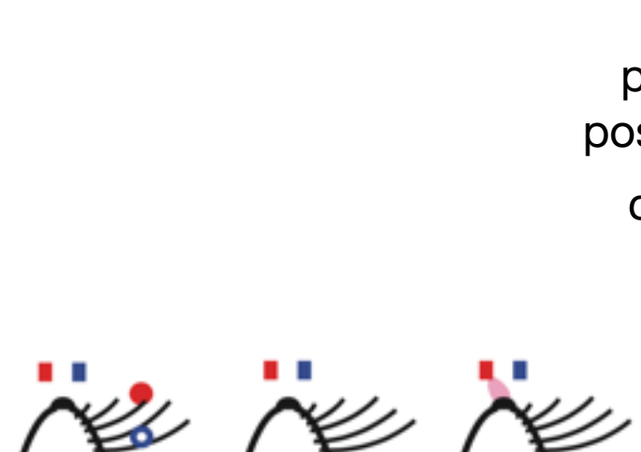
Perturbation

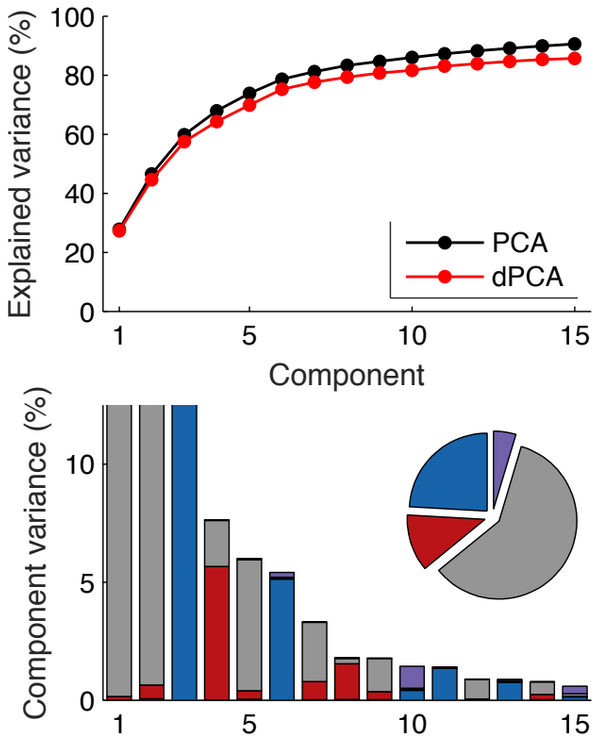
Decision

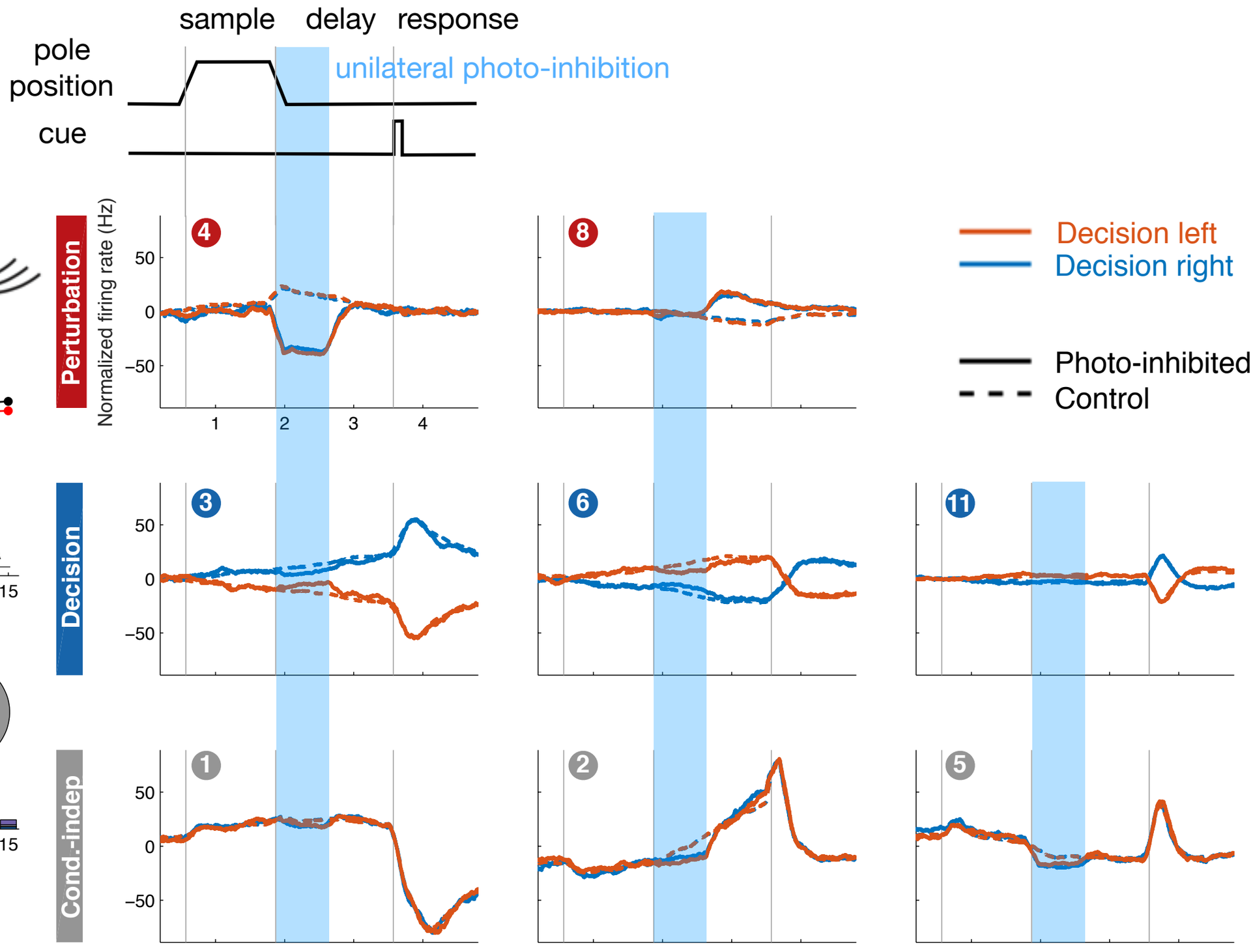
Cond.-indep.

data courtesy of Nuo Li*, Kayvon Daie*, Karel Svoboda, Shaul Druckmann

Whisker Detection Task / ALM
(Li, Daie et al, Nature, 2016)







data courtesy of Nuo Li*, Kayvon Daie*, Karel Svoboda, Shaul Druckmann

http://github.com/machenslab/dPCA

Conclusion

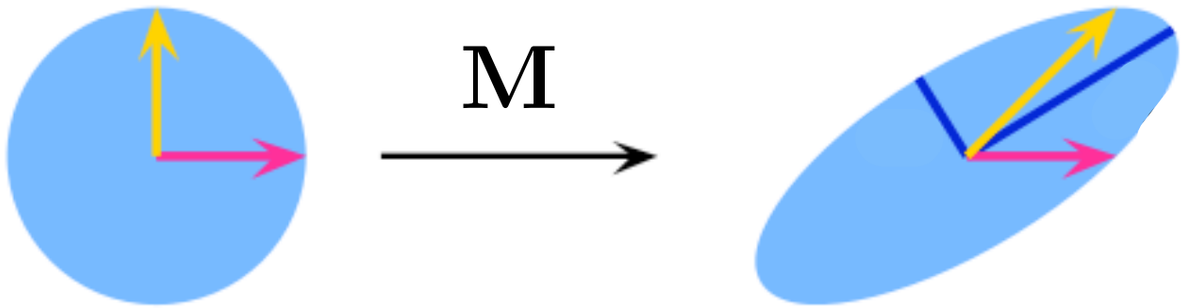
- Our best guess for population representations are linear readouts from the population.
- Demixed PCA seeks a set of decoders that (1) provide readouts for individual task parameters (2) without losing essential features of the data
- Demixed PCA rests on a decomposition of the data into ‘marginalized averages’, each of which is fitted to the actual data using reduced-rank regression
- Application of the method can highlight similarities of information presentation across tasks and cortical areas or animal species.

Singular Value Decomposition

Every matrix **M** can be uniquely 'decomposed' into an orthogonal matrix **U**, a diagonal matrix **S** and an orthogonal matrix **V**:

$$\mathbf{M} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$$

default convention sets a transpose here

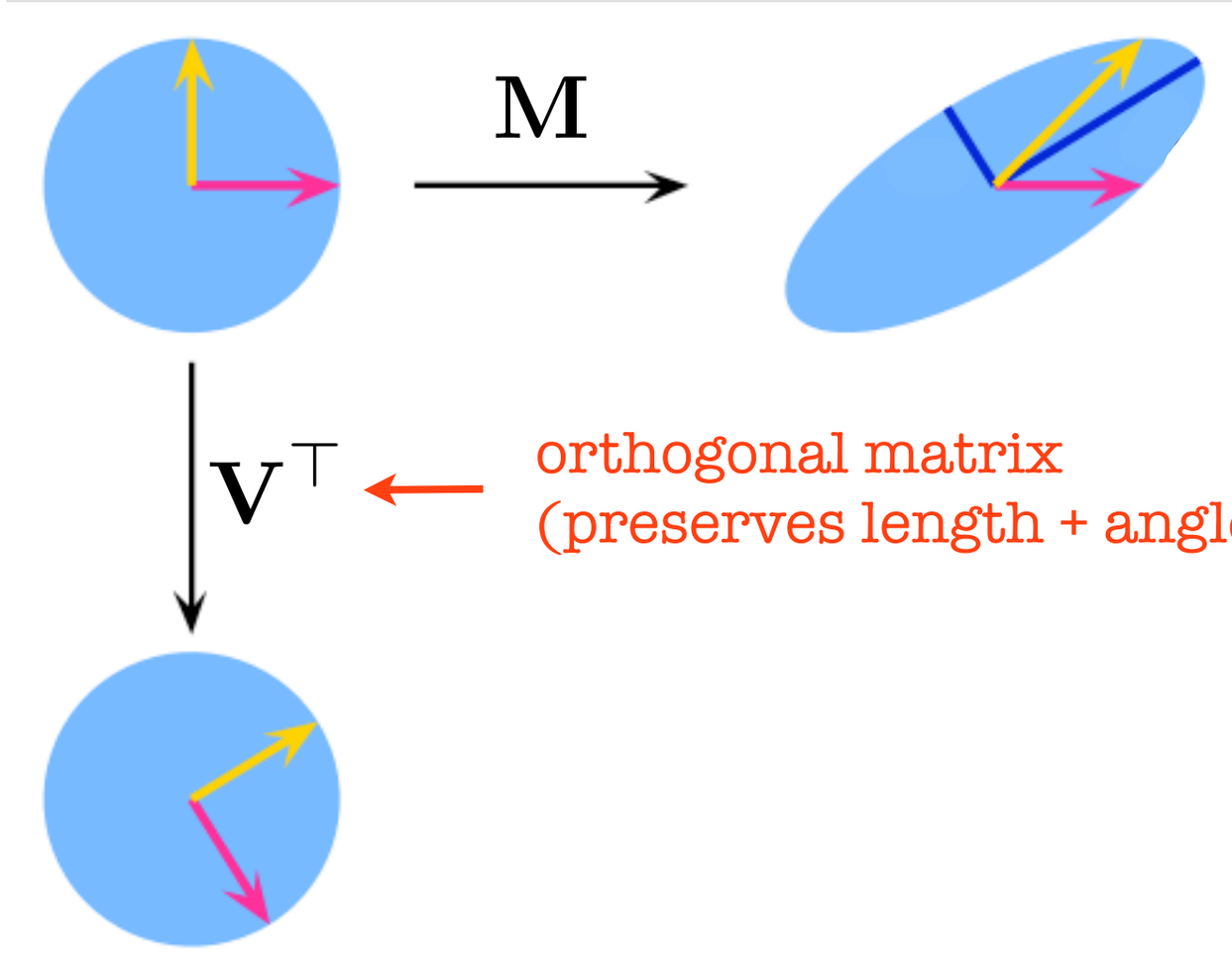


Singular Value Decomposition

Every matrix **M** can be uniquely 'decomposed' into an orthogonal matrix **U**, a diagonal matrix **S** and an orthogonal matrix **V**:

$$\mathbf{M} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$$

default convention sets a transpose here

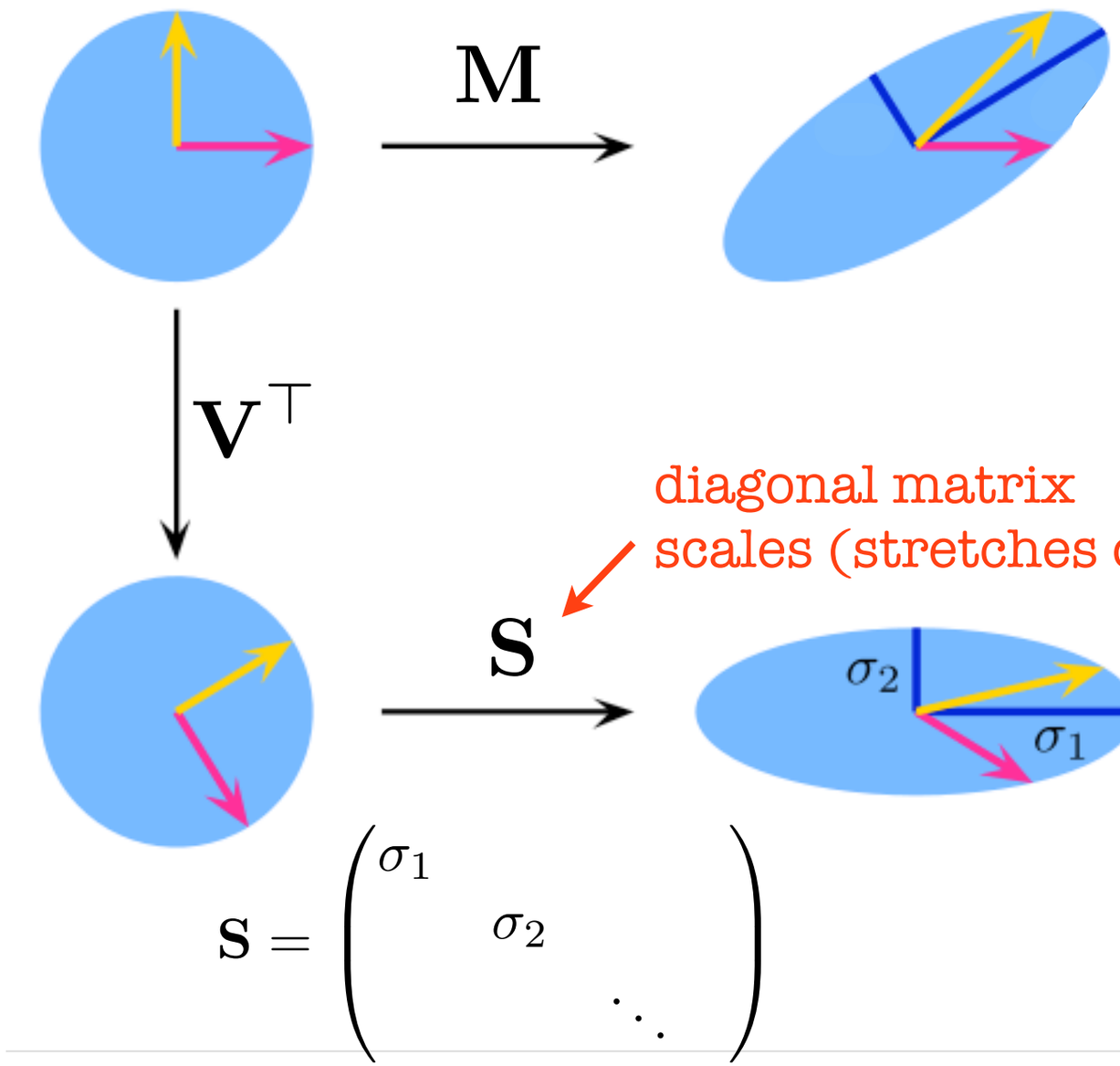


Singular Value Decomposition

Every matrix **M** can be uniquely 'decomposed' into an orthogonal matrix **U**, a diagonal matrix **S** and an orthogonal matrix **V**:

$$\mathbf{M} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$$

default convention sets a transpose here

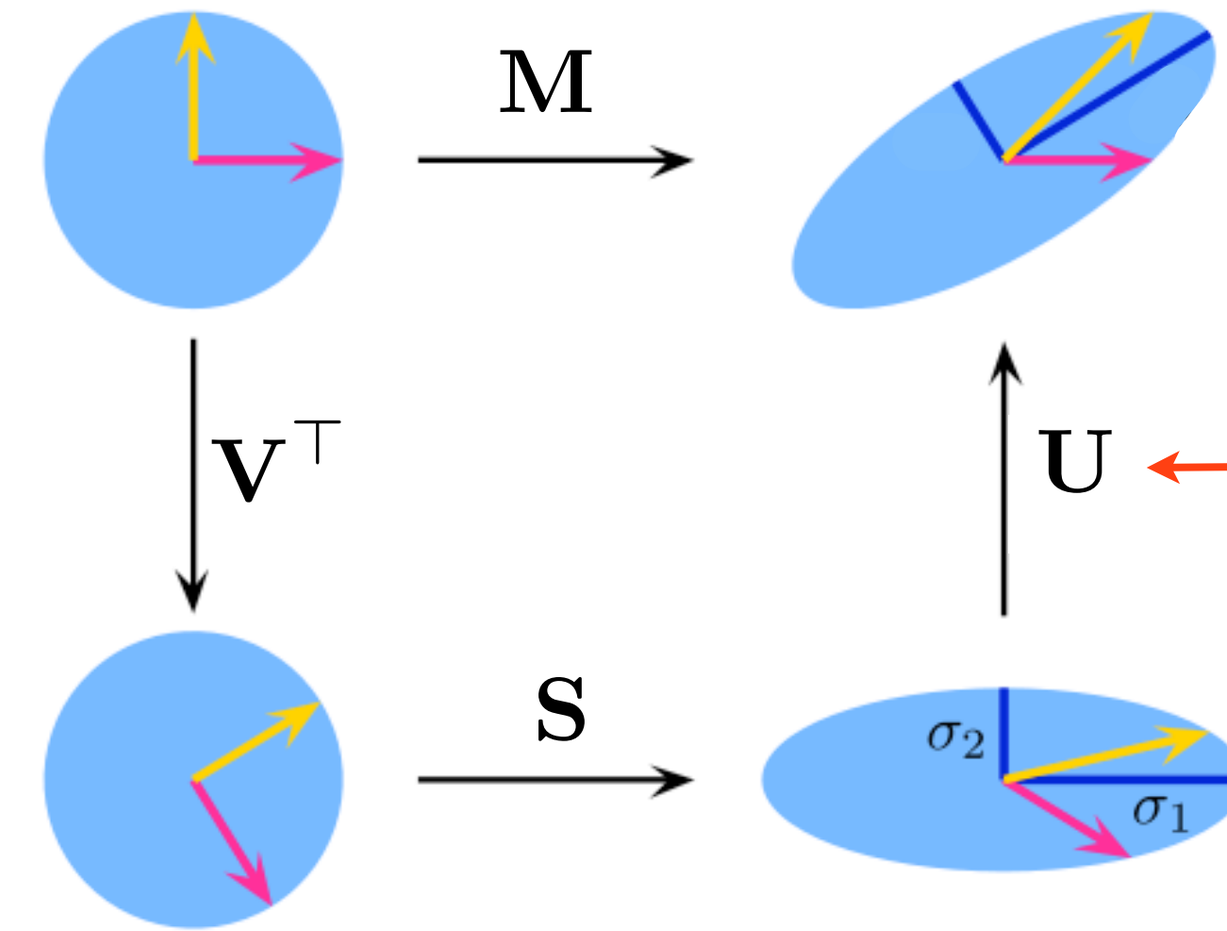


Singular Value Decomposition

Every matrix **M** can be uniquely 'decomposed' into an orthogonal matrix **U**, a diagonal matrix **S** and an orthogonal matrix **V**:

$$\mathbf{M} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$$

default convention sets a transpose here



orthogonal matrix (preserves length + angle)

see also: en.wikipedia.org/wiki/Singular_value_decomposition