

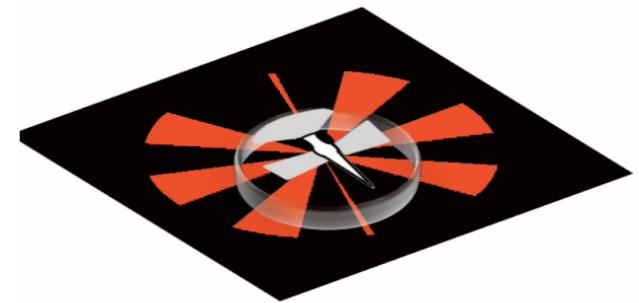
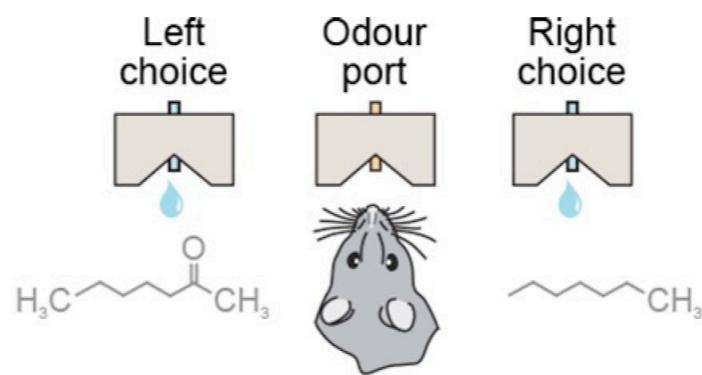
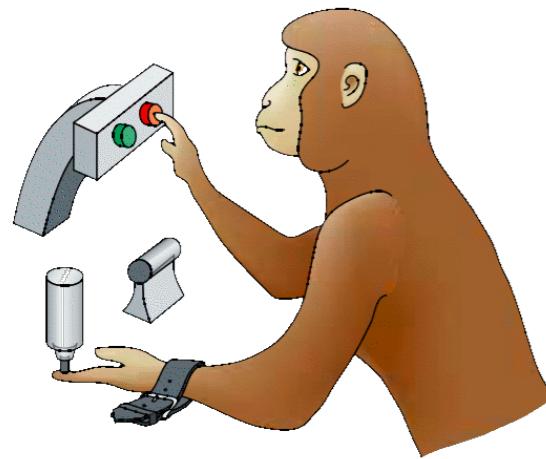
Population coding and distributed representations in the brain

Christian Machens

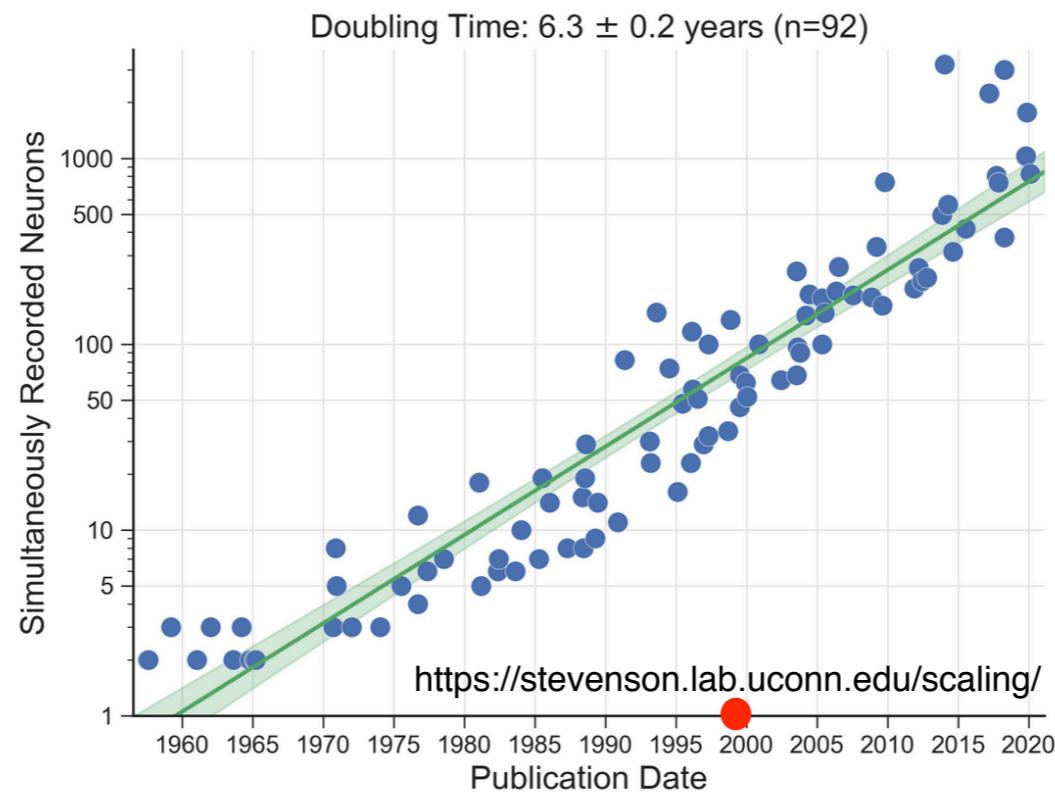
Champalimaud Centre for the Unknown, Lisbon, Portugal



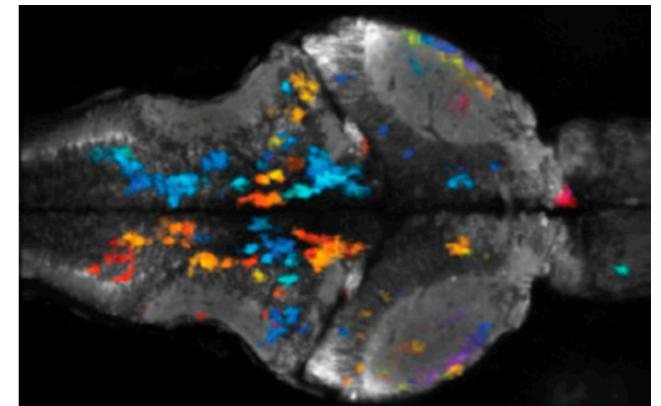
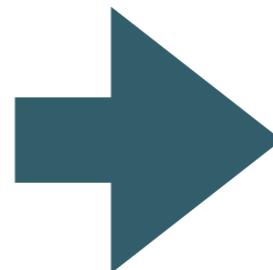
Animal behaviour: simple, lab-controlled tasks



Recording techniques: more and more cells

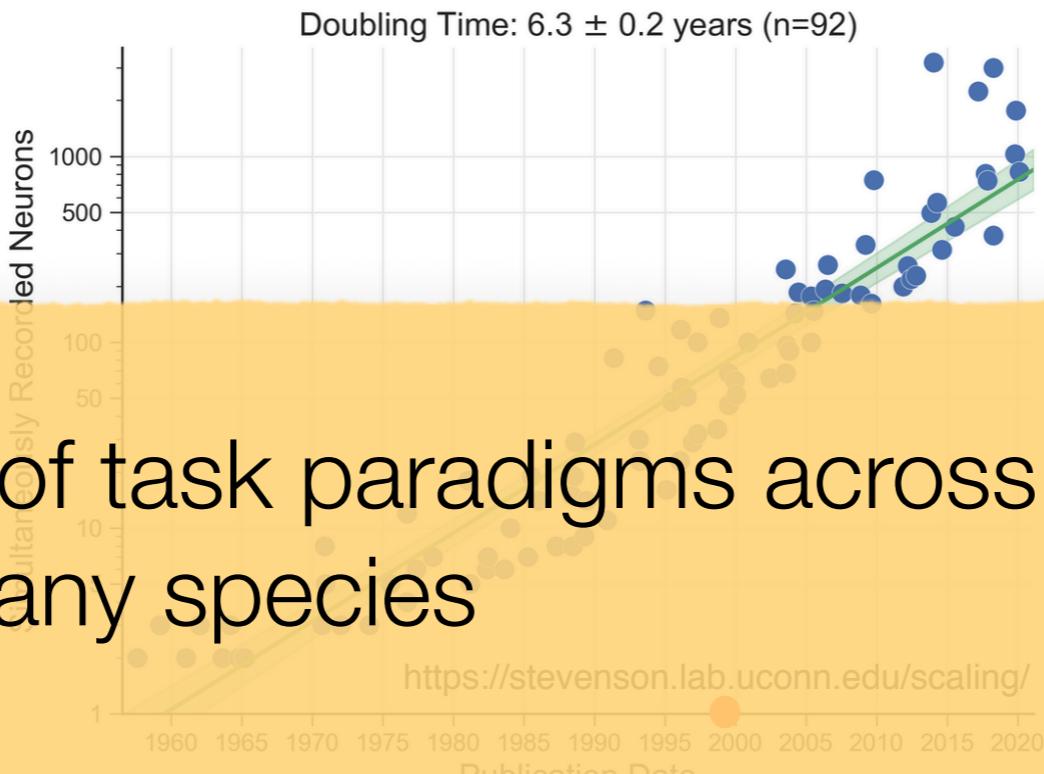


Single-cell
recording



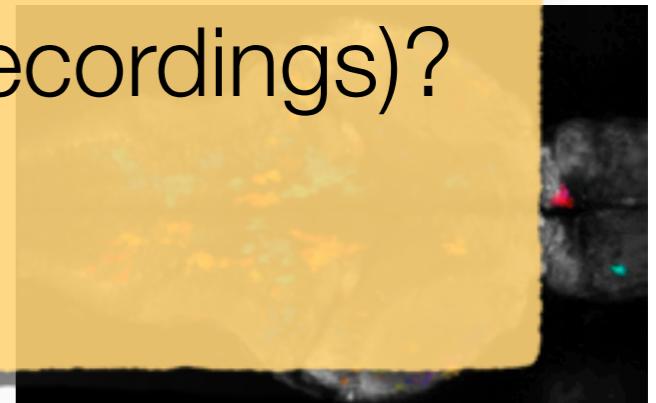
Whole brain
recording

Recording techniques: more and more cells



Hundreds of task paradigms across many brain areas in many species

How do we ‘look’ at the data (neural recordings)?
What did we learn?



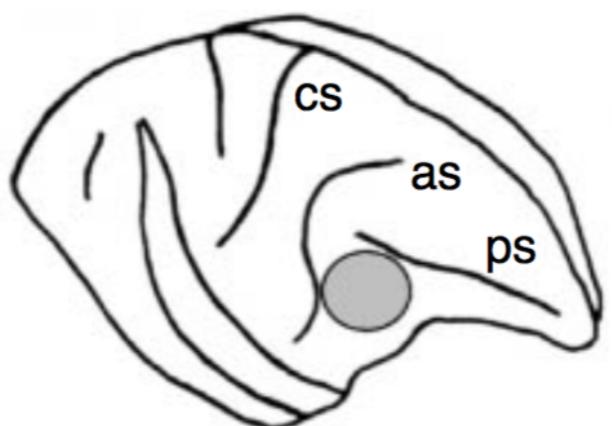
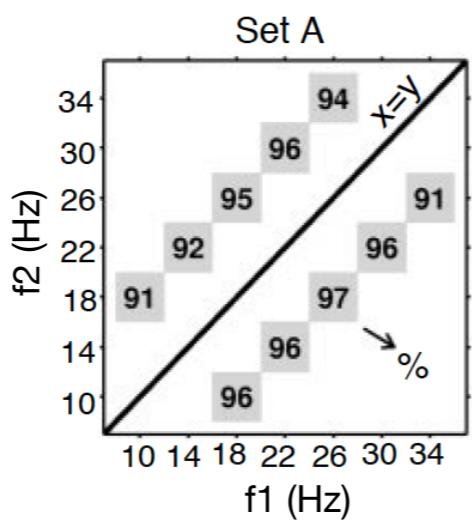
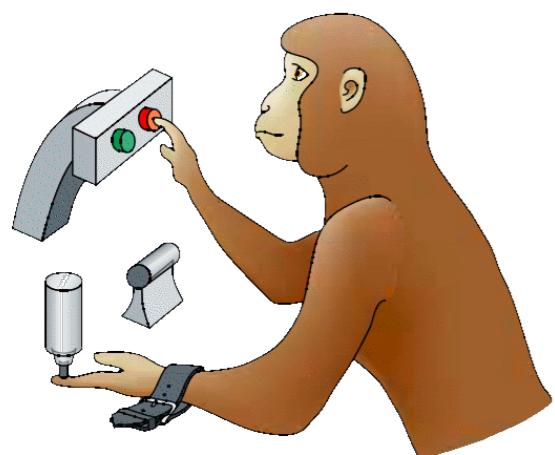
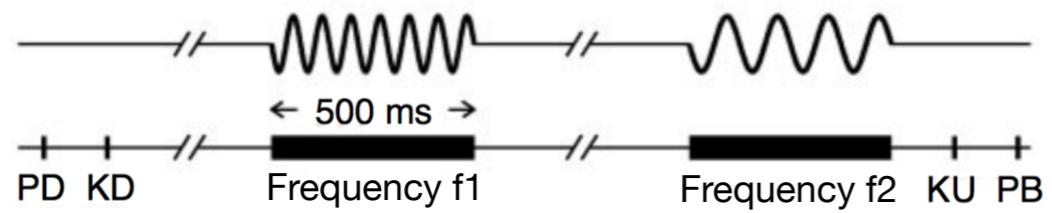
Single-cell recording

Whole brain recording

How to ‘look’ at the data (1)

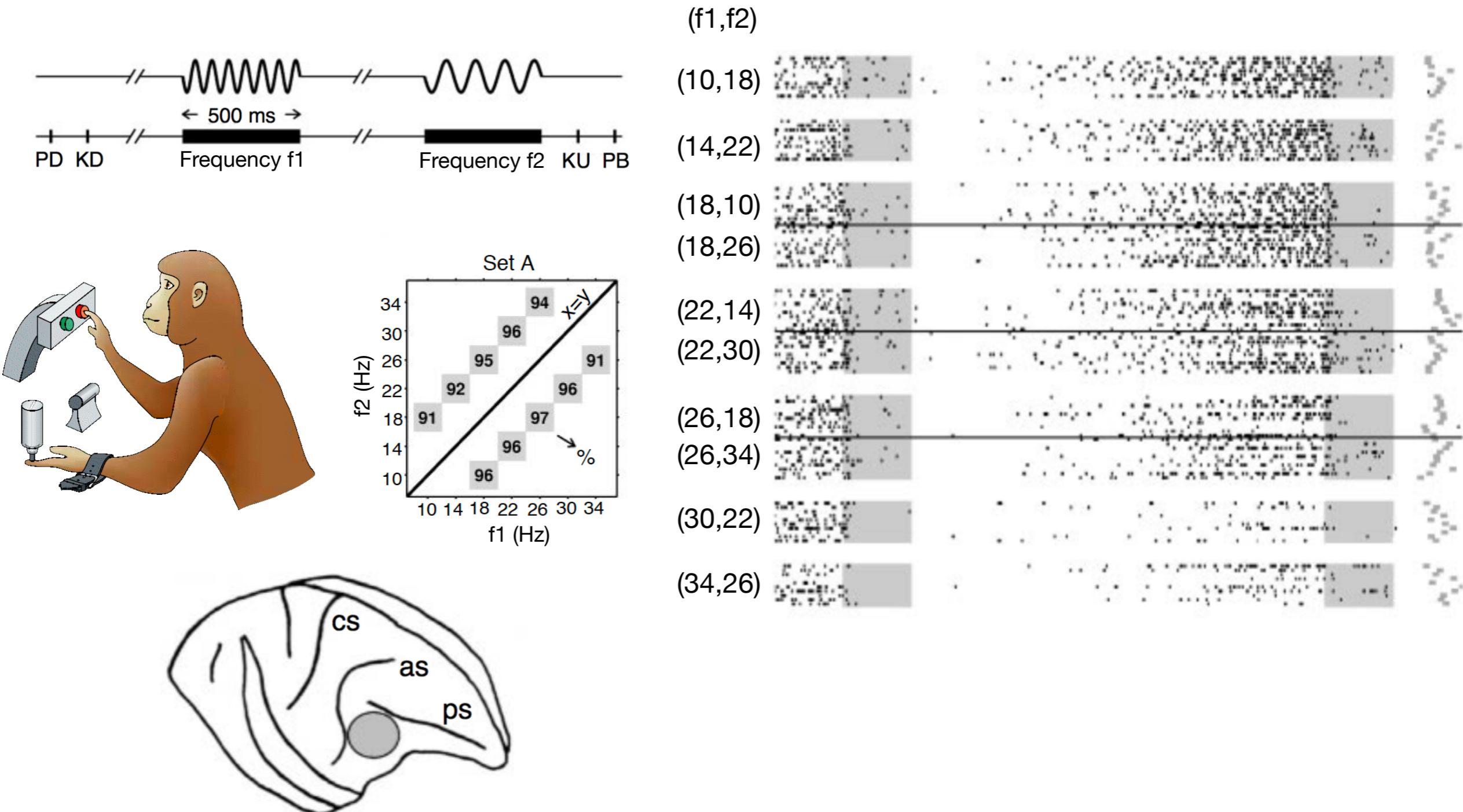
Cell-by-cell

PFC, working memory task



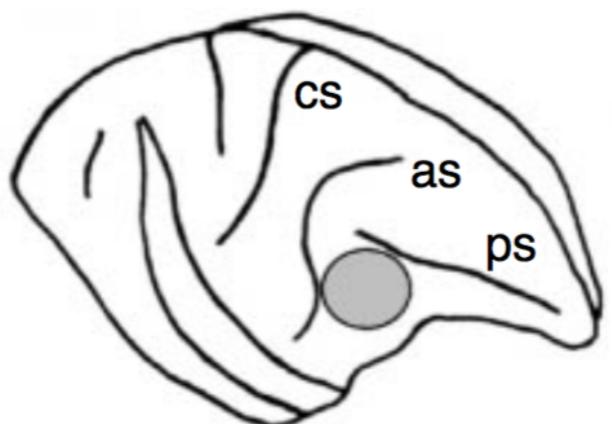
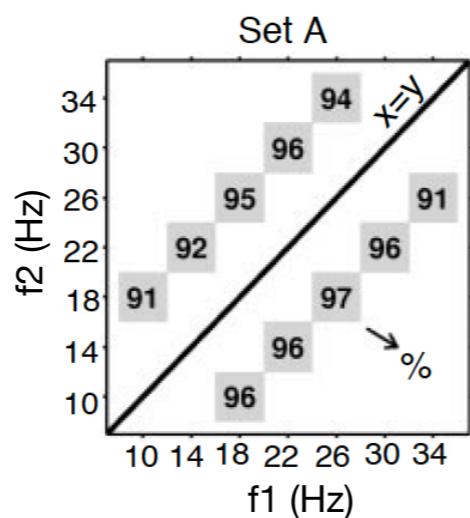
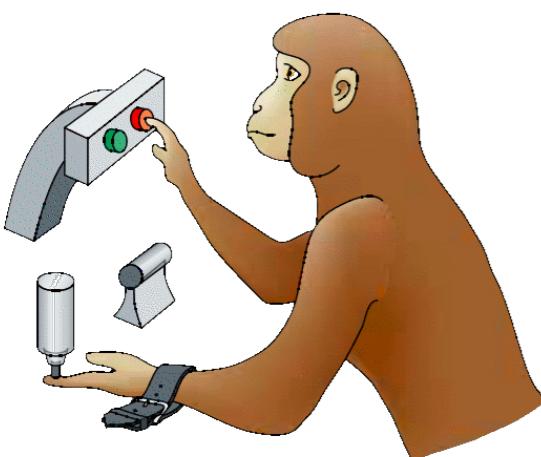
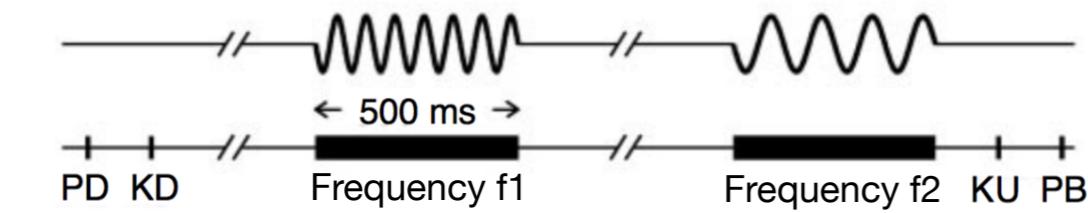
PFC, working memory task

Single cell recording: trial-to-trial spike trains



PFC, working memory task

Single cell: Peri-stimulus time histogram (PSTH)



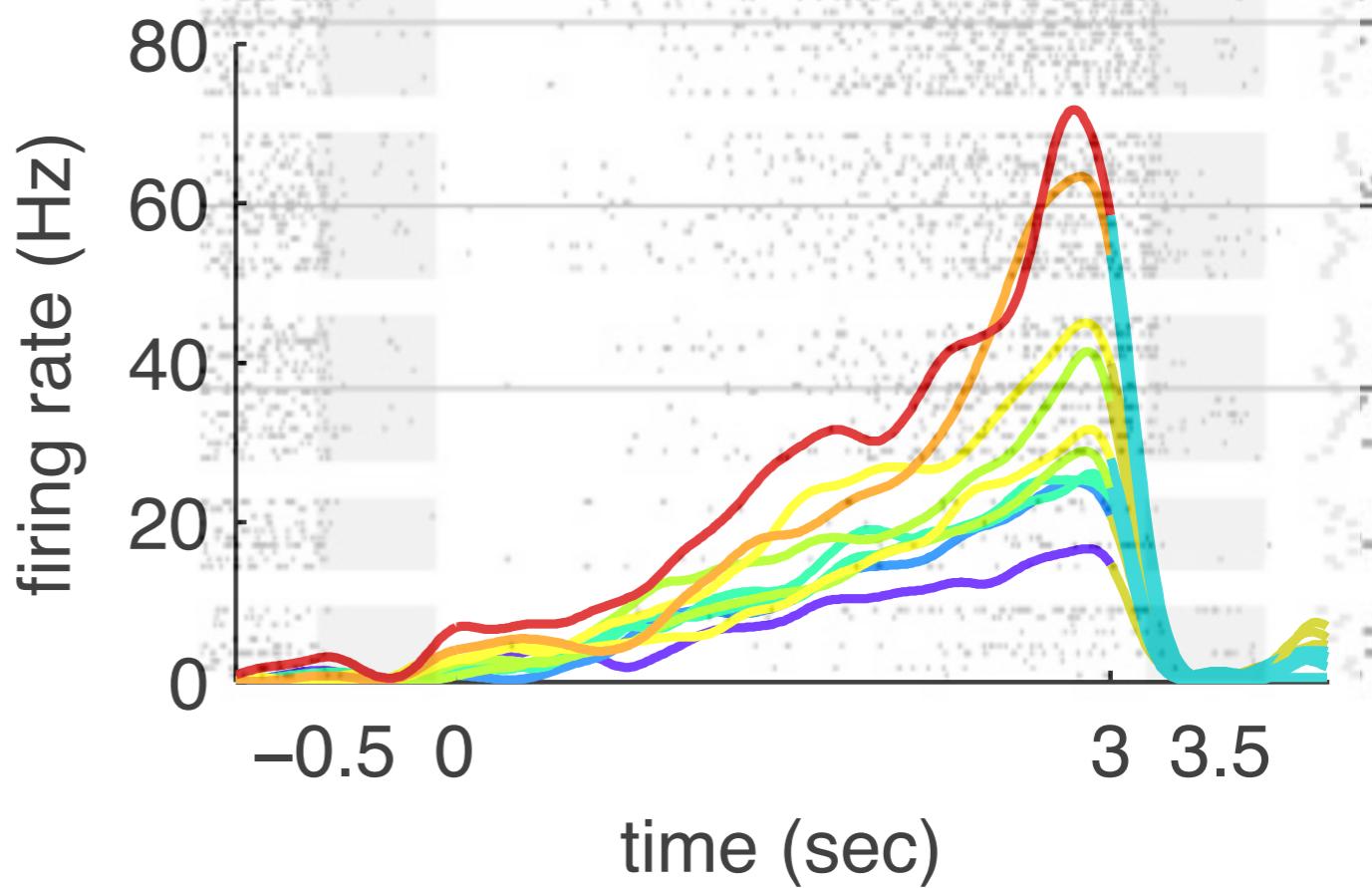
task parameters

stimulus f1

10 Hz 34 Hz

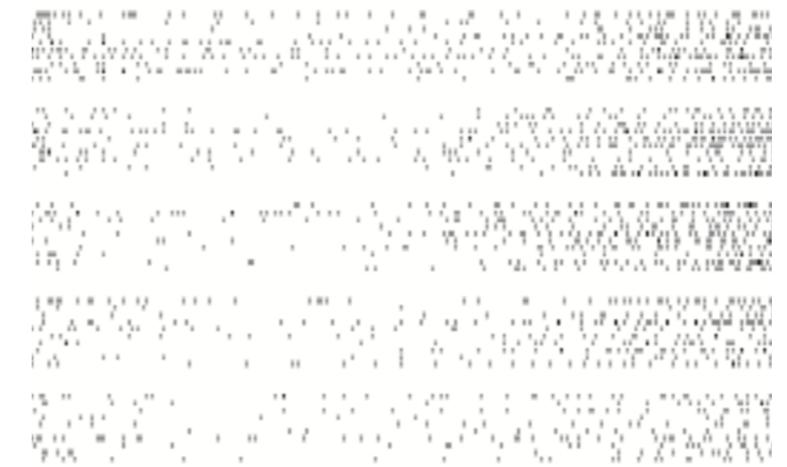
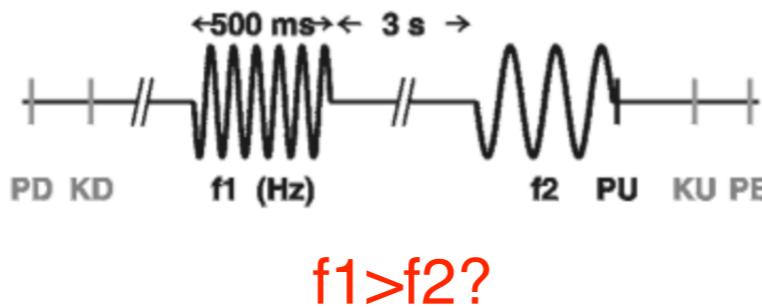
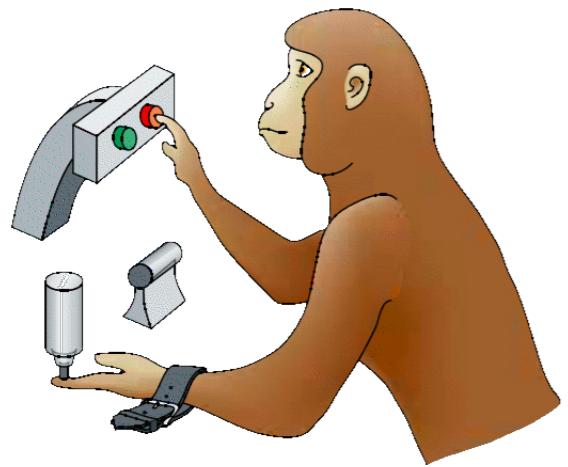
decision

Yes No



Exercise:

Spike Raster / PSTH of PFC neurons



Data:

Recordings over trials:

Stimulus f1

f1 = (10

trial 1

Stimulus f2

f2 = (18

18

...)

Decisions:

d = (1

26

...)

0

...)

Spike times #1

s1 = (12 28 53 104 112 256 234 ... 15 29 508 223 145 ...)

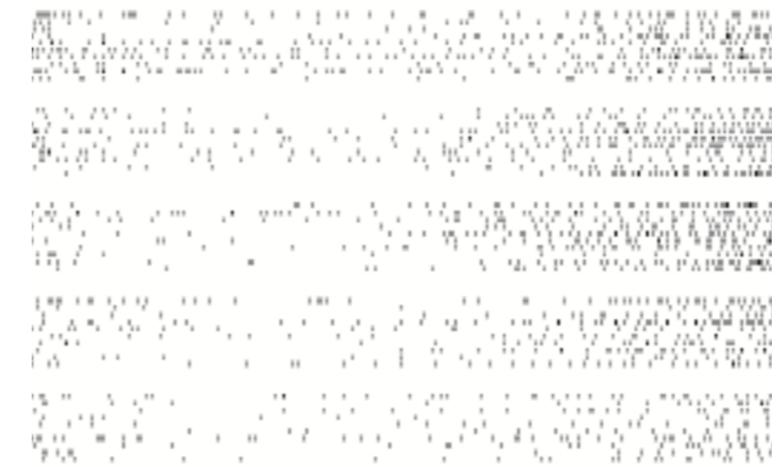
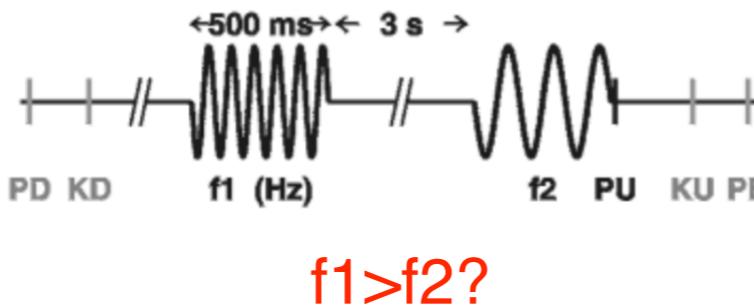
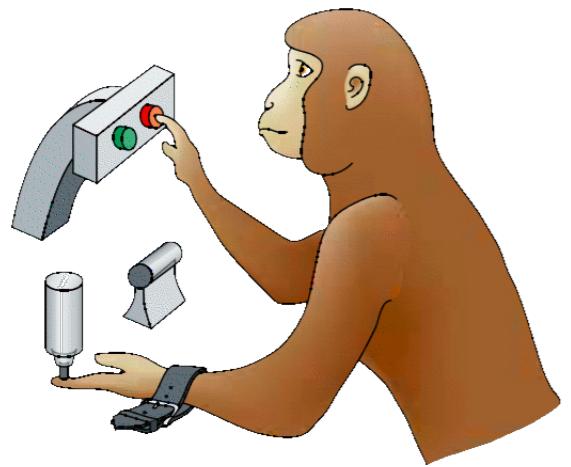
Spike times #2

s2 = (44 80 178 245 335 399 400 ... 66 94 100 174 205 ...)

...

Exercise:

Spike Raster / PSTH of PFC neurons



To-do-list

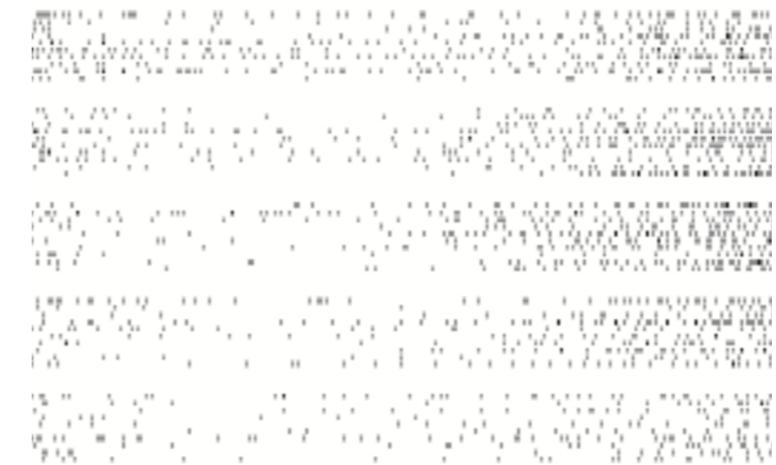
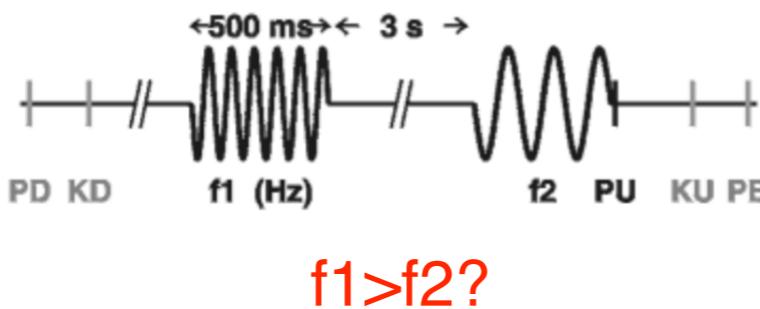
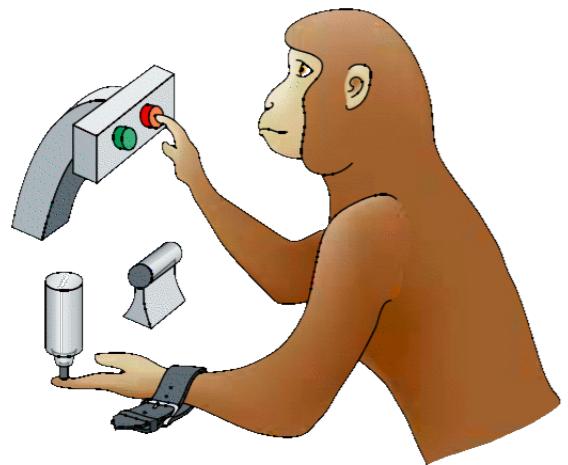
- (1) load a file from romo folder
- (2) extract all spike times of one neuron
- (3) make a raster plot for that neuron
- (4) sort spike times according to trial type
- (5) compute peri-stimulus time histograms (PSTH) for each trial type
- (6) [Advanced:] PSTH standard deviation
- (7) loop over all electrodes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

MATLAB commands

- (1)
- (2) cell arrays
- (3) ind2sub; sub2ind
- (4) unique;
- (5)
- (6)
- (7)
- (8)

Exercise:

Solution files



Matlab scripts ...

- (1) romo_unsorted
- (2) romo_sorted
- (3) romo_psth
- (4) romo_allpsth

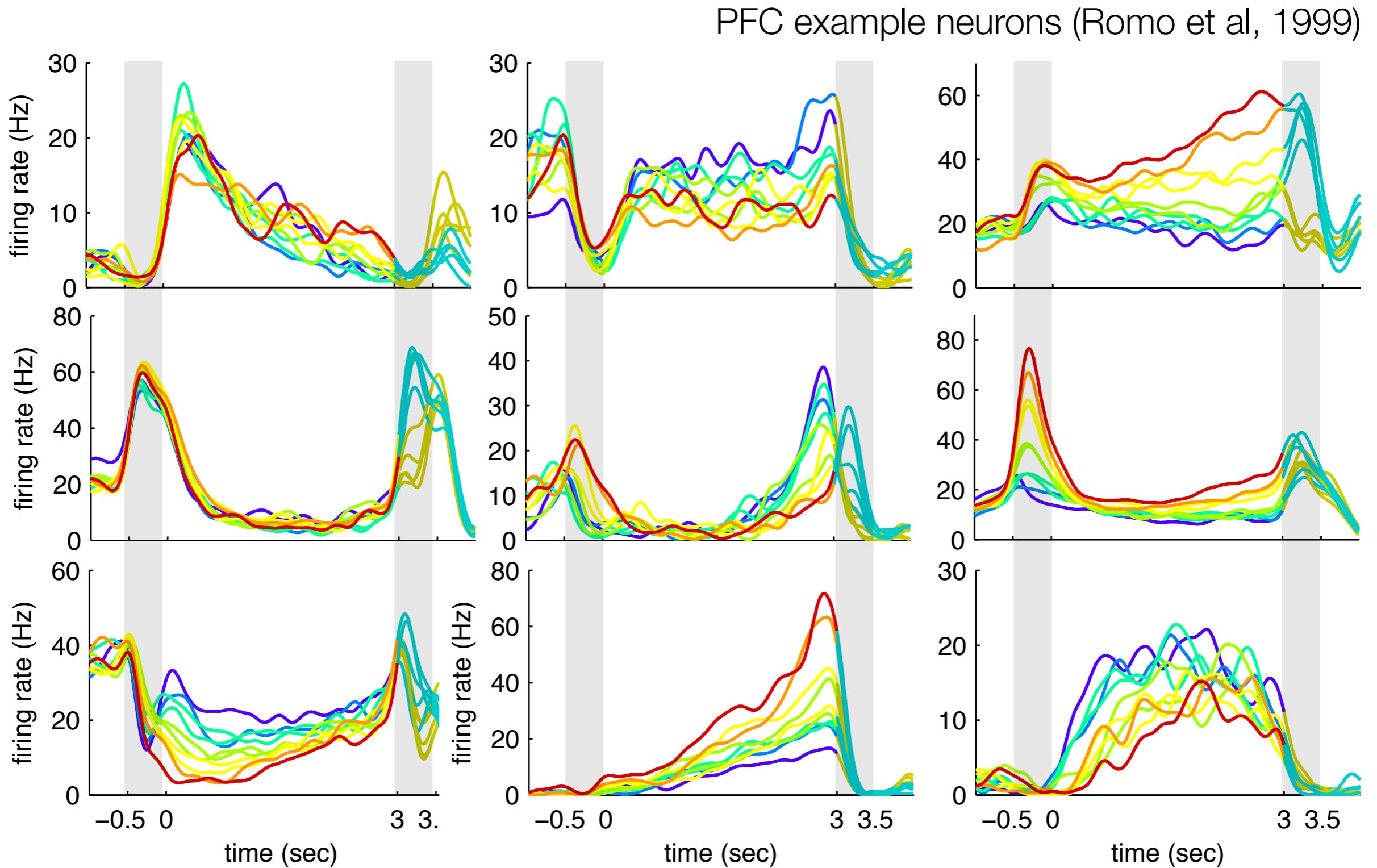
... and what they do

- spike raster plot of one cell, trials over time
- spike raster plot of one cell, sorted by f1 and d
- psth of one cell, plotted as function of (f1, d, t)
- extract psths of all cells in data folder rr014

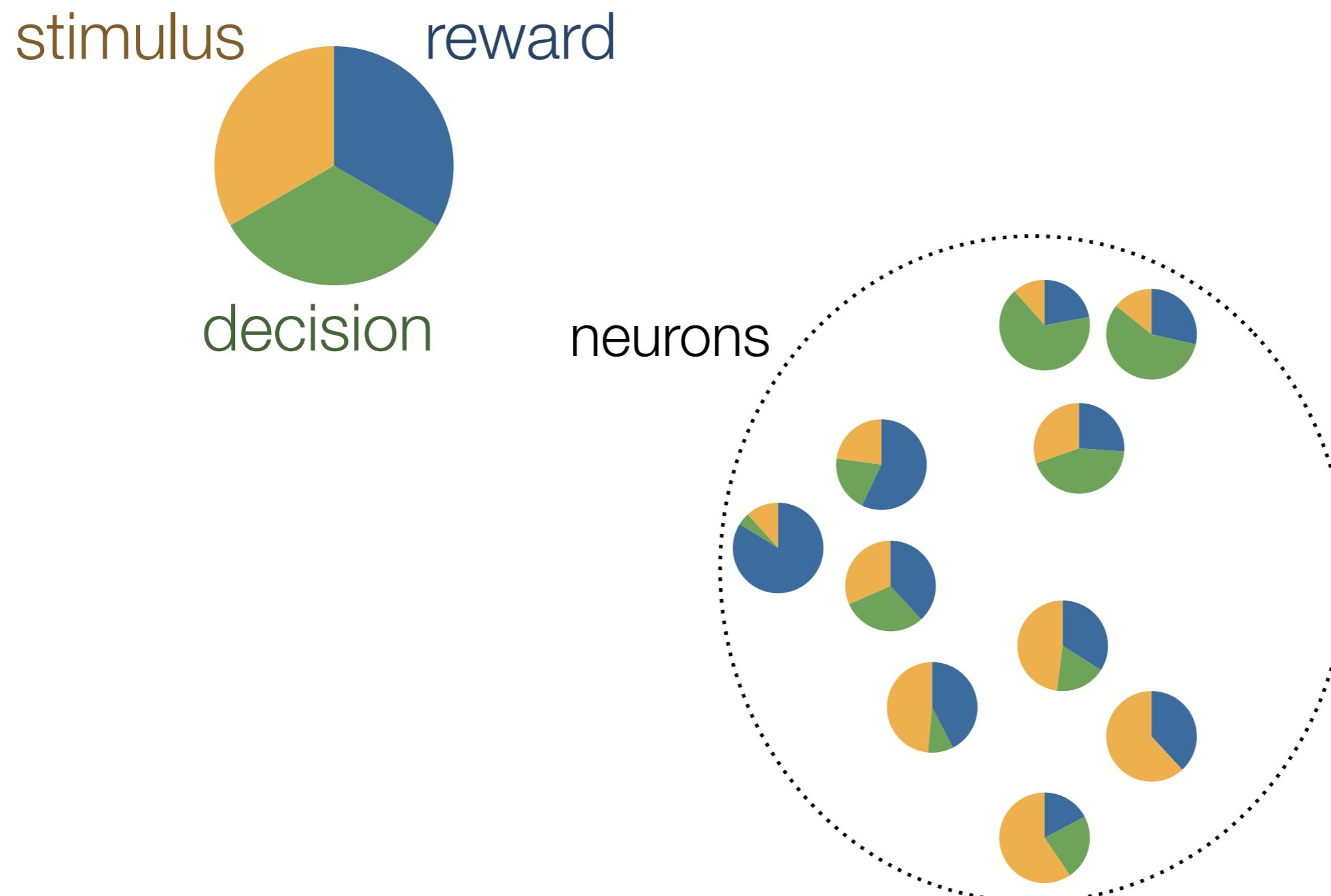
$X = [\# \text{ neurons}] \times [\# \text{ conditions}] \times [\# \text{ time points}]$

PFC, working memory task

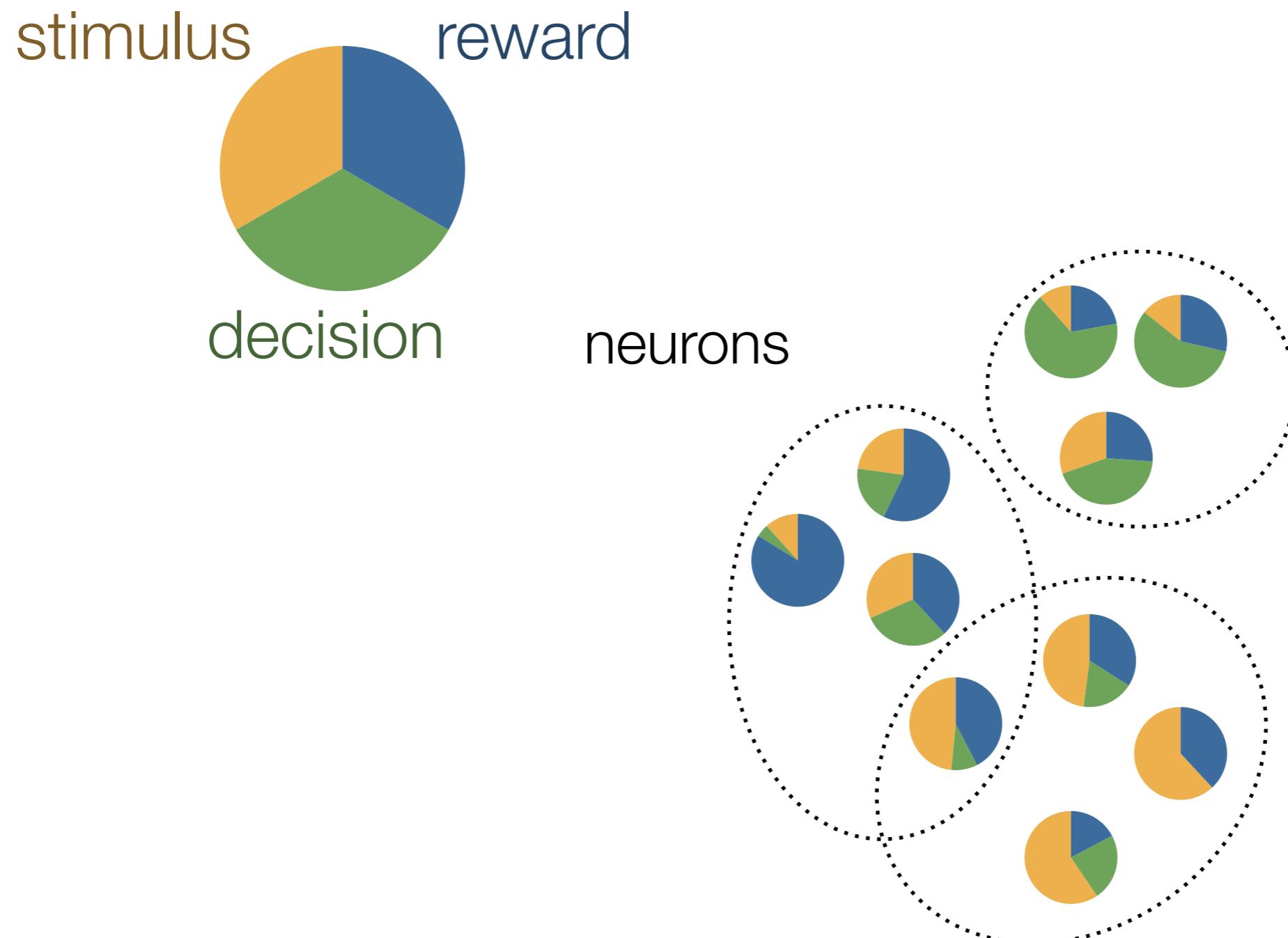
PSTHs of nine (out of~1000) cells



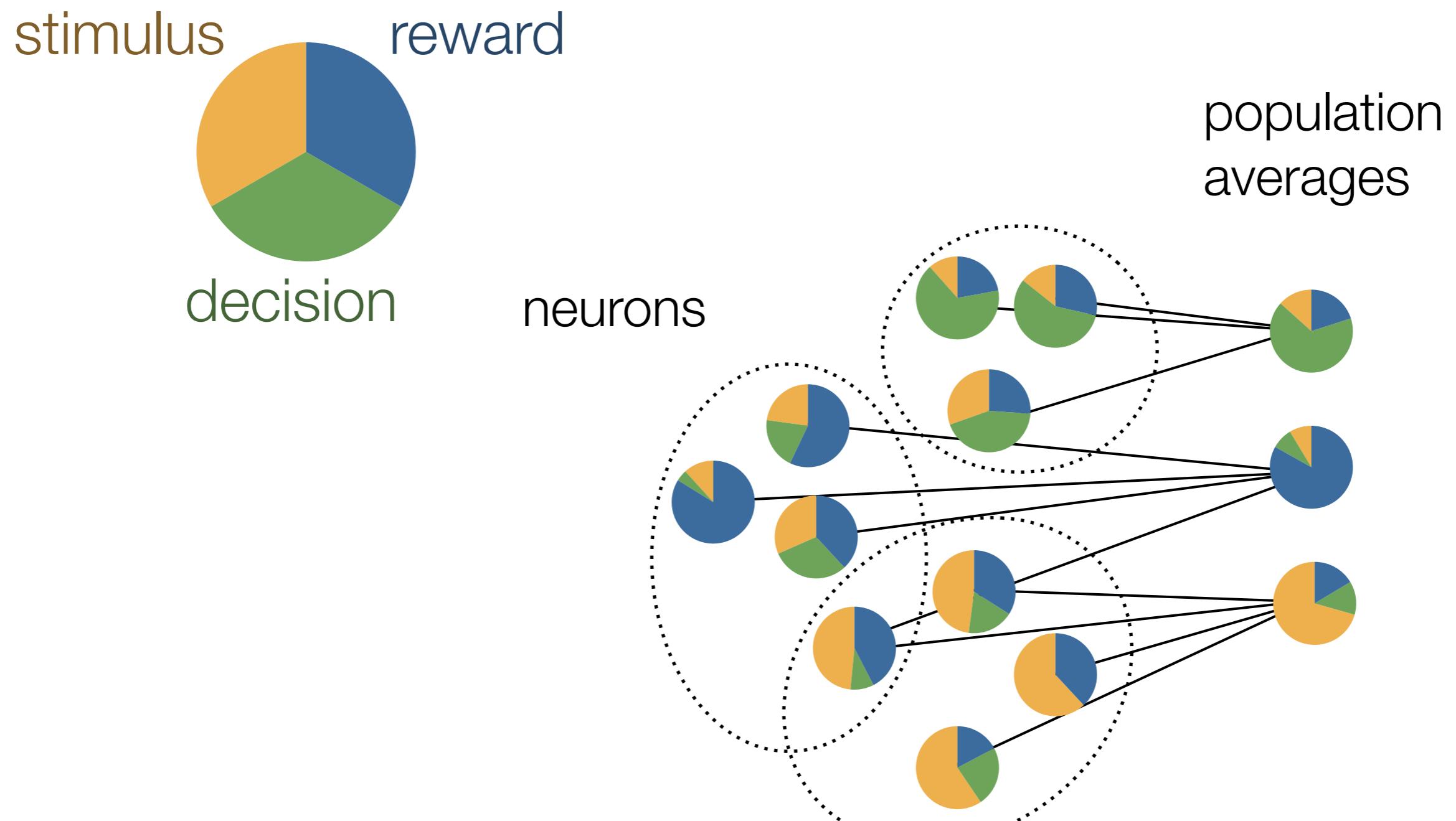
The classical approach: Sorting cells by selectivity (tuning)



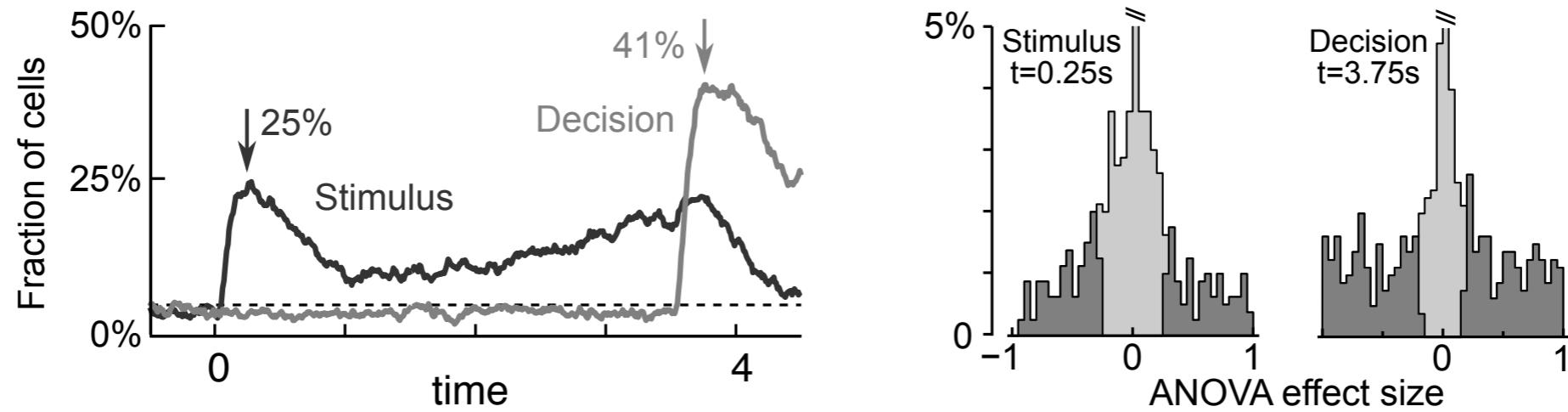
The classical approach: Sorting cells by selectivity (tuning)



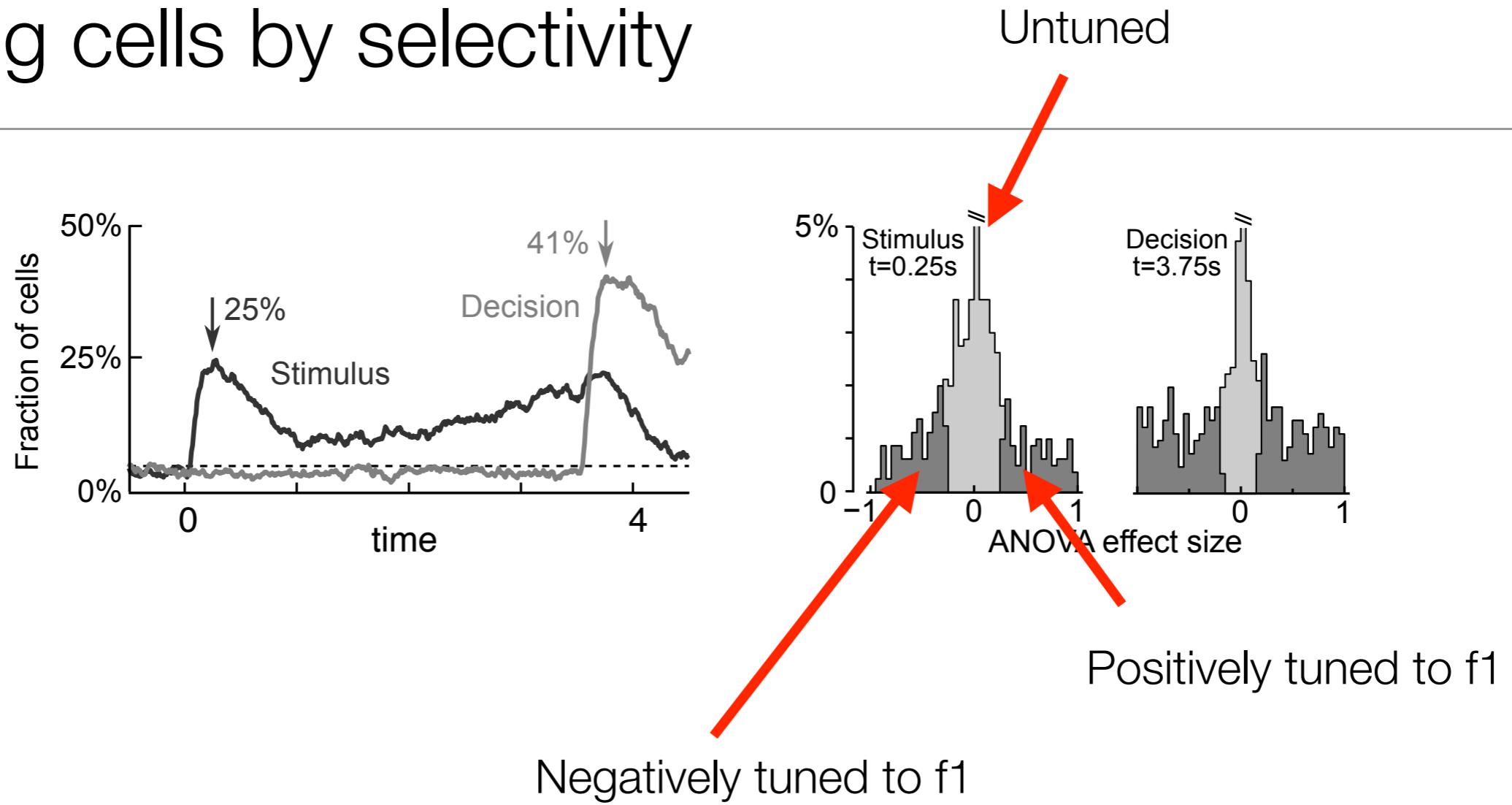
The classical approach: Sorting cells by selectivity & averaging



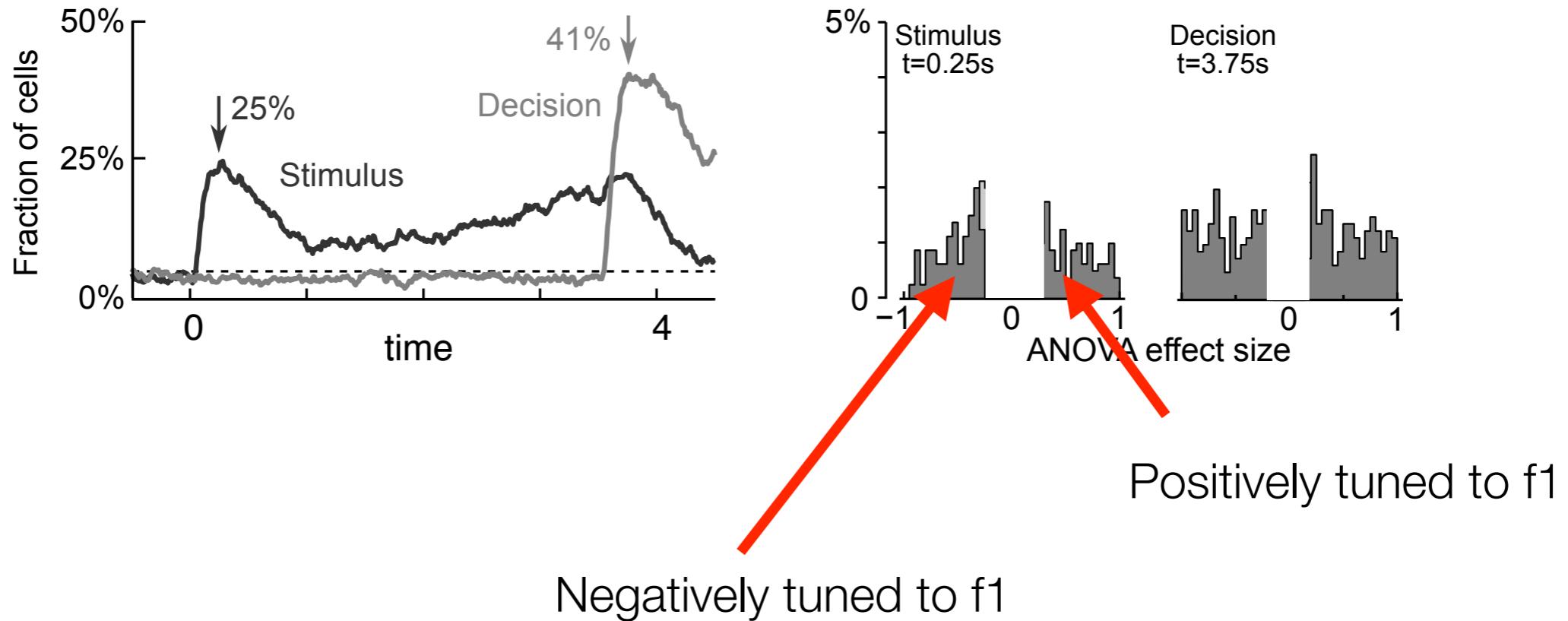
The classical approach: Sorting cells by selectivity



The classical approach: Sorting cells by selectivity



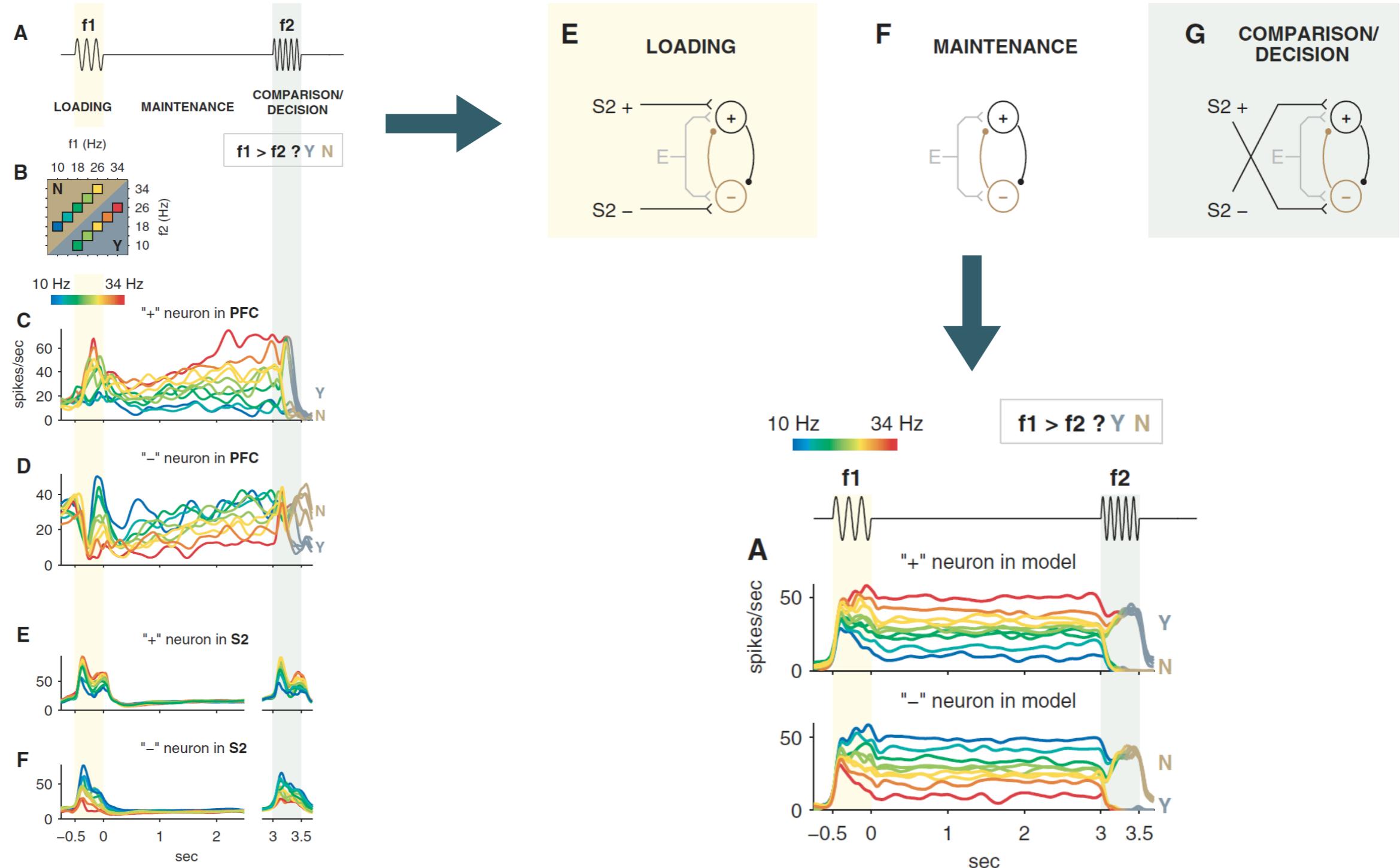
The classical approach: Sorting cells by selectivity



Other examples:

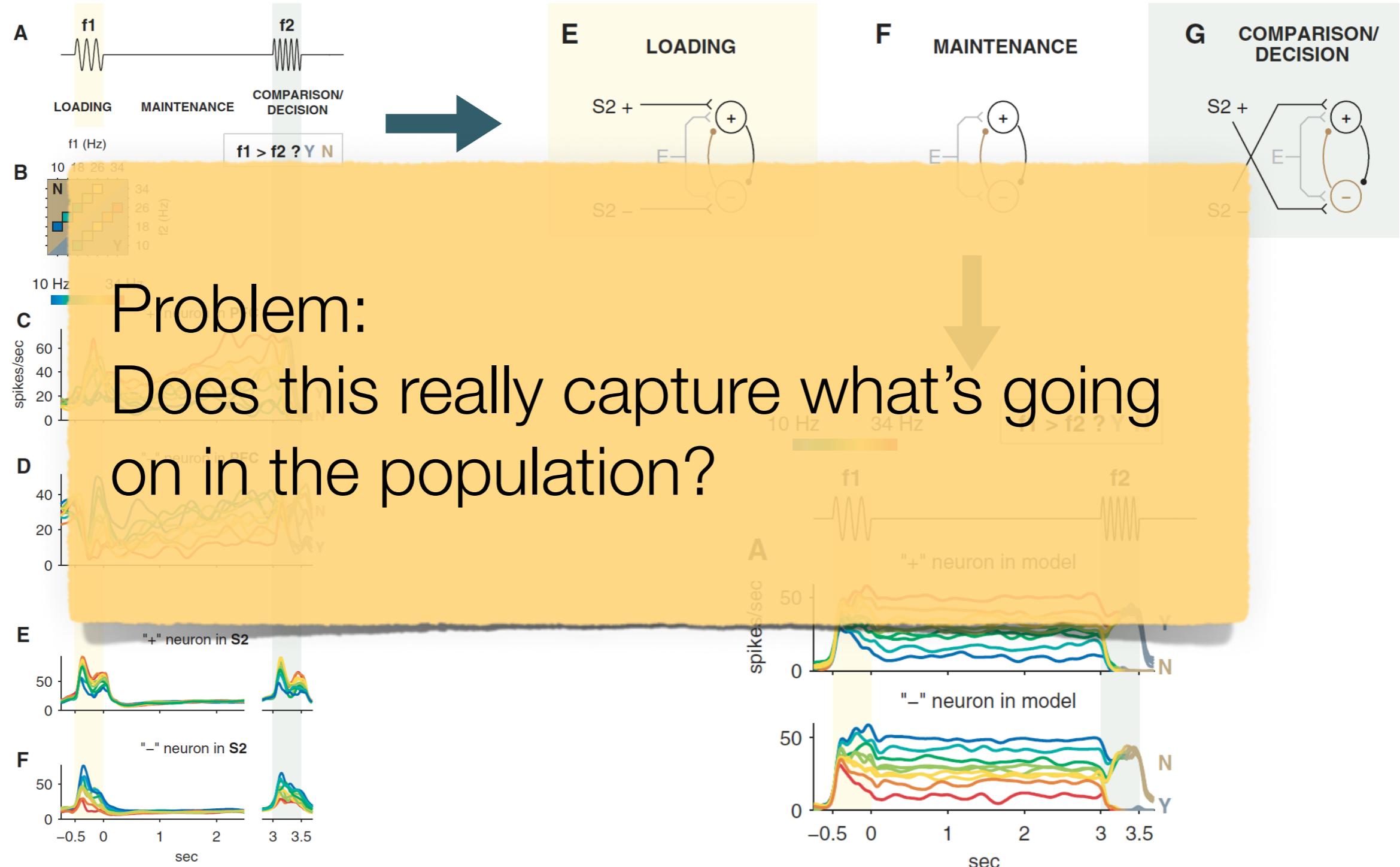
- Same task, other areas (S2, Premotor Cortex, M1)
- V1: Simple cells vs complex cells (see e.g. Mechler & Ringach, Vision Research, 2002)

Personal aside: 'Looking' at the data influences theory



Machens et al, Science, 2005

Personal aside: 'Looking' at the data influences theory



How to ‘look’ at the data (2)

Dimensionality reduction applied to neural populations

Population data and what to do with it

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

Population data and what to do with it

Objective:

find structure in x

Method:

unsupervised

Examples:

PCA

Clustering

...

Population data and what to do with it

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

$$\mathbf{Y} = \begin{bmatrix} \text{time 1} & \text{time 2} & \text{time 3} & & \text{time T} \\ y_{11} & \dots & & & y_{2T} \end{bmatrix} \quad \begin{array}{l} \text{label 1} \\ \text{label 2} \end{array}$$

Population data and what to do with it

Objective:

find structure in x

Method:

unsupervised

Examples:

PCA

Clustering

...

Objective:

find structure in x
that predicts y

Method:

supervised

Examples:

Regression

Classification

...

Population data and what to do with it

Objective:

find structure in x

Method:

unsupervised

Examples:

PCA

Clustering

...

low-dimensional

Objective:

find structure in x
that predicts y

Method:

supervised

Examples:

Regression

Classification

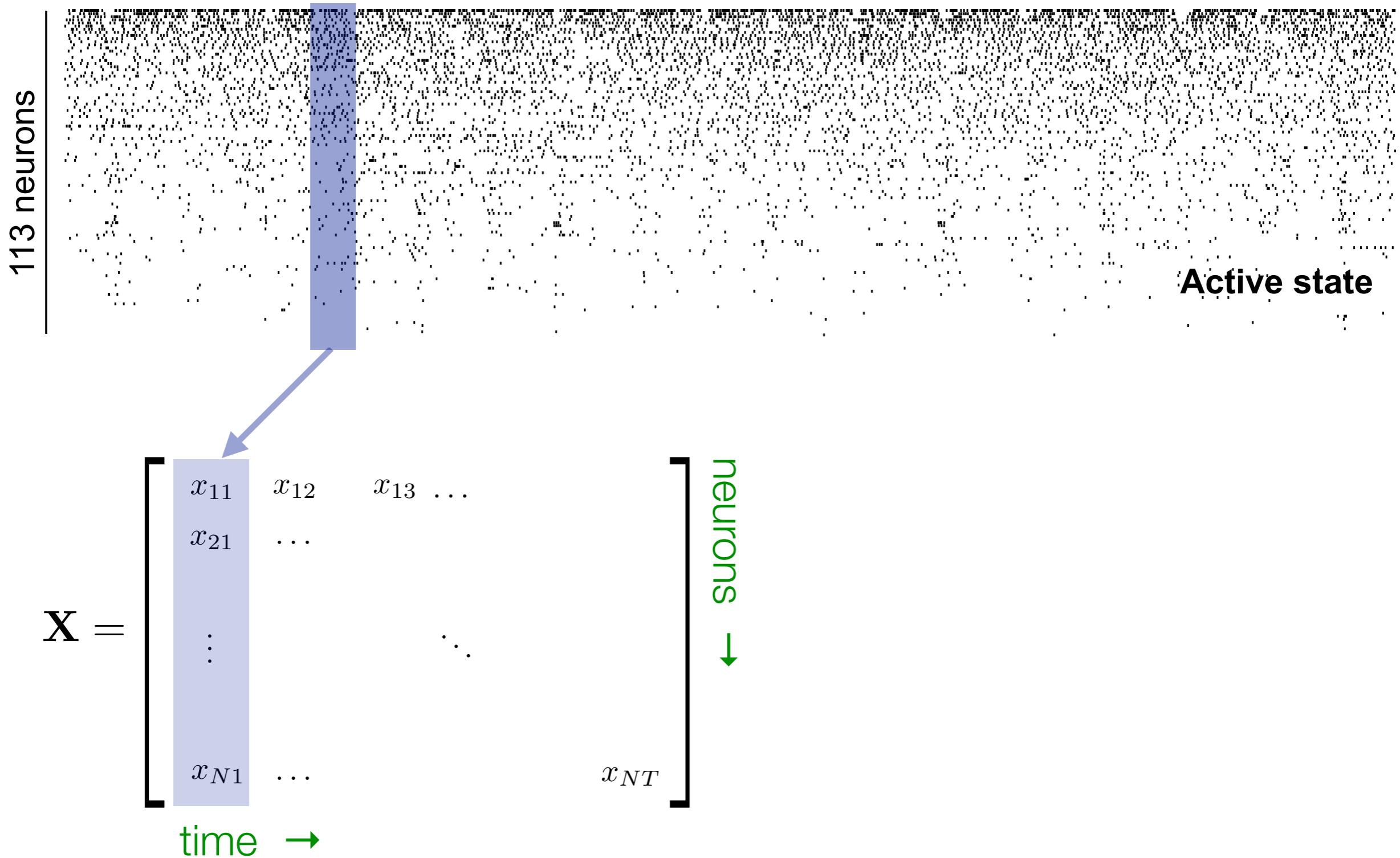
...

What is dimensionality reduction and why do we need it?

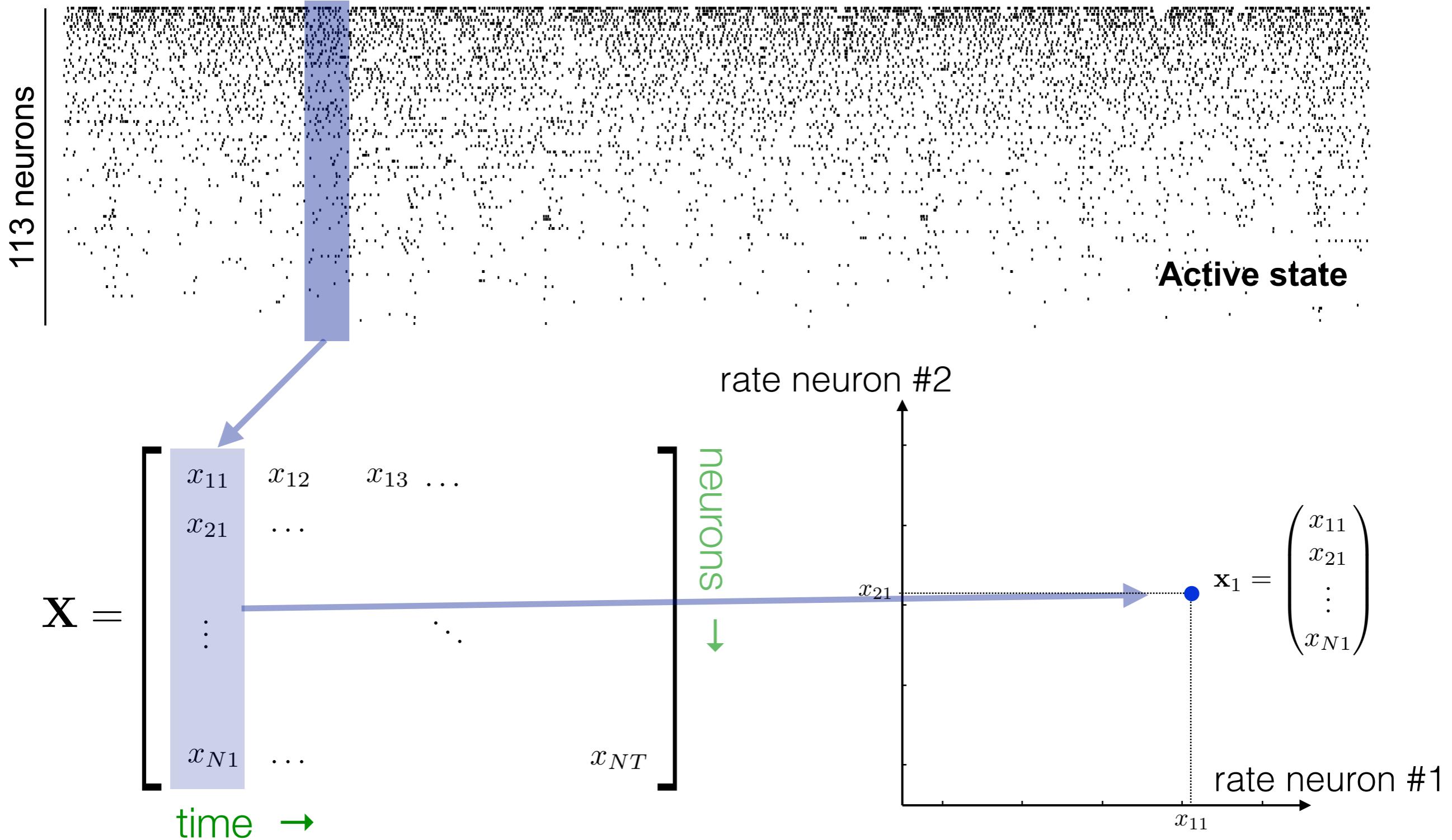
Reduce the data but keep the essential features

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

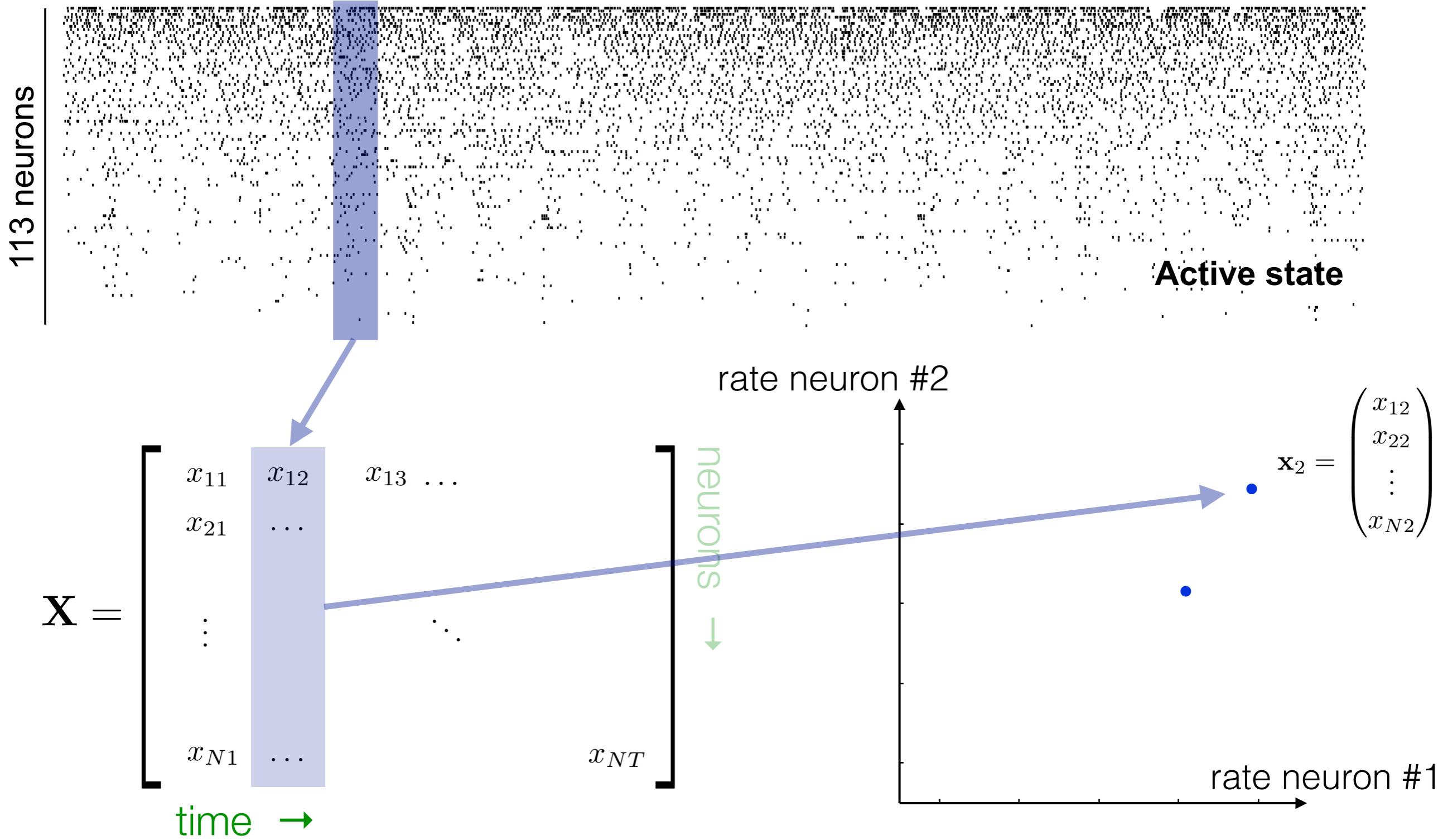
Population activity reconsidered (ignoring the task context for now)



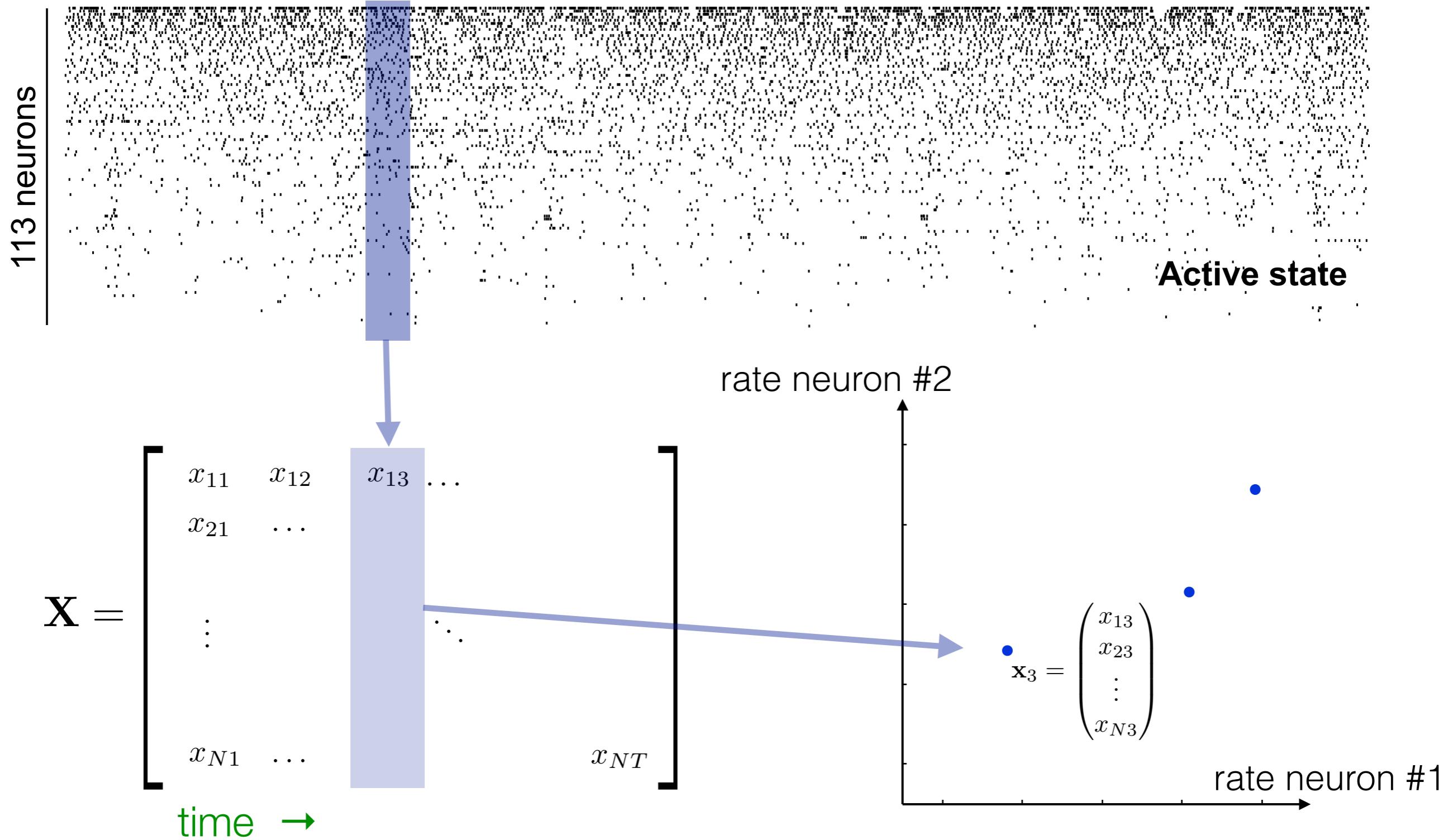
Population activity reconsidered (ignoring the task context for now)



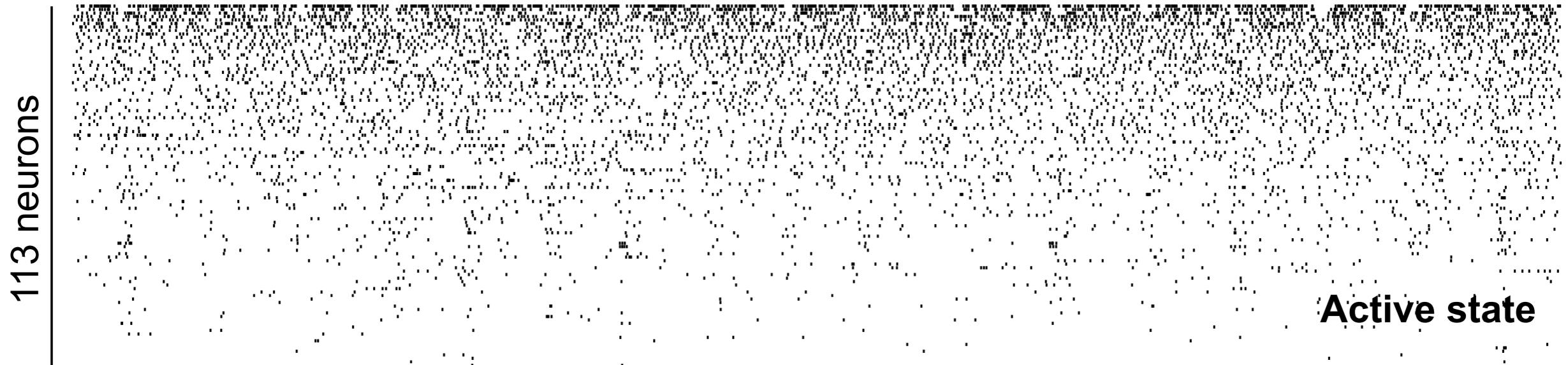
Population activity reconsidered (ignoring the task context for now)



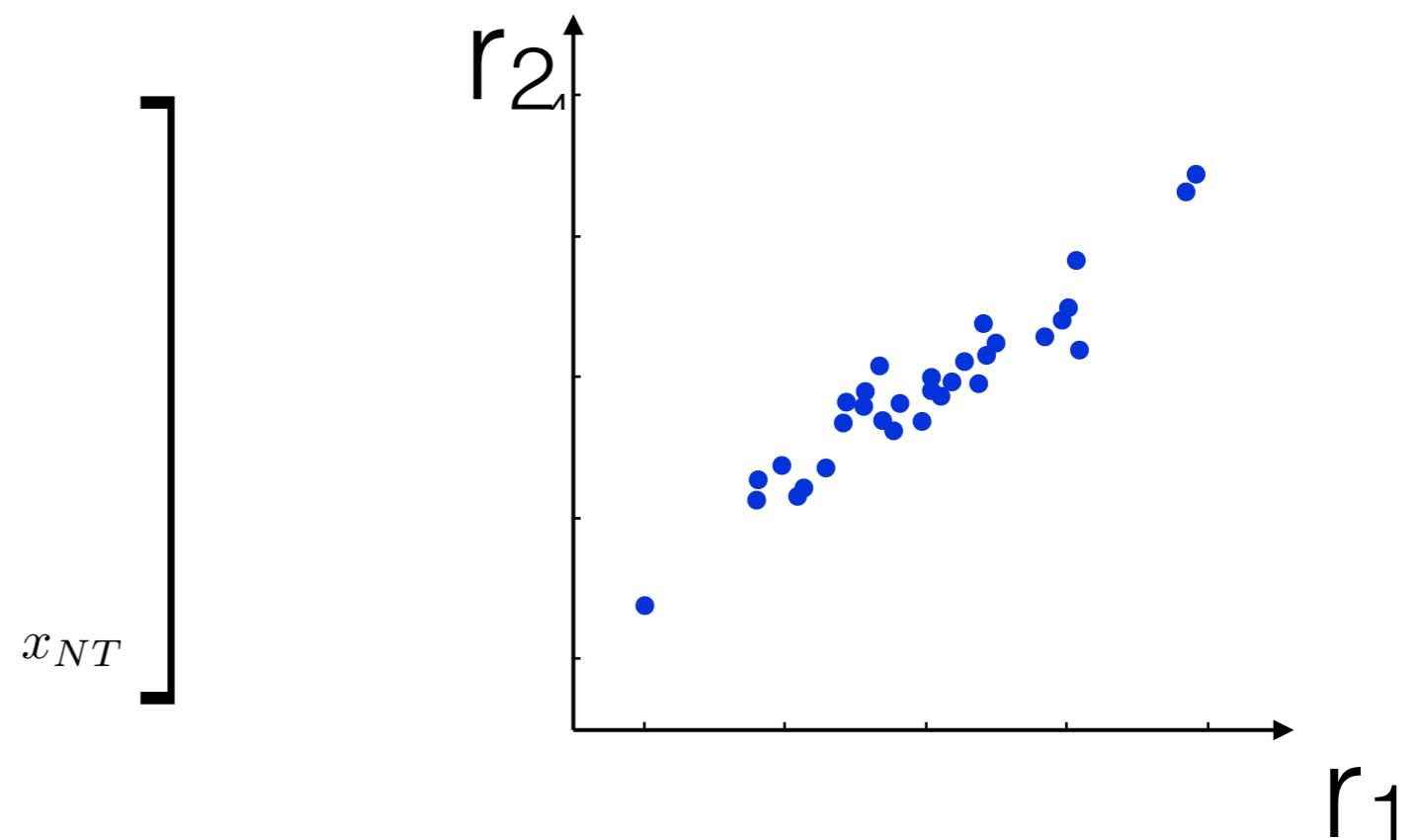
Population activity reconsidered (ignoring the task context for now)



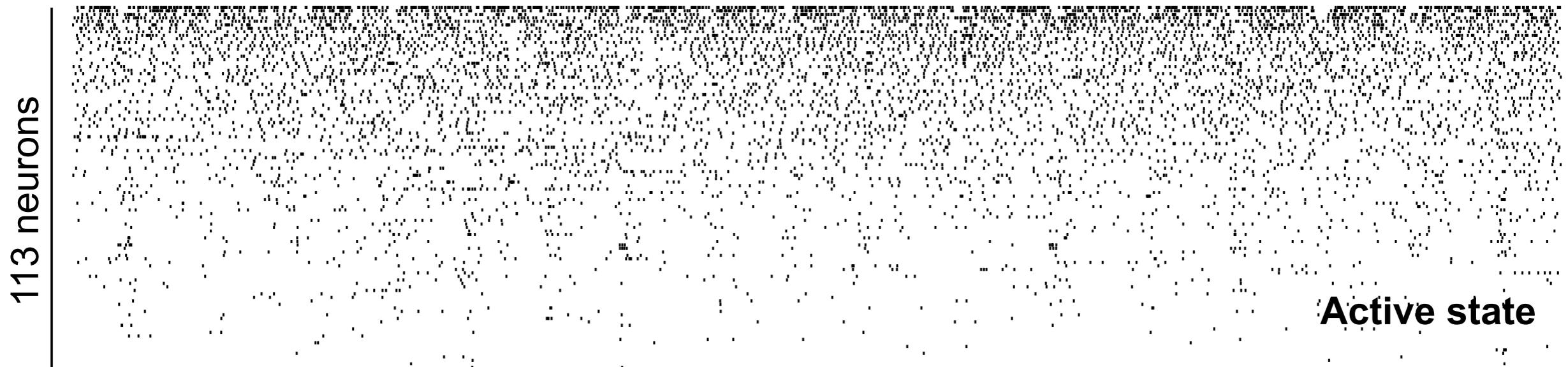
Population activity reconsidered (ignoring the task context for now)



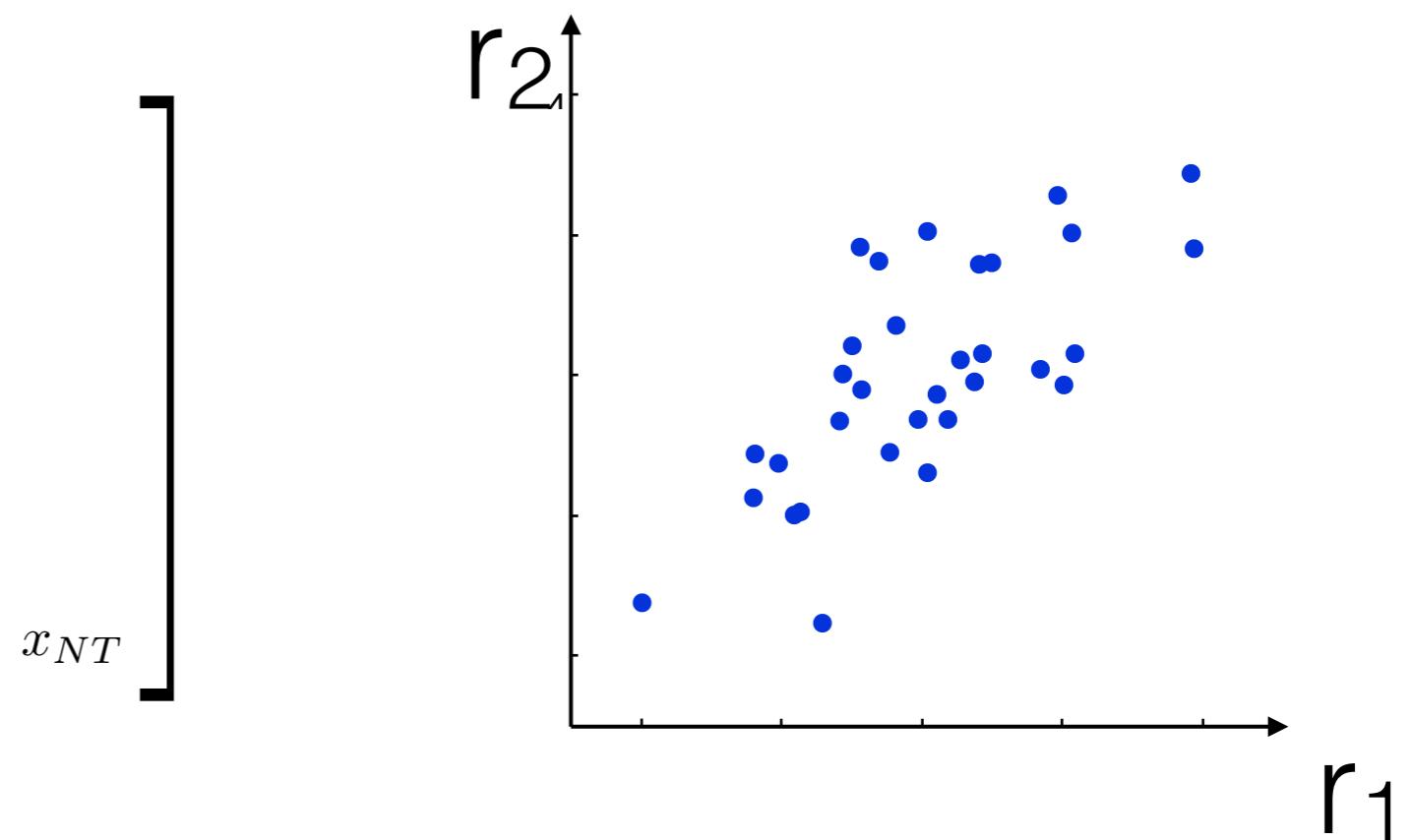
$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots \\ x_{21} & \dots & & \\ \vdots & & \ddots & \\ x_{N1} & \dots & & x_{NT} \end{bmatrix}$$



Population activity reconsidered (ignoring the task context for now)

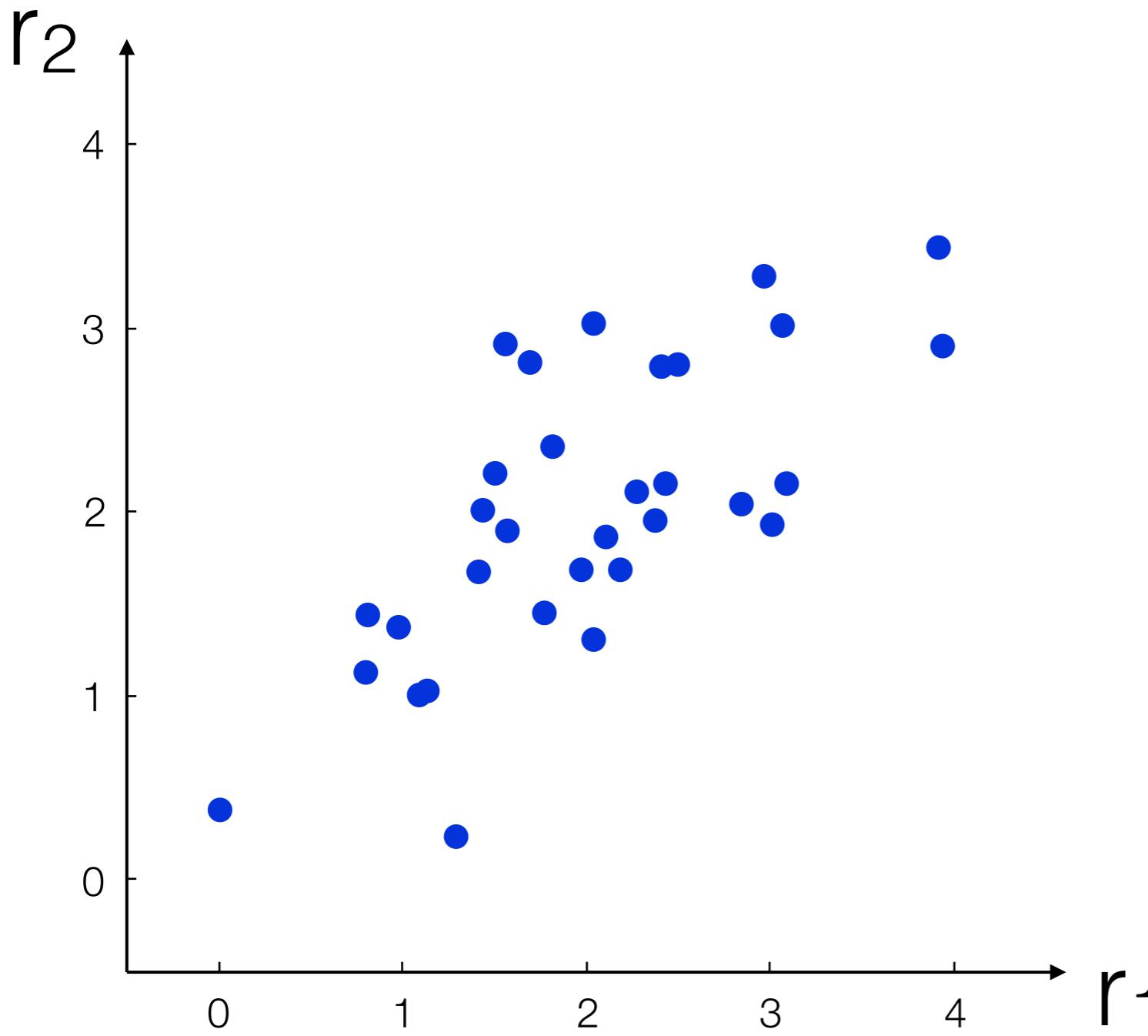


$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots \\ x_{21} & \dots & & \\ \vdots & & \ddots & \\ x_{N1} & \dots & & x_{NT} \end{bmatrix}$$



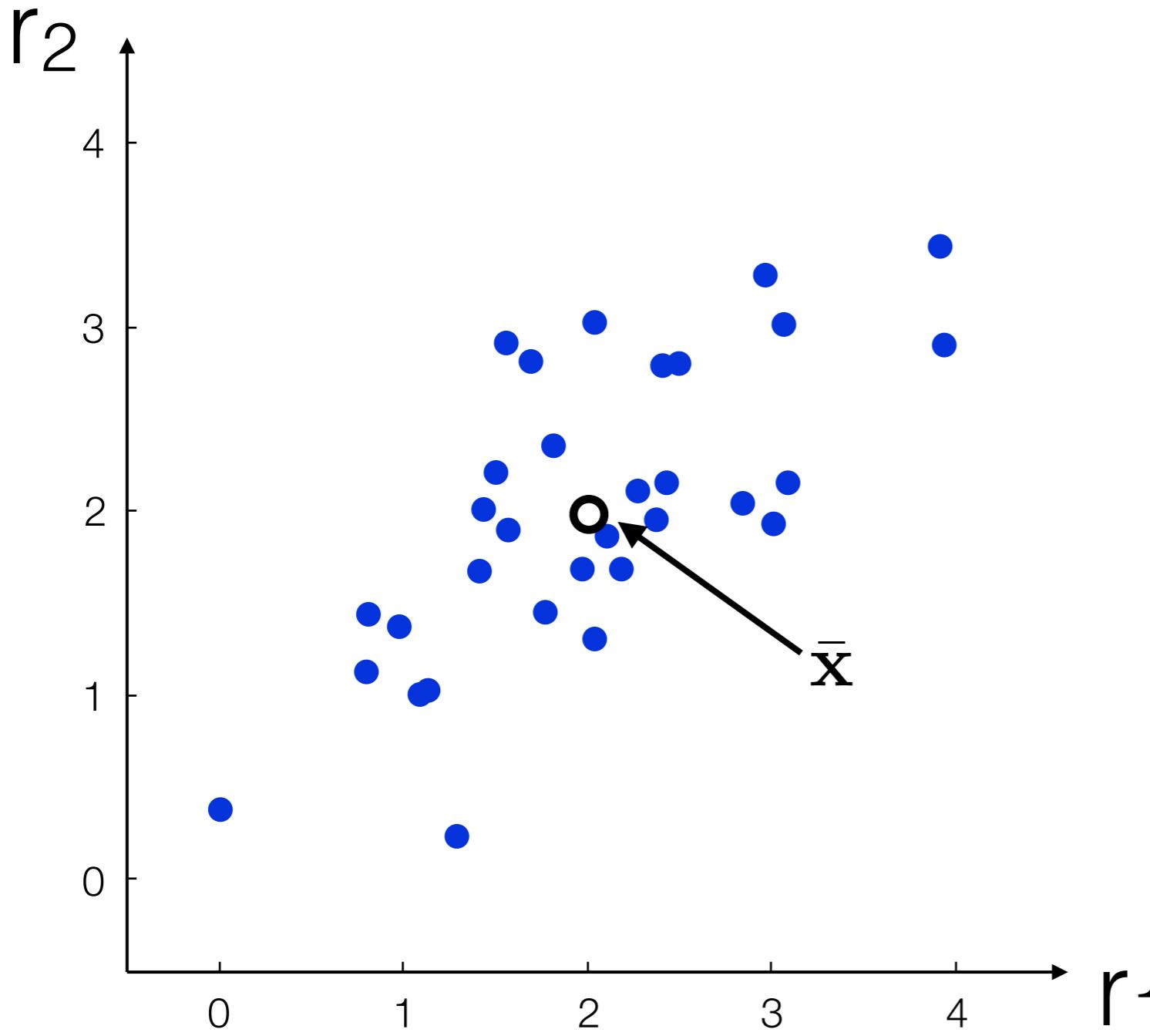
Population activity reconsidered

State space of firing rates



Principal component analysis step-by-step

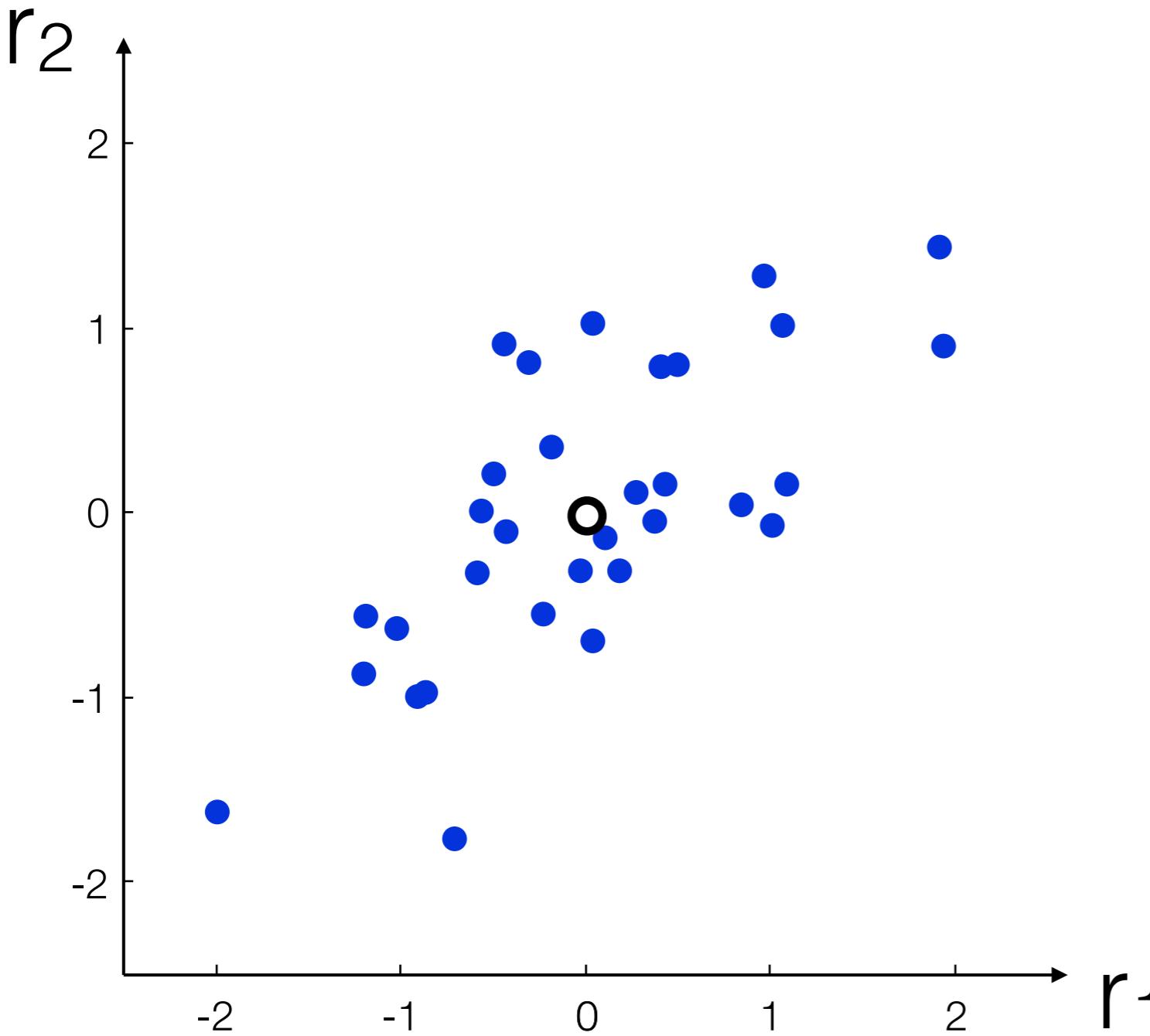
(1) Centering the data



$$\bar{\mathbf{x}} = \frac{1}{T} \sum_{i=1}^T \mathbf{x}_i$$
$$\mathbf{x}_i \rightarrow \mathbf{x}_i - \bar{\mathbf{x}}$$

Principal component analysis step-by-step

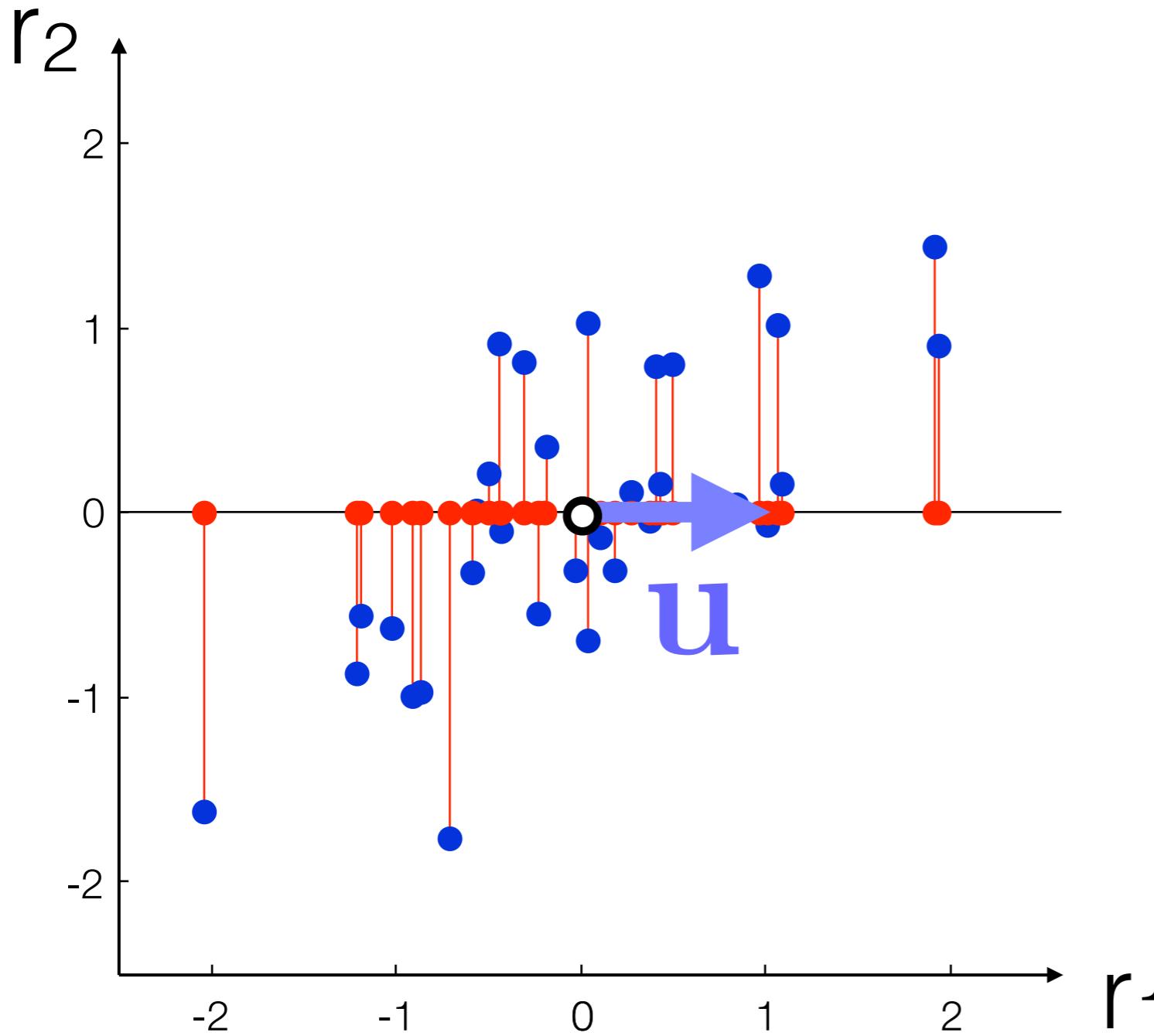
(1) Centering the data



$$\bar{\mathbf{x}} = \frac{1}{T} \sum_{i=1}^T \mathbf{x}_i$$
$$\mathbf{x}_i \rightarrow \mathbf{x}_i - \bar{\mathbf{x}}$$

Principal component analysis step-by-step

(2) Projecting the data onto a subspace



$$\bar{\mathbf{x}} = \frac{1}{T} \sum_{i=1}^T \mathbf{x}_i$$

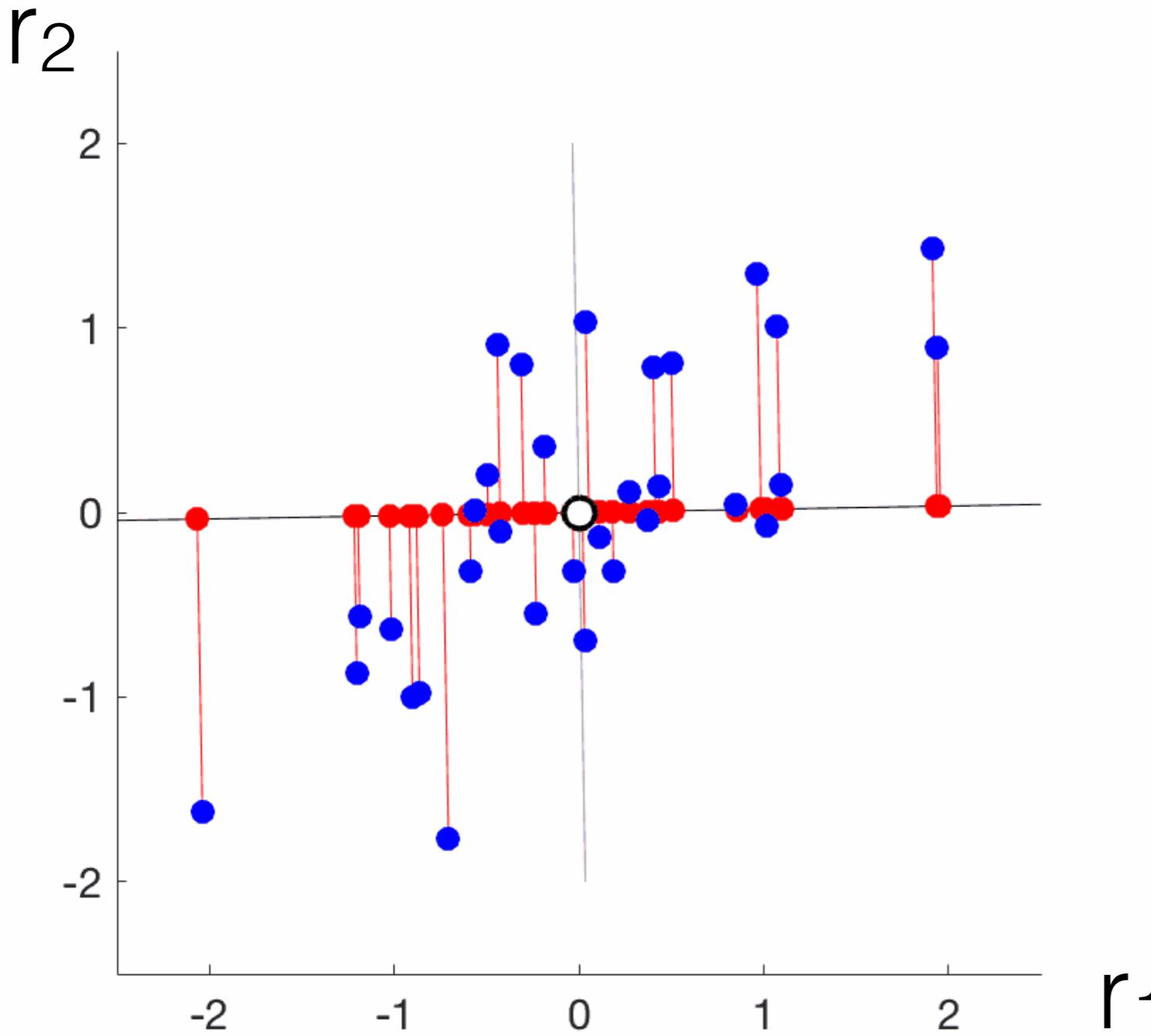
$$\mathbf{x}_i \rightarrow \mathbf{x}_i - \bar{\mathbf{x}}$$

$$\mathbf{p}_i = \mathbf{u} \mathbf{u}^\top \mathbf{x}_i$$

$$\text{with } \|\mathbf{u}\| = 1$$

Principal component analysis step-by-step

(2) Projecting the data onto a subspace



$$\bar{\mathbf{x}} = \frac{1}{T} \sum_{i=1}^T \mathbf{x}_i$$

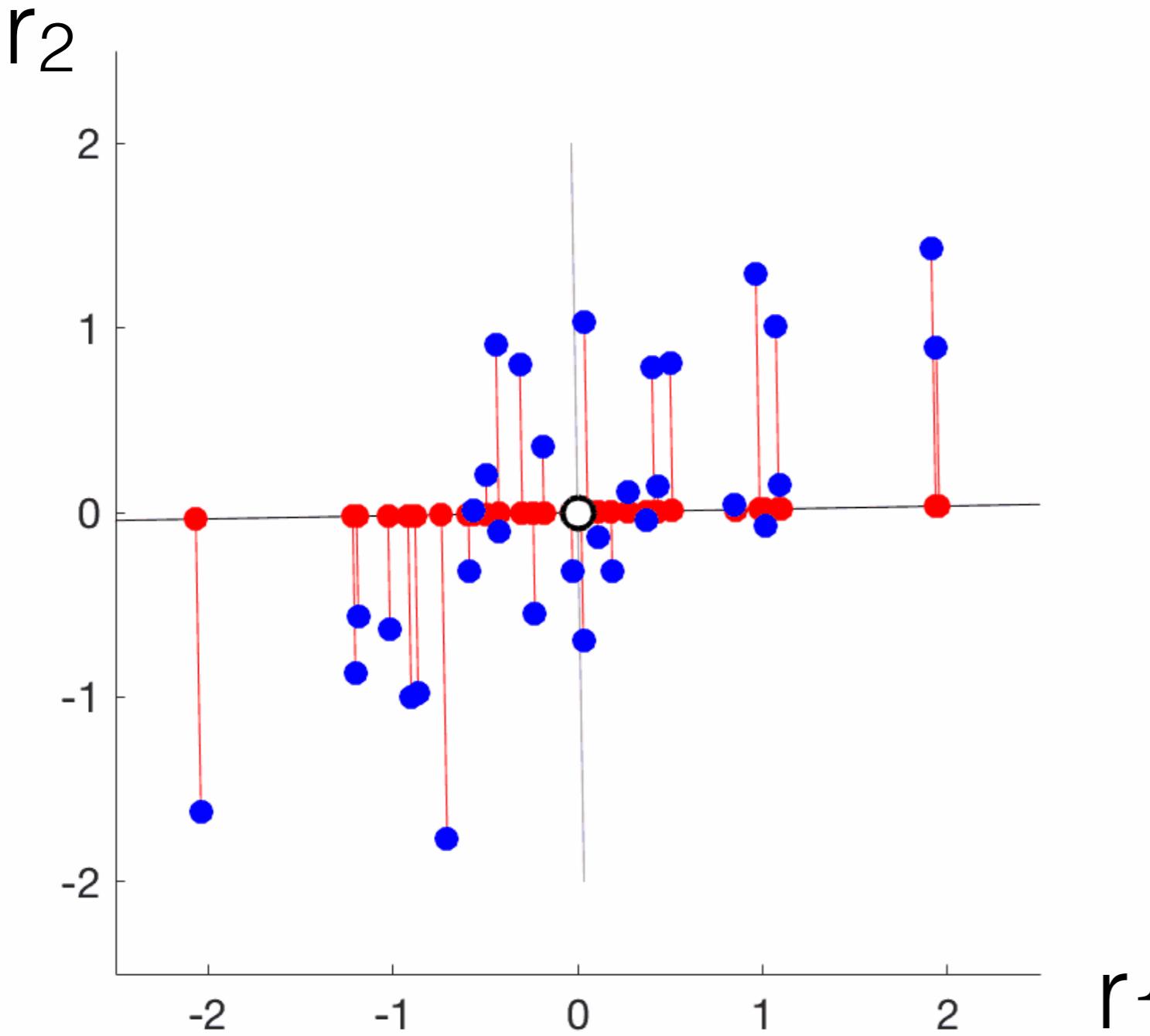
$$\mathbf{x}_i \rightarrow \mathbf{x}_i - \bar{\mathbf{x}}$$

$$\mathbf{p}_i = \mathbf{u}\mathbf{u}^\top \mathbf{x}_i$$

$$\text{with } \|\mathbf{u}\| = 1$$

Principal component analysis step-by-step

(3) Finding the minimum error projection ...



$$\bar{\mathbf{x}} = \frac{1}{T} \sum_{i=1}^T \mathbf{x}_i$$

$$\mathbf{x}_i \rightarrow \mathbf{x}_i - \bar{\mathbf{x}}$$

$$\mathbf{p}_i = \mathbf{u}\mathbf{u}^\top \mathbf{x}_i$$

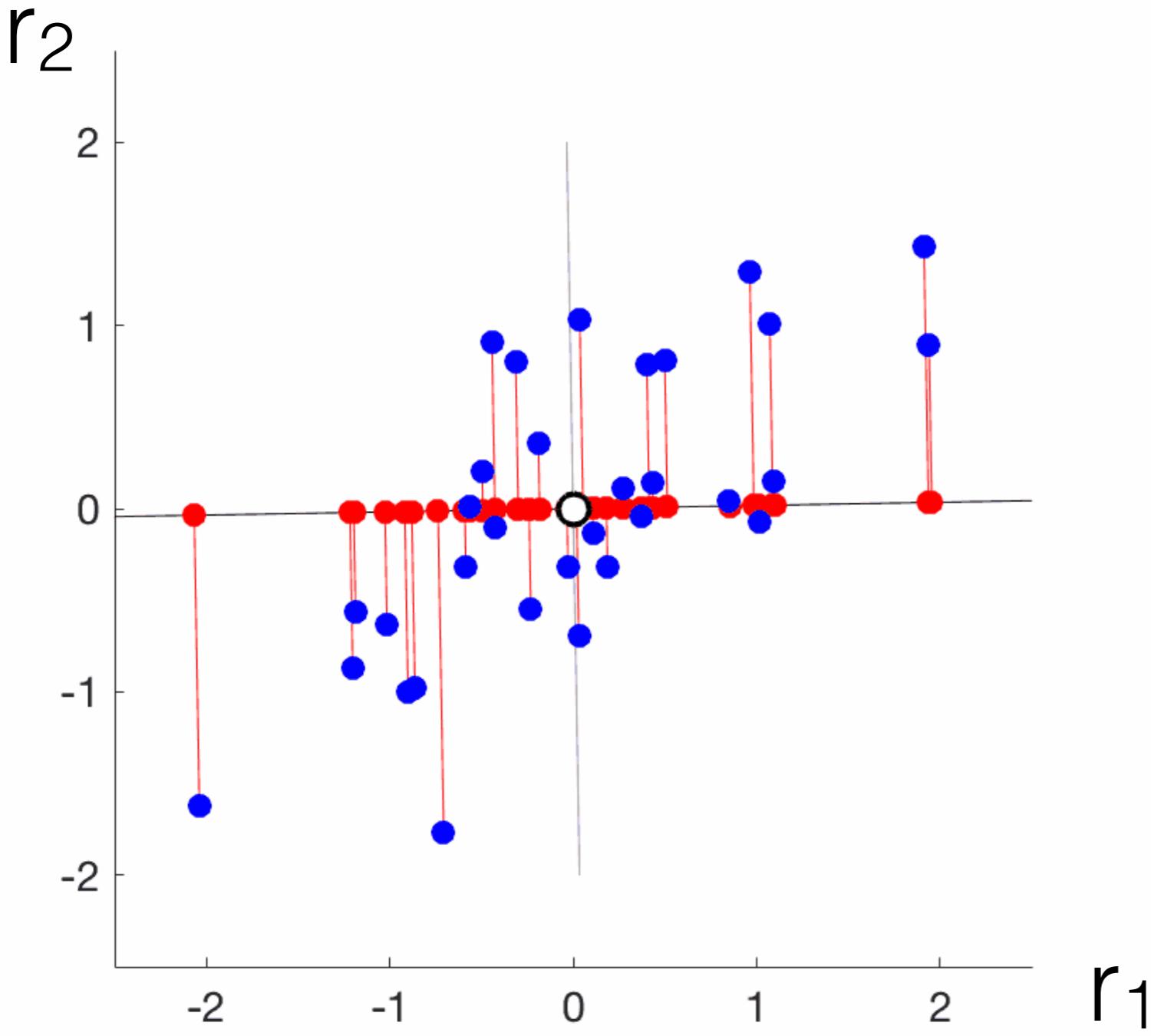
$$\text{with } \|\mathbf{u}\| = 1$$

Projection error:

$$L = \sum_{i=1}^T \|\mathbf{x}_i - \mathbf{u}\mathbf{u}^\top \mathbf{x}_i\|^2$$

Principal component analysis step-by-step

(3) ... or finding the maximum variance



$$\bar{\mathbf{x}} = \frac{1}{T} \sum_{i=1}^T \mathbf{x}_i$$

$$\mathbf{x}_i \rightarrow \mathbf{x}_i - \bar{\mathbf{x}}$$

$$\mathbf{p}_i = \mathbf{u}\mathbf{u}^\top \mathbf{x}_i$$

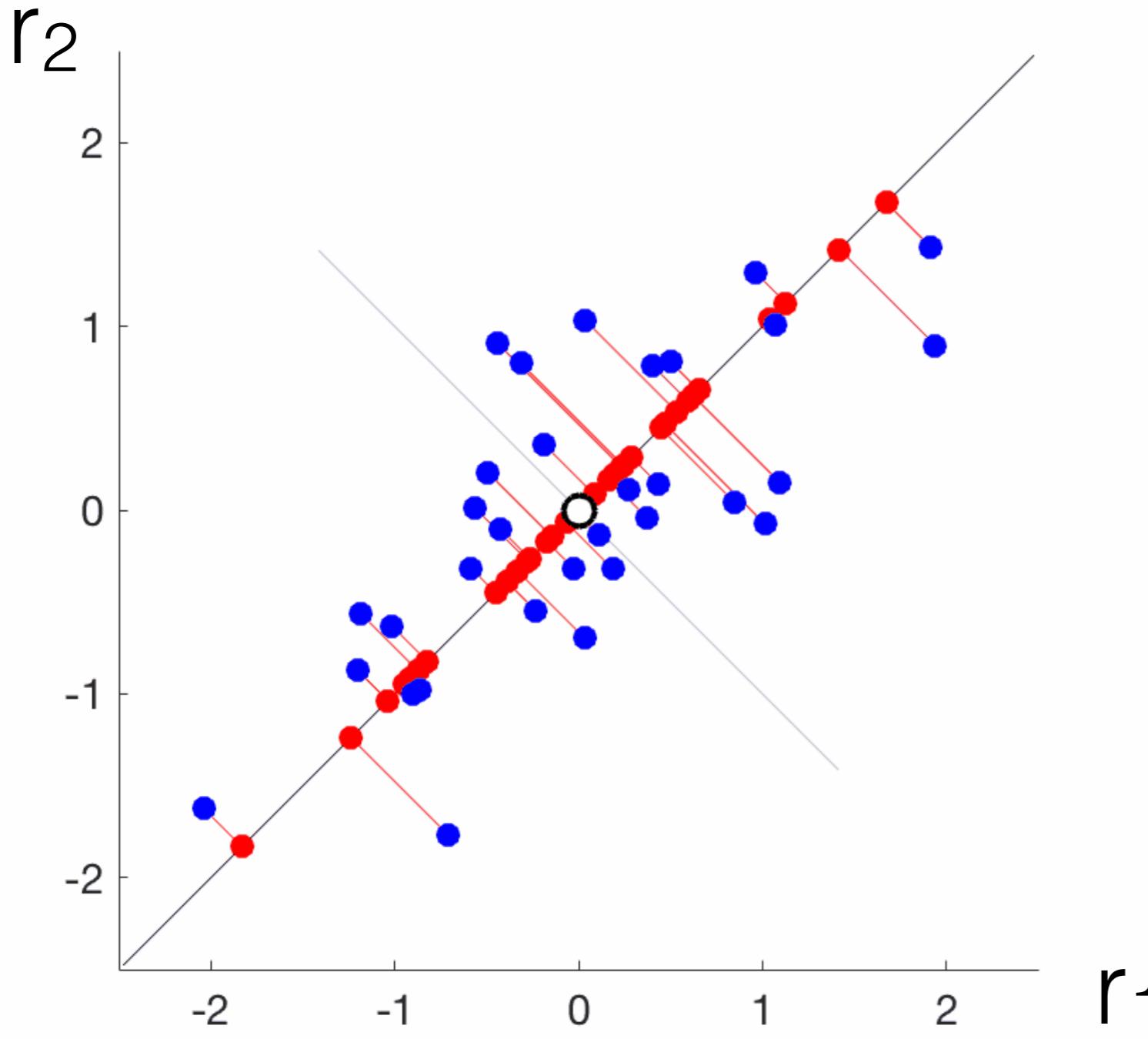
$$L = \sum_{i=1}^T \|\mathbf{x}_i - \mathbf{u}\mathbf{u}^\top \mathbf{x}_i\|^2$$

$$= -\mathbf{u}^\top \left[\sum_{i=1}^T \mathbf{x}_i \mathbf{x}_i^\top \right] \mathbf{u}$$

with $\|\mathbf{u}\| = 1$

Principal component analysis step-by-step

(3) minimum projection error = maximum variance



$$\bar{\mathbf{x}} = \frac{1}{T} \sum_{i=1}^T \mathbf{x}_i$$
$$\mathbf{x}_i \rightarrow \mathbf{x}_i - \bar{\mathbf{x}}$$
$$\mathbf{p}_i = \mathbf{u}\mathbf{u}^\top \mathbf{x}_i$$
$$L = \sum_{i=1}^T \|\mathbf{x}_i - \mathbf{u}\mathbf{u}^\top \mathbf{x}_i\|^2$$
$$= -\mathbf{u}^\top \left[\sum_{i=1}^T \mathbf{x}_i \mathbf{x}_i^\top \right] \mathbf{u}$$

with $\|\mathbf{u}\| = 1$

Technical solution

$$L = \mathbf{u}^\top \left[\sum_{i=1}^T \mathbf{x}_i \mathbf{x}_i^\top \right] \mathbf{u}$$

with $\|\mathbf{u}\| = 1$

covariance matrix (centered data): $\mathbf{C} = \frac{1}{T} \sum_{i=1}^T \mathbf{x}_i \mathbf{x}_i^T$



Technical solution

$$L = \mathbf{u}^\top \mathbf{C} \mathbf{u} \quad \text{with } \|\mathbf{u}\| = 1$$

↑

covariance matrix (centered data): $\mathbf{C} = \frac{1}{T} \sum_{i=1}^T \mathbf{x}_i \mathbf{x}_i^T$

Technical solution

$$L = \mathbf{u}^\top \mathbf{C} \mathbf{u} \quad \text{with } \|\mathbf{u}\| = 1$$

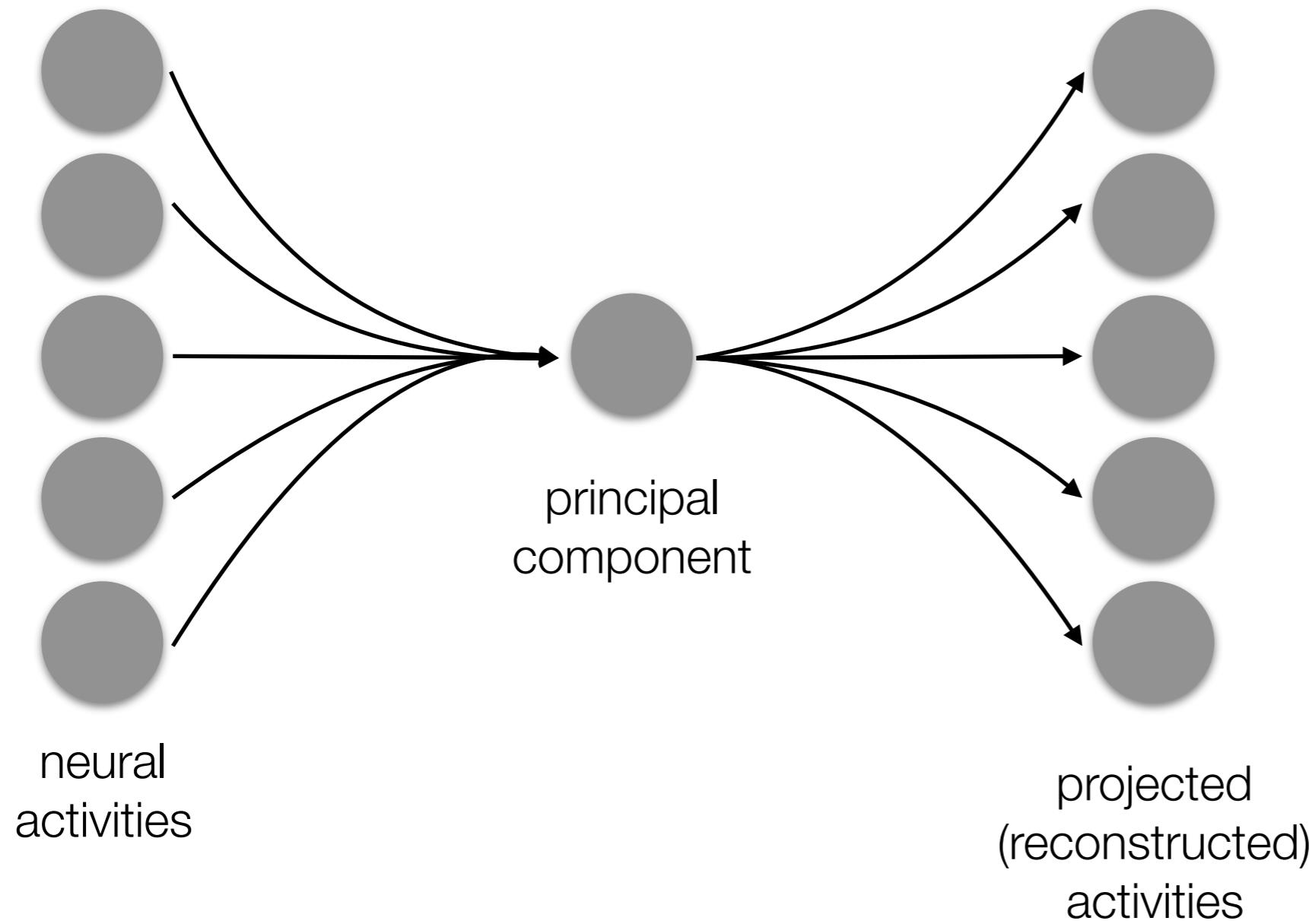
covariance matrix (centered data): $\mathbf{C} = \frac{1}{T} \sum_{i=1}^T \mathbf{x}_i \mathbf{x}_i^T$

Solution:

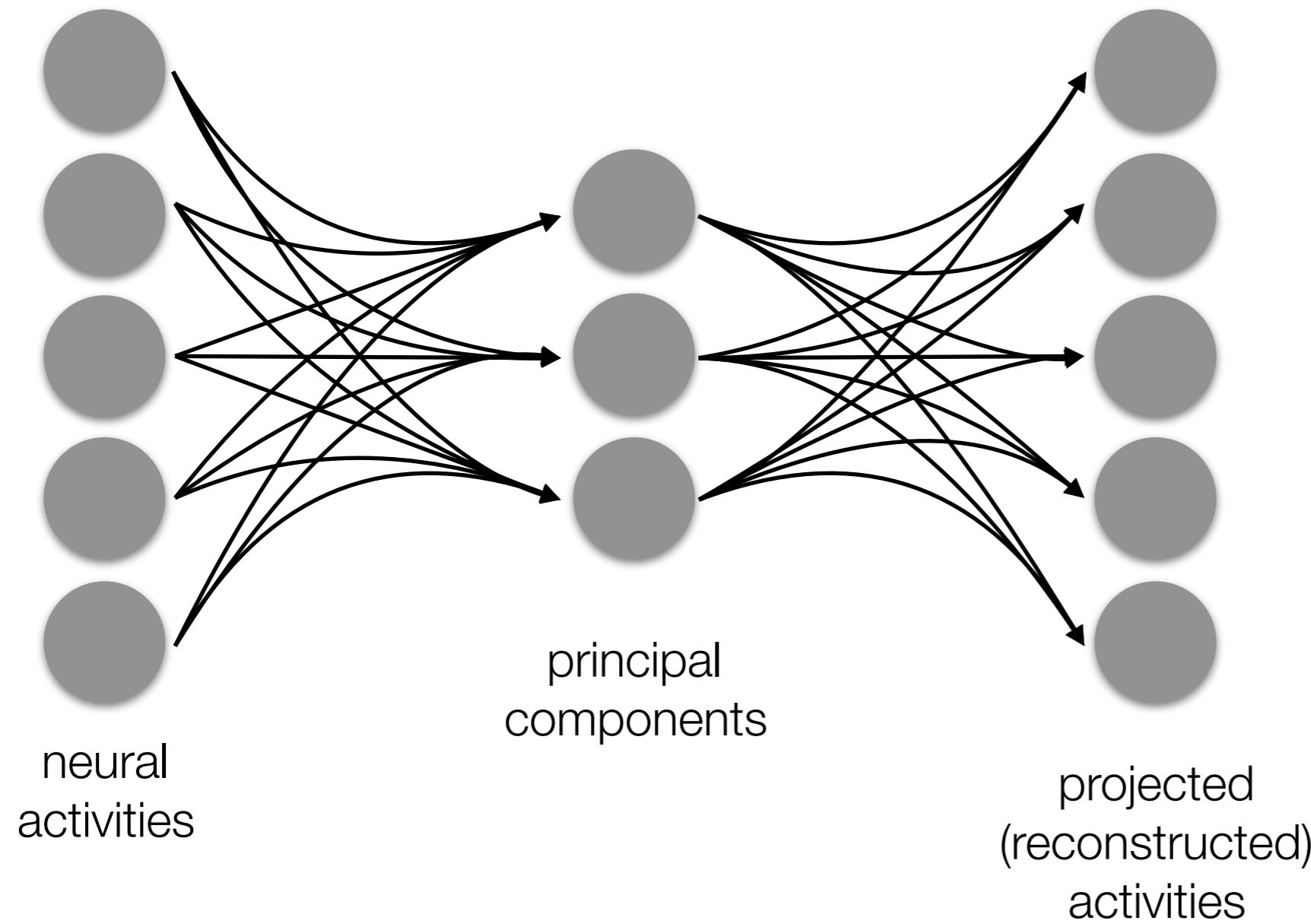
$$\mathbf{C}\mathbf{u} = \lambda\mathbf{u}$$

Nice: solution allows you also to find subspaces with dimensions > 1

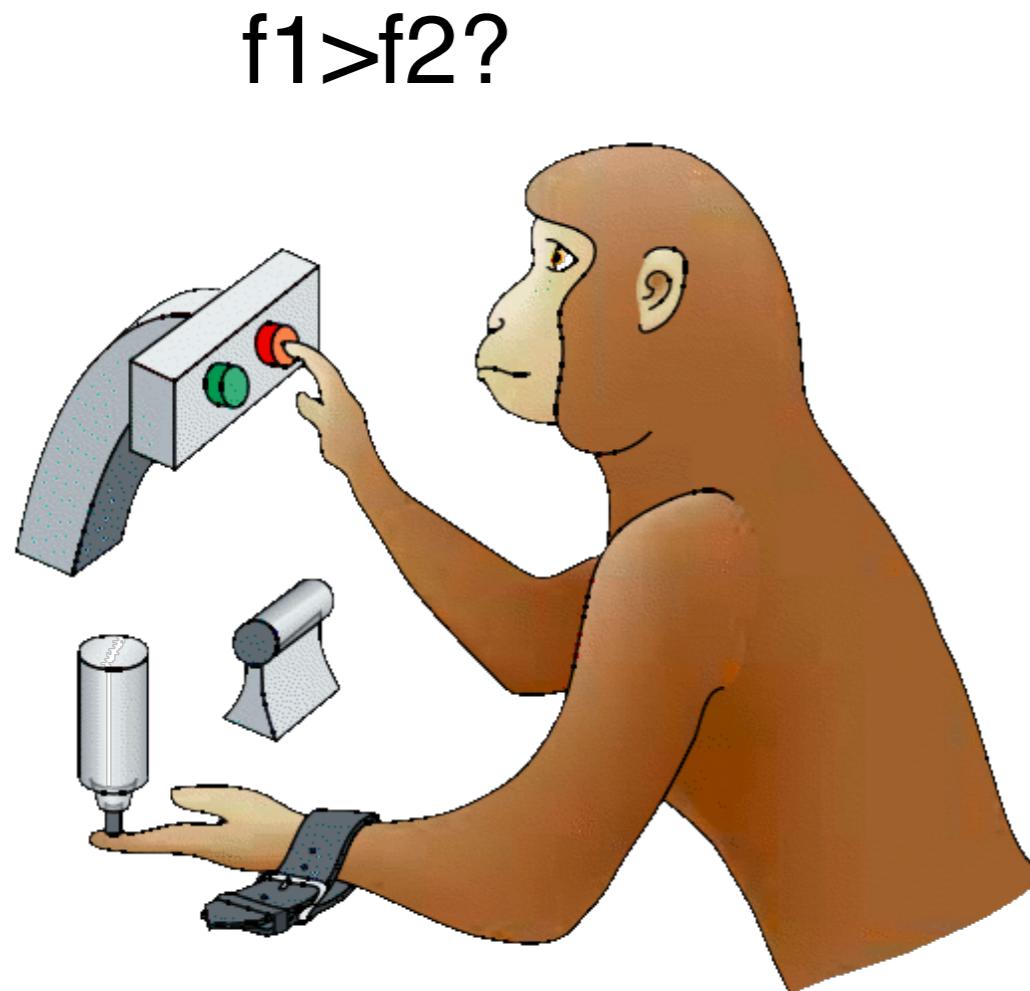
PCA as a bottleneck



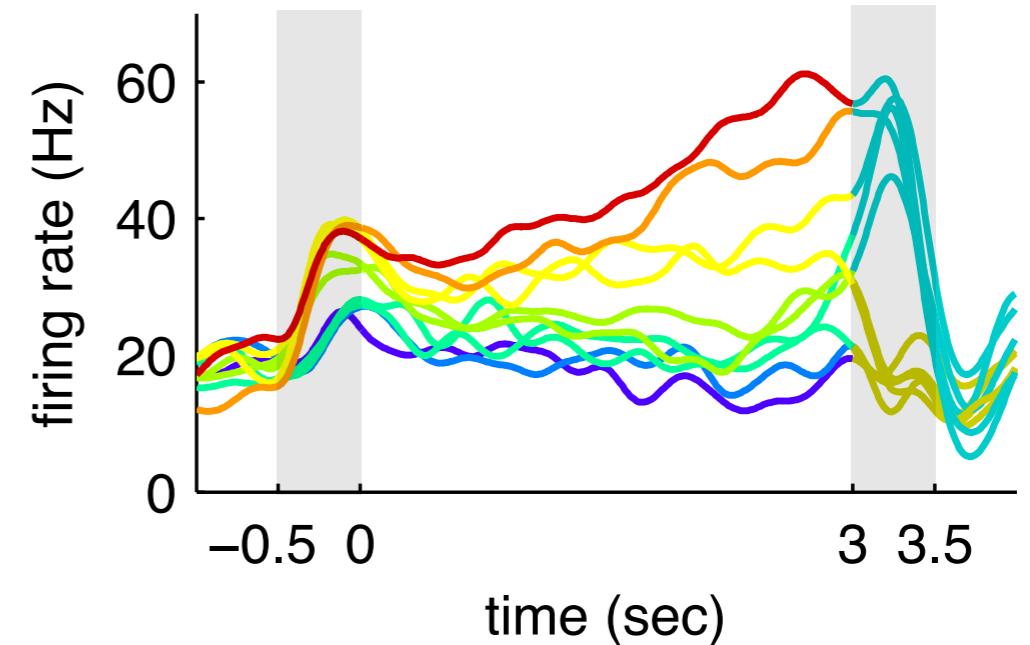
PCA as a bottleneck (beyond 1d-subspaces)



Application: PFC working memory task



PFC example neuron (Romo et al, 1999)



task parameters

stimulus f1



decision



Yes No

Population Data

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

time 1 time 2 time 3 time T

neuron 1
neuron 2
neuron N

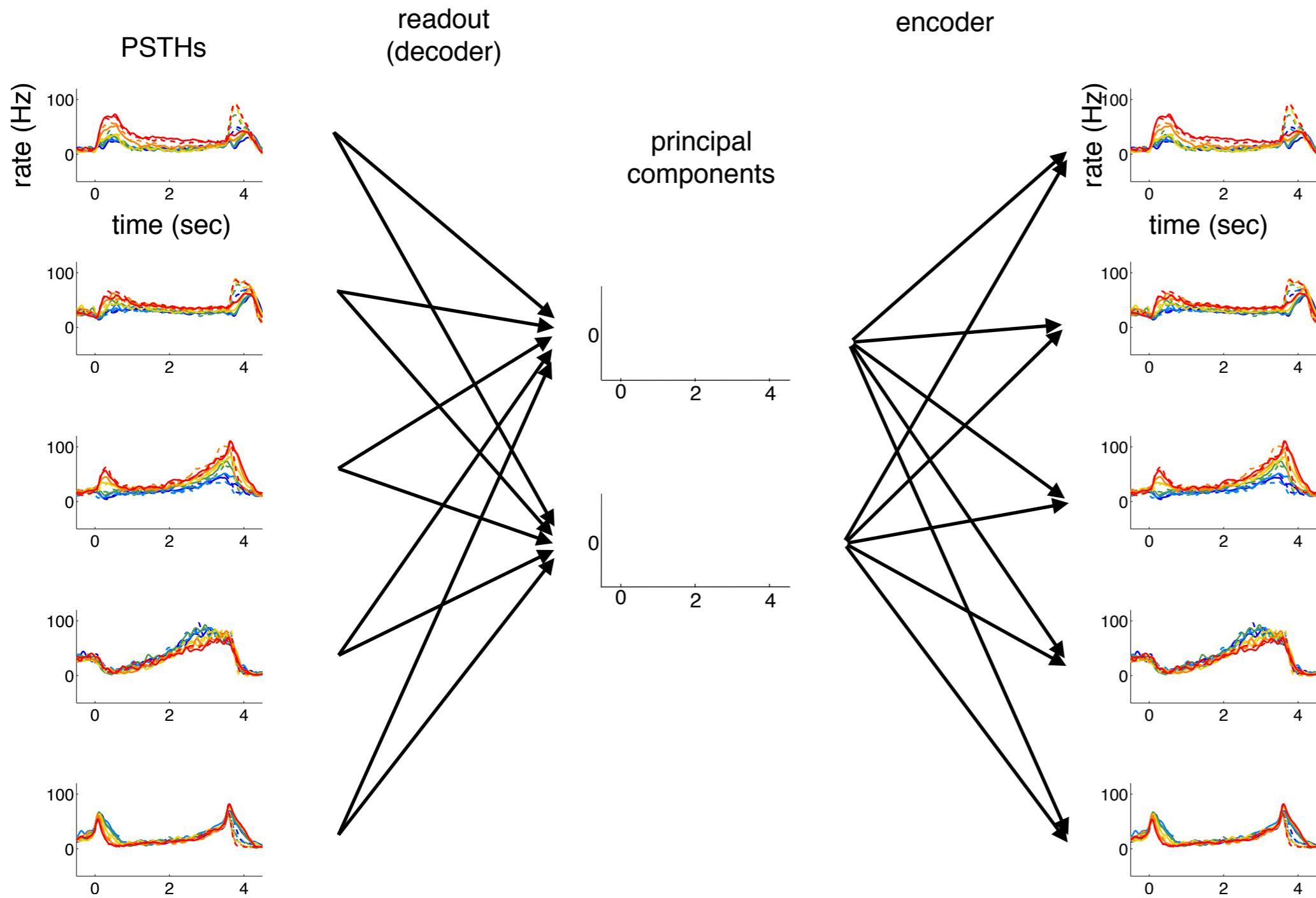
Population Data, organized by task parameters

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

neuron 1
neuron 2
neuron N

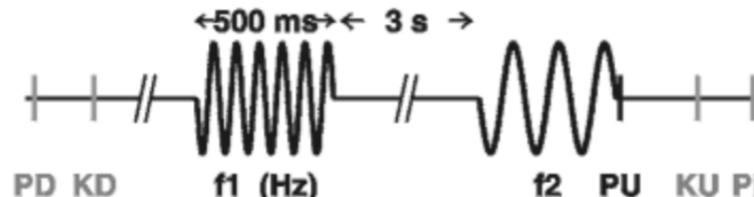
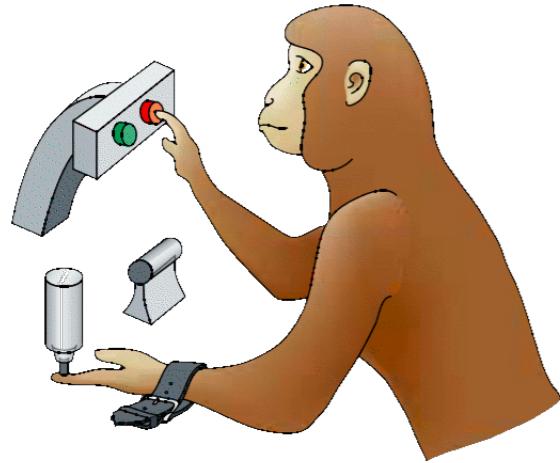
condition 1 condition 2 ...

PCA in a nutshell

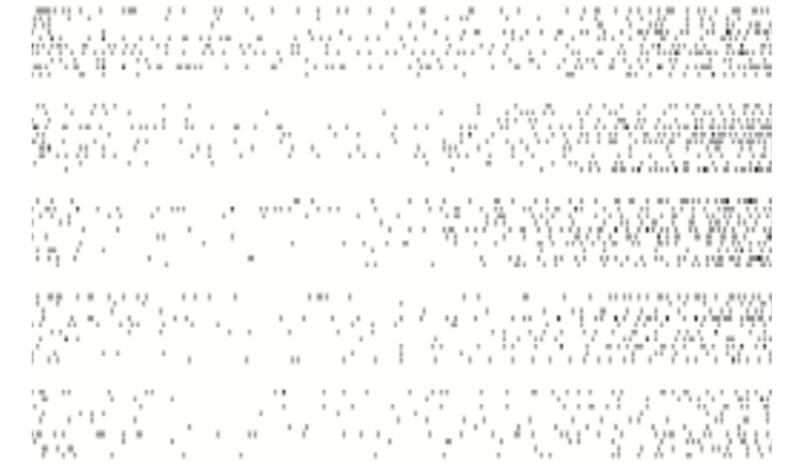


Exercise:

PCA on Romo data



f1>f2?

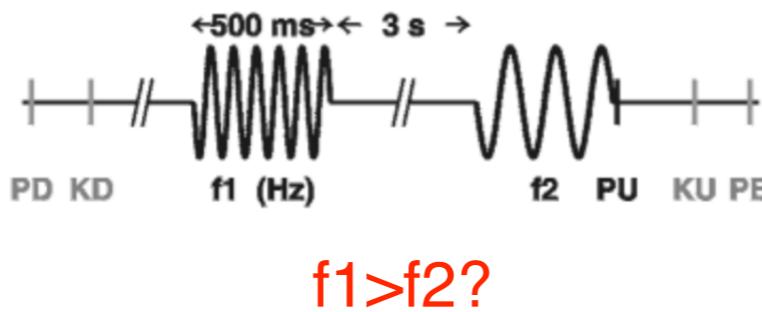
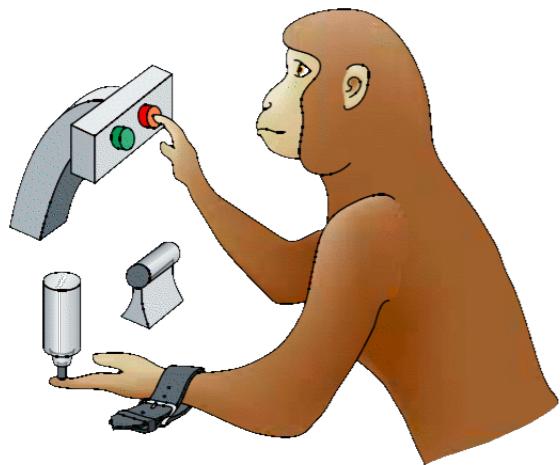


To-do-list

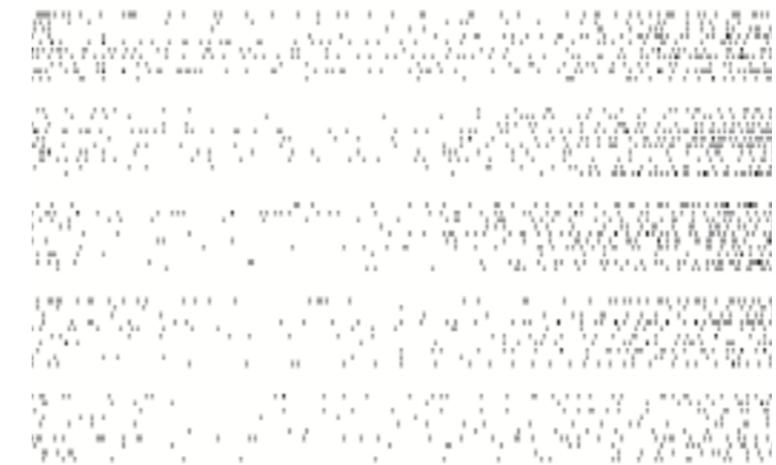
- (1) load data 'romo_allpsths.mat'
- (2) reshape data array
 $X = [\# \text{neurons}] \times [\# \text{conditions}] \times [\# \text{time points}]$
into data matrix
 $X = [\# \text{neurons}] \times [\# \text{conditions} \times \# \text{time points}]$
- (3) compute and plot data matrix X
- (4) 'center' the data
- (5) compute and plot covariance matrix
- (6) determine eigenvalues and eigenvectors of this matrix
- (7) plot eigenvalues
- (8) compute and plot the first principal components

Exercise:

Solution files



f1>f2?



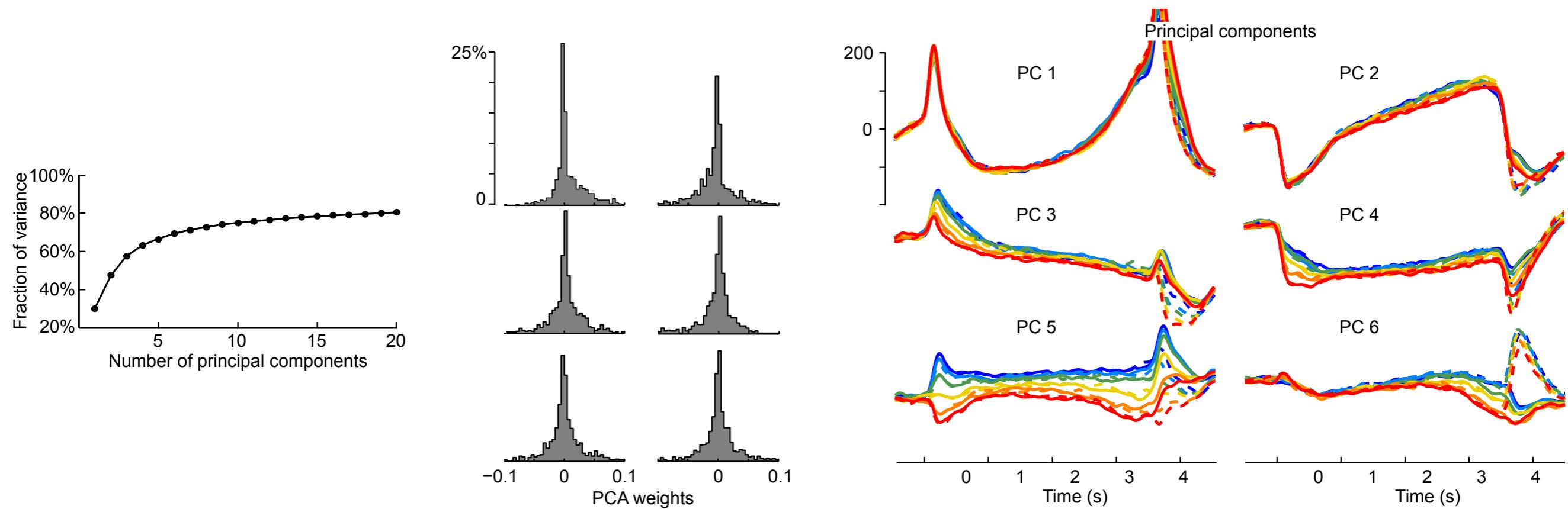
Matlab scripts ...

- (1) romo_allpsths
- (2) romo_pca

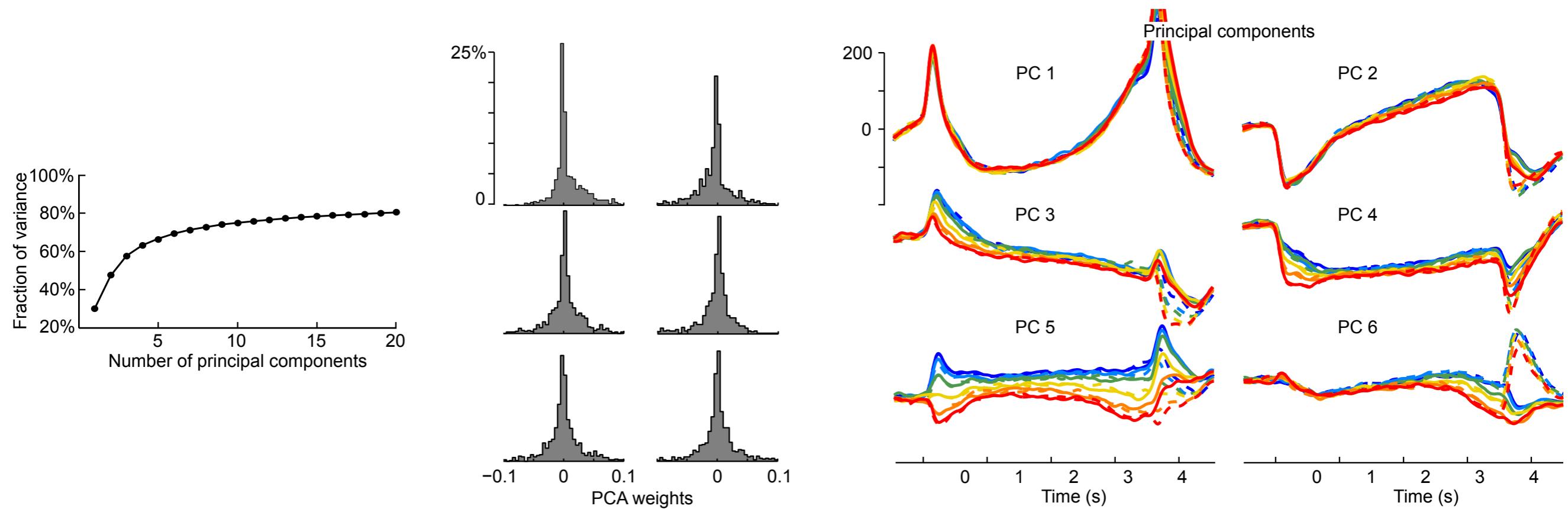
... and what they do

- extract psths of all cells in data folder rr014
- compute PCA on data matrix X

Application: PFC working memory task



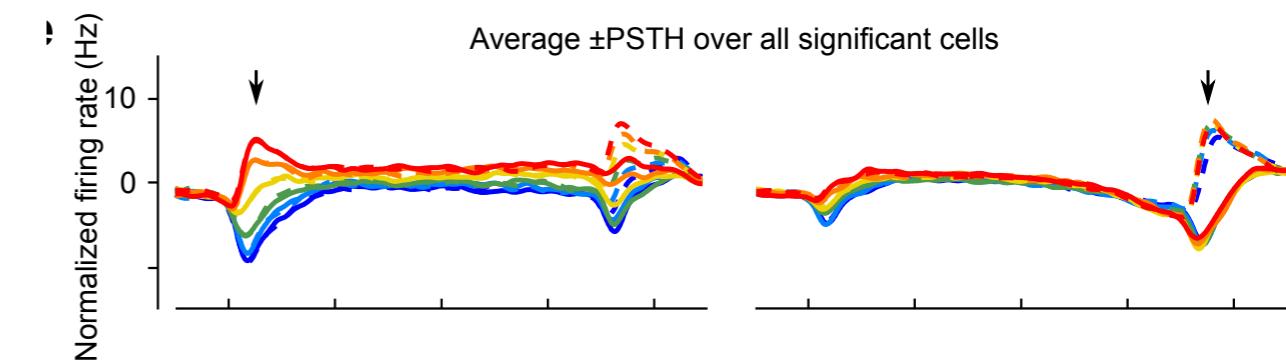
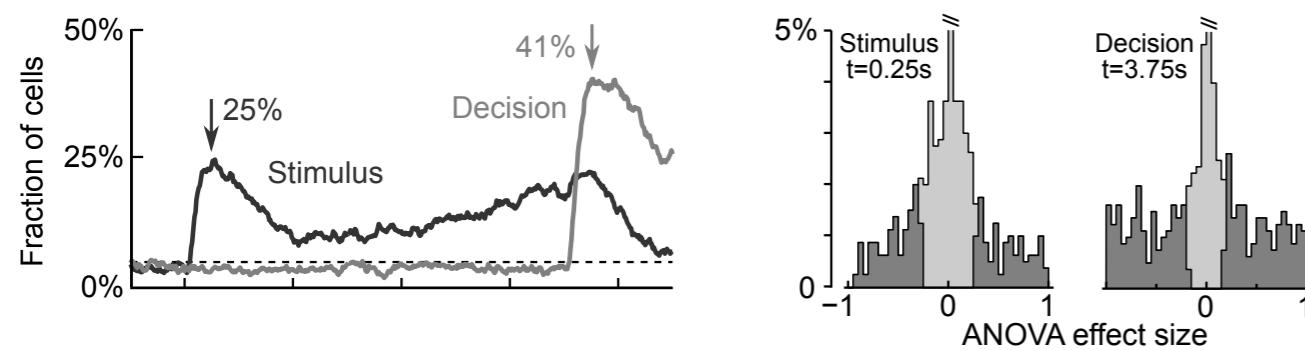
Application: PFC working memory task



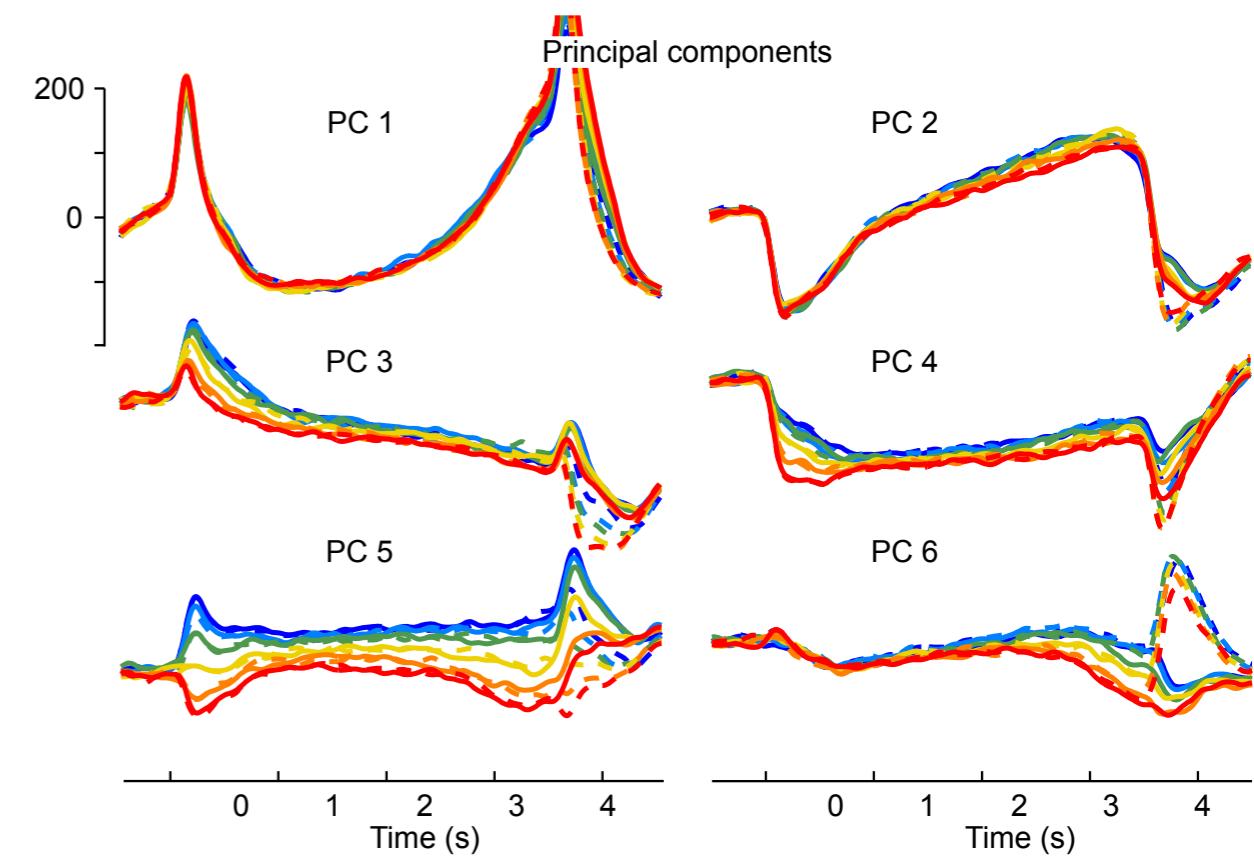
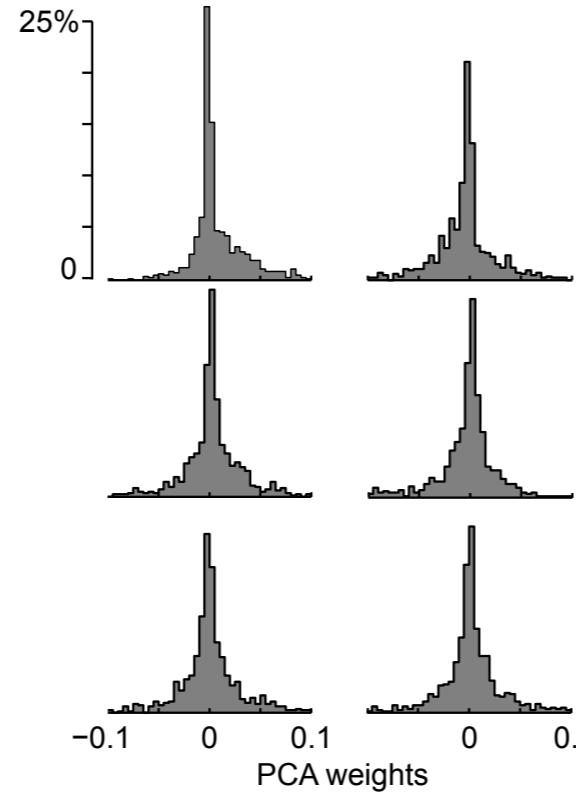
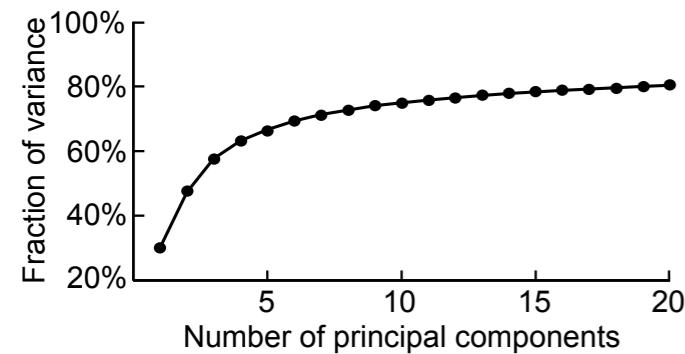
... so wait ... what did we learn?

'Looking at the data': Classical single-cell approach vs. PCA

Sorting cells via ANOVA

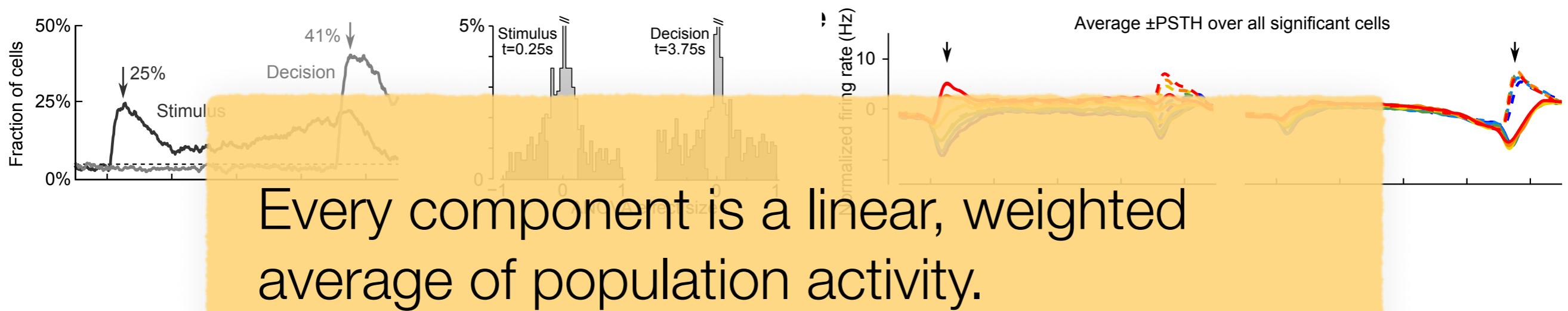


Principal Component Analysis



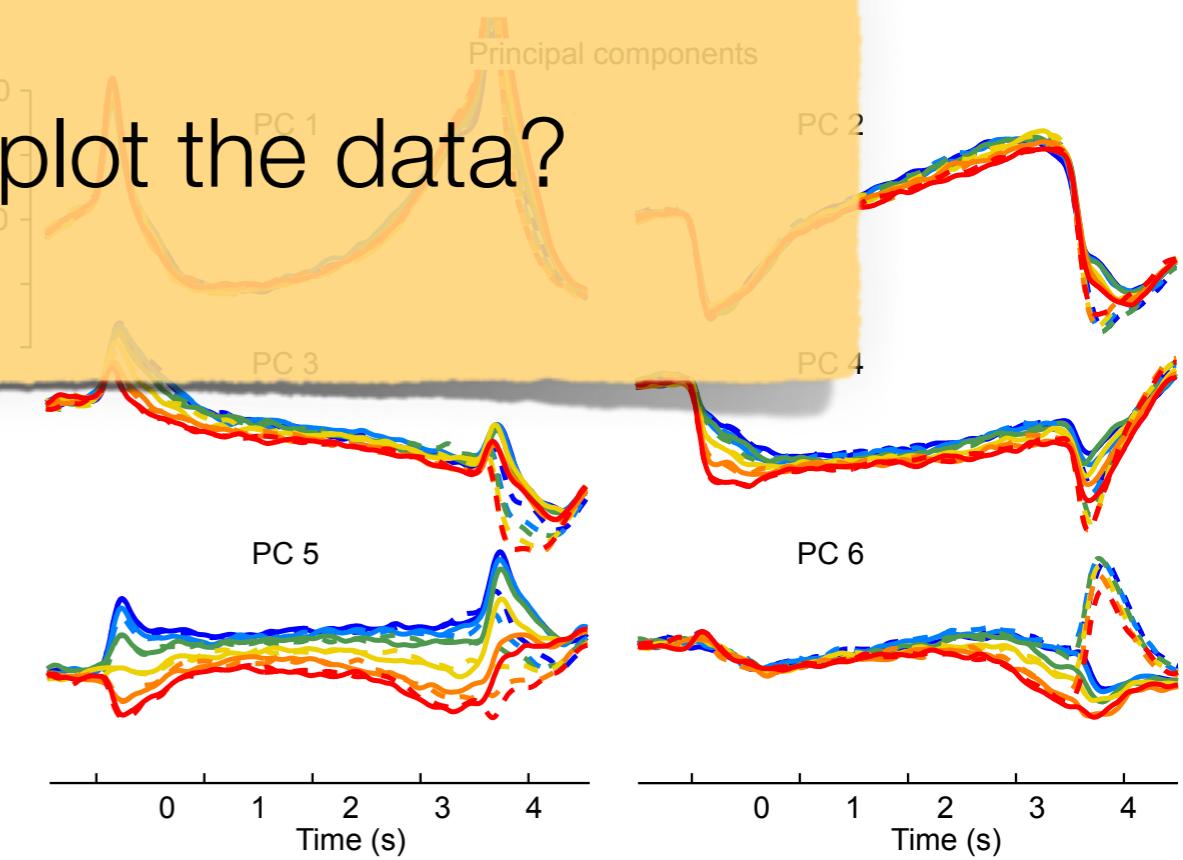
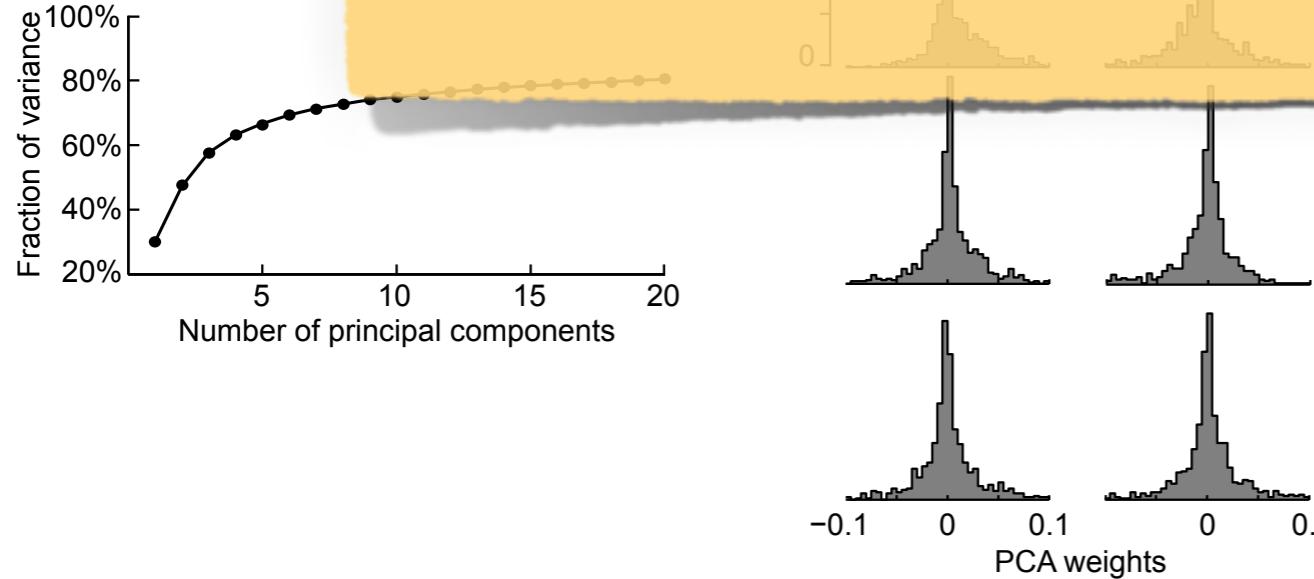
'Looking at the data': Classical single-cell approach vs. PCA

Sorting cells via ANOVA



Principal Component Analysis

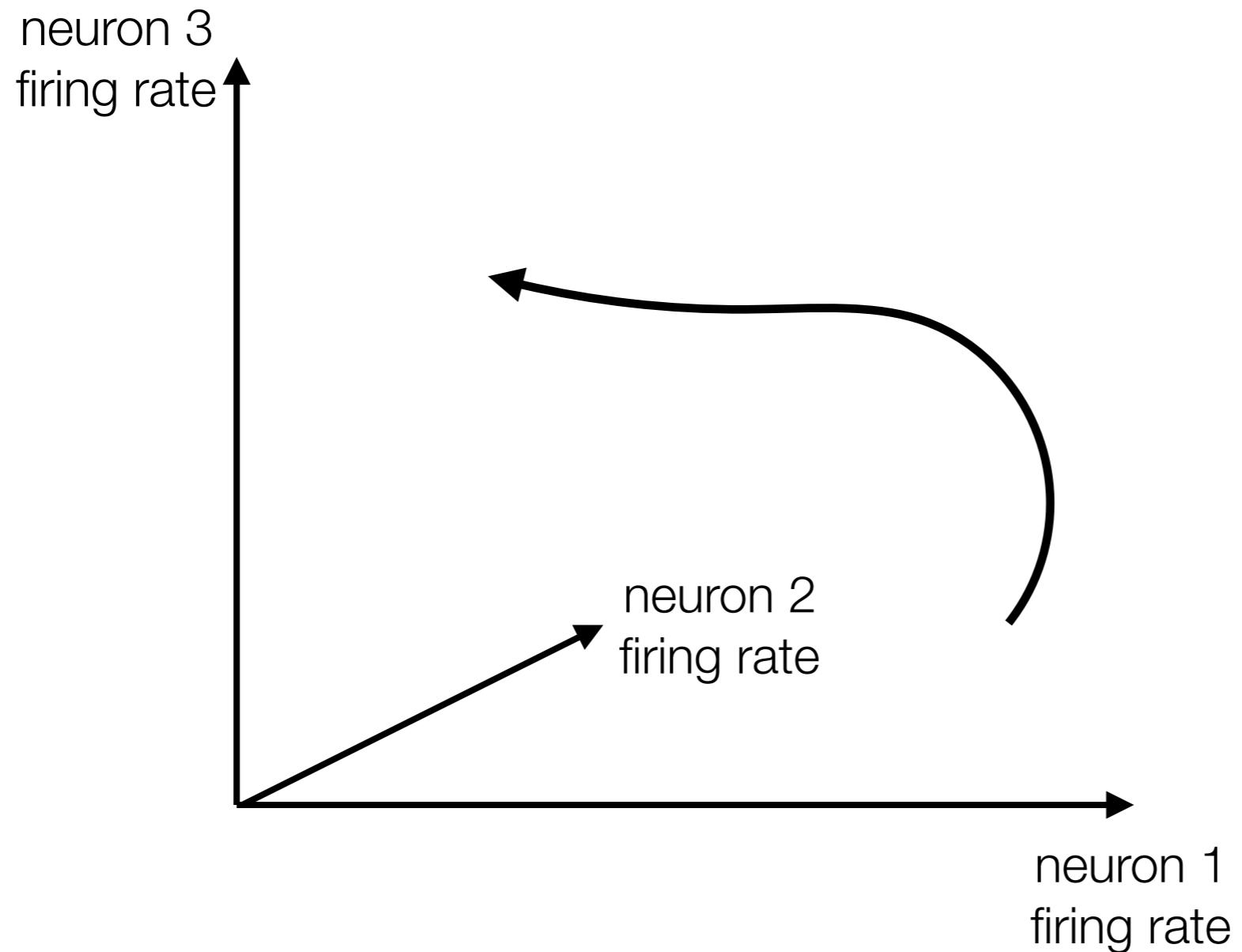
What's the 'best' way to plot the data?



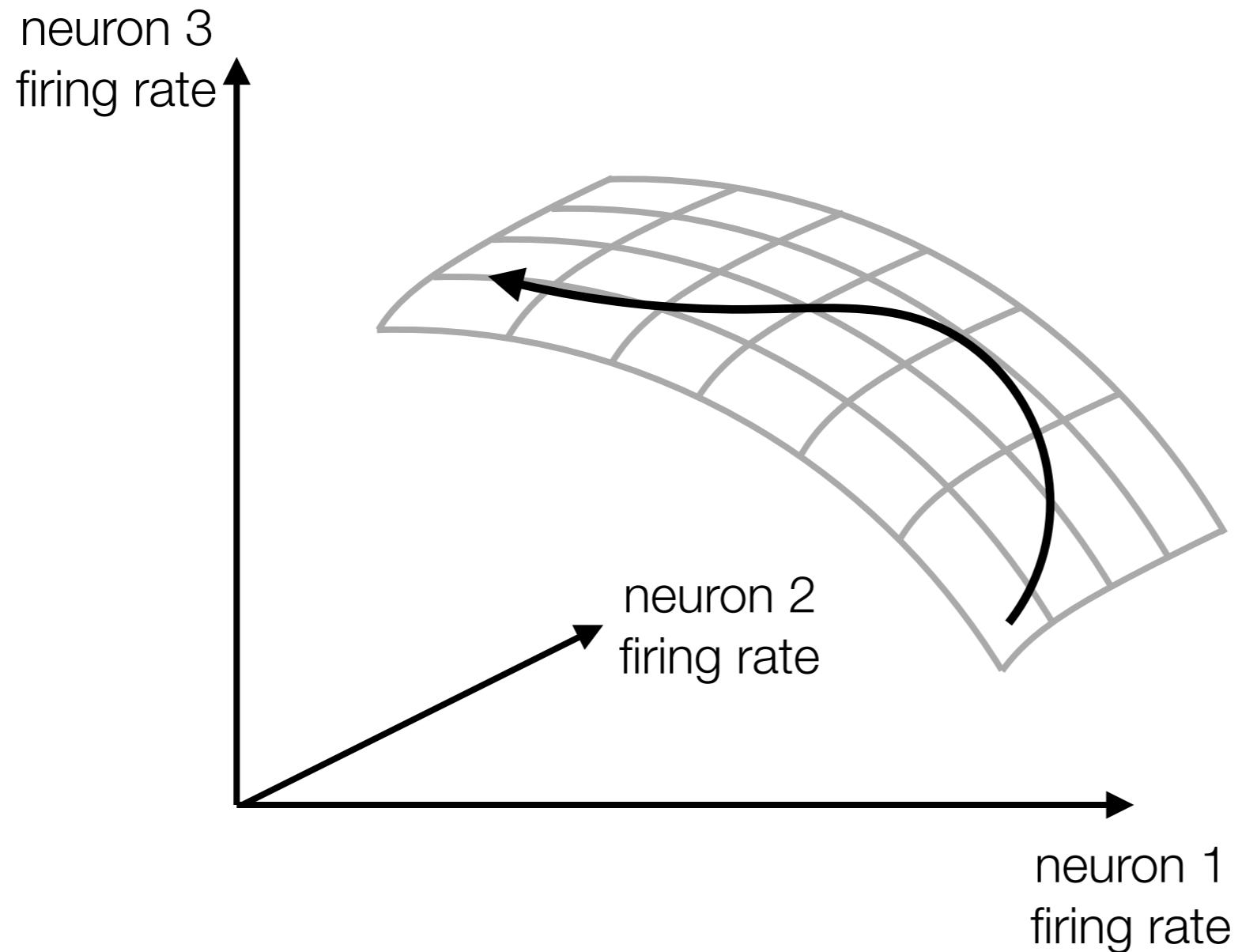
How to ‘look’ at the data (3)

Demixed principal component analysis

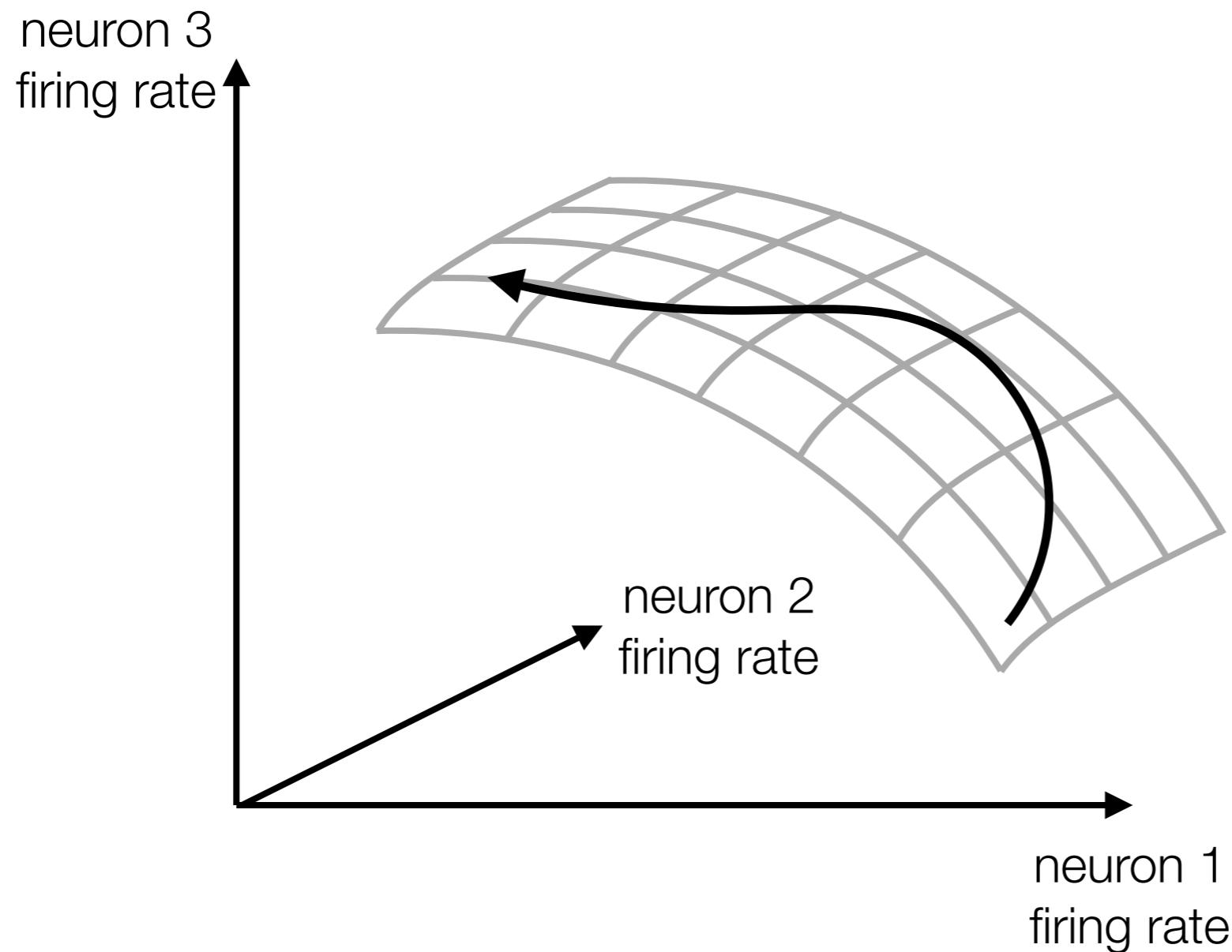
State space embedding and ‘neural manifolds’



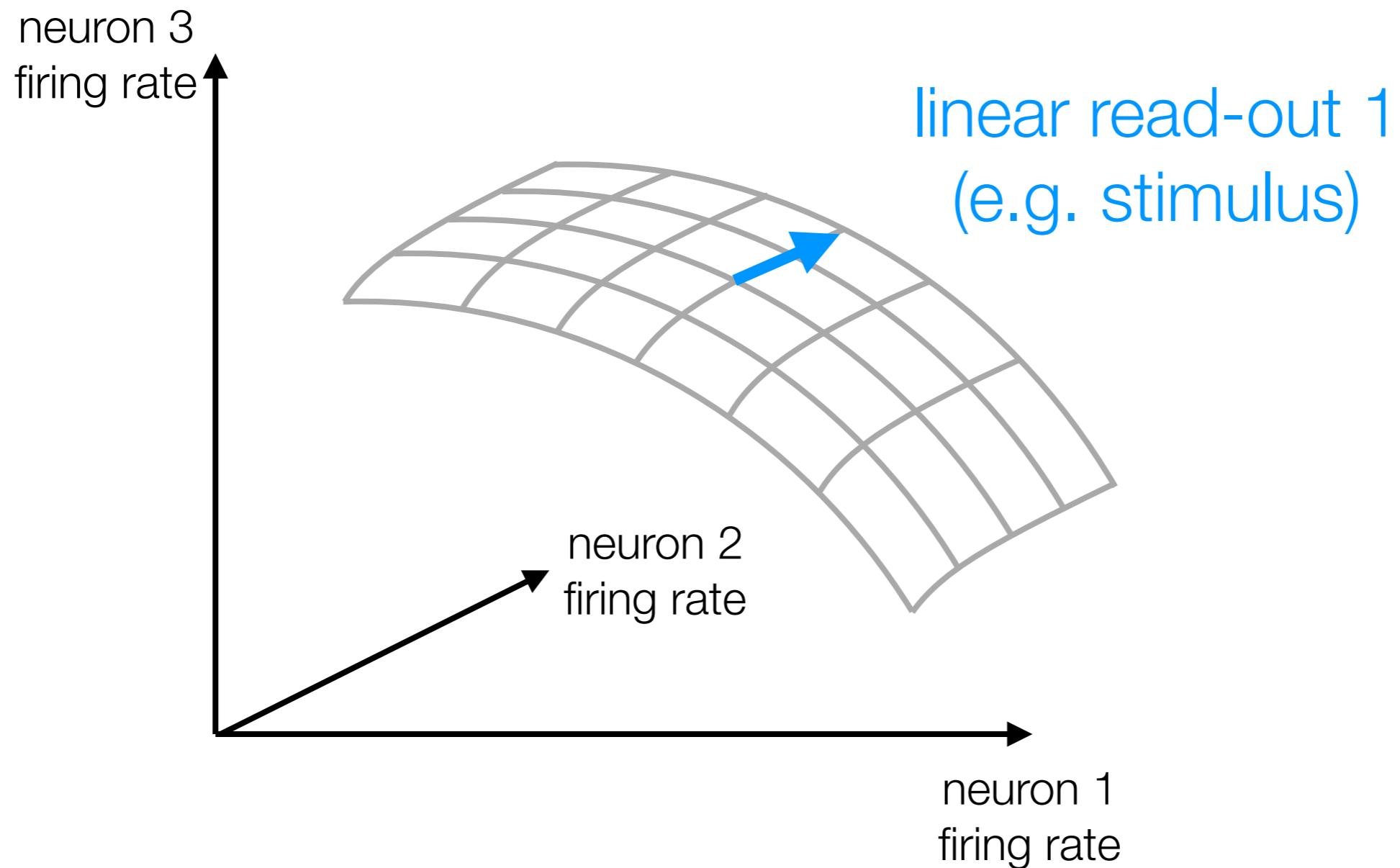
State space embedding and ‘neural manifolds’



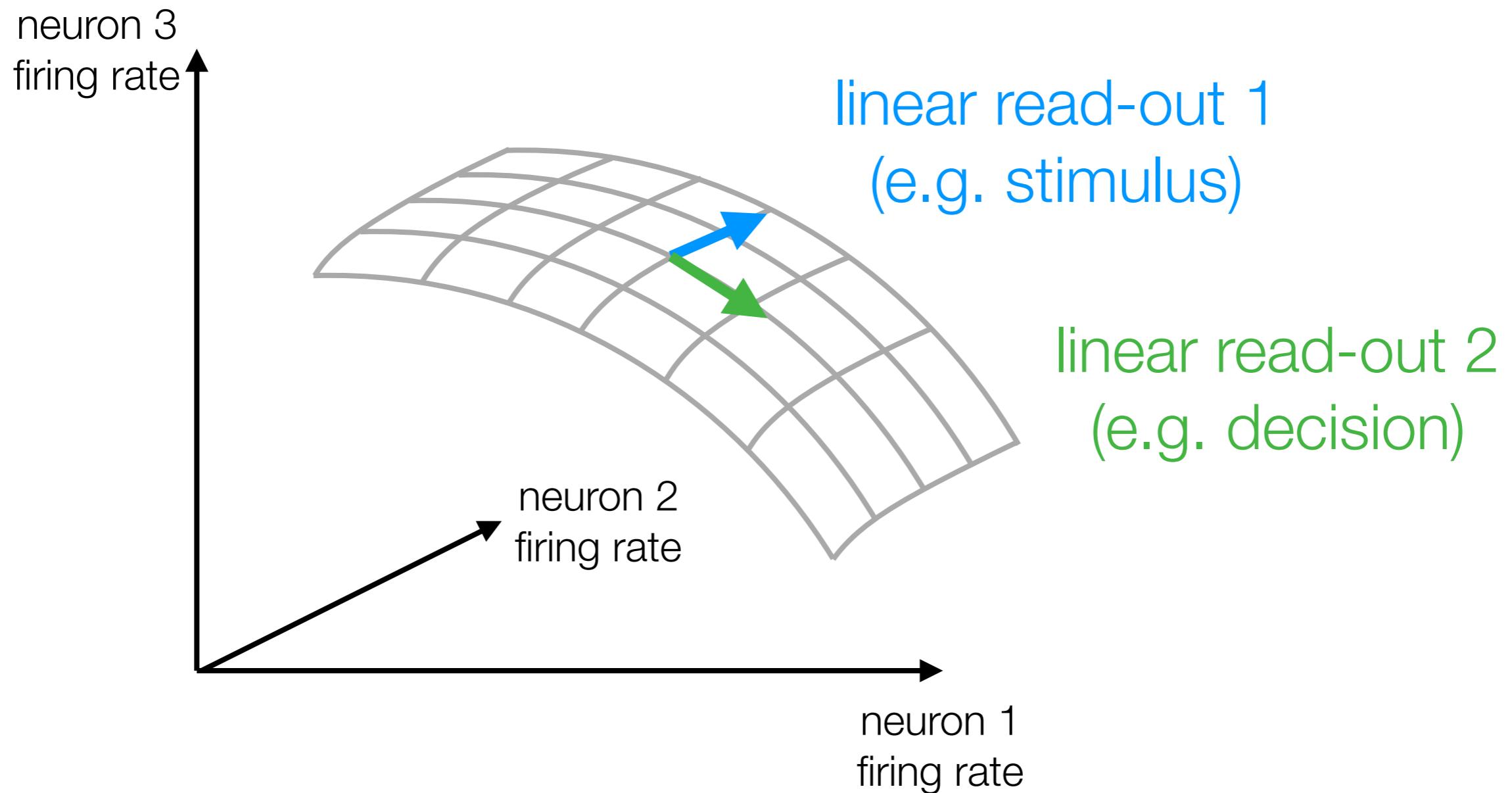
Can we assign meaning to the manifold?



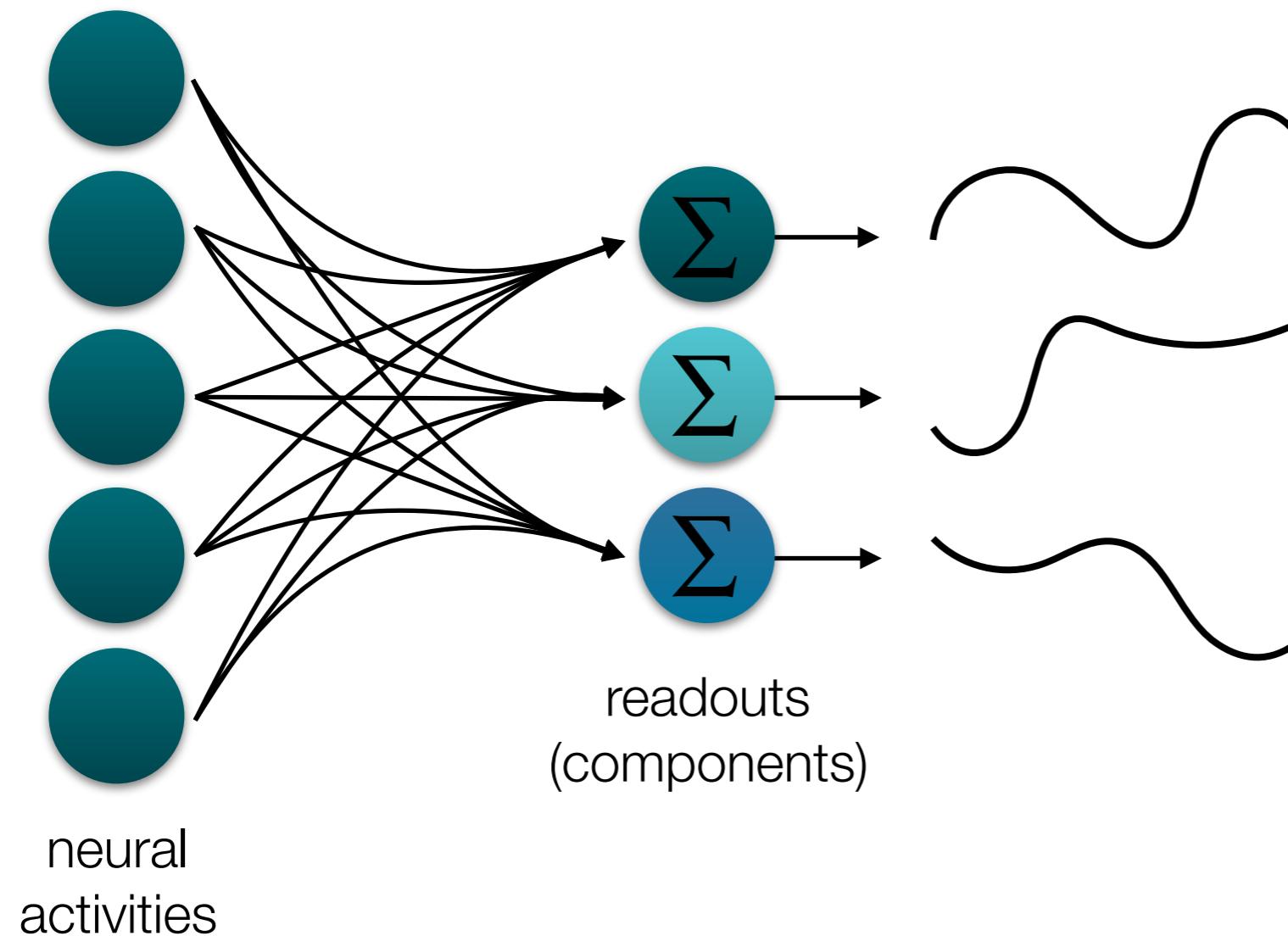
Movement on manifold = change in linear read-out



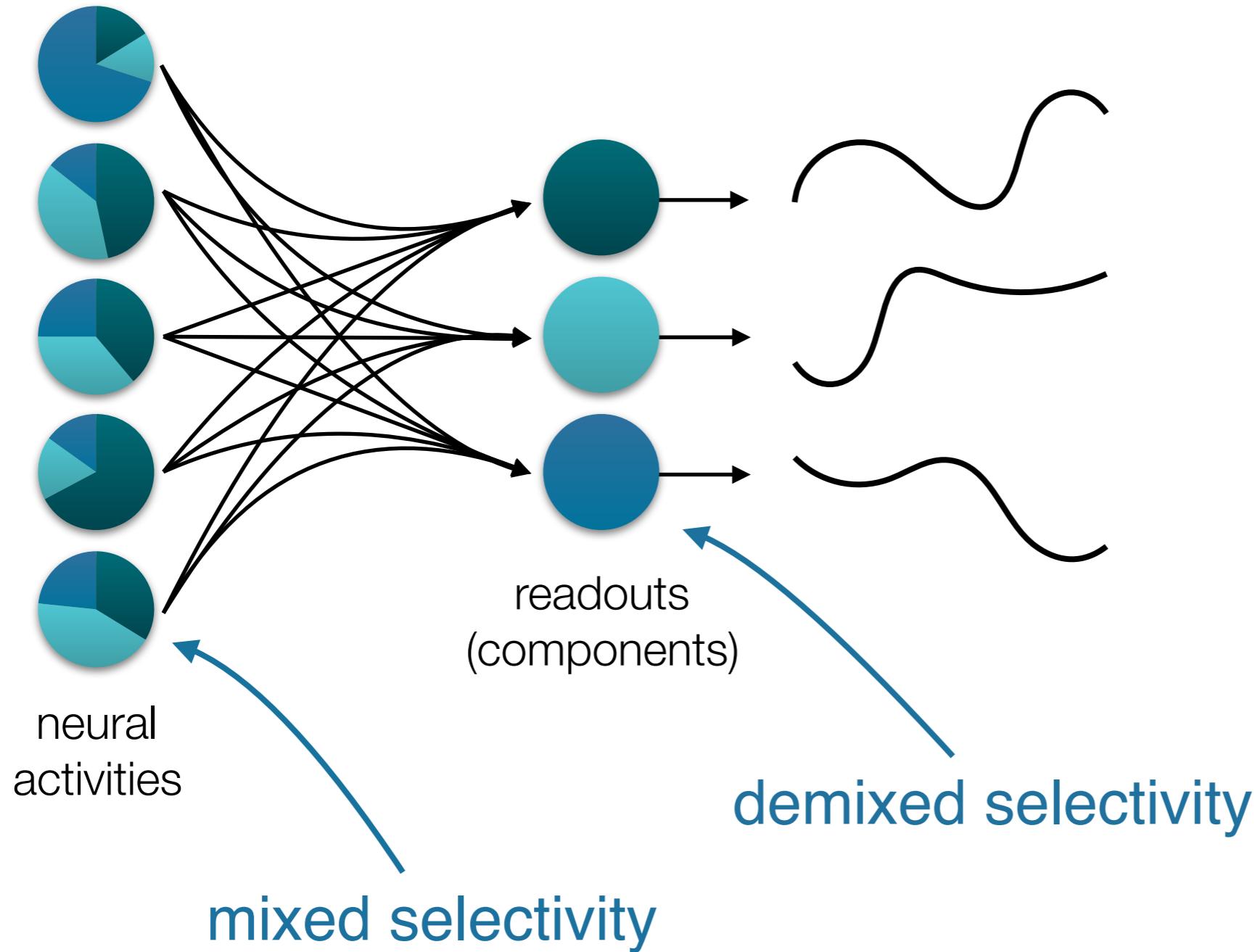
Movement on manifold = change in linear read-out



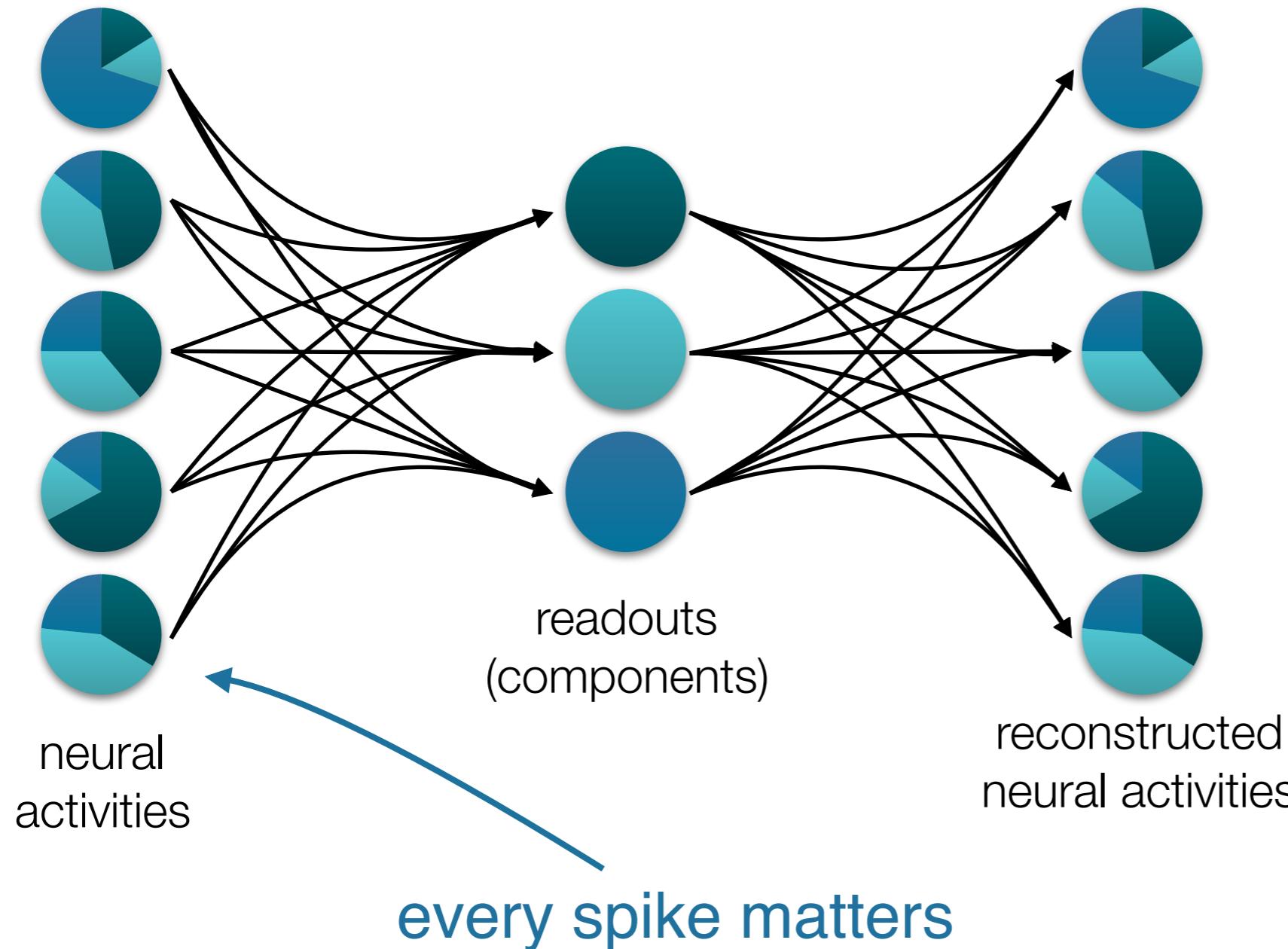
All approaches are based on multiple linear readouts (=components) from the population



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

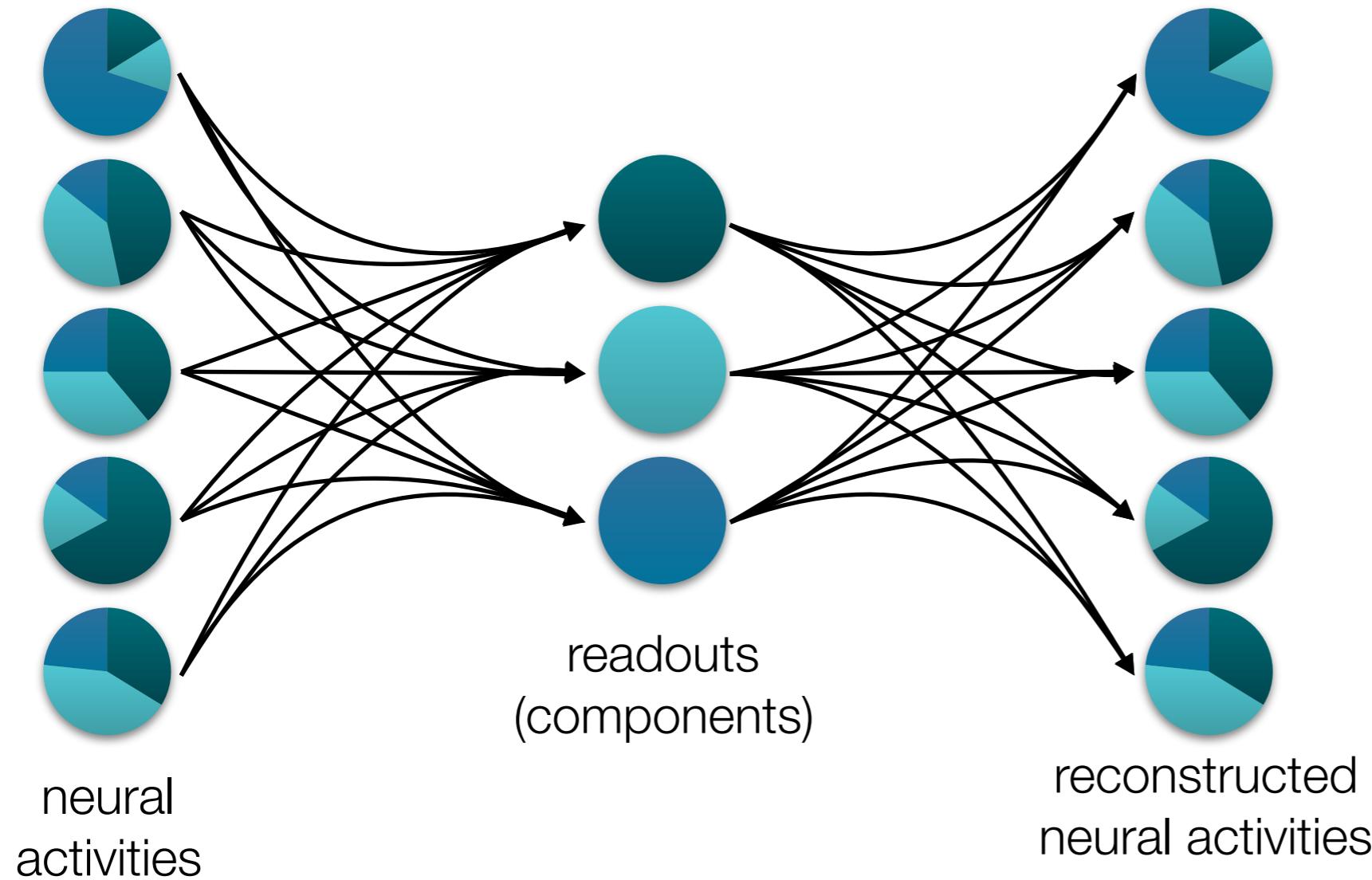


Our goal: choose readouts that
(2) and capture all aspects of activity

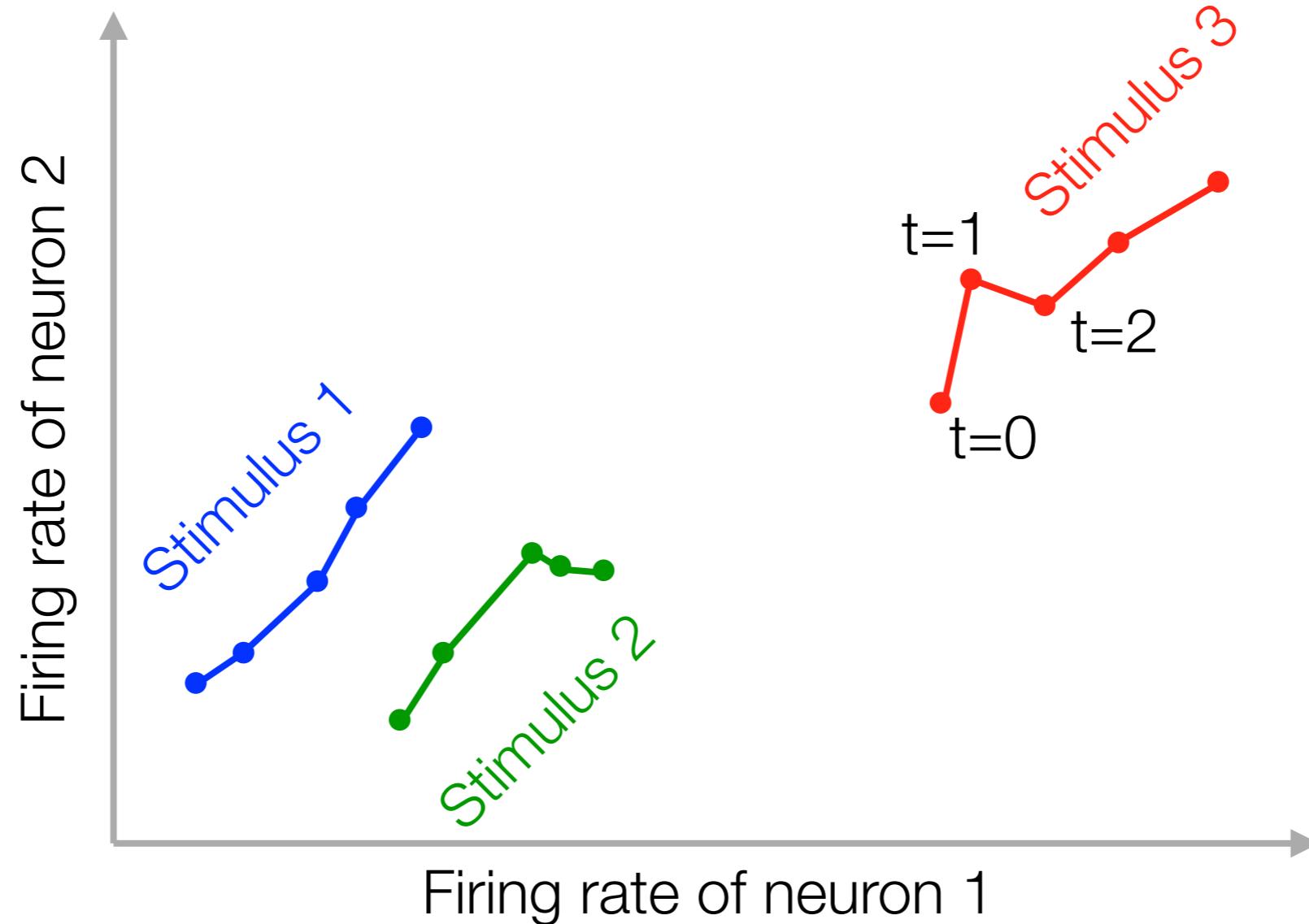
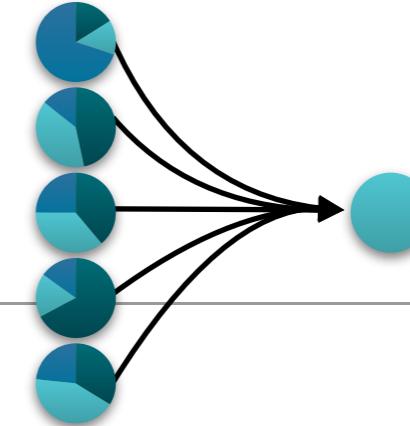


'Demixed' principal component analysis

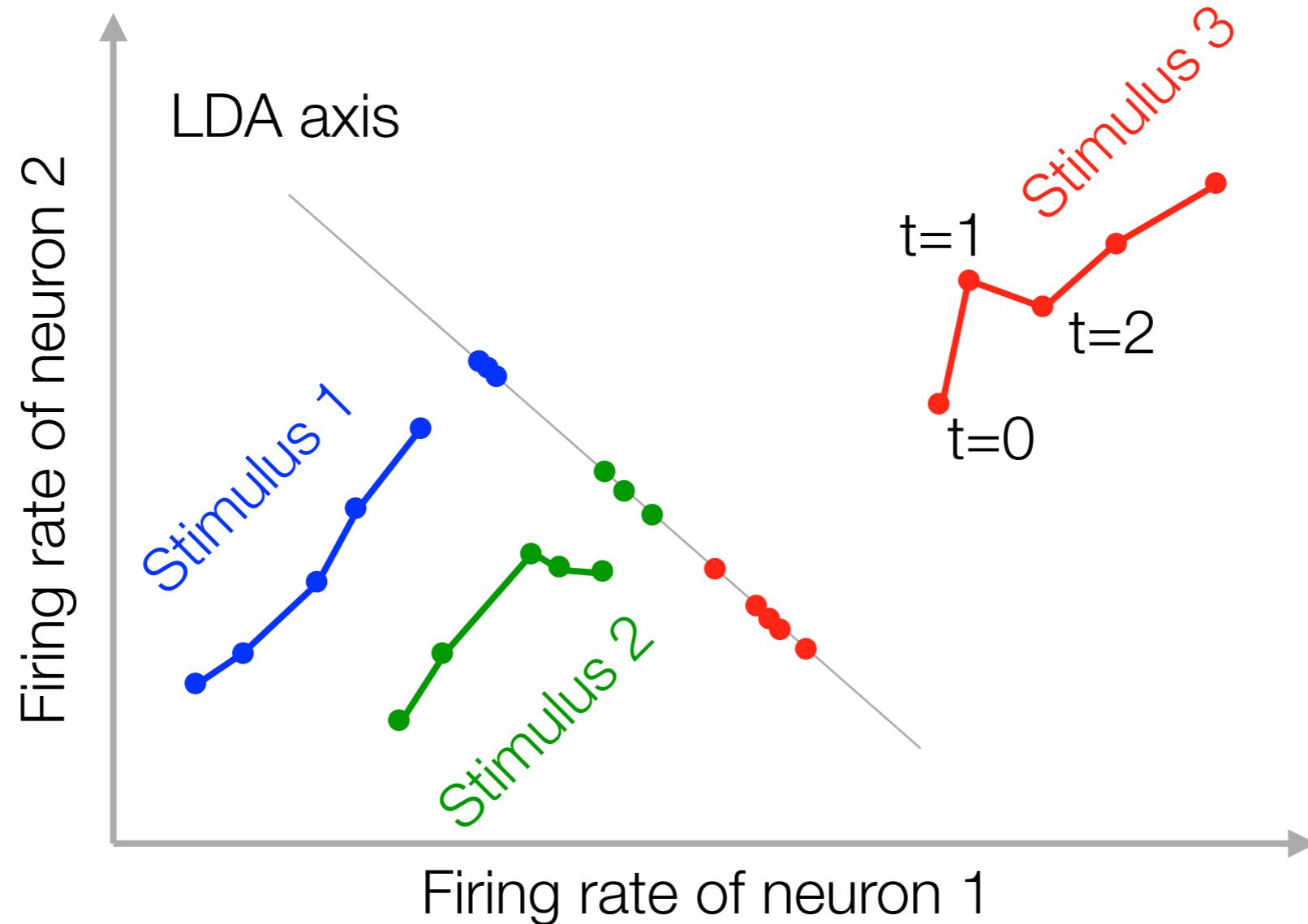
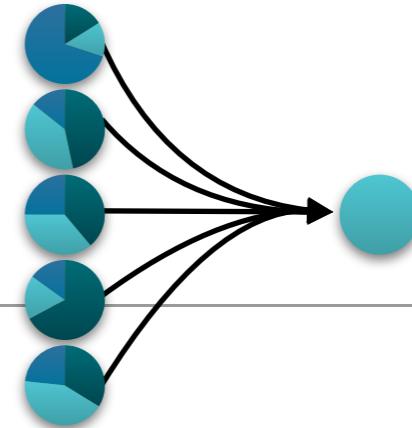
(Kobak, Brendel et al, eLife, 2016; older version in NIPS 2011)



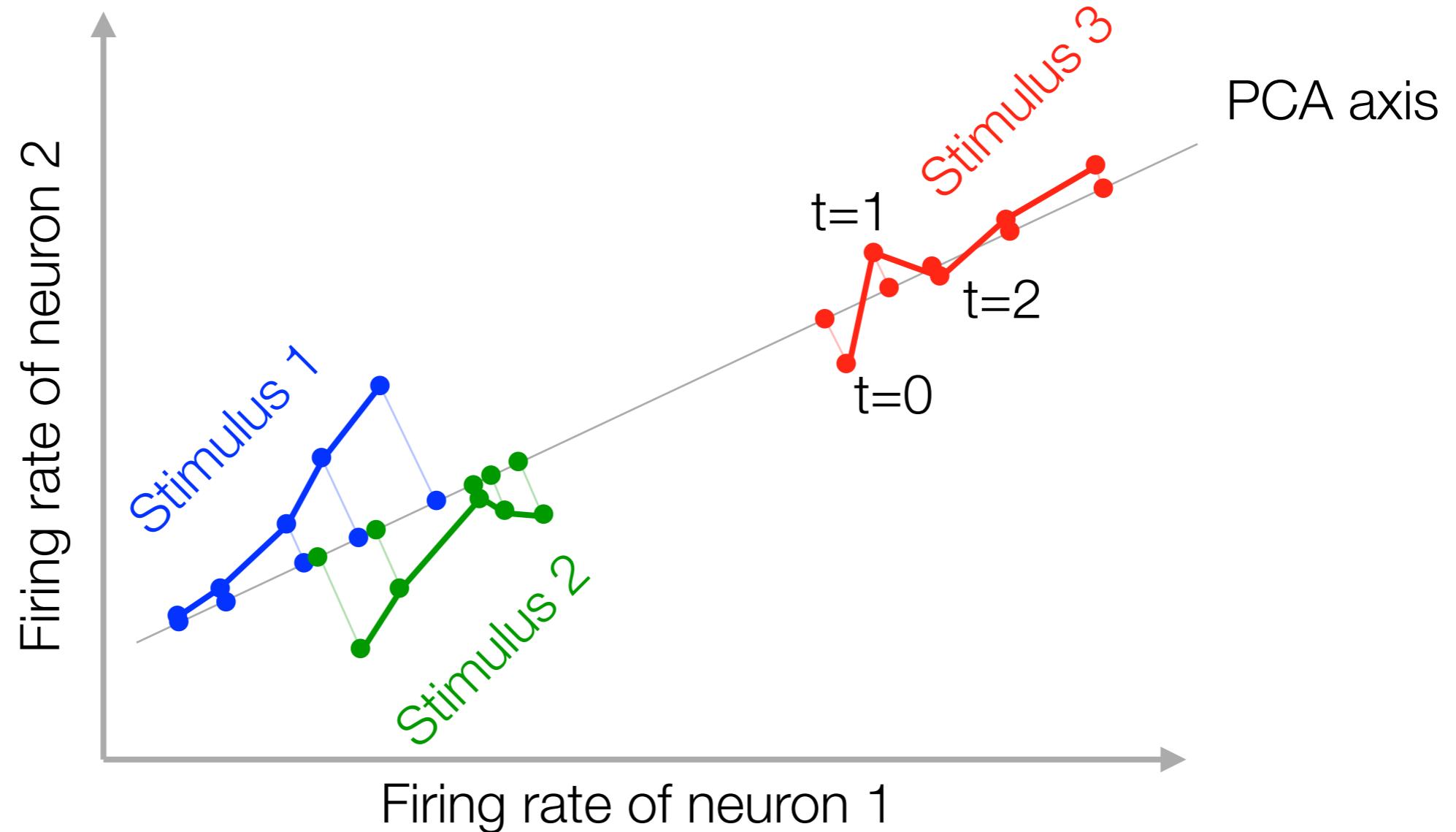
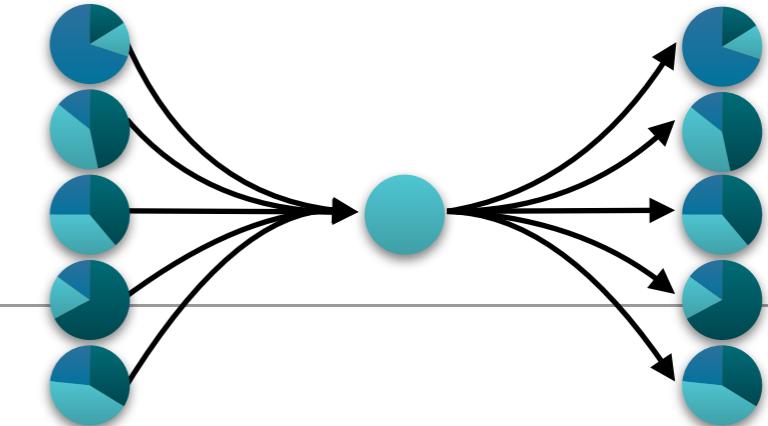
Linear readouts revisited ...



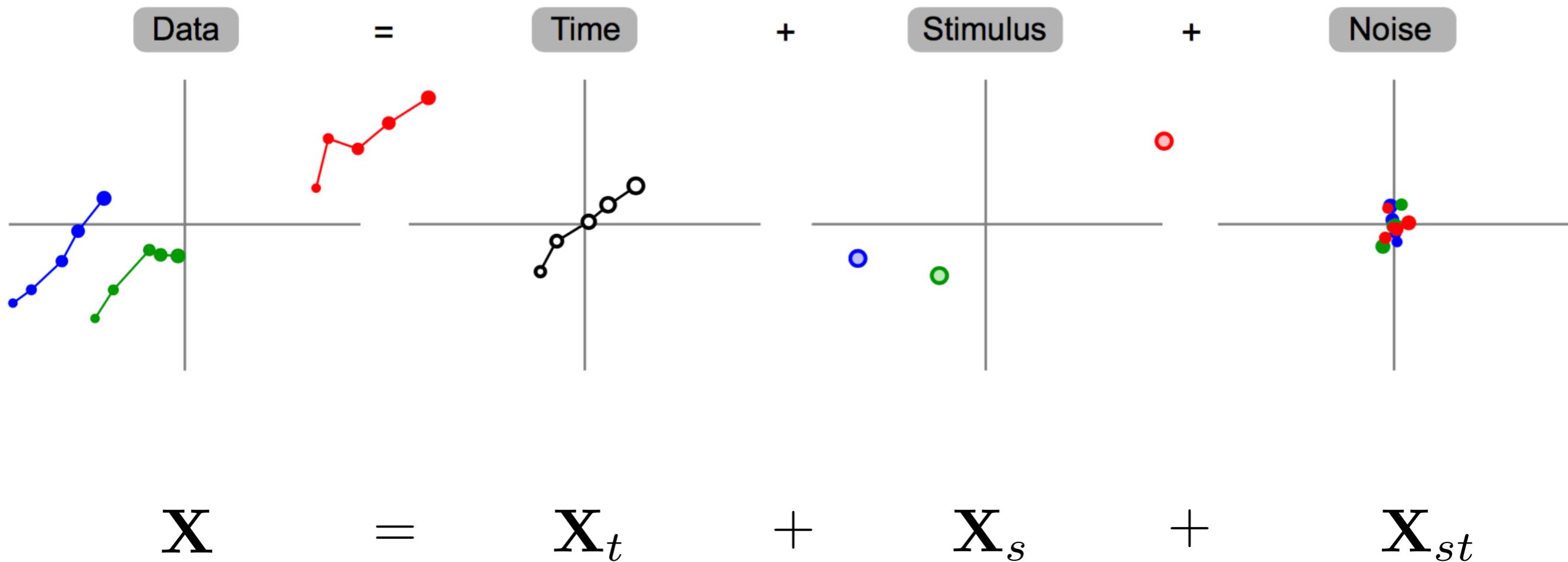
Linear readout with LDA



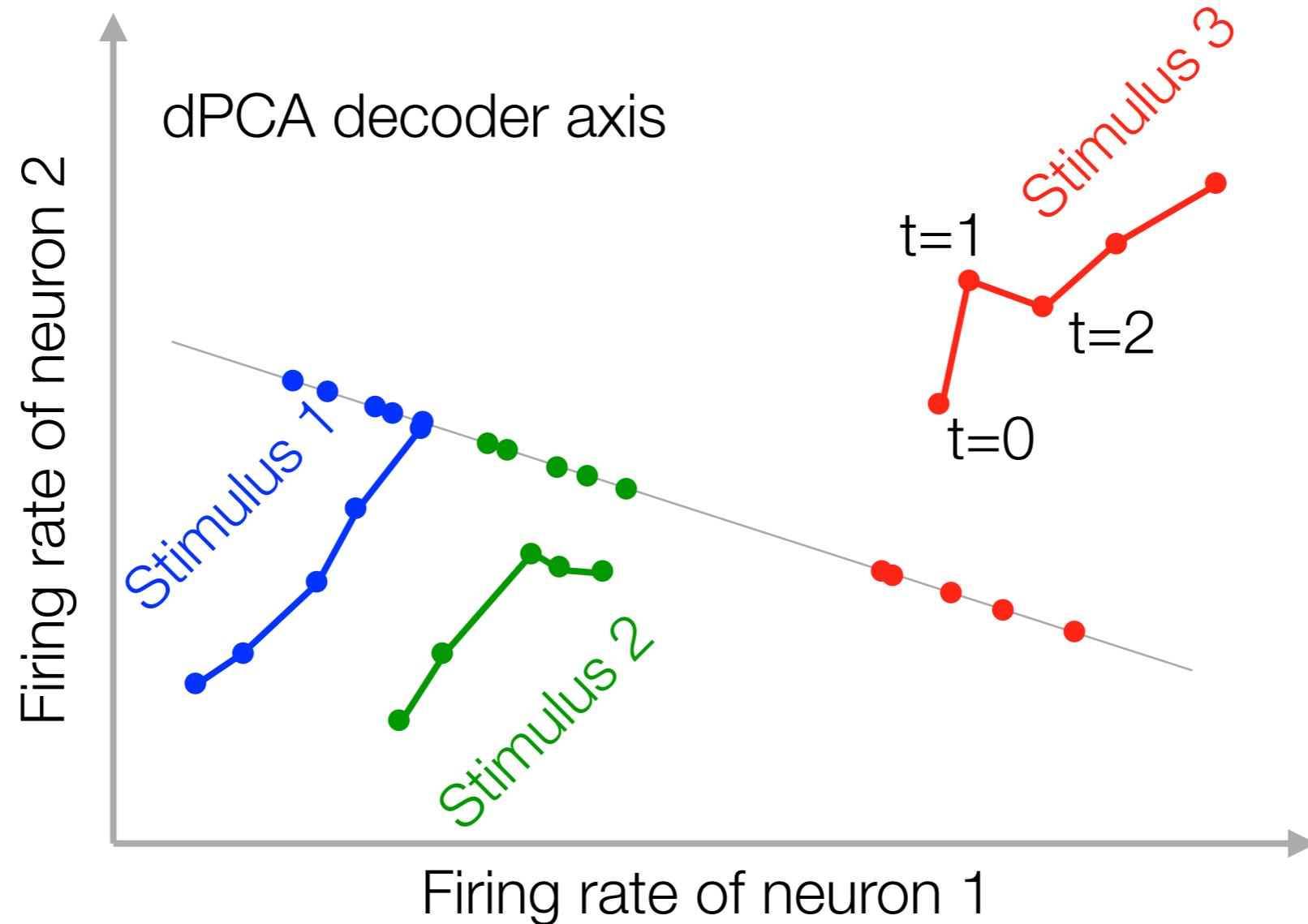
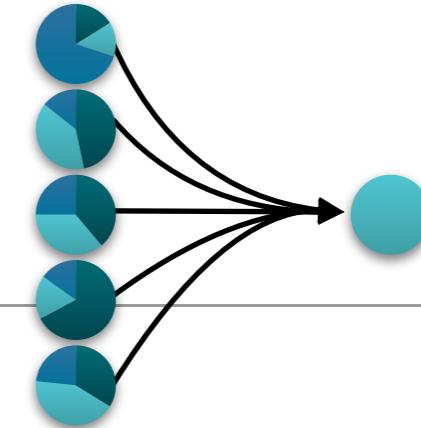
Linear readout with PCA



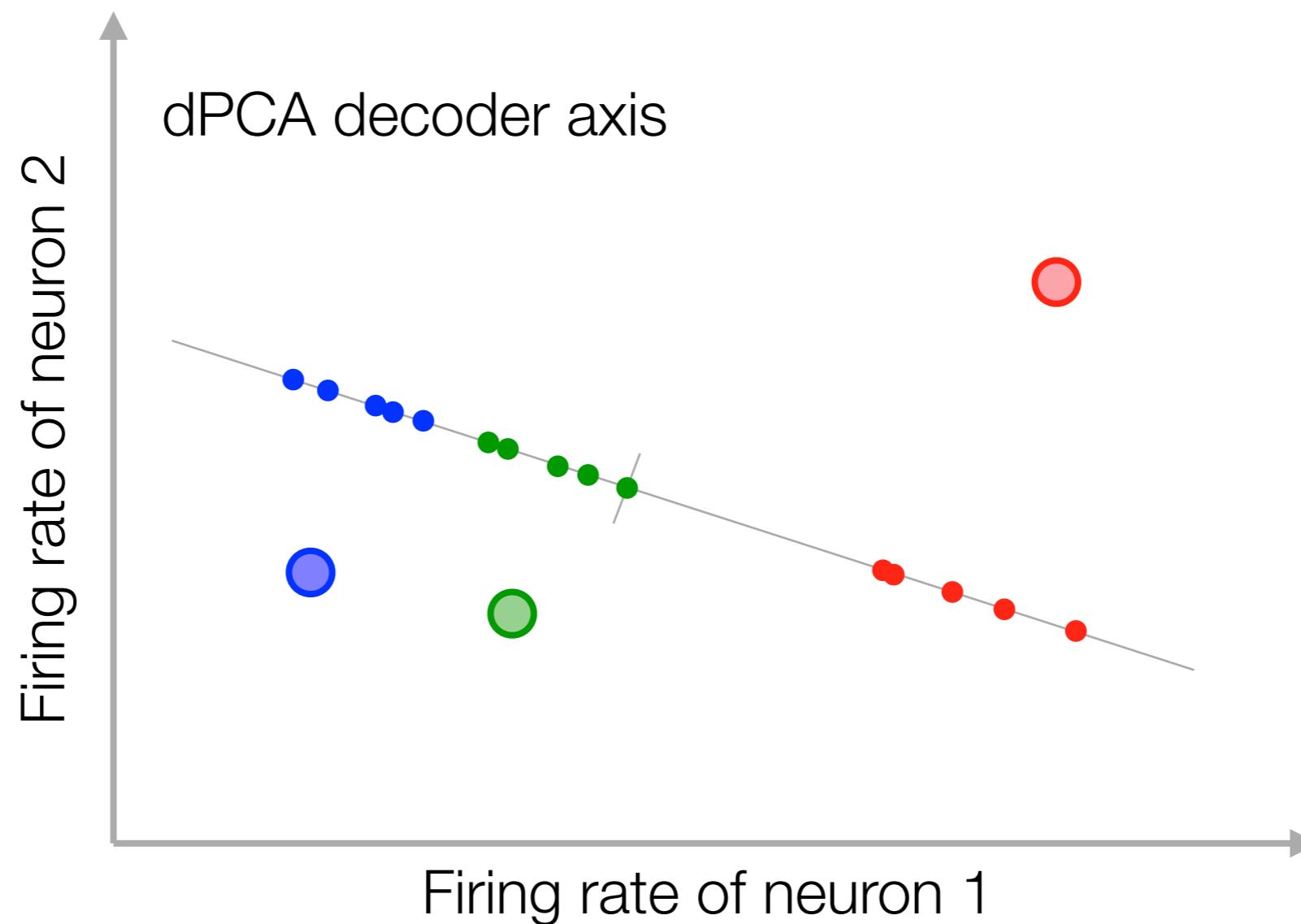
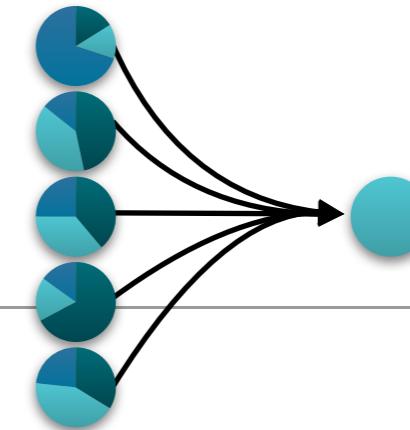
Data can be decomposed
into ‘marginalized averages’



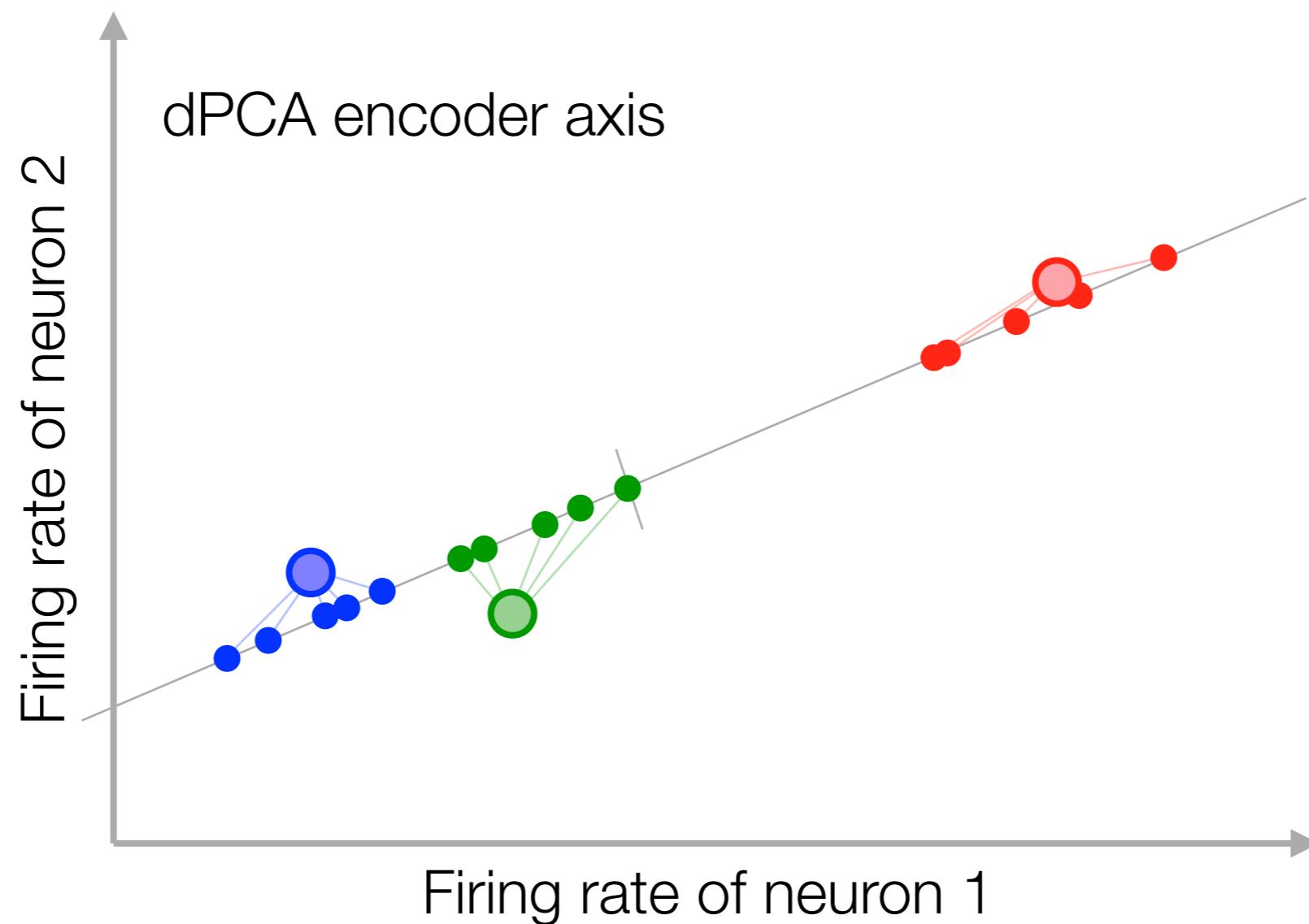
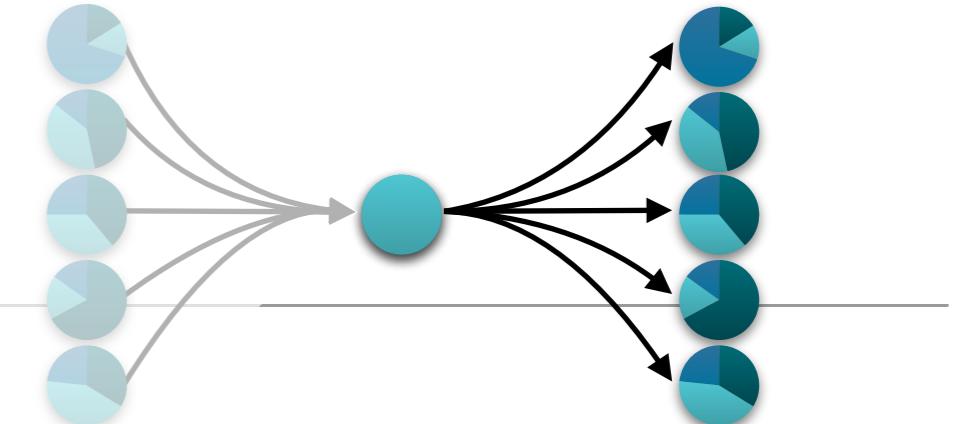
Linear readout with dPCA 'metric-preserving decoder'



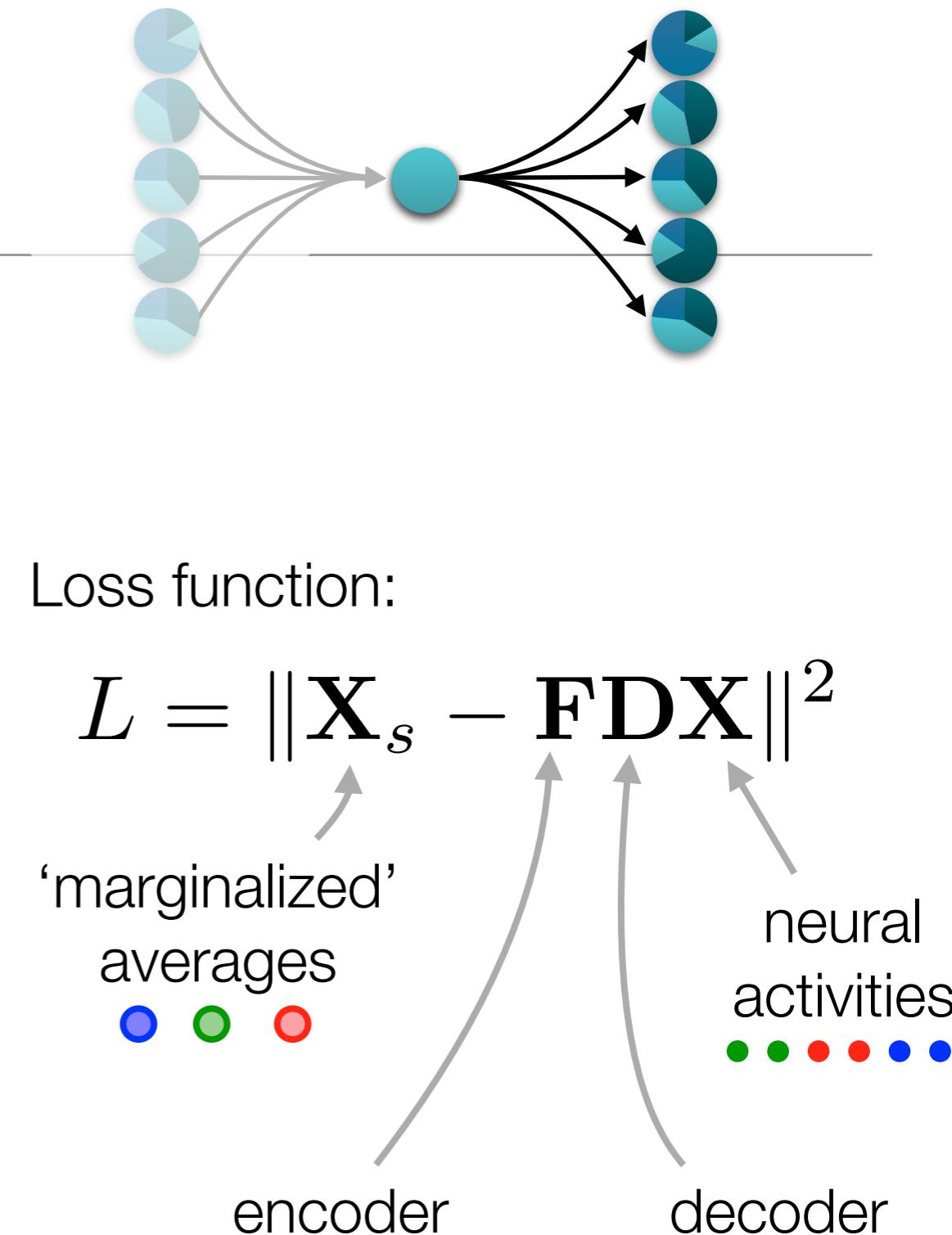
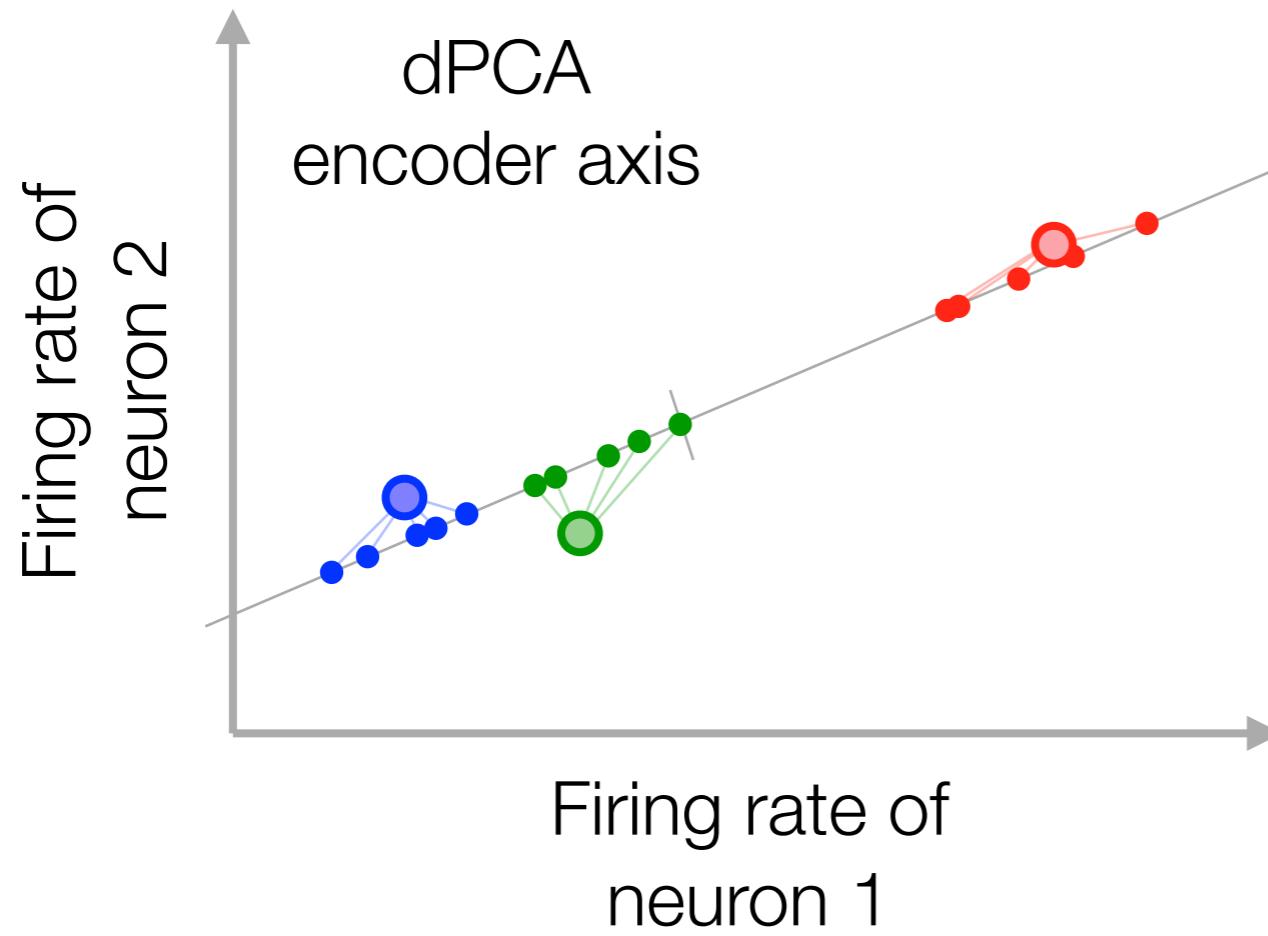
Linear readout with dPCA 'metric-preserving decoder'



Reconstruction along different ‘encoder’ axis

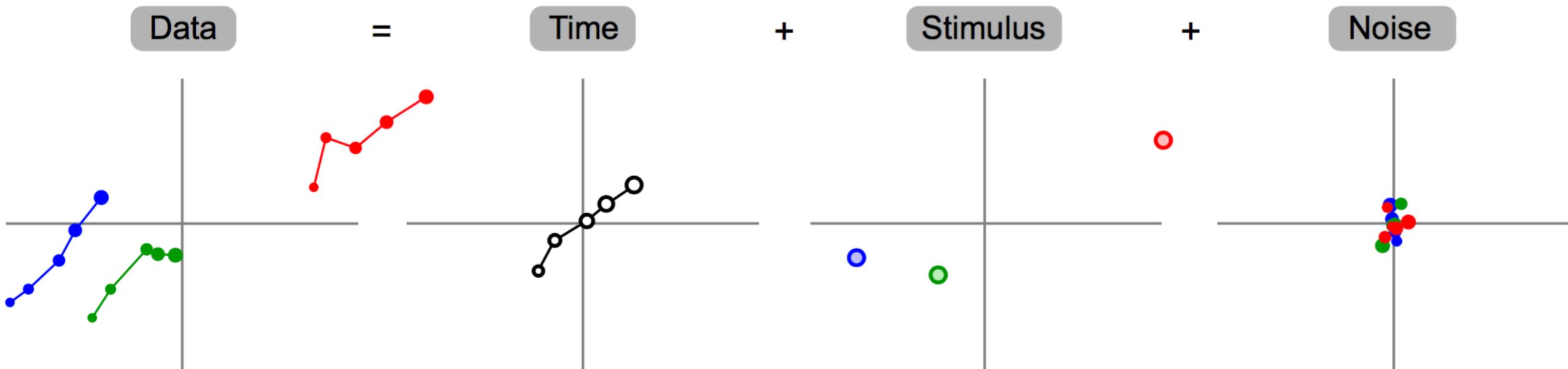


Reconstruction along different ‘encoder’ axis



“reduced rank regression”

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

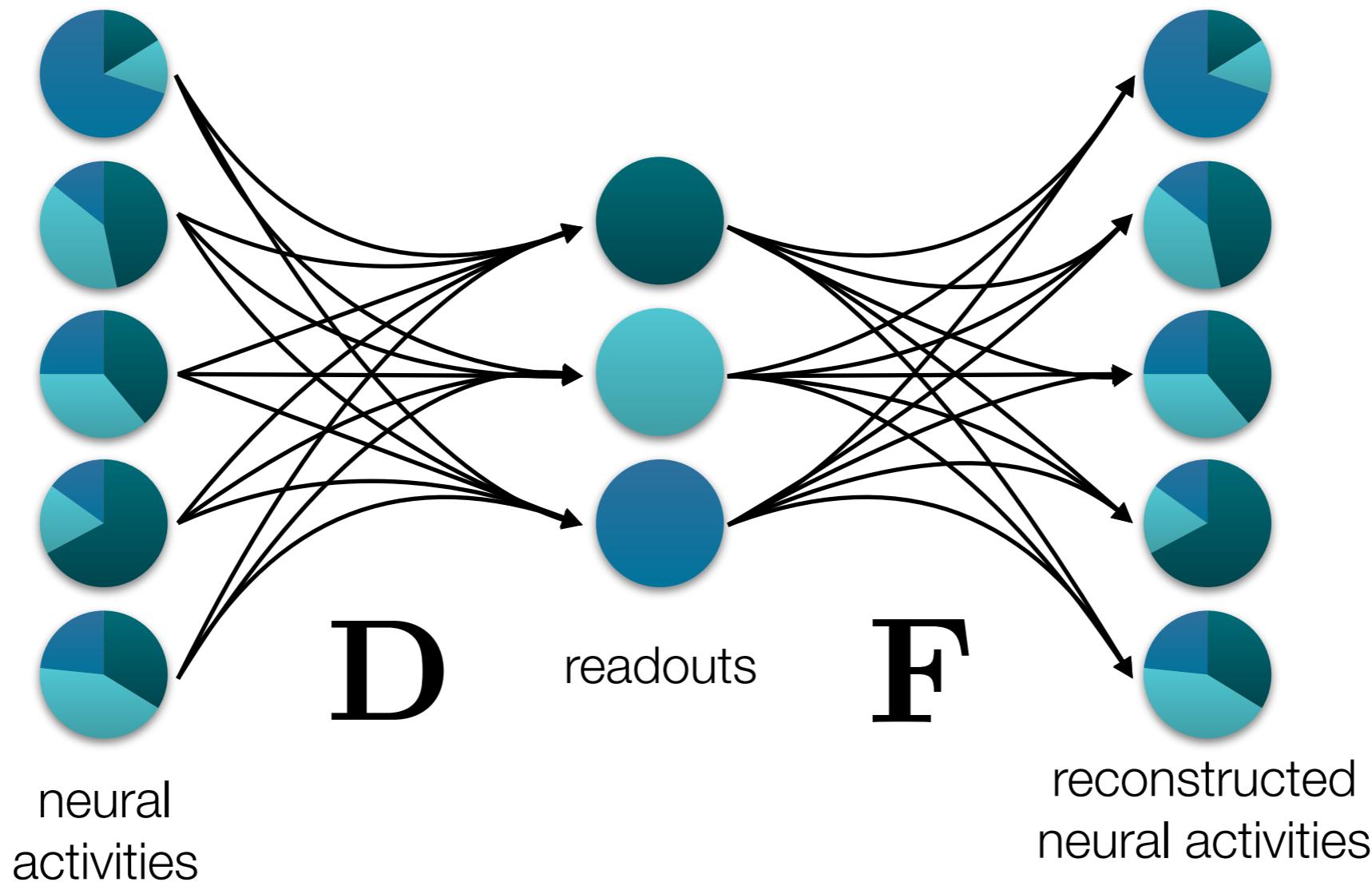


$$\mathbf{X} = \mathbf{X}_t + \mathbf{X}_s + \mathbf{X}_{st}$$

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

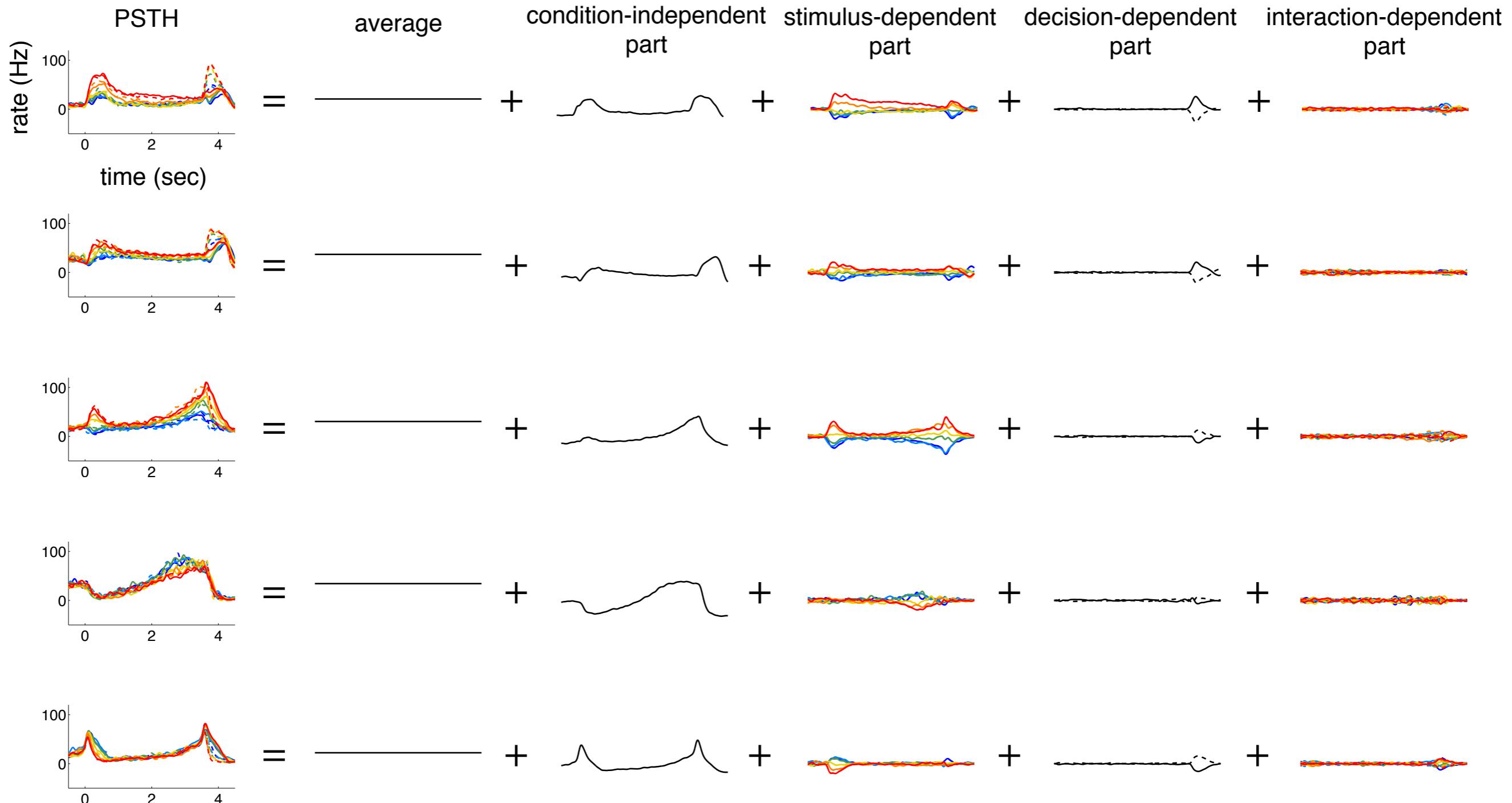
'Demixed' principal component analysis 2.0

(Kobak, Brendel et al, 2016, eLife)



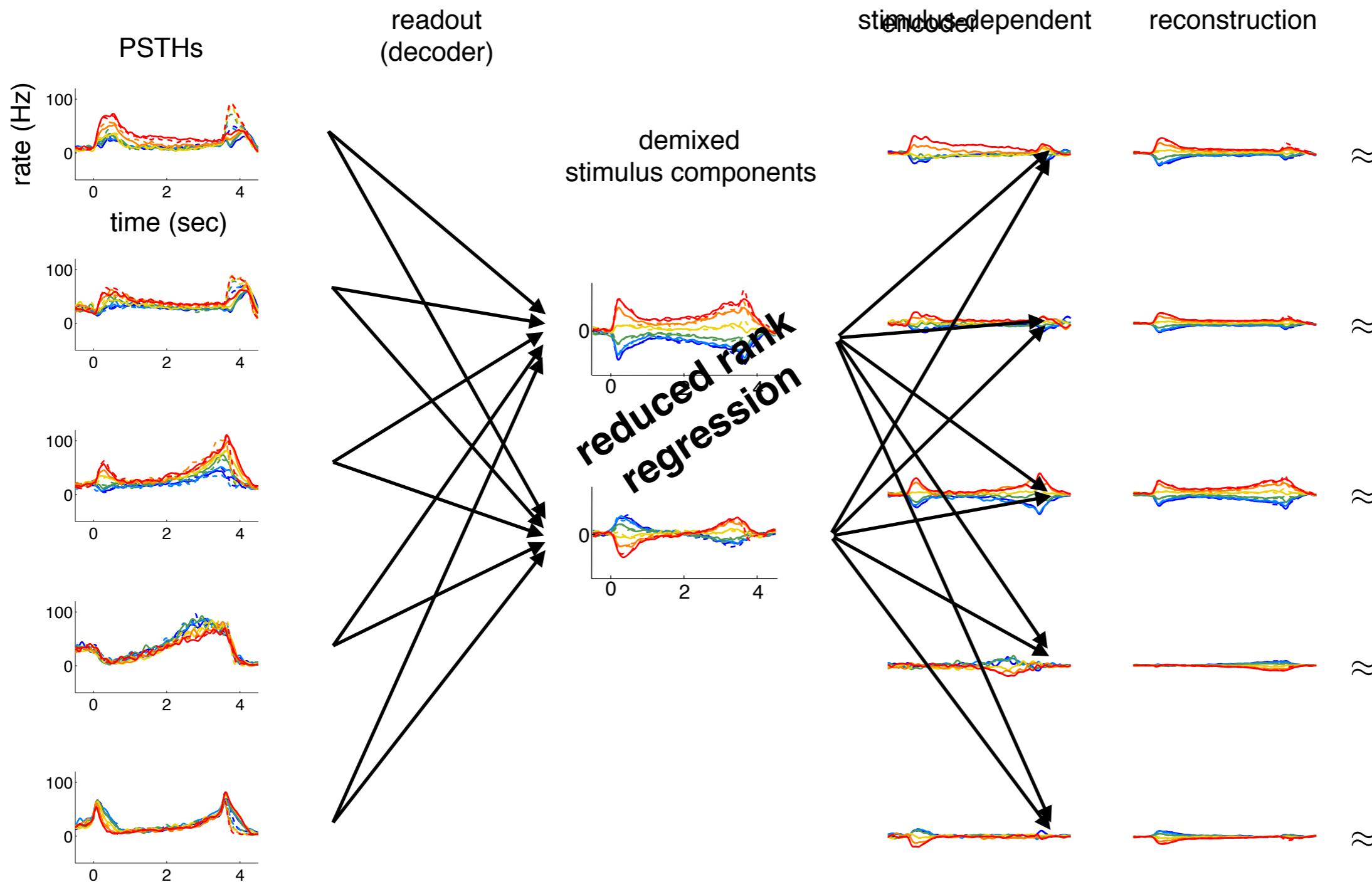
Algorithm

Step 1: decompose PSTHs into marginalized averages



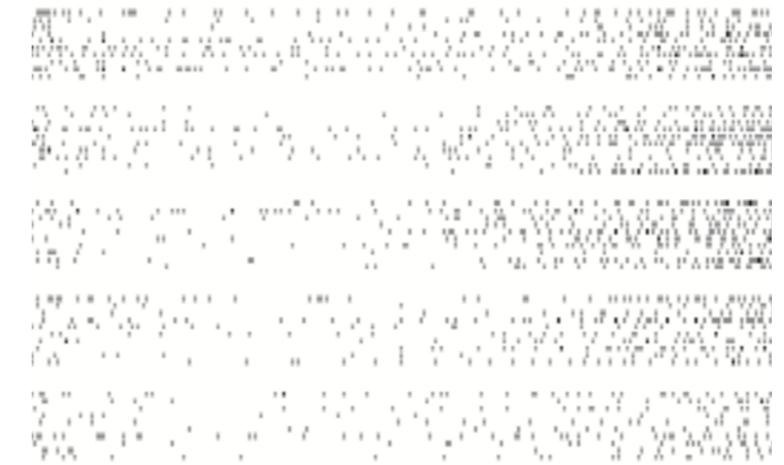
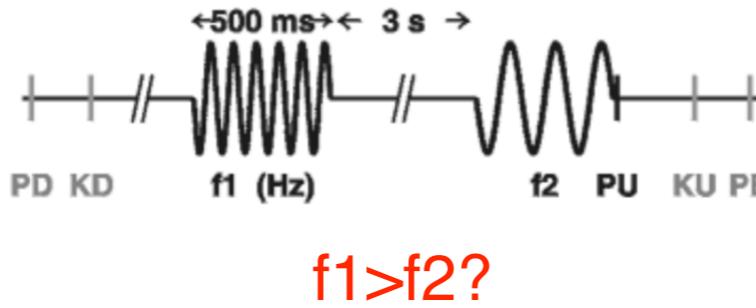
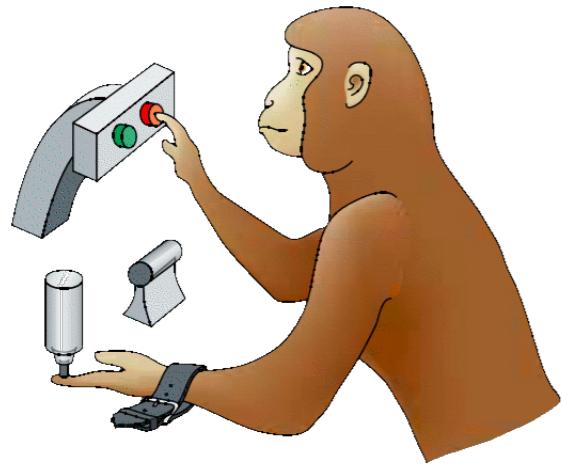
Algorithm

Step 2: reduced-rank regression



Exercise:

dPCA on Romo data



To-do-list

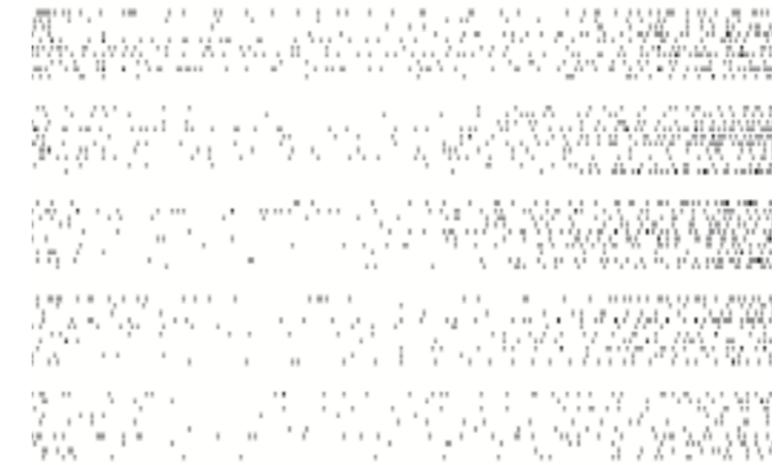
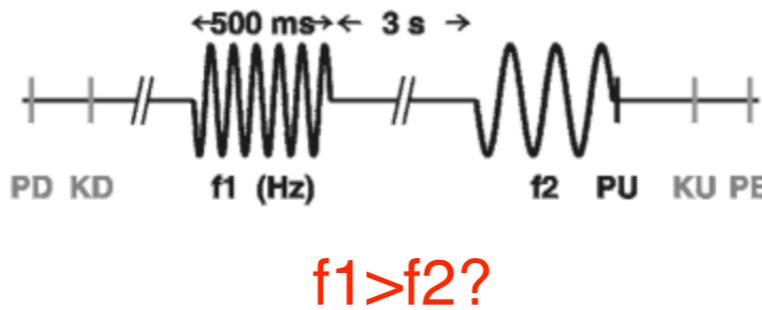
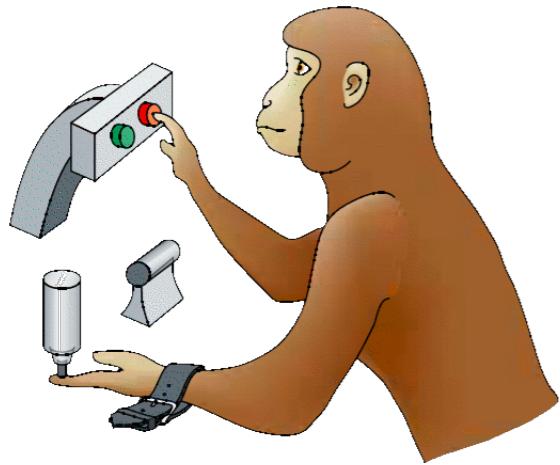
- (1) load data 'romo_allpsths.mat'
- (2) read help for dPCA.m
- (3) define parameters for use in dPCA (separate time from condition)
- (4) compute explained variance using dPCA_explainedVariance
- (5) use dPCA_plot for plotting

Advanced

- (1) Re-organize data matrix such that
$$X = [\# \text{neurons}] \times [\#\text{stimuli}] \times [\#\text{decisions}] \times [\#\text{time points}]$$
- (2) re-run dPCA

Exercise:

Solution files



Matlab scripts ...

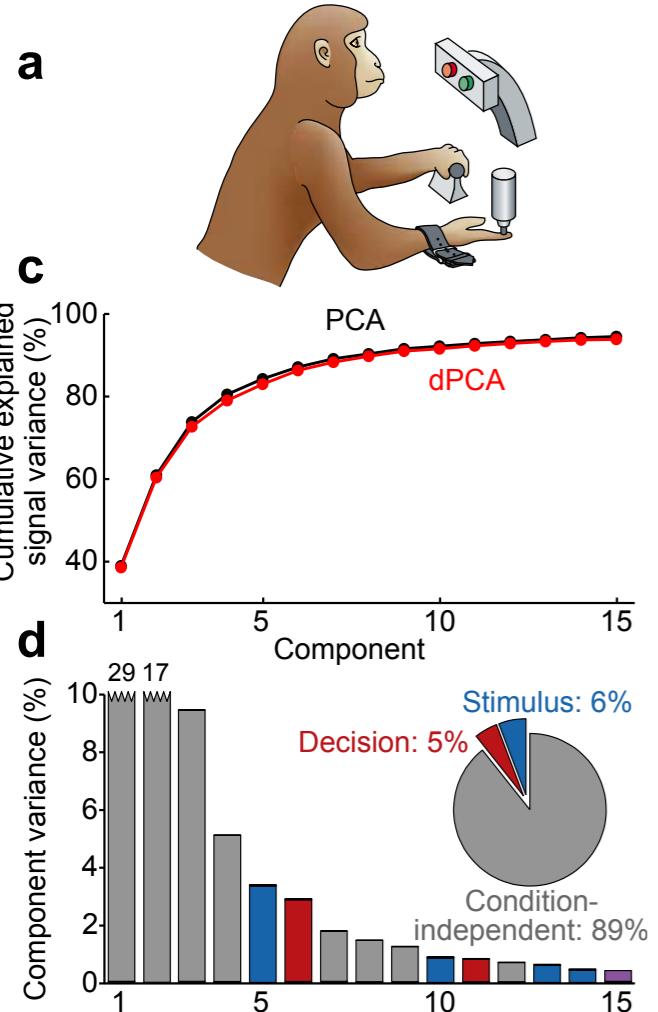
- (1) romo_allpsths
- (2) romo_dpcal
- (3) romo_dpca2

... and what they do

- extract psths of all cells in data folder rr014
- compute dPCA on data matrix X
(separating time and condition)
- compute dPCA on data matrix X
(separating time, stimulus, and decision)

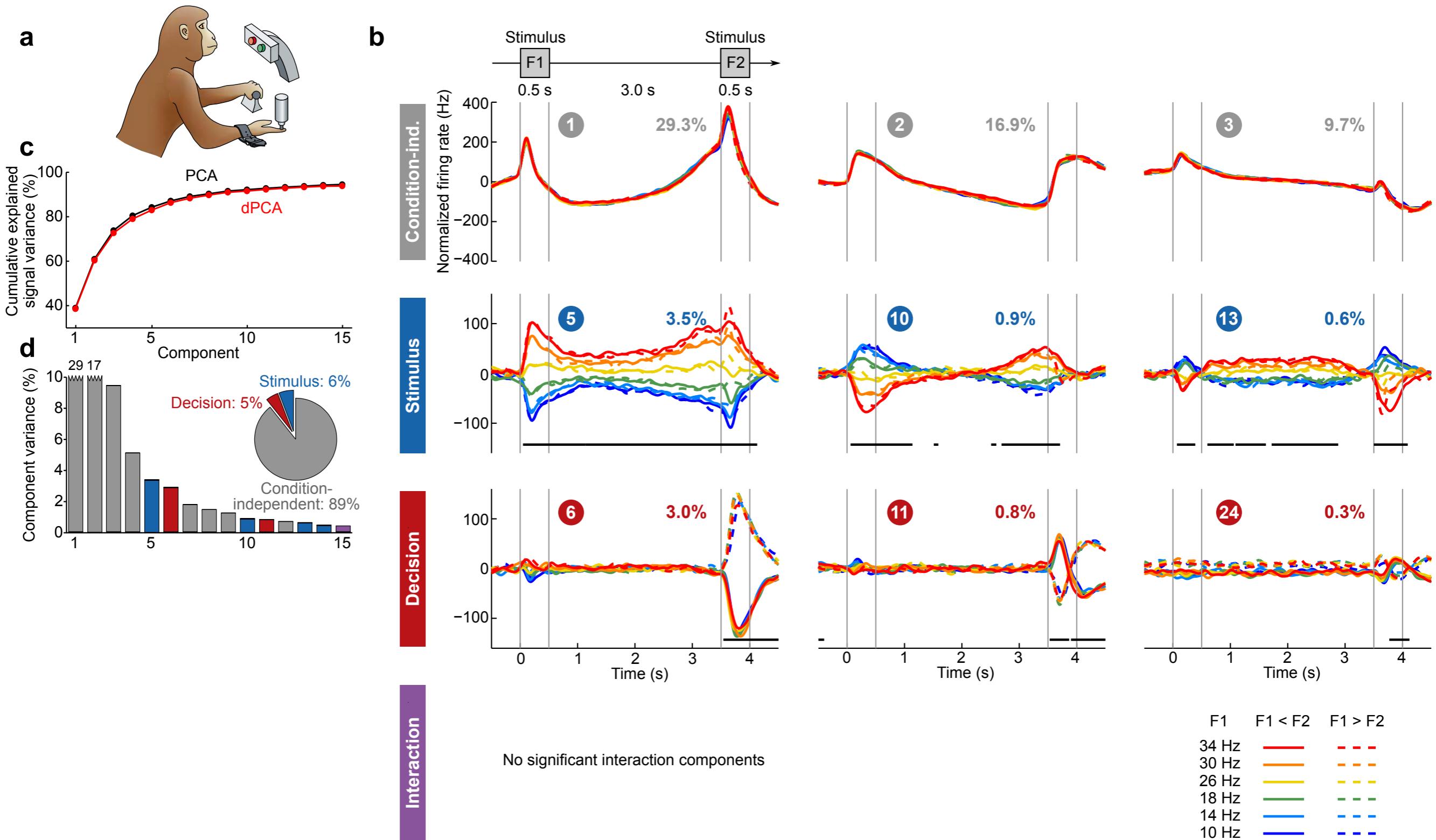
Somatosensory Working Memory Task / PFC

(Romo et al, Nature, 1999)

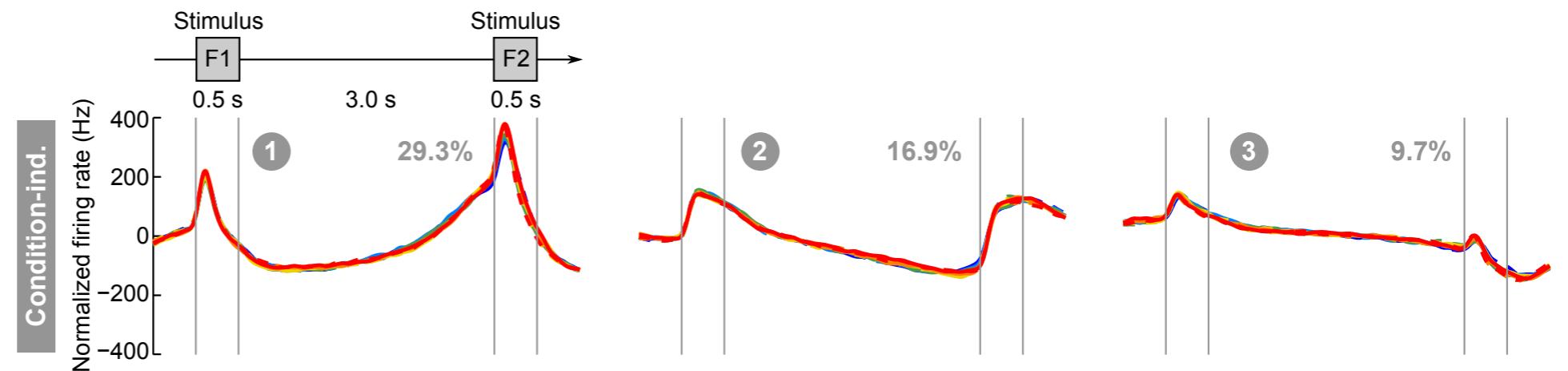


Somatosensory Working Memory Task / PFC

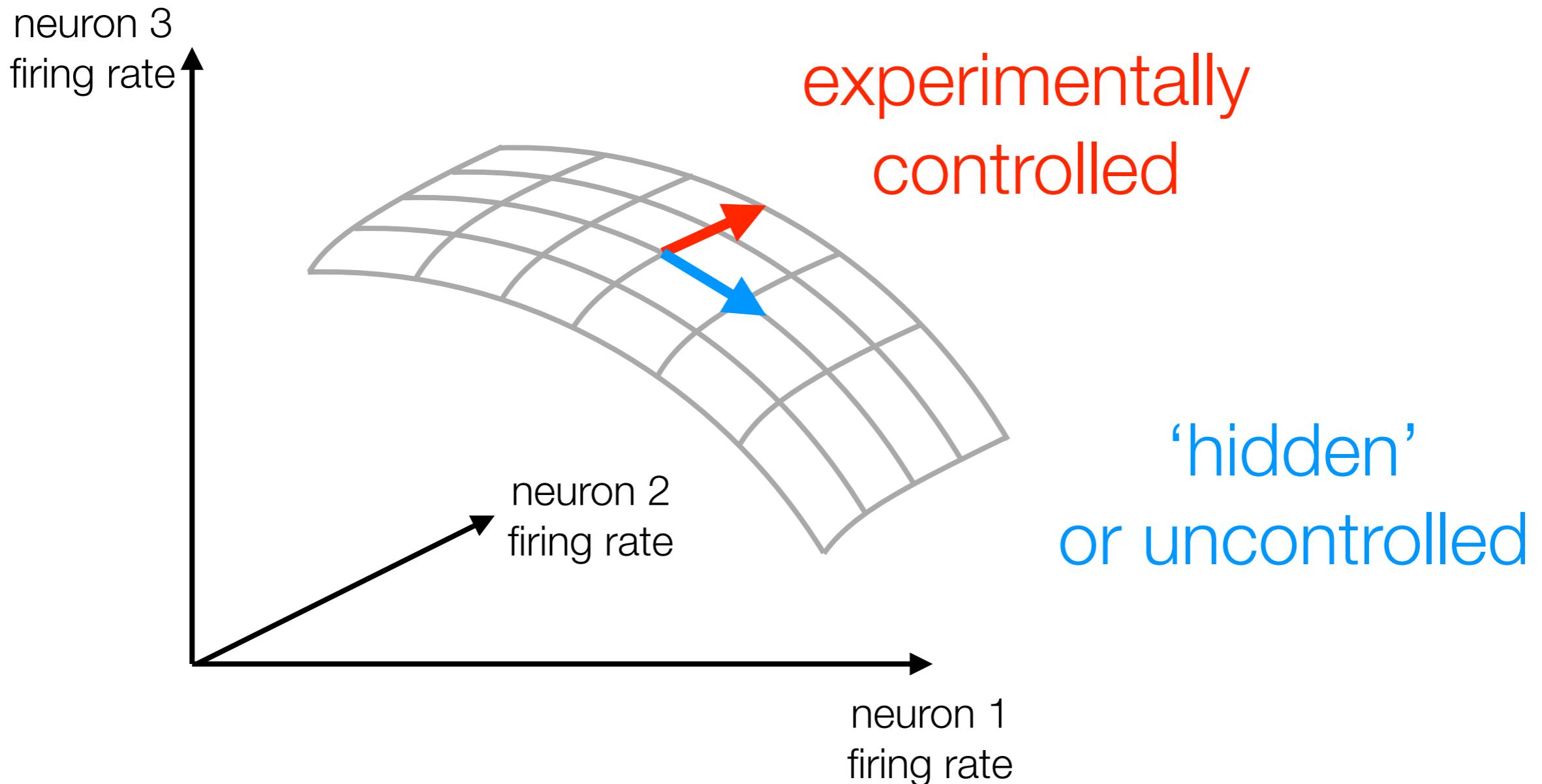
(Romo et al, Nature, 1999)



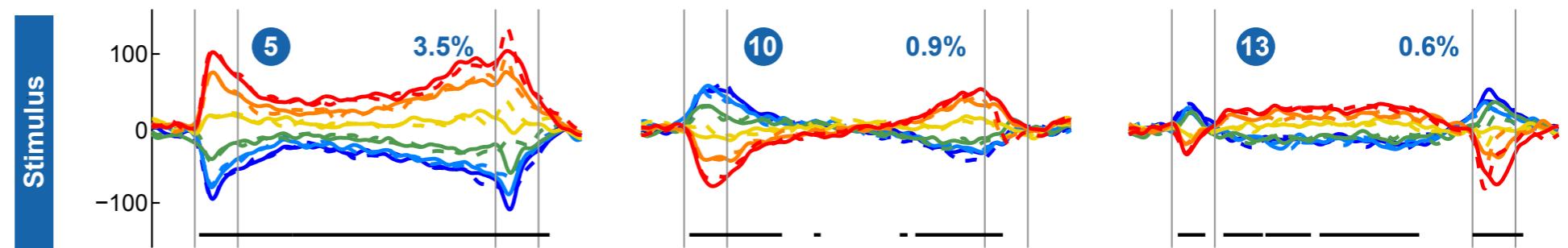
Condition-independent components:



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



Visualization of stimulus supspace



Visualization of stimulus supspace

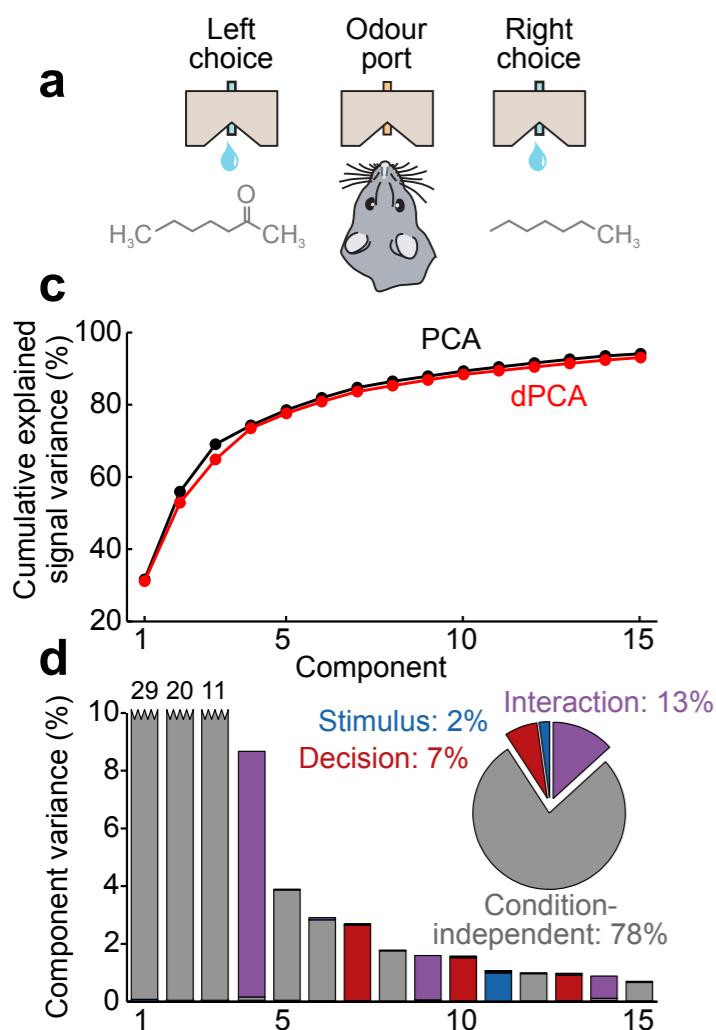


What did we learn?

Low-dimensional dynamics across many tasks

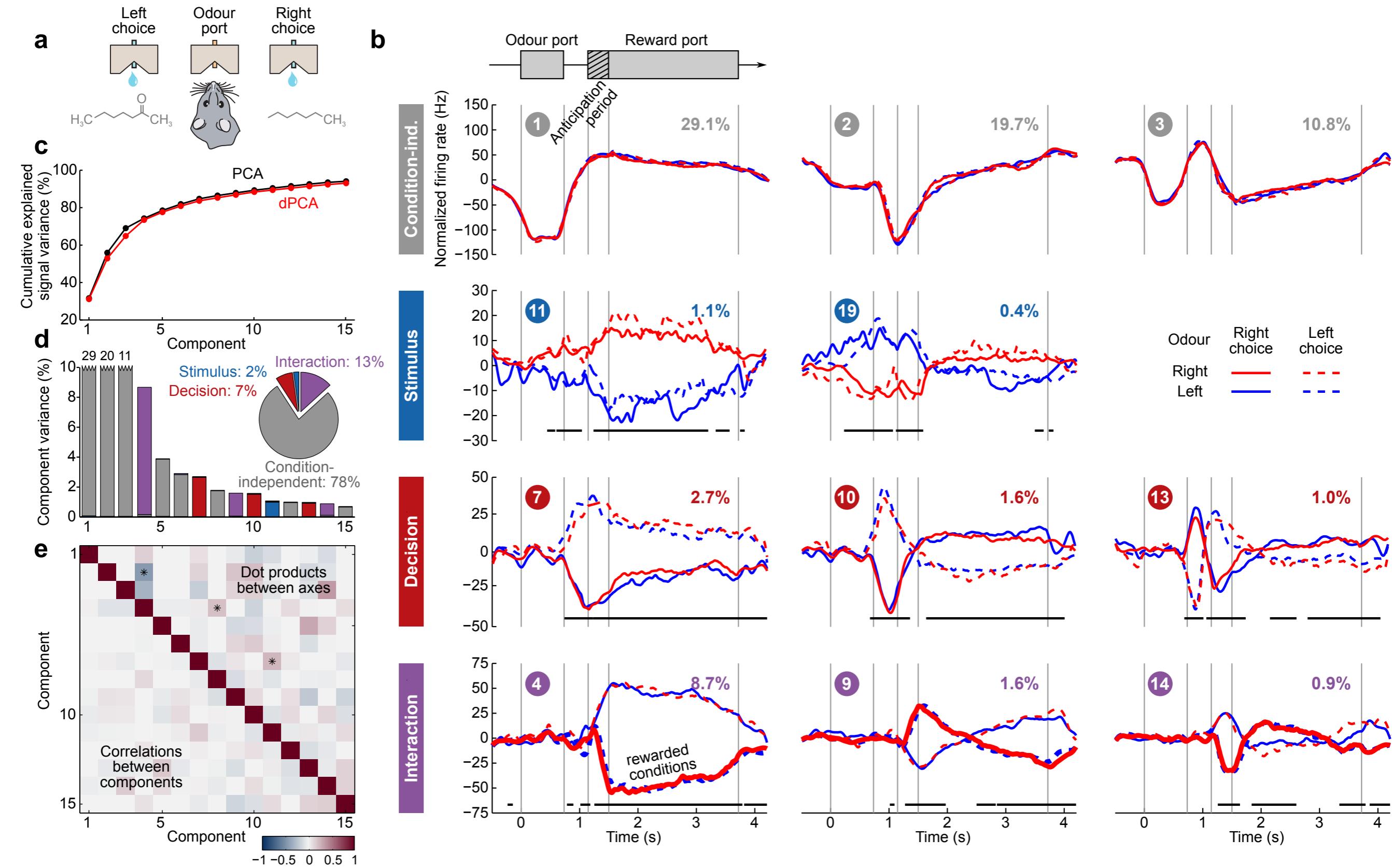
Olfactory Discrimination Task - OFC

(Feierstein et al, Neuron, 2006)



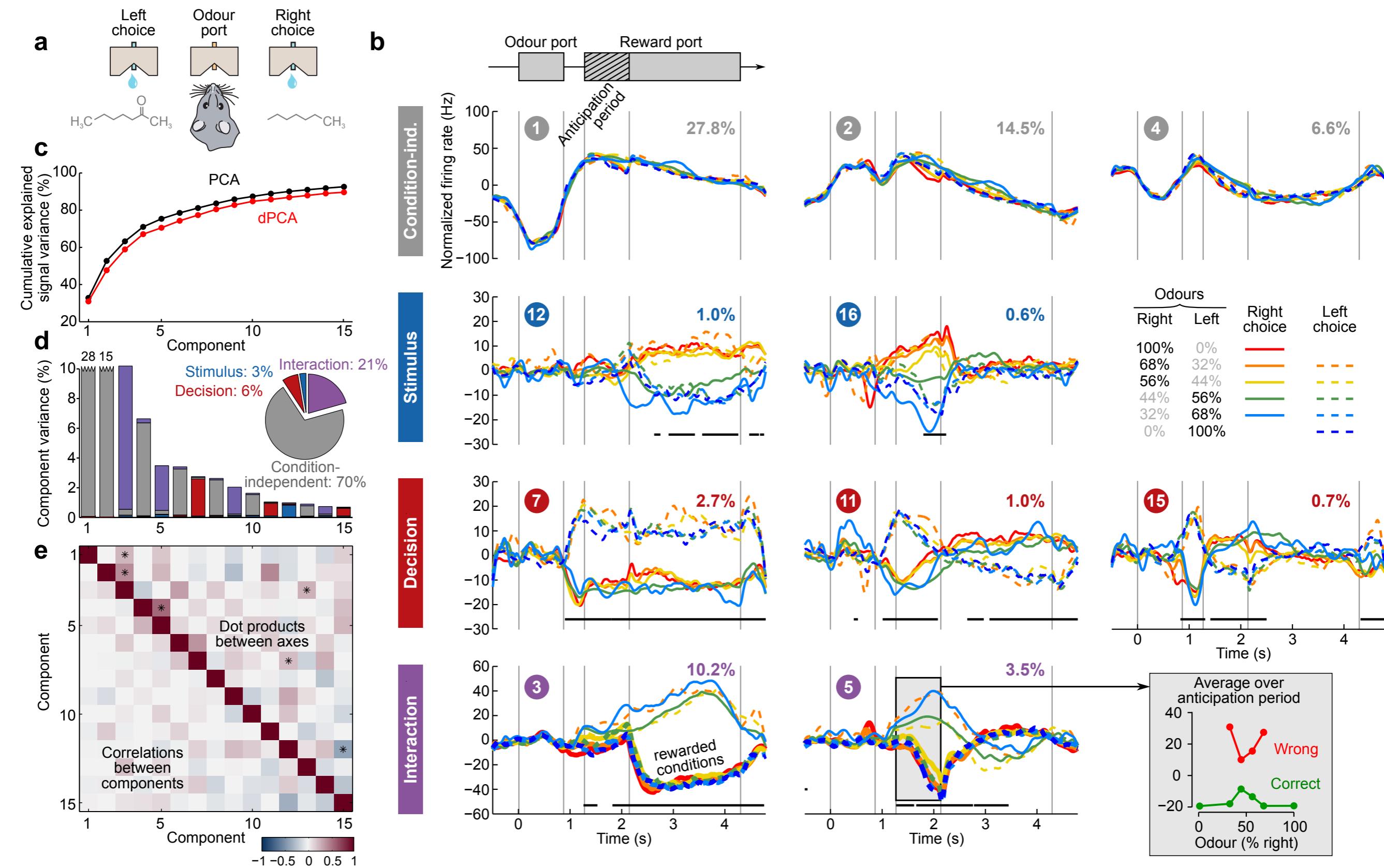
Olfactory Discrimination Task - OFC

(Feierstein et al, Neuron, 2006)

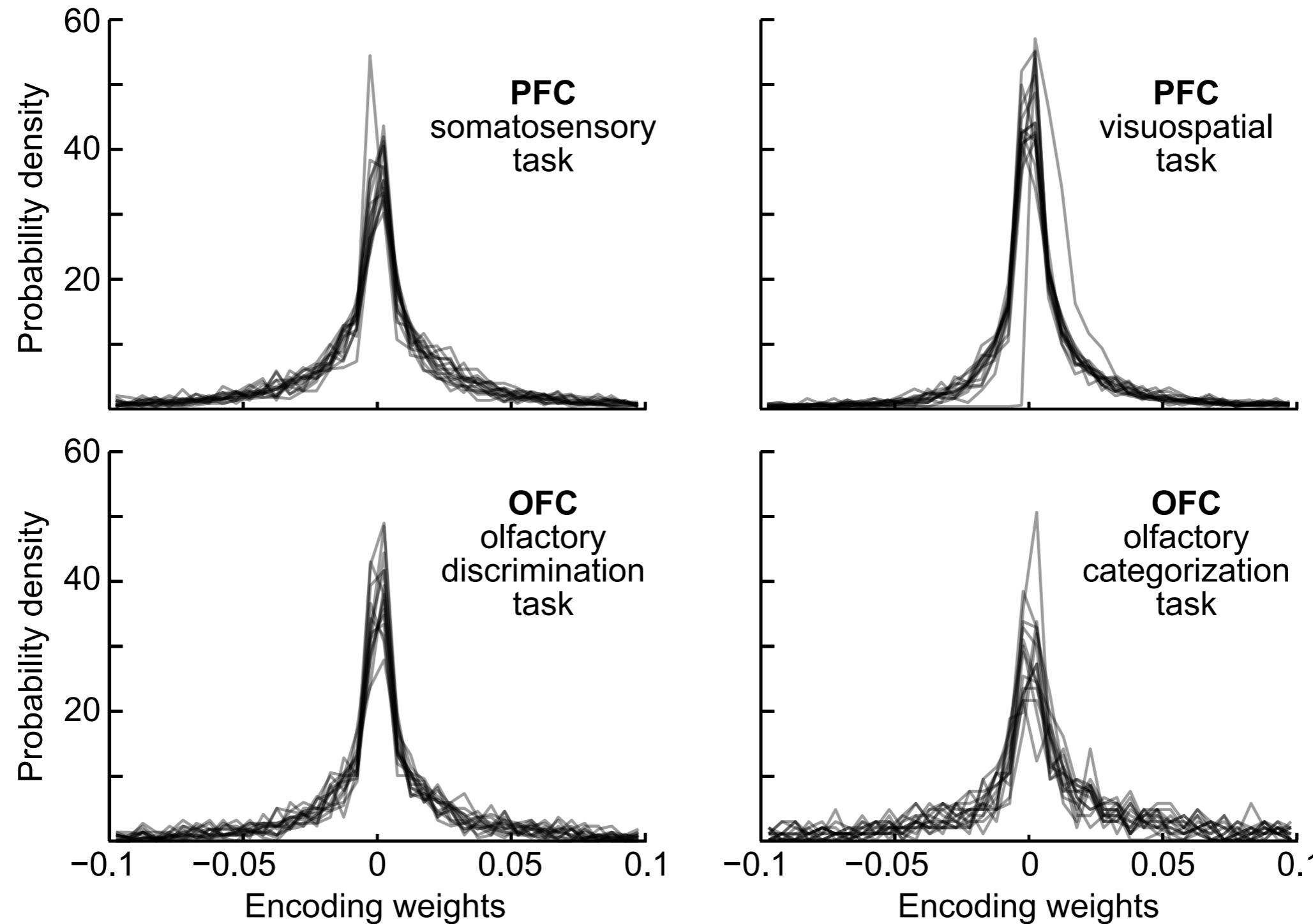


Olfactory Categorization Task - OFC

(Kepecs et al, Nature, 2008)



Distribution of encoding weights roughly Laplacian



Low-dimensional, orthogonal representation all over...

Primate:

- PFC: Mante et al (2013) : color \perp motion \perp decision along separate axes in state space
- M1: Ames & Churchland (2019): right arm movement \perp left arm movement
- M1: Gallego et al (2019): movement types create cartesian coordinate system
- IT: Chang & Tsaio (2017): cartesian coordinates for a ‘face space’
- ...

Rodents:

- A1: Kobak et al (2019): absolute sound level \perp inter-aural level difference
- Visual cortex: Stringer et al (2019): facial movement \perp visual input
- ...

The population view: Common observations

- **Low-dimensionality:** linear projections of neural activity eliminate the diversity/complexity of single-cell tuning in favour of a simpler ‘population picture’

The population view: Common observations

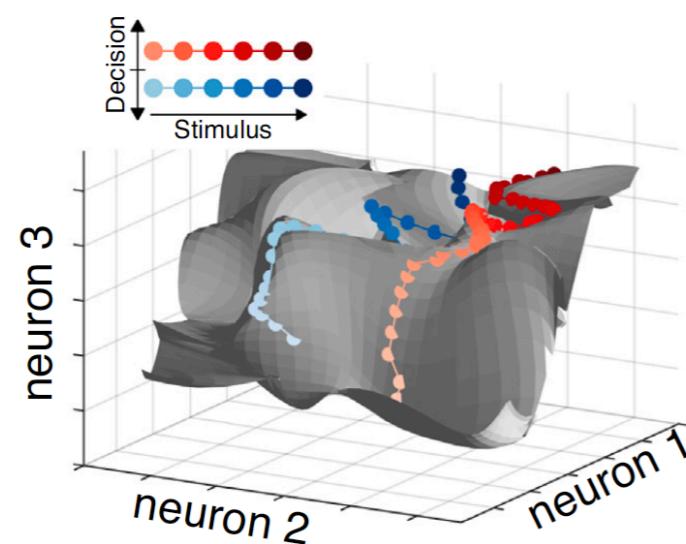
- **Low-dimensionality:** linear projections of neural activity eliminate the diversity/complexity of single-cell tuning in favour of a simpler ‘population picture’
- **Demixability:** Different task parameters often represented along ‘orthogonal axes’ in state space

The population view: Common observations

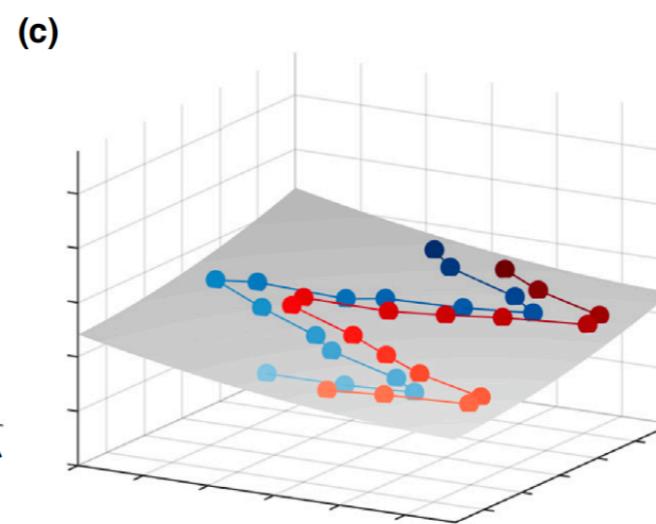
- **Low-dimensionality:** linear projections of neural activity eliminate the diversity/complexity of single-cell tuning in favour of a simpler ‘population picture’
- **Demixability:** Different task parameters often represented along ‘orthogonal axes’ in state space
- **Distributed Codes:** Each projection is made up by combining many individual neurons; weights are often distributed according to a Laplacian

The population view: Common observations: trivial or not?

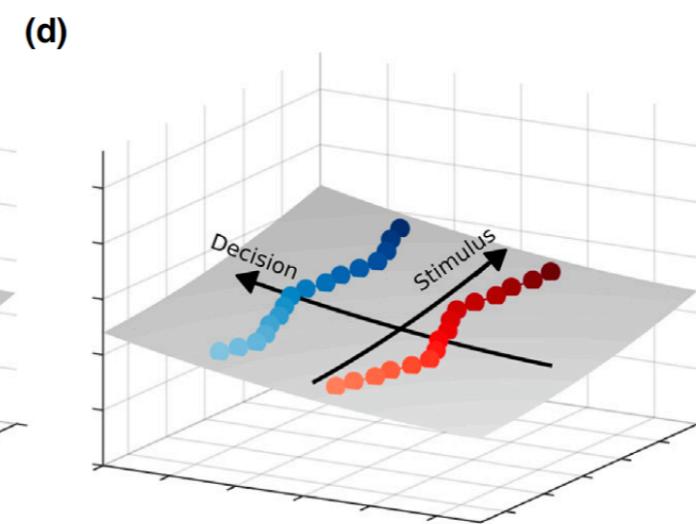
- **Low-dimensionality:** linear projections of neural activity eliminate the diversity/complexity of single-cell tuning in favour of a simpler ‘population picture’
- **Demixability:** Different task parameters often represented along ‘orthogonal axes’ in state space
- **Distributed Codes:** Each projection is made up by combining many individual neurons; weights are often distributed according to a Laplacian



demixable
not low-dimensional



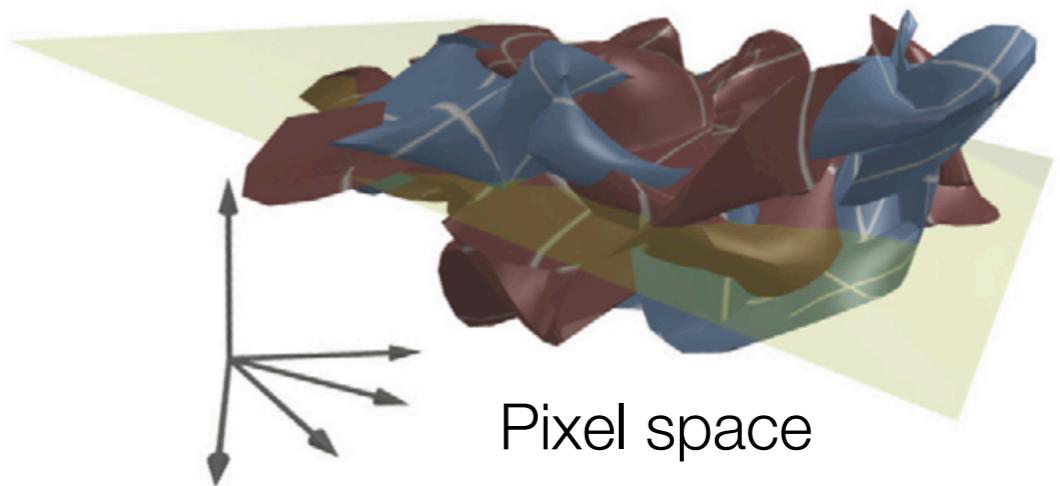
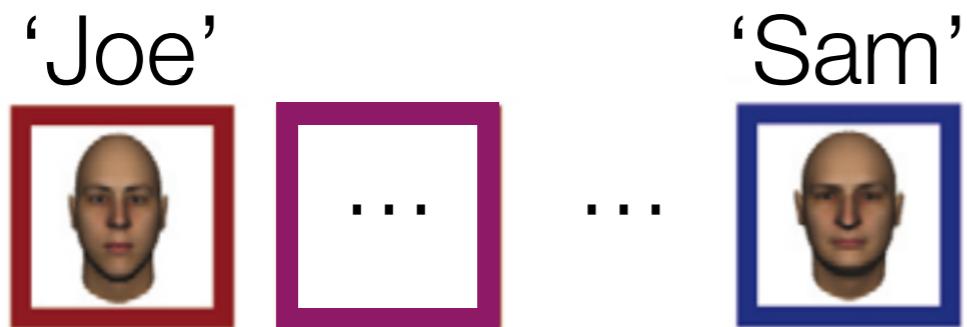
not demixable
low-dimensional



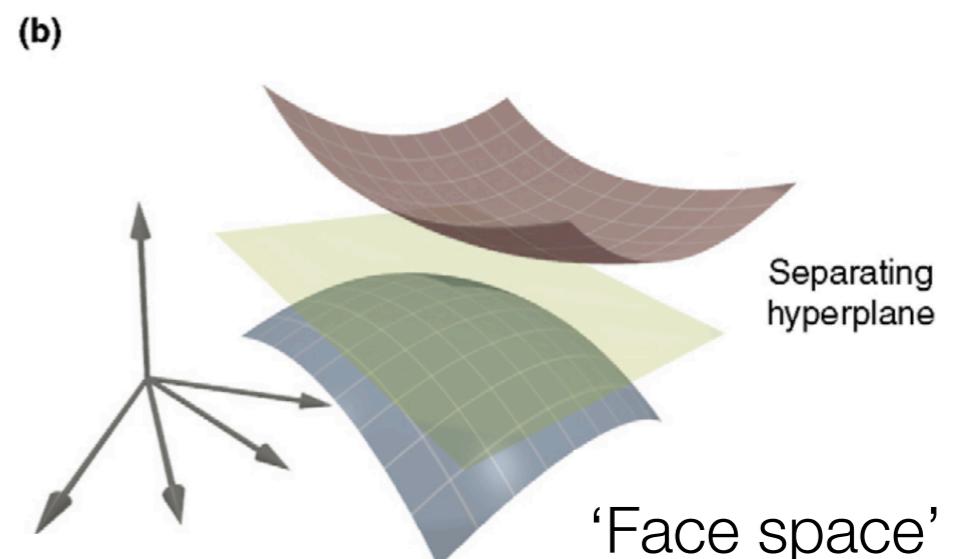
demixable
low-dimensional

‘Task parameters’ are defined by the experimenter

Categorization task



What are the ‘task parameters’?



‘Task parameters’ are defined by the experimenter

Categorization task

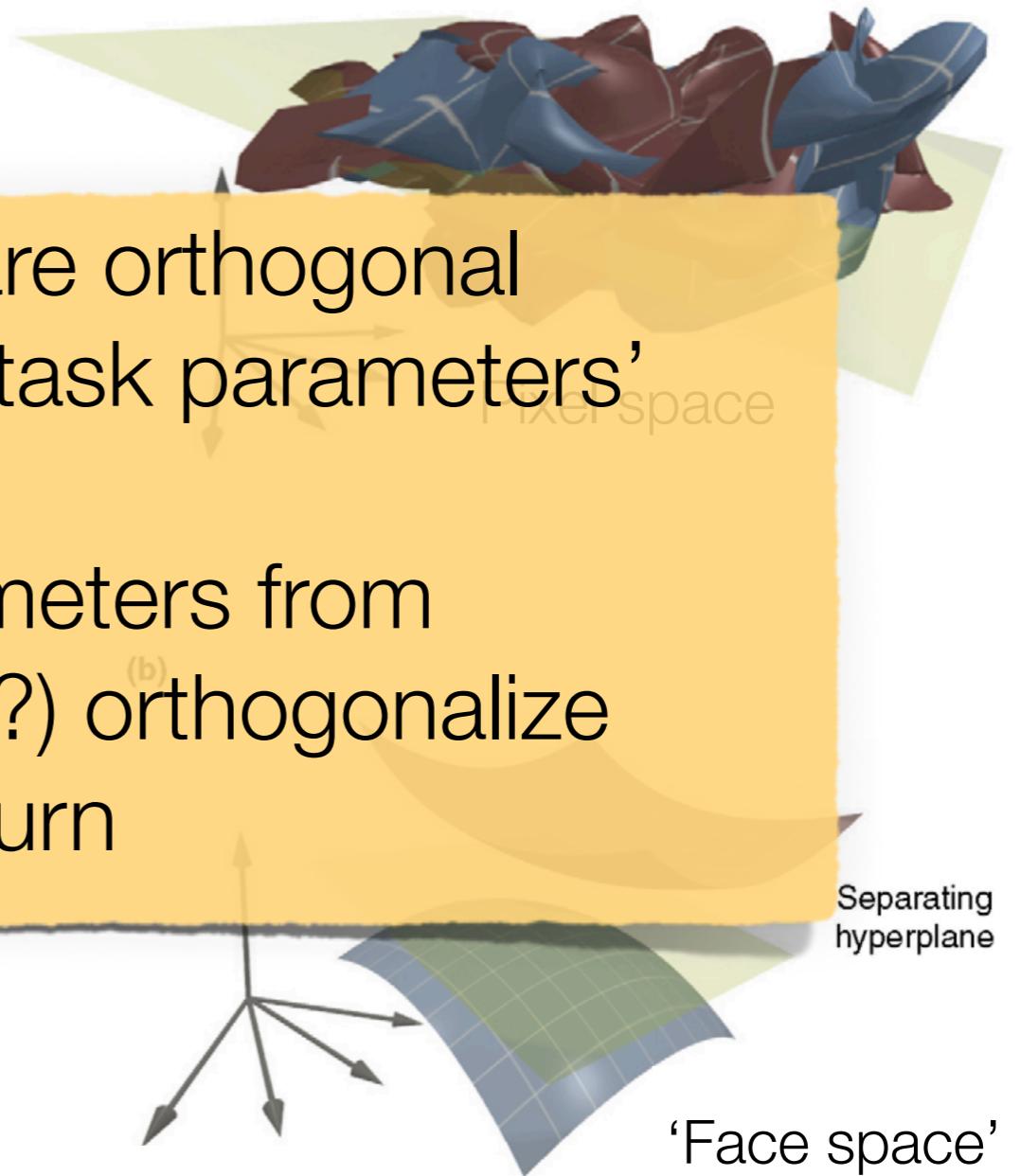
‘Joe’



‘Sam’

Neural representations are orthogonal
with the ‘right choice of task parameters’

What are the task parameters?
Animals learn task parameters from
experience and maybe (?) orthogonalize
their representations in turn



‘Face space’

The population view: What did we lose track of? Where should we go?

The population view:

What did we lose track of? Where should we go?

- Higher-order components
- Outlier ‘single’ cells
- Anything not visible in firing rates (e.g. phase coding)
- ...?
- Non-linear methods?

Conclusion

- Classically, people ‘looked’ at single cells and sought to categorise them
- Single cell responses are often quite complex and heterogeneous (‘mixed selectivity’)
- Nowadays, people ‘look’ at linear projections of neural activities because these eliminate the complexity and heterogeneity of single-cell responses
- Linear projections highlight similarities of information representation across tasks, cortical areas, and species.
- We’ll know the ‘right way’ of looking at the data once we understand how the brain works .. ;-)