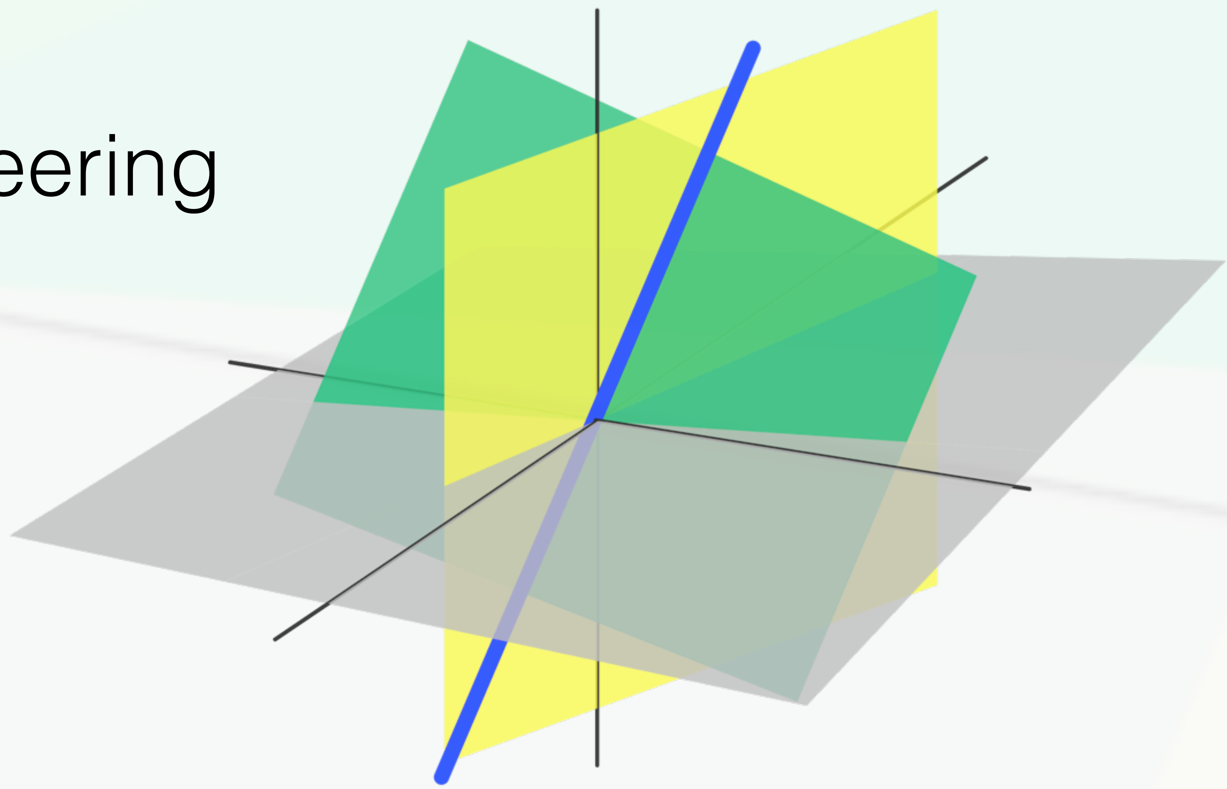
Finding **low-dimensional** structure in **large-scale** neural datasets

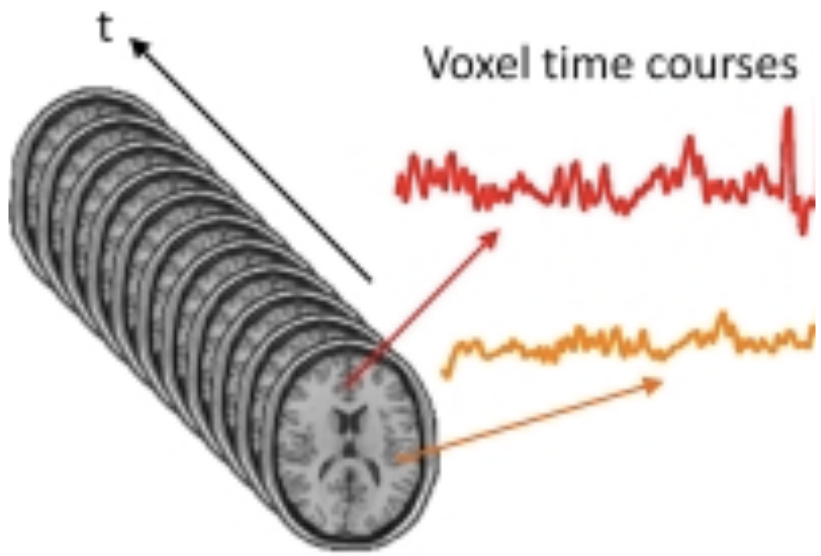
Eva Dyer

Department of Biomedical Engineering
Georgia Institute of Technology //
Emory University

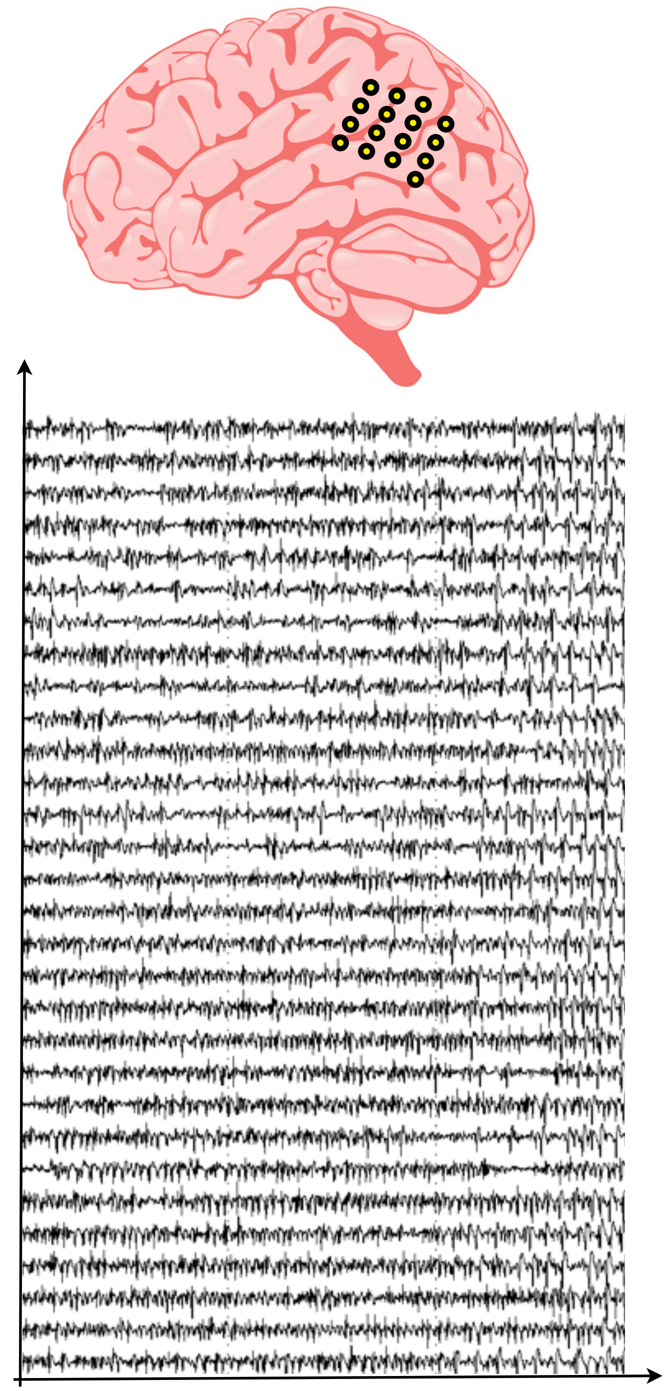


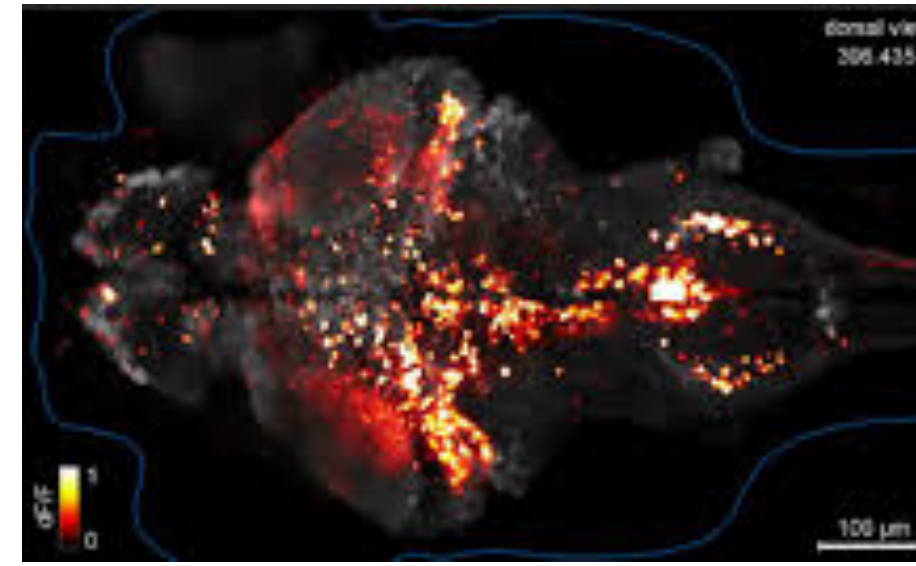
NeuroHackademy - Aug 3 2018

neural data deluge

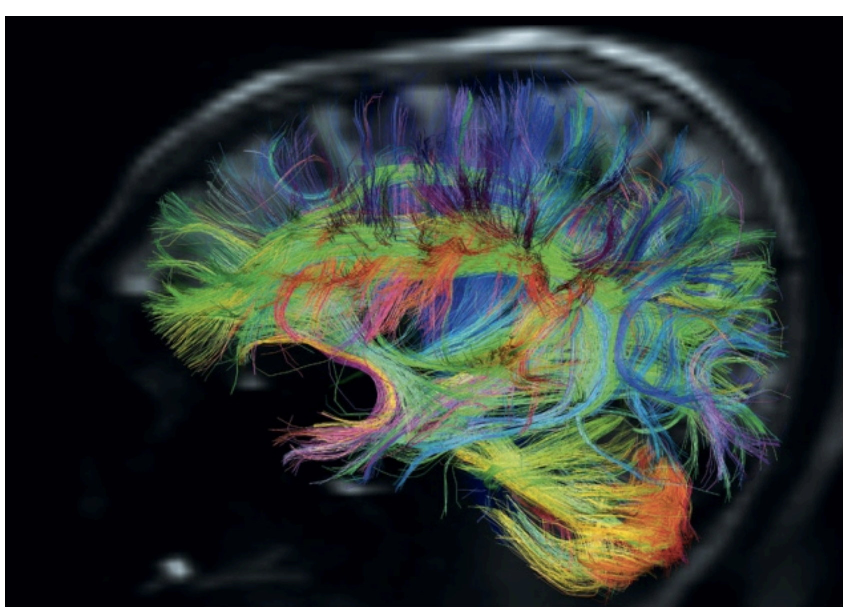


fMRI



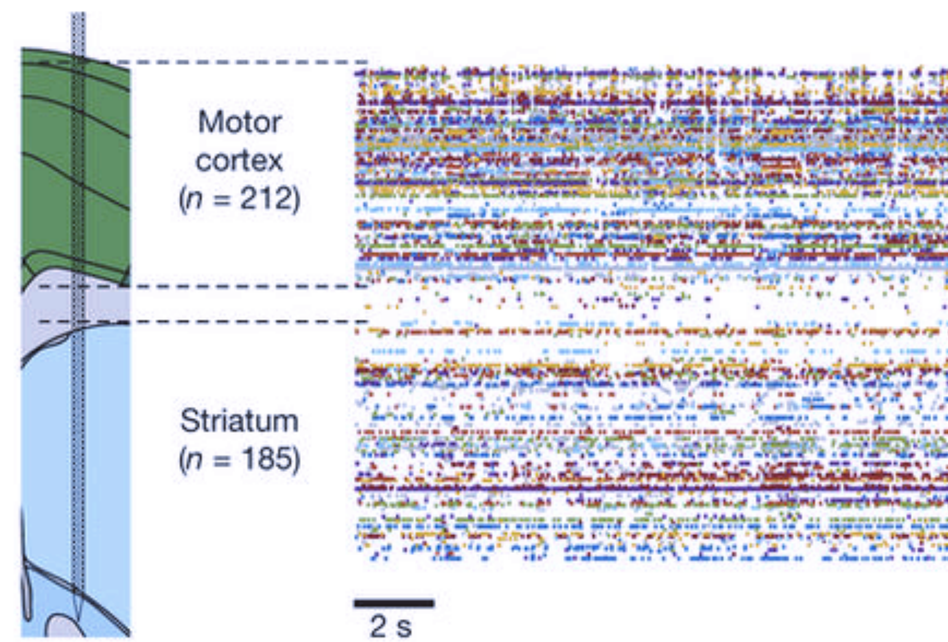


calcium imaging



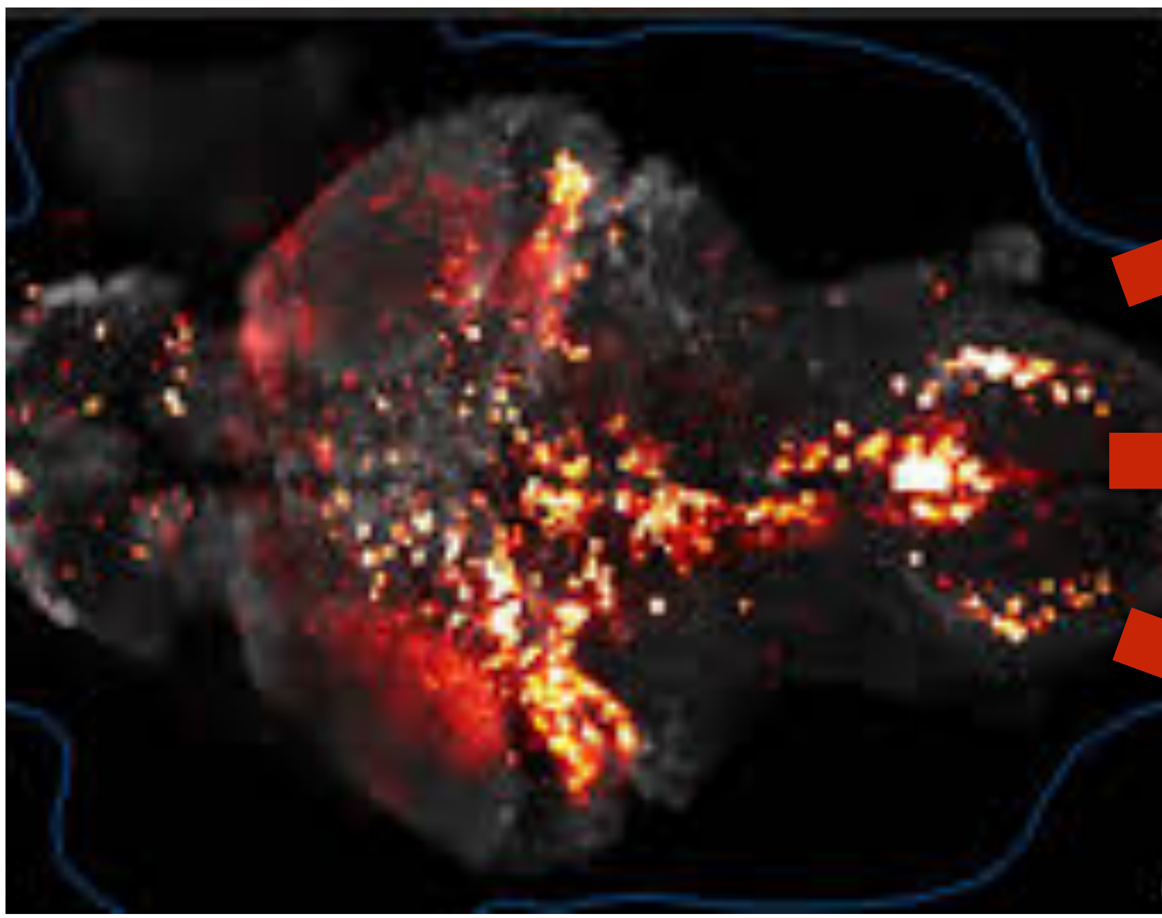
DTI

ecog



extracellular arrays

why reduce dimensionality?

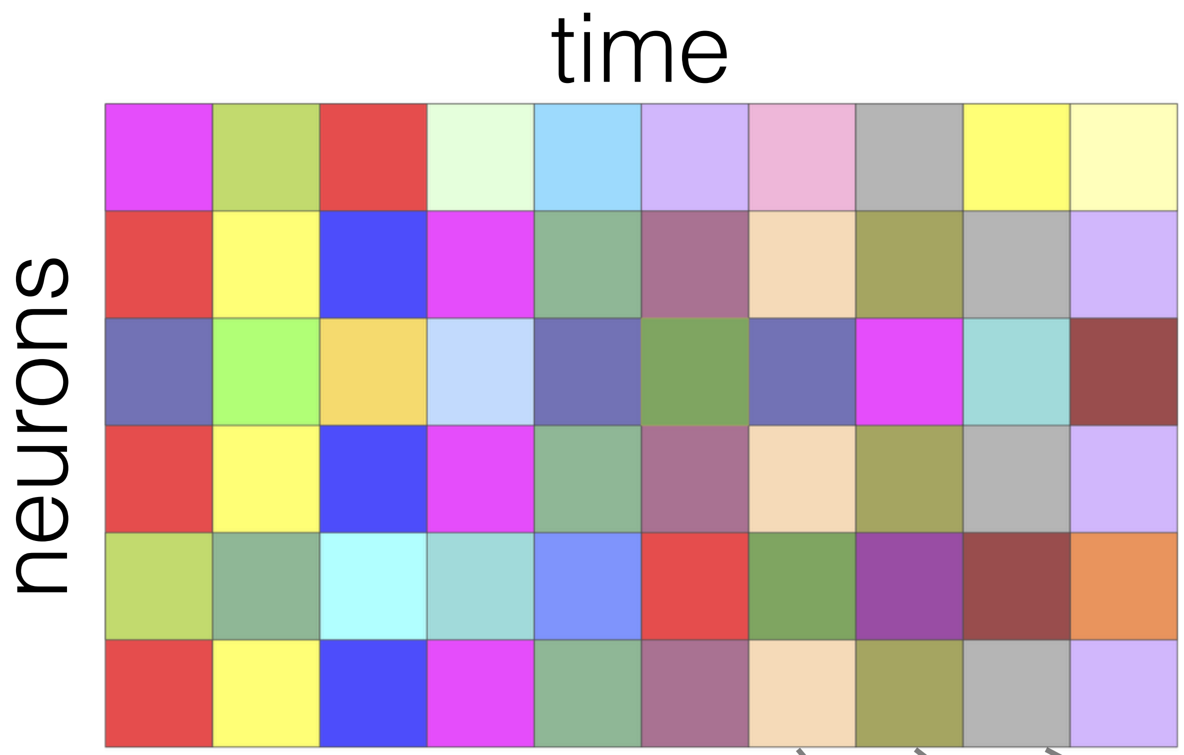


compression

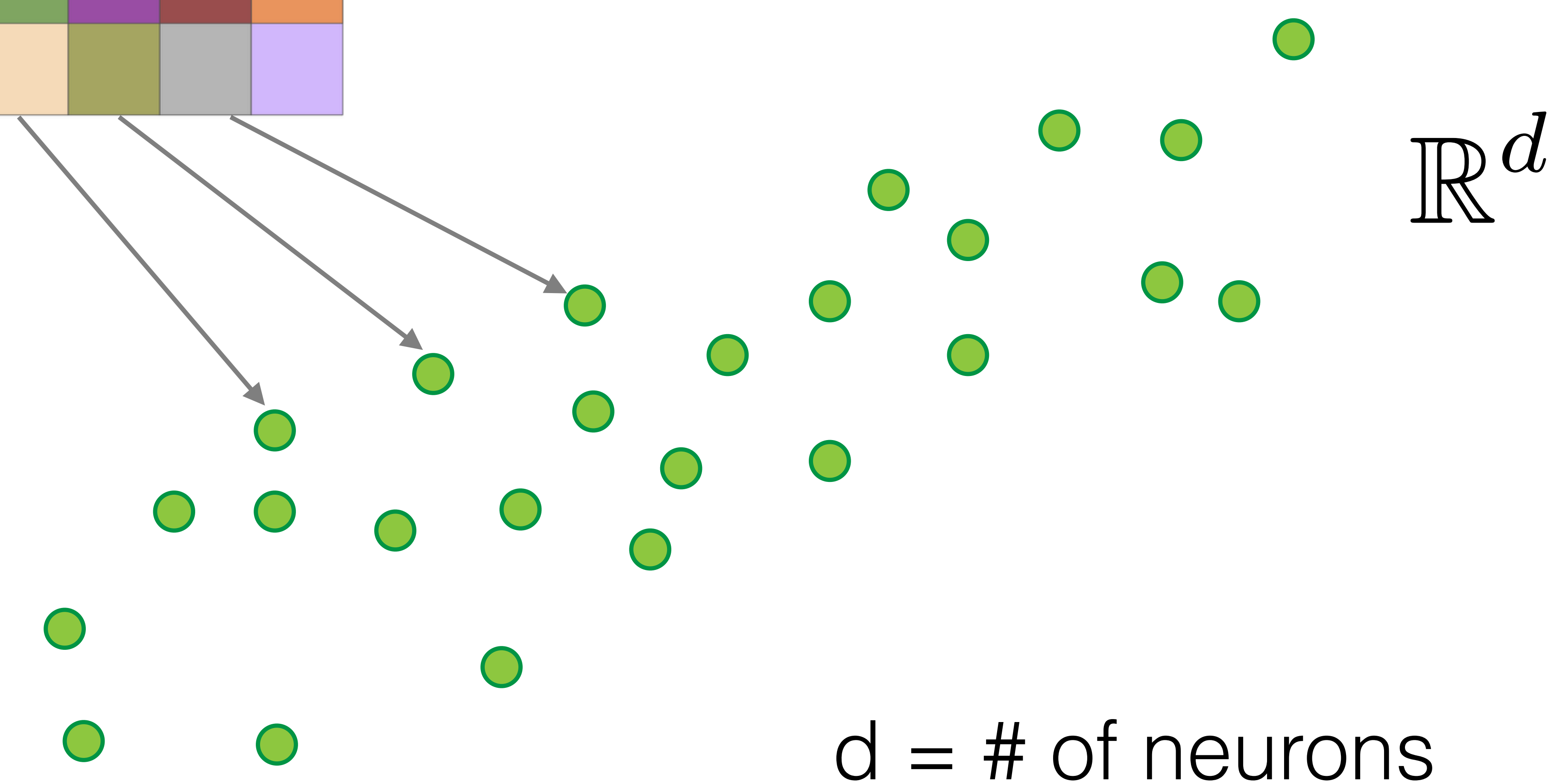
denoising

interpret complex data

low-dimensional models

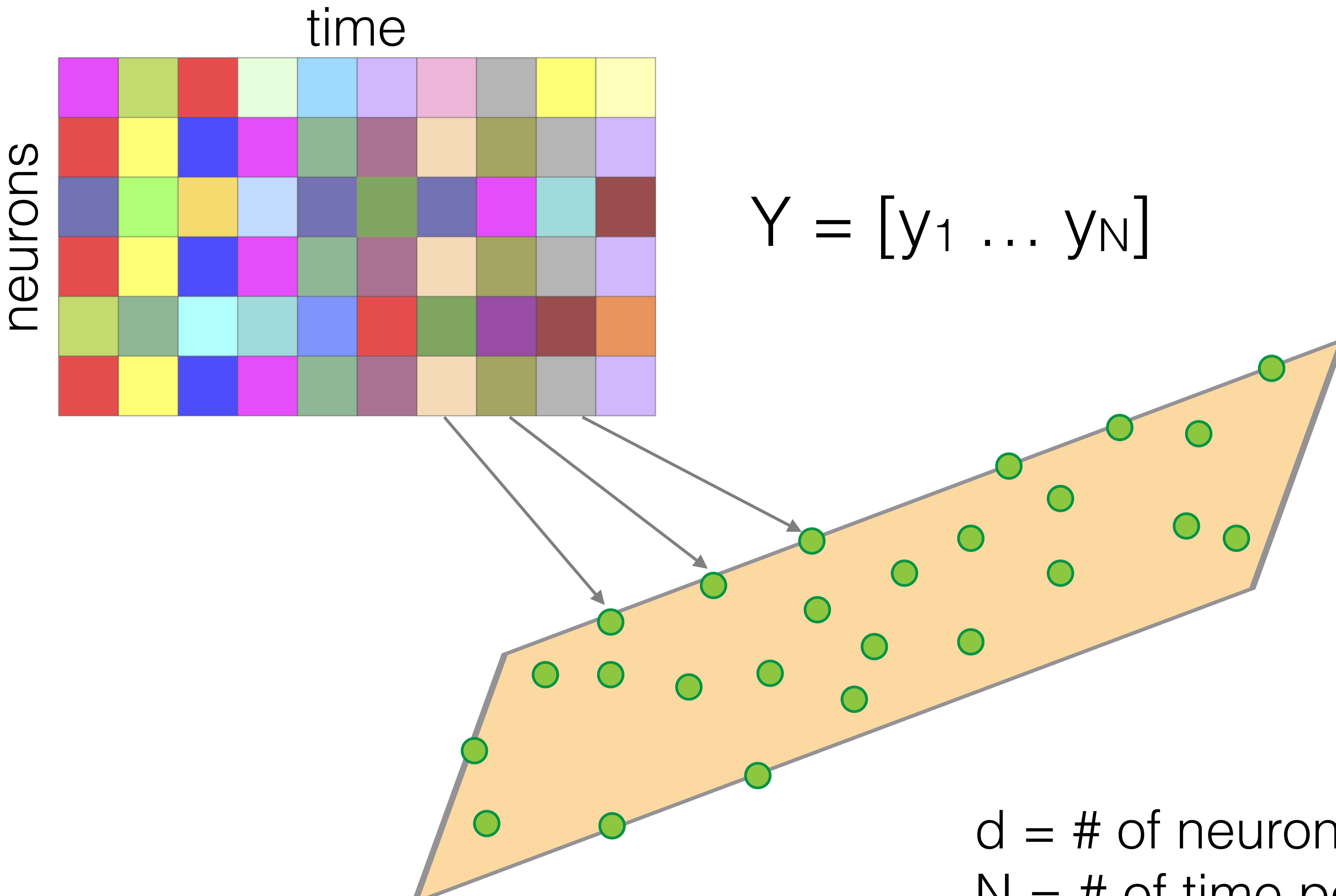


$Y = [y_1 \ldots y_N]$



d = # of neurons
N = # of time points

low-dimensional models

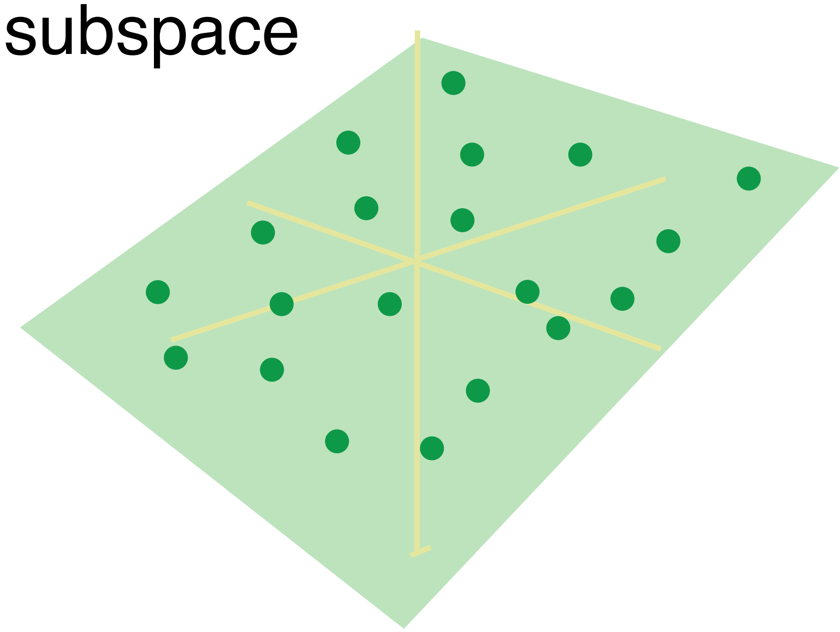


$Y = [y_1 \ldots y_N]$

$\mathbb{R}^d$

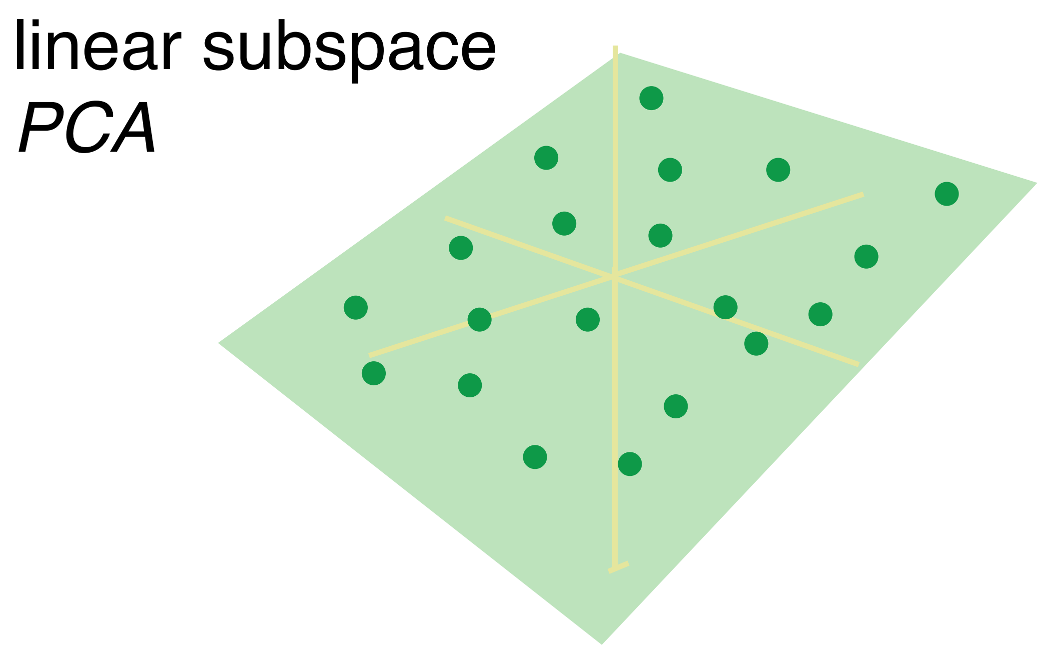
d = # of neurons
N = # of time points

low rank model



$$\min_{\mathbf{A}} \; \|\mathbf{Y} - \mathbf{A}\|_F \quad \text{s.t.} \quad \text{rank}(\mathbf{A}) \leq \text{k}$$

low rank model

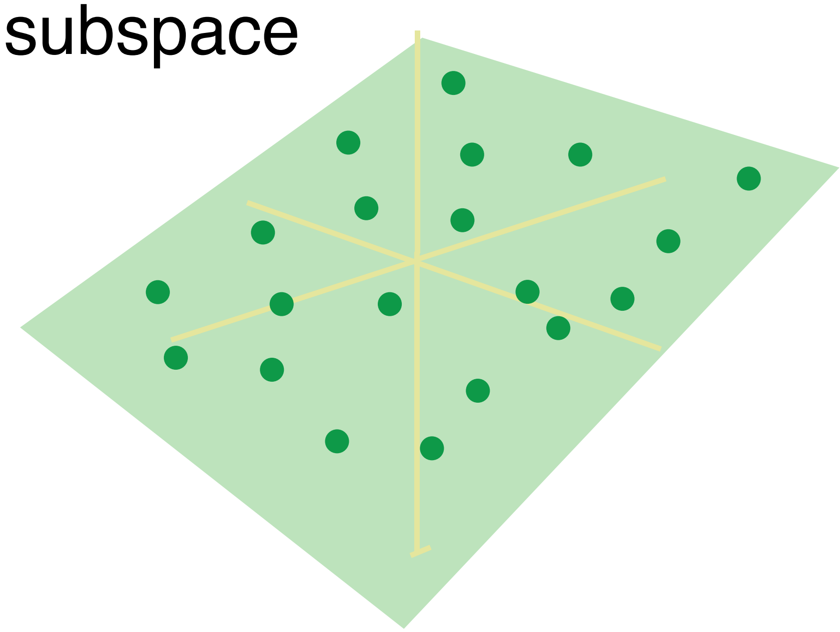


$$\min_{\mathbf{A}} \ \|\mathbf{Y} - \mathbf{A}\|_F \quad \text{s.t.} \quad \text{rank}(\mathbf{A}) \leq \text{k}$$

data matrix

low rank model

linear subspace
PCA

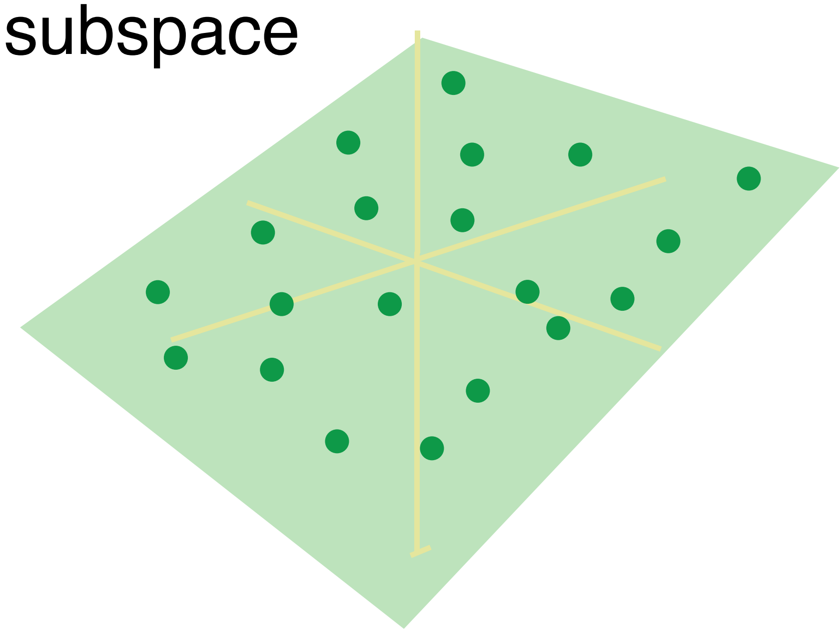


$$\min_{\mathbf{A}} \ \|\mathbf{Y} - \mathbf{A}\|_F \quad \text{s.t.} \quad \text{rank}(\mathbf{A}) \leq \text{k}$$

"low rank" approximation

low rank model

linear subspace
PCA

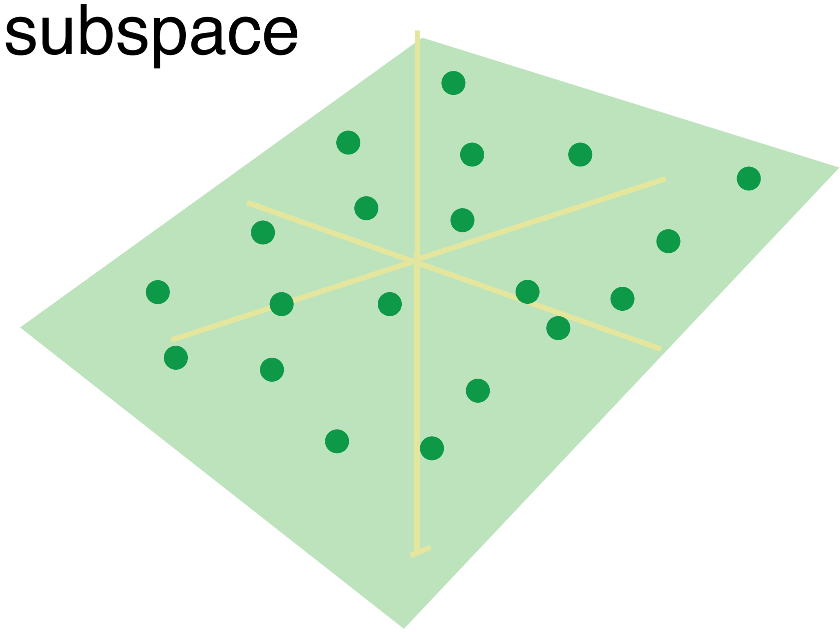


$$\min_{\mathbf{A}} \; \|\mathbf{Y} - \mathbf{A}\|_F \quad \text{s.t.} \quad \text{rank}(\mathbf{A}) \leq \text{k}$$

subject to = constraints

low rank model

linear subspace
PCA

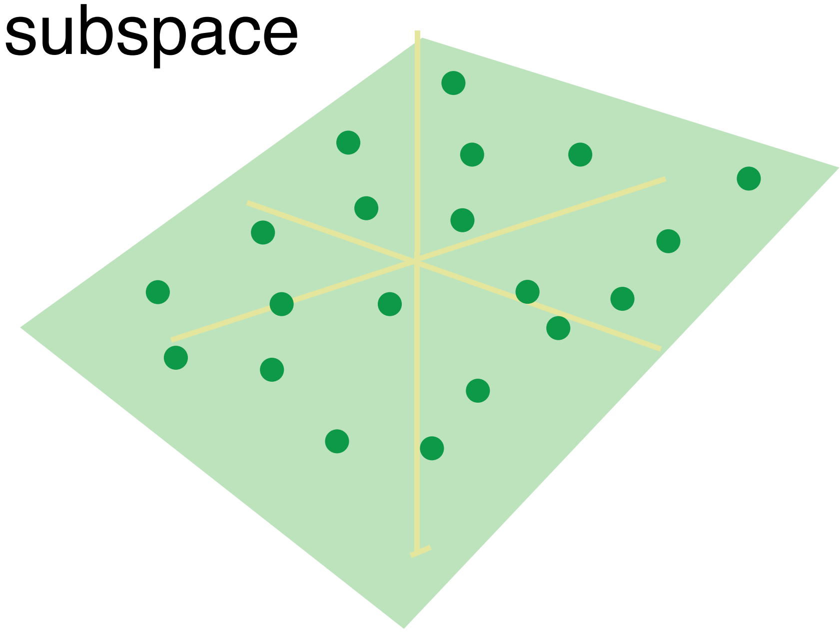


$$\min_{\mathbf{A}} \ \|\mathbf{Y} - \mathbf{A}\|_F \quad \text{s.t.} \quad \text{rank}(\mathbf{A}) \leq \text{k}$$

rank(A) = ?

low rank model

linear subspace
PCA

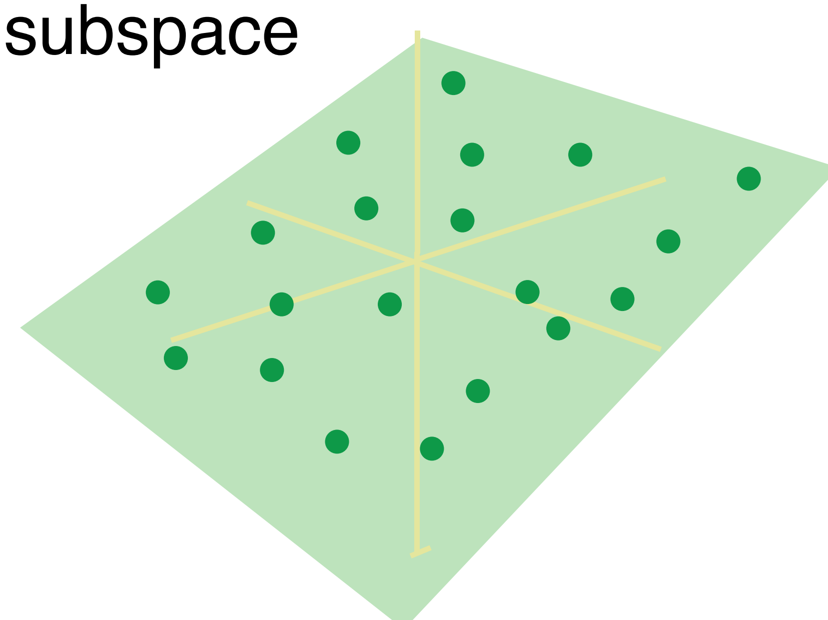


$$\min_{\mathbf{A}} \|\mathbf{Y} - \mathbf{A}\|_F \quad \text{s.t.} \quad \text{rank}(\mathbf{A}) \leq \text{k}$$

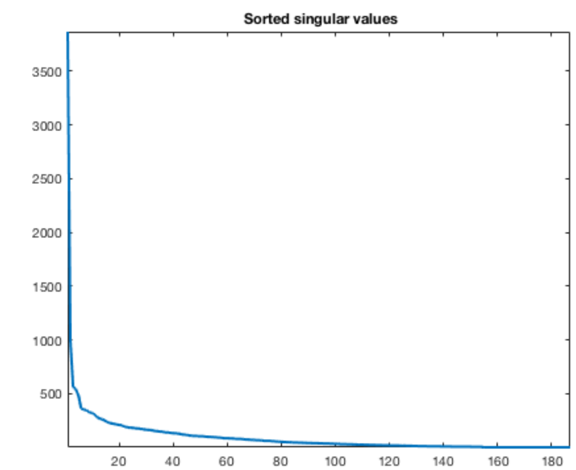
rank(A) = ?

low rank model

linear subspace
PCA



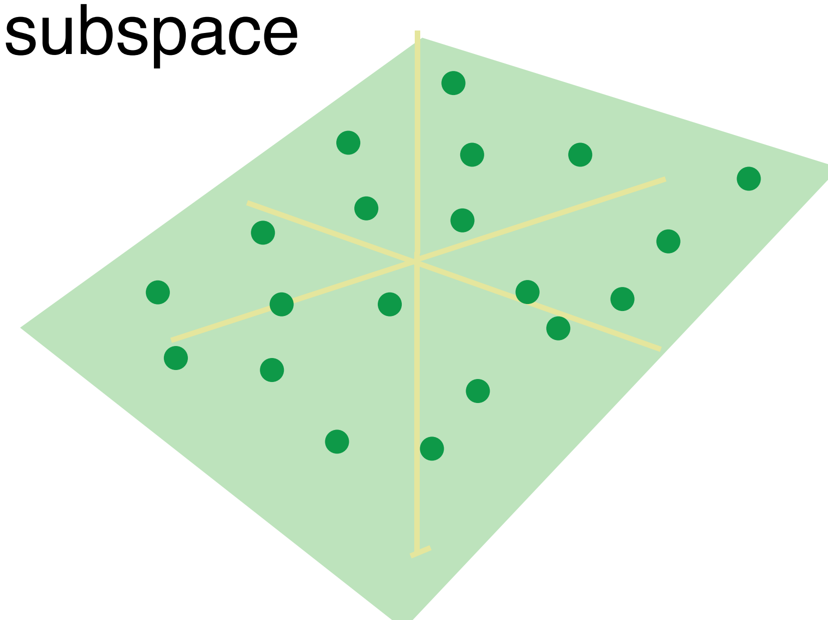
[U,S,V] =svd(Y)



$$\min_{\mathbf{A}} \ \|\mathbf{Y} - \mathbf{A}\|_F \quad \text{s.t.} \quad \text{rank}(\mathbf{A}) \leq \text{k}$$

$$\mathbf{A} = \mathbf{U}_k \mathbf{S}_k \mathbf{V}_k^T$$ (truncated SVD)

low rank model



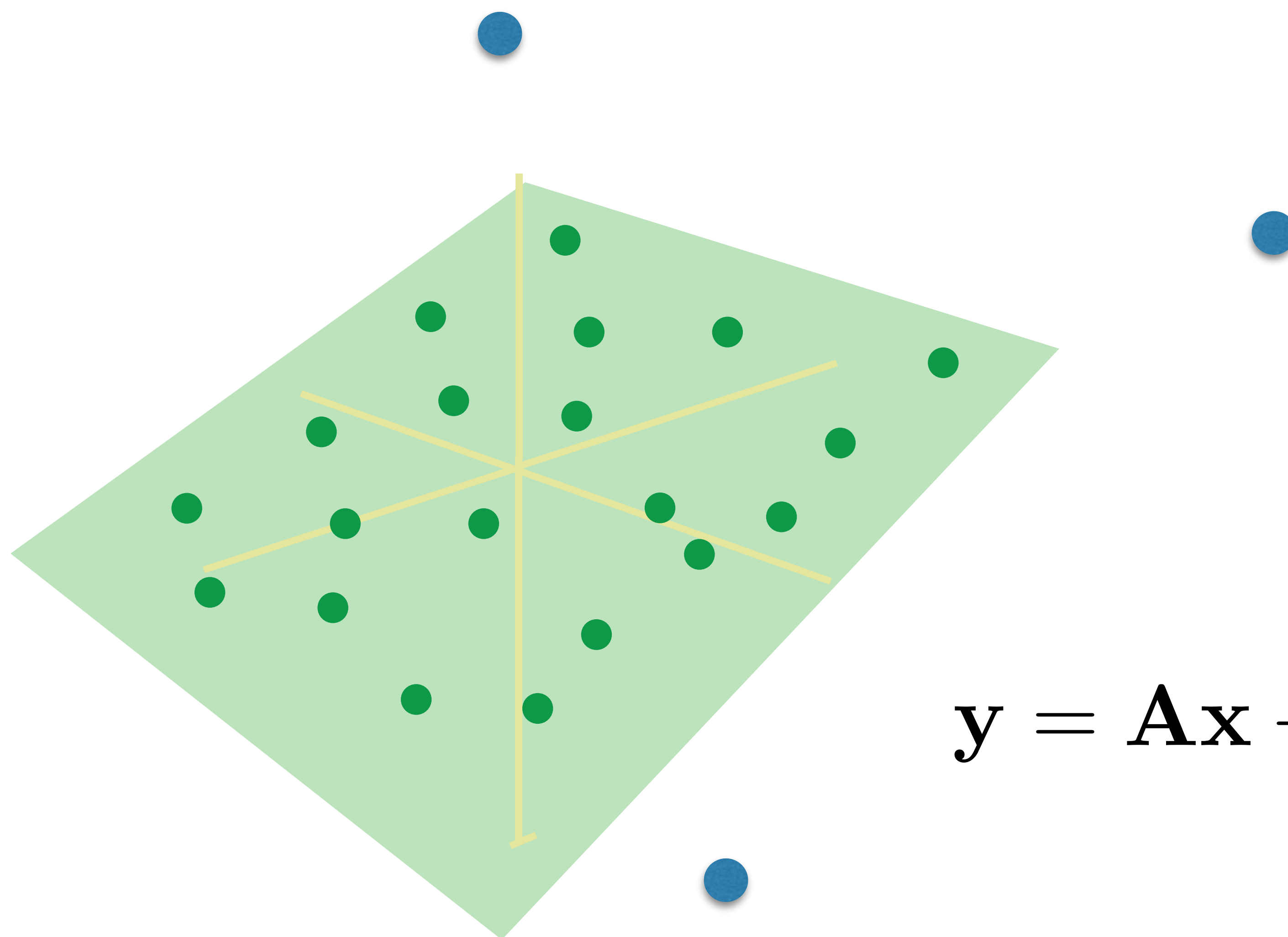
Covariance matrix

$$C = (\mathbf{Y} - \overline{\mathbf{y}})^T (\mathbf{Y} - \overline{\mathbf{y}})^T$$

PCA:

1. Compute the covariance matrix (C)
2. Compute eigenvalue decomposition of C
3. Output > top k eigenvectors and their eigenvalues

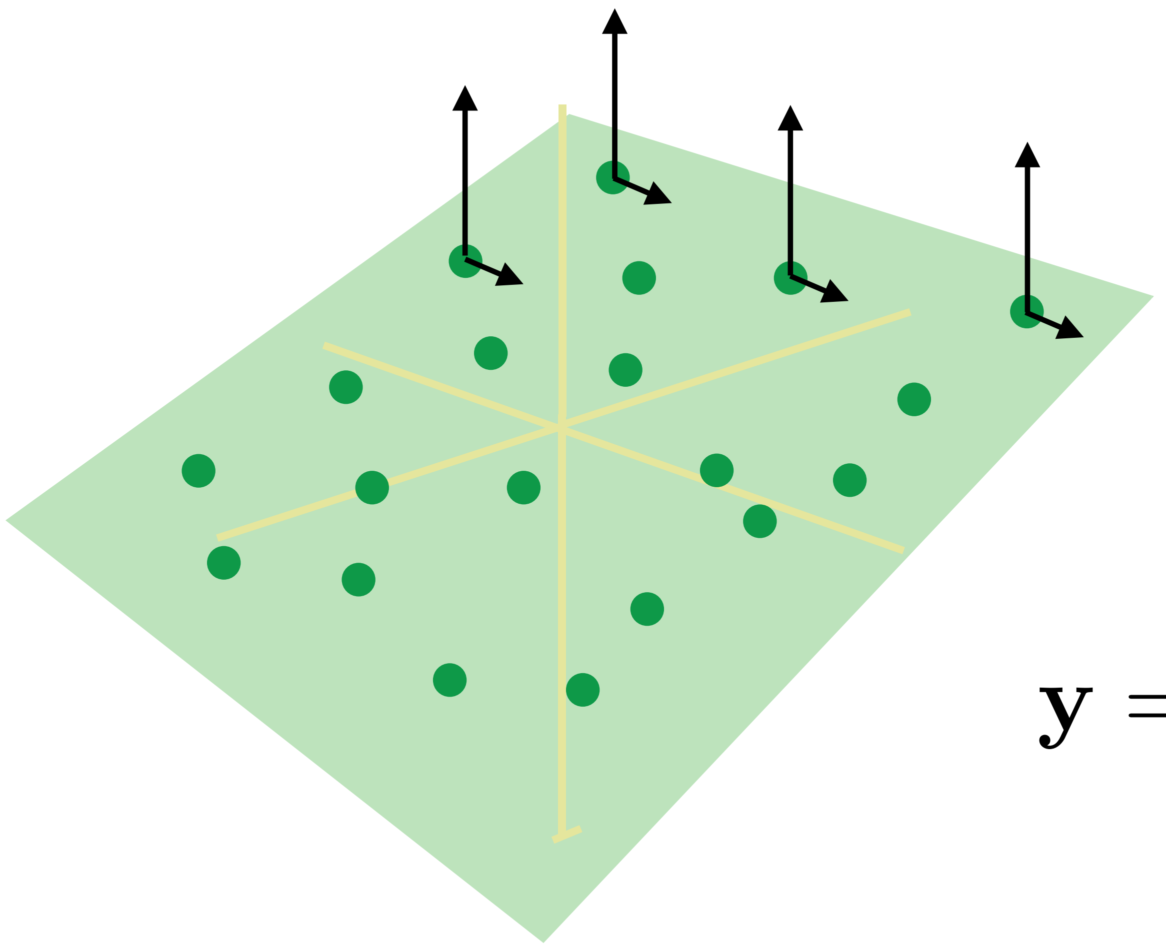
extensions of PCA



$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$$

1. **Robust PCA - sparse LARGE errors**
2. Factor analysis (FA) - noise of unequal variance
3. Non-negative matrix factorization (NMF)

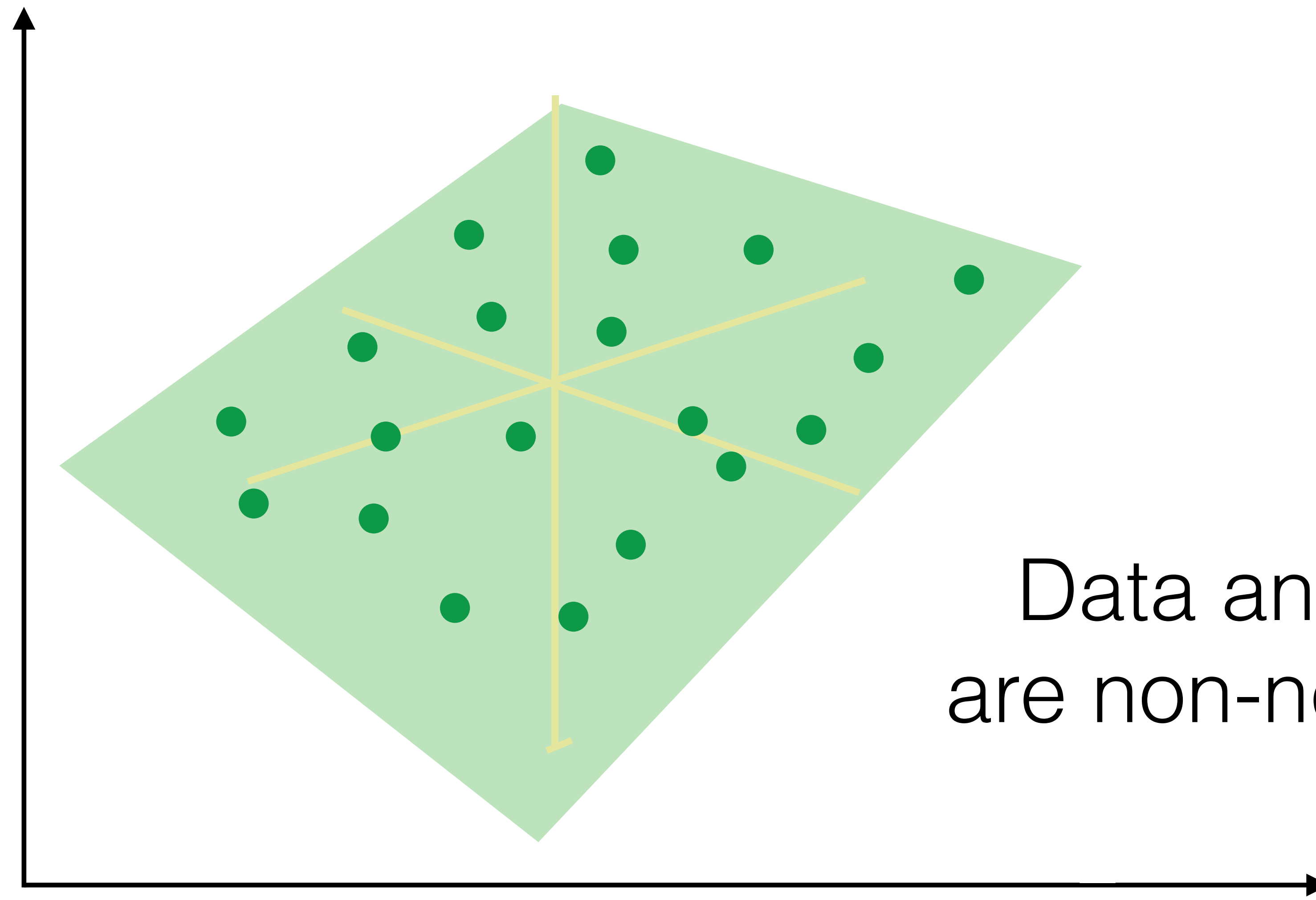
extensions of PCA



$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$$

1. Robust PCA - sparse errors
2. **Factor analysis (FA) - noise of unequal variance**
3. Non-negative matrix factorization (NMF)

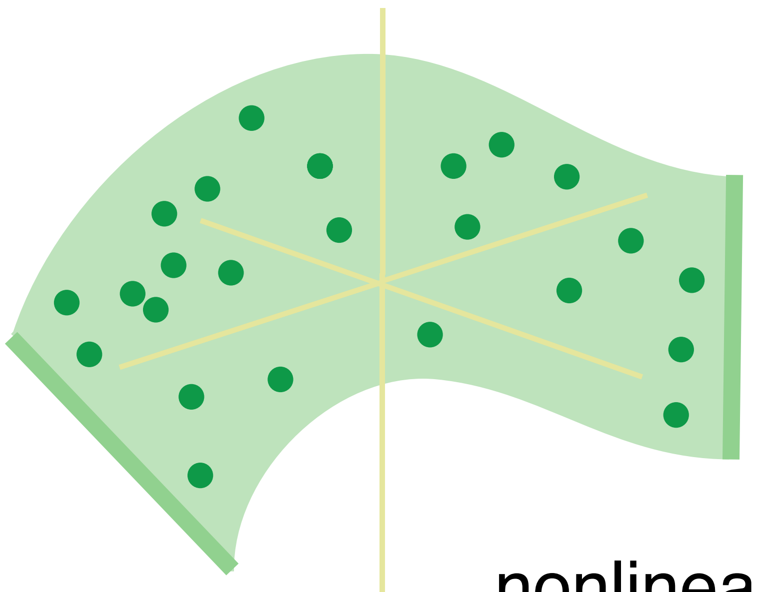
extensions of PCA



Data and PCs
are non-negative!

1. Robust PCA - sparse errors
2. Factor analysis (FA) - noise of unequal variance
3. **Non-negative matrix factorization (NMF)**

nonlinear models (manifolds)

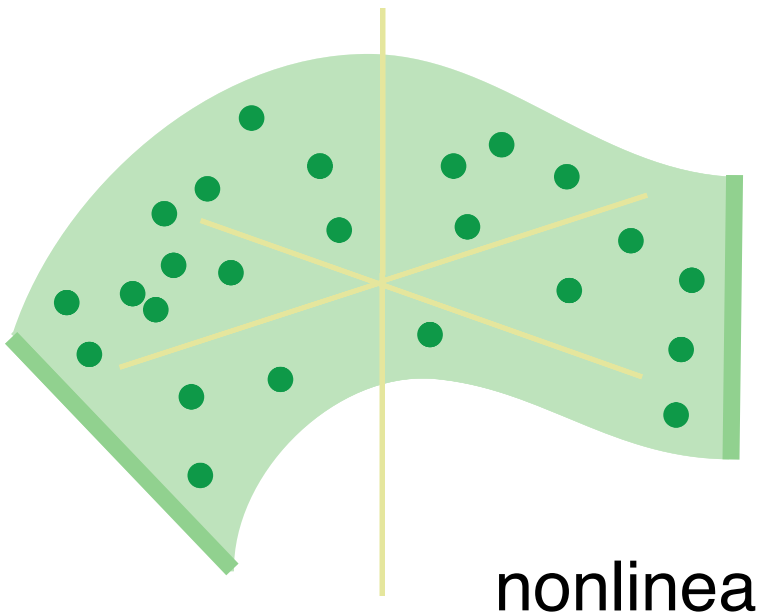


nonlinear manifold
Isomap, LLE

$$\min_{\mathcal{P}} \ |d(y_i, y_j) - d(\mathcal{P}y_i, \mathcal{P}y_j)|$$

distance between original data points

nonlinear models (manifolds)



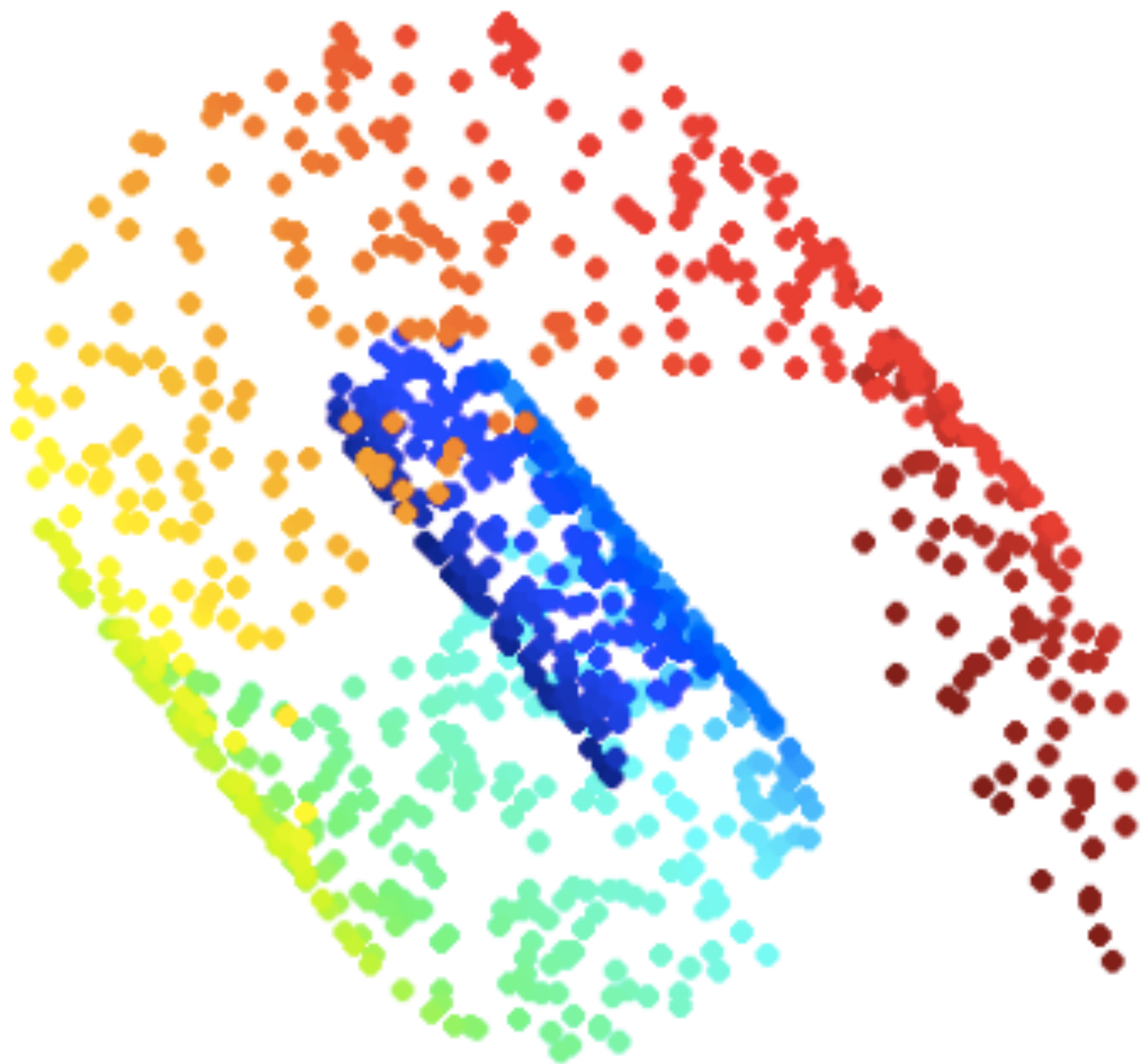
nonlinear manifold
Isomap, LLE

$$\min_{\mathcal{P}} \; \left| d(y_i, y_j) - d(\mathcal{P}y_i, \mathcal{P}y_j) \right|$$

distance between projected data points

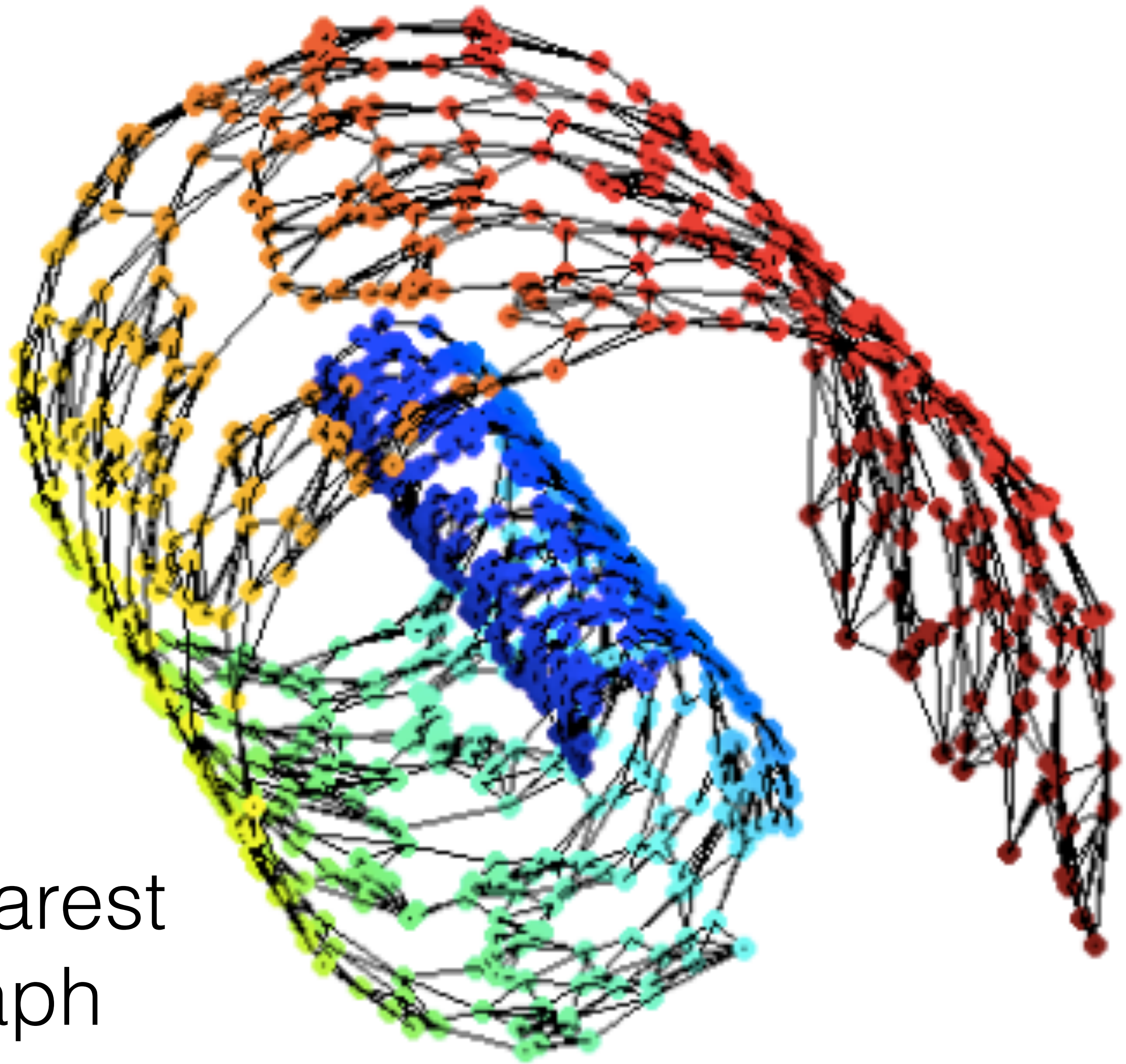
nonlinear models (manifolds)

swiss roll



http://www.numerical-tours.com/matlab/shapes_7_isomap/

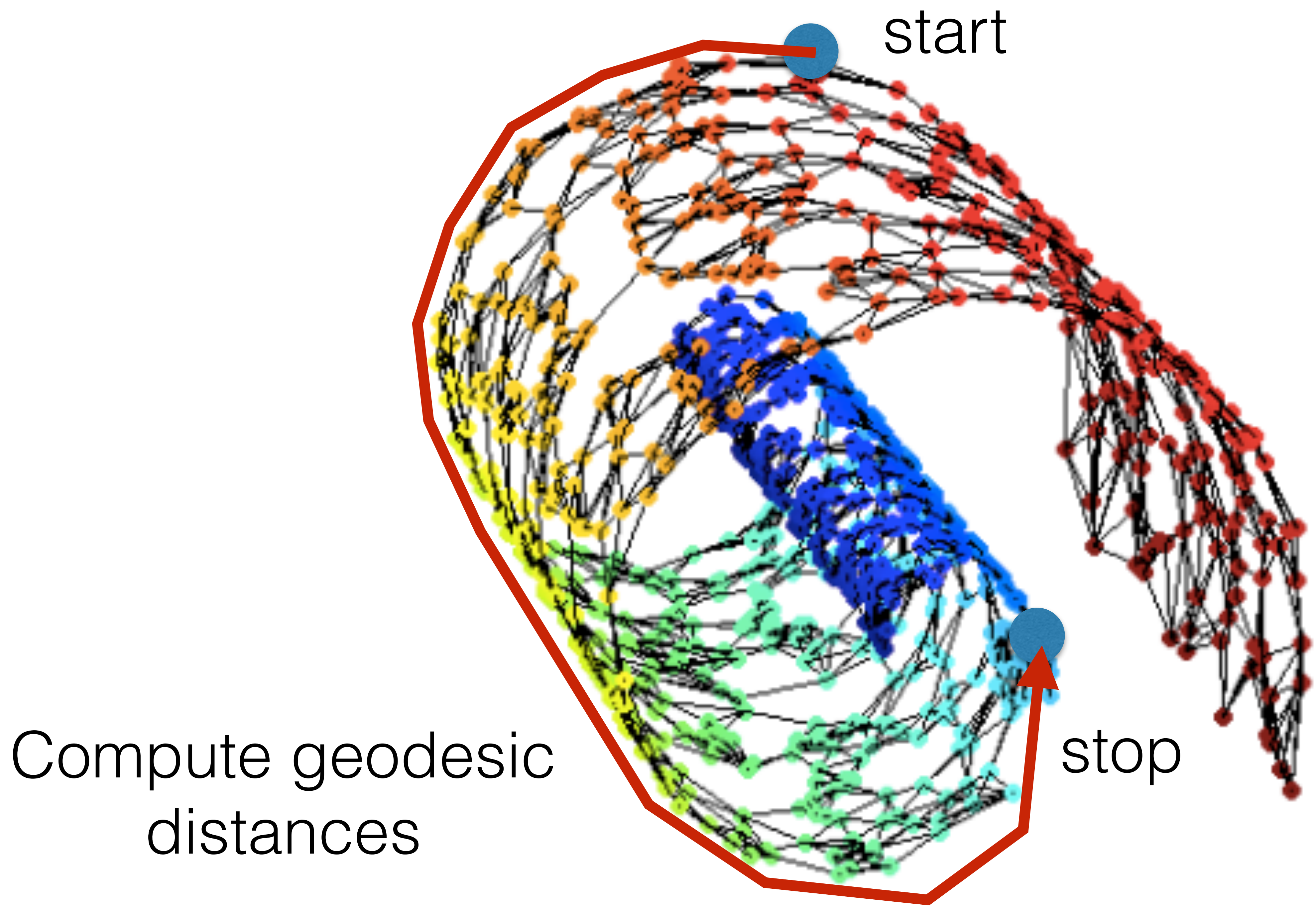
nonlinear models (manifolds)



Compute k-nearest neighbor graph

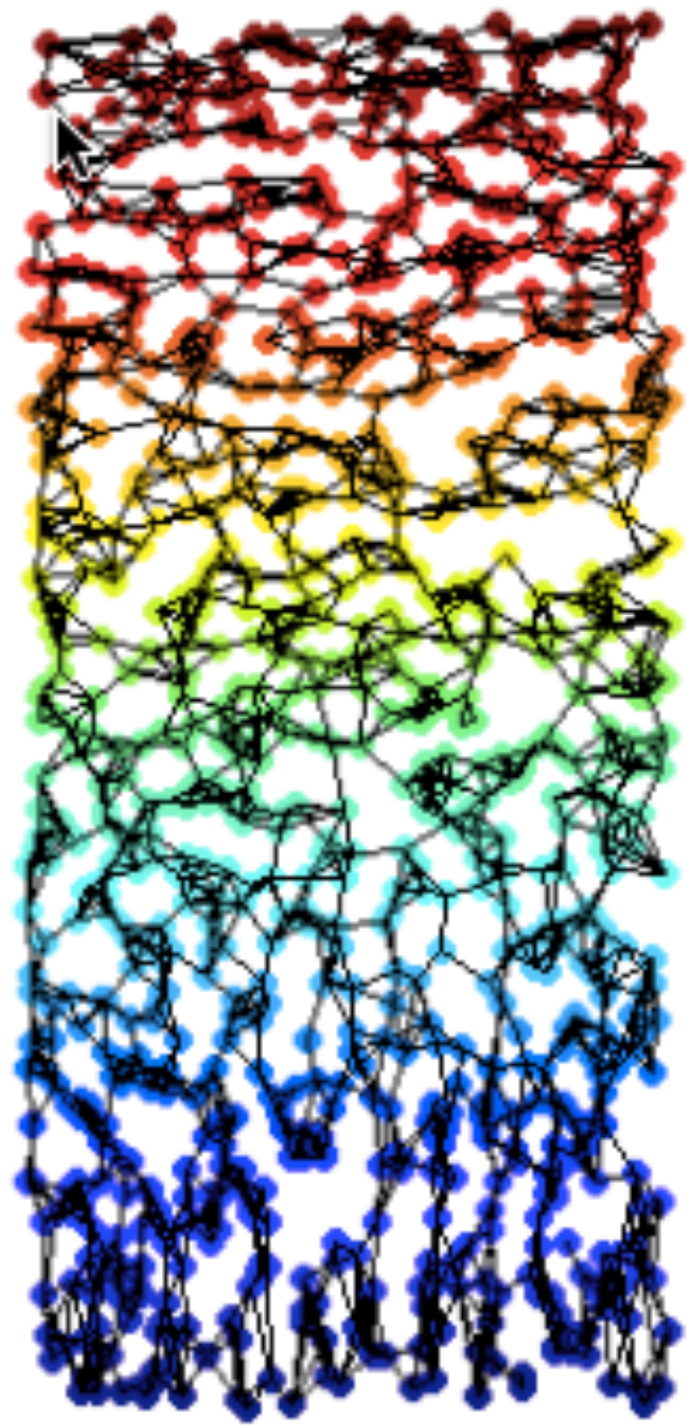
http://www.numerical-tours.com/matlab/shapes_7_isomap/

nonlinear models (manifolds)



http://www.numerical-tours.com/matlab/shapes_7_isomap/

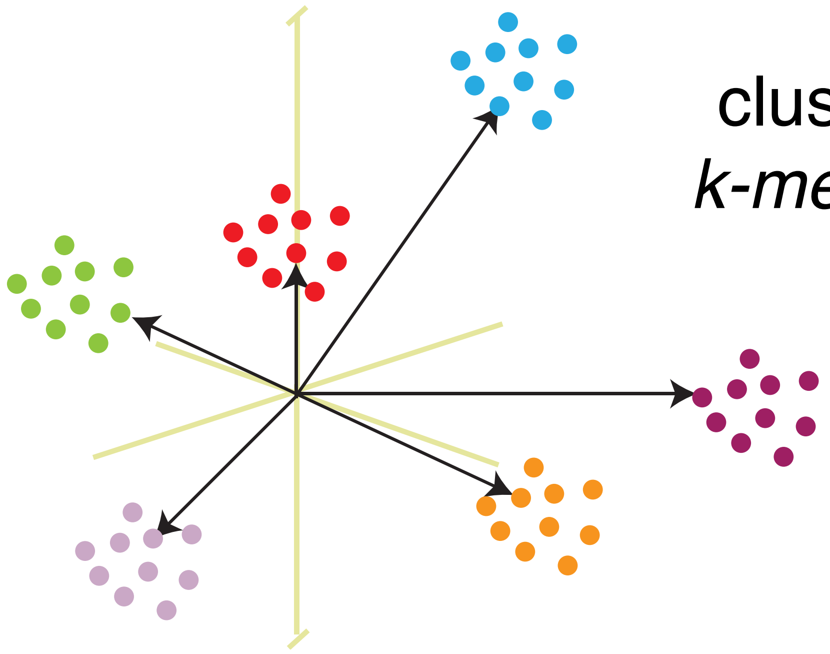
nonlinear models (manifolds)



Compute leading eigenvectors

http://www.numerical-tours.com/matlab/shapes_7_isomap/

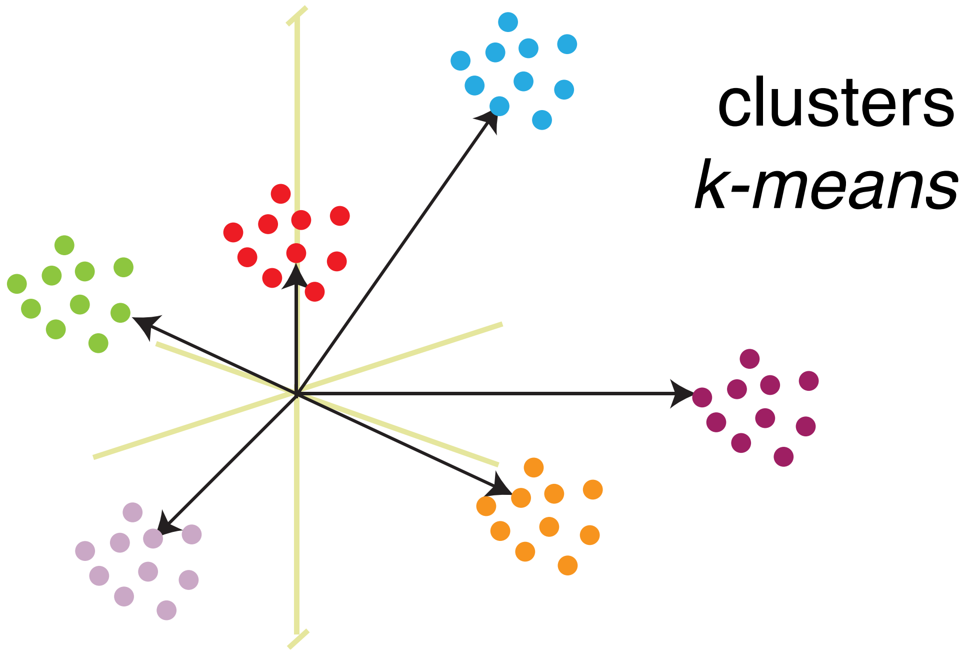
cluster model



$$\min_{\mathbf{c}_1,\cdots,\mathbf{c}_k} \sum_{j=1}^{k} \sum_{i\in\Omega_j} \|y_i - c_j\|_2$$

ith data point

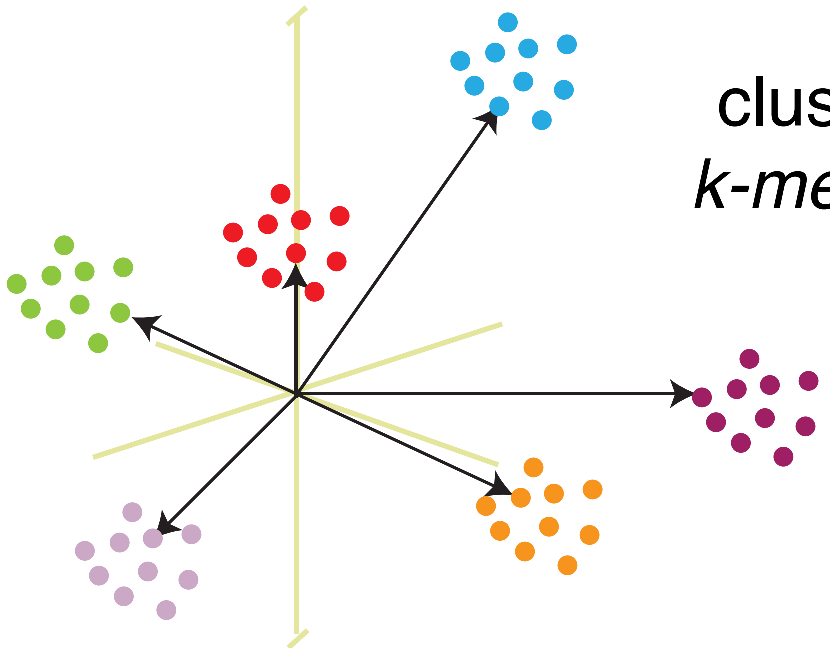
cluster model



$$\min_{\mathbf{c}_1, \cdots, \mathbf{c}_k} \sum_{j=1}^{k} \sum_{i \in \Omega_j} \|y_i - c_j\|_2$$

jth cluster center

cluster model

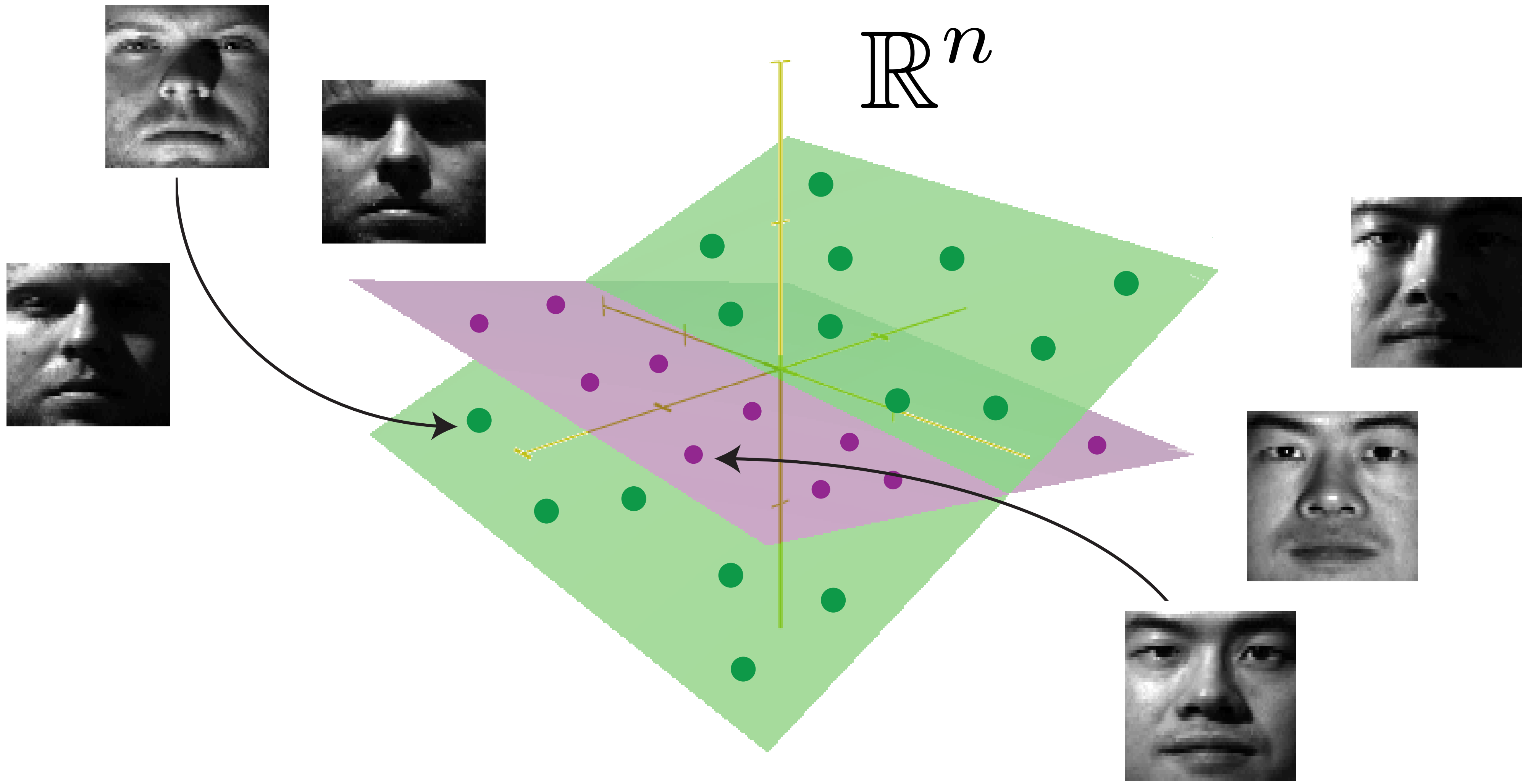


$$\min_{\mathbf{c}_1,\cdots,\mathbf{c}_k} \sum_{j=1}^{k} \sum_{i\in\Omega_j} \|y_i - c_j\|_2$$

kmeans:

1. Randomly initialize cluster centers
2. Assign each data point to its closest cluster center
3. Update cluster centers (mean of all assigned points)
4. Iterate steps 2-3 until convergence

union of subspaces

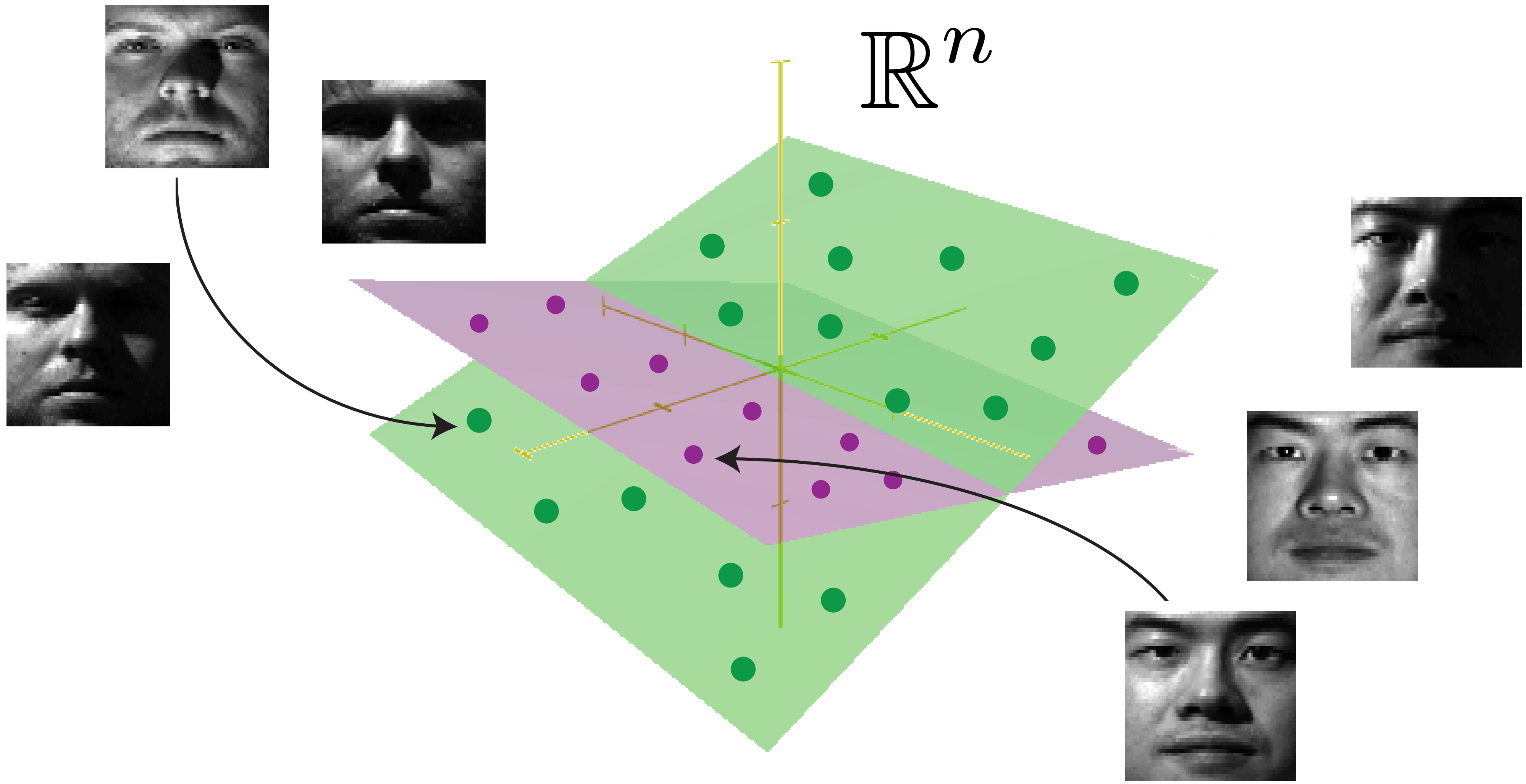


$$\min_{\mathbf{A}_i,\Omega_i} \quad \sum \|\mathbf{Y}_{\Omega_i} - \mathbf{A}_i\|_F \quad \text{s.t.} \quad \text{rank}(\mathbf{A}_i) \leq k_i$$

↑

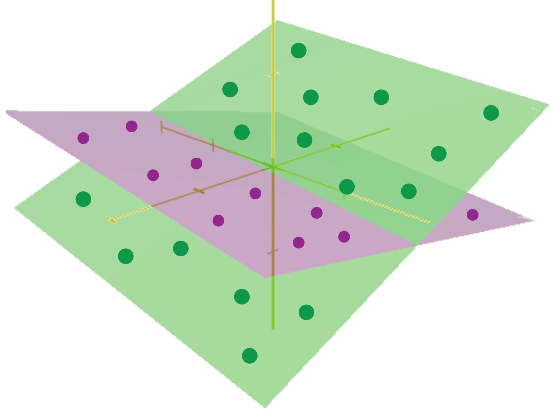
subset of data in ith subspace

union of subspaces

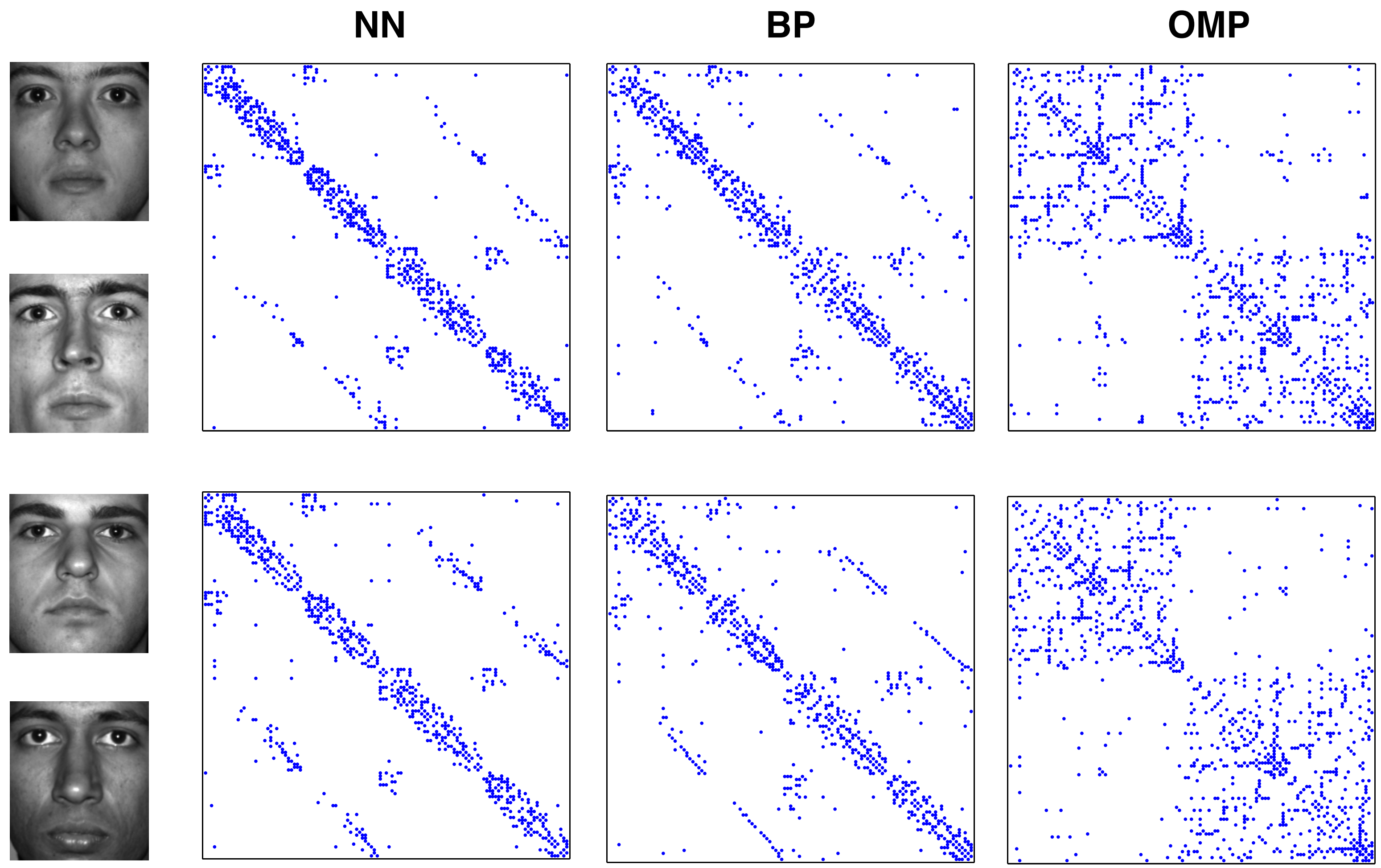


$$\min_{\mathbf{A}_i, \Omega_i} \quad \sum \|\mathbf{Y}_{\Omega_i} - \mathbf{A}_i\|_F \quad \text{s.t.} \quad \text{rank}(\mathbf{A}_i) \leq k_i$$

low rank approx of ith cluster



union of subspaces

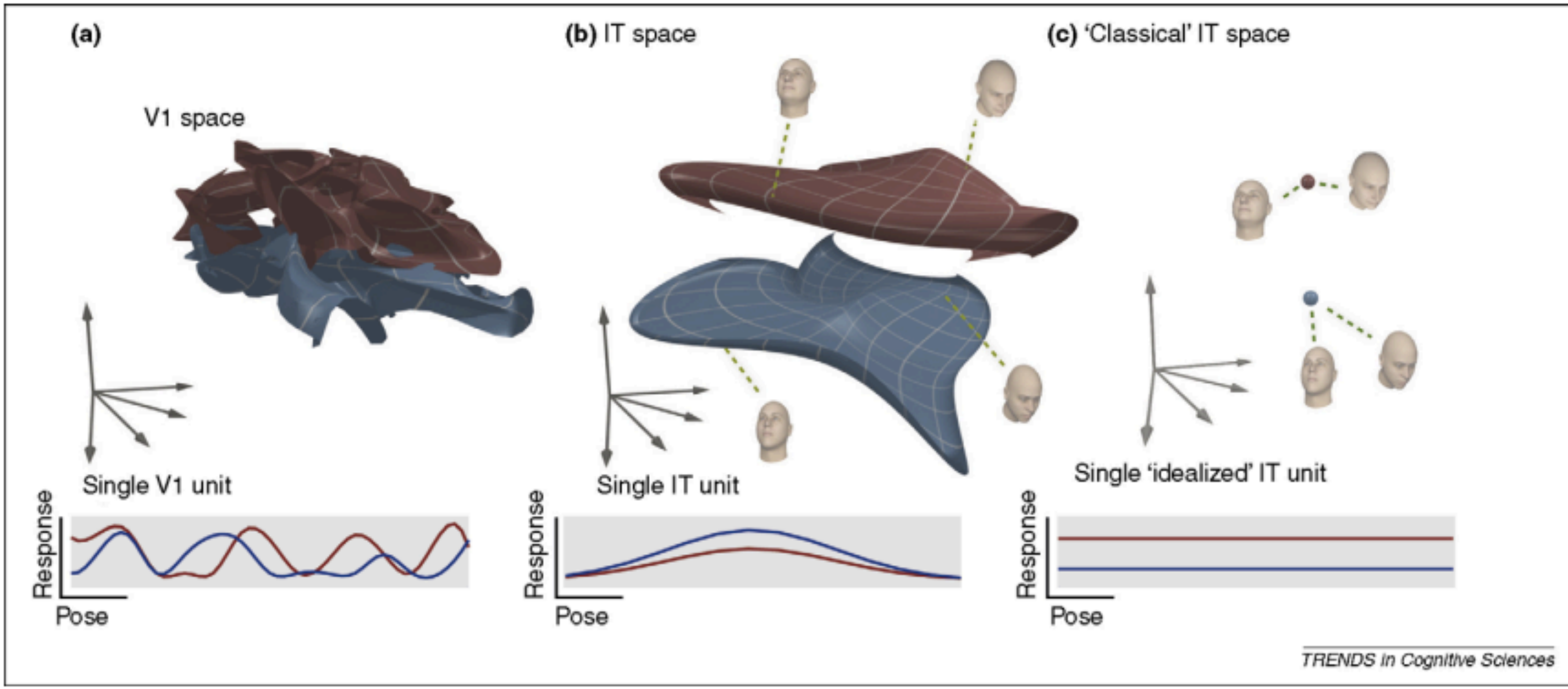


sparse subspace clustering (SSC):

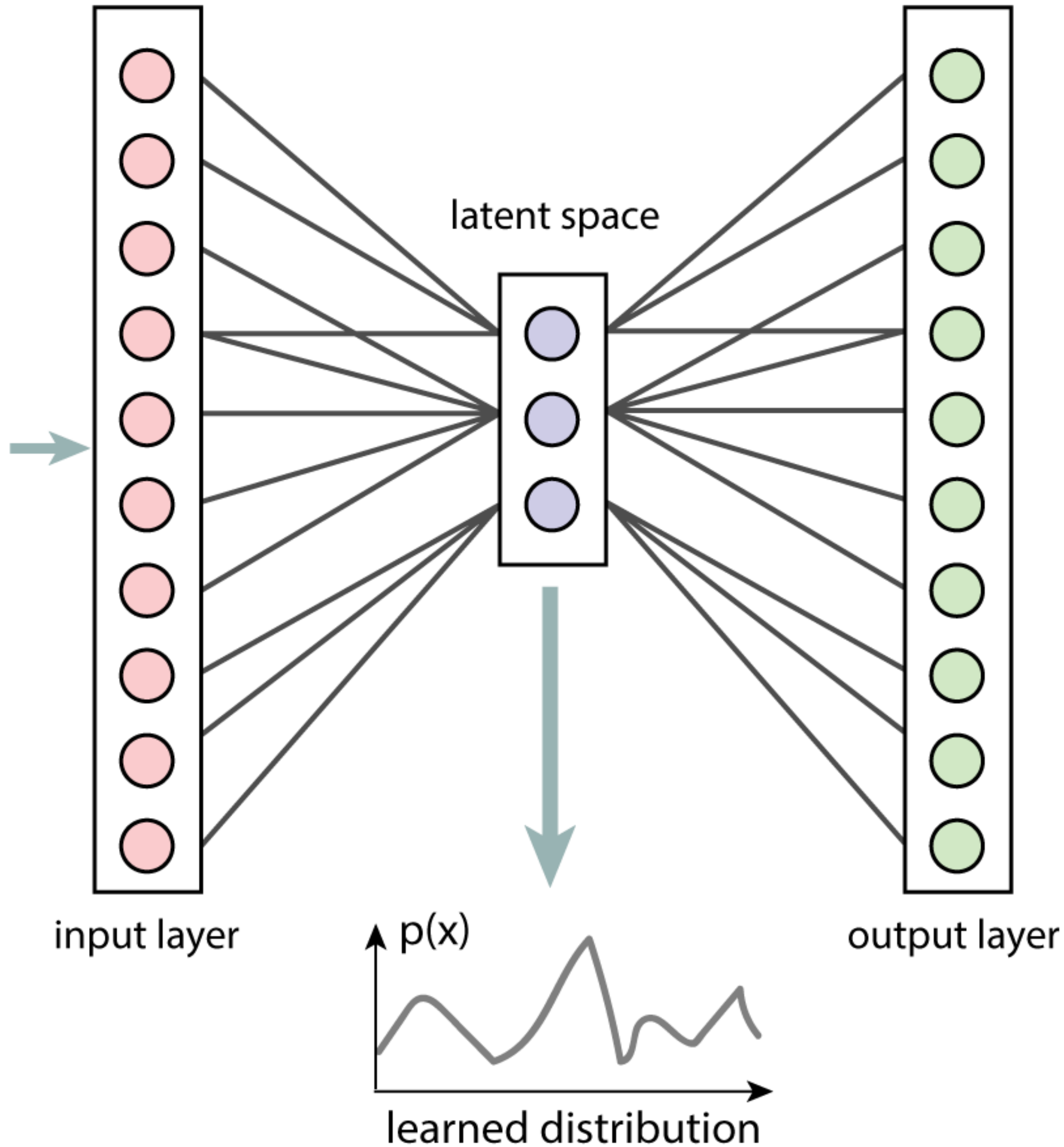
1. Compute the subspace affinity matrix (C)
2. Cluster the affinity matrix C
3. For each subspace cluster, run SVD and get low rank approximation

Dyer, JMLR 2013

“tangled” manifolds

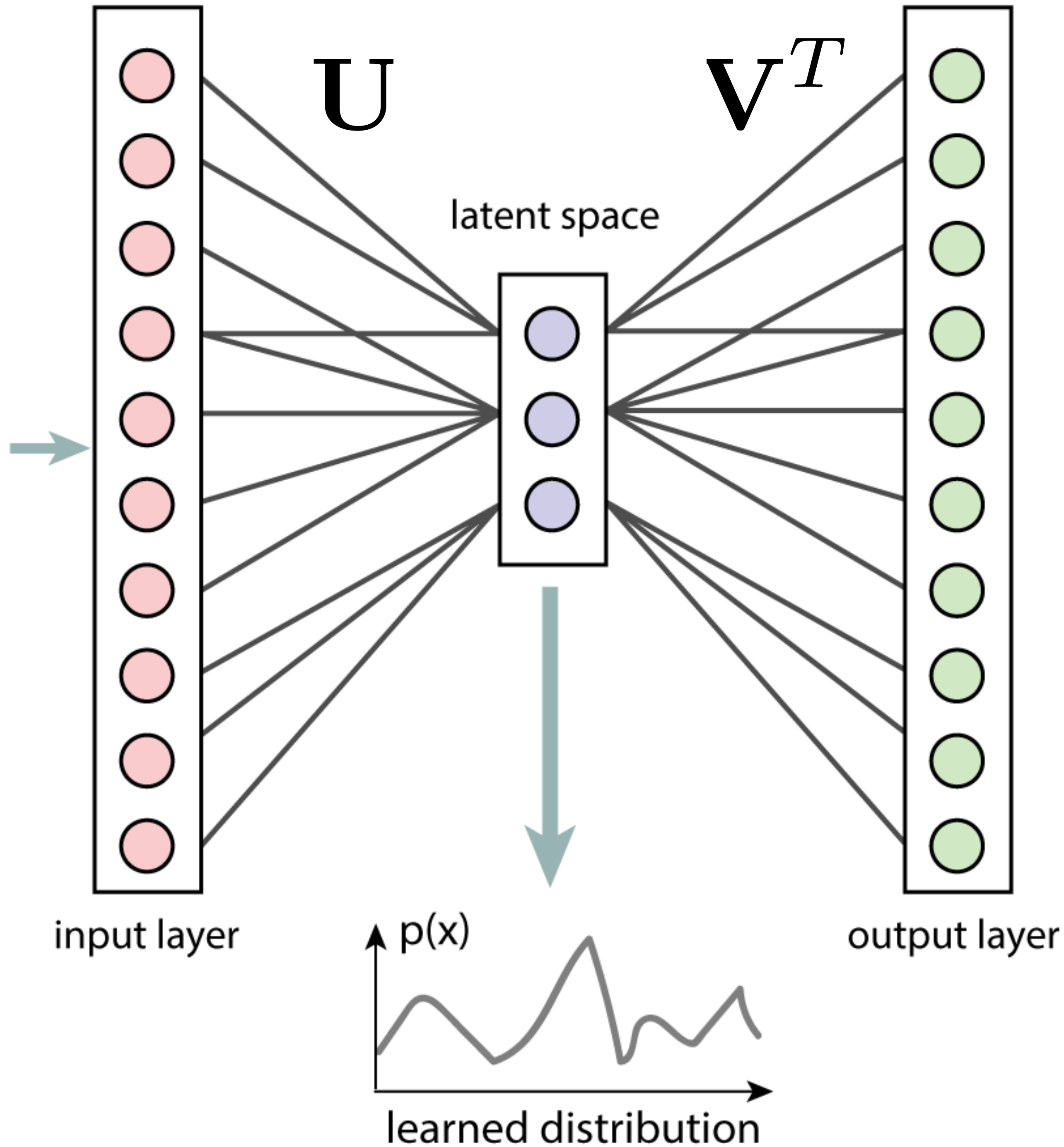


autoencoders



$$\min_{f} \quad f(\mathbf{Y}_{in}, \mathbf{Y}_{out})$$

linear autoencoder -> PCA

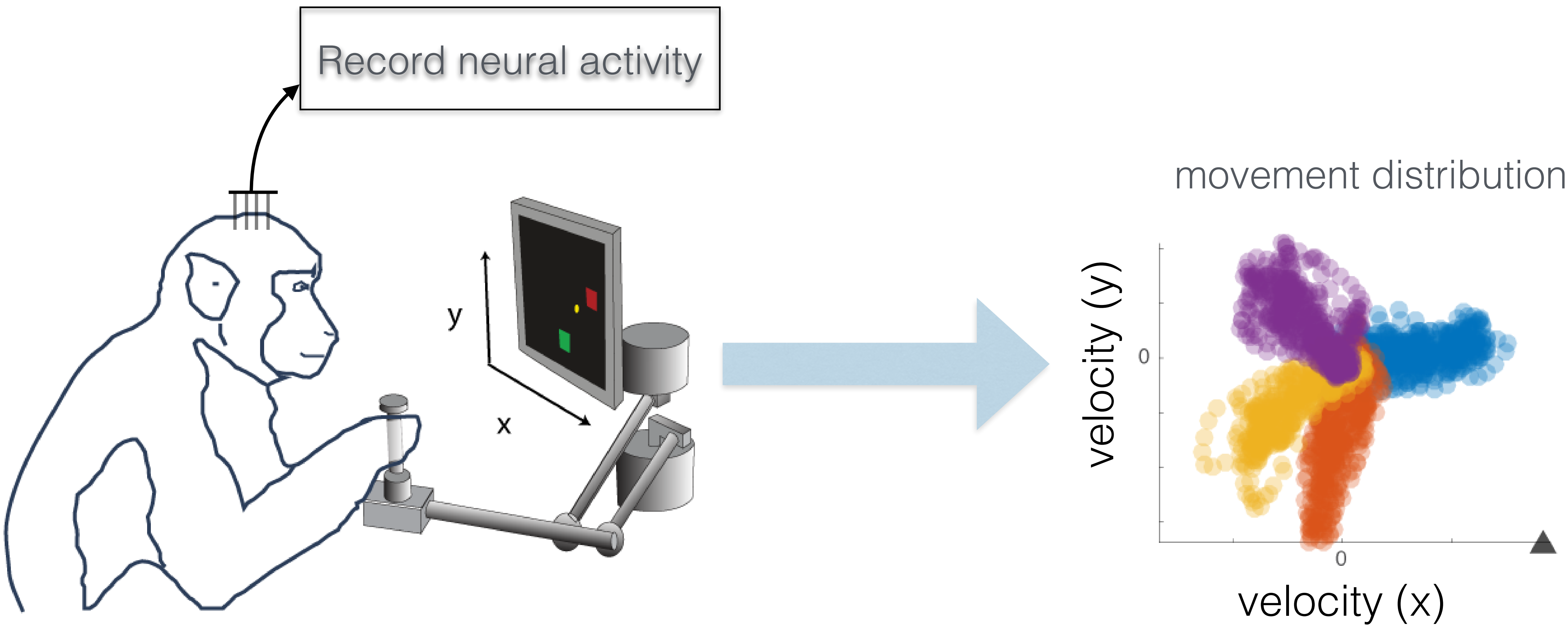


$$\mathbf{Y} \approx \mathbf{U}\mathbf{S}\mathbf{V}^T$$

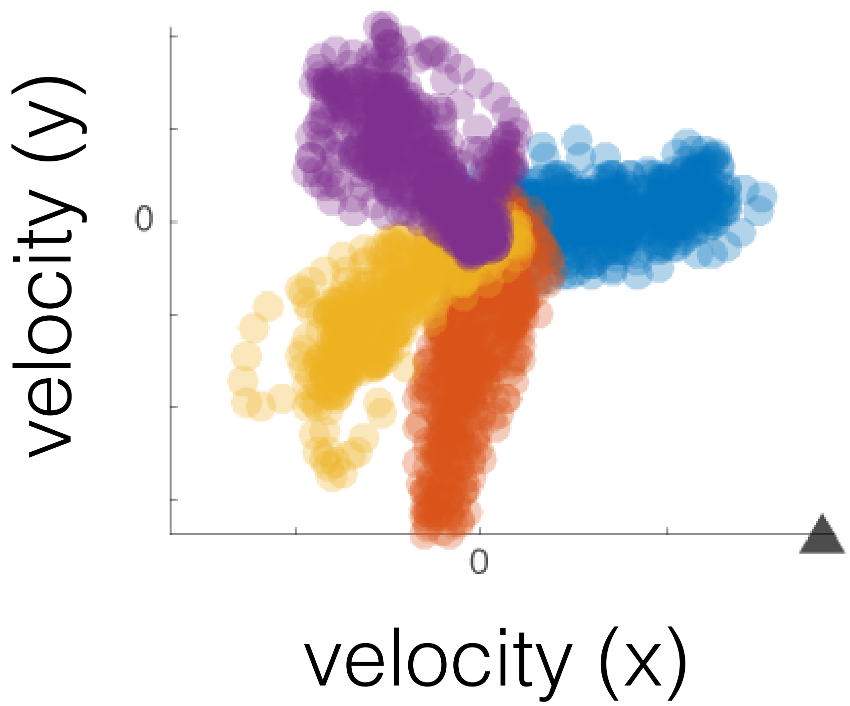
$$\min_{f} \quad f(\mathbf{Y}_{in}, \mathbf{Y}_{out})$$

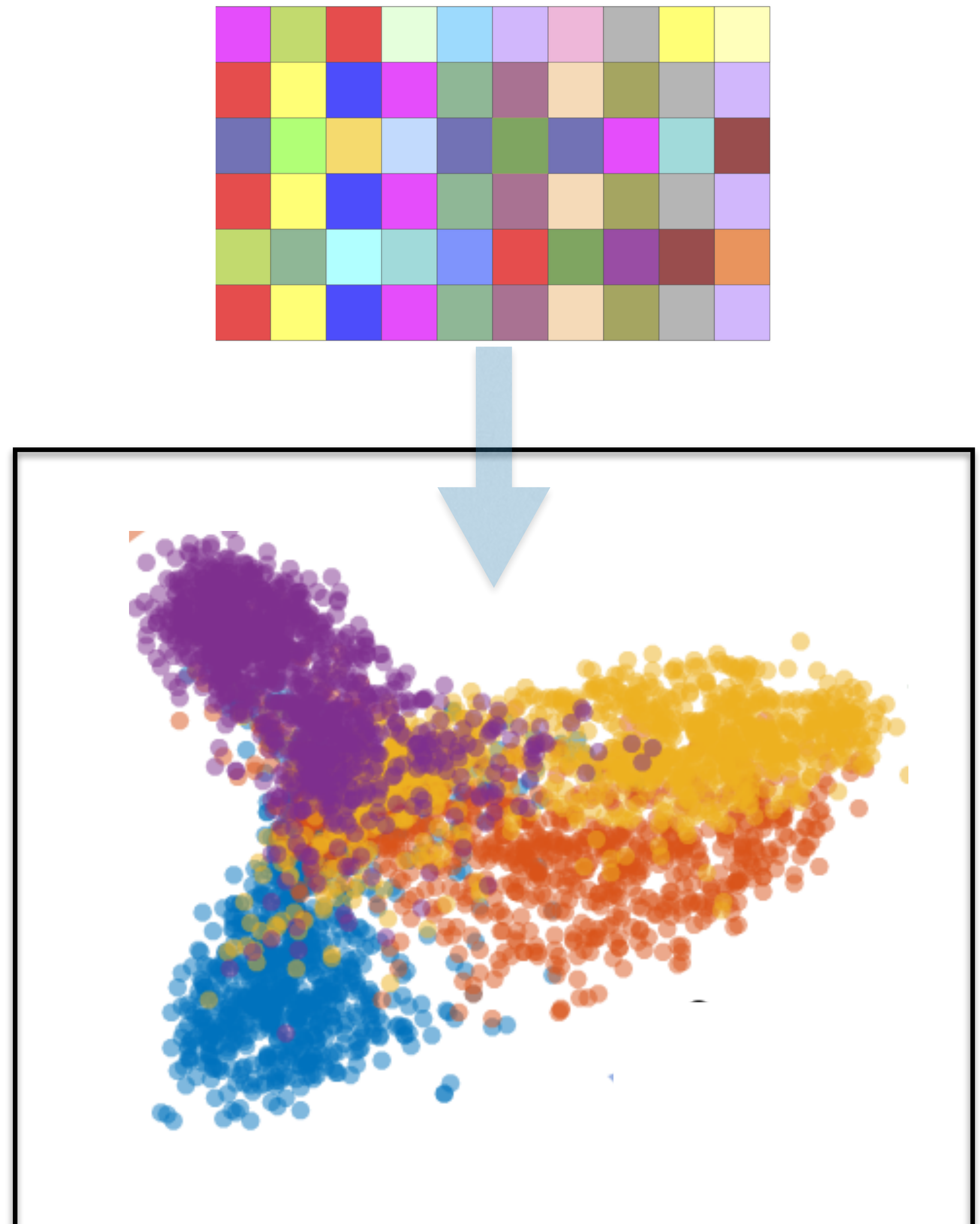
application to
movement decoding

movement decoding



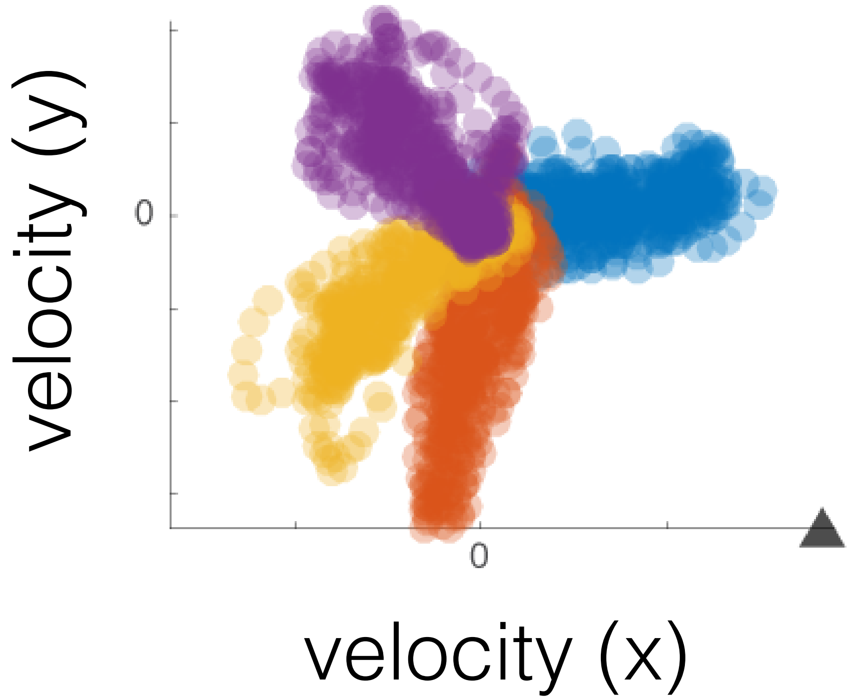
neural responses are low-d

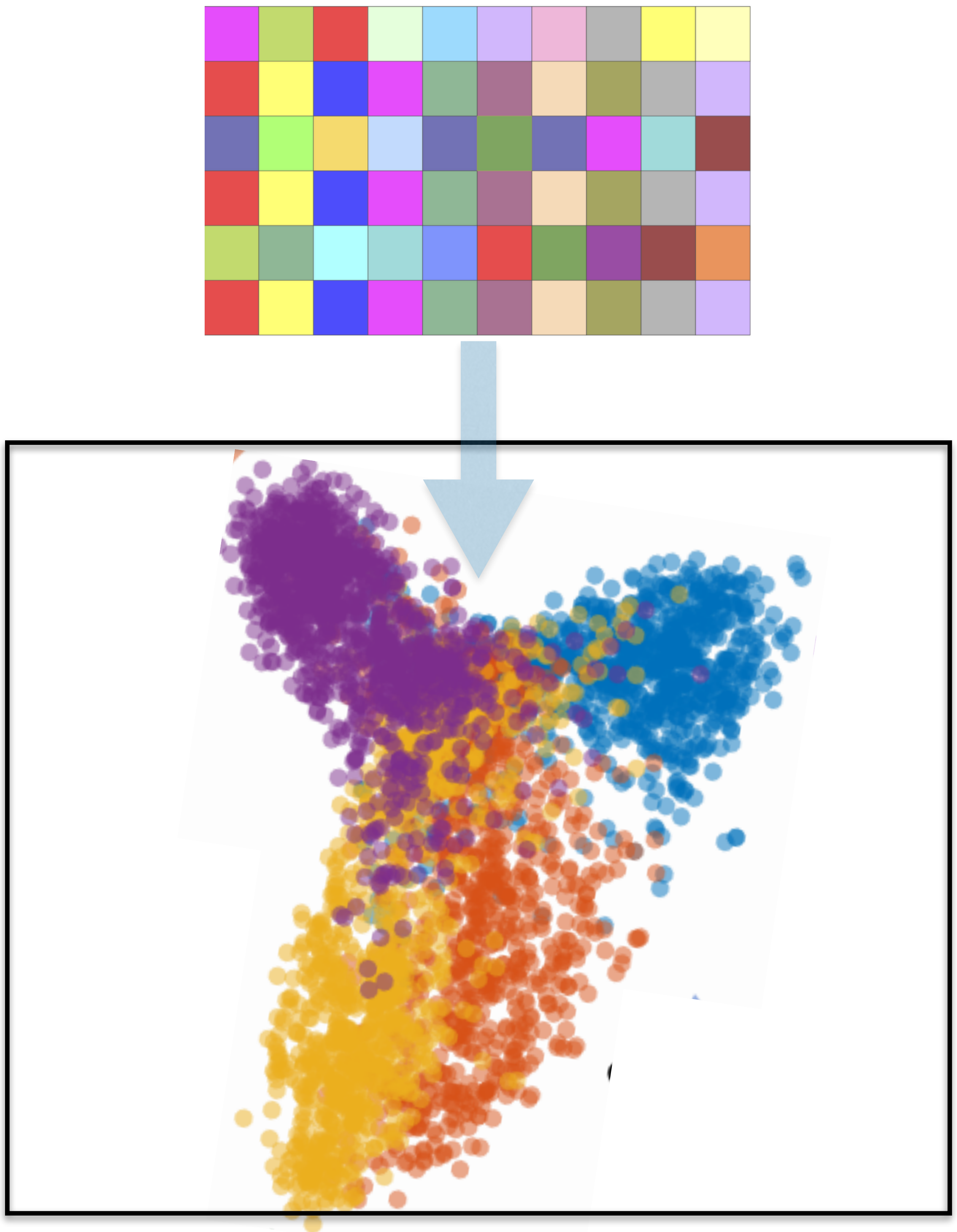




good low-d representation

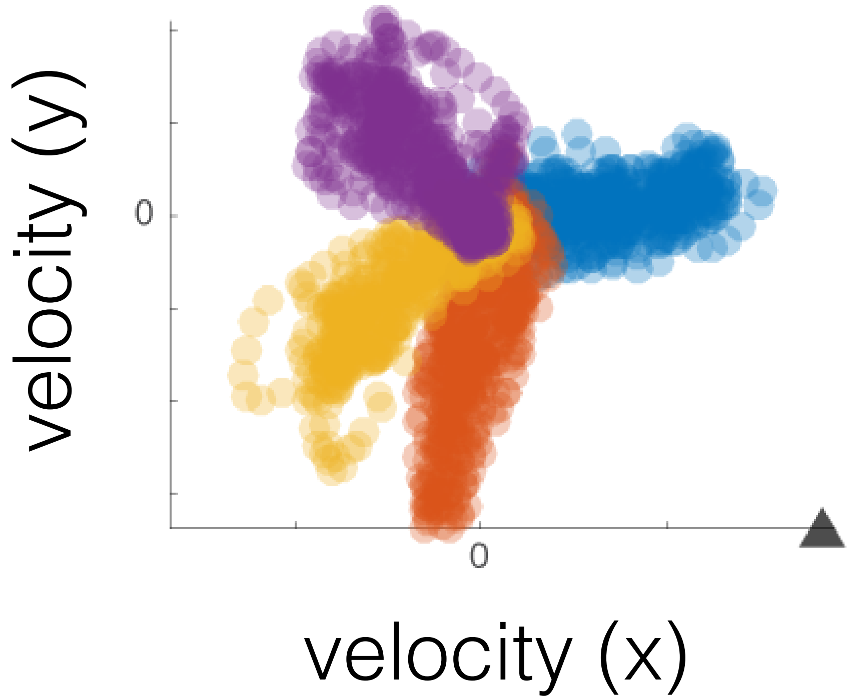
neural responses are low-d

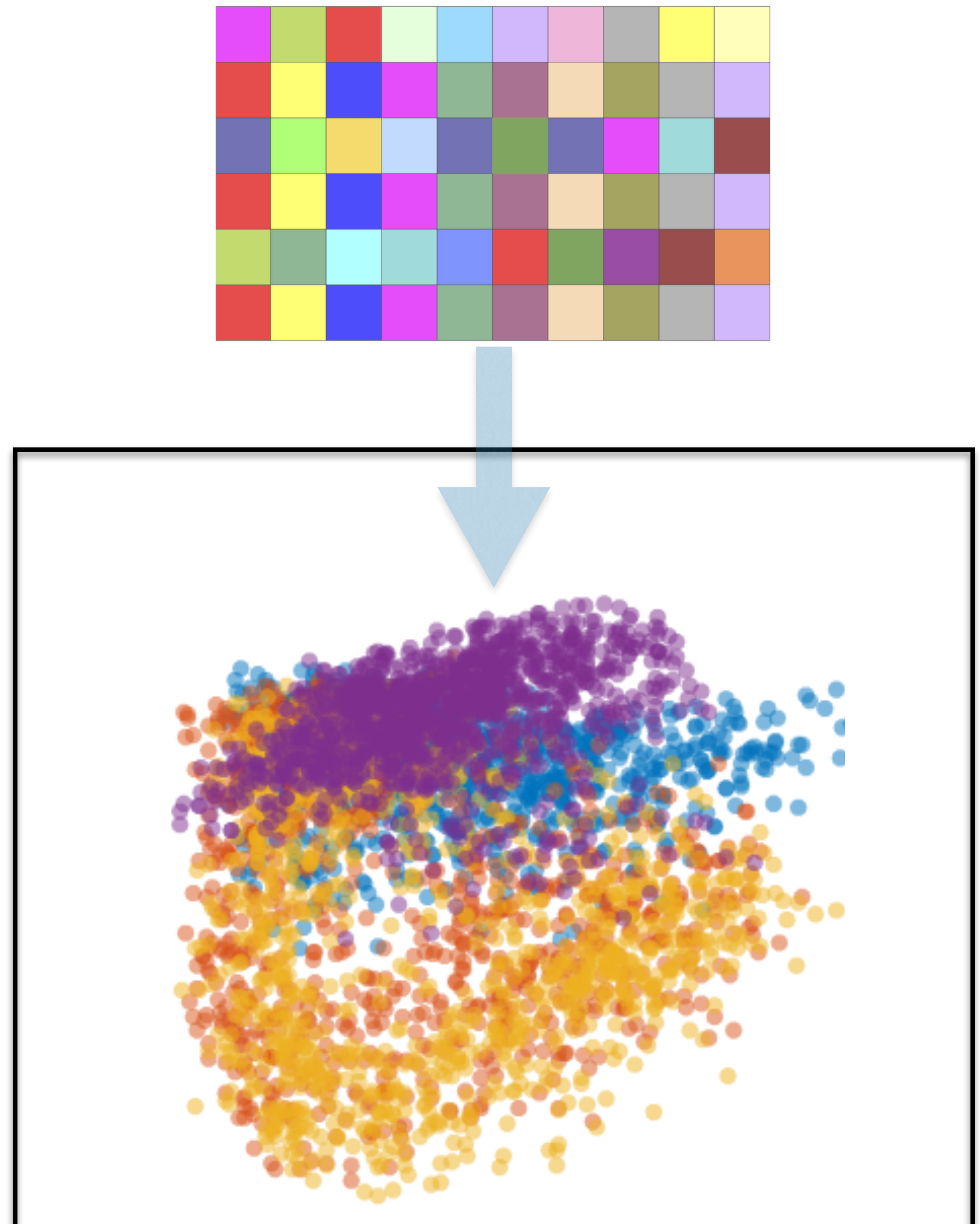




good low-d representation

neural responses are low-d

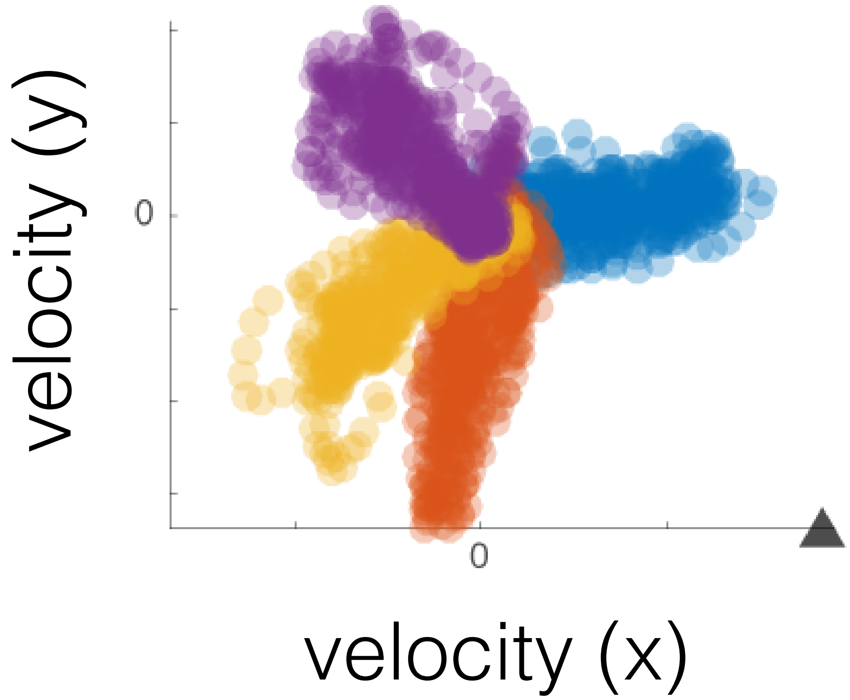


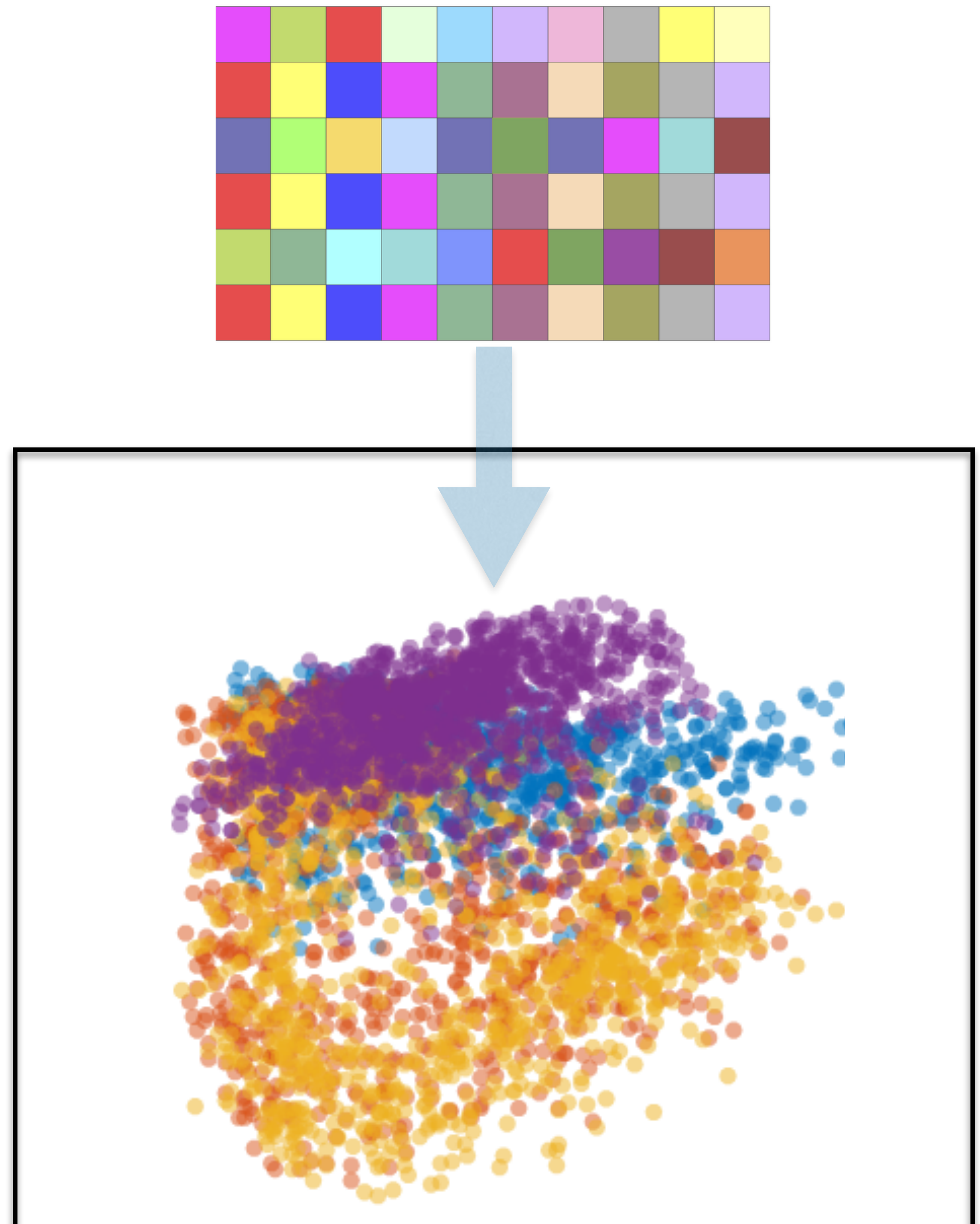


challenge: signal and noise structure vary!

bad low-d representation

neural responses are low-d

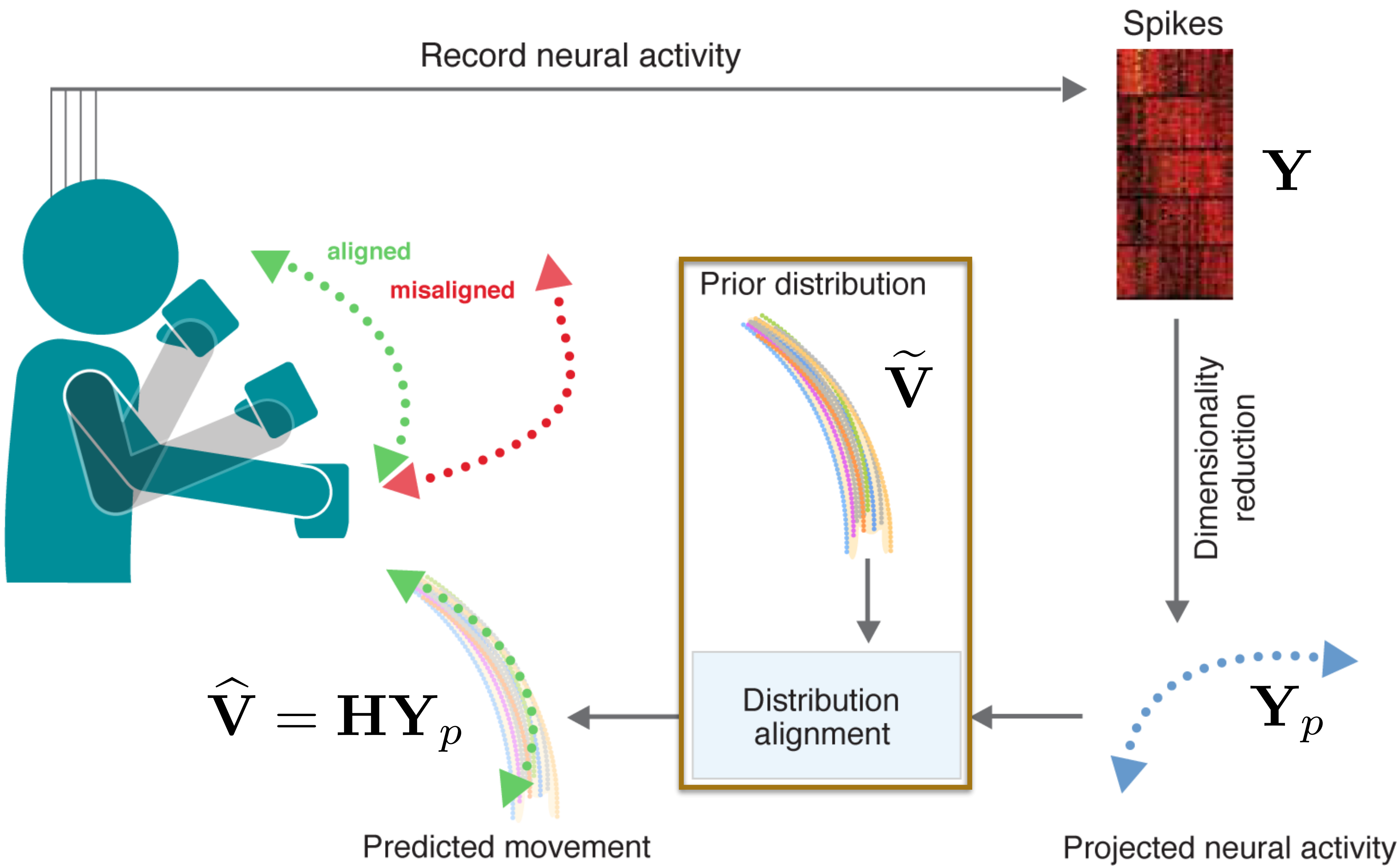




bad low-d representation

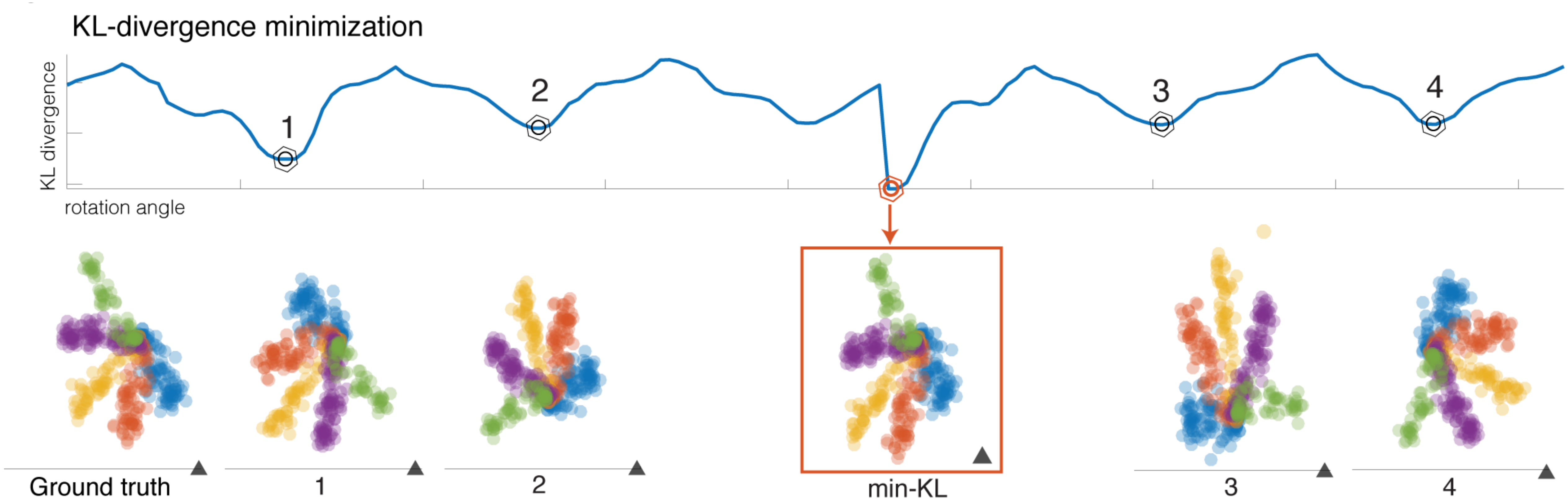
solution: leverage distribution of known movements

distribution alignment approach



Dyer, Nature BME, 2017 github.com/nerdslab/DAD

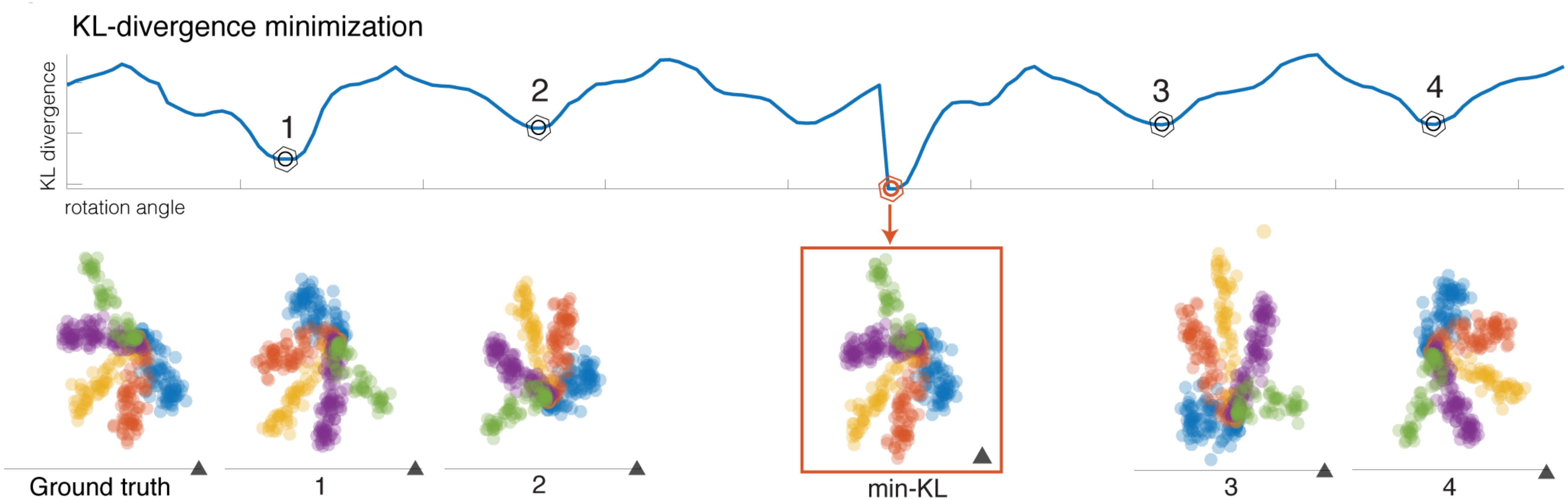
KL-divergence minimization



goal: align neural activities with prior movement distribution

Dyer, Nature BME, 2017

KL-divergence minimization



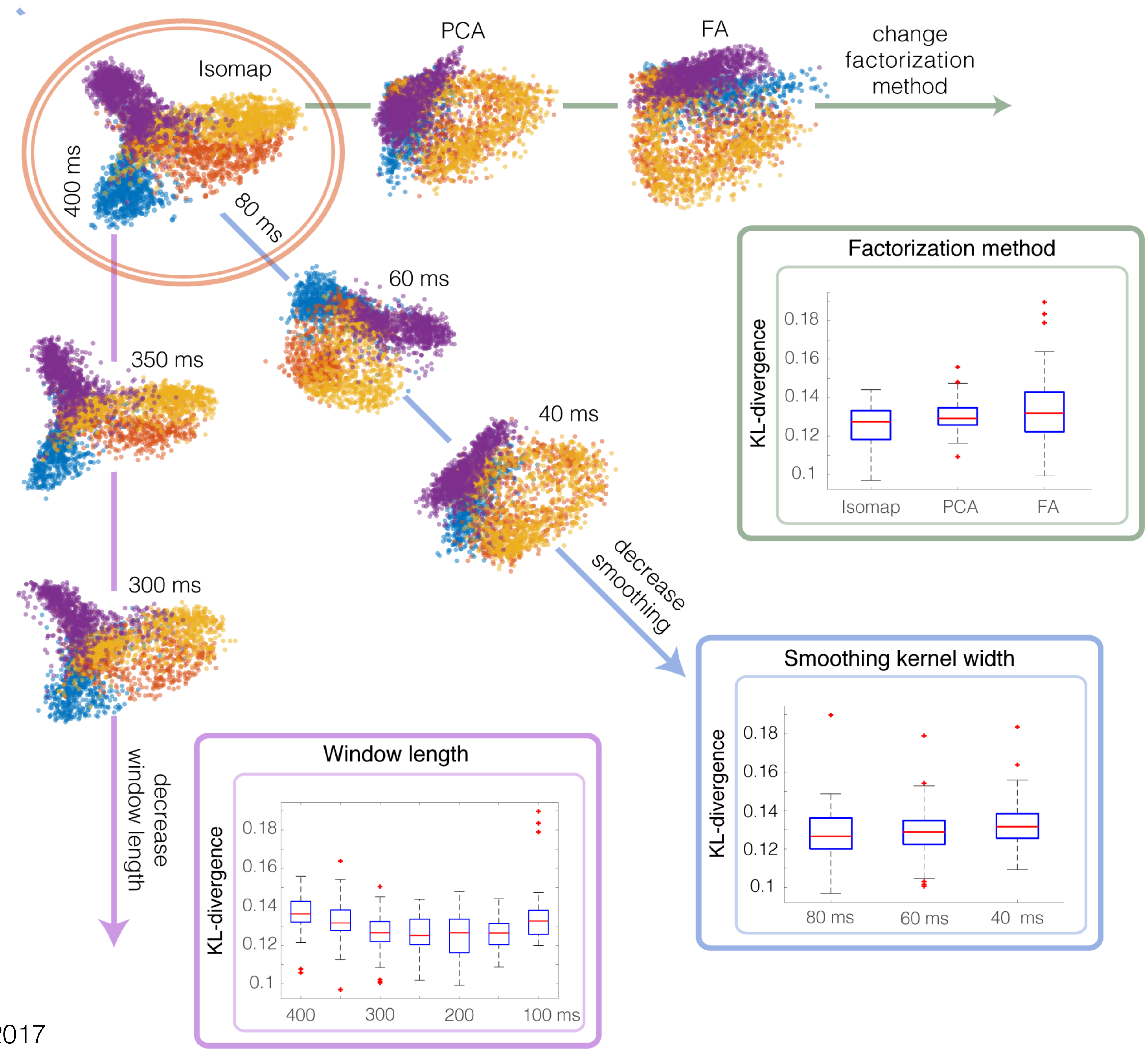
$$\mathbf{H}^* = \underset{\mathbf{H} \in \mathbb{R}^{d \times 3}}{\arg\min} \, \mathbf{KL}(p||q)$$

Estimate **p** from $\widetilde{\mathbf{V}}$

Estimate **q** from $\widehat{\mathbf{V}} = \mathbf{HY}_p$

Dyer, Nature BME, 2017

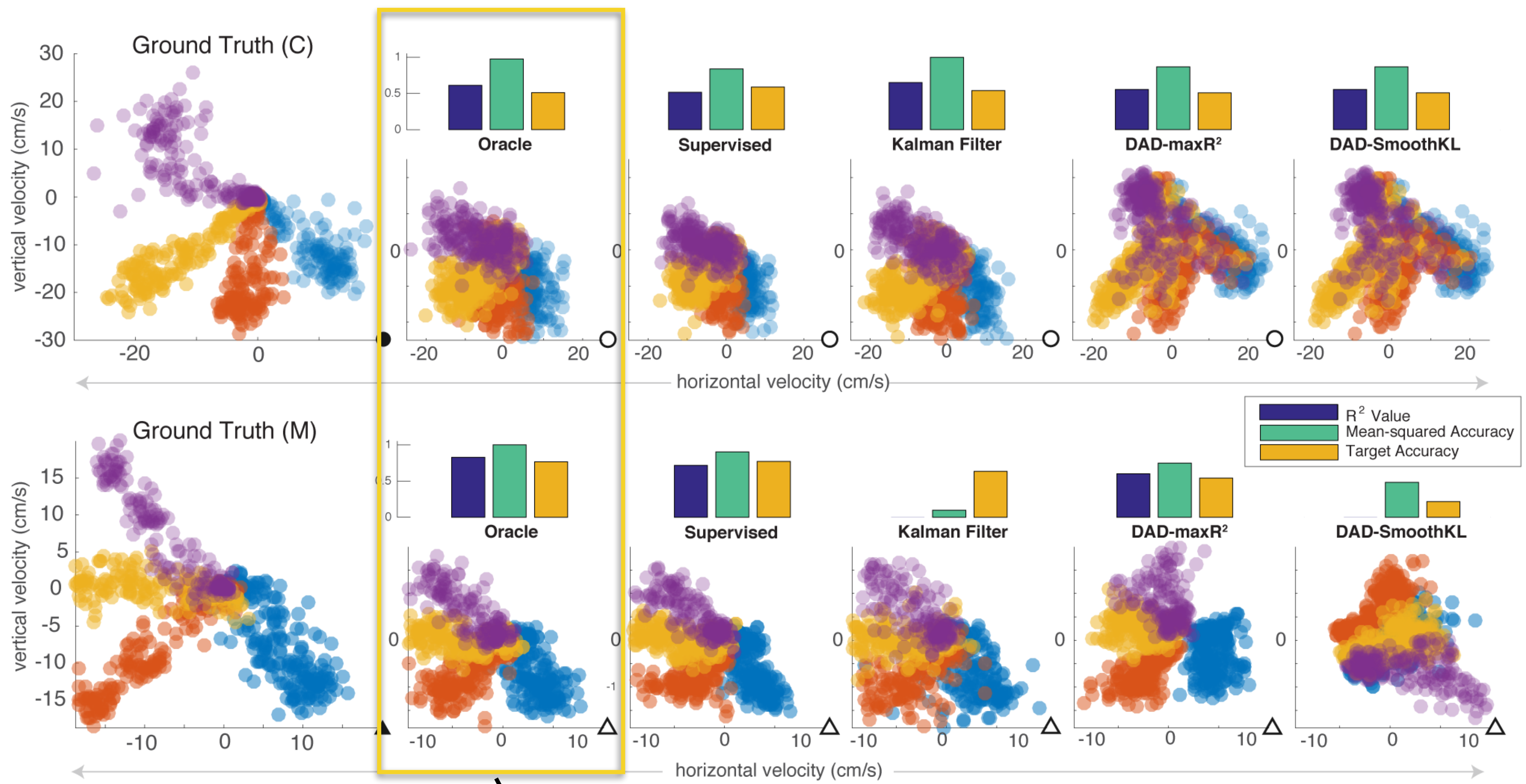
model selection



Dyer, Nature BME, 2017

decoding results

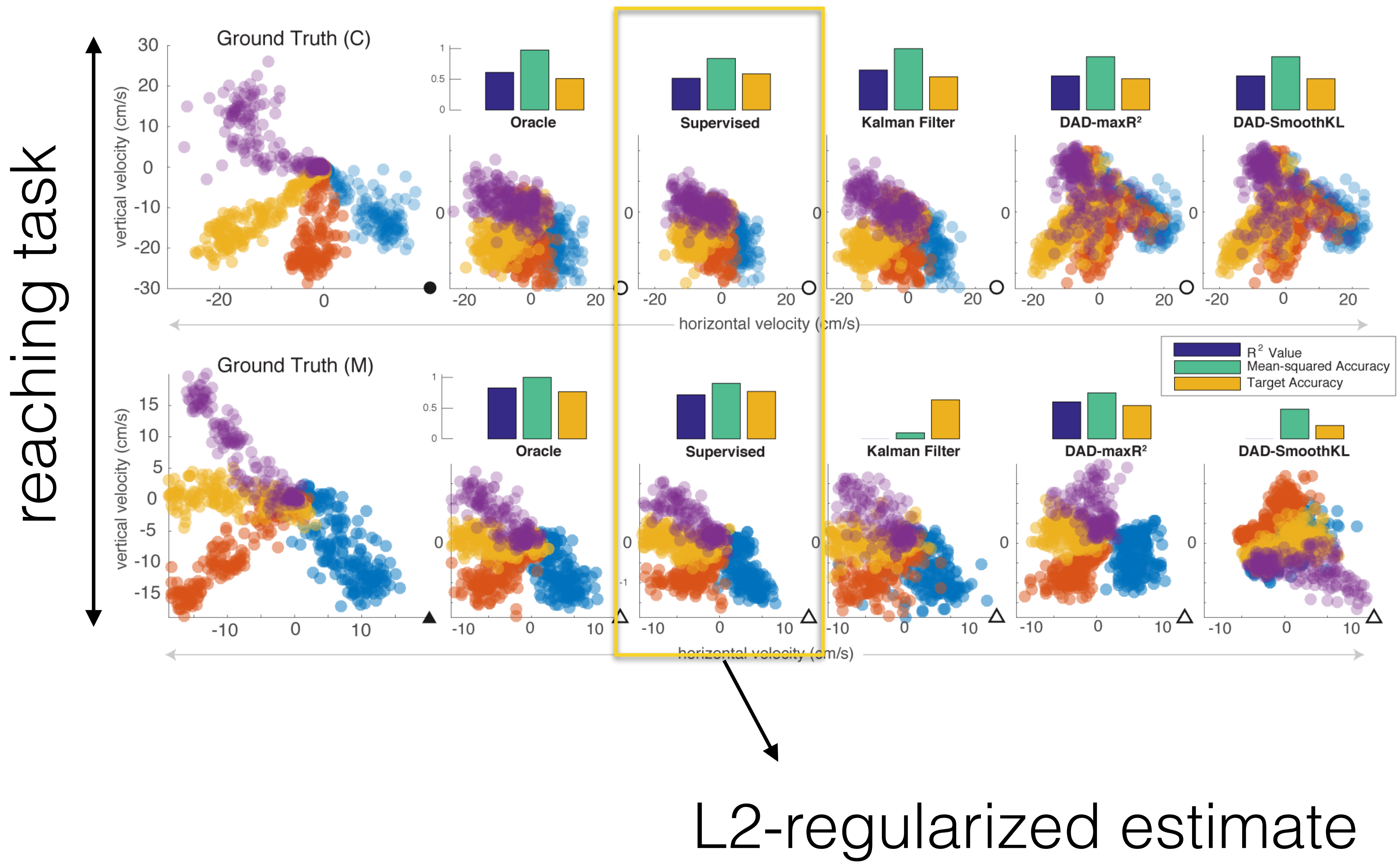
reaching task



optimal linear map on test set

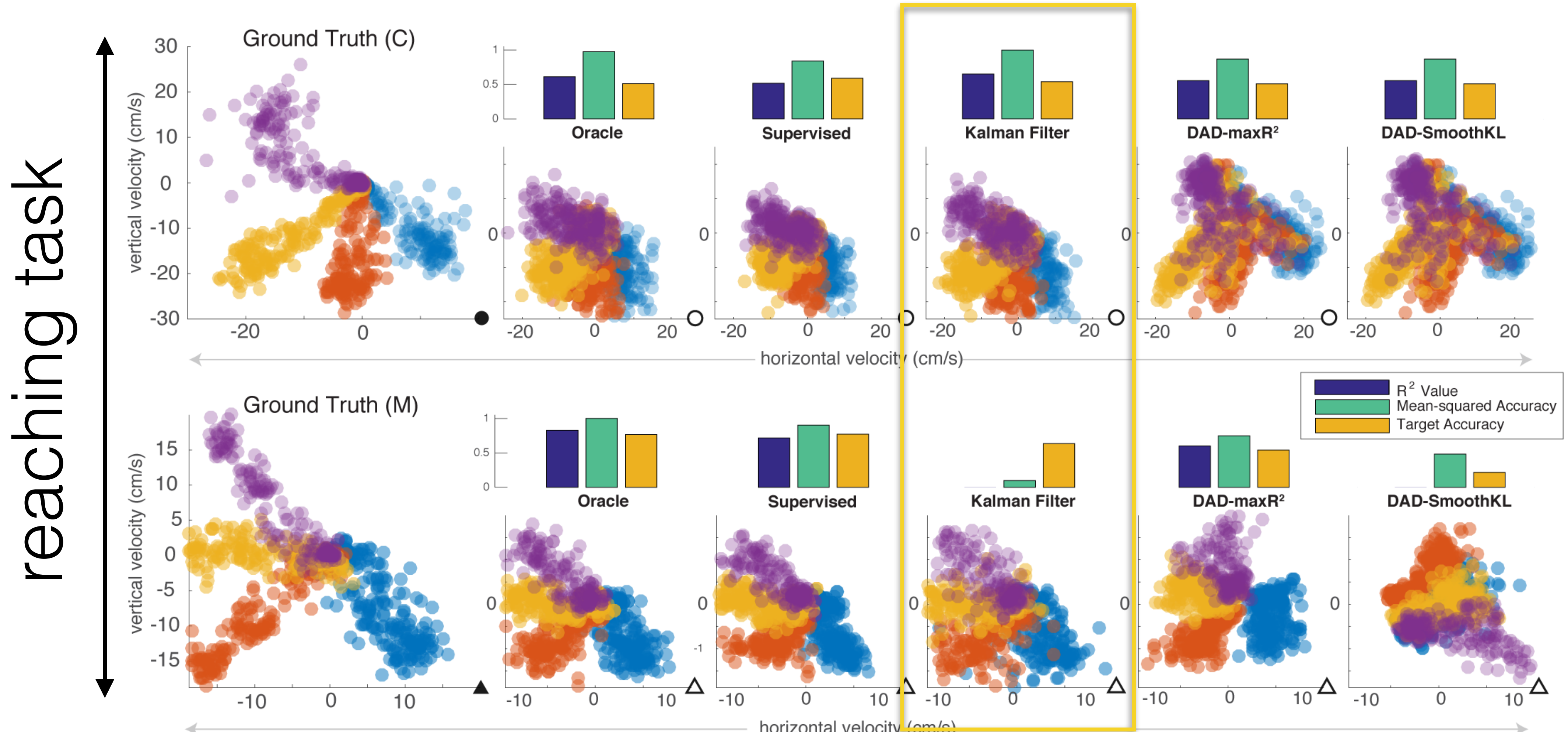
Dyer, Nature BME, 2017

decoding results



Dyer, Nature BME, 2017

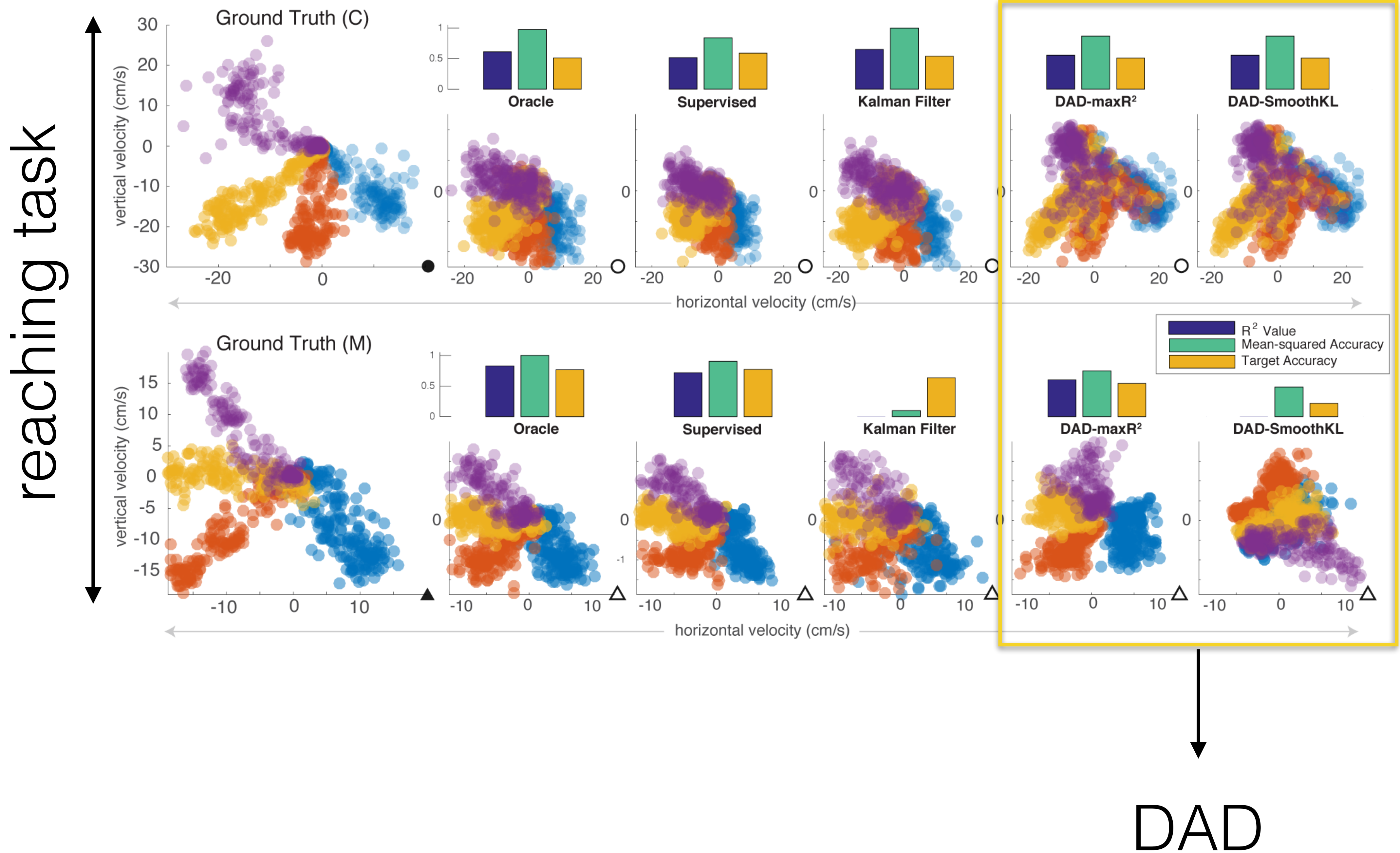
decoding results



supervised method which leverages dynamics

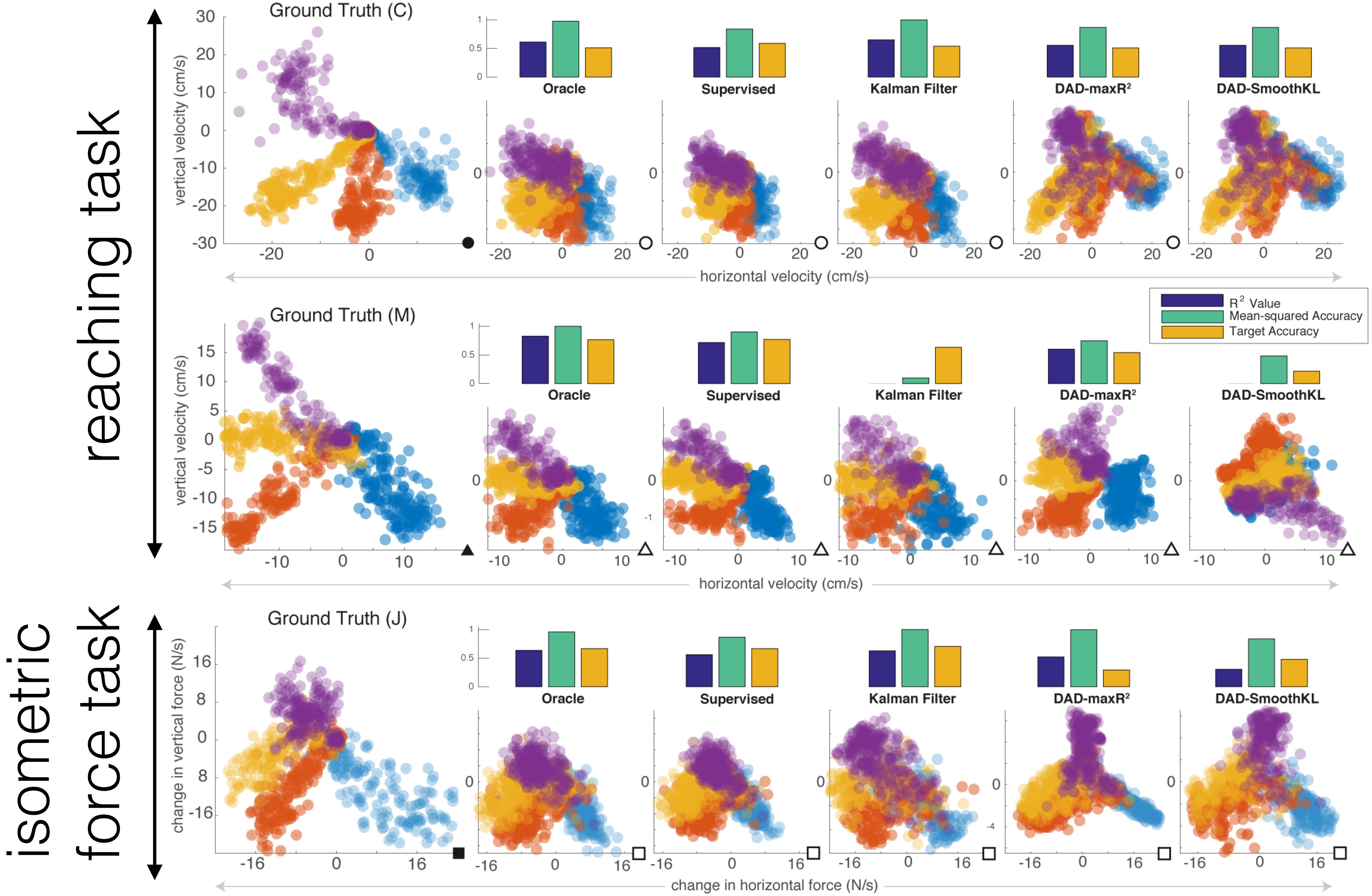
Dyer, Nature BME, 2017

decoding results



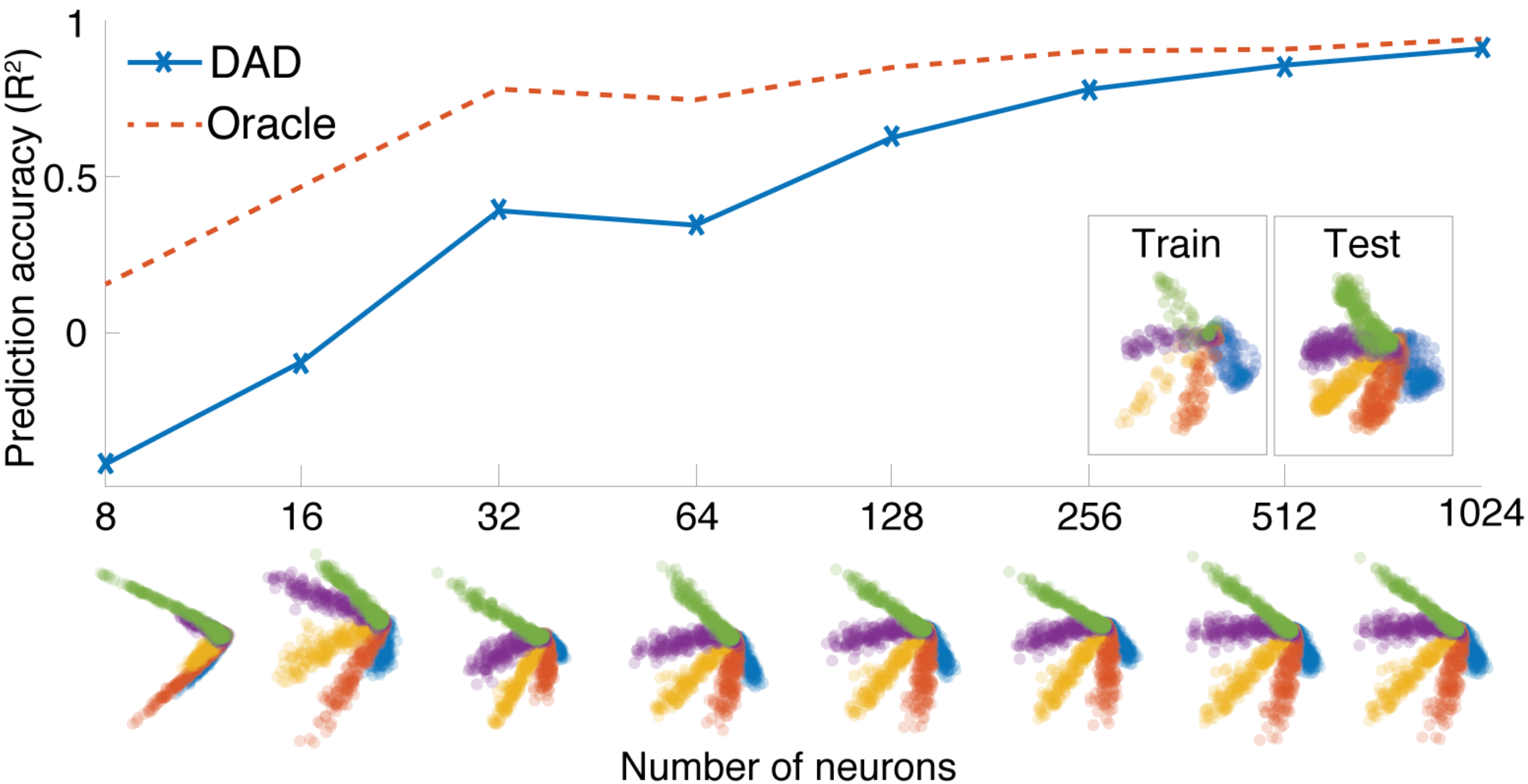
Dyer, Nature BME, 2017

decoding results



Dyer, Nature BME, 2017

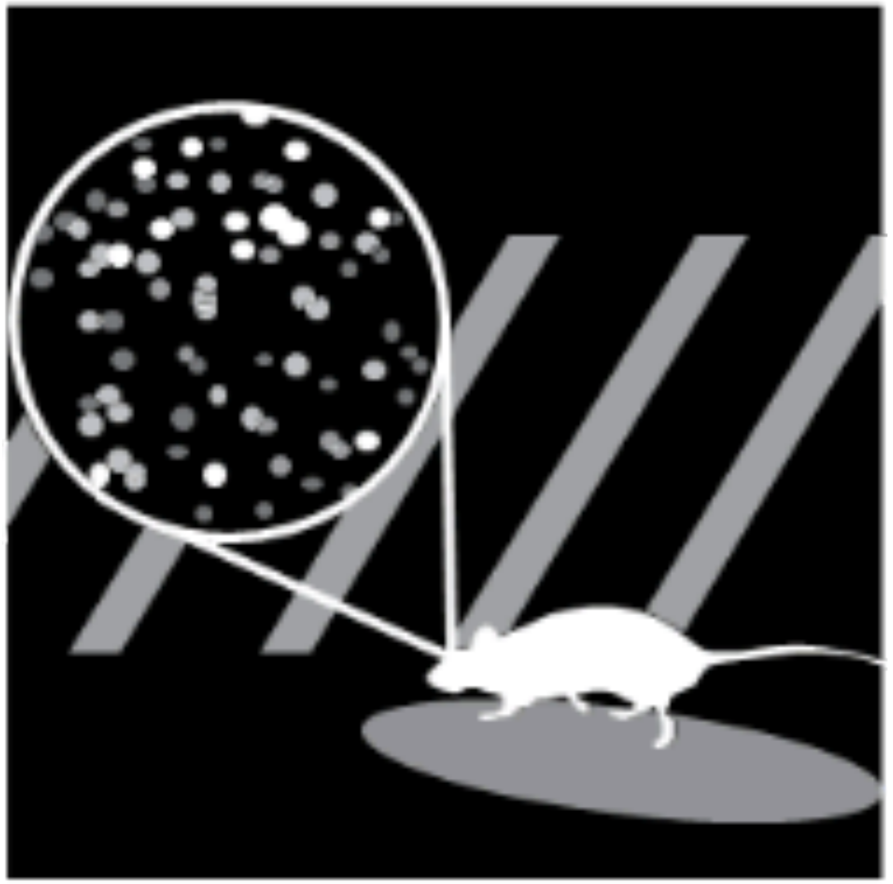
increasing the population size



application to
visual coding

visual coding

Allen Institute Brain Observatory



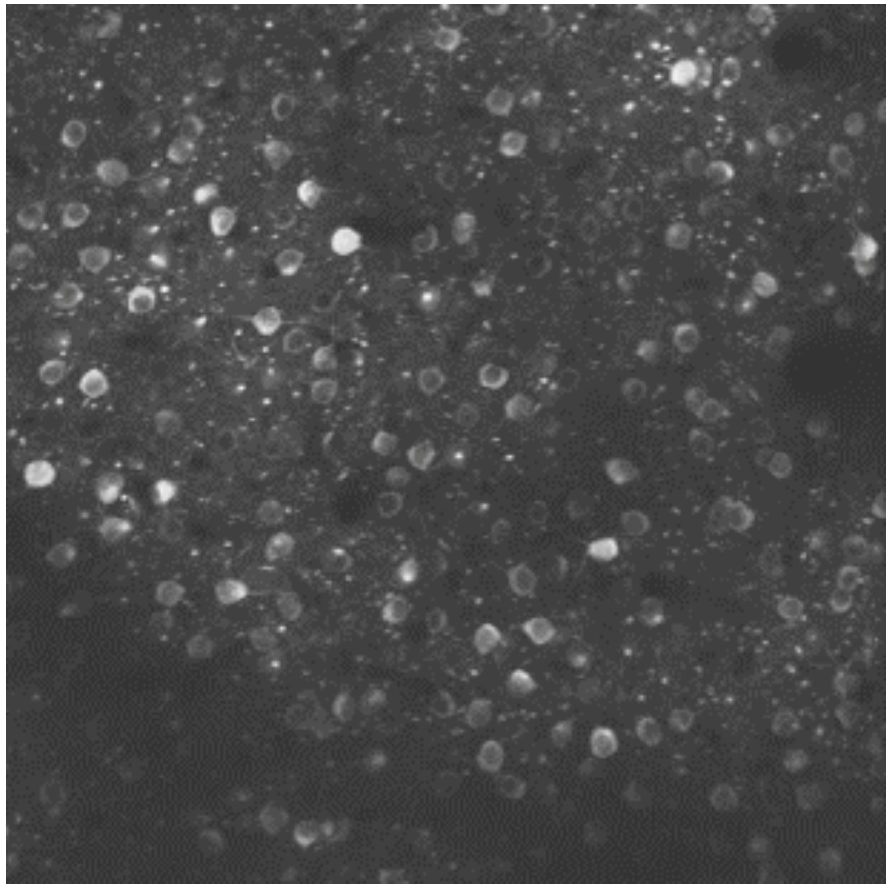
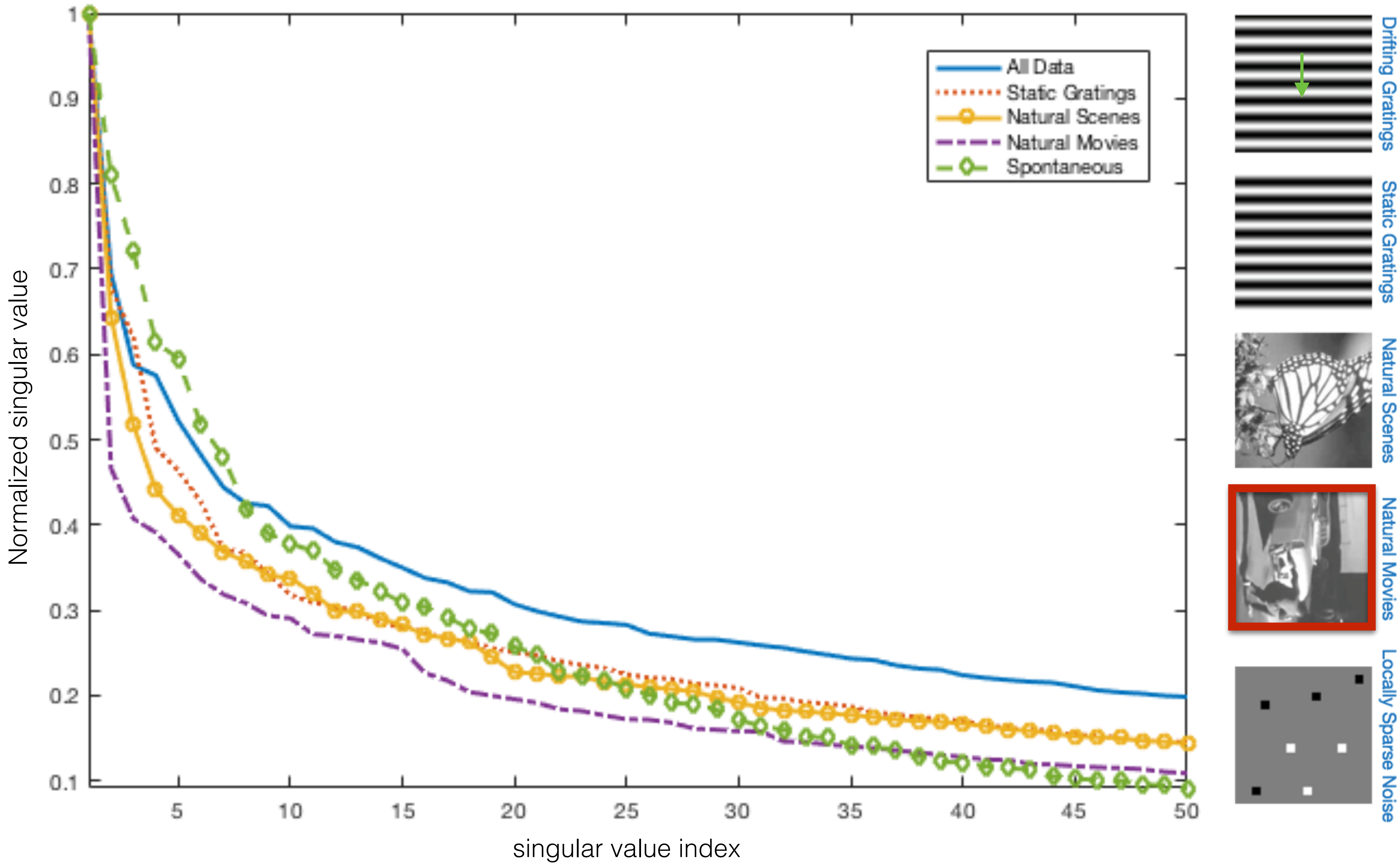


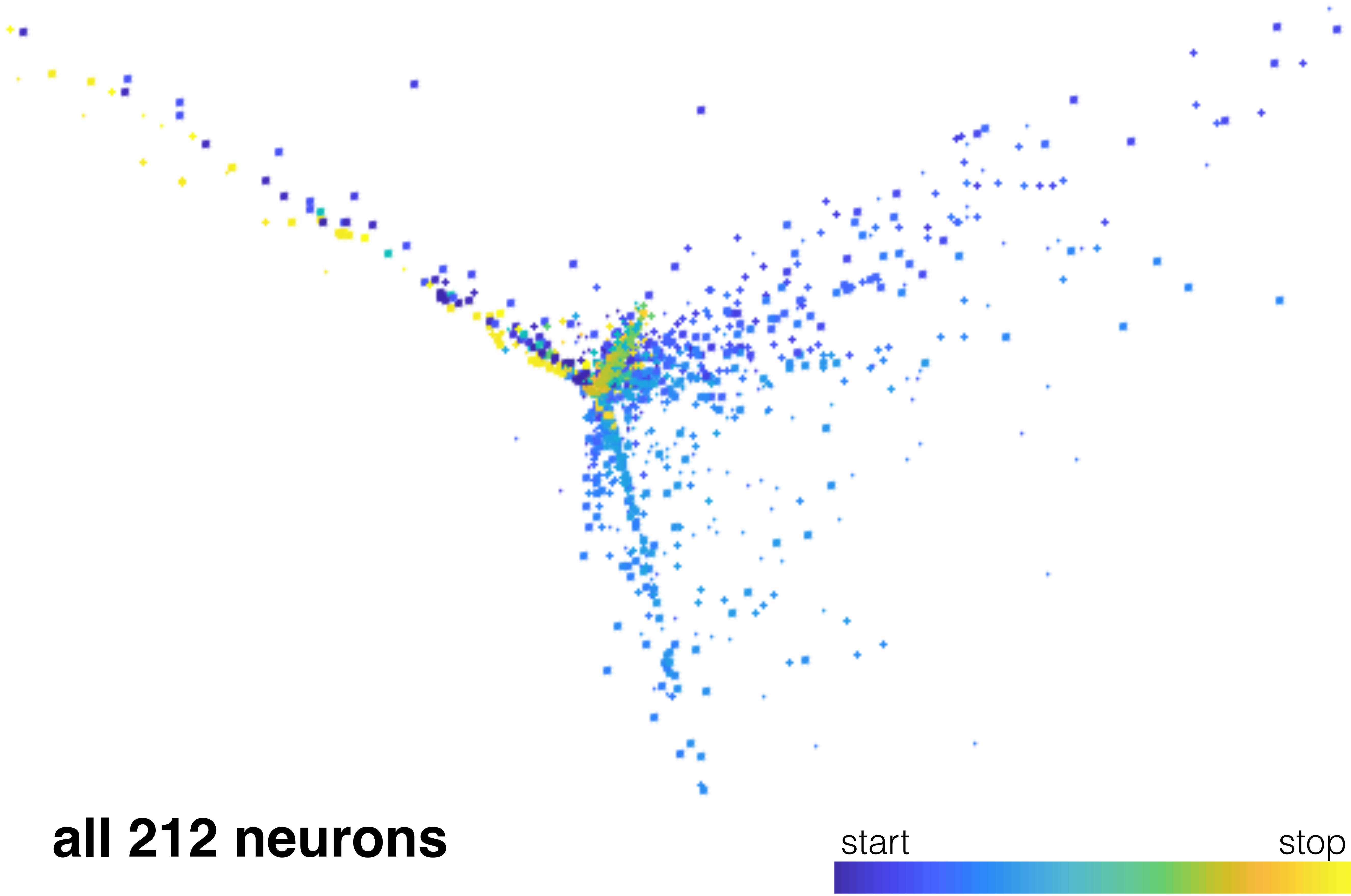
Photo Credit: Allen Institute Brain Observatory

De Vries, bioRxiv, 2018

naturalistic stimuli are low-d

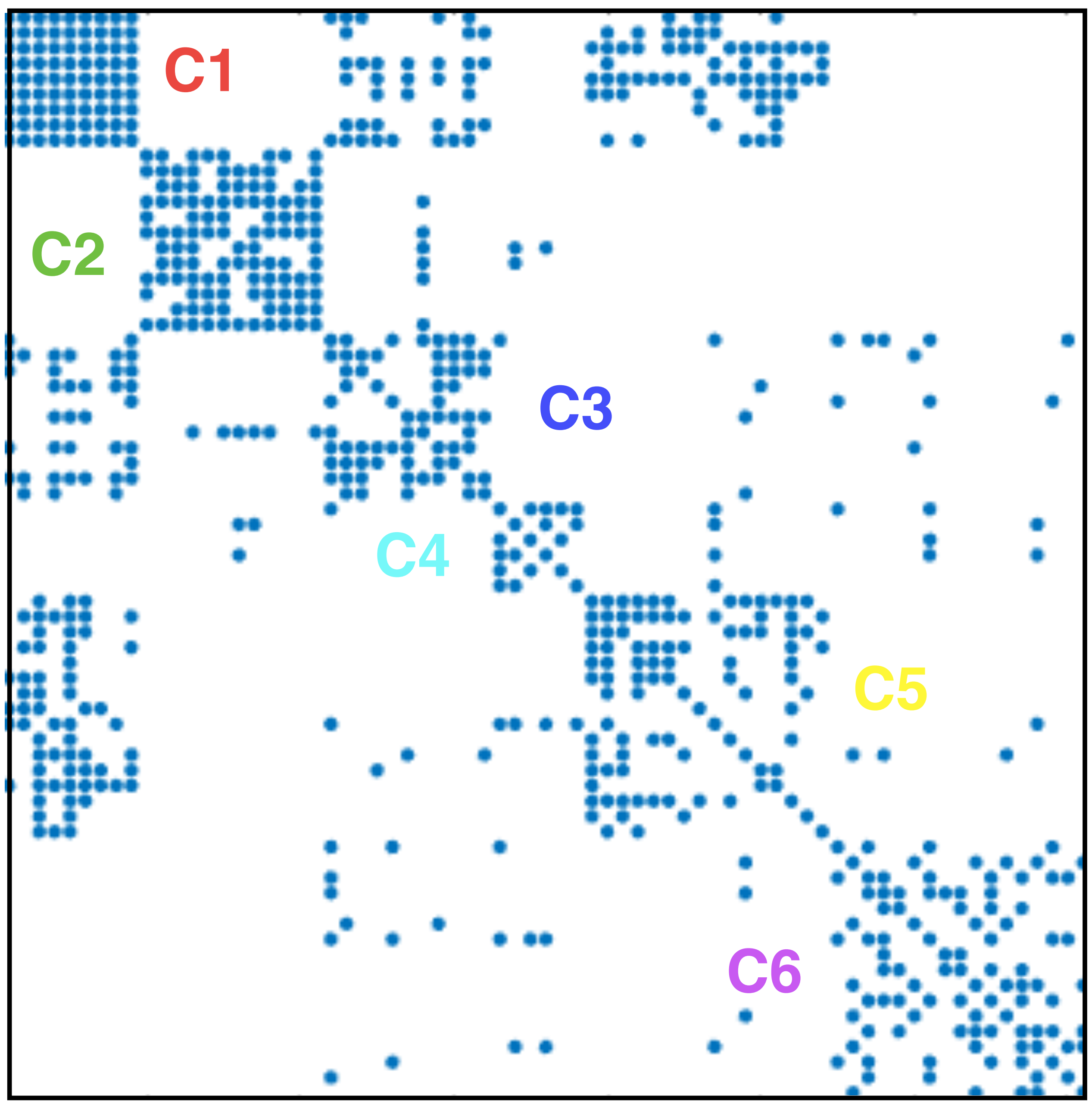


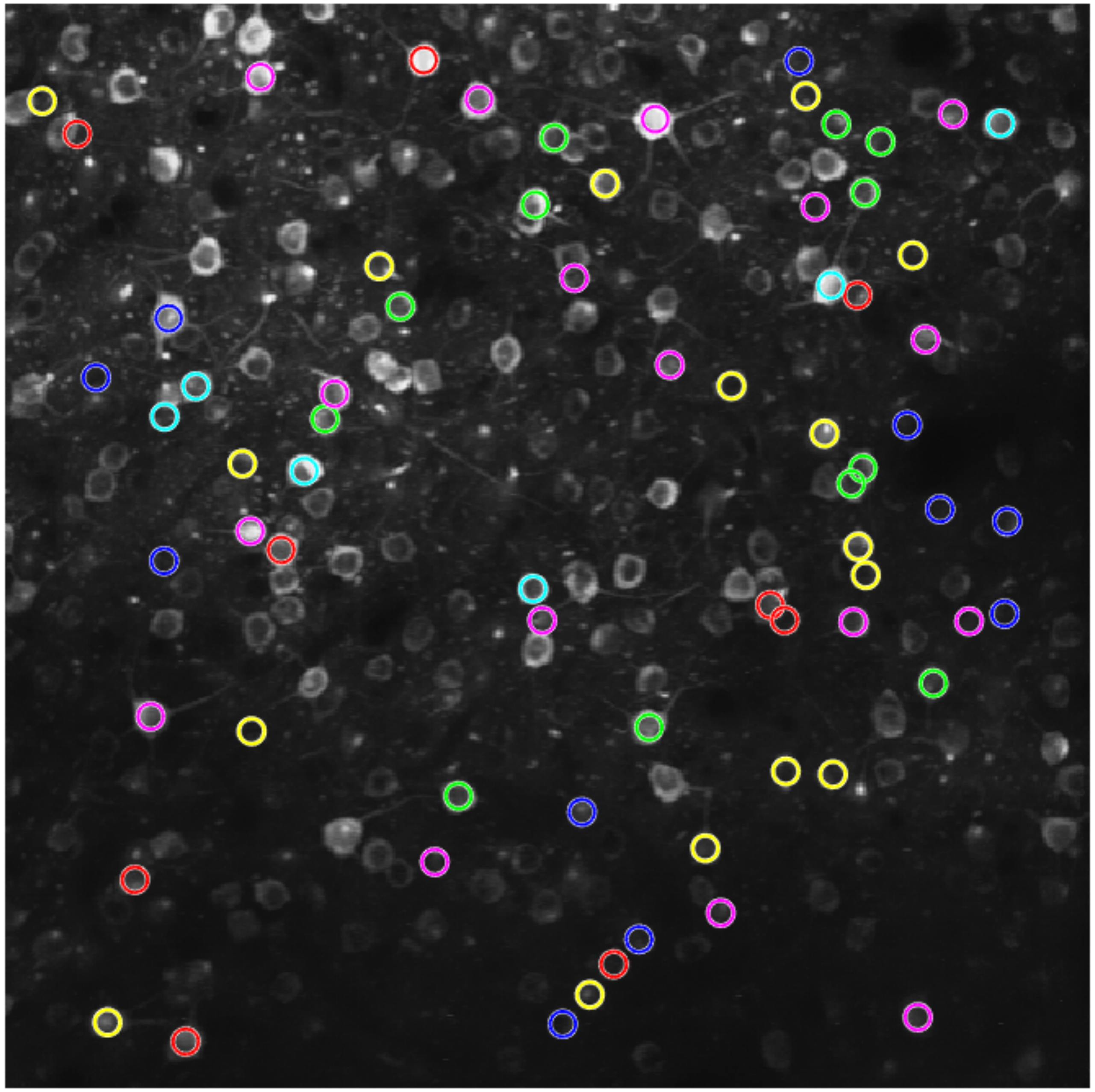
visualizing population dynamics



all 212 neurons

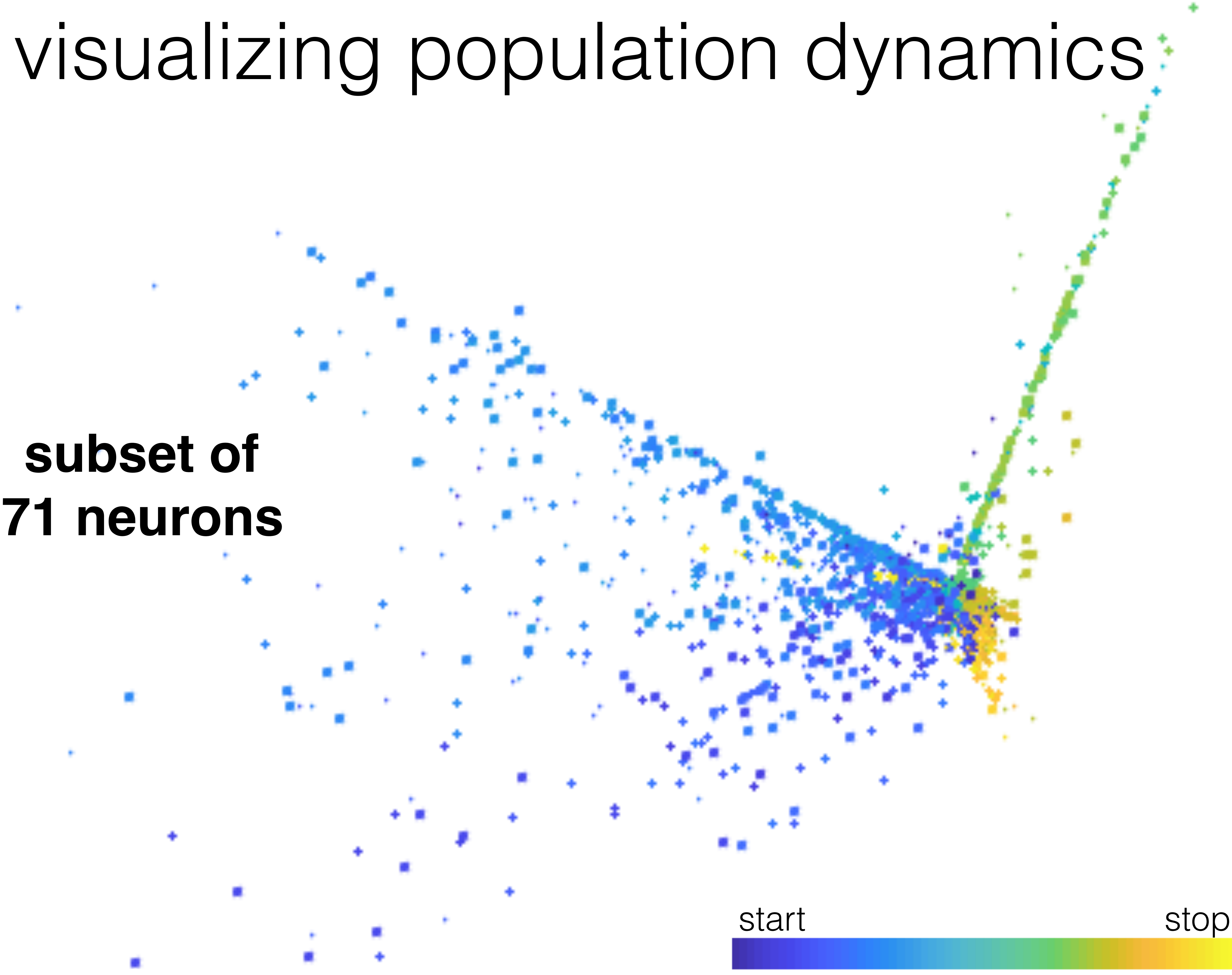
cluster analysis - natural movies

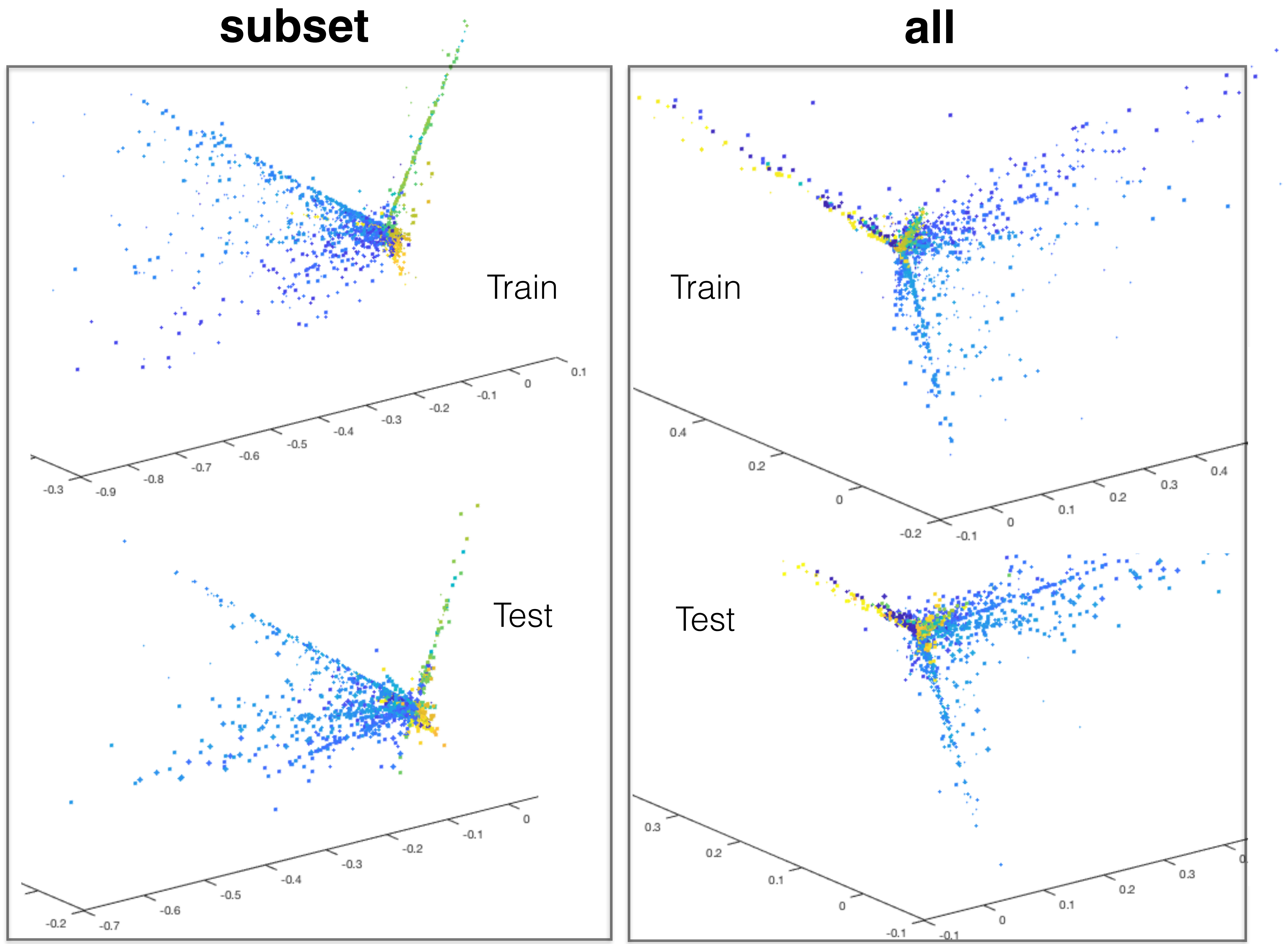




visualizing population dynamics

subset of 71 neurons




subset
all
Train
Test
Train
Test
-0.3
-0.9
-0.8
-0.7
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
-0.2
-0.7
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.4
0.2
0
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.3
0.2
0.1
0
-0.1
-0.1
0
0.1
0.2
0.3

summary

overview of low-dimensional models

- Linear subspace models (PCA, FA, NMF)
- Manifold models (Isomap, LLE)
- Clustering models (kmeans)
- Unions of subspaces (SSC)

example 1: movement decoding

- use movement priors to guide factorizations
- decode without supervised data

example 2: visual coding

- cluster neurons then factorize
- not all neurons are created equal

collaborators

<u>movement decoding</u>
Mohammad Gheshlagi Azar (DeepMind)
Konrad Kording (UPenn)
Lee Miller (Northwestern)

<u>visual coding</u>
Saskia de Vries (Allen Institute)

code/data refs

Matlab Toolbox for dimensionality reduction

- https://lvdmaaten.github.io/drtoolbox/

Python Tutorials on PCA

- https://sebastianraschka.com/Articles/2015_pca_in_3_steps.html

MATLAB Tutorial on Isomap

- http://www.numerical-tours.com/matlab/shapes_7_isomap/

Distribution Alignment Decoding (DAD)

- https://github.com/KordingLab/DAD/tree/master/data/demo

paper refs

Dimensionality reduction for neural data (Review)

- https://stat.columbia.edu/~cunningham/pdf/CunninghamNN2014.pdf

Distribution Alignment Decoding (DAD)

- http://rdcu.be/Bafy

Brain Observatory Pipeline Paper

- https://www.biorxiv.org/content/early/2018/06/29/359513

thank you!

(web)
dyerlab.gatech.edu