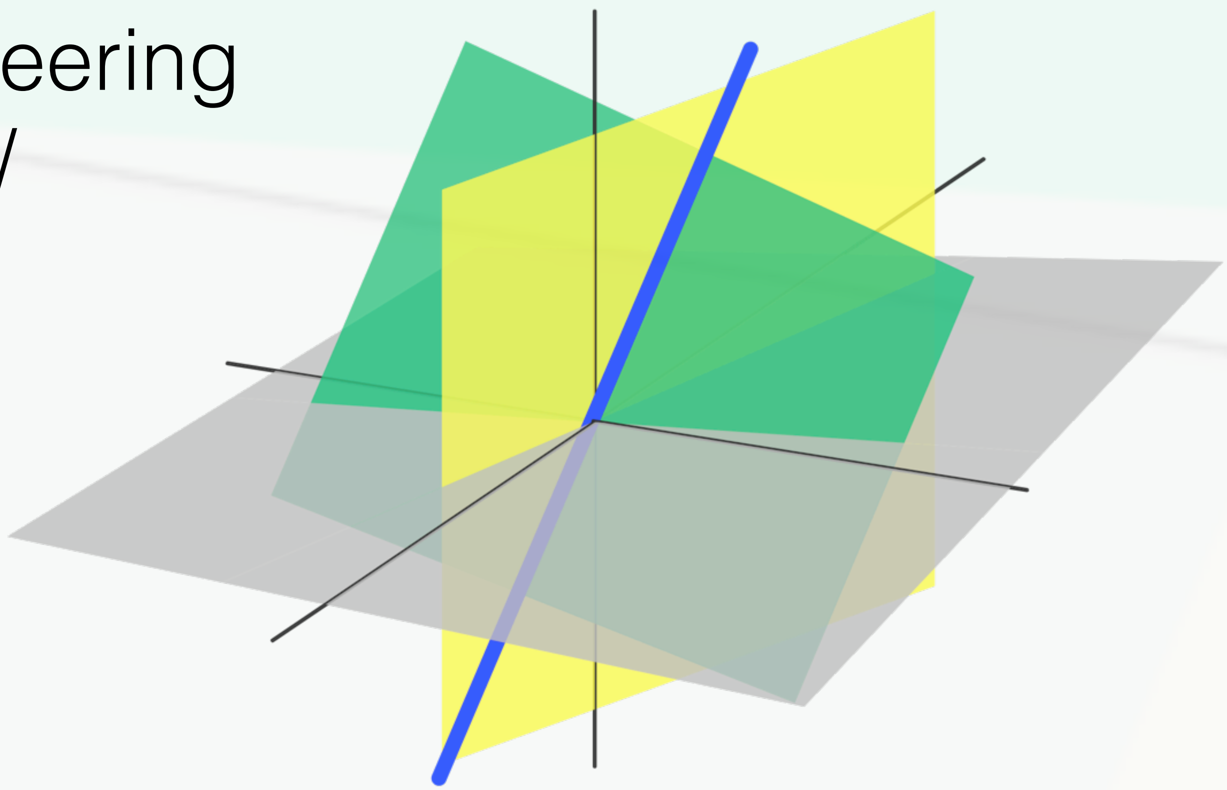
Finding **low-dimensional** structure in **high-dimensional** datasets

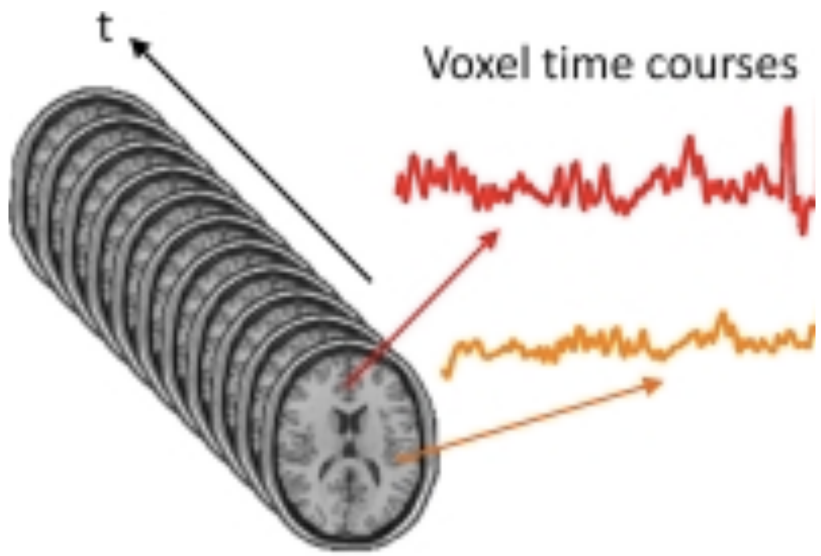
Eva Dyer

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

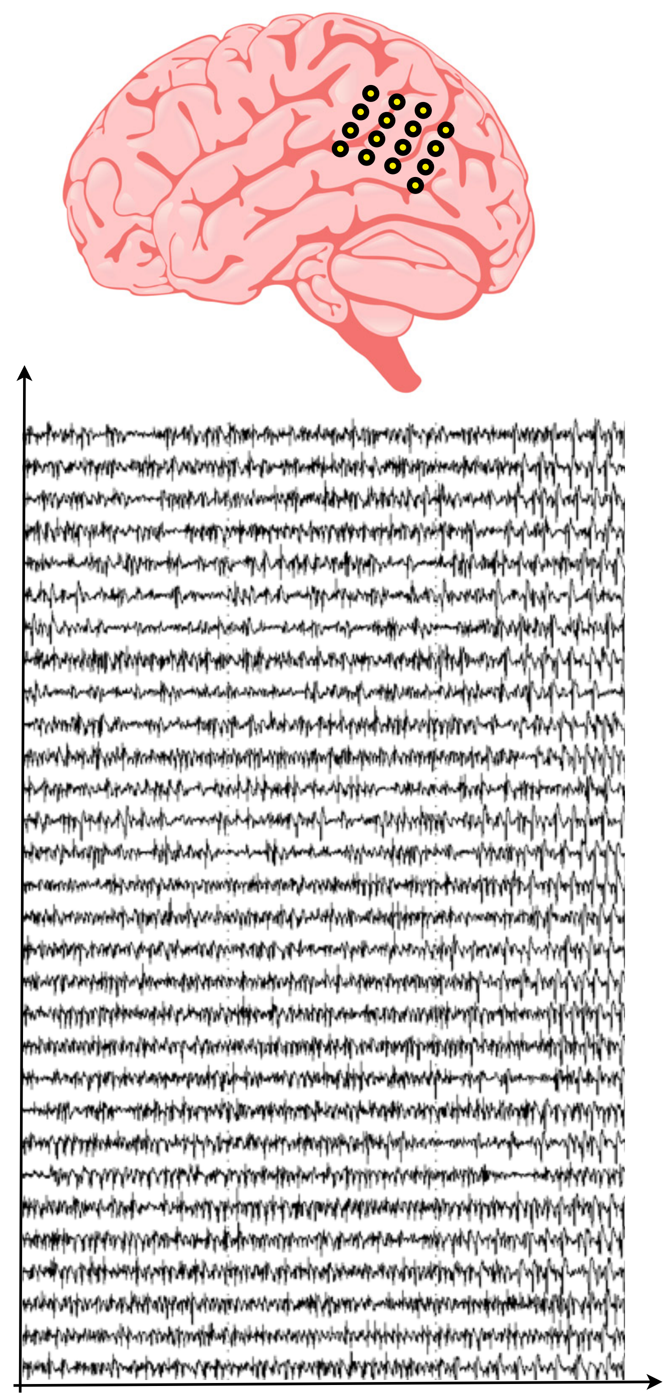


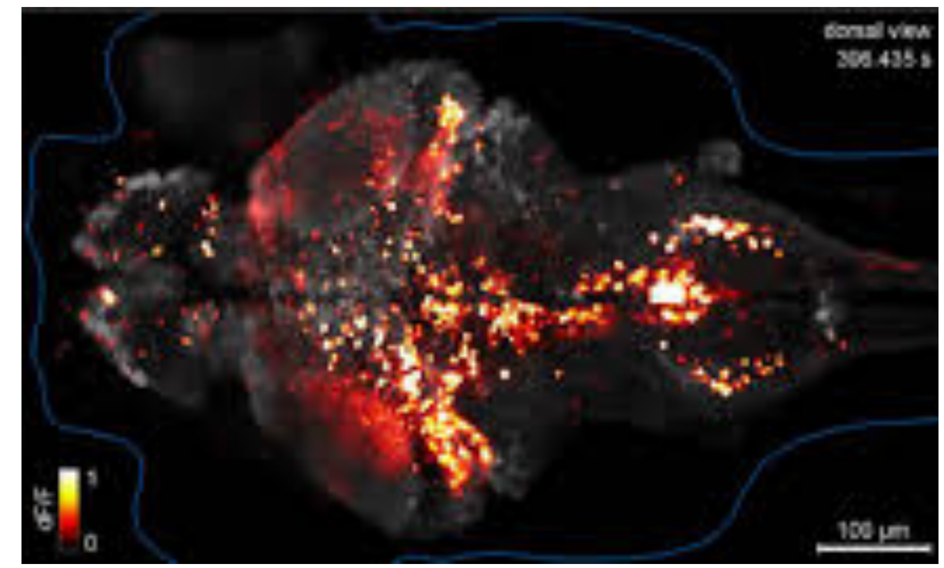
Georgia Tech
EMORY UNIVERSITY

neural data deluge

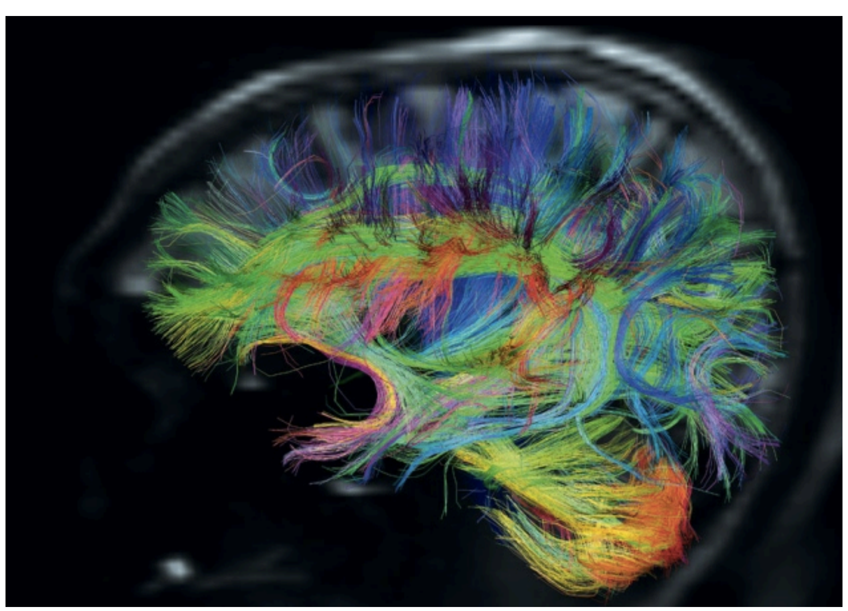


fMRI



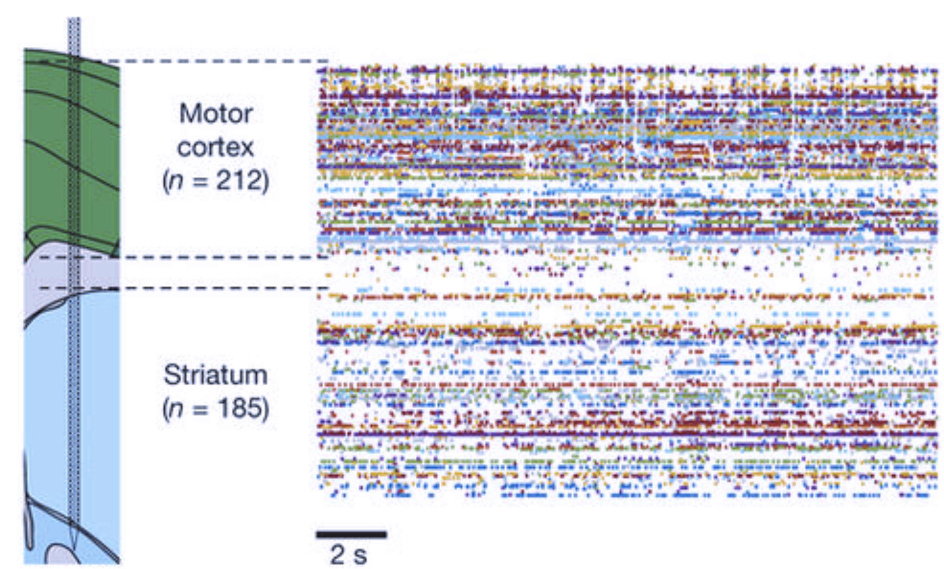


calcium imaging



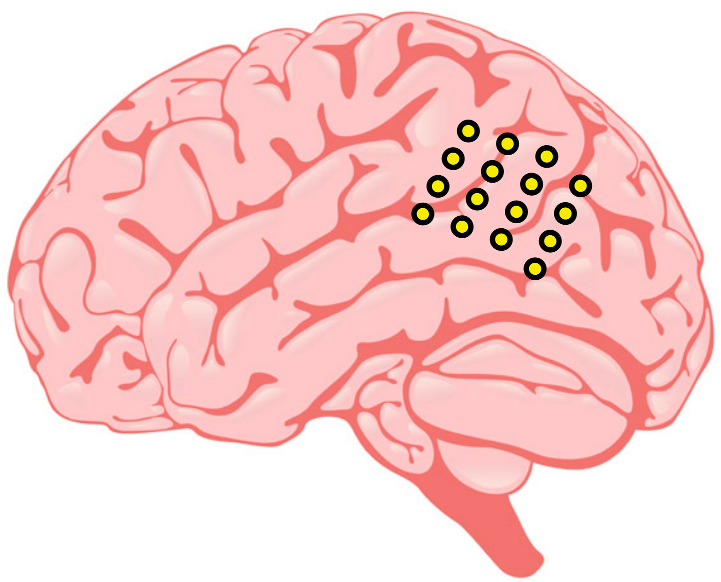
DTI

ecog



electrophysiology

why reduce dimensionality?

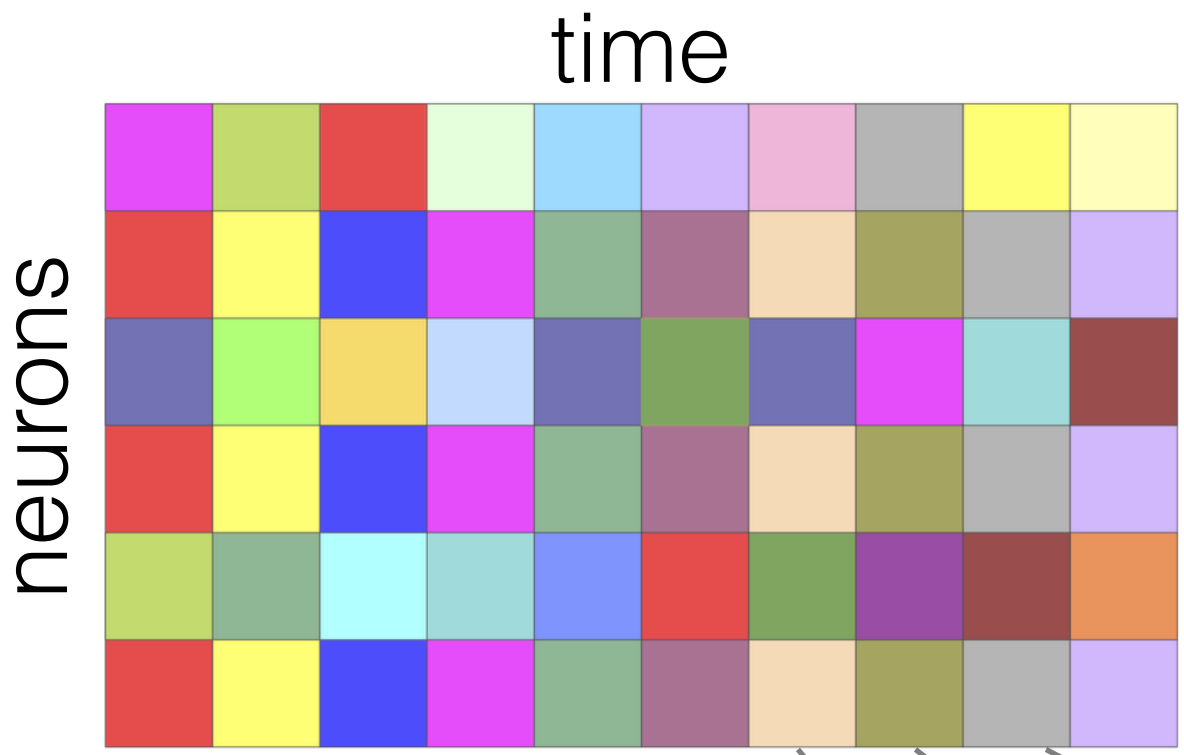


compression

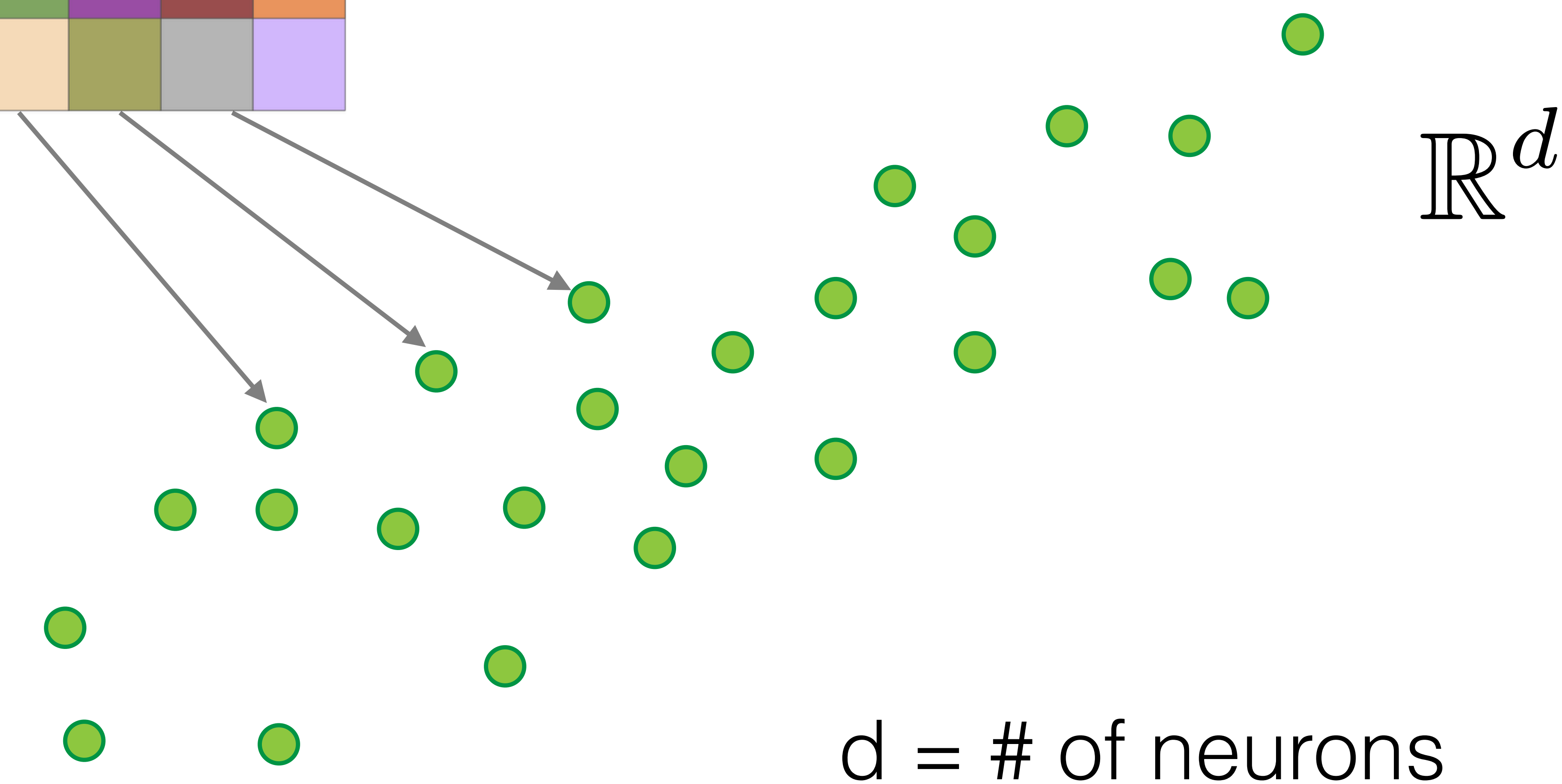
denoising

interpret complex data

low-dimensional models

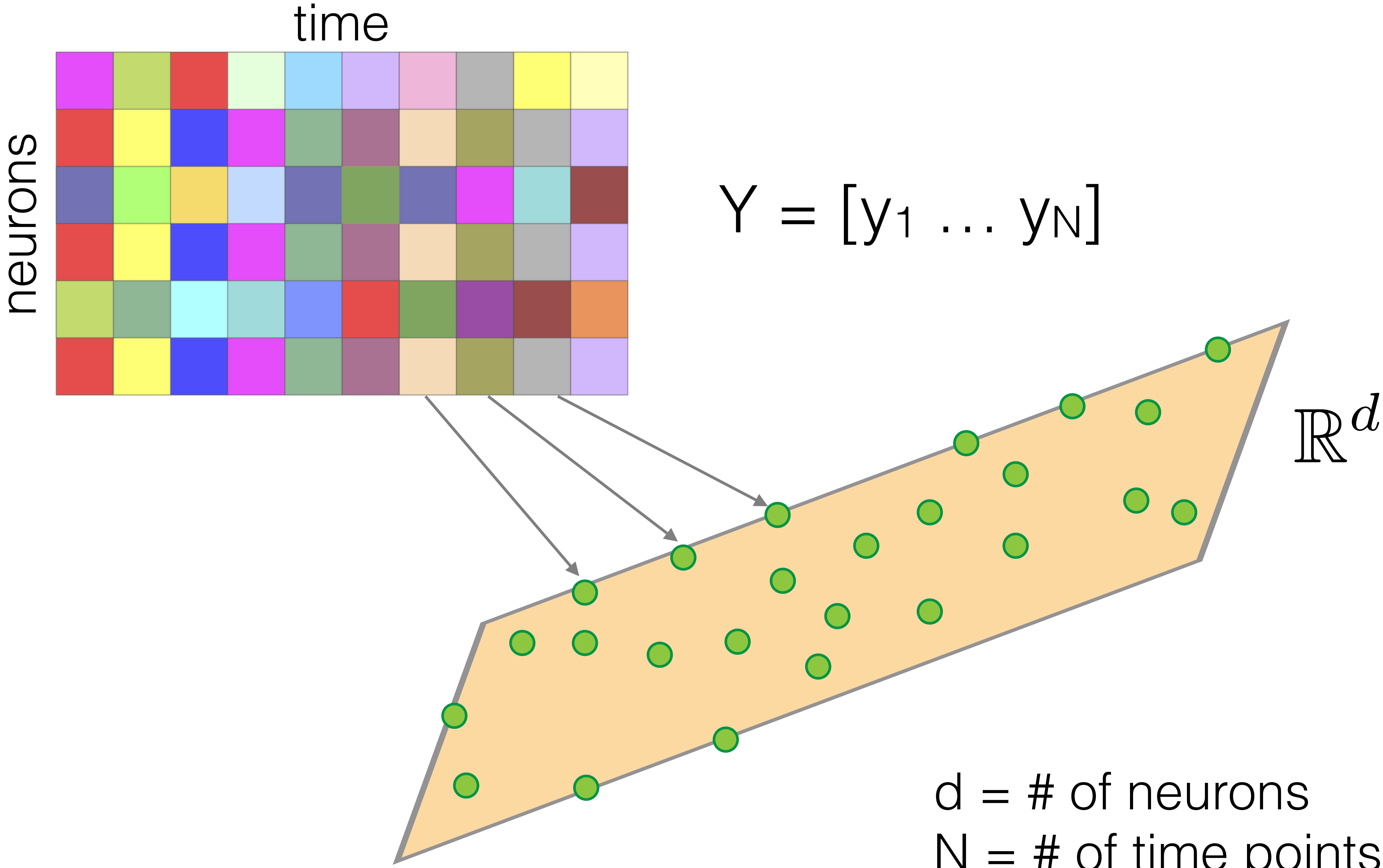


$Y = [y_1 \ldots y_N]$



d = # of neurons
N = # of time points

low-dimensional models

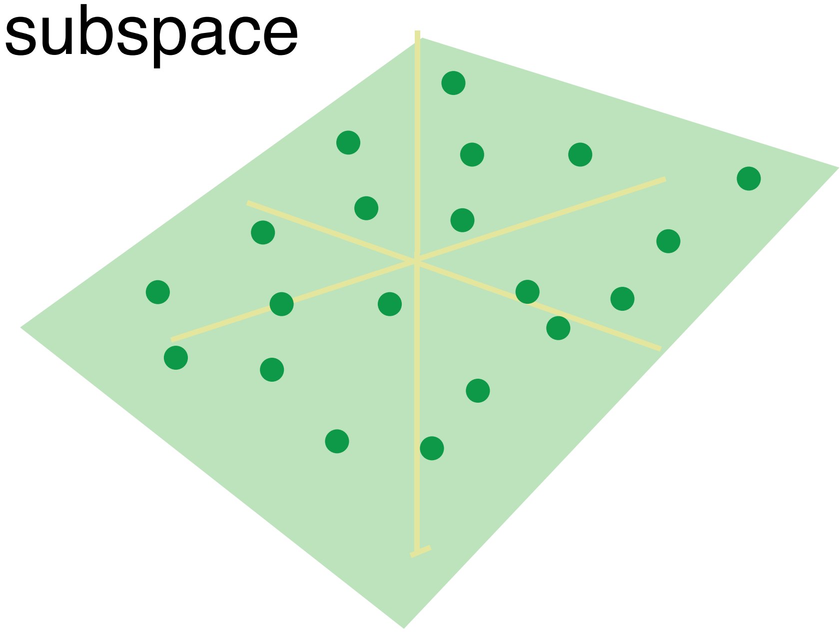


$Y = [y_1 \ldots y_N]$

d = # of neurons
N = # of time points

low rank model

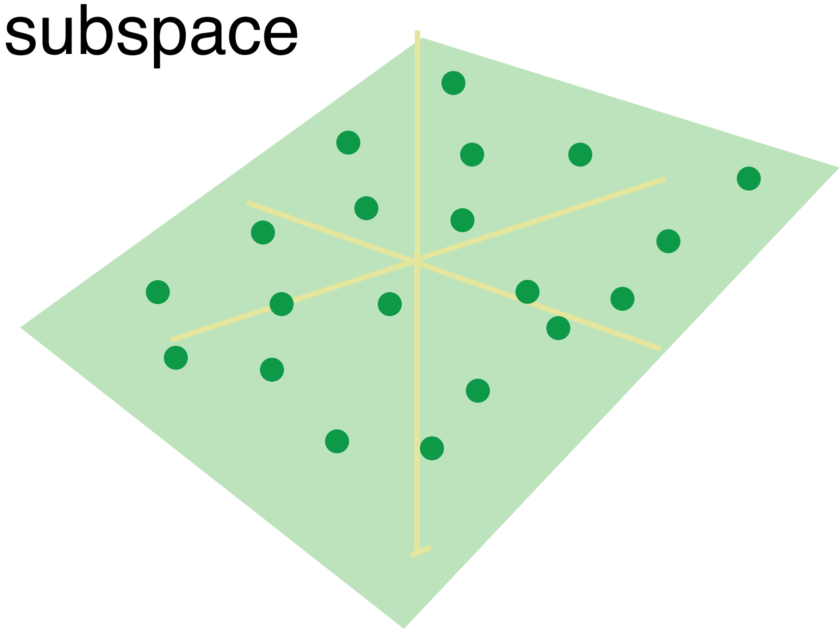
linear subspace
PCA



what is the objective underlying PCA?

low rank model

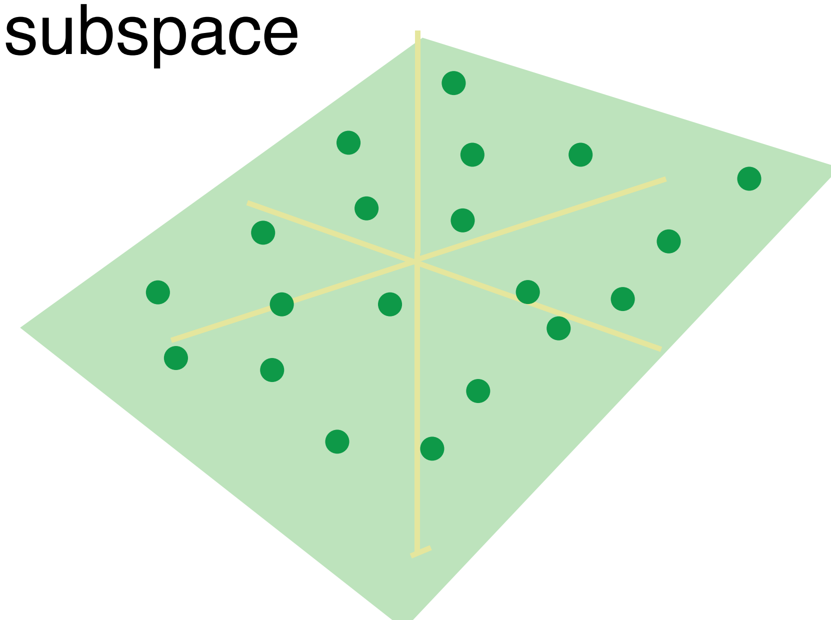
linear subspace
PCA



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

low rank model

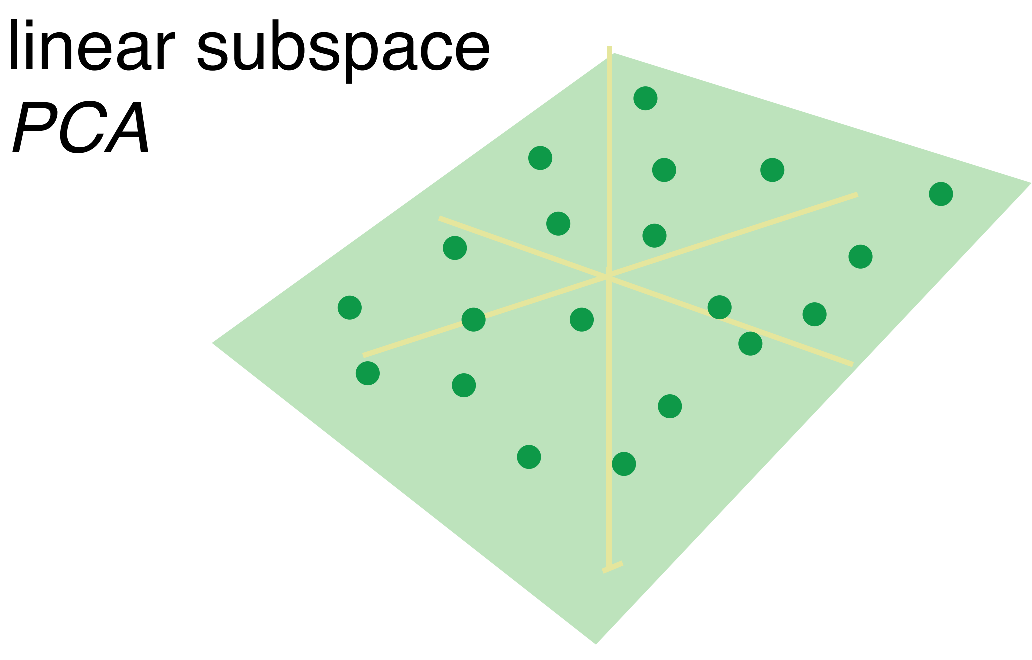
linear subspace
PCA



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

data matrix

low rank model

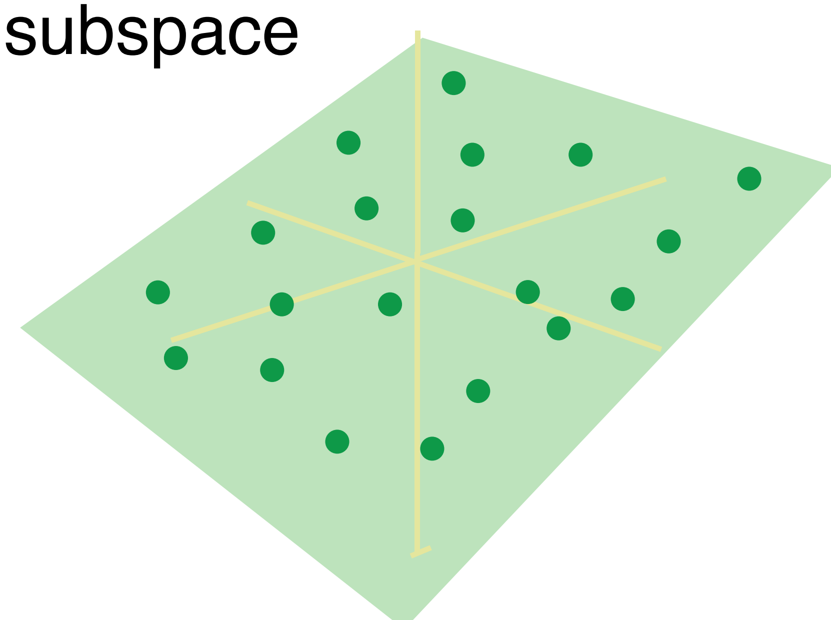


$$\min_{\mathbf{A}} \ \|\mathbf{Y} - \mathbf{A}\|_F \quad \text{s.t.} \quad \text{rank}(\mathbf{A}) \leq \text{k}$$

"low rank" approximation

low rank model

linear subspace
PCA

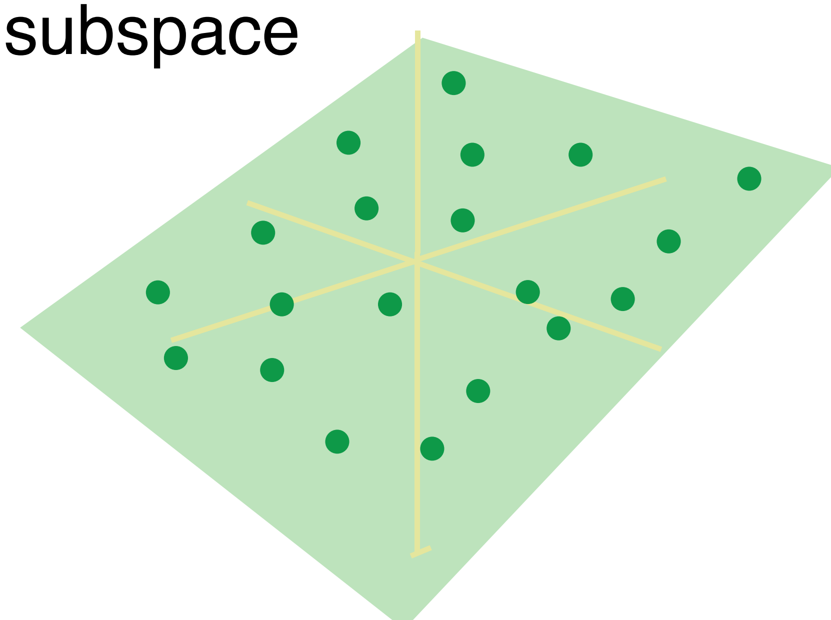


$$\min_{\mathbf{A}} \ \|\mathbf{Y} - \mathbf{A}\|_F \quad \text{s.t.} \quad \text{rank}(\mathbf{A}) \le \text{k}$$

subject to == constraints

low rank model

linear subspace
PCA

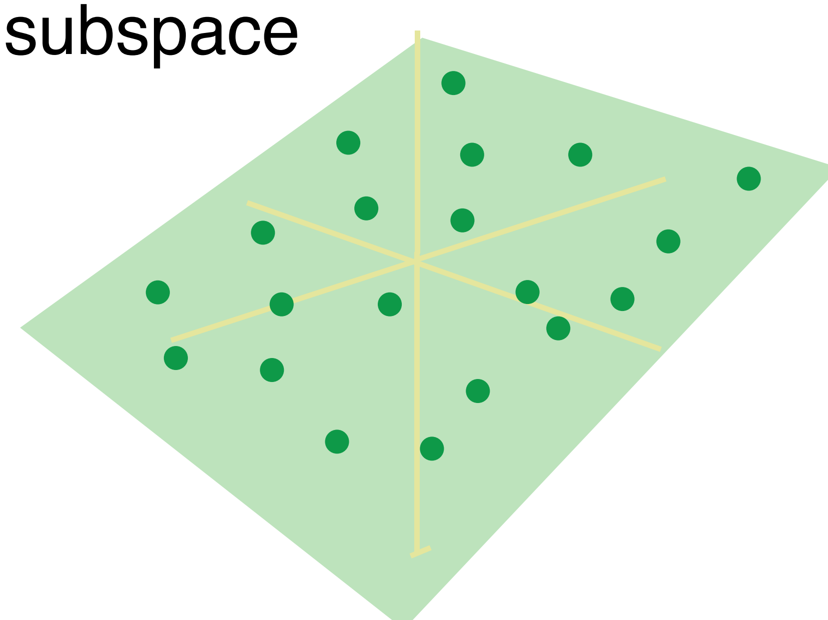


$$\min_{\mathbf{A}} \ \|\mathbf{Y} - \mathbf{A}\|_F \quad \text{s.t.} \quad \text{rank}(\mathbf{A}) \leq \text{k}$$

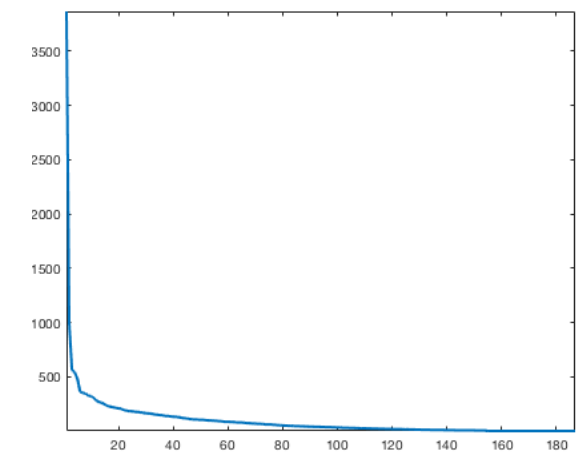
rank(A) = ?

low rank model

linear subspace
PCA



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



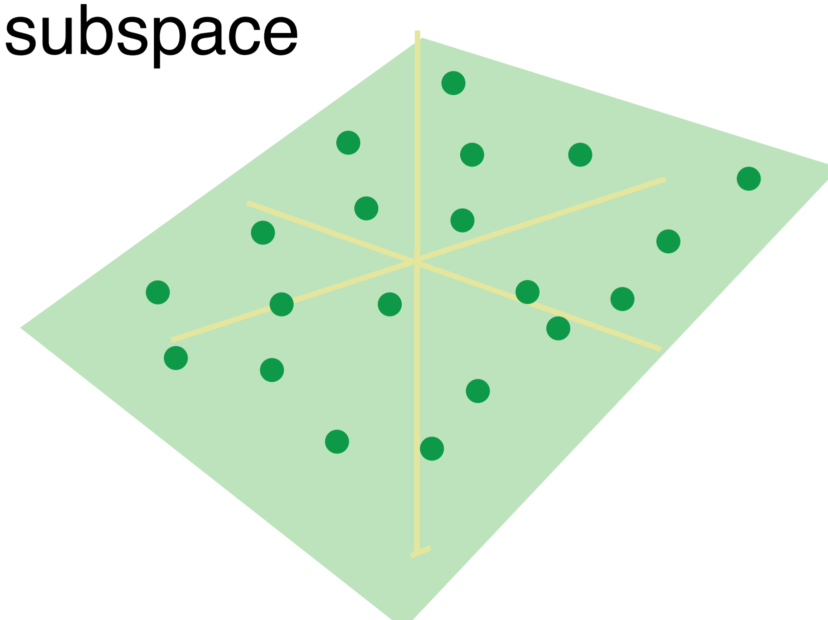
singular values

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

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

low rank model

linear subspace
PCA



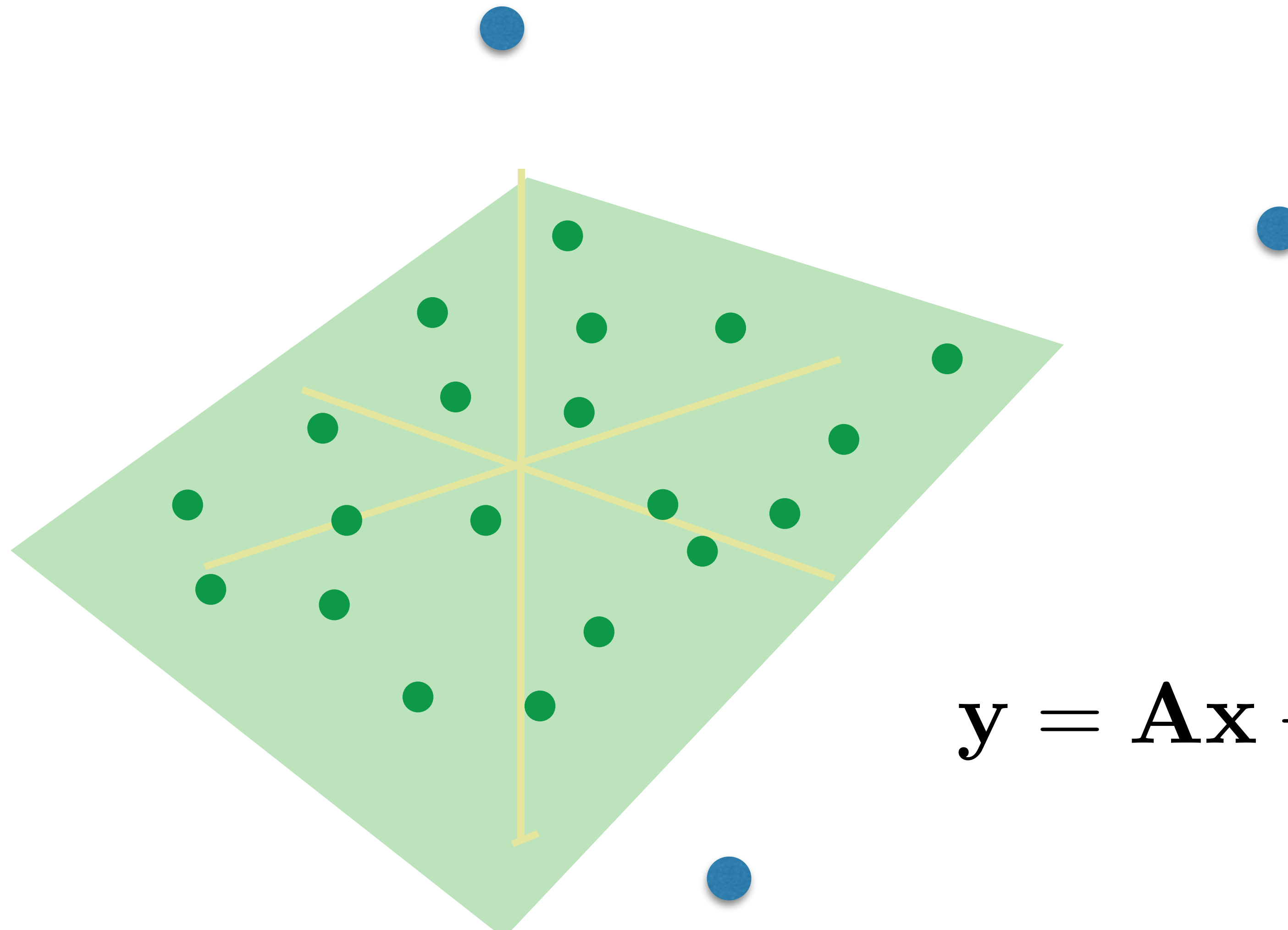
Covariance matrix

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

PCA:

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

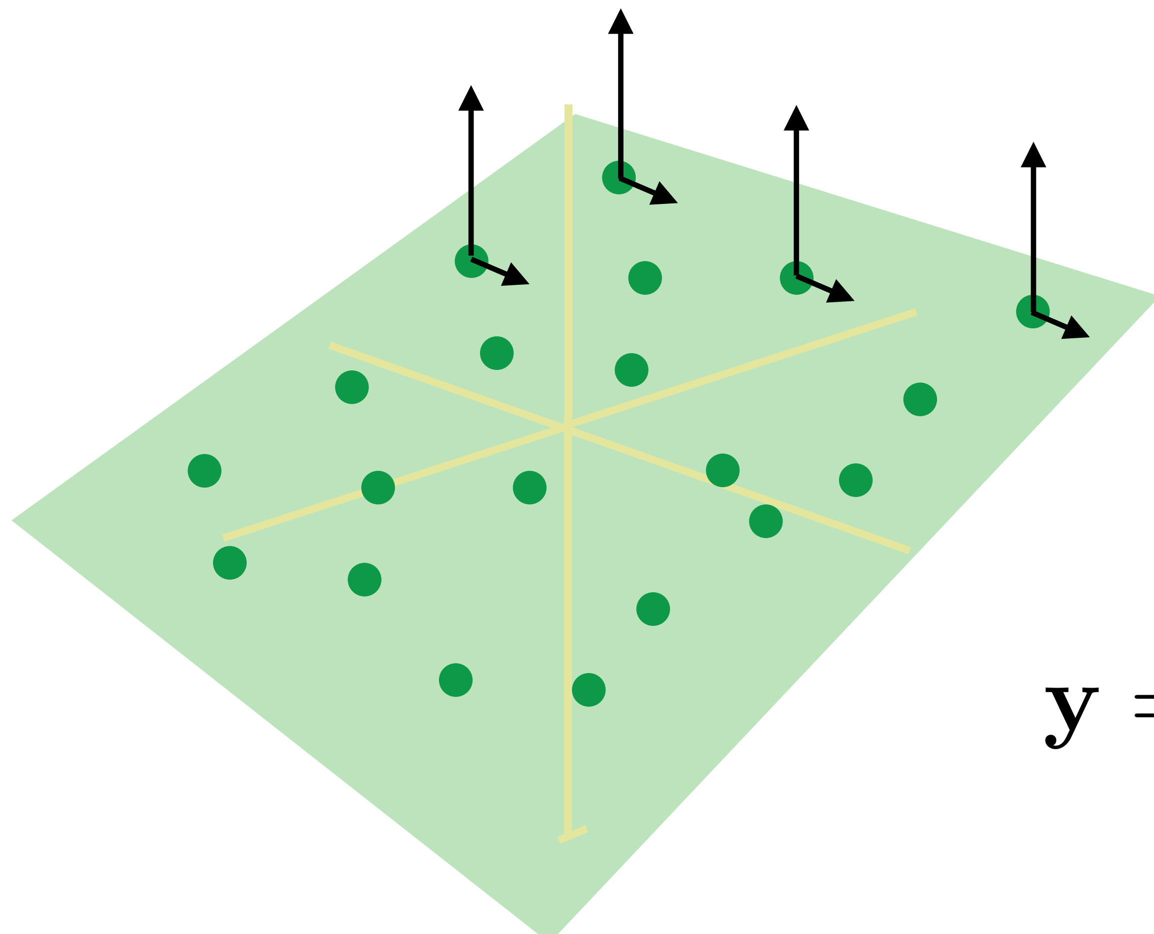
extensions of PCA



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

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

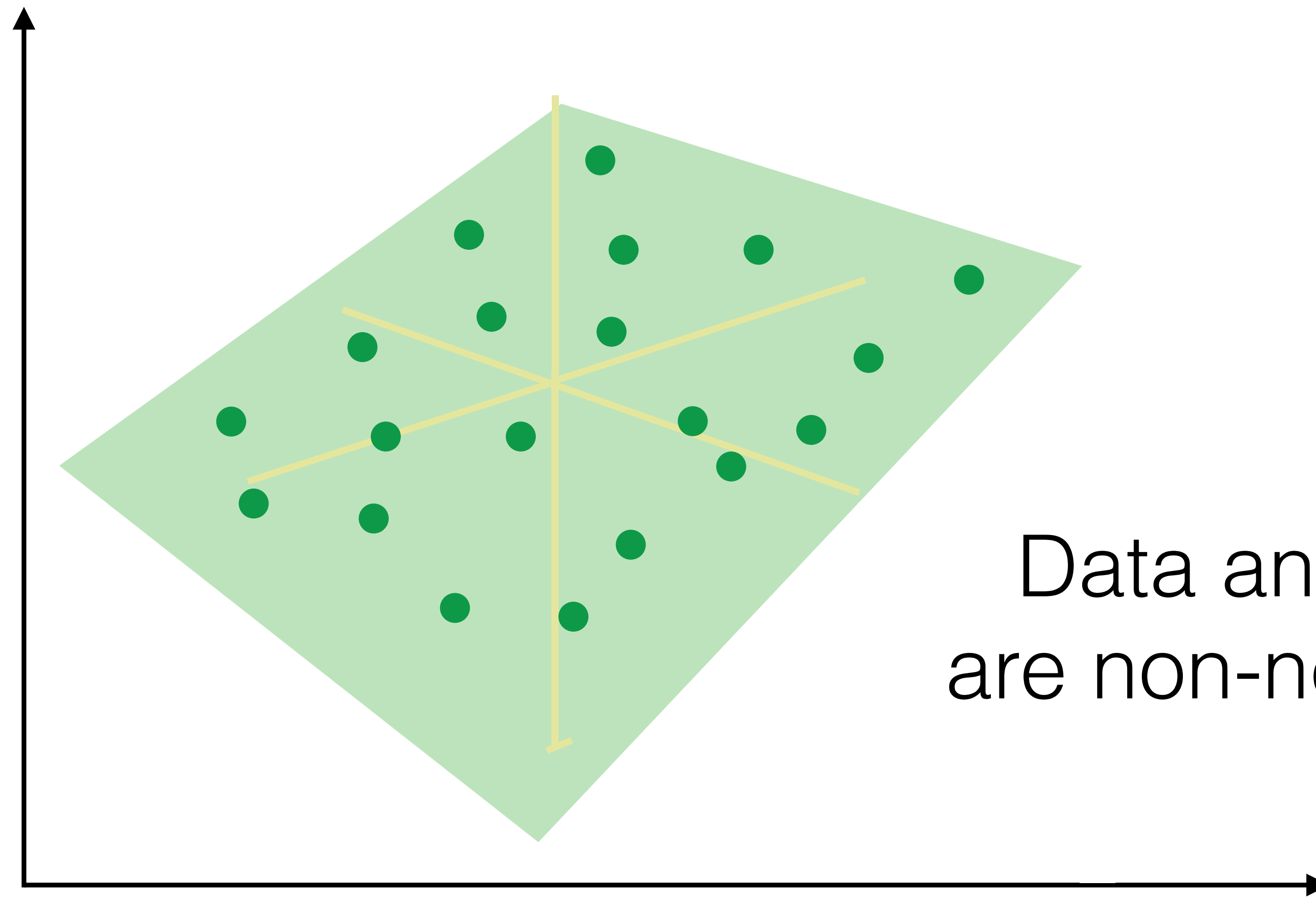
extensions of PCA



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

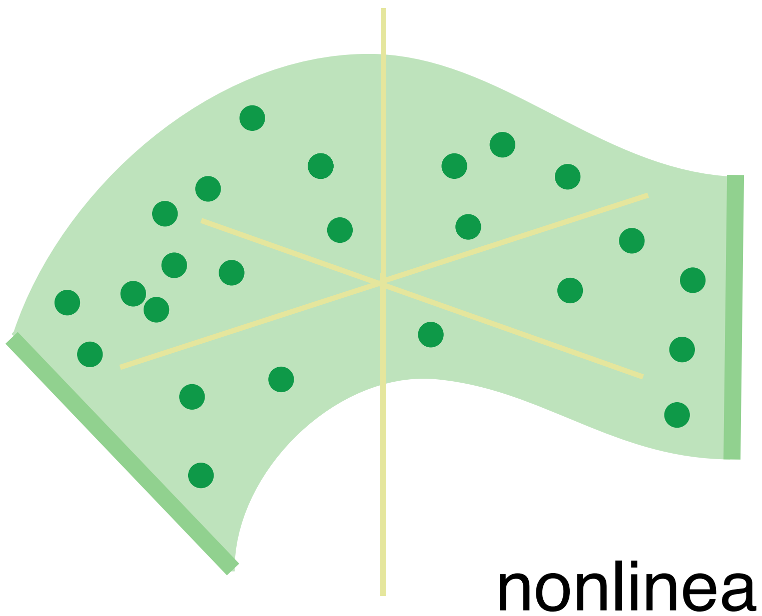
1. Robust PCA - sparse errors
2. **Factor analysis (FA) - noise of unequal variance**
3. Non-negative matrix factorization (NMF)

extensions of PCA



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

nonlinear models (manifolds)



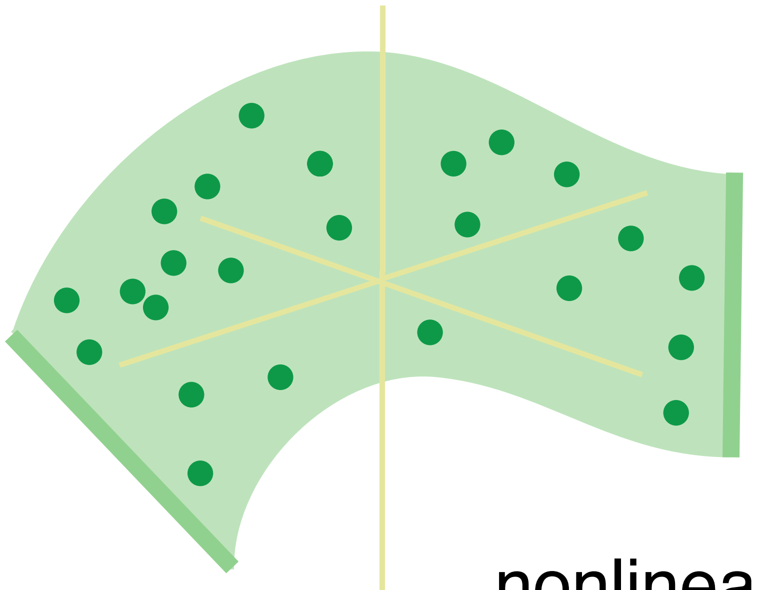
nonlinear manifold
Isomap, LLE

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

distance between original data points

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

nonlinear models (manifolds)



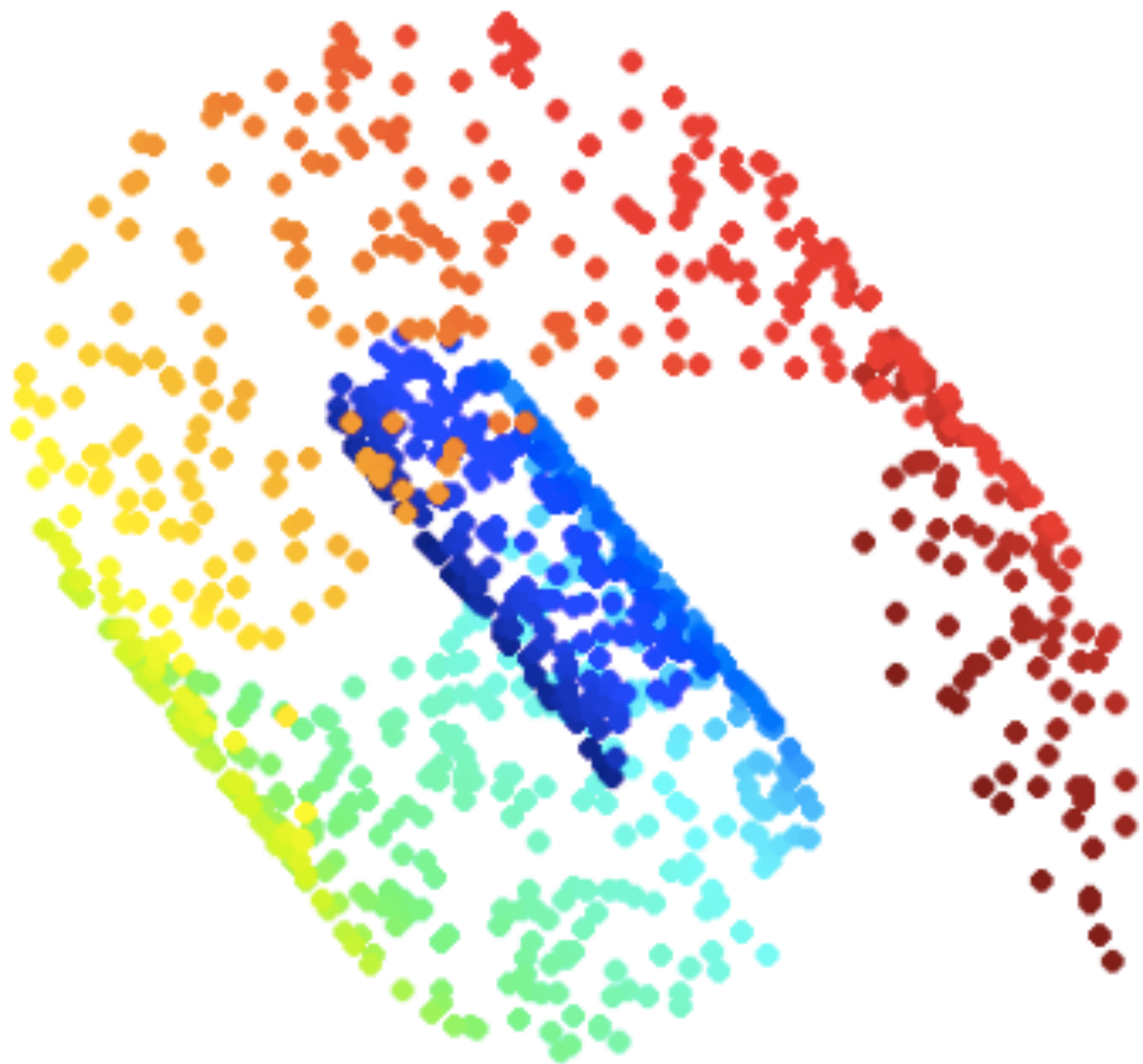
nonlinear manifold
Isomap, LLE

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

distance between projected data points

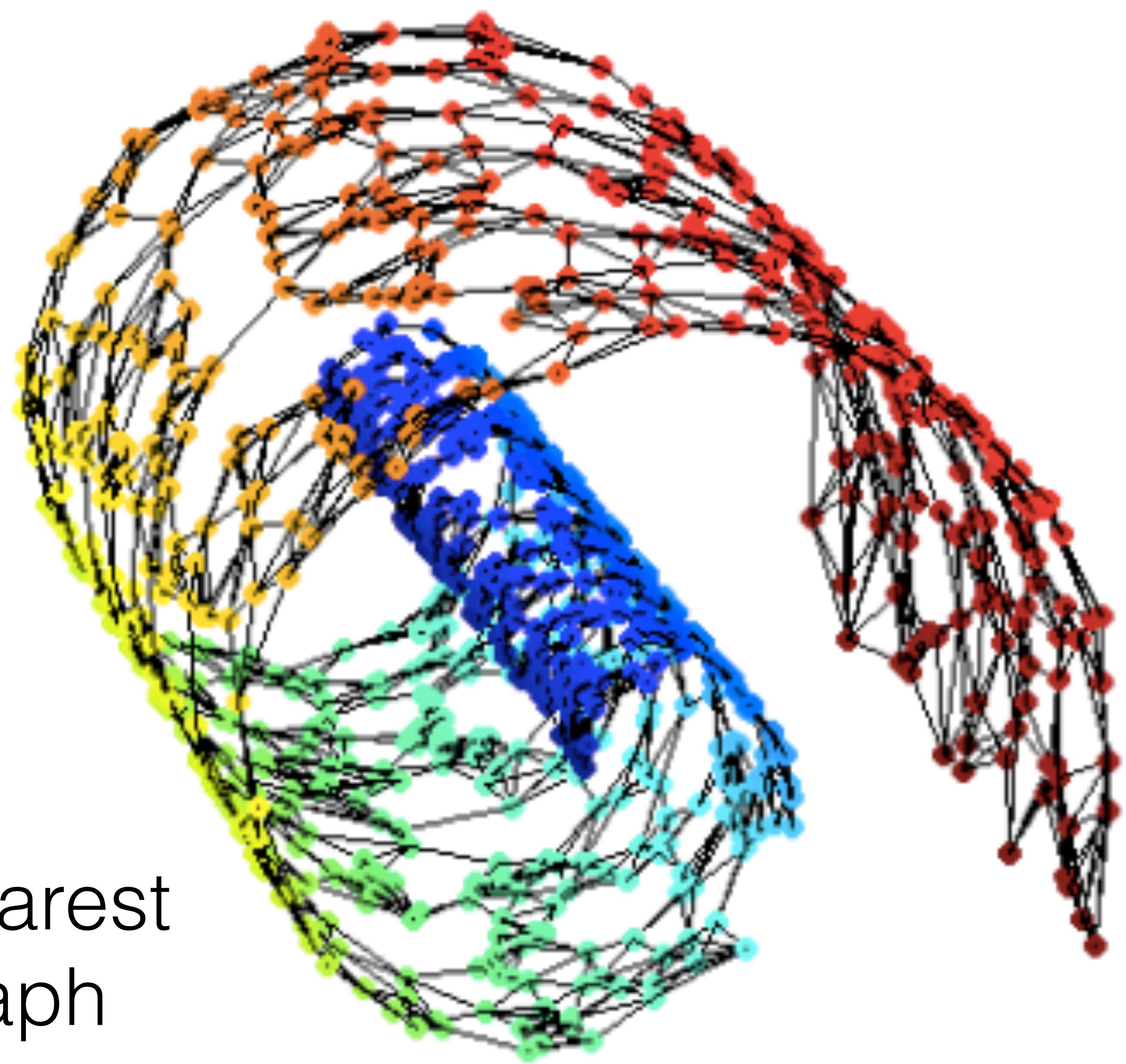
nonlinear models (manifolds)

swiss roll



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

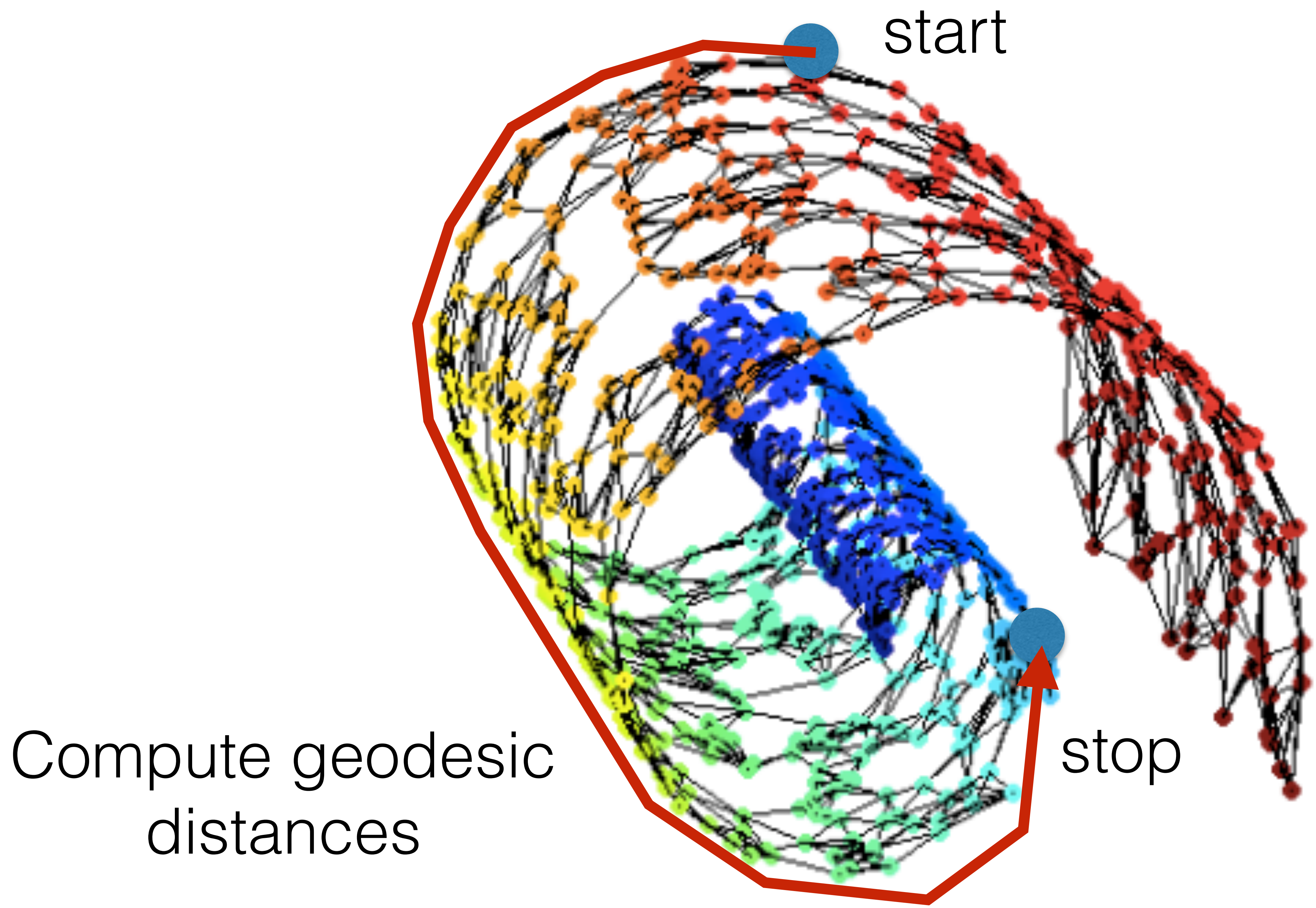
nonlinear models (manifolds)



Compute k-nearest neighbor graph

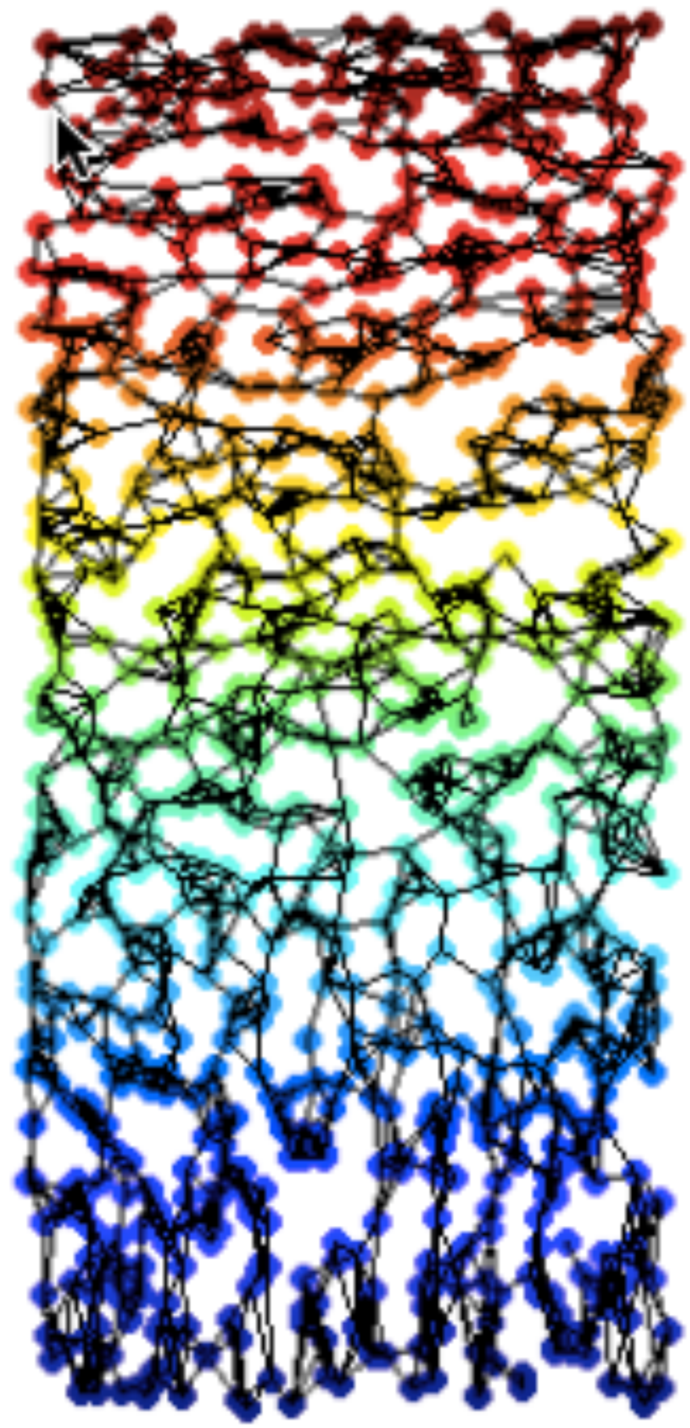
http://www.numerical-tours.com/matlab/shapes_7_isomap/

nonlinear models (manifolds)



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

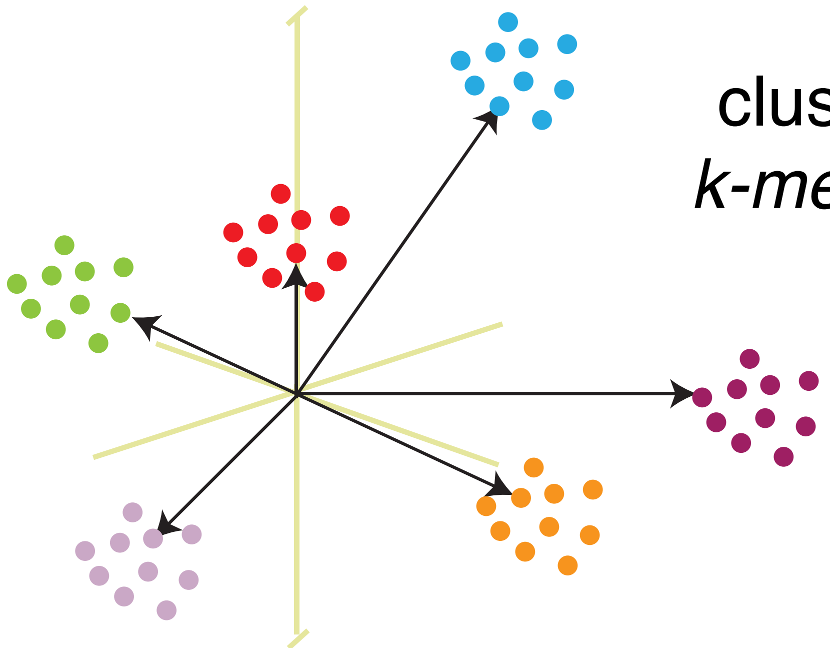
nonlinear models (manifolds)



Compute leading eigenvectors

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

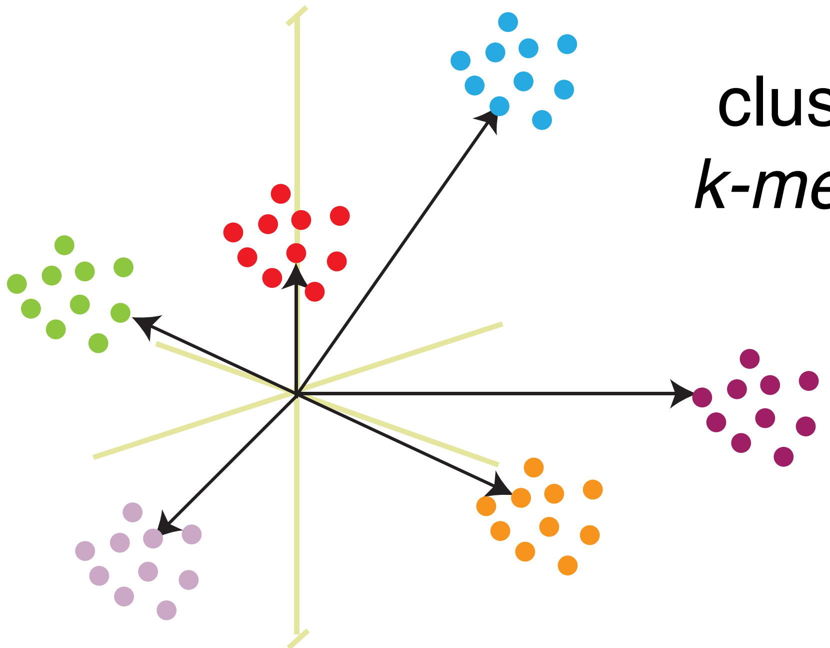
cluster model



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

ith data point

cluster model

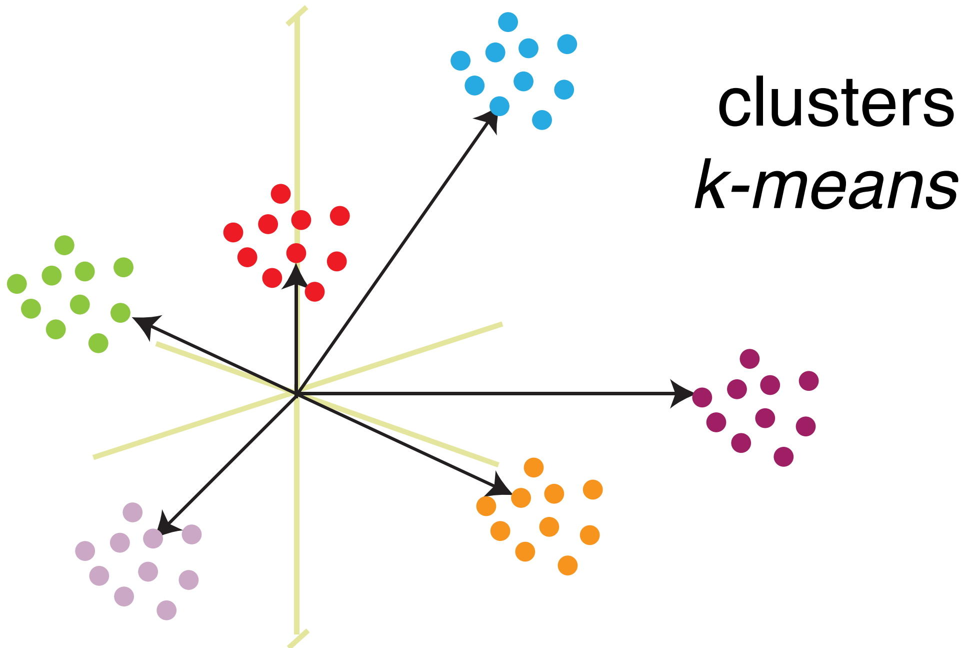


clusters
k-means

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

jth cluster center

cluster model

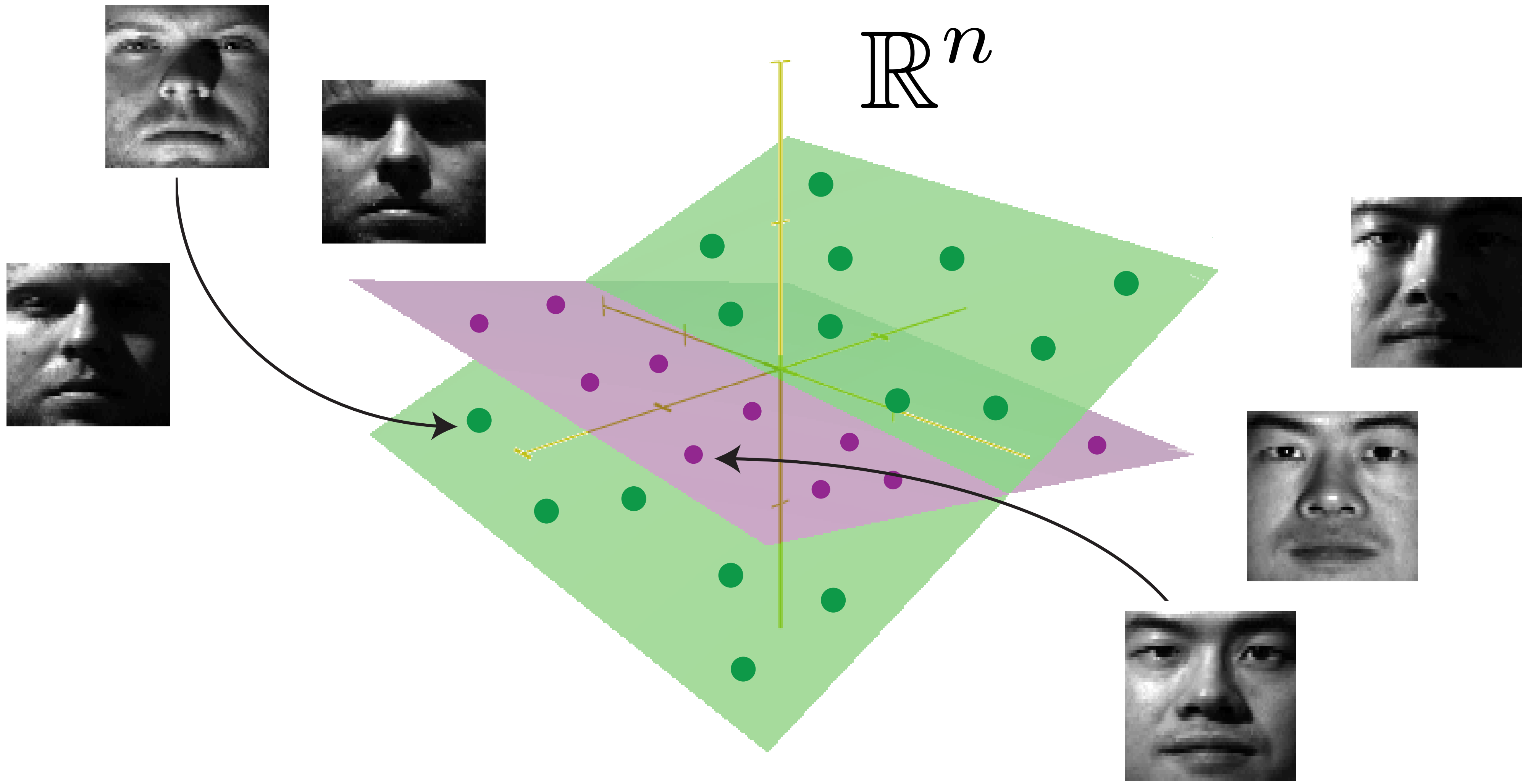


$$\min_{\mathbf{c}_1,\cdots,\mathbf{c}_k} \sum_{j=1}^{k} \sum_{i\in\Omega_j} \|y_i - c_j\|_2$$

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

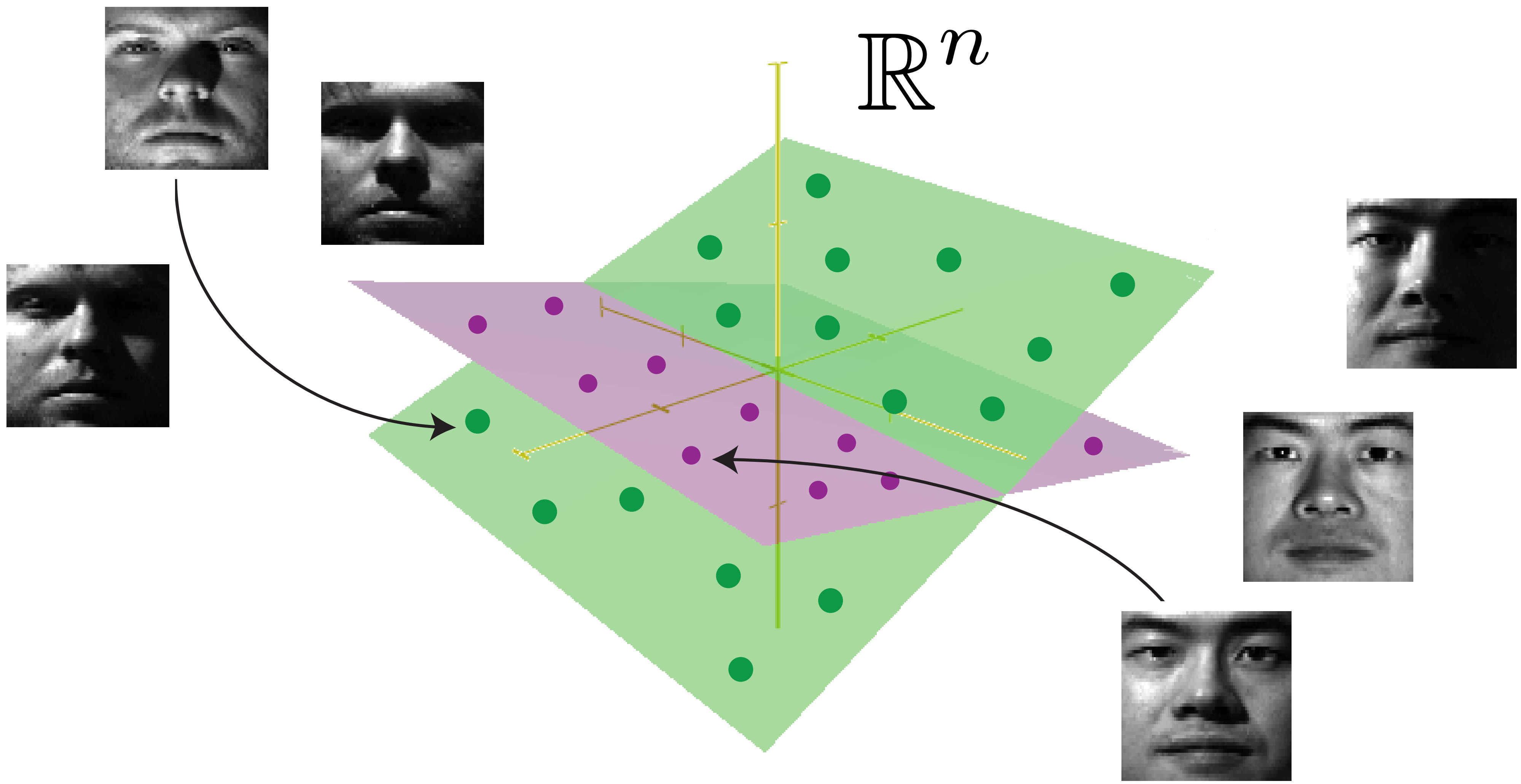


$$\min_{\mathbf{A}_i, \Omega_i} \quad \sum \|\mathbf{Y}_{\Omega_i} - \mathbf{A}_i\|_F \quad \text{s.t.} \quad \text{rank}(\mathbf{A}_i) \leq k_i$$

subset of data in ith subspace

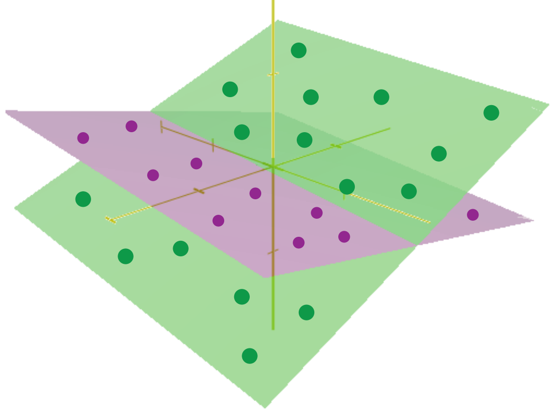
Elhamifar, 2012

union of subspaces

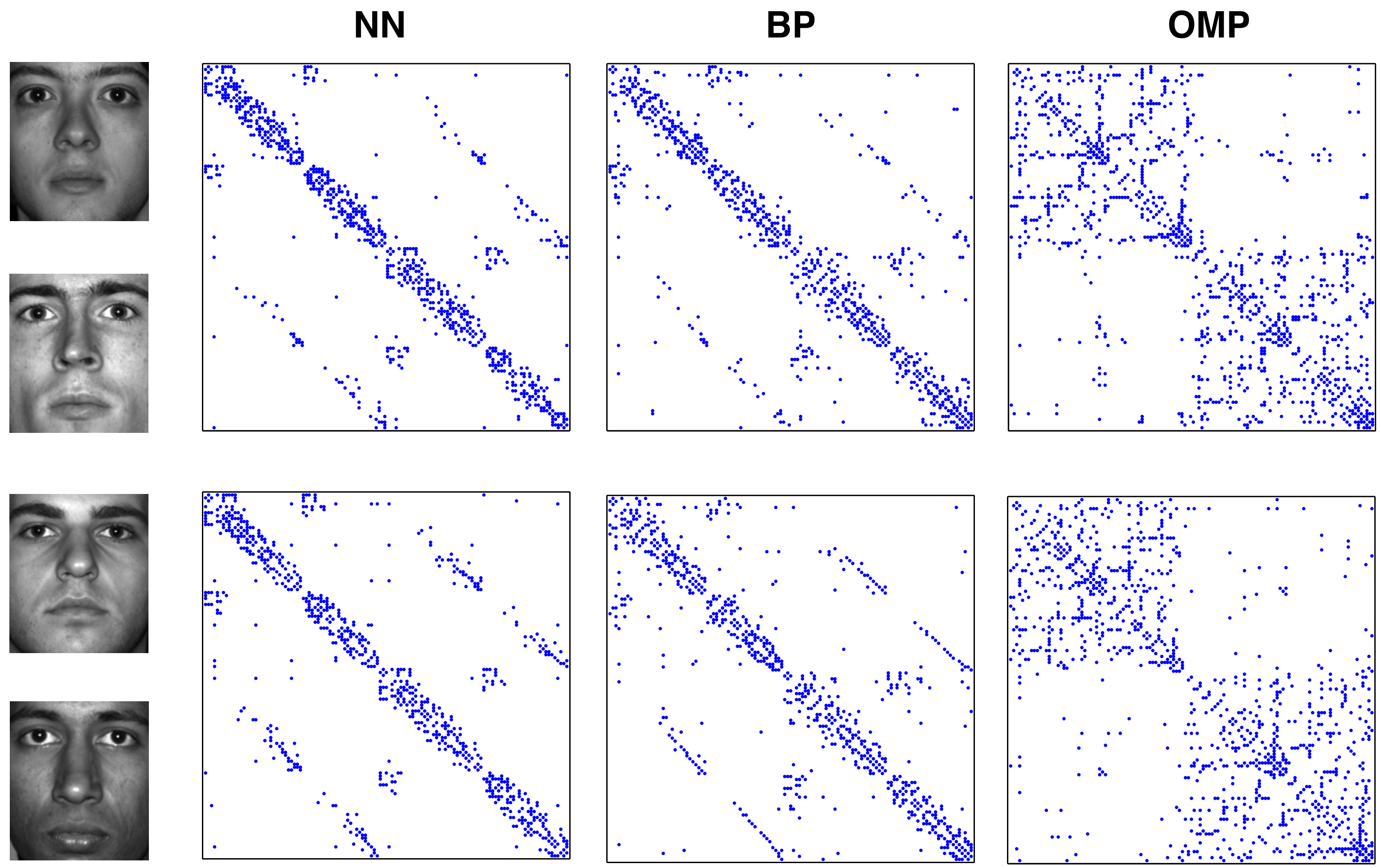


$$\min_{\mathbf{A}_i,\Omega_i} \quad \sum \|\mathbf{Y}_{\Omega_i} - \mathbf{A}_i\|_F \quad \text{s.t.} \quad \text{rank}(\mathbf{A}_i) \leq k_i$$

low rank approx of ith cluster



union of subspaces

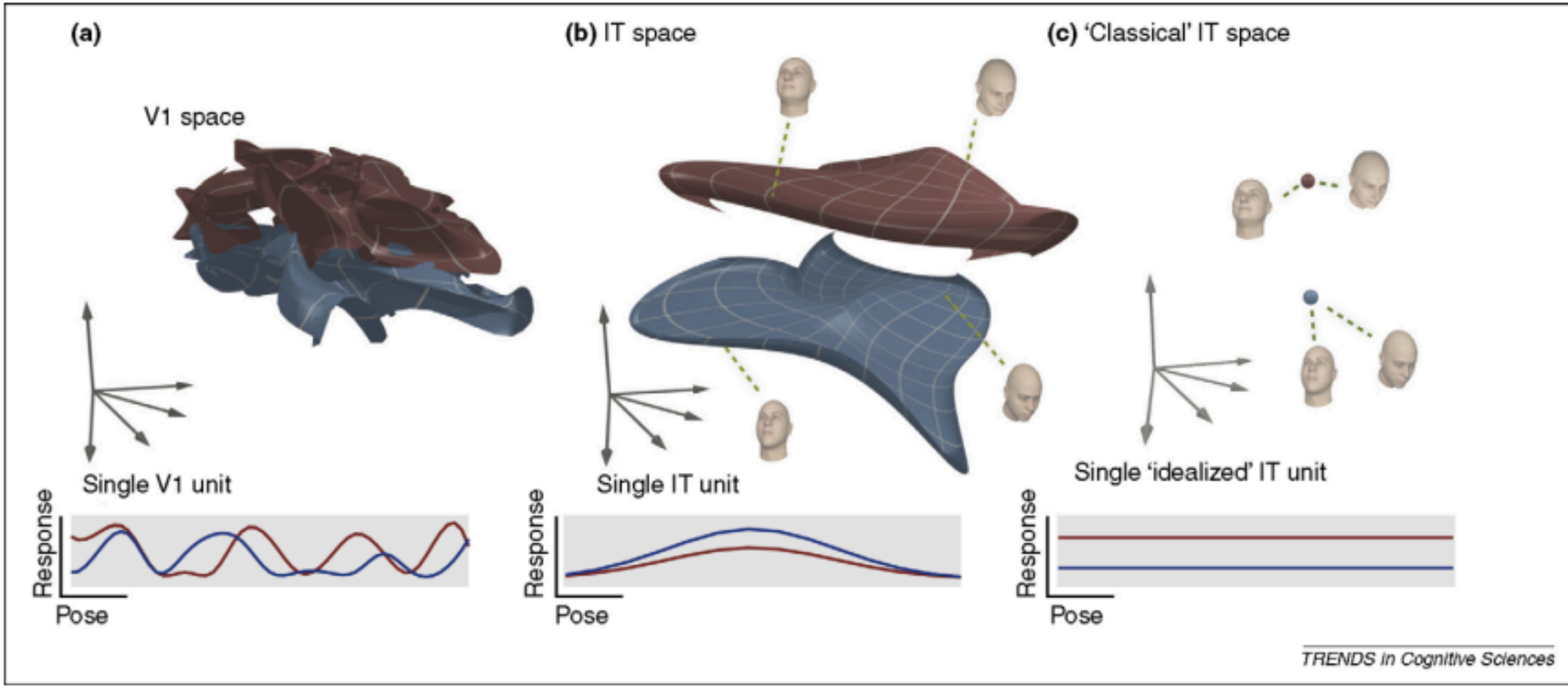


Greedy SSC:

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

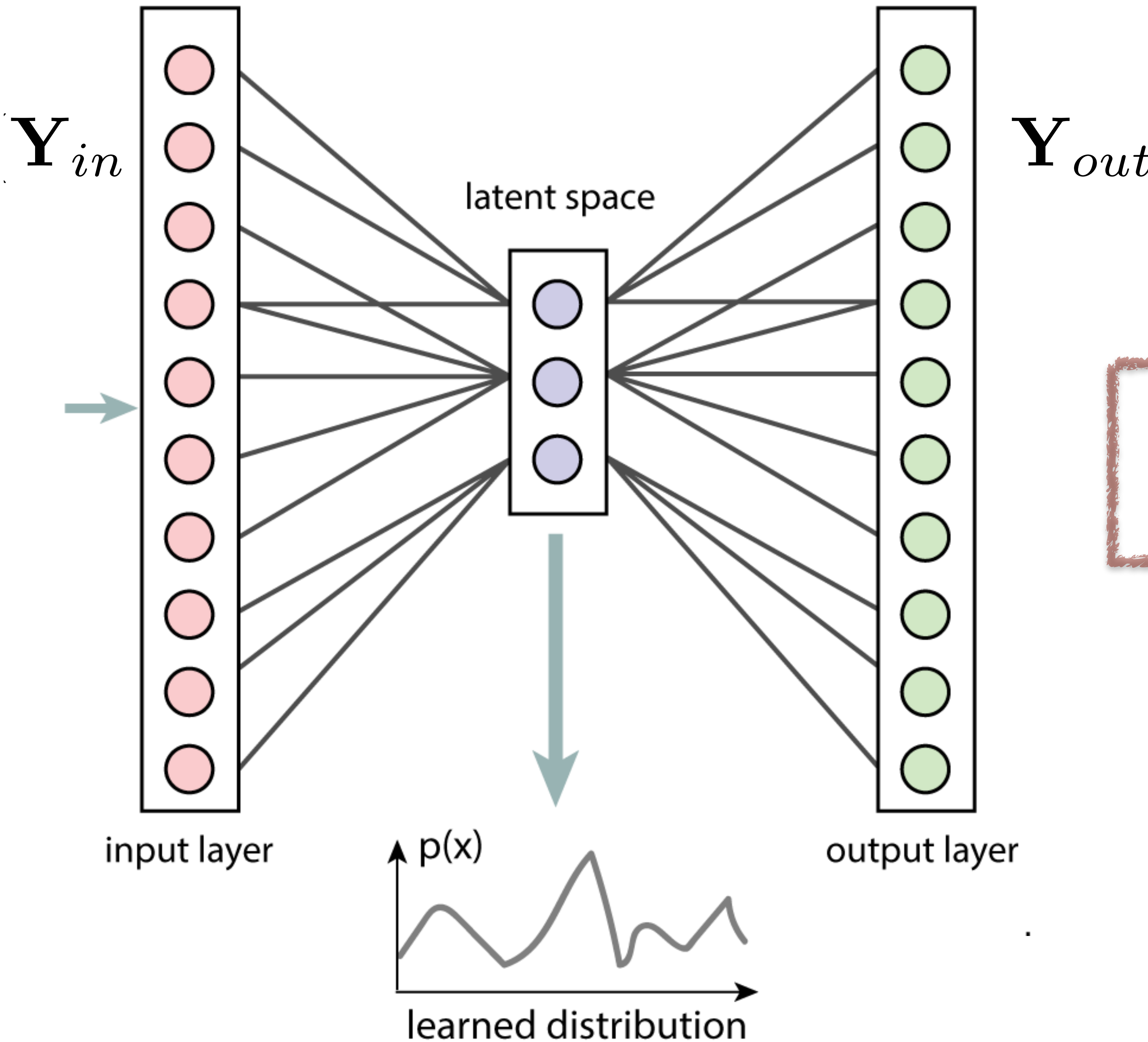
Dyer, JMLR 2013

“tangled” manifolds



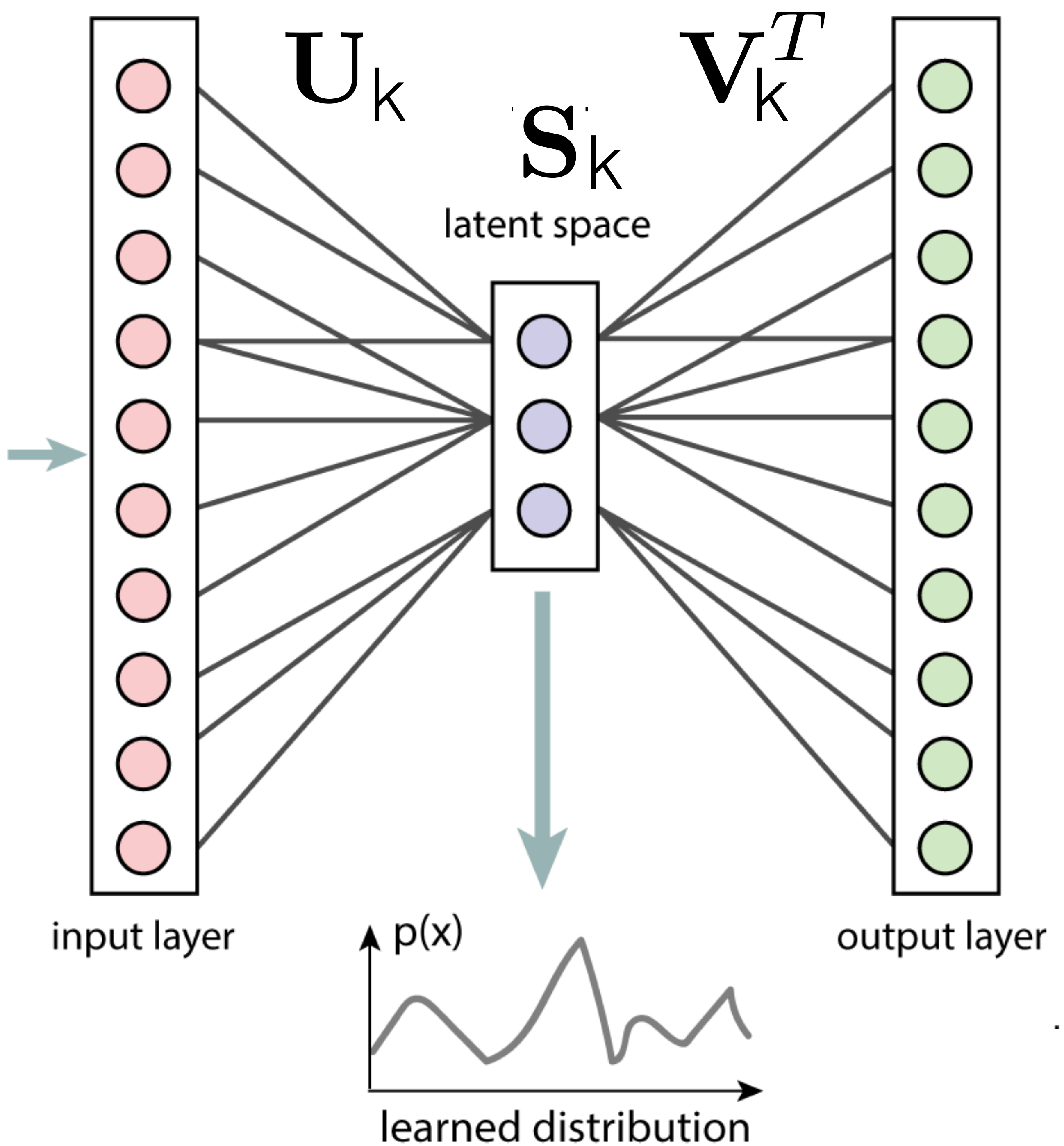
DiCarlo, Cox, 2007

autoencoder architecture



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

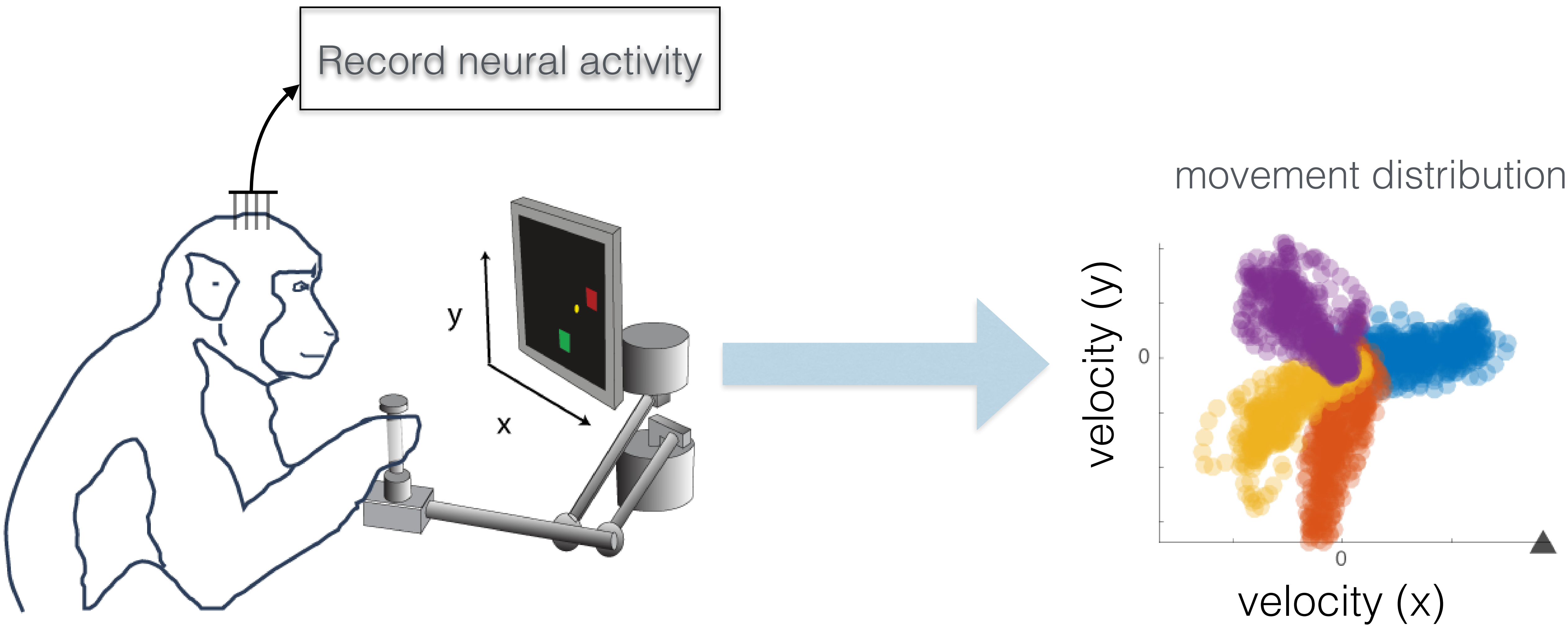
linear autoencoder -> PCA



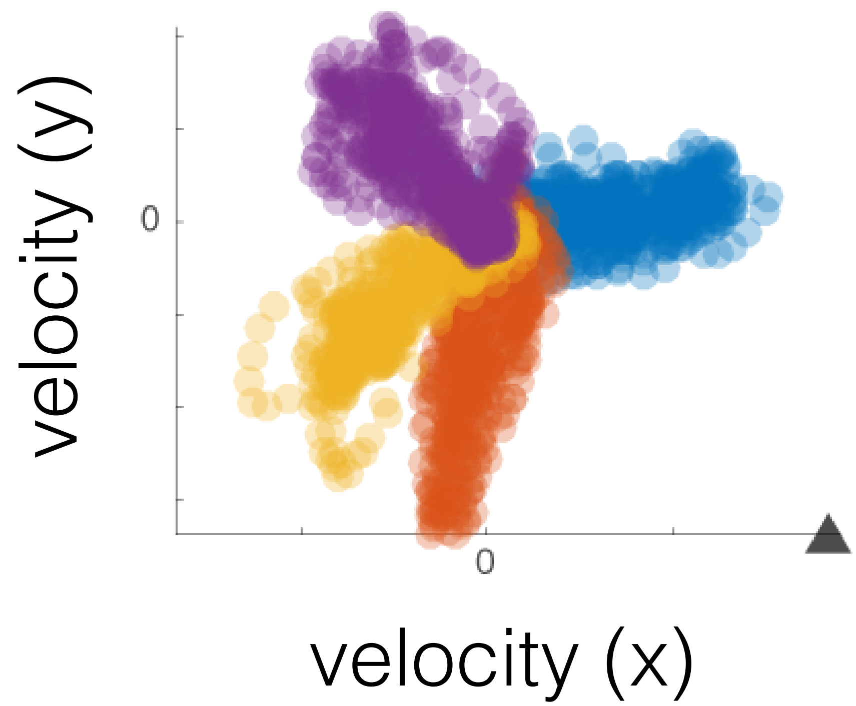
$$\mathbf{Y} \approx \mathbf{U}_k \mathbf{S}_k \mathbf{V}_k^T$$

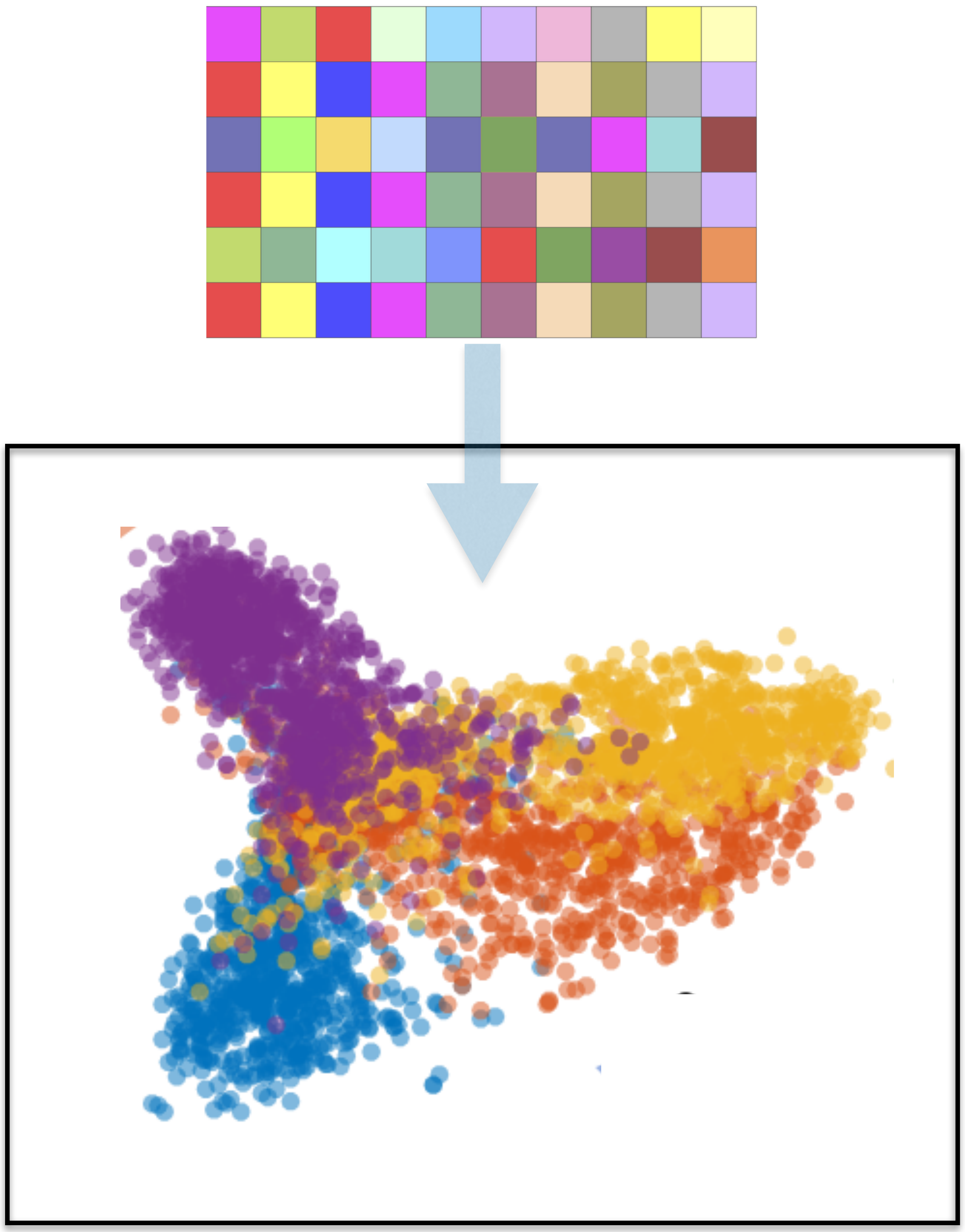
application to neural decoding

movement decoding

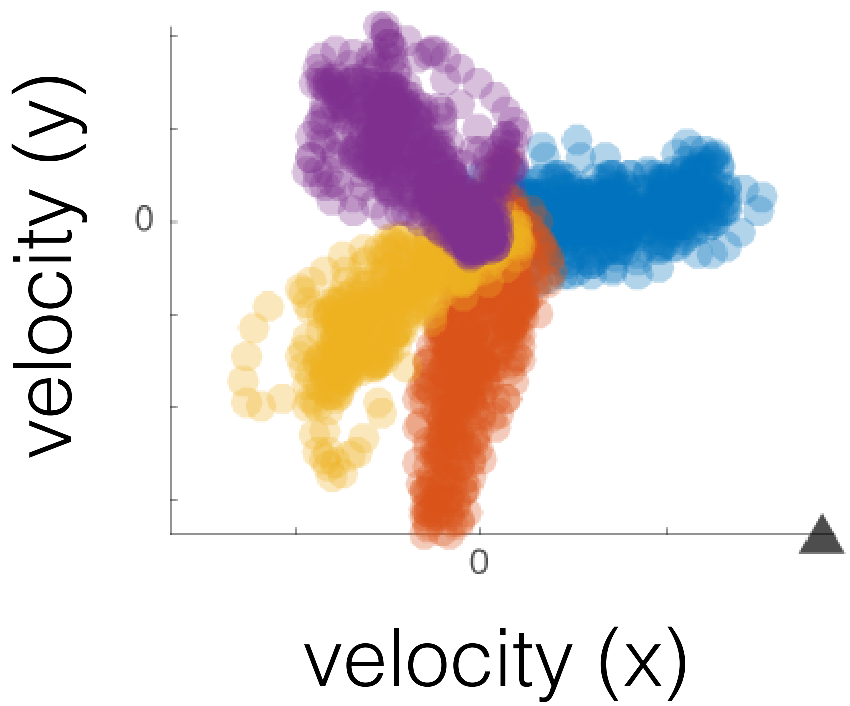


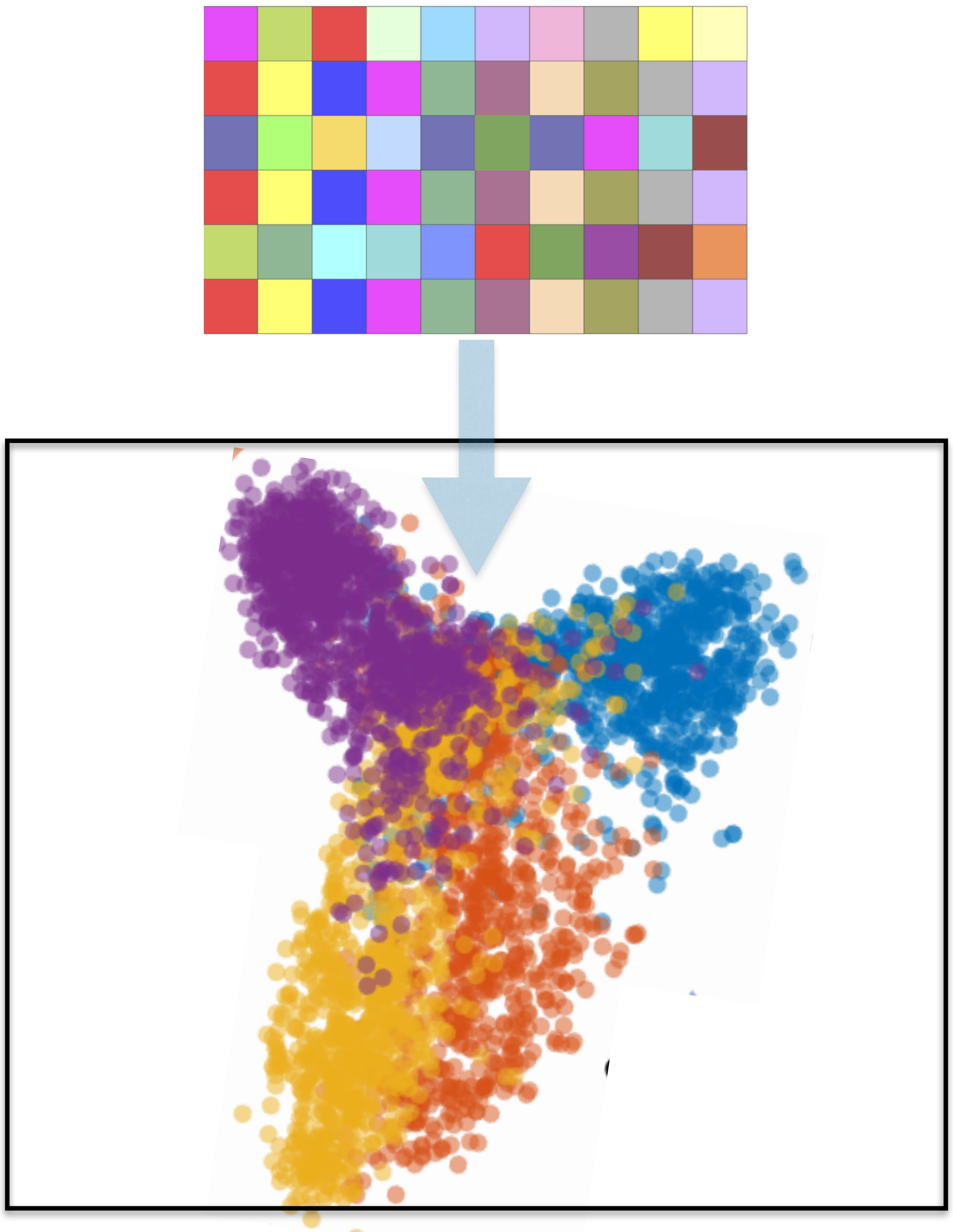
neural responses are low-d



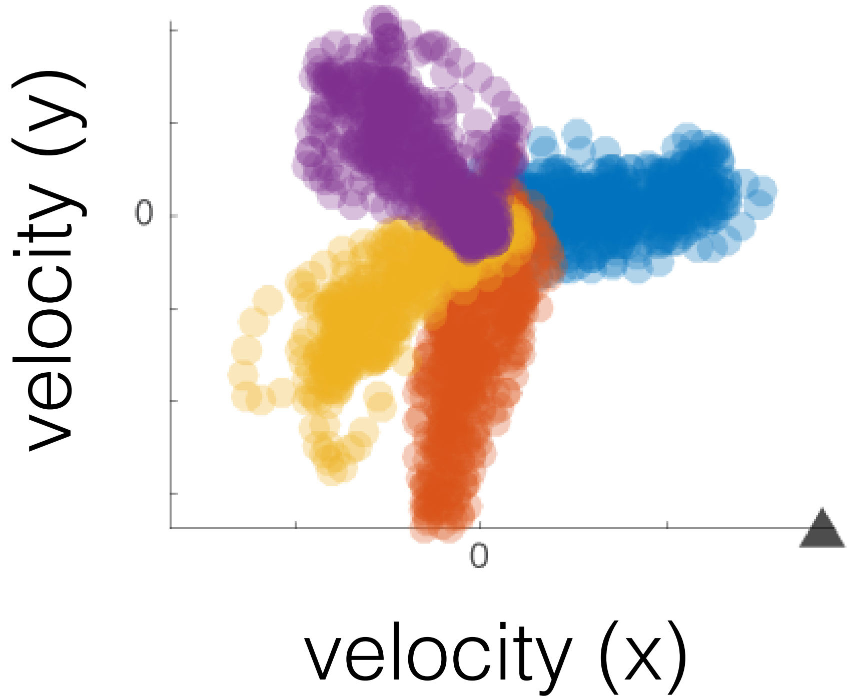


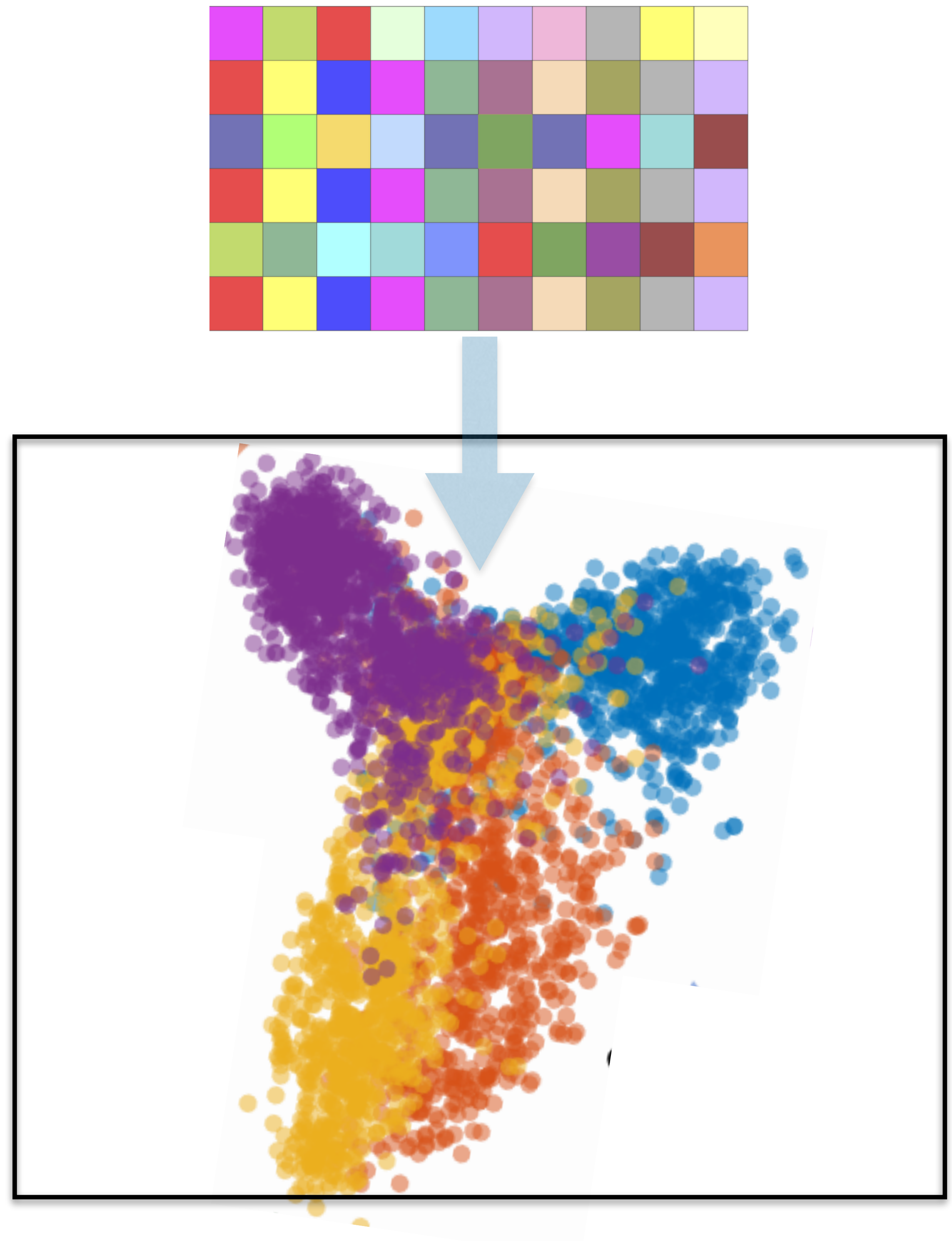
neural responses are low-d





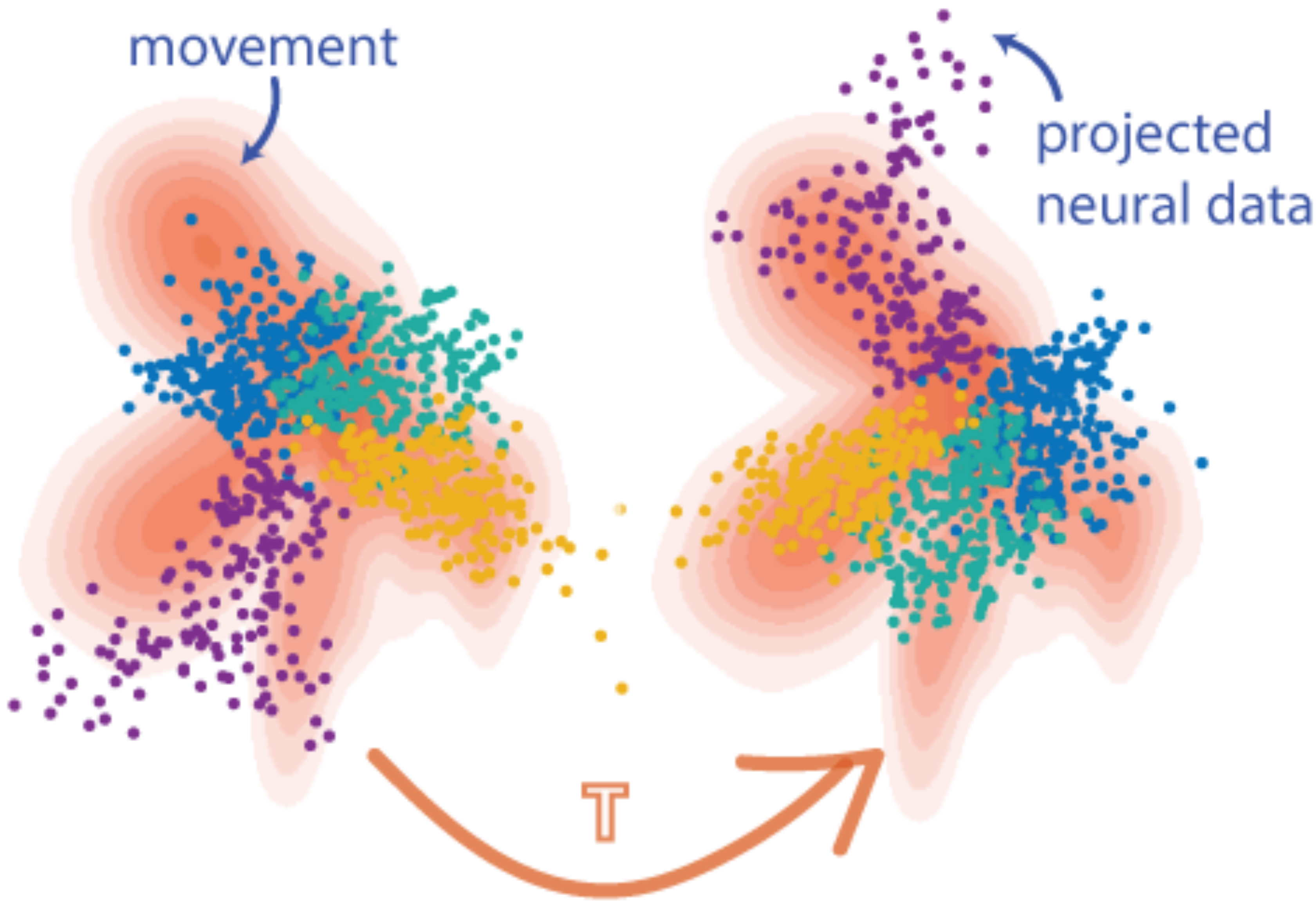
neural responses are low-d





solution: leverage distribution of known movements

neural distribution alignment

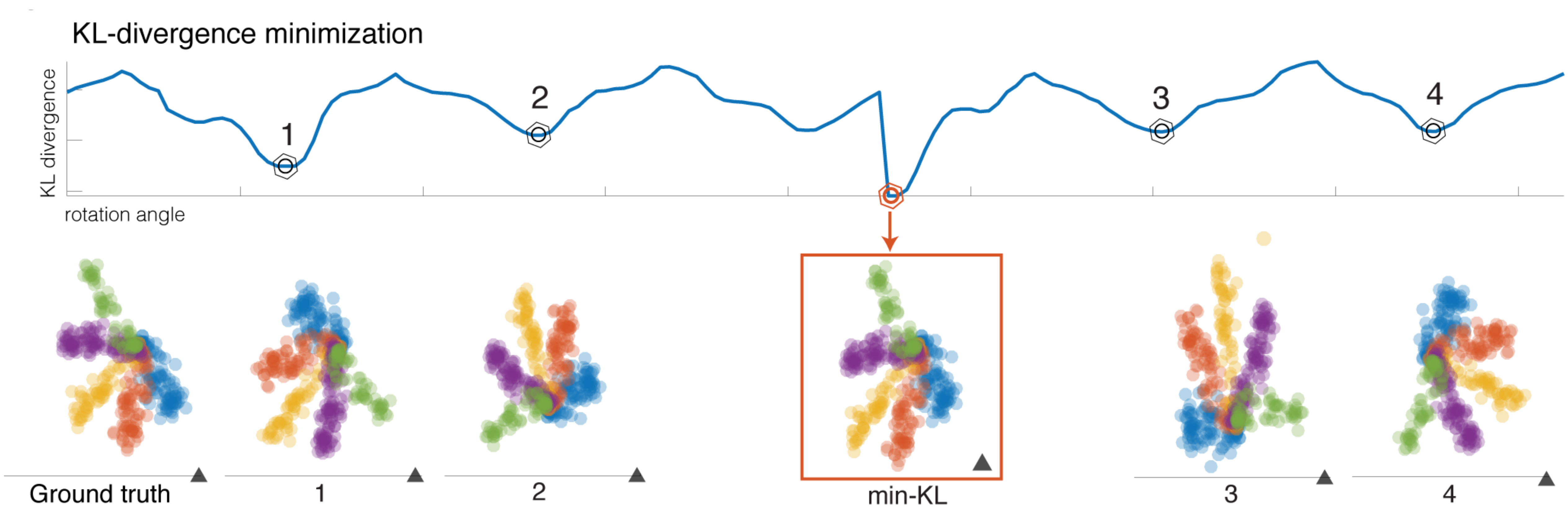


$$\min_{T \in \mathcal{T}} \mathcal{D}(T(\mu), \nu)$$

Dyer, Nature BME, 2017

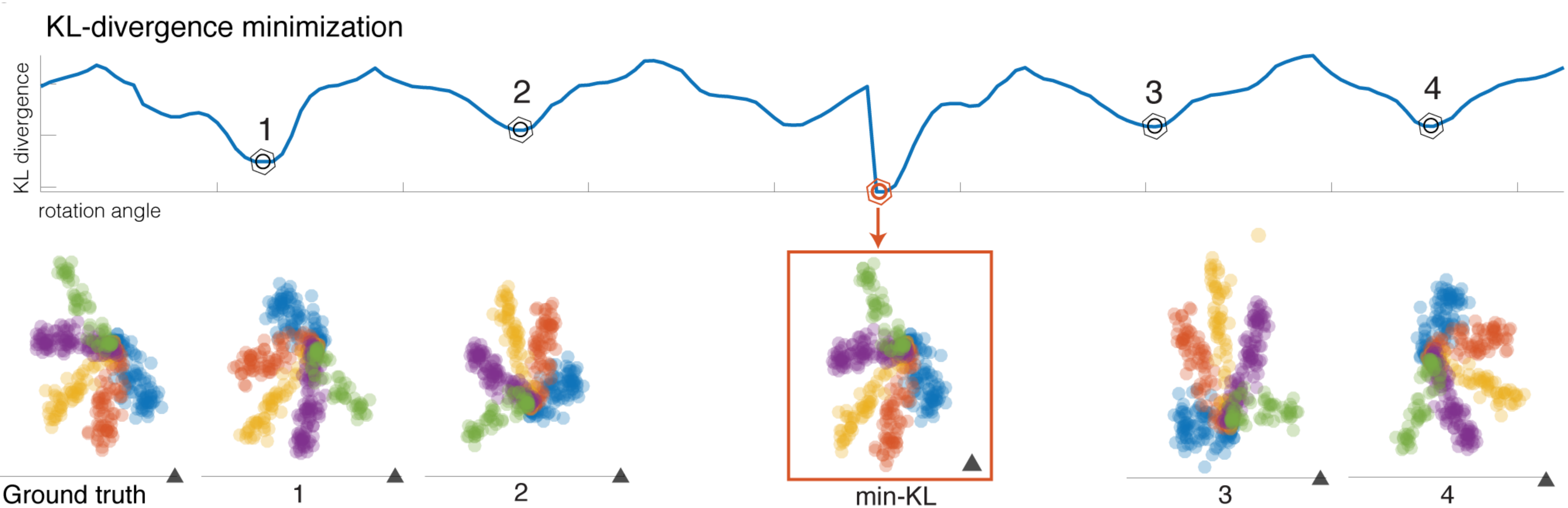
github.com/nerdslab/DAD

KL-divergence minimization



goal: align neural activities with prior movement distribution

KL-divergence minimization

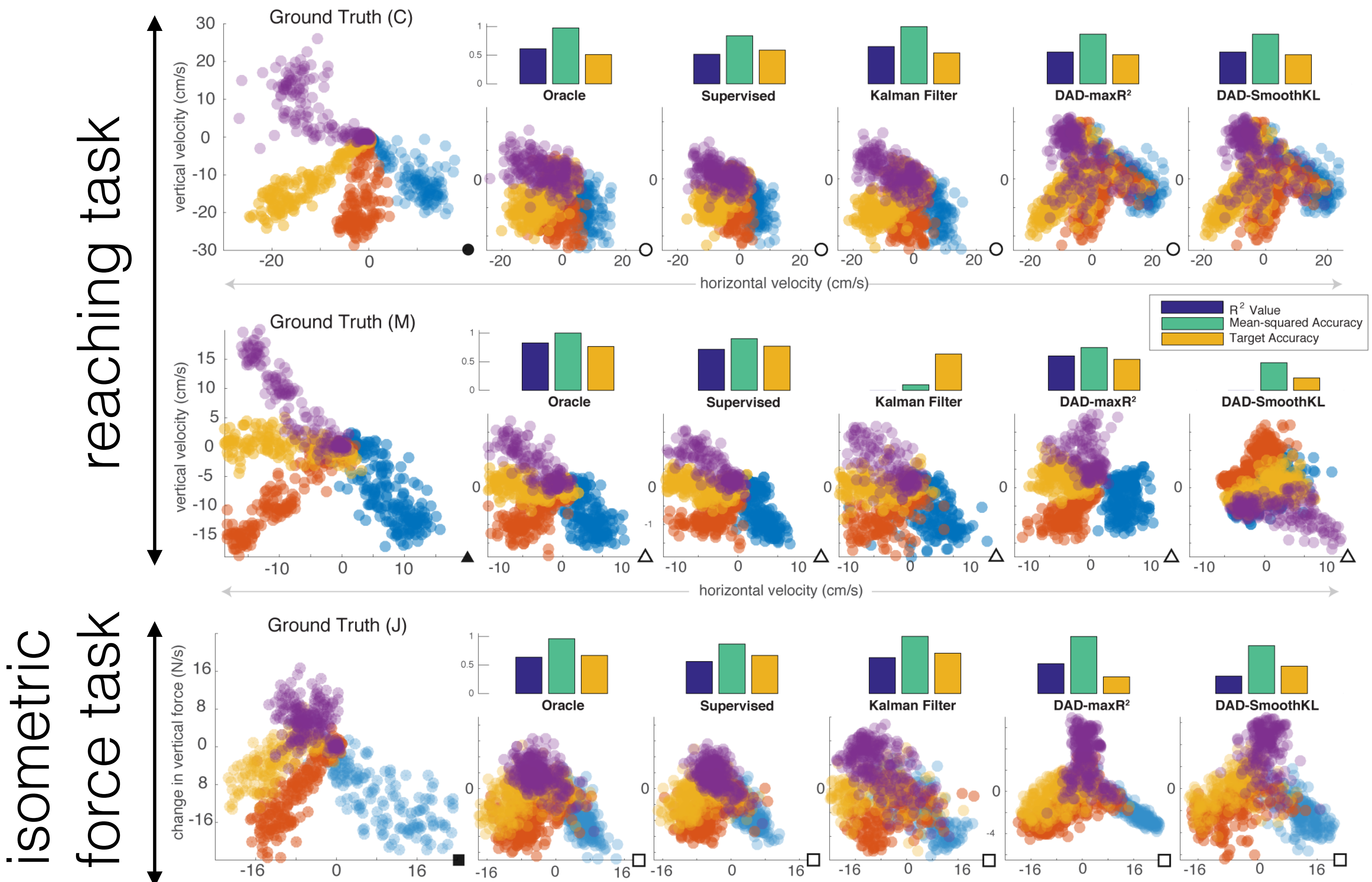


$$\mathbf{H}^* = \arg\min_{\mathbf{H}\in\mathbb{R}^{d\times 3}} \mathbf{KL}(p||q)$$

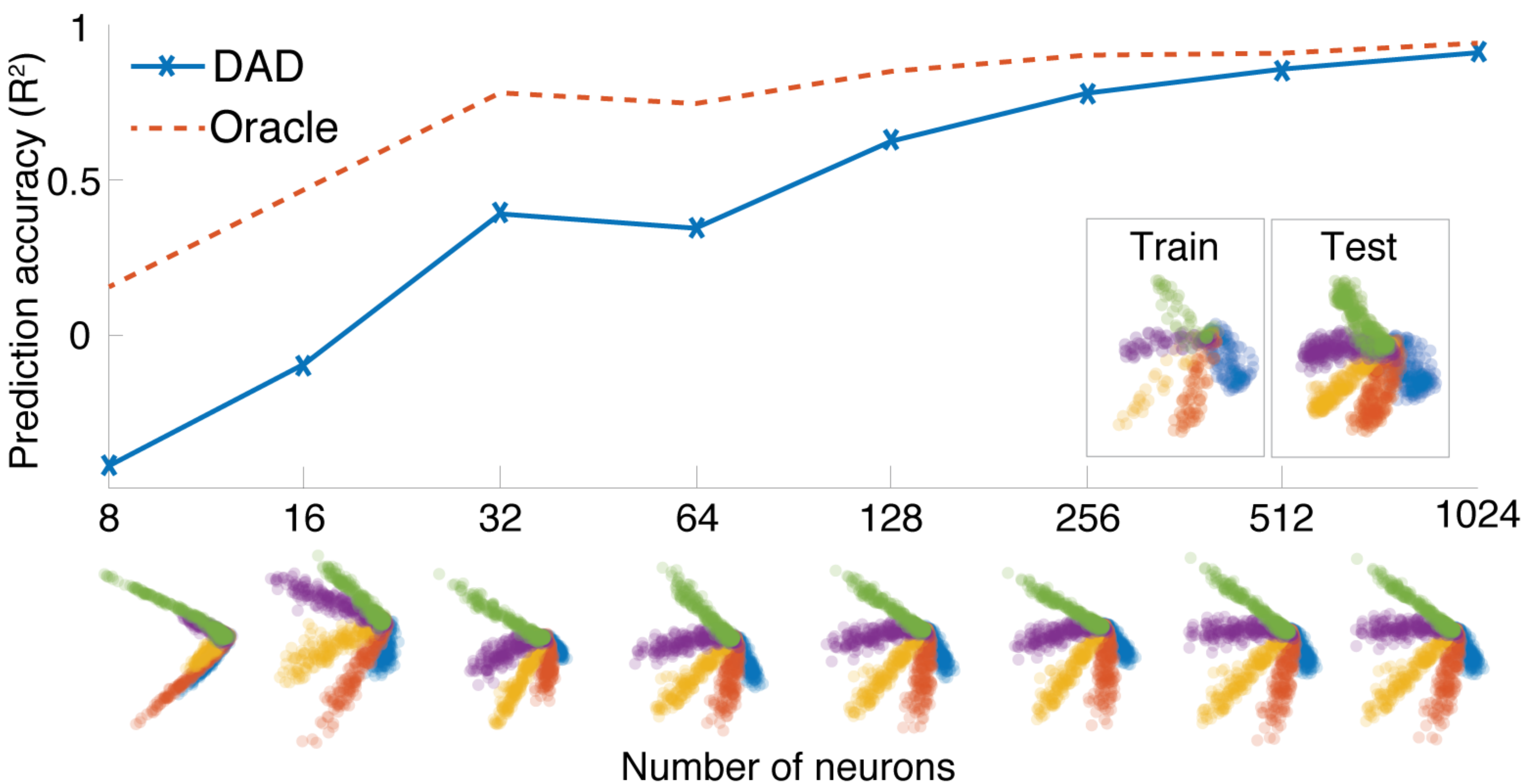
Estimate **p** from $\widetilde{\mathbf{V}}$

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

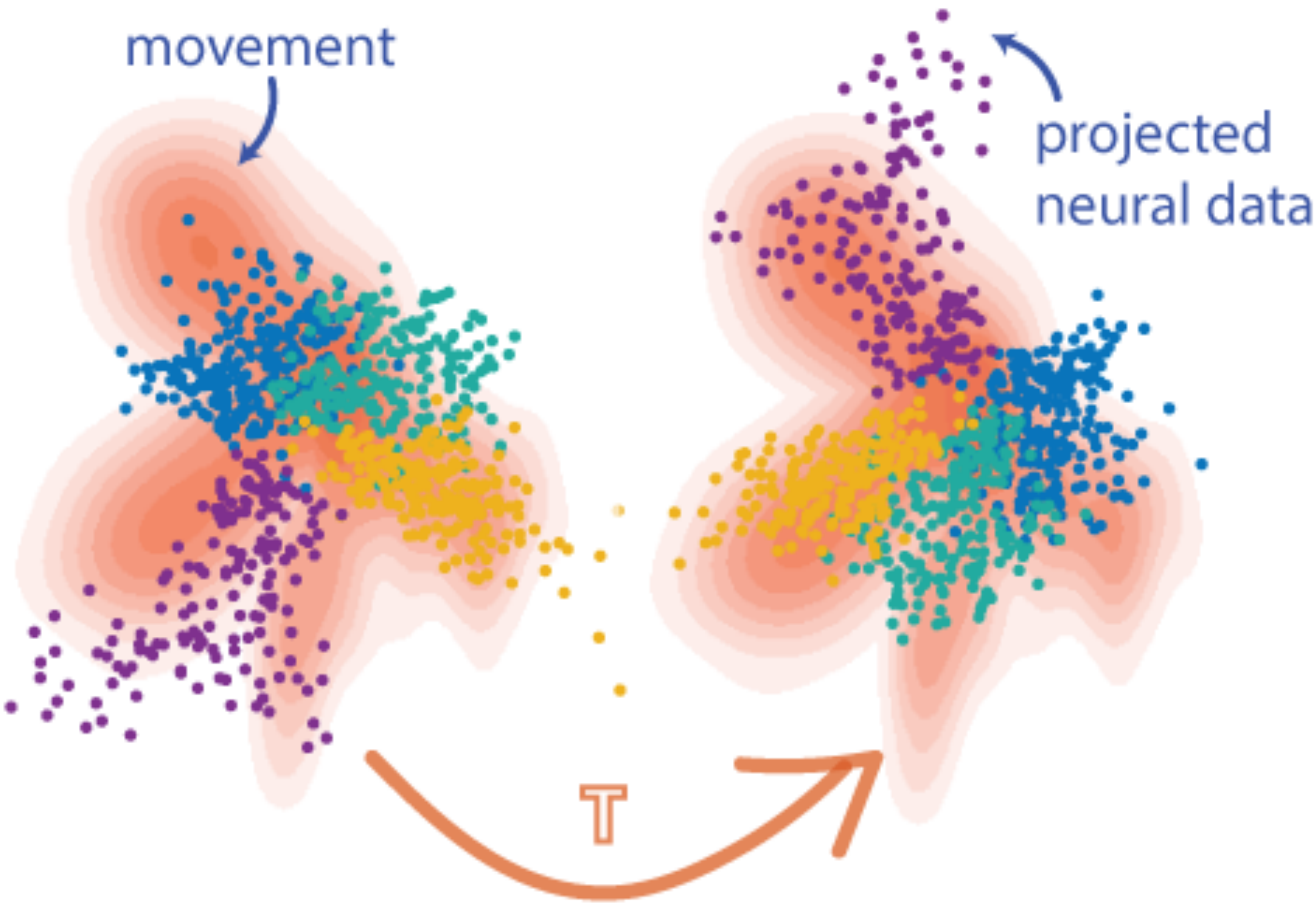
decoding results



increasing the population size



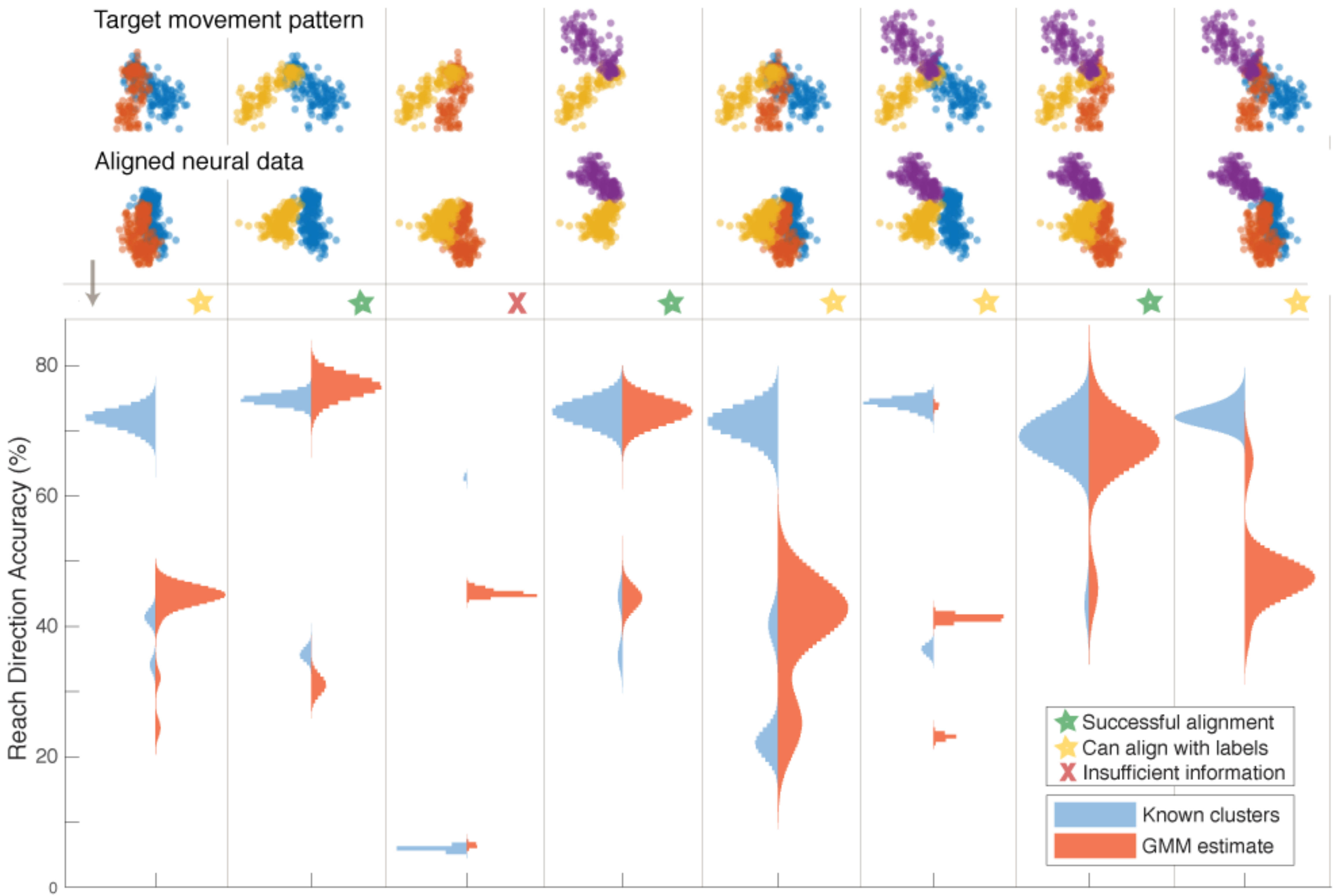
alignment with optimal transport



Lee et al., 2019

https://github.com/siplab-gt/hiwa-matlab

alignment with optimal transport



summary and references

overview of low-dimensional models

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

example: movement decoding

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

acknowledgements

Max Dabagia (Georgia Tech, NerDS Lab)
John Lee (Georgia Tech)
Chris Rozell (Georgia Tech)
Mohammad Gheshlagi Azar (DeepMind)
Konrad Kording (UPenn)
Lee Miller (Northwestern)

code / tutorials

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

Hierarchical Optimal Transport

- https://github.com/siplab-gt/hiwa-matlab

thank you!

(web)
dyerlab.gatech.edu
github.com/nerdslab