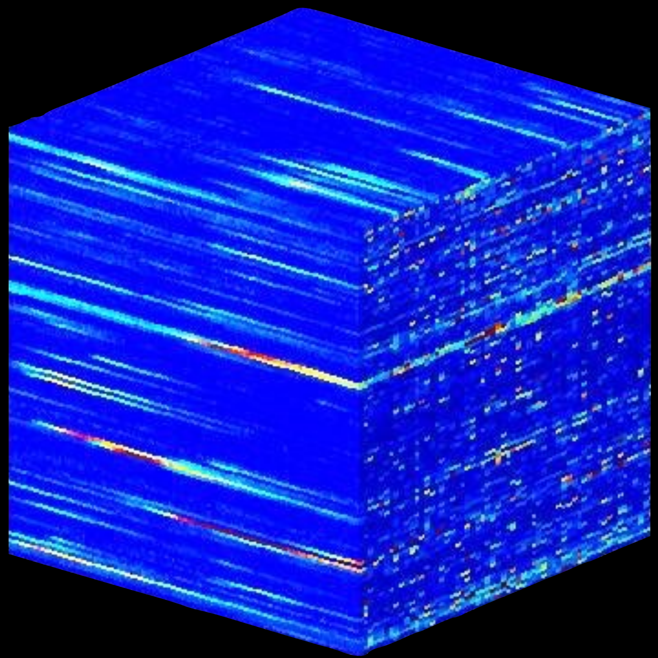
Multiway Tensor Analysis with Neuroscience Applications

Gal Mishne
Halıcıoğlu Data Science Institute, UCSD
January 2020

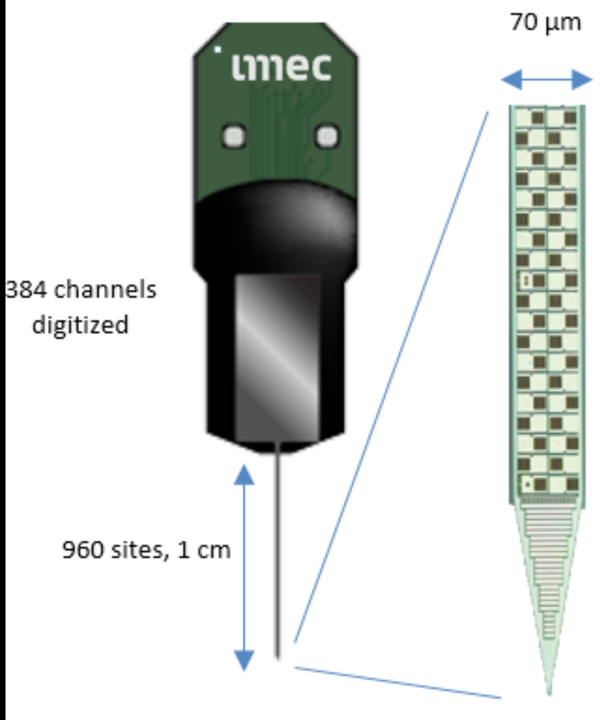


Outline

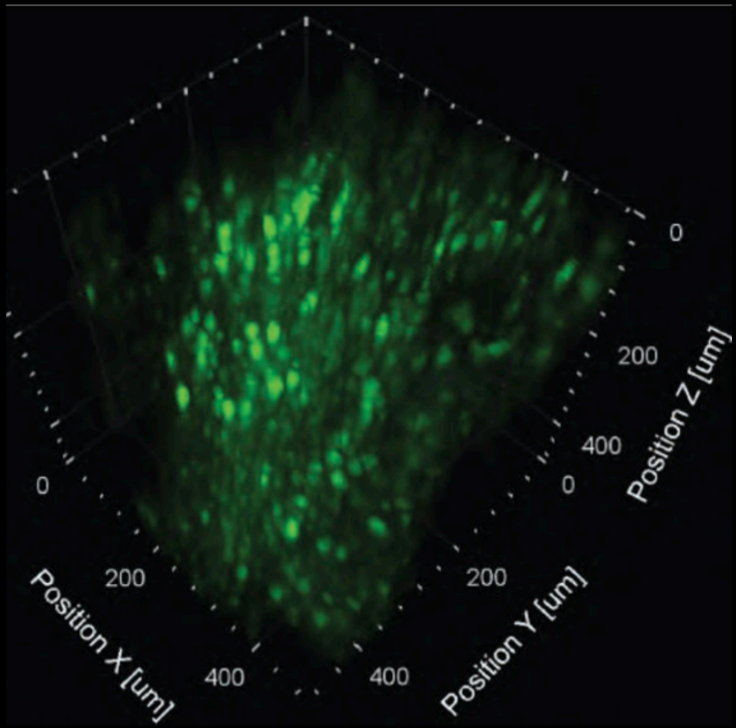
- **Background**
- Hierarchical Coupled Geometry Analysis
- Multi-step embedding of brain dynamics
- Visualizing the PHATE of neural networks

Neuro-data-science

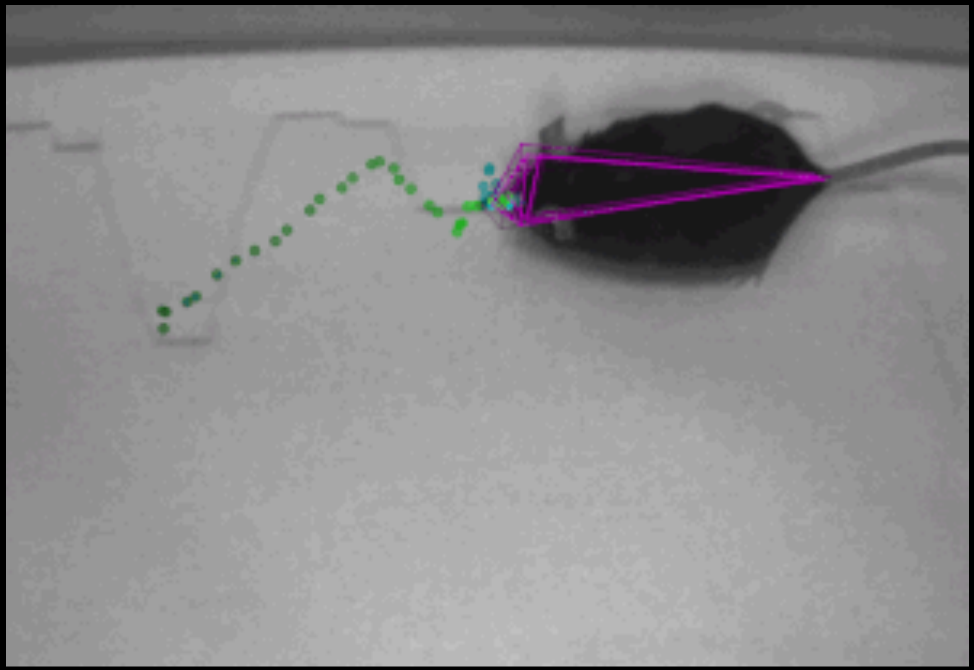
- Data explosion in neuroscience



Neuropixel probes
hundreds of neurons
across multiple brain regions



Volumetric calcium imaging
thousands of neurons
across multiple cortical layers
Source: [Prevedel et al. 2016]

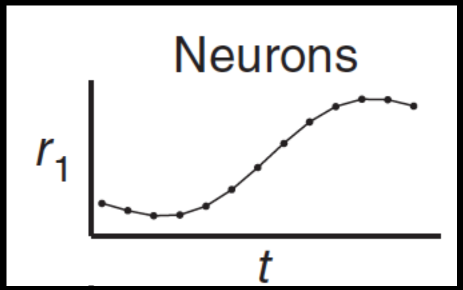


Behavioral videos of animal models
[Deeplabcut, 2019]

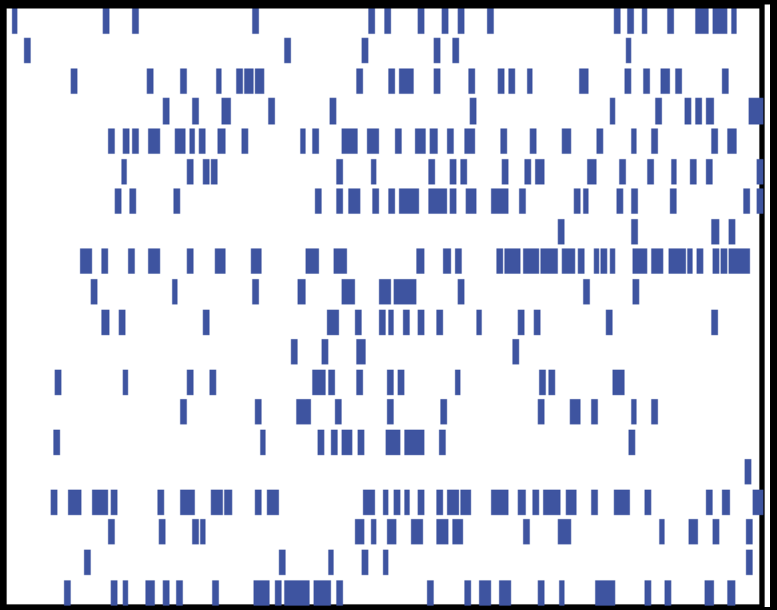
Neural data analysis

Shift from single neuron to population-level hypotheses, recordings and analysis

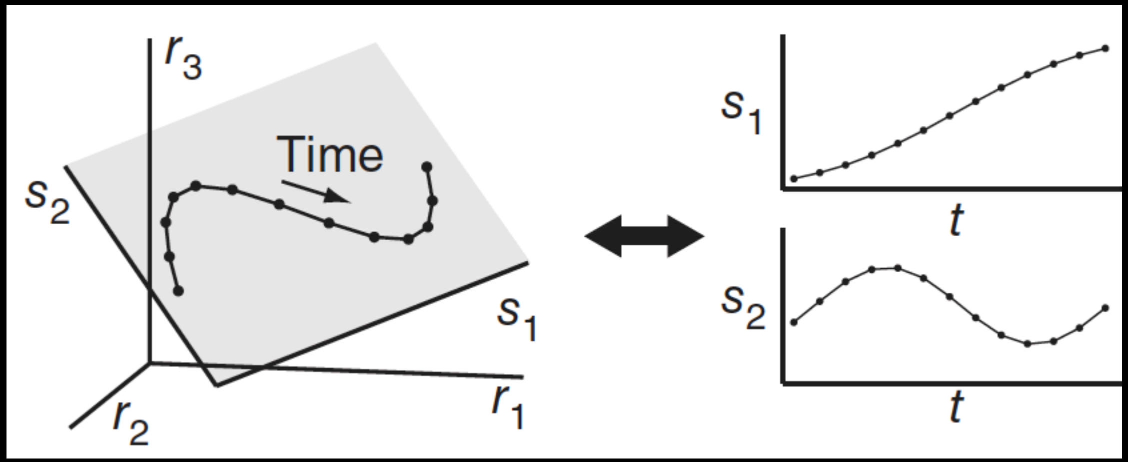
single neuron



multiple neurons



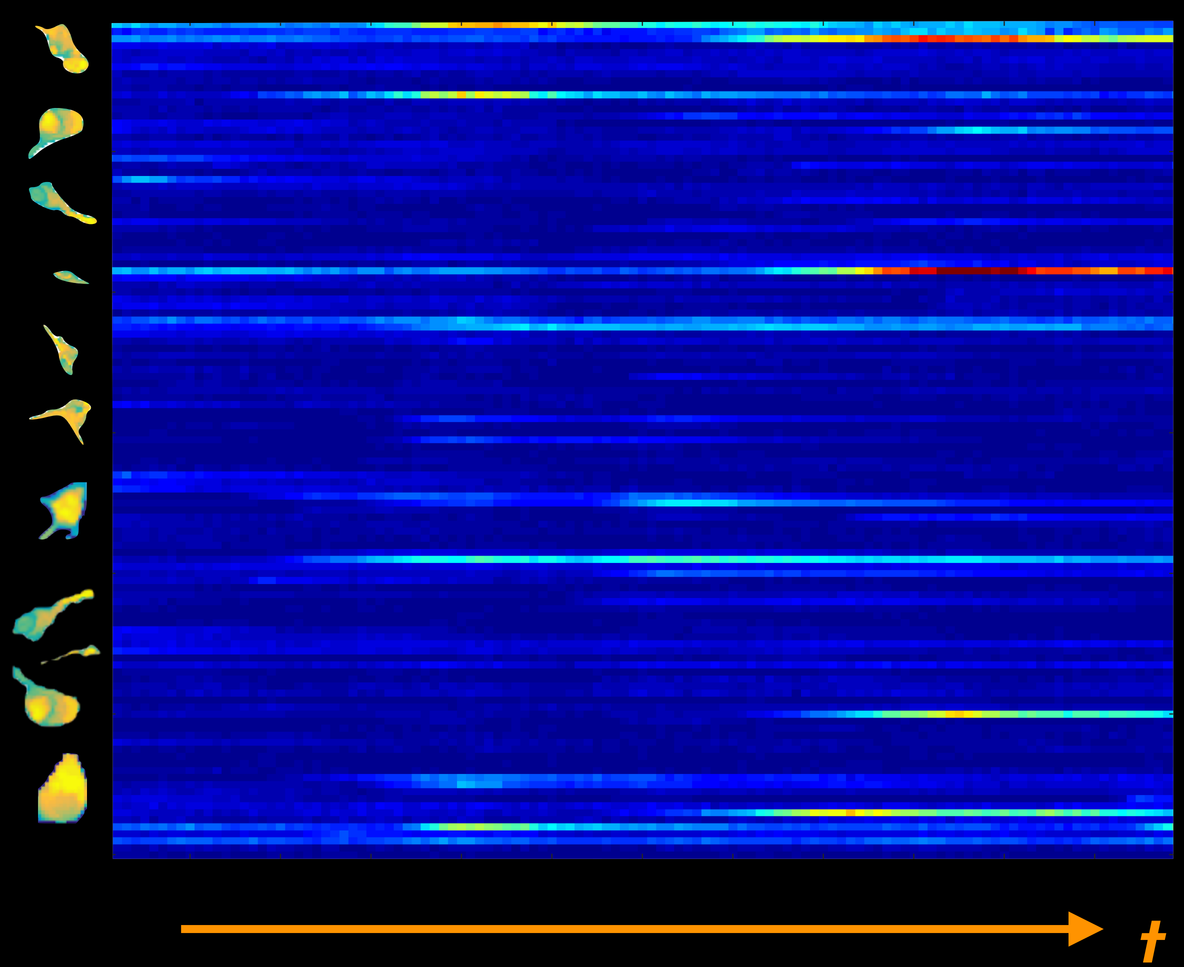
state space **latent variables**



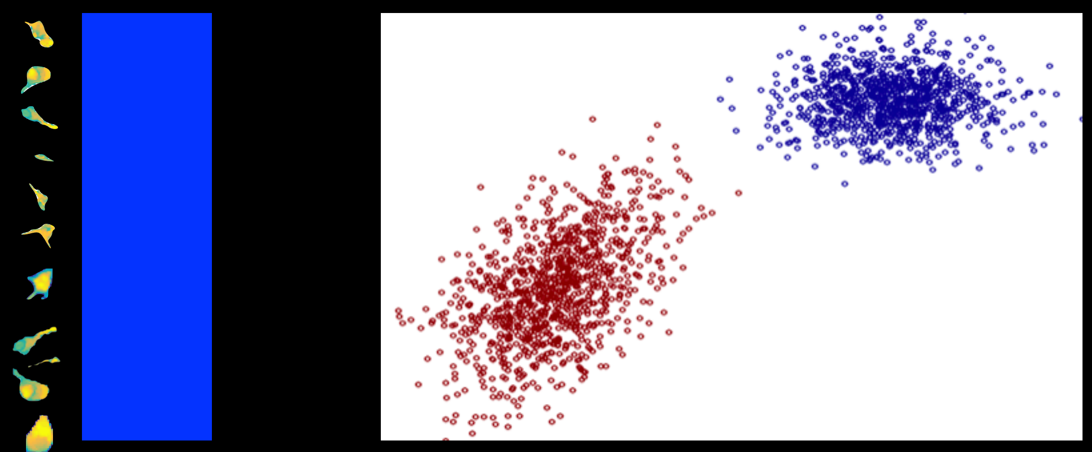
How do we find these latent variables?

[Cunningham and Yu, 2014]

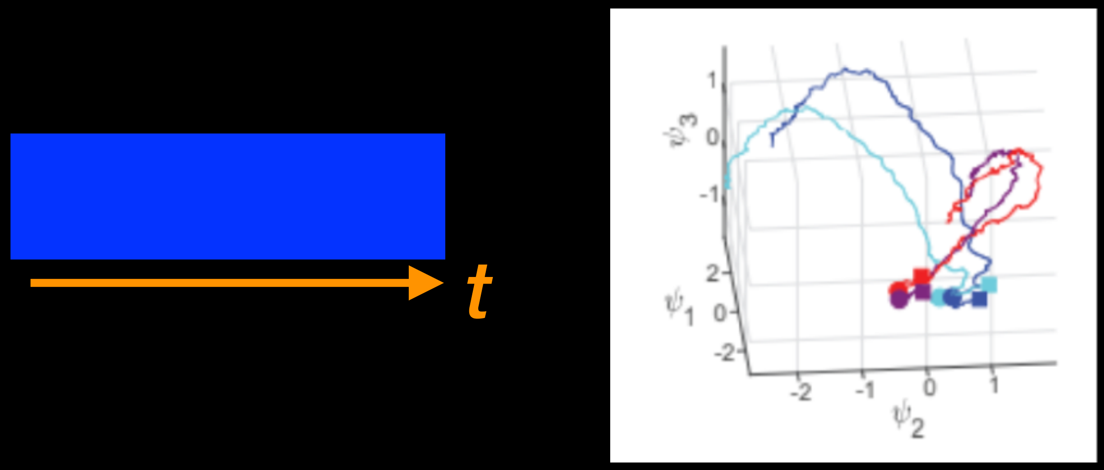
Dimensionality reduction for neural data analysis



Embedding neurons = network structure



Embedding time = dynamics



High-dimensional data analysis

Manifold learning

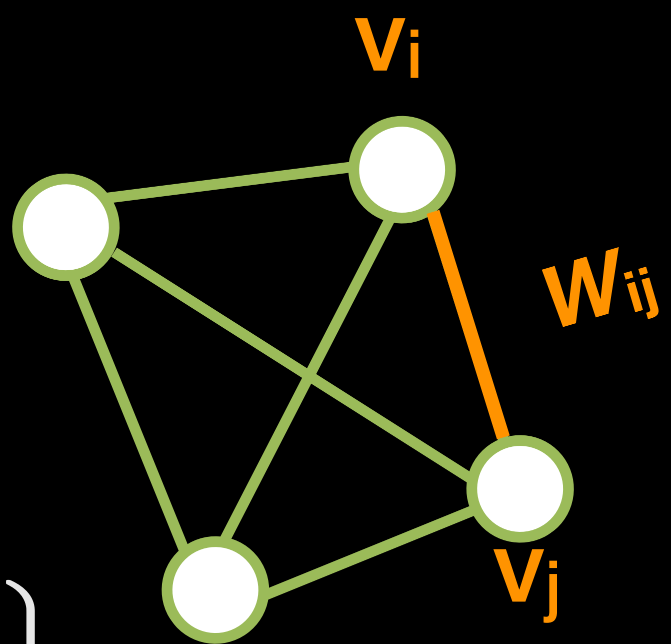
- Learn manifold from data.
- Dimensionality reduction:
  Non-linear **representation** of low-dimensional manifold.
- Preserve **geometric** properties.
- Data visualization, exploration, pre-processing (de-noising), compression, machine learning

[Tenenbaum et al., 2000,
Roweis and Saul, 2000,
Belkin and Niyogi, 2001,
Donoho and Grimes, 2002,
Coifman and Lafon, 2004]

Diffusion Maps

- Undirected weighted graph on *n* nodes:

G={V,E,W}



- W - weighted adjacency matrix

$$\mathbf{W}[i,j] = w_{ij} = \exp\left\{-\frac{d^2(i,j)}{\sigma^2}\right\}$$

- Degree matrix

$$\mathbf{D}_{ii} = d_i = \sum_j \mathbf{W}[i,j]$$

[Coifman and Lafon, 2004]

Diffusion Maps

- Construct weighted graph from *n* datapoint
- Eigendecomposition of random-walk matrix

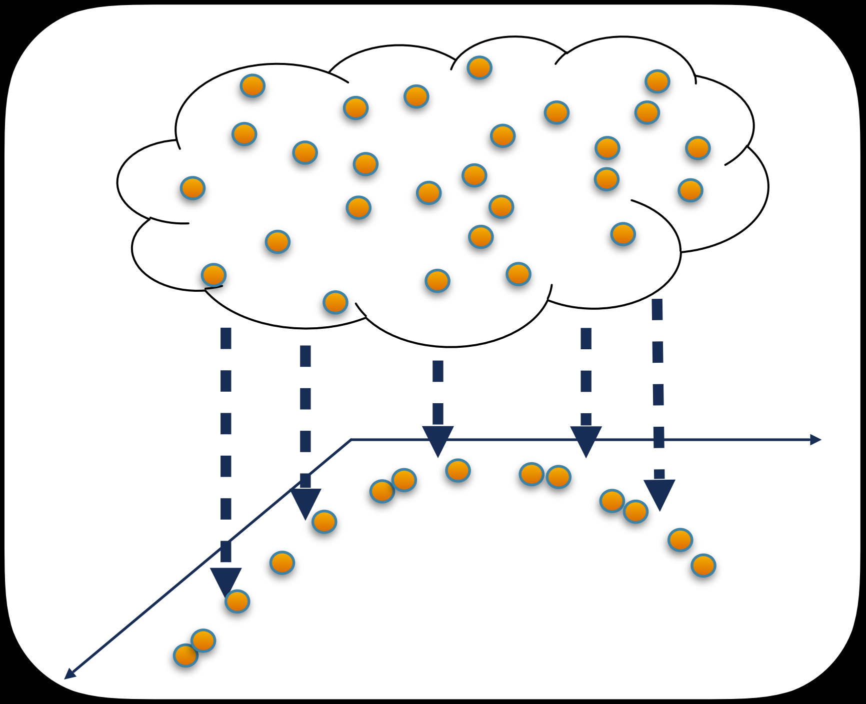
$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$$

$$1 = \lambda_0 \geq \lambda_1 \geq \lambda_2 ... \qquad \{\phi_\ell\}_{\ell=1}^{d} \quad \{\psi_\ell\}_{\ell=1}^{d}$$

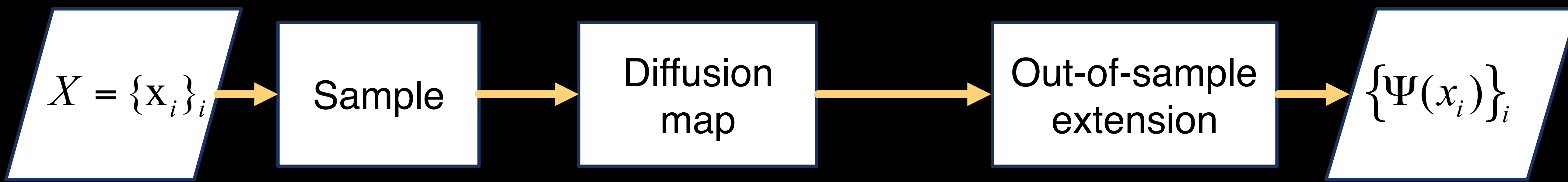
- Nonlinear embedding

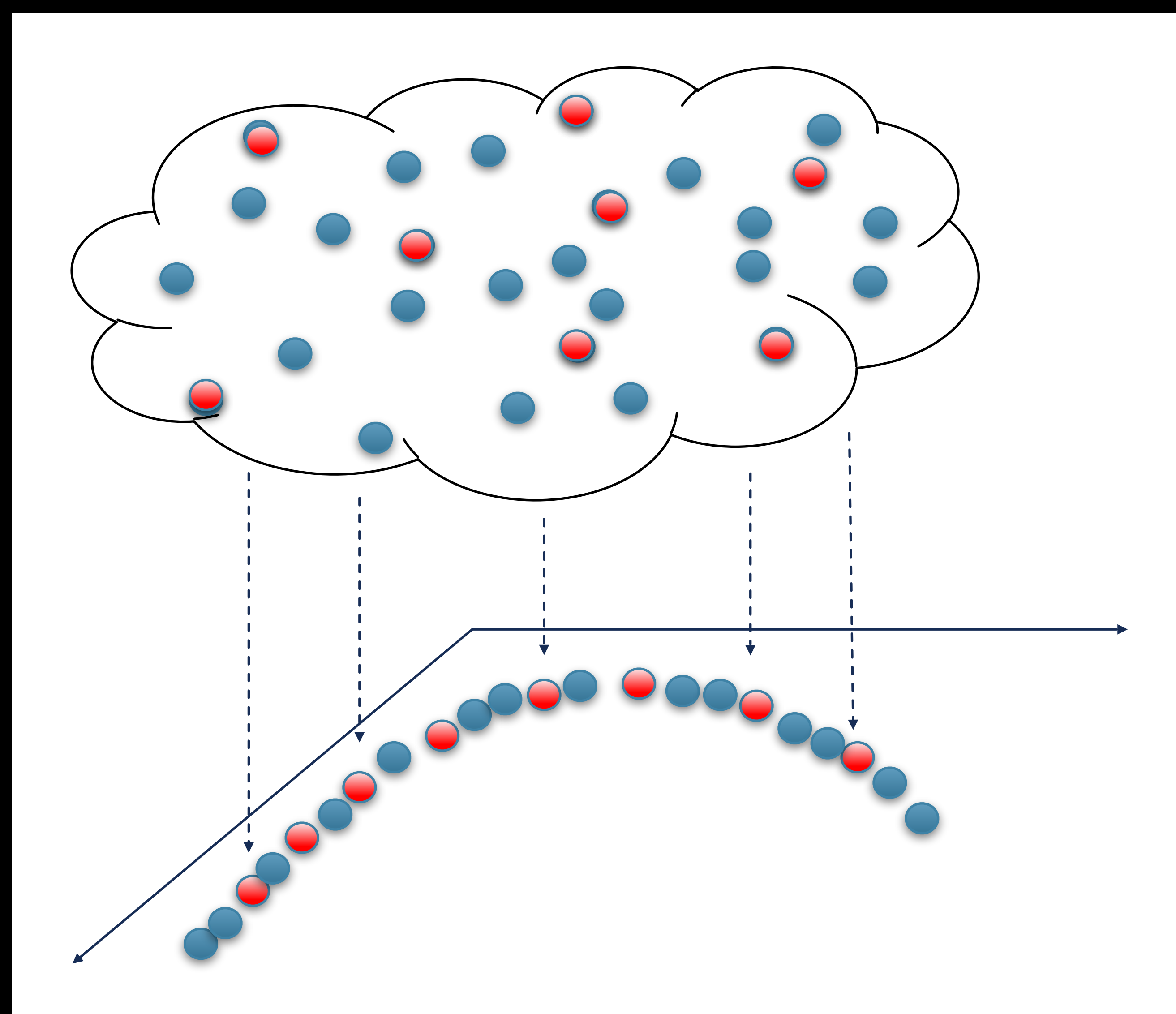
$$\Psi_t : x_i \rightarrow \left[\lambda_1^t \psi_1(i), \cdots, \lambda_\ell^t \psi_\ell(i)\right]^T$$

- There is no linear mapping!

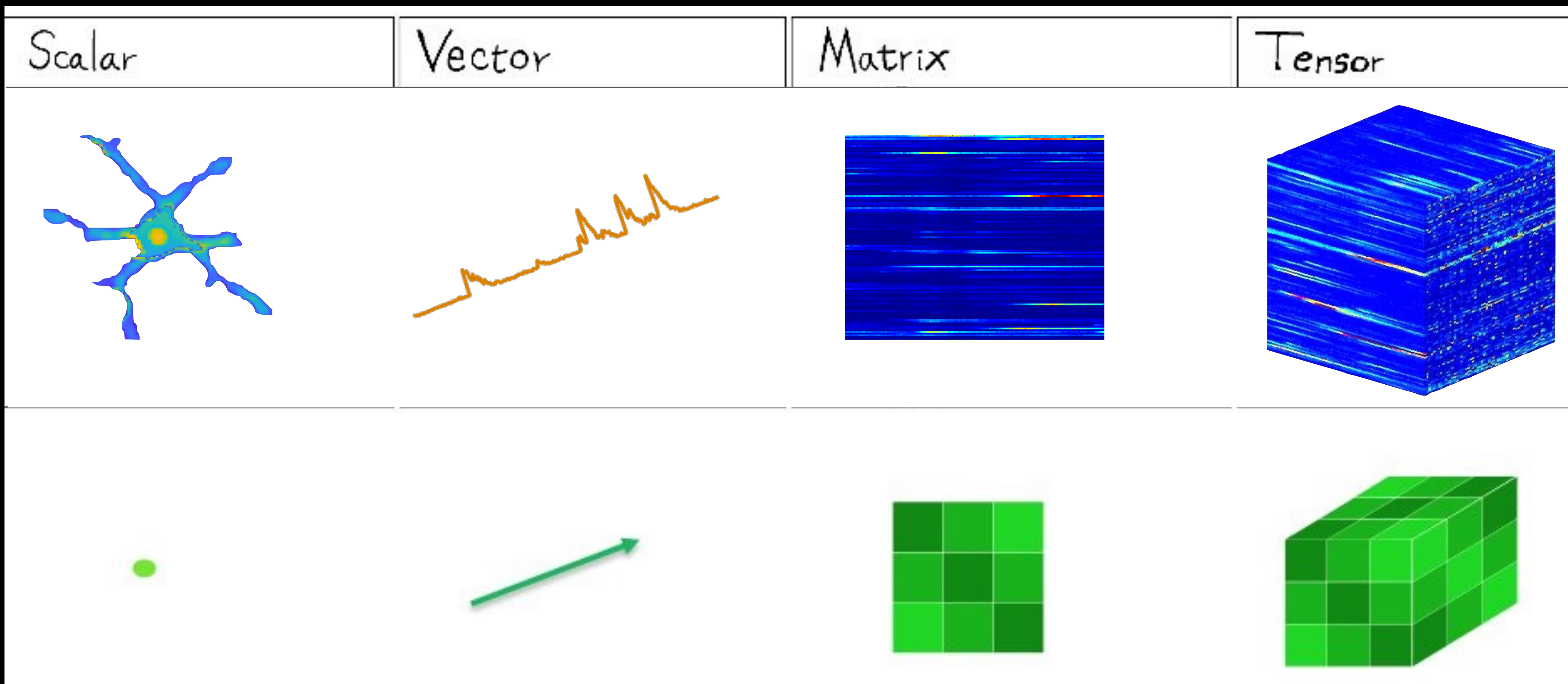


Out-of-sample Function Extension



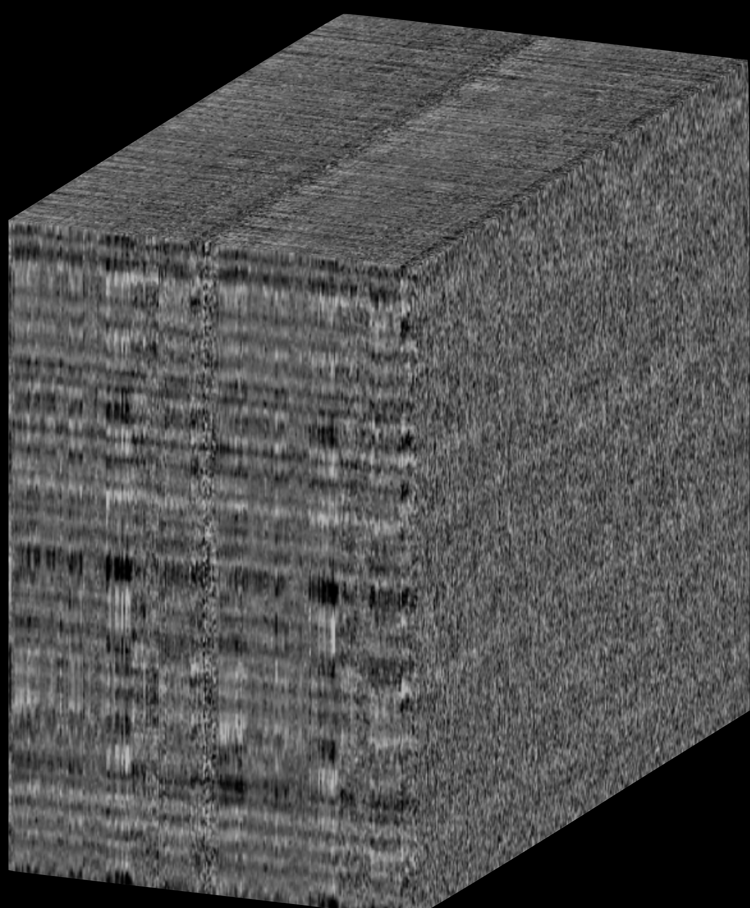


Tensors - the neuro version

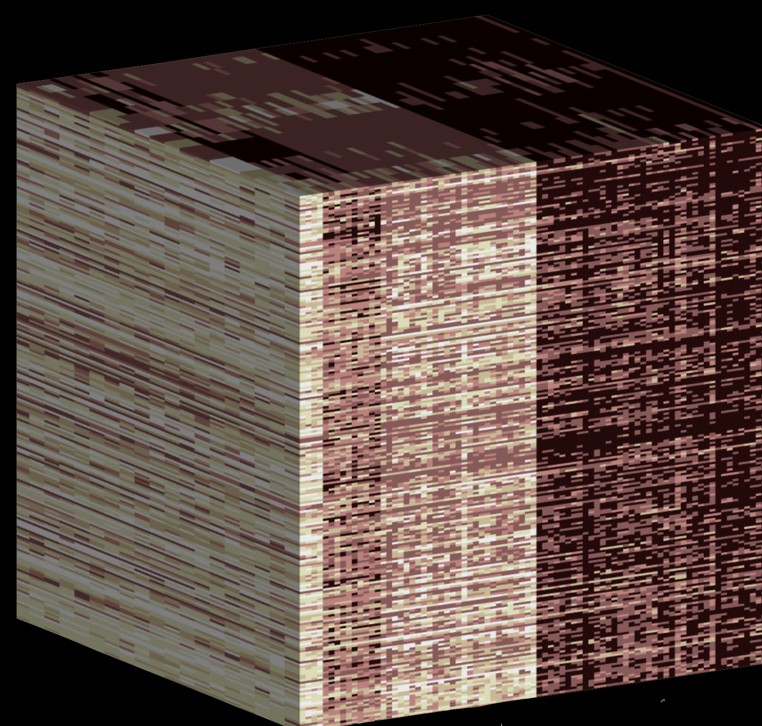


[Adapted from: Anima Anandkumar]

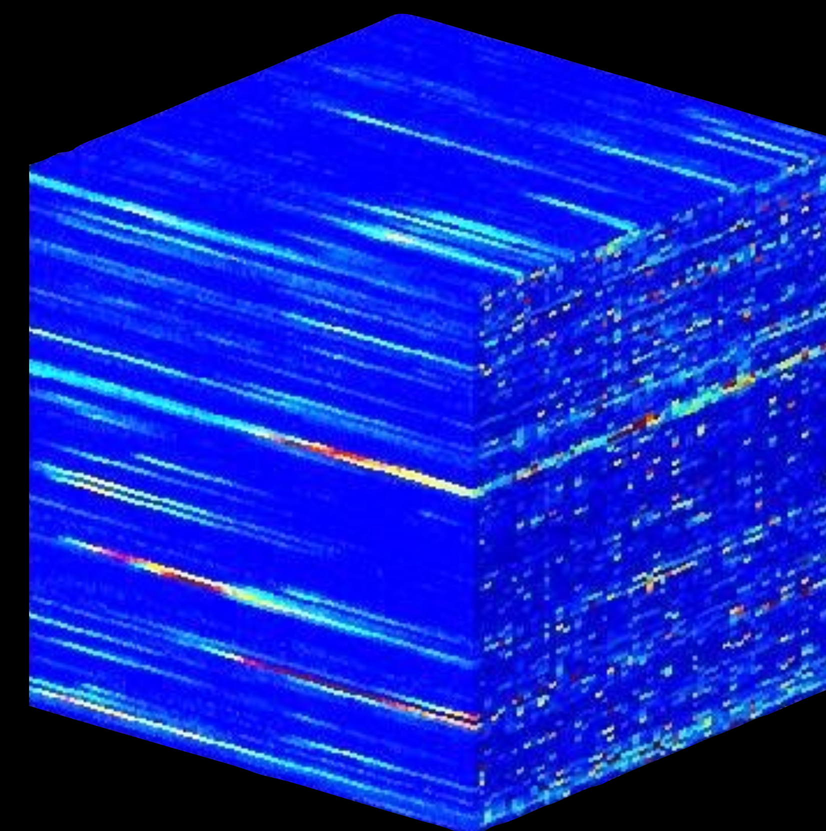
Multi-way Tensors



task-fMRI



Psychological questionnaires



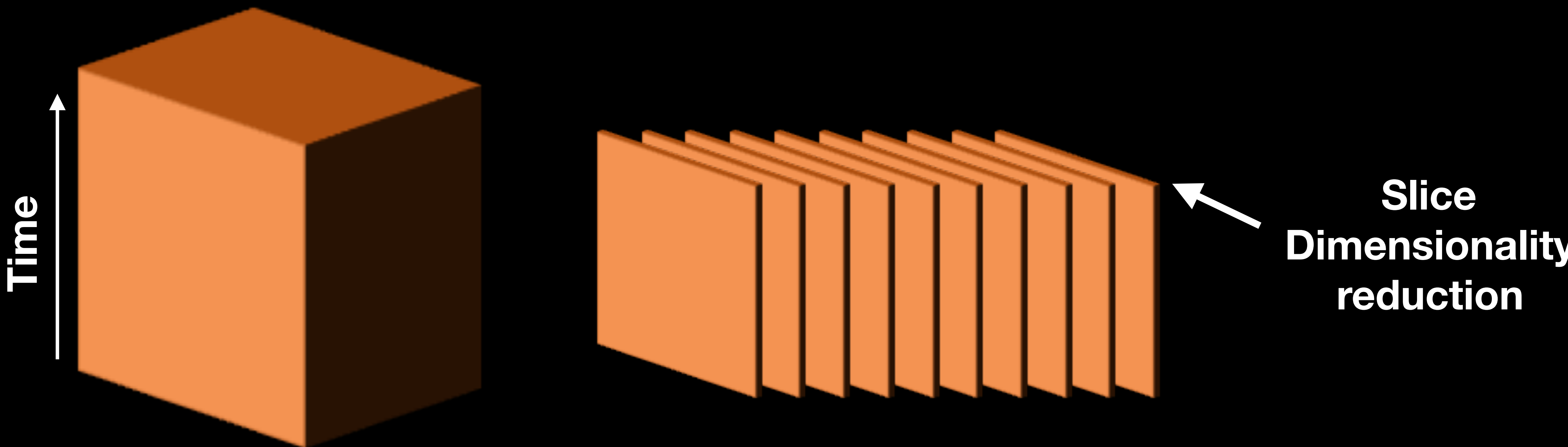
Multi-trial neuronal activity

Outline

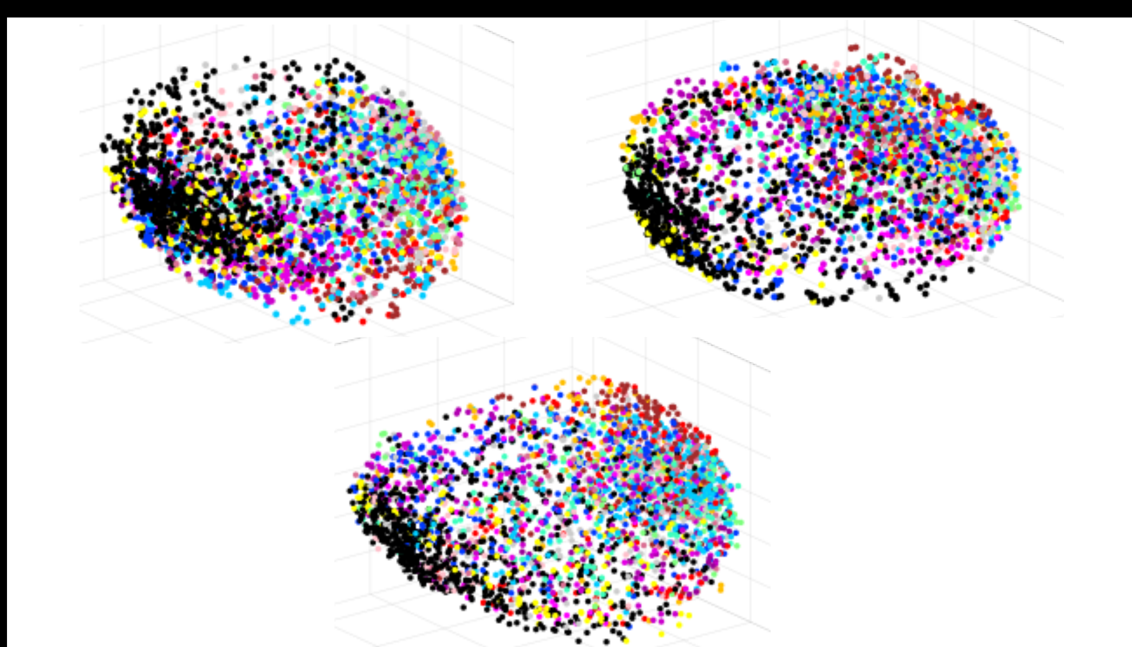
- Background
- **Multi-step embedding of brain dynamics**
- Hierarchical Coupled Geometry Analysis
- Visualizing the PHATE of neural networks

[Gao, Mishne, Scheinost, CCN-2018, IPMI-2019]

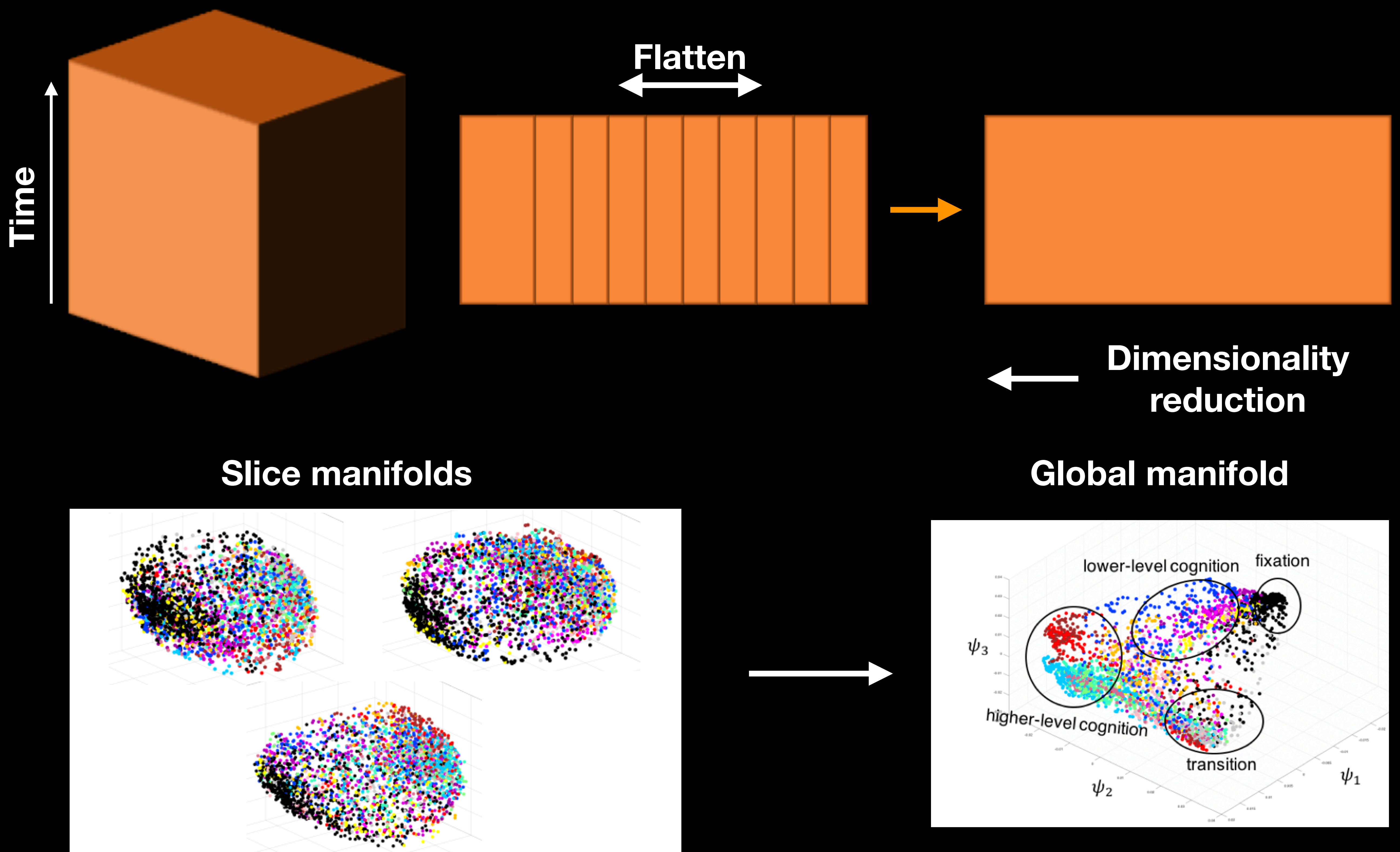
Multi-step Embedding



Slice manifolds

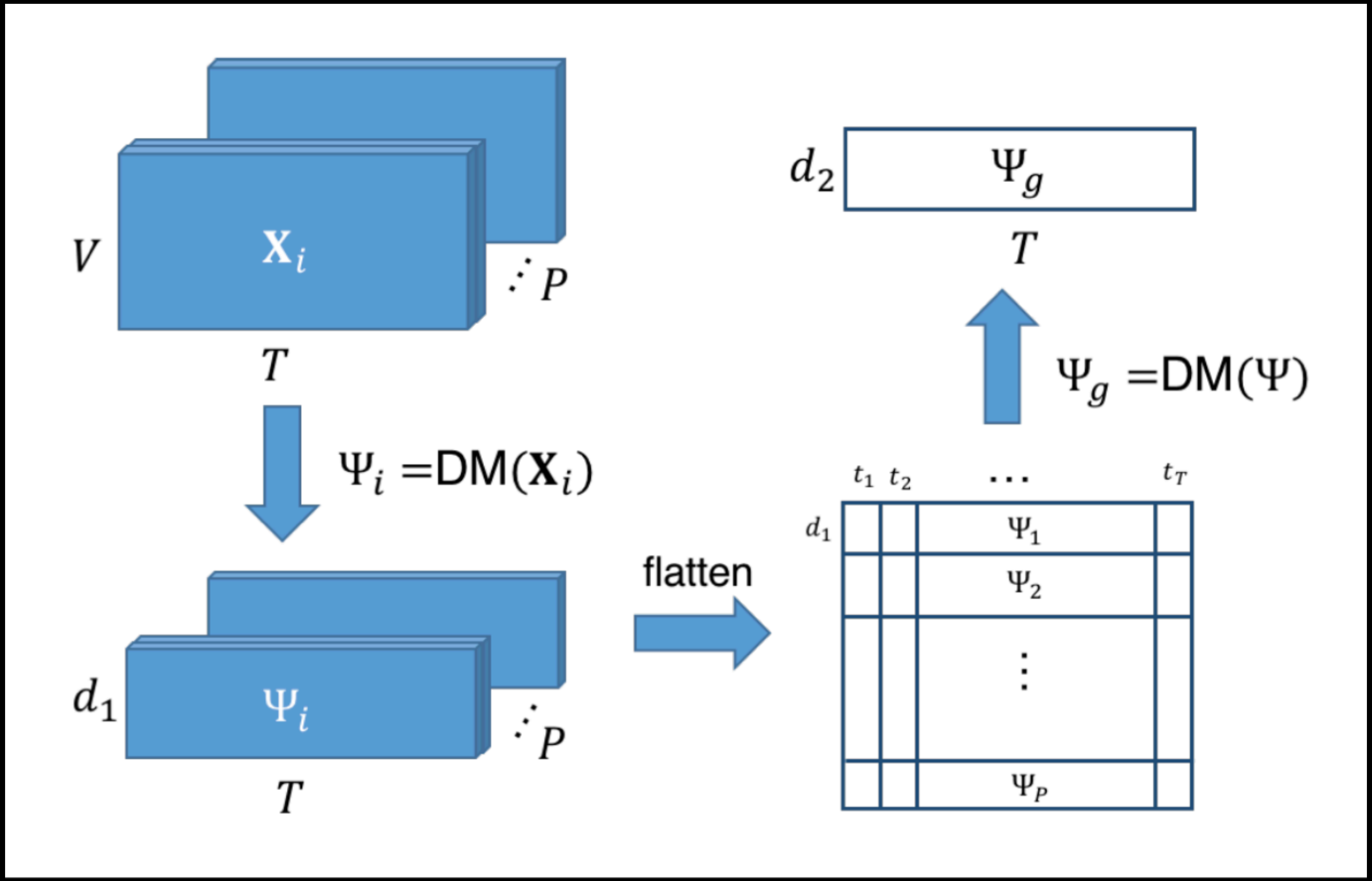


Multi-step Embedding



Temporal embedding with 2sDM

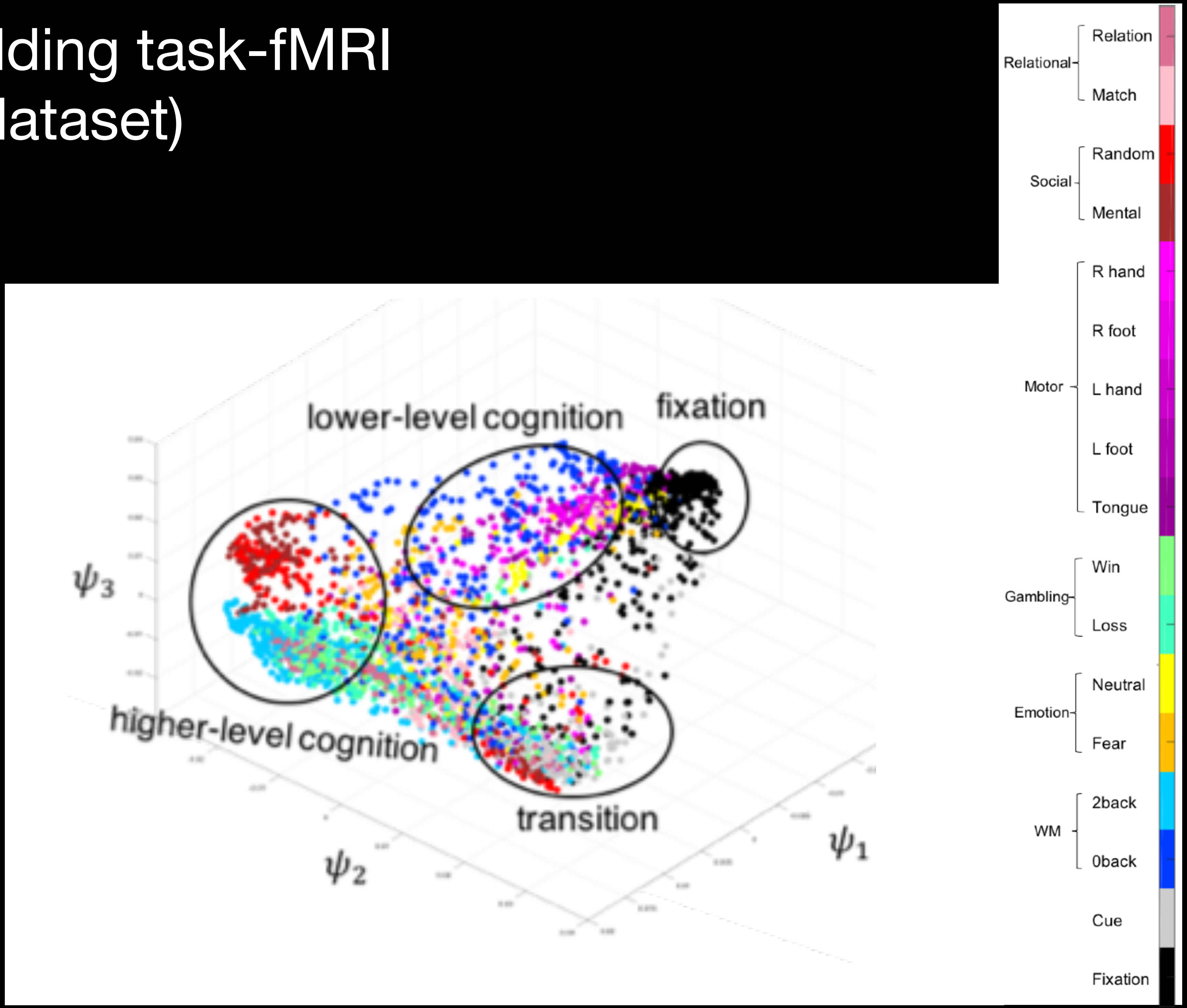
- Hierarchical manifold learning framework
- Time-synchronized multi-individual fMRI BOLD time series



$$||\mathbf{\Psi}^{(1)}(t) - \mathbf{\Psi}^{(1)}(s)||_2^2 = \sum_{i=1}^{M} D_t^2(\mathbf{X}_{i,t,\cdot}, \mathbf{X}_{i,s,\cdot})$$

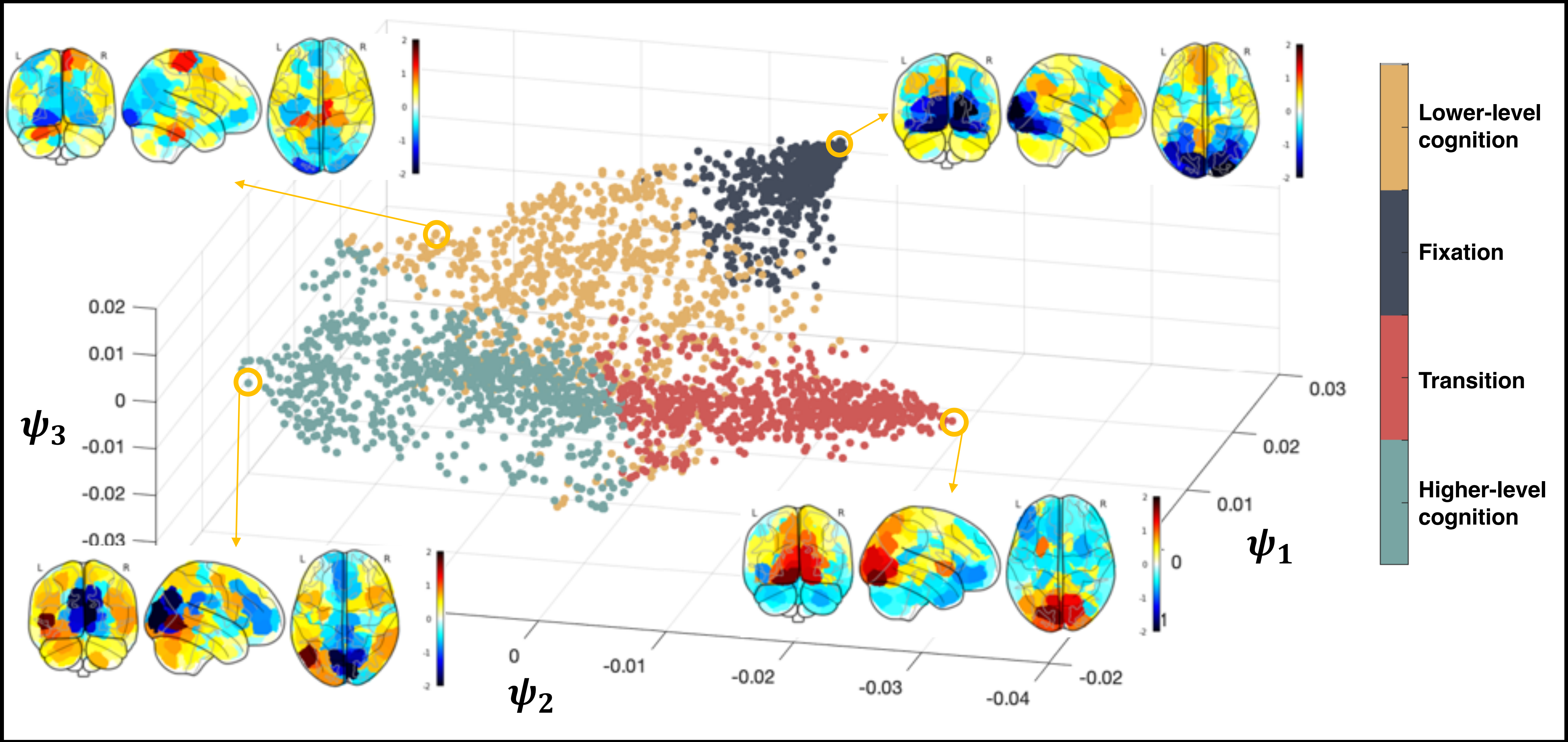
fMRI Temporal Manifolds

- Embedding task-fMRI (HCP dataset)



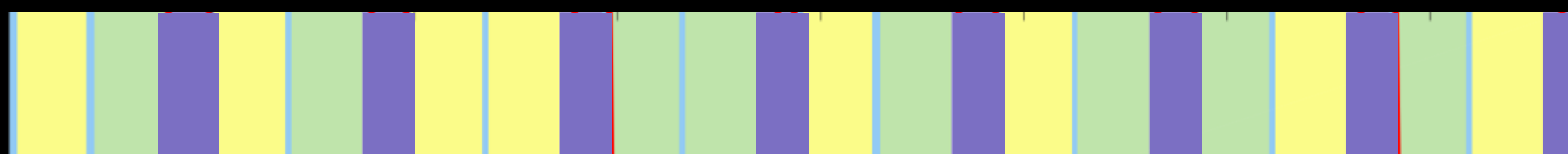
Brain States

- Clustering into 4 states

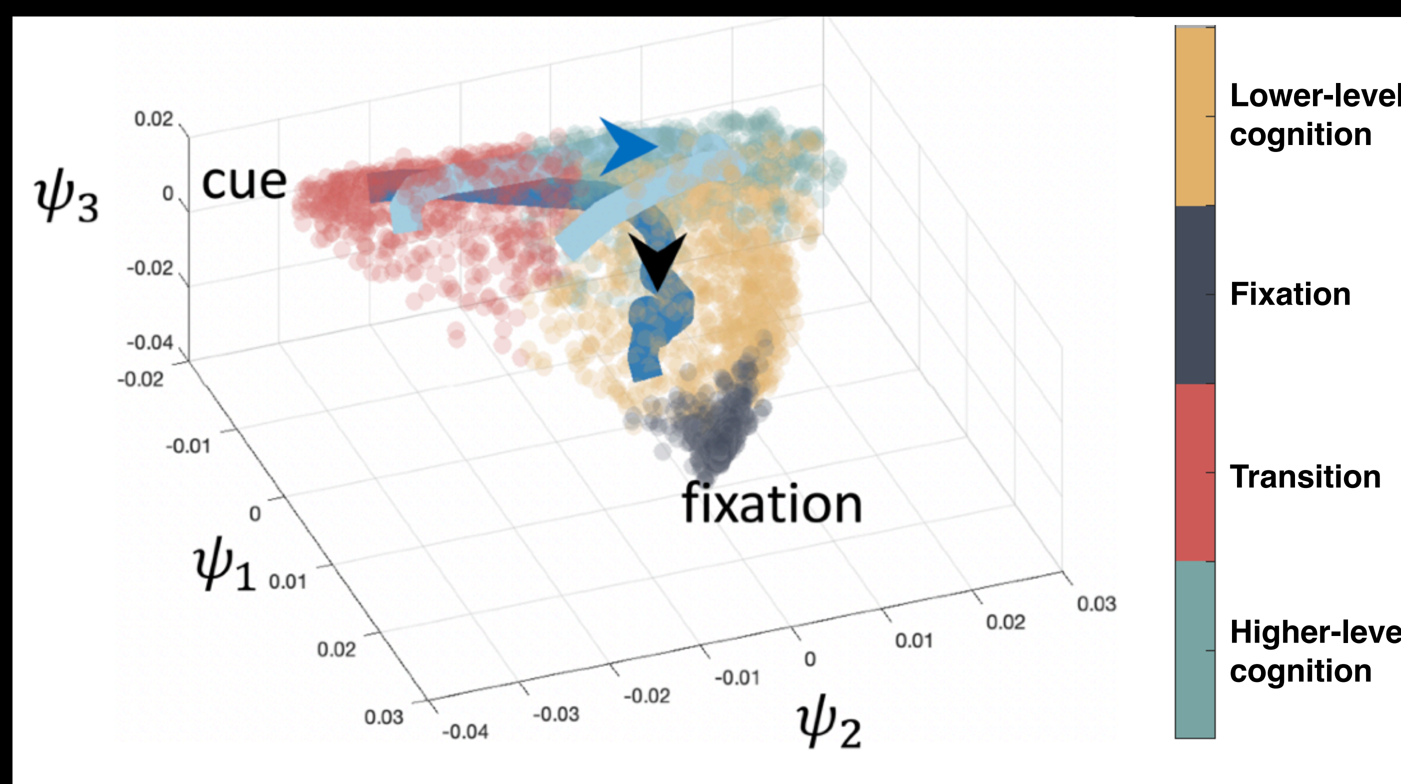


Task-fMRI Dynamics

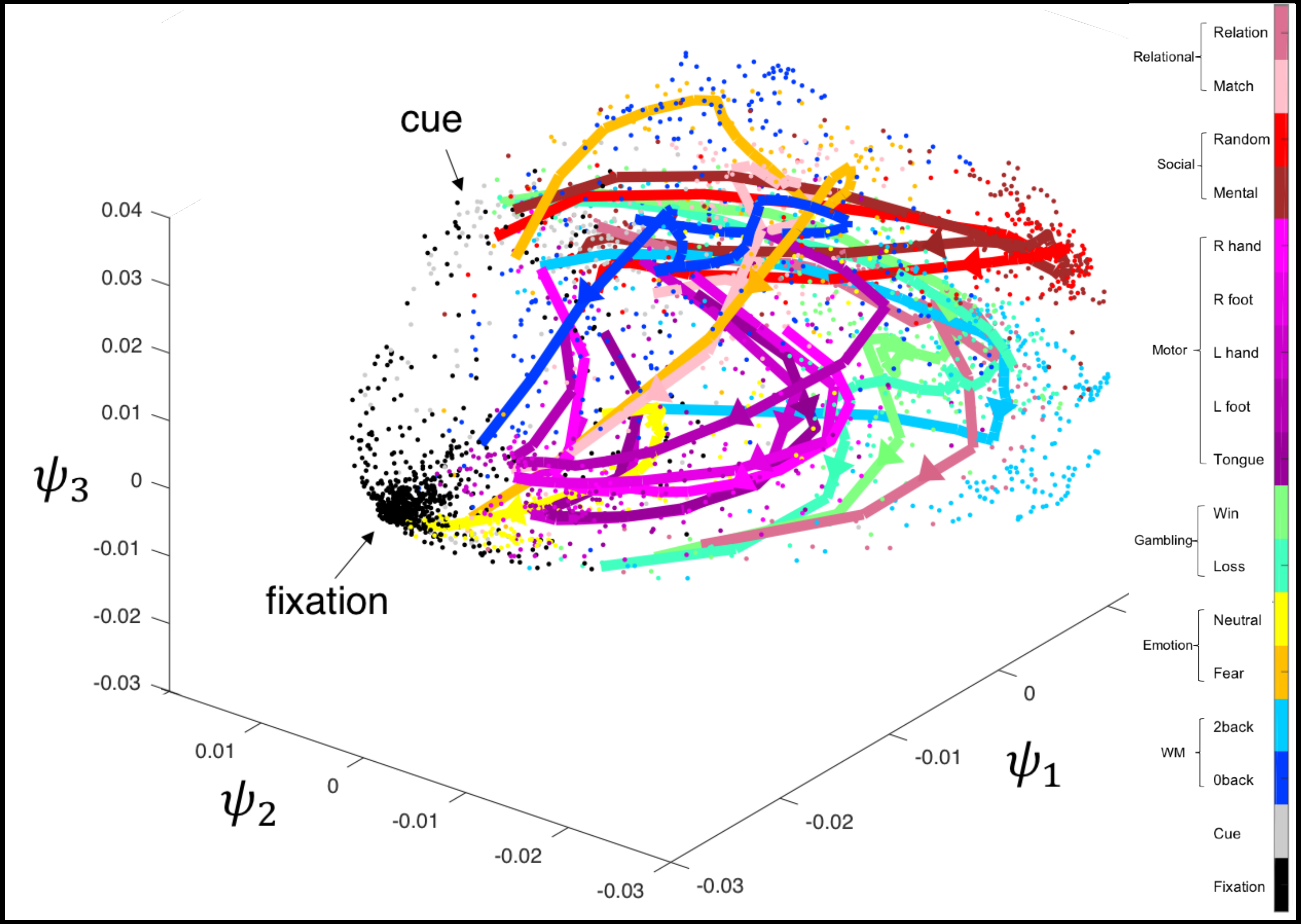
Task blocks: fixation cue 0-back 2-back



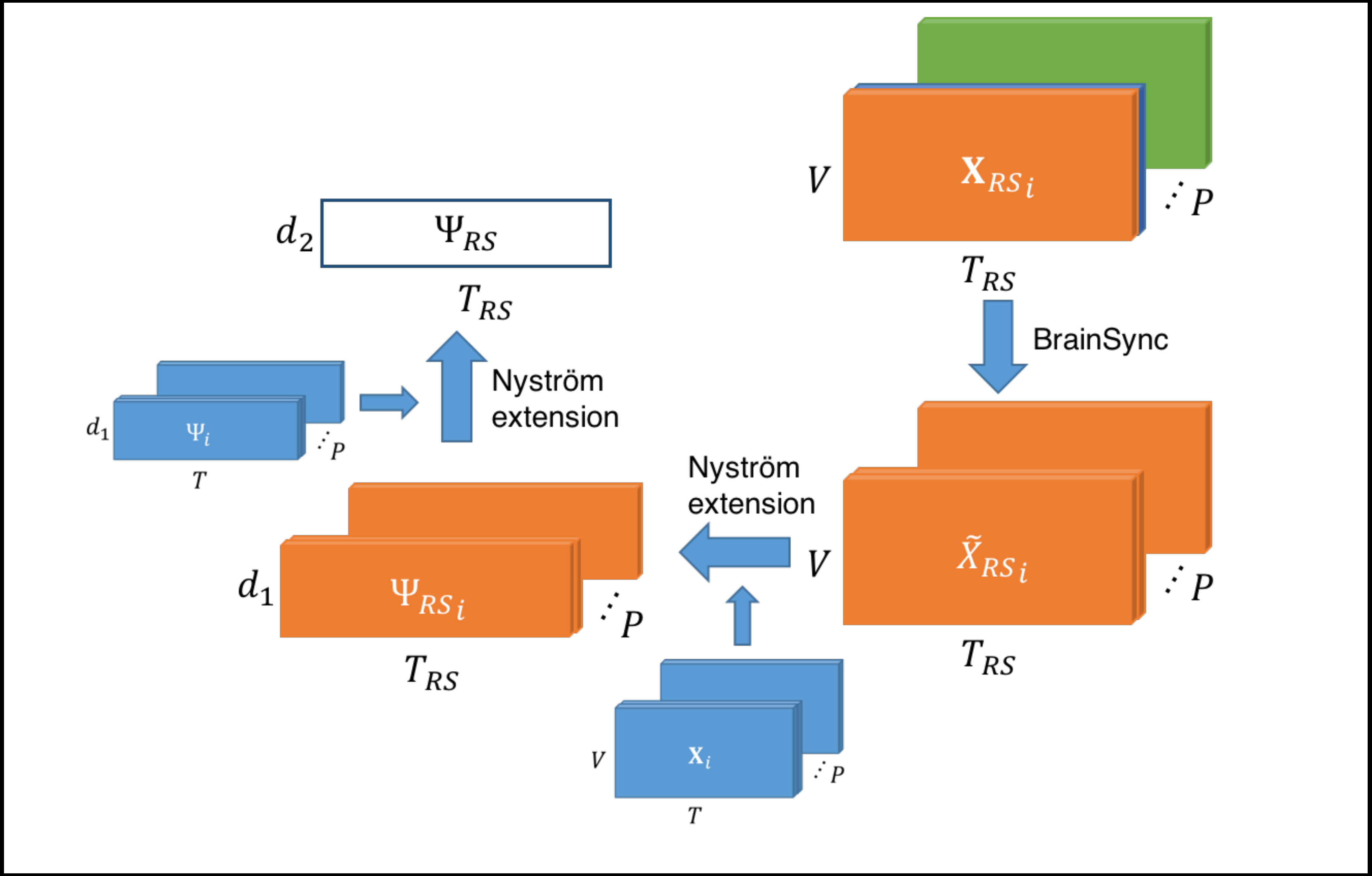
Time →



Task-fMRI Dynamics

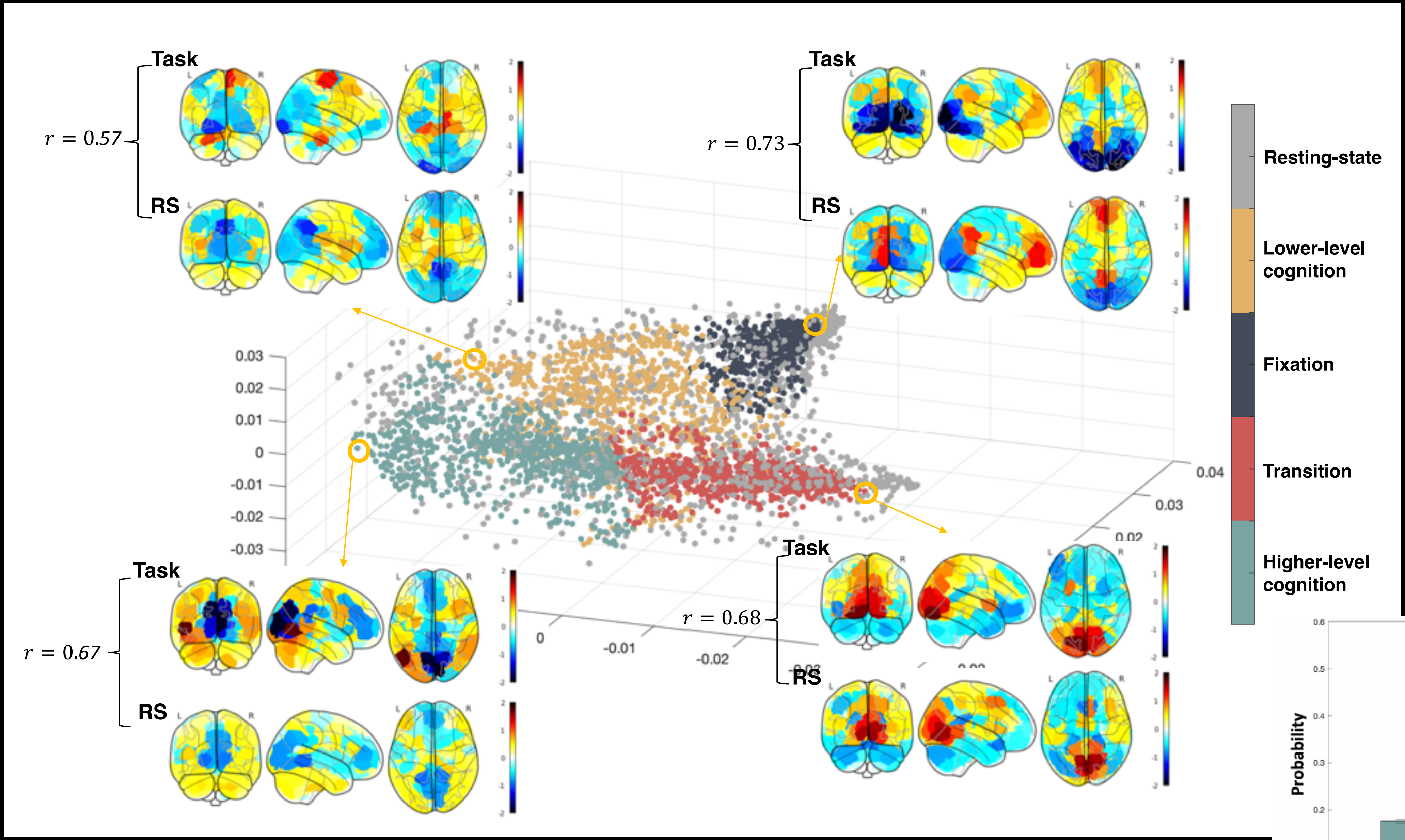


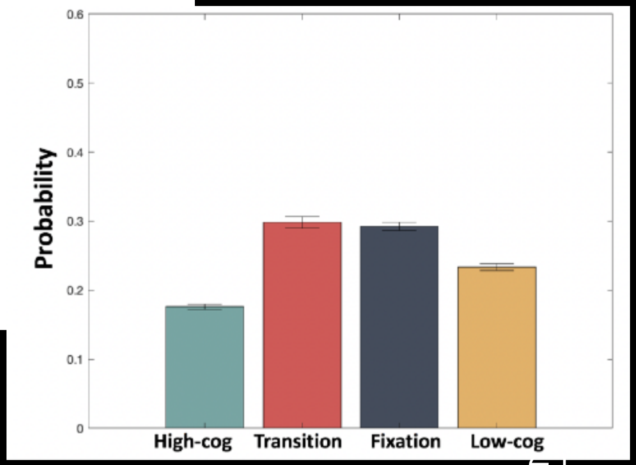
Out-of-sample Extension



Embedding resting-state

- Resting state is not single state, but a collection of multiple task-like states



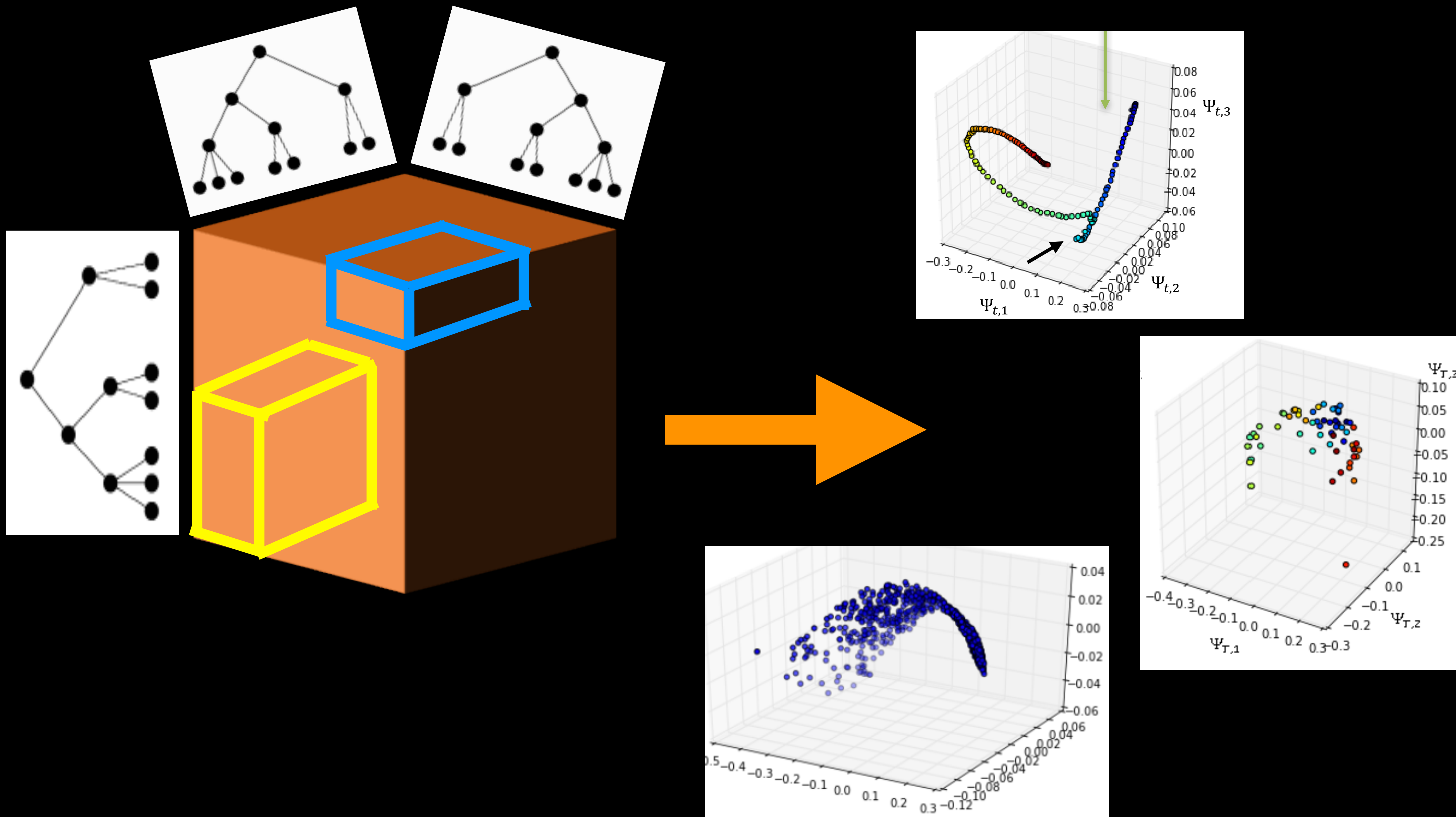


Outline

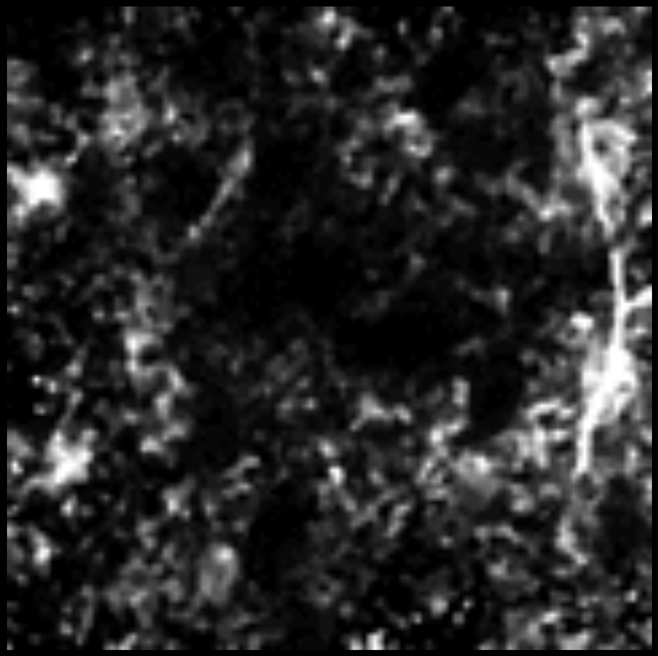
- Background
- Multi-step embedding of brain dynamics
- **Hierarchical Coupled Geometry Analysis**
- Visualizing the PHATE of neural networks

[Mishne, Talmon, Meir, Schiller, Lavzin, Dubin and Coifman, IEEE JSTSP 2016]

Hierarchical Coupled Geometry Analysis



Calcium imaging



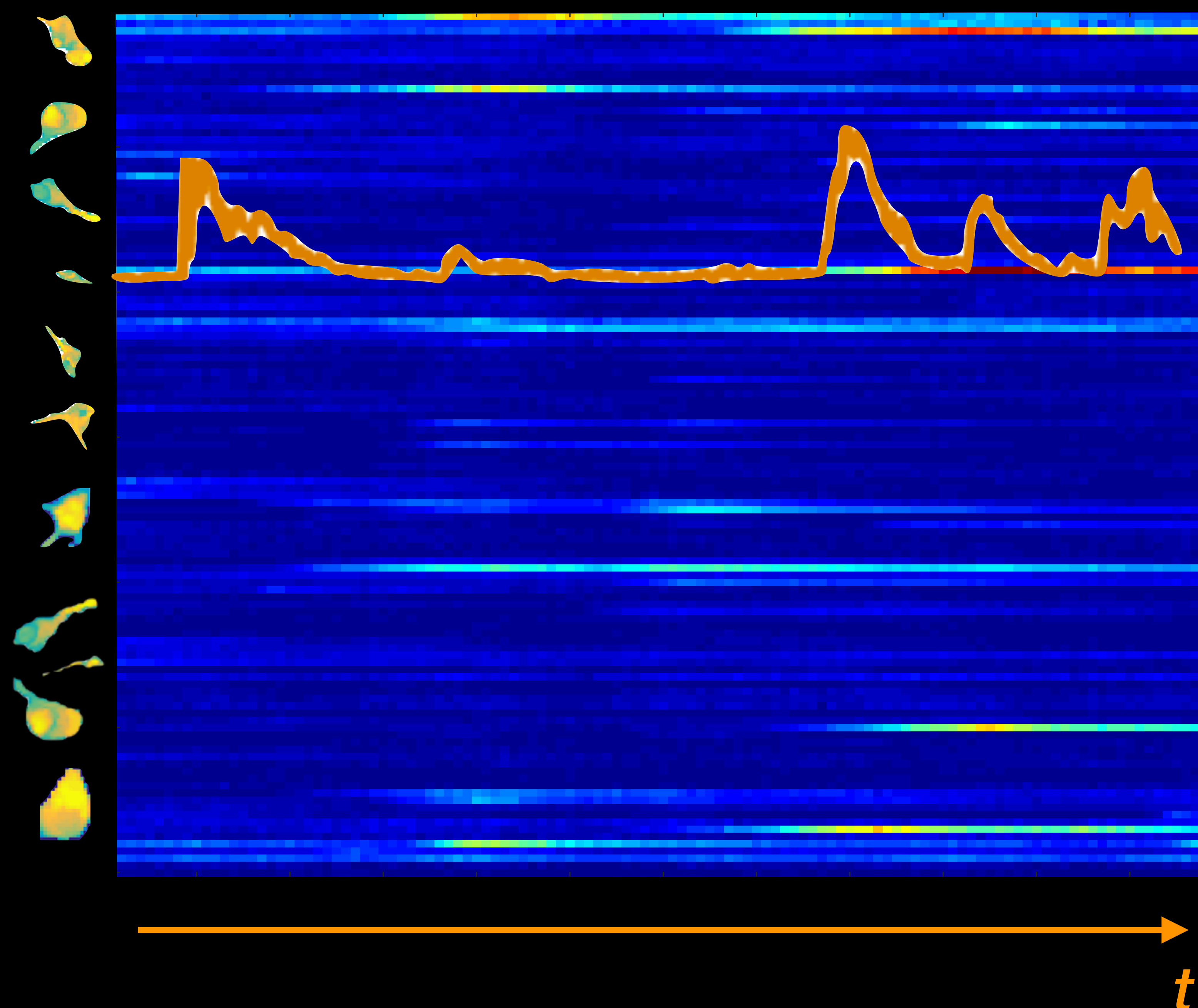
- Enables recording *in vivo* **hundreds** of neurons
- Tracking same population across days
- Sub-cellular spatial resolution
- Temporal resolution relevant to behavior
- **Challenge**: complex large-scale neuronal and behavioral datasets

[Schiller lab]

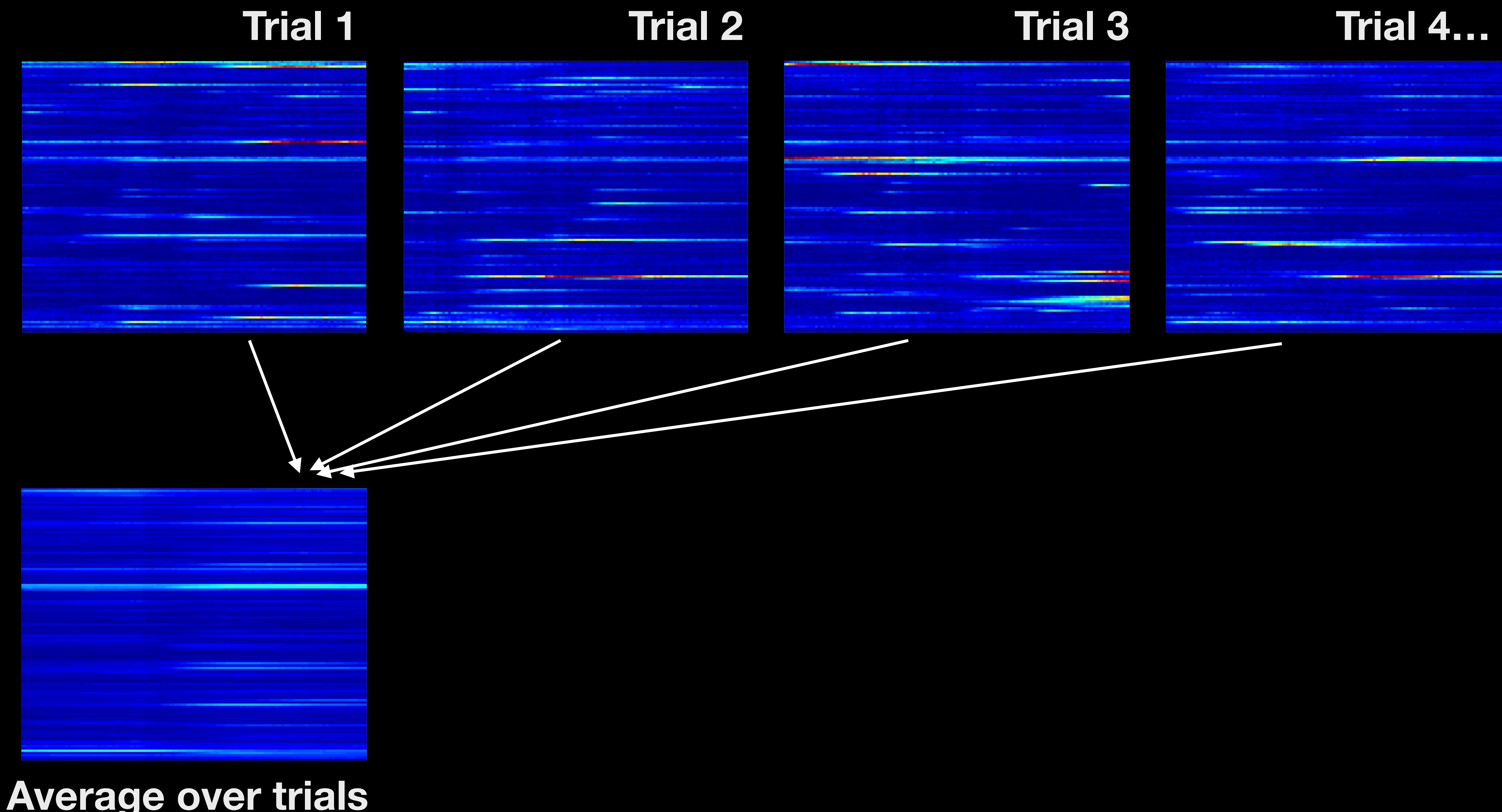
Neuronal Activity Analysis

Goals:

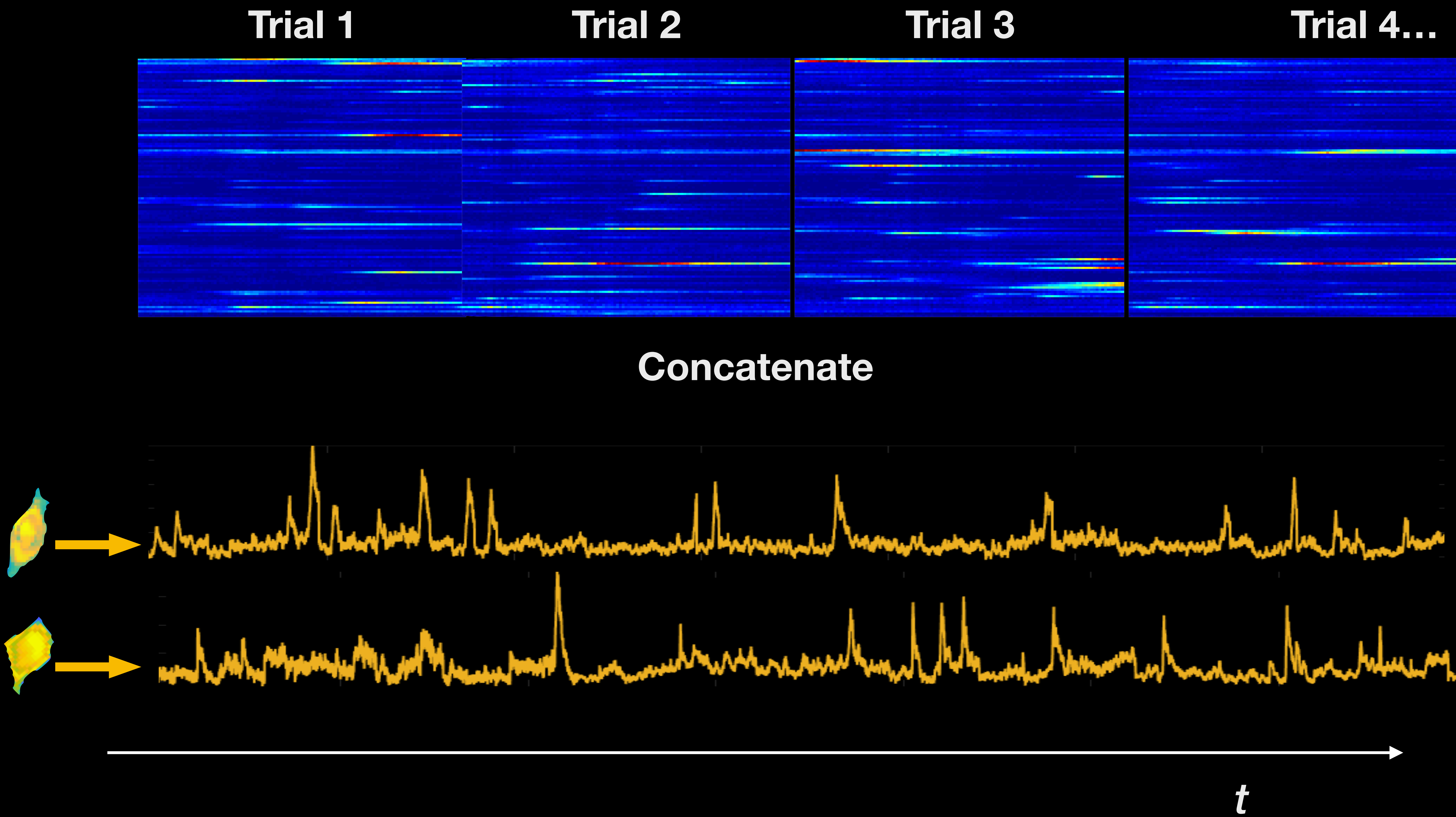
- **Network** of neurons
- Low-dimensional **dynamics**
- **Data-driven**
- **Unbiased**



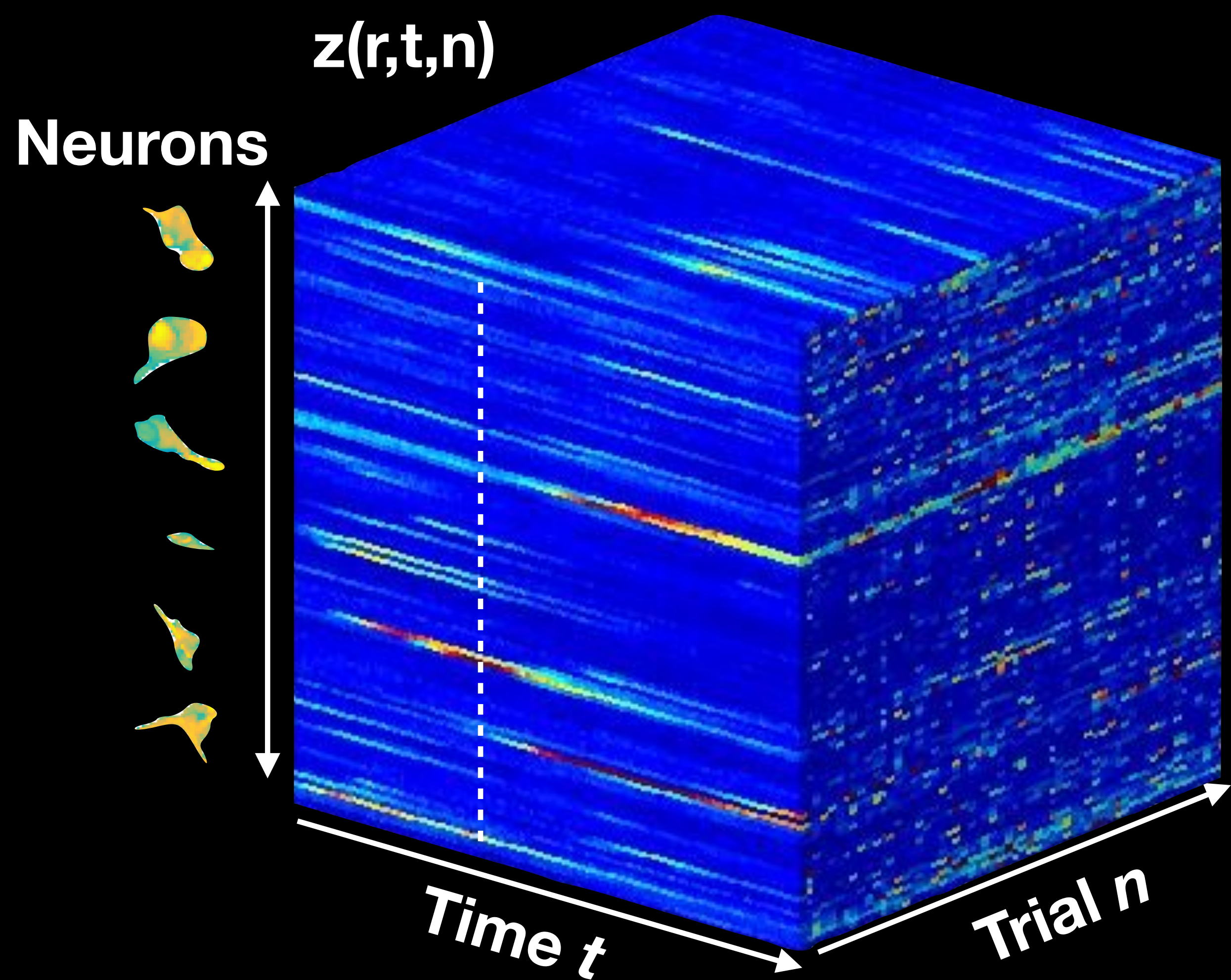
Multi-trial Analysis

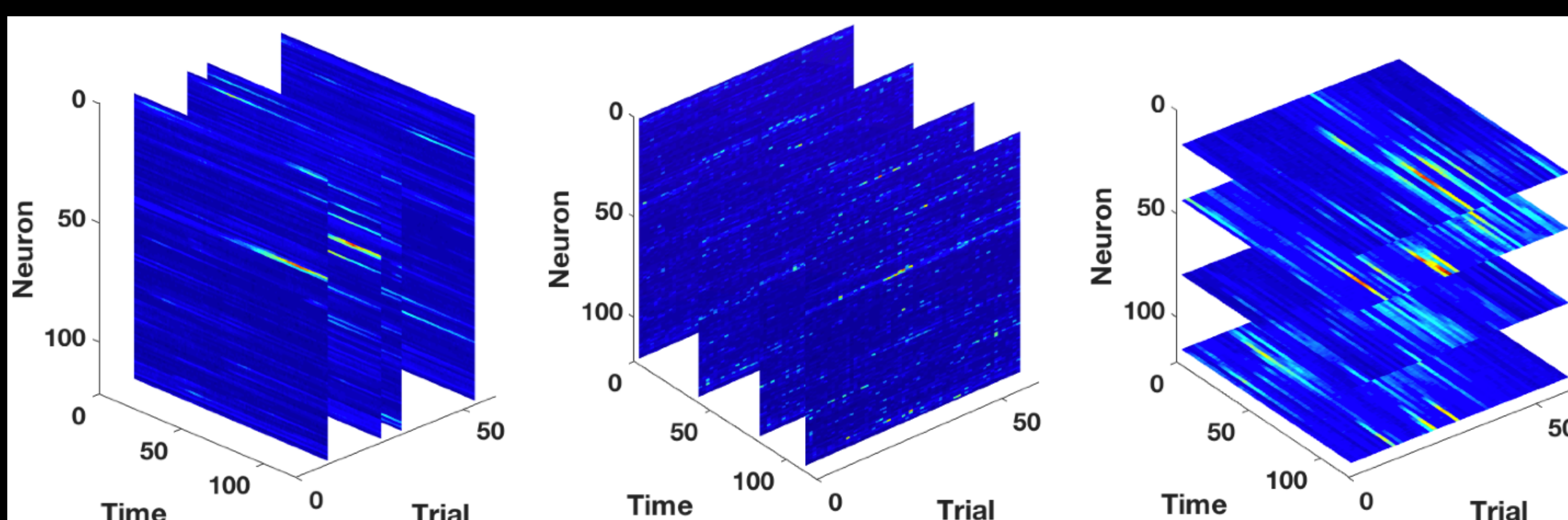


Multi-trial Analysis



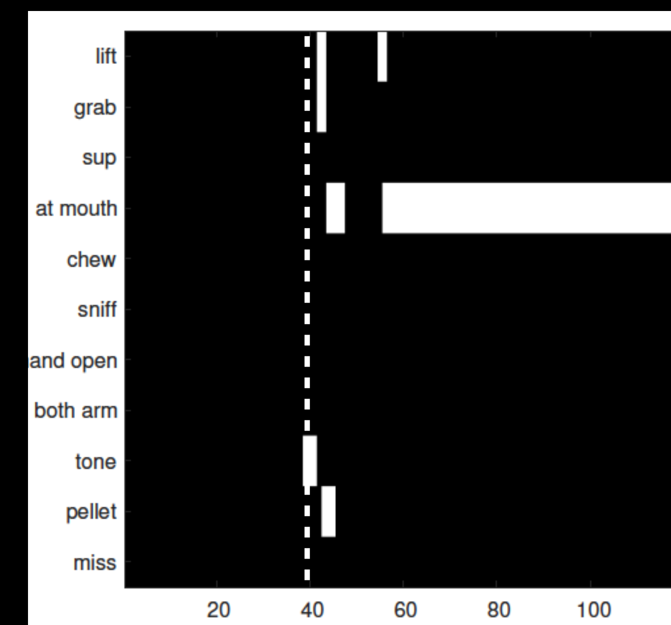
Multi-trial Analysis with Tensors



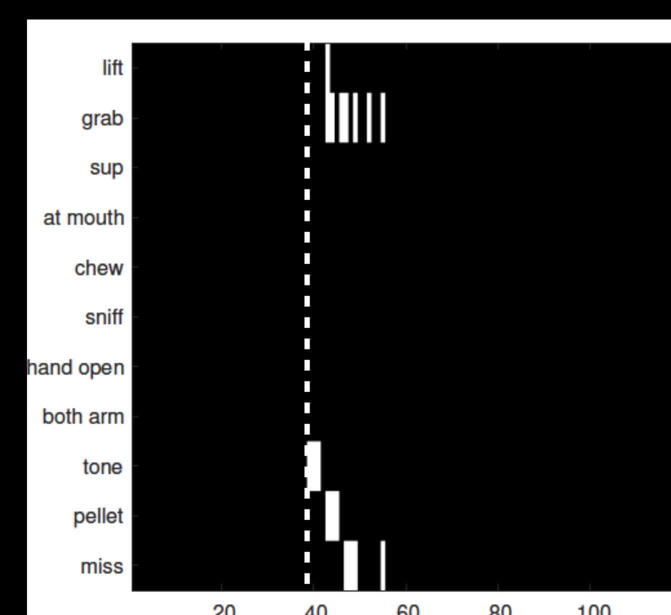


Behavioral events

Succeeded at task



Failed at task



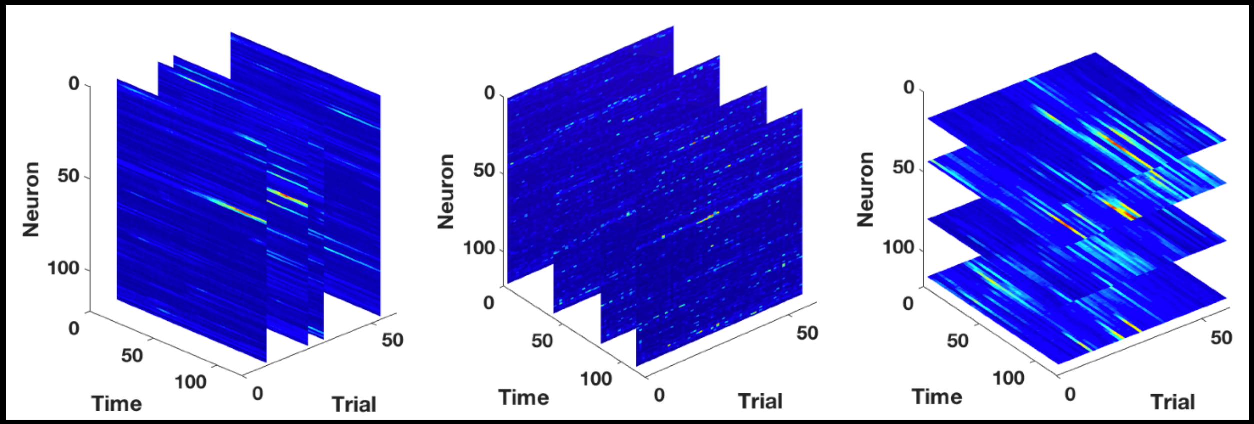
Coupled Metrics

In the context of kernel methods, we need a metric

$$\mathbf{W}[i,j] = \exp\{-d(\mathbf{X}_i, \mathbf{X}_j)/\sigma\}$$

How to appropriately compare slices?

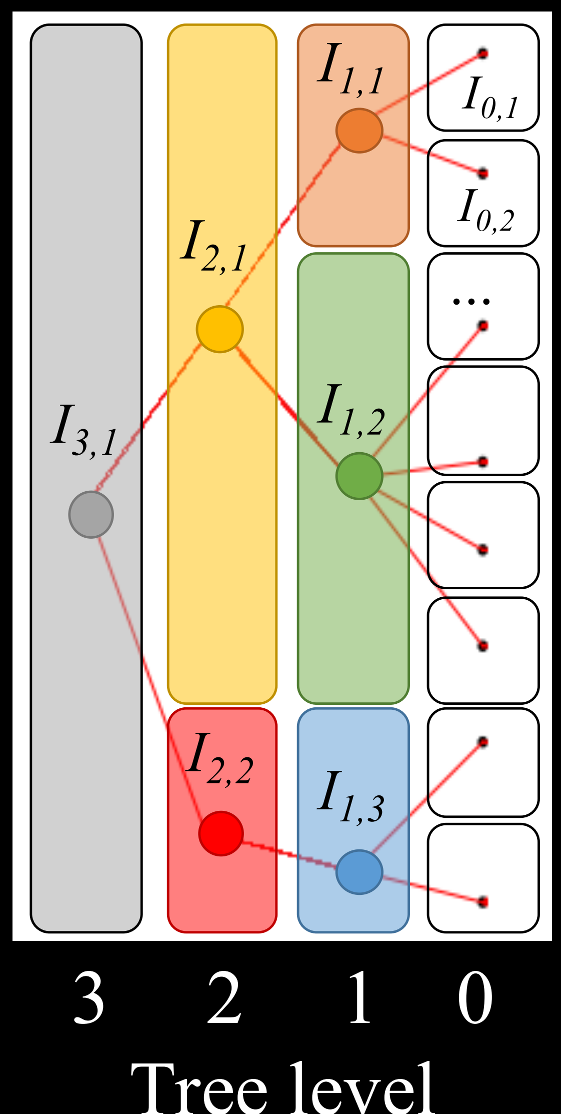
- Maintain high resolution within a slice
- ***"Data-driven metric"*** - incorporates the coupling/dependencies both within a slice and between the axes
- Unsupervised

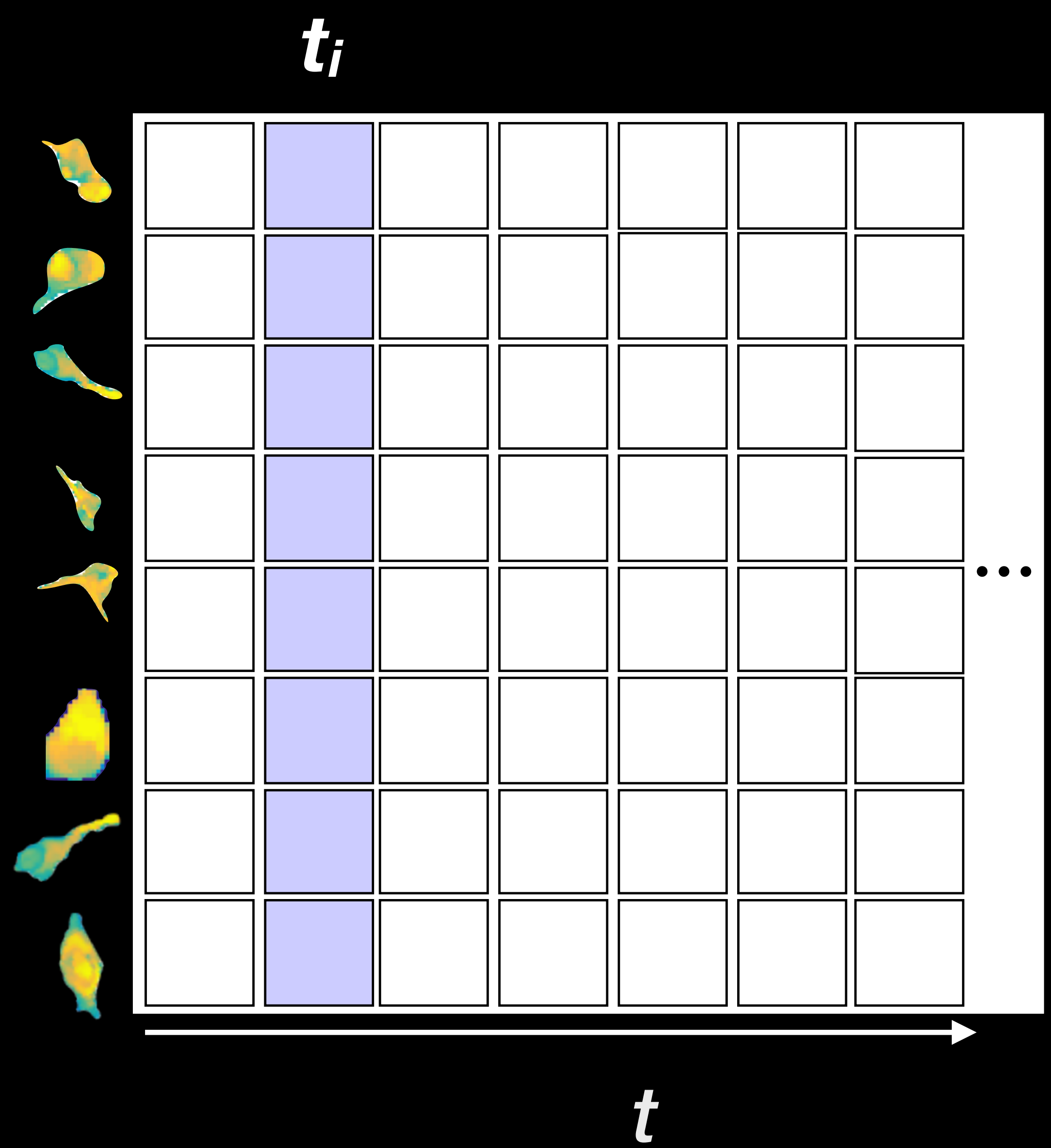


Data Organization

- Trial = 2D slice of neurons x time
- Multiscale hierarchical clustering of the neurons

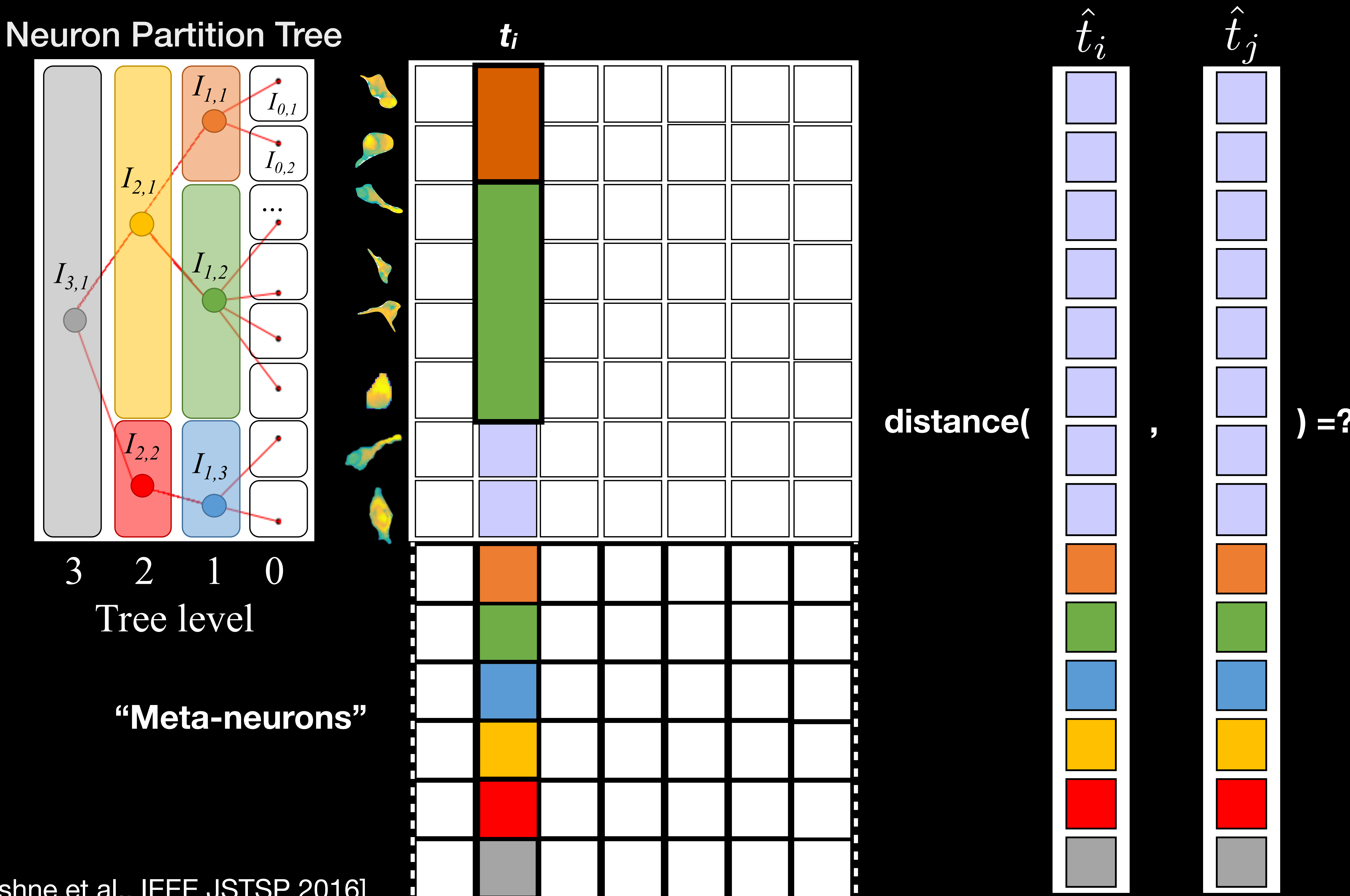
Neuron Partition Tree





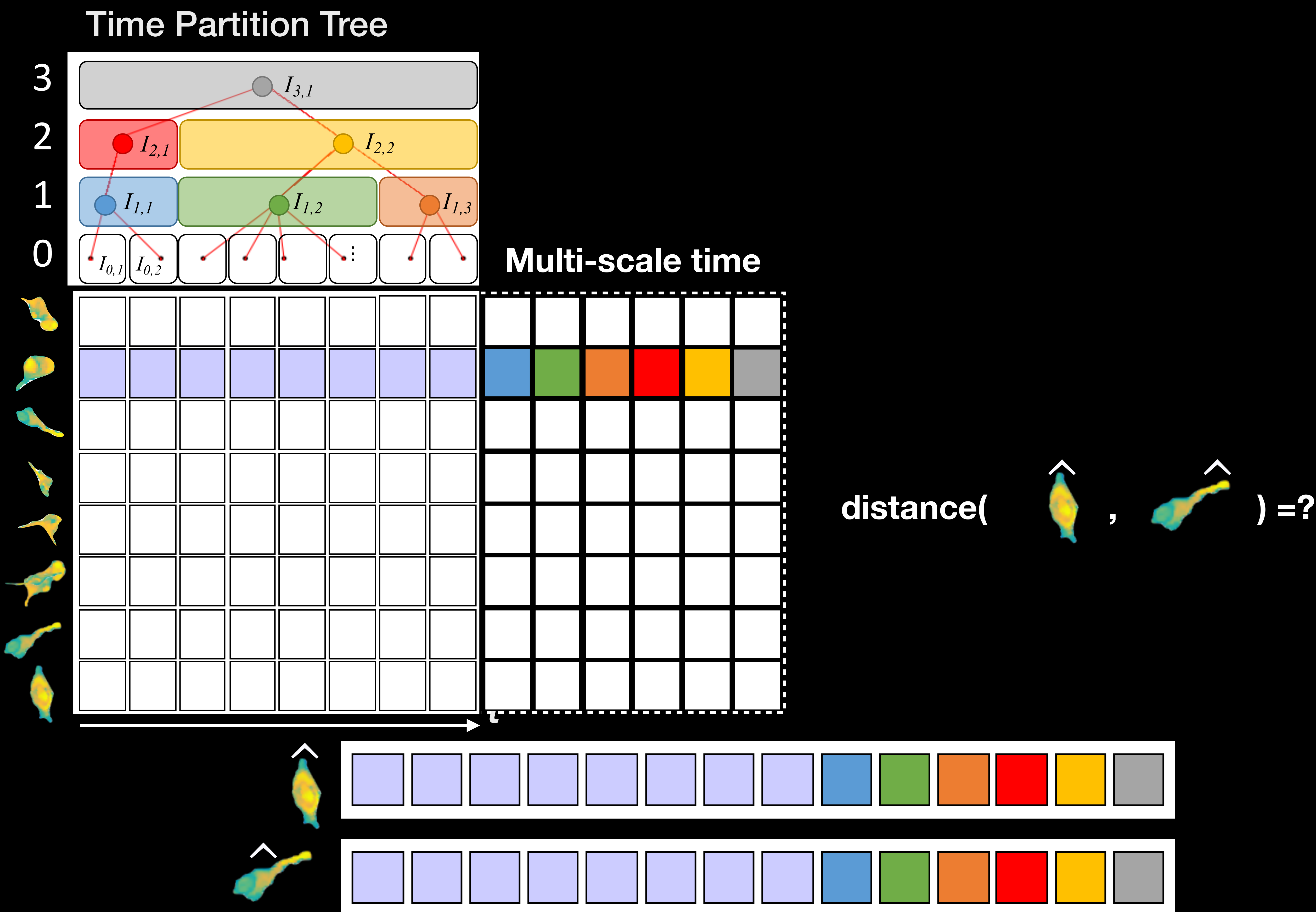
[Gavish et al., 2010
Ankenman, 2014
Leeb, 2015]

Tree Transform



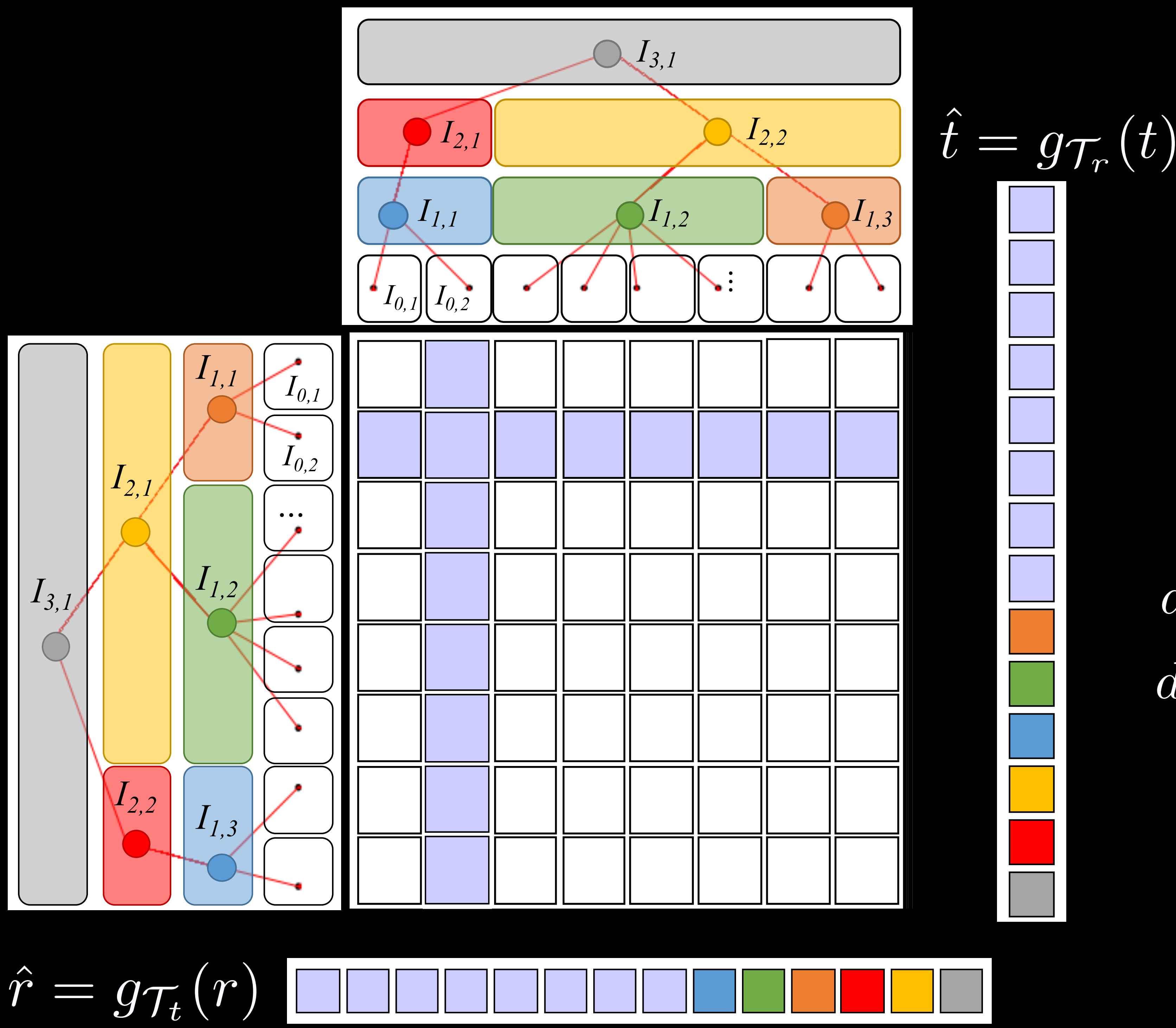
[Mishne et al., IEEE JSTSP 2016]

Tree Transform



[Mishne et al., IEEE JSTSP 2016]

Tree Metrics



The tree forms a set of data-driven **multiscale “lowpass”** filters and a new metric

$$d_{\mathcal{T}_r}(t, t') = \|g_{\mathcal{T}_r}(t) - g_{\mathcal{T}_r}(t')\|_1$$
$$d_{\mathcal{T}_t}(r, r') = \|g_{\mathcal{T}_t}(r) - g_{\mathcal{T}_t}(r')\|_1$$

[Mishne et al., IEEE JSTSP 2016]

Generalized Earth Mover's Distance

Result: [Leeb, 2014, Mishne et al., 2016]

The tree metric is equivalent to a generalized Earth Mover's Distance (EMD):

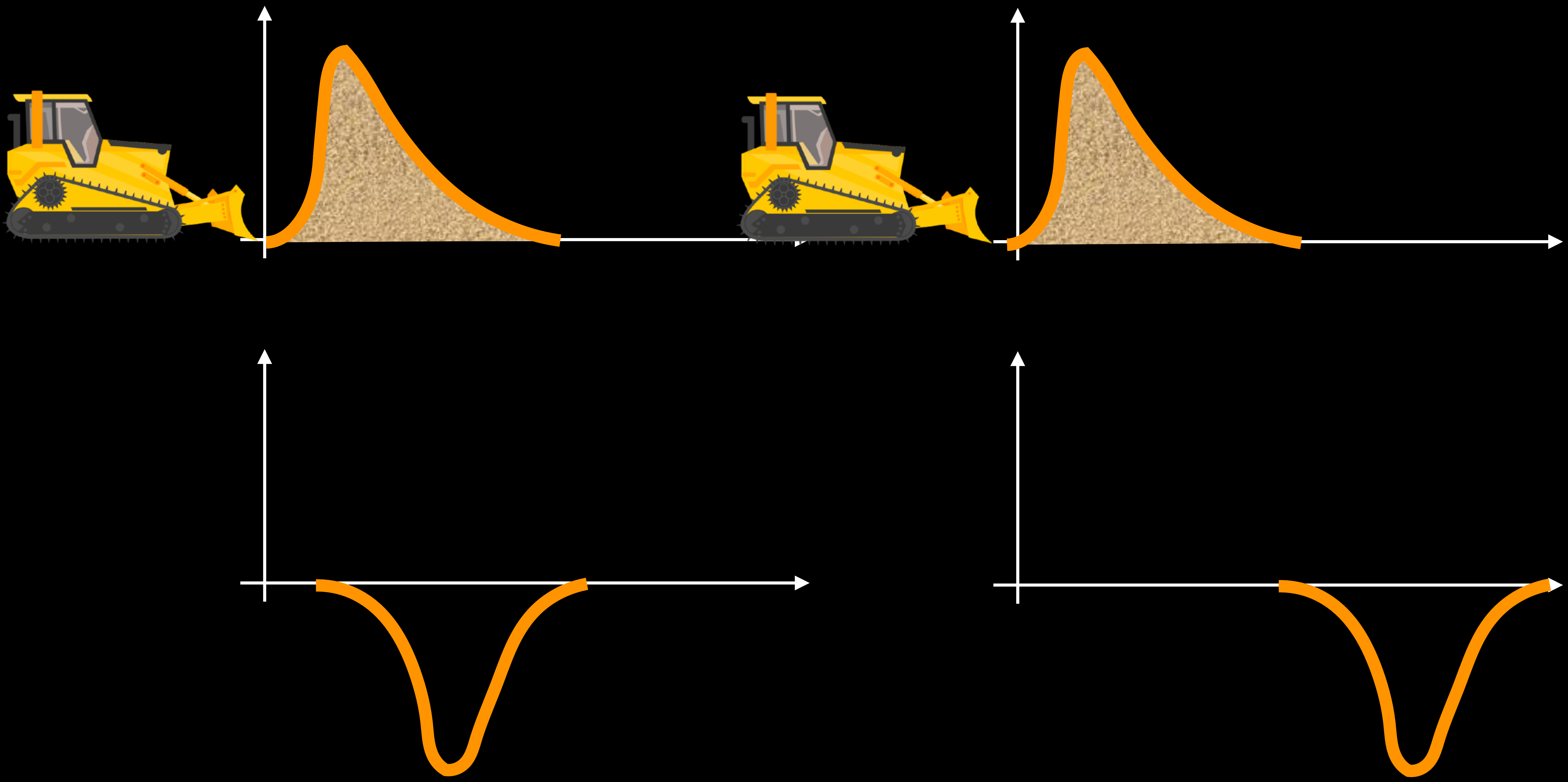
$$d_{\mathcal{T}_t}(r_i, r_j) = \|\mathbf{W}_{\mathcal{T}_t}(\hat{r}_i - \hat{r}_j)\|_1$$

$$\boxed{\omega(\mathcal{I}) = \left(\frac{|\mathcal{I}|}{M}\right)^{\beta+1}}$$

Earth Mover's Distance

- Distance between two probability distributions (Wasserstein metric)
- Popular in computer vision [Rubner 1998]

Generalized Earth Mover's Distance

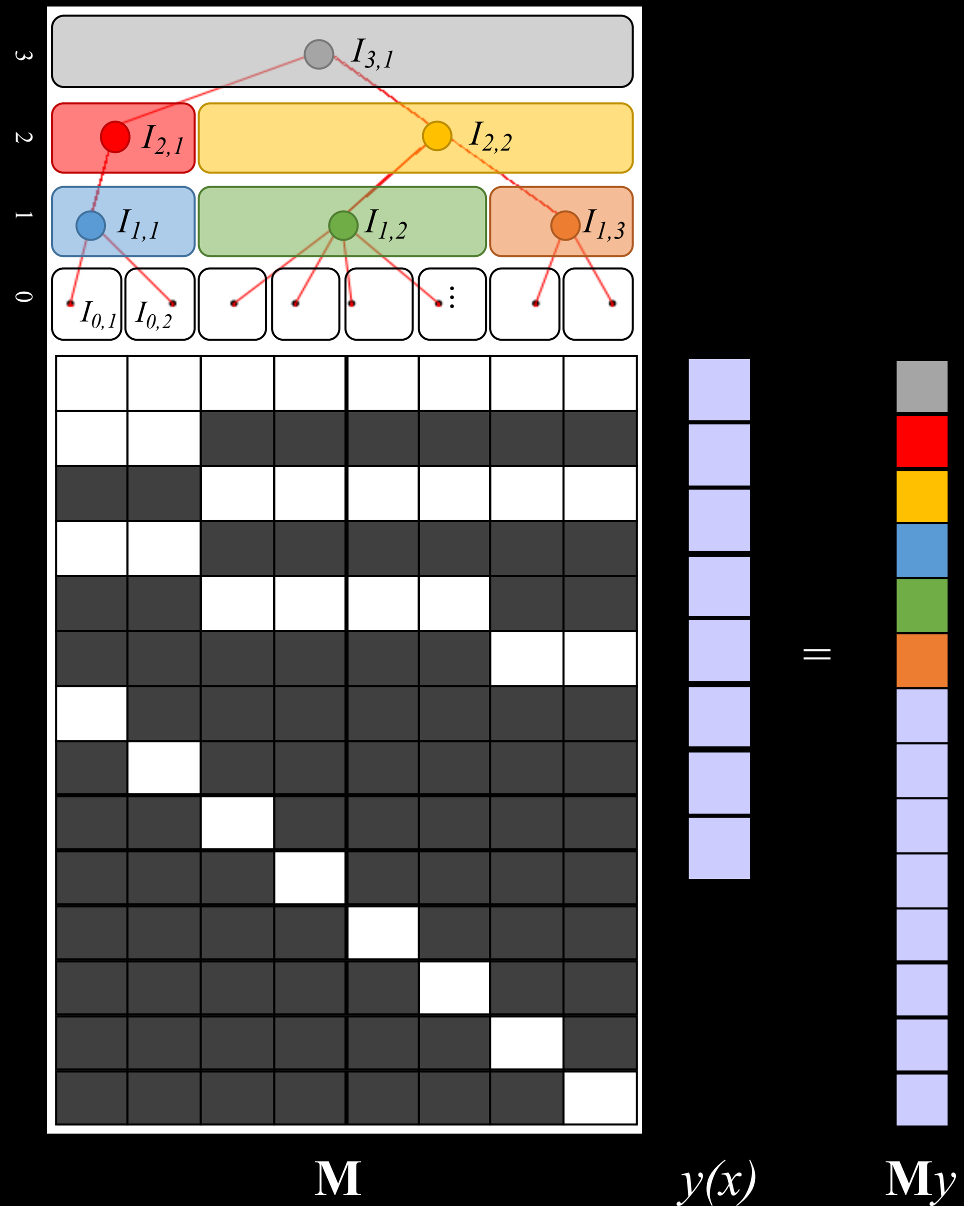


Tree Transform

Tight relation to **Signal Processing on Graphs** [Shuman et al., 13]

- Our filters are data-driven
- Formulated as a linear transform
- Features whose comparison is ‘meaningful’

$$d_{\mathcal{T}_{\mathcal{X}}}(y, y') = \|\mathbf{W}\mathbf{M}_{\mathcal{T}_{\mathcal{X}}}(y - y')\|_1$$

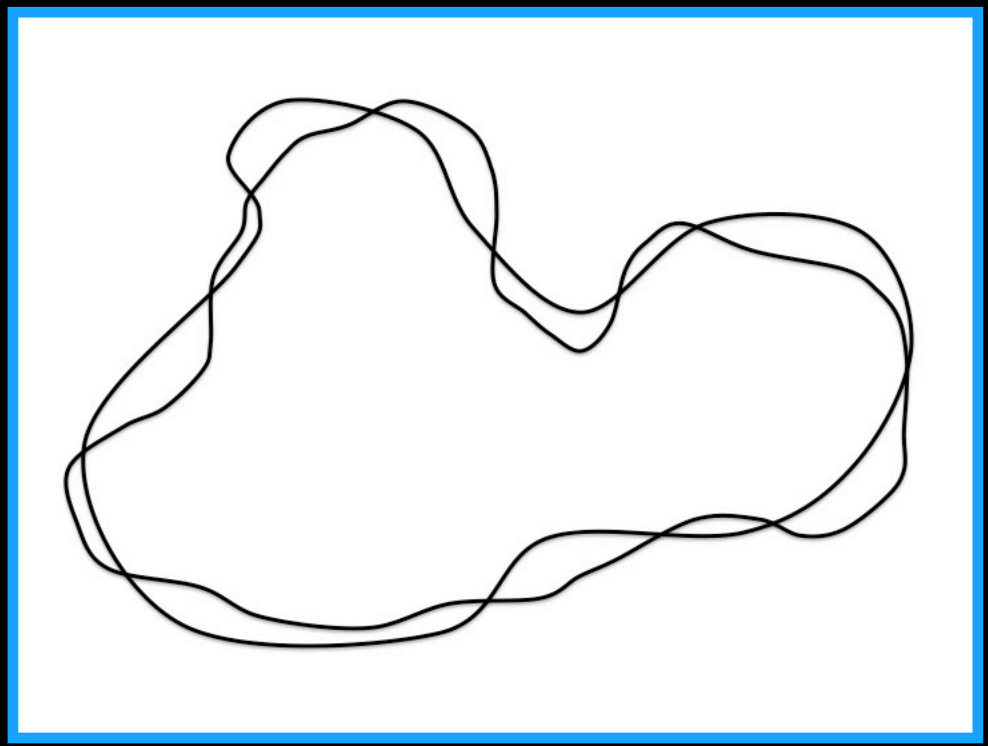


[Mishne, Talmon, Cohen, Coifman, Kluger, IEEE TSIPN 2017]

Illustrative Example

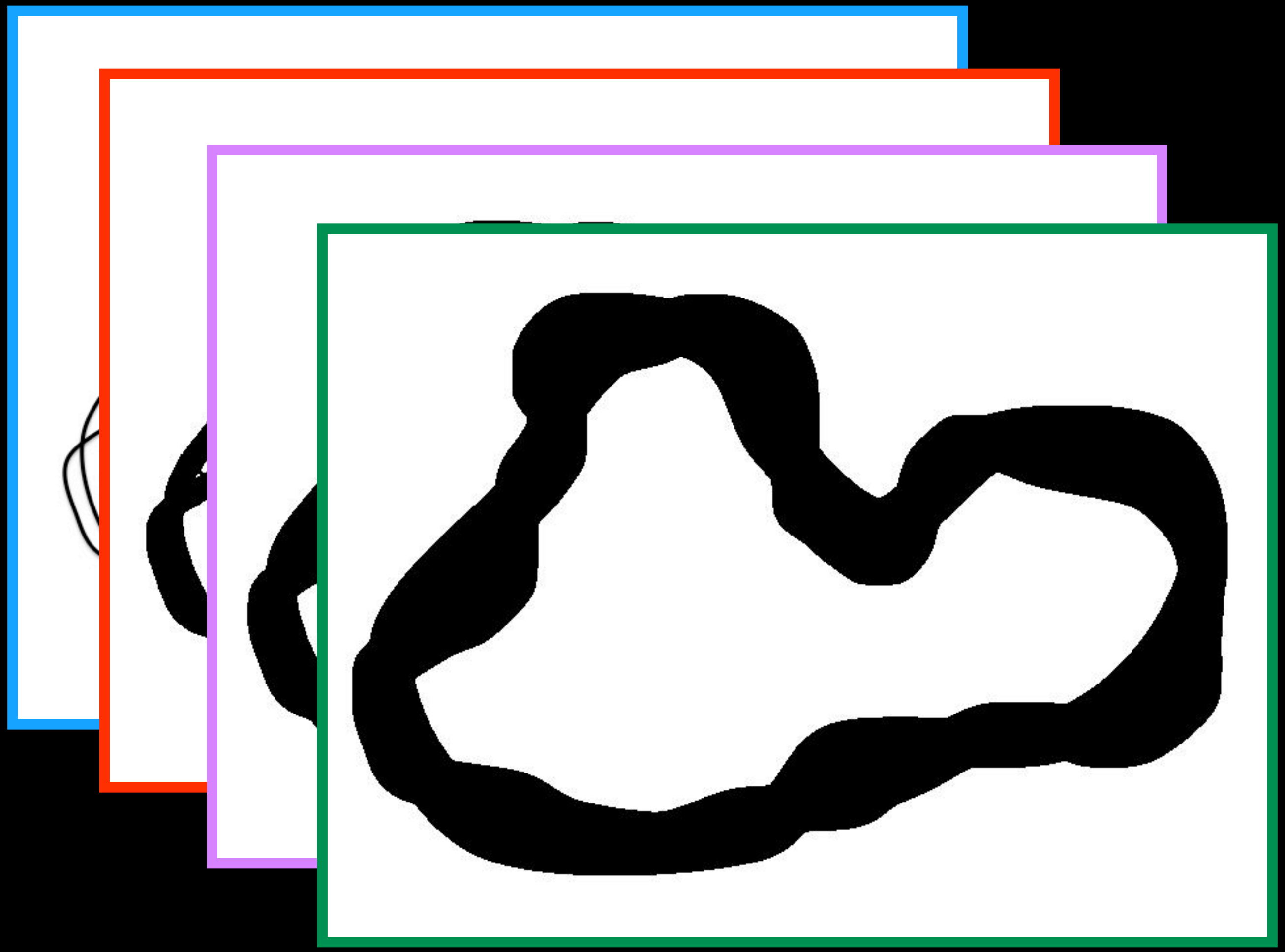
Coming back to comparing observations/features…

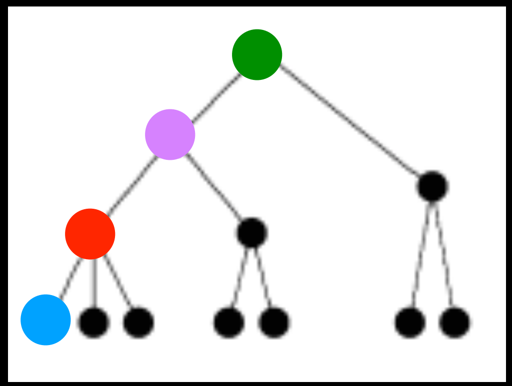
Euclidean distances *in high dimension* are meaningless



Illustrative Example

Efficient implementation via **lowpass filters**:
obtaining coarser and coarser views



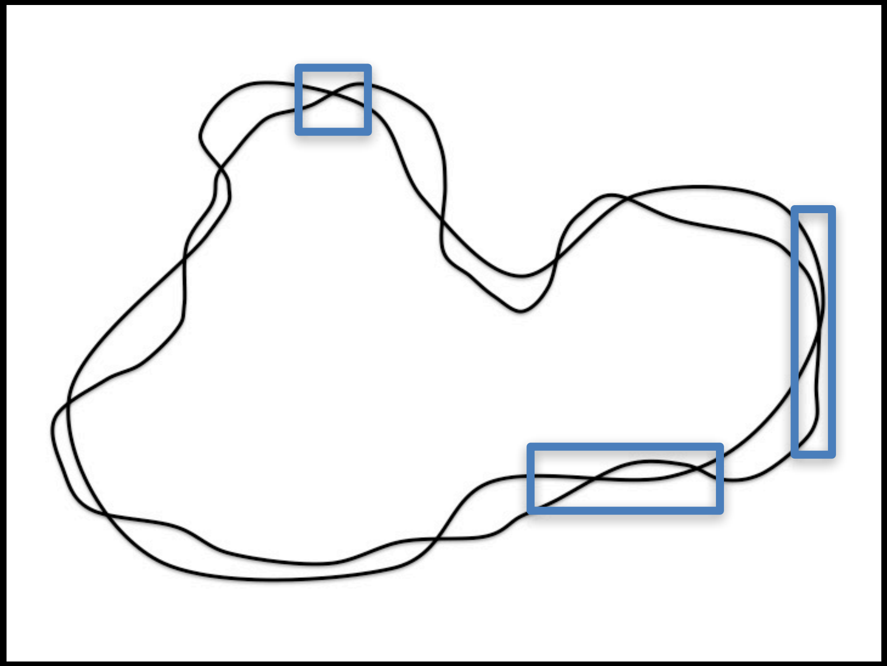


Tree-based EMD

Comparisons based on all levels

The right <u>neighborhoods</u> in multiple scales at every point and their respective weight

- Not necessarily local…
- The definition of the neighborhoods is data-driven

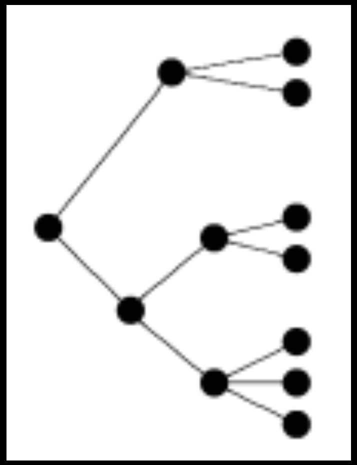


The proper <u>**weight**</u> of each level

Iterative Algorithm

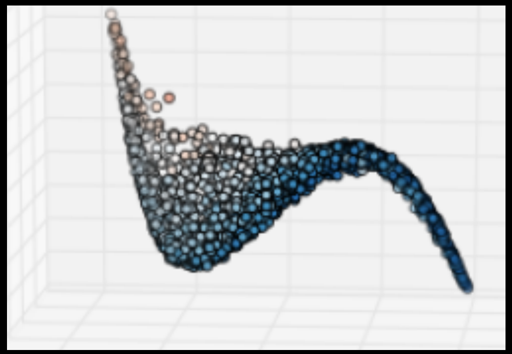
$d(r, r')$

$\mathcal{T}_r$

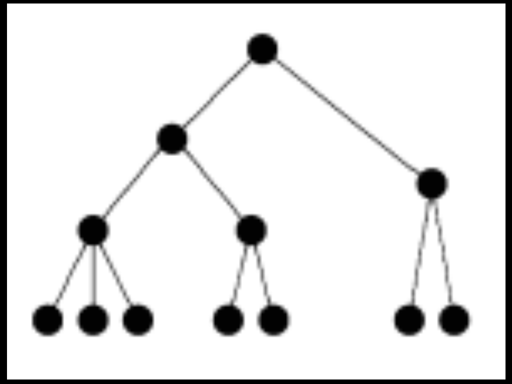


$d_{\mathcal{T}_r}(t, t') = \|\mathbf{W}_{\mathcal{T}_r}(\hat{t} - \hat{t}')\|_1$

Ψ_t

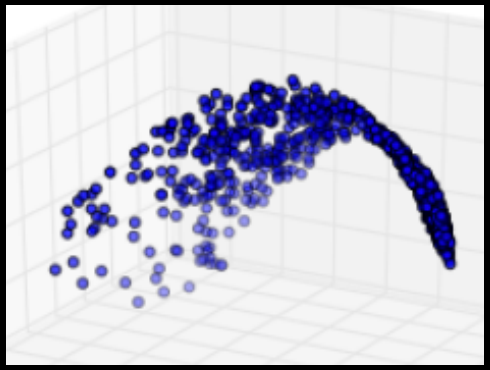


$\mathcal{T}_t$



$d_{\mathcal{T}_t}(r, r') = \|\mathbf{W}_{\mathcal{T}_t}(\hat{r} - \hat{r}')\|_1$

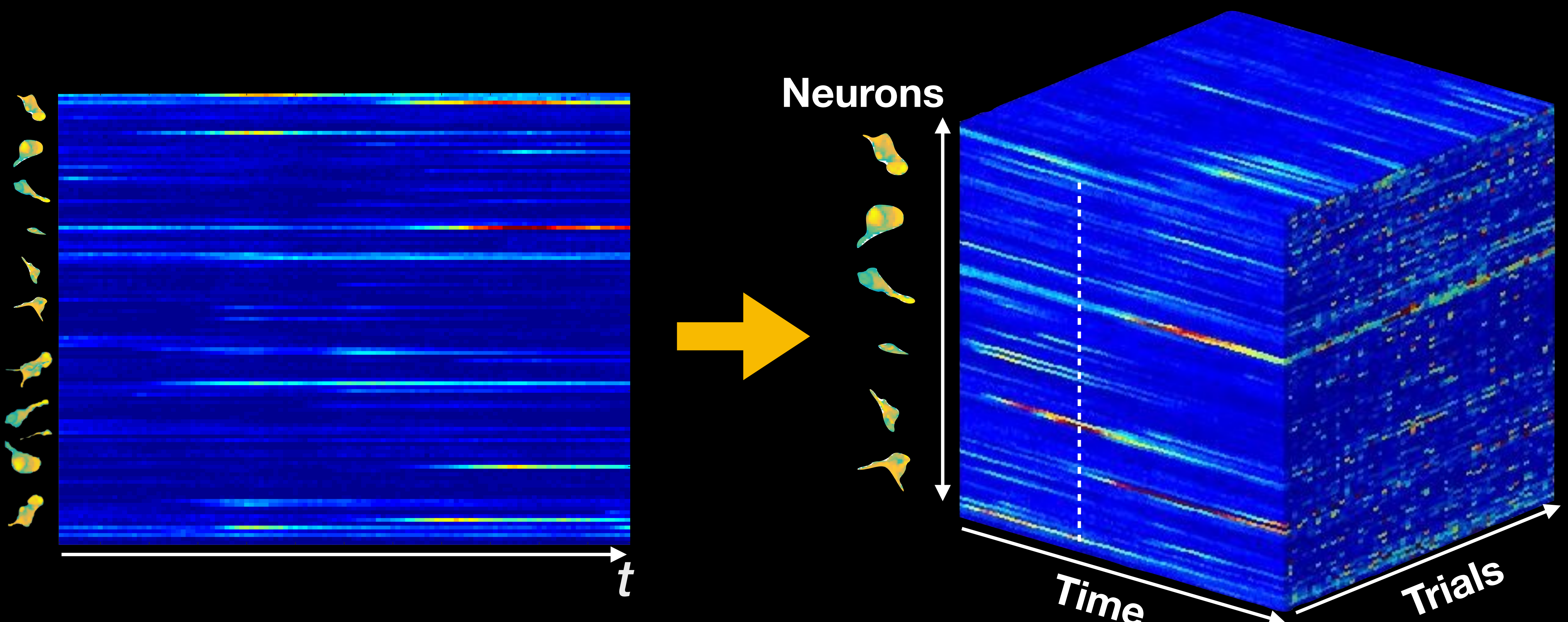
Ψ_r



Tensor organization

- The extension to 3D is analogous
- Bi-tree metric: Compare two neurons based on multi-scale time and trial structures

$$d_{\mathcal{T}_t,\mathcal{T}_T}(r_i, r_j) = \|g_{\mathcal{T}_t,\mathcal{T}_T}(r_i) - g_{\mathcal{T}_t,\mathcal{T}_T}(r_j)\|_1$$



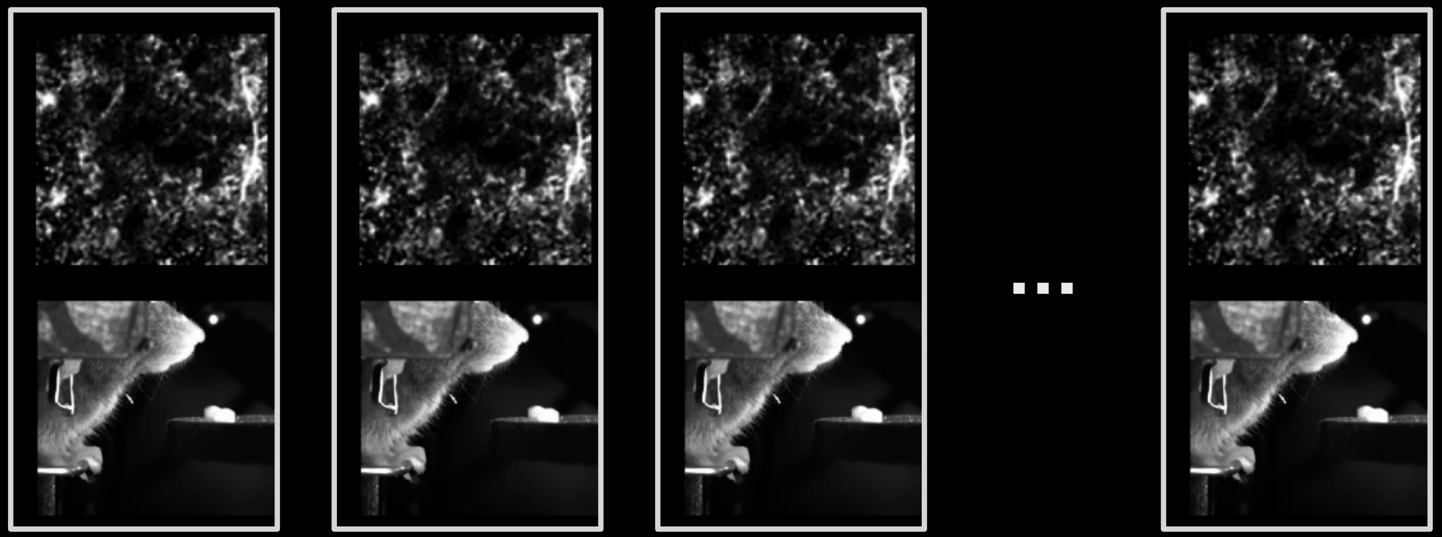
Multi-trial experiments

Mouse learns motor task over 3 weeks

Measurements:

1. Neuronal activity

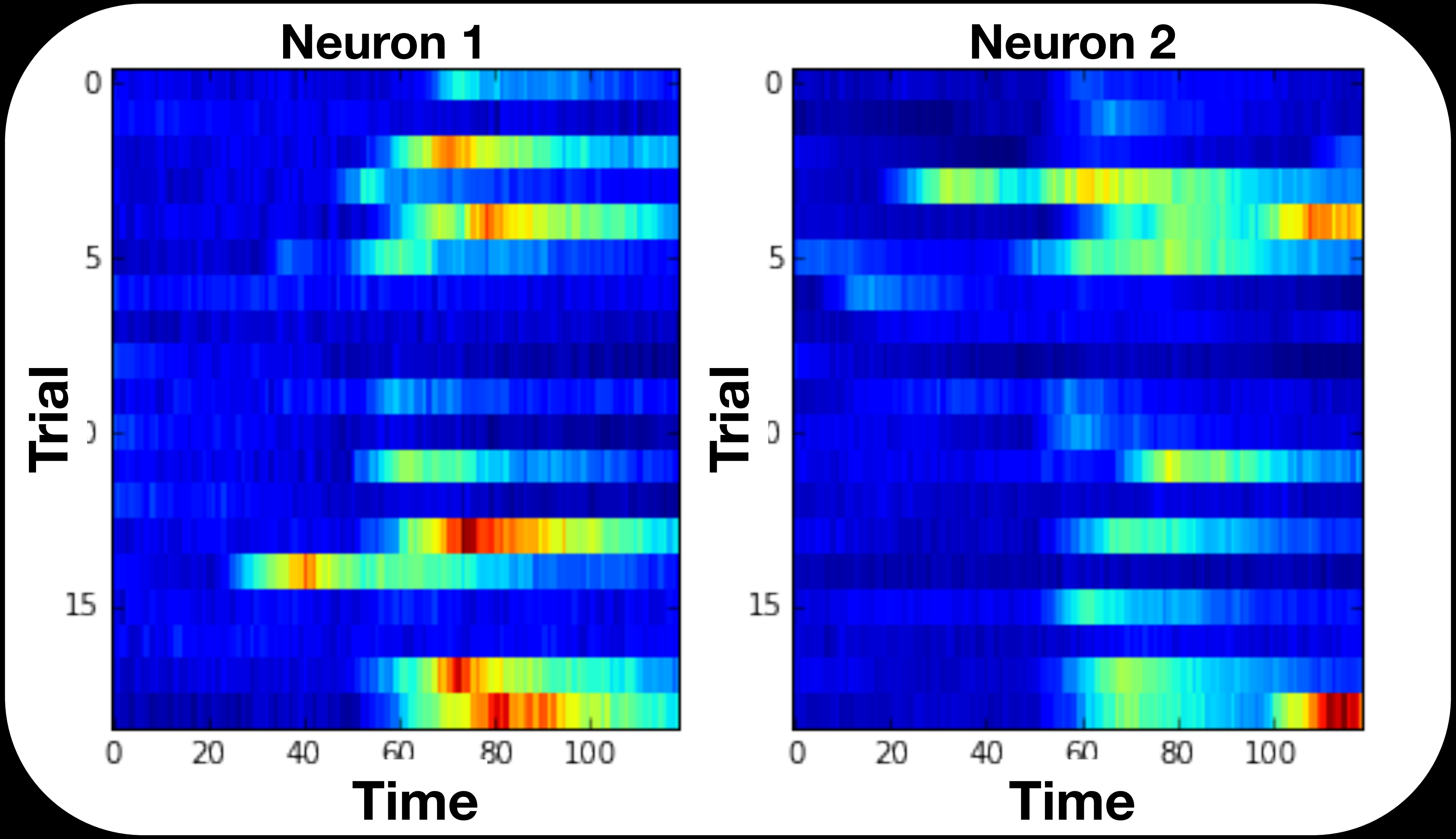
2. Behavior

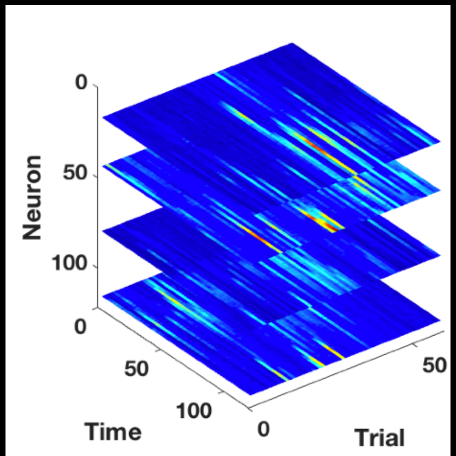


Protocol:

- Head-fixed motor reach **task**
- **Repeated fixed-length trials**

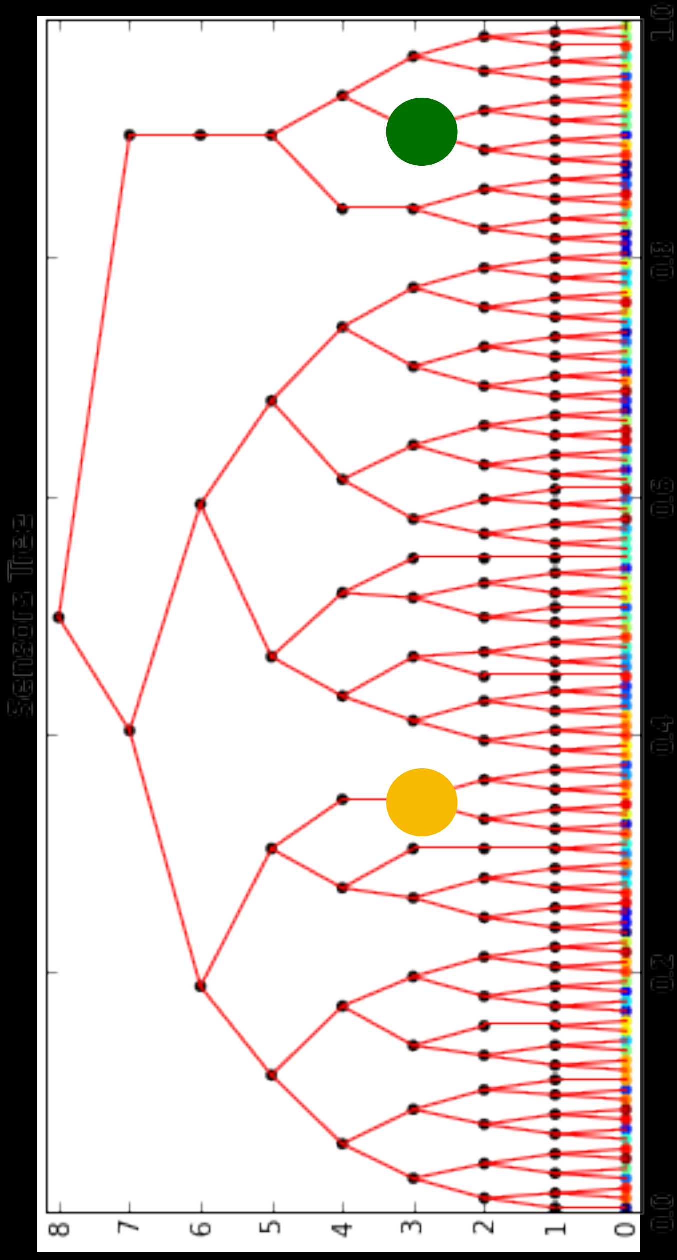
Comparing neurons

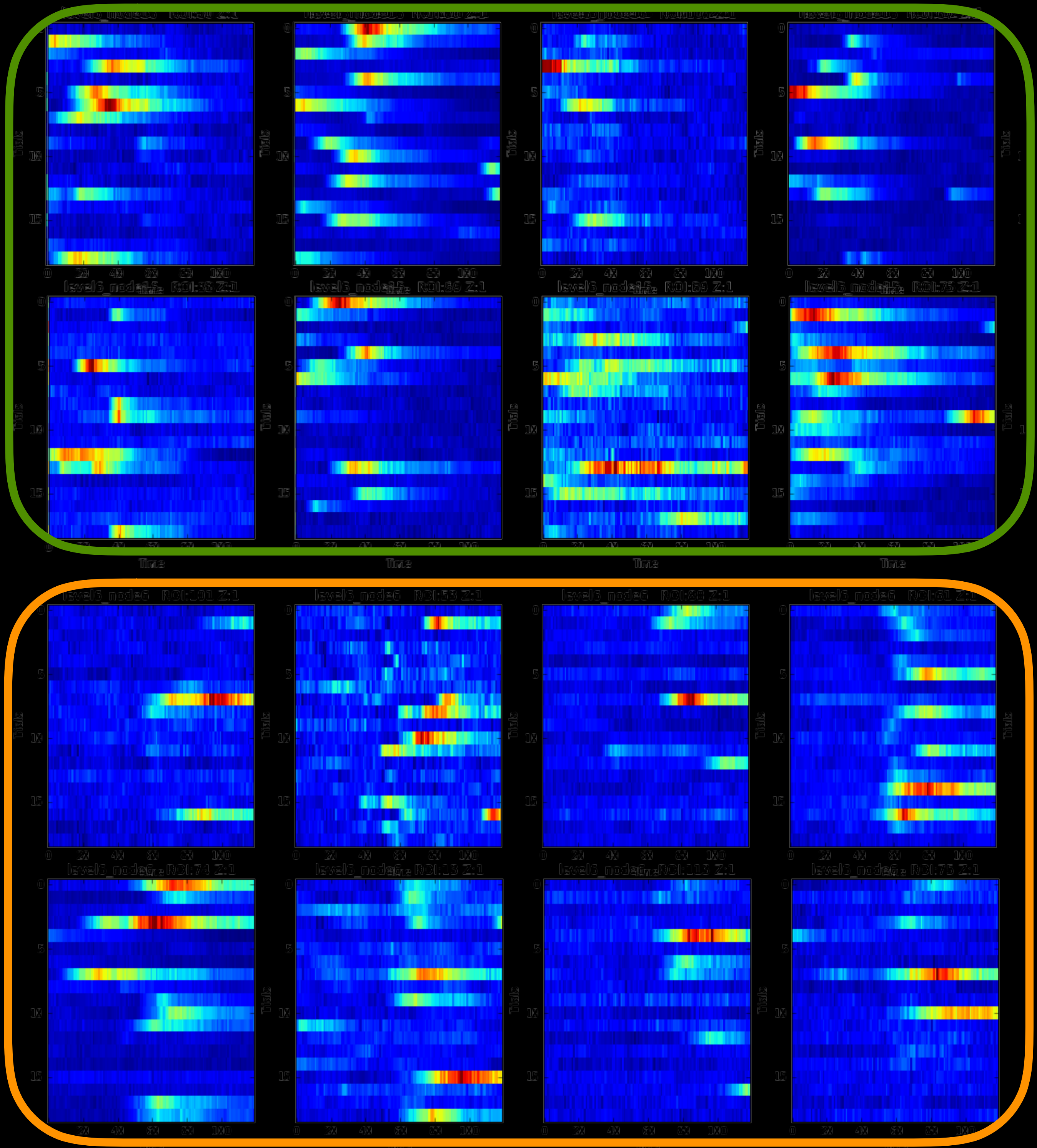


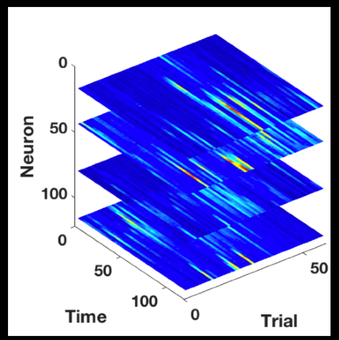


Neuronal data analysis

Neuron tree

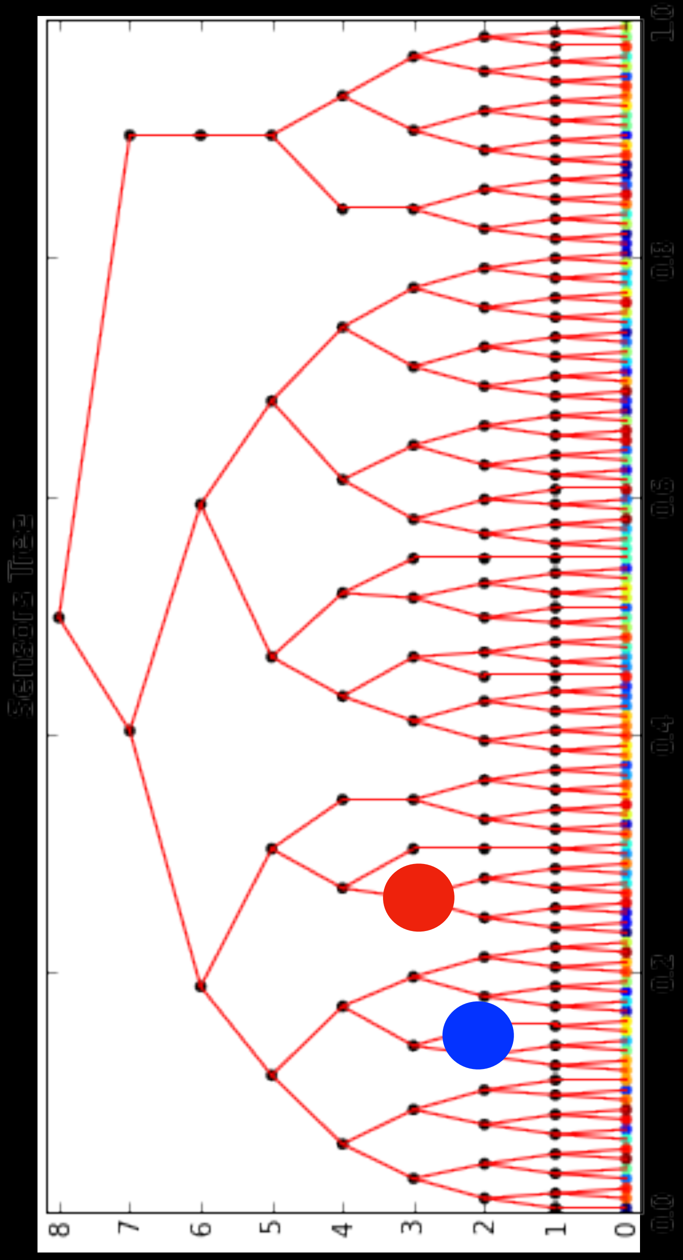


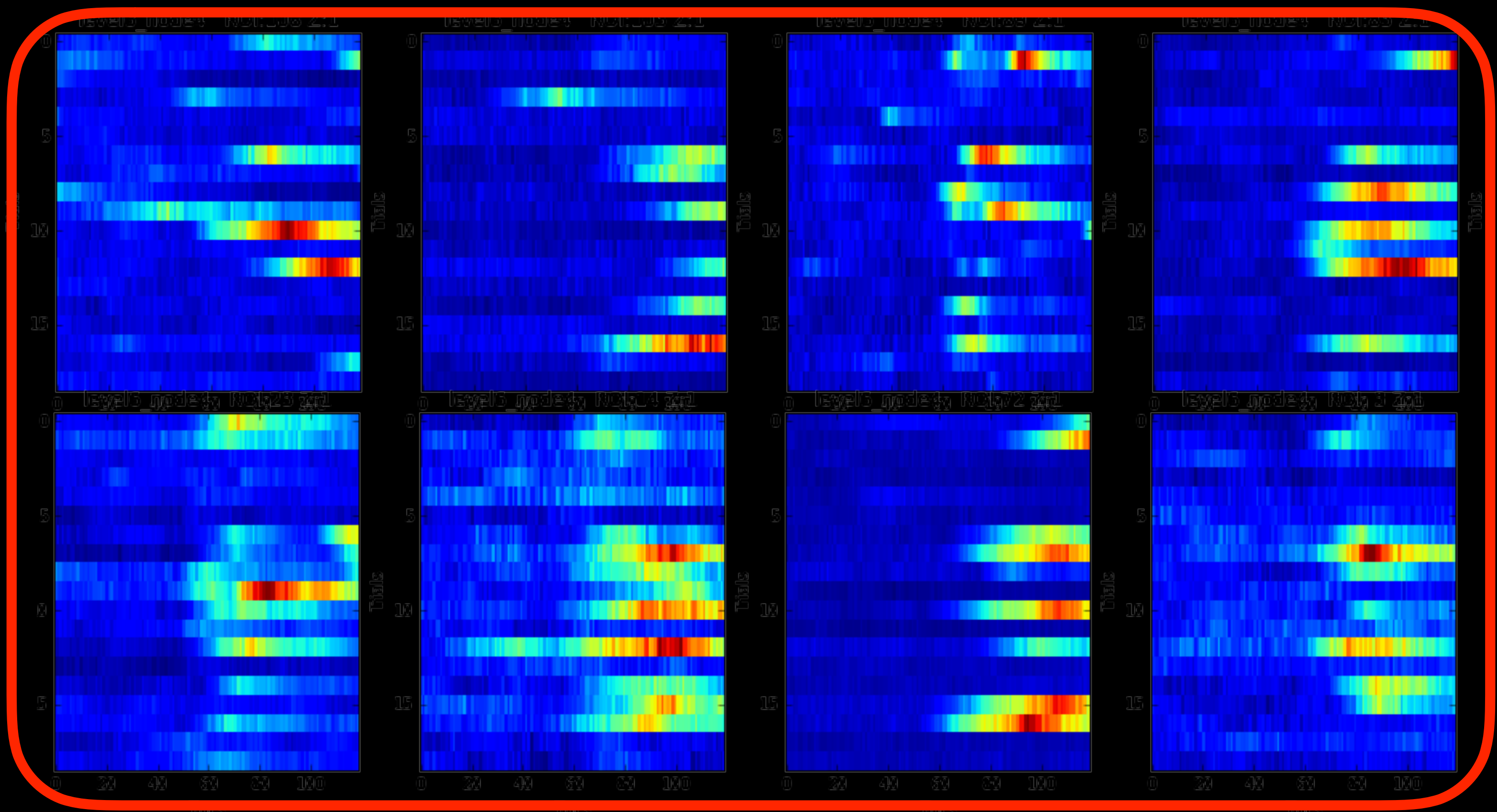


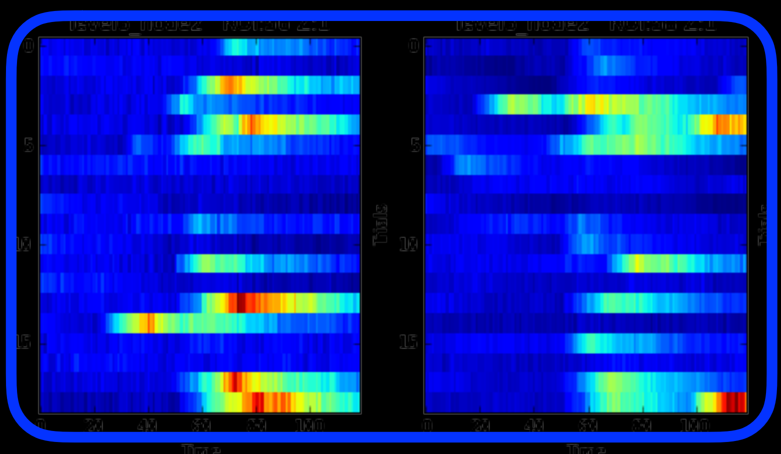


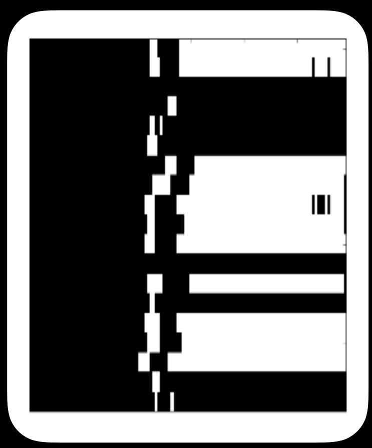
Neuronal data analysis

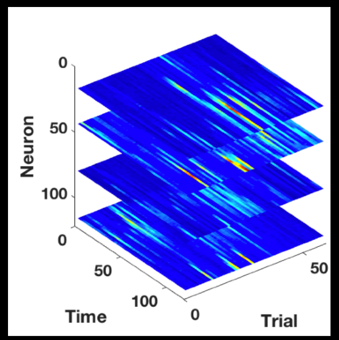
Neuron tree







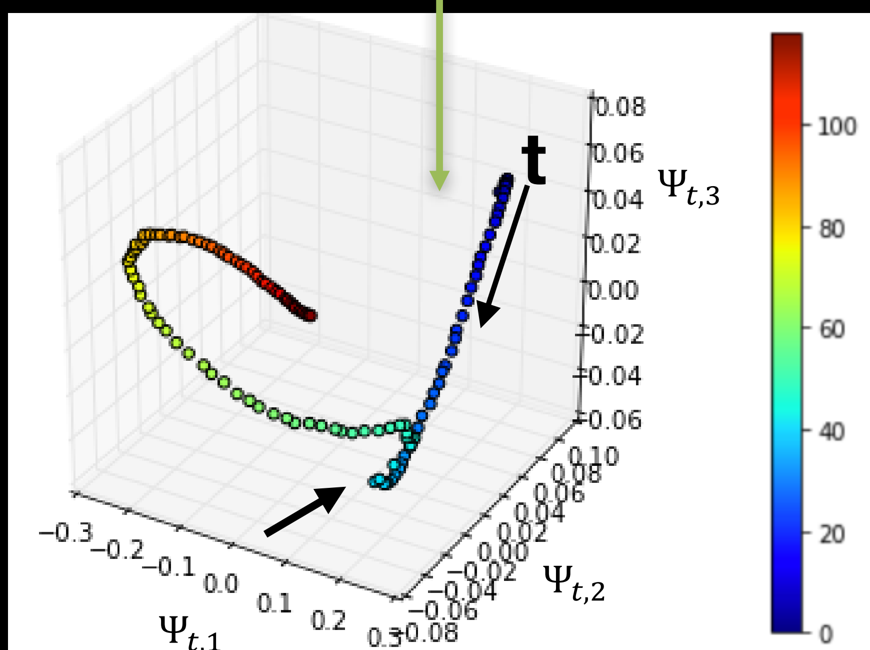




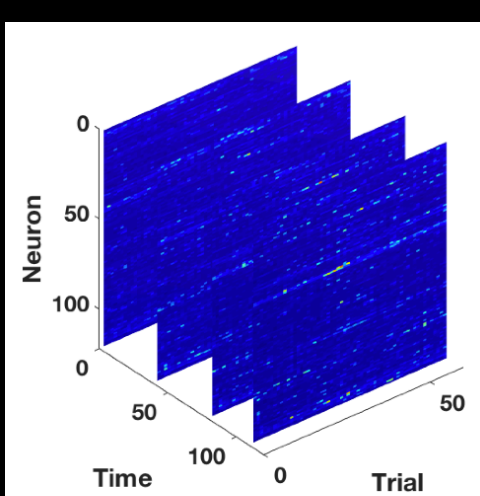
Neuronal data analysis

Time evolution in manifold

Before tone



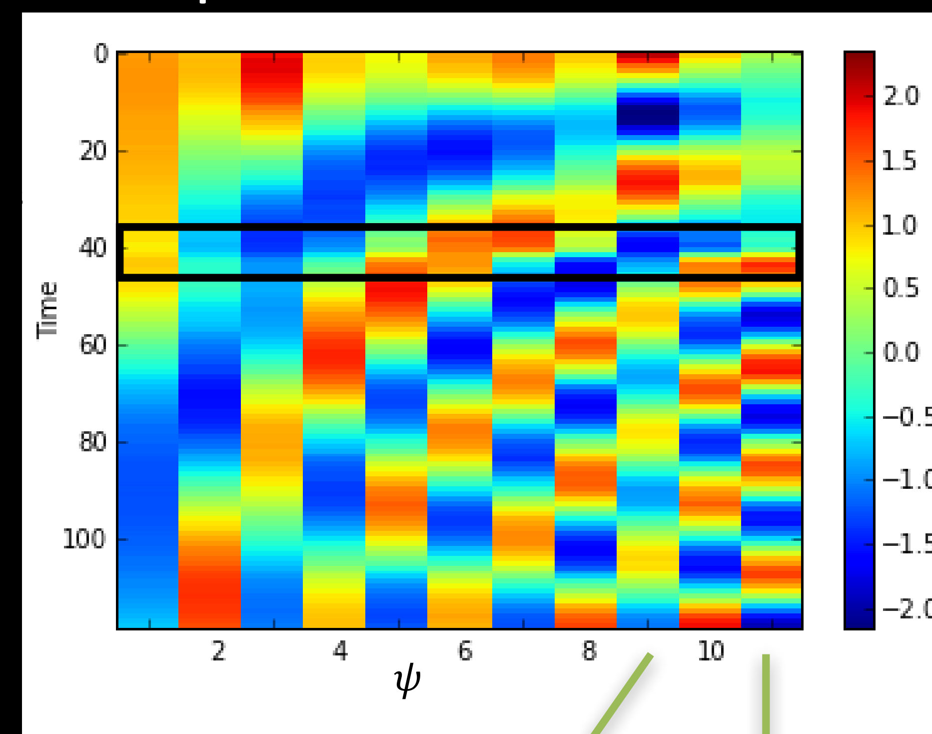
Tone - discovered from data



First 11 eigenvectors

Data-dependent harmonic functions

Tone

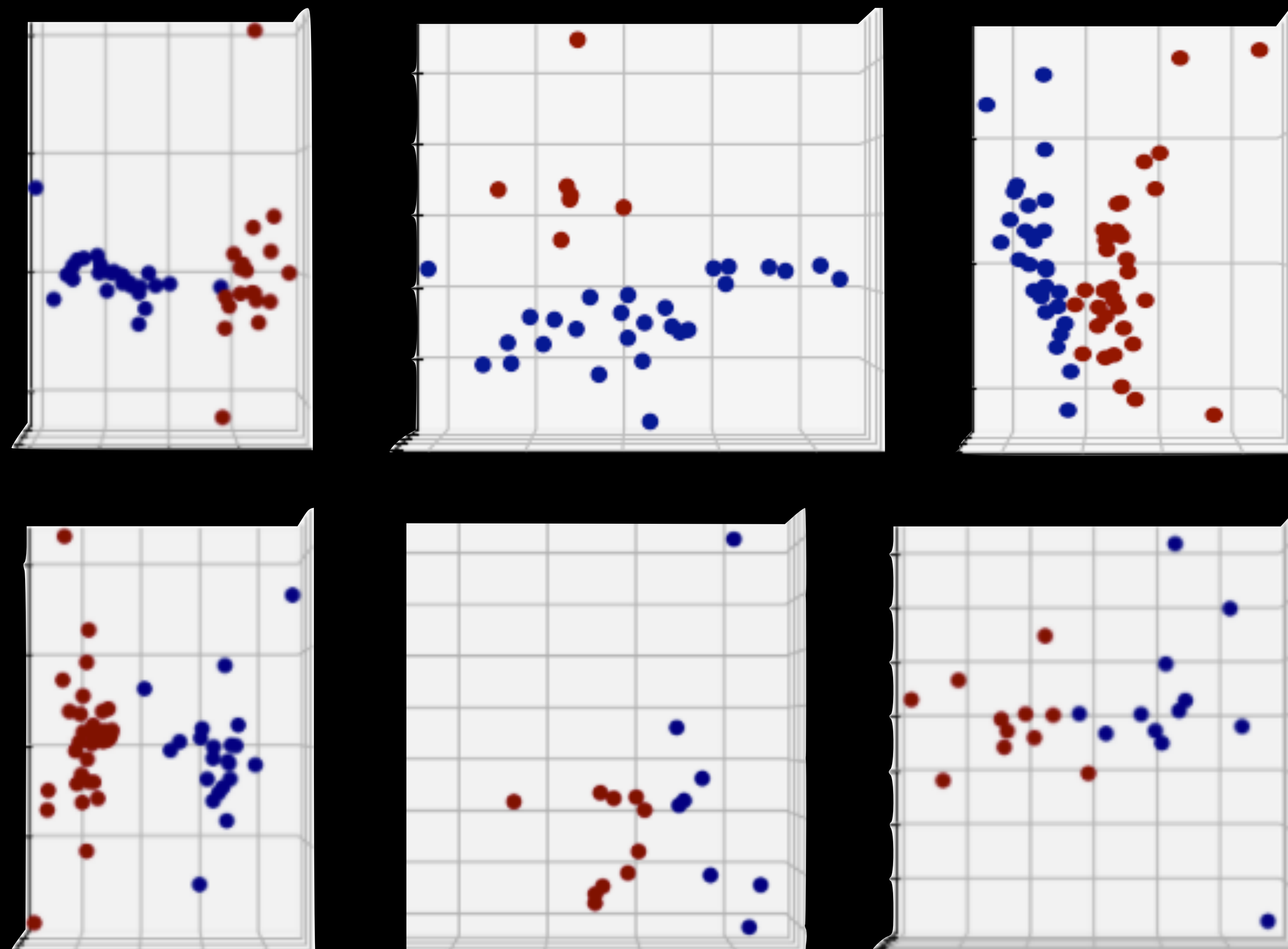


Localized pre-tone

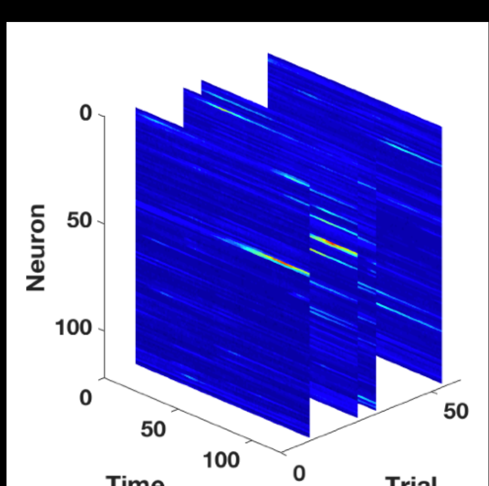
Localized post-tone

Neuronal data analysis

Trial organization in embedding space

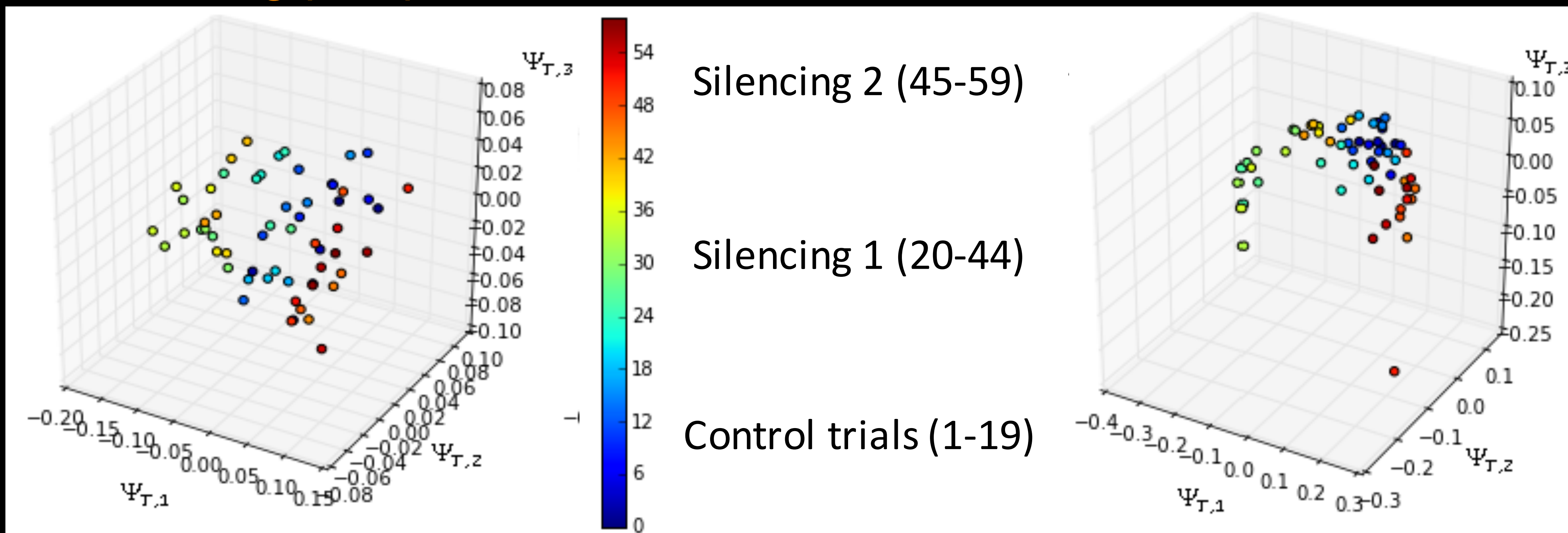


Successful at task
Failure at task



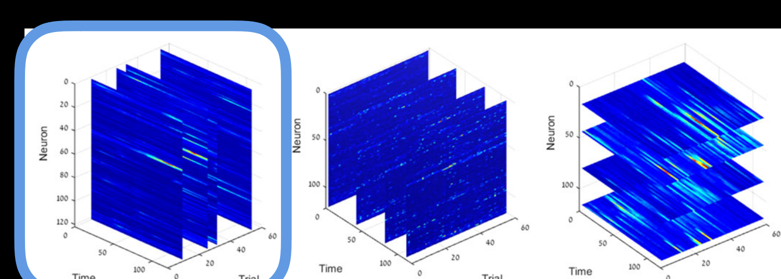
Neuronal data analysis

Silencing (CNO) trials 20-59



Initial **Iterated**

- Functional self-organization
- Detect pathological dysfunction
- Silencing has a delay – adaptation



Neuronal data analysis

Solely from the neuronal measurements we discover:

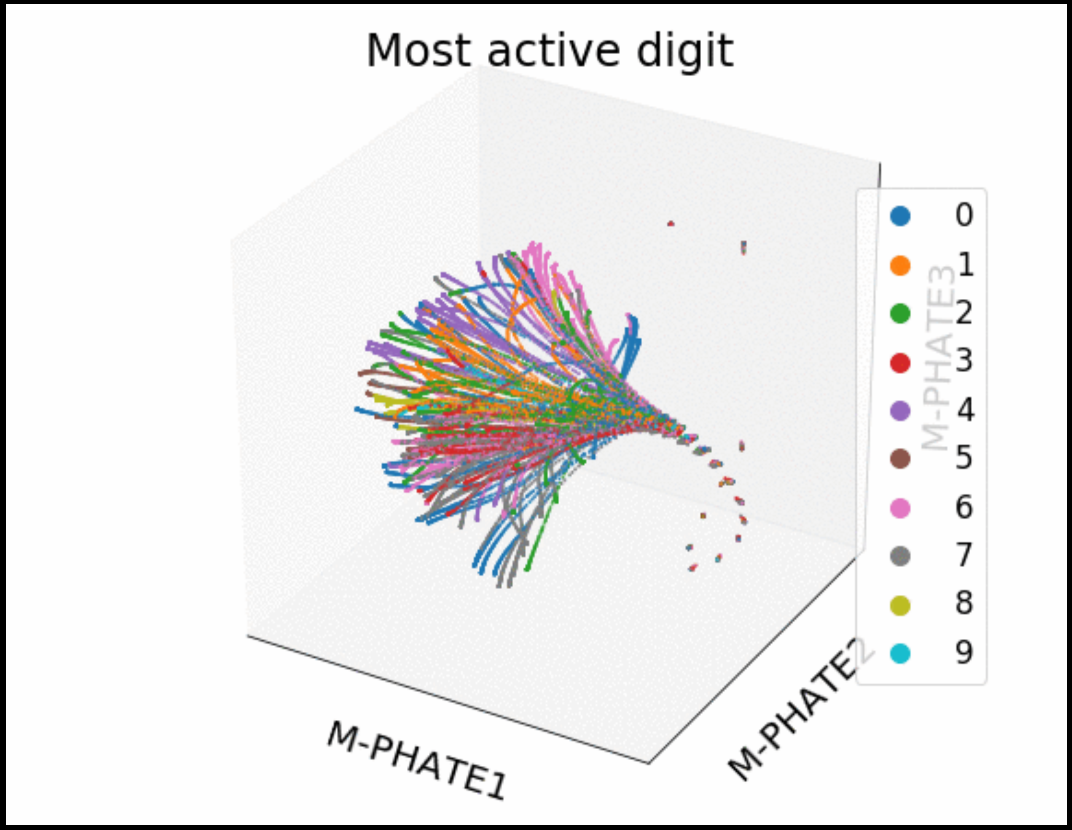
- Neuronal **activity patterns** associated with external triggers and behavioral events
- Different **time scales**, from recovering the local “script” of the trial, to global behavioral organization of trials
- Functional subsets of neurons

Applicable to other data: EEG, fMRI

https://github.com/gmishne/pyquest

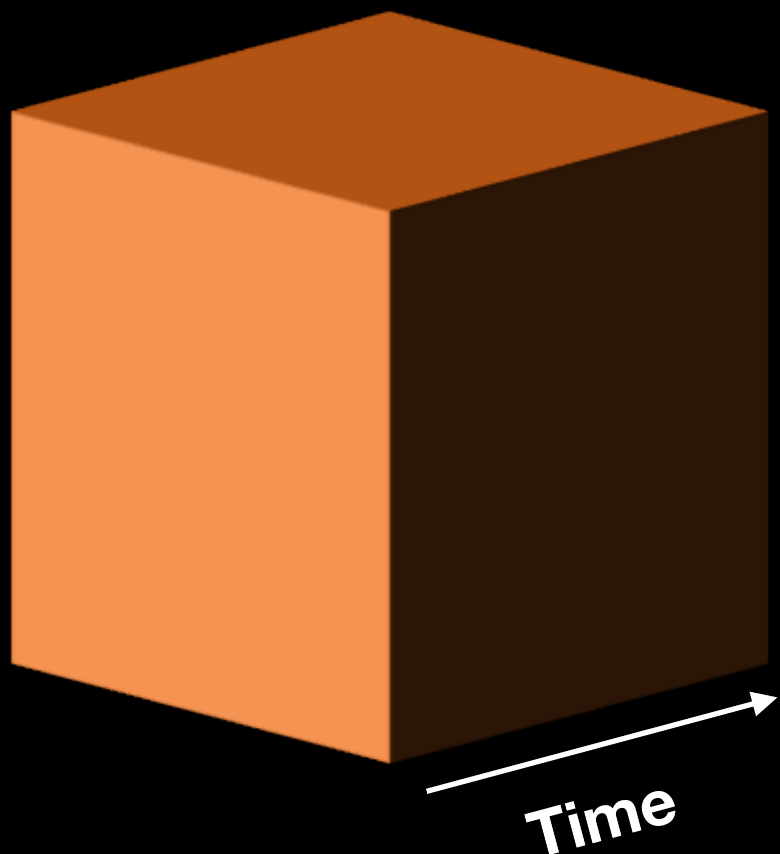
Outline

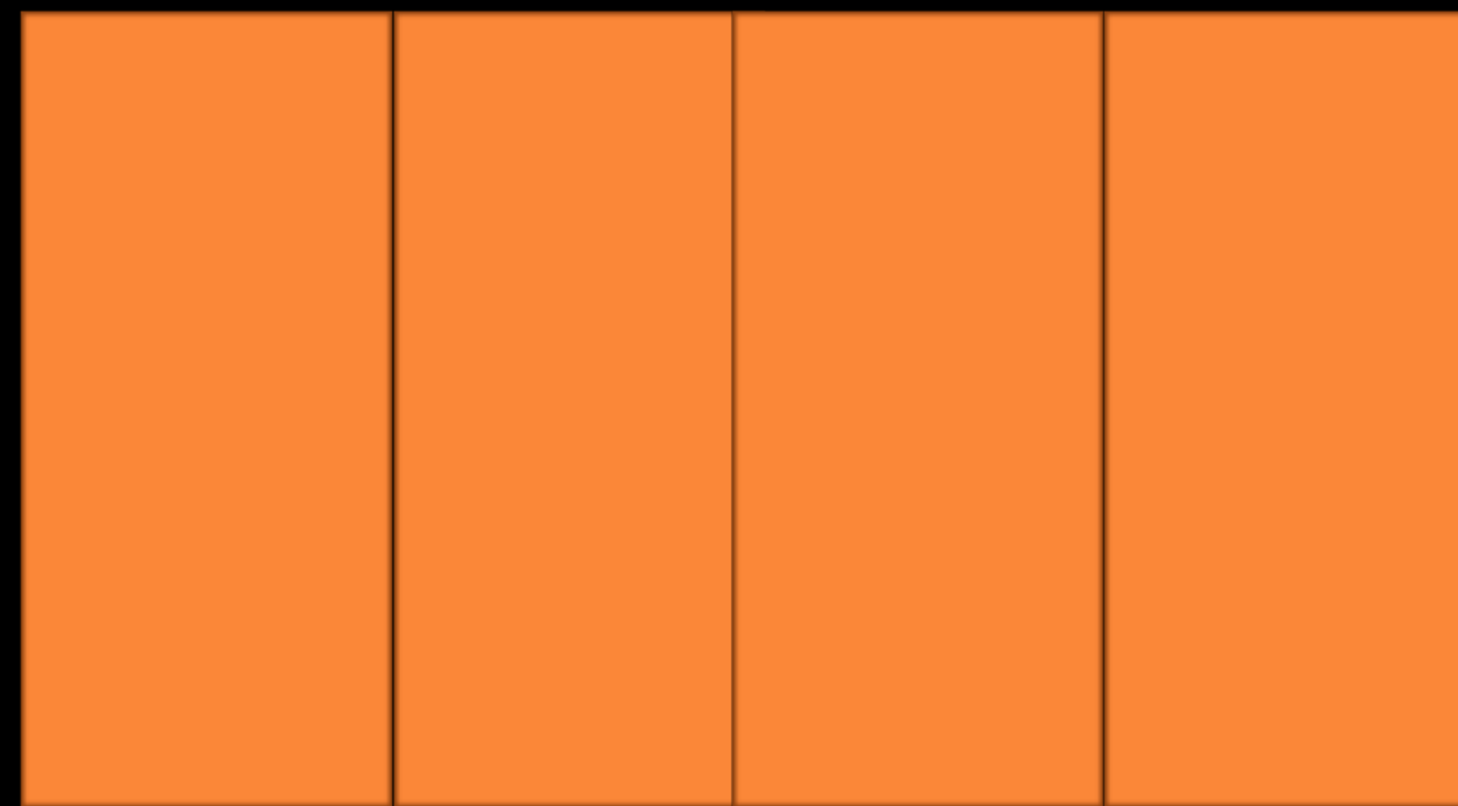
- Background
- Multi-step embedding of brain dynamics
- Hierarchical Coupled Geometry Analysis
- **Visualizing the PHATE of neural networks**



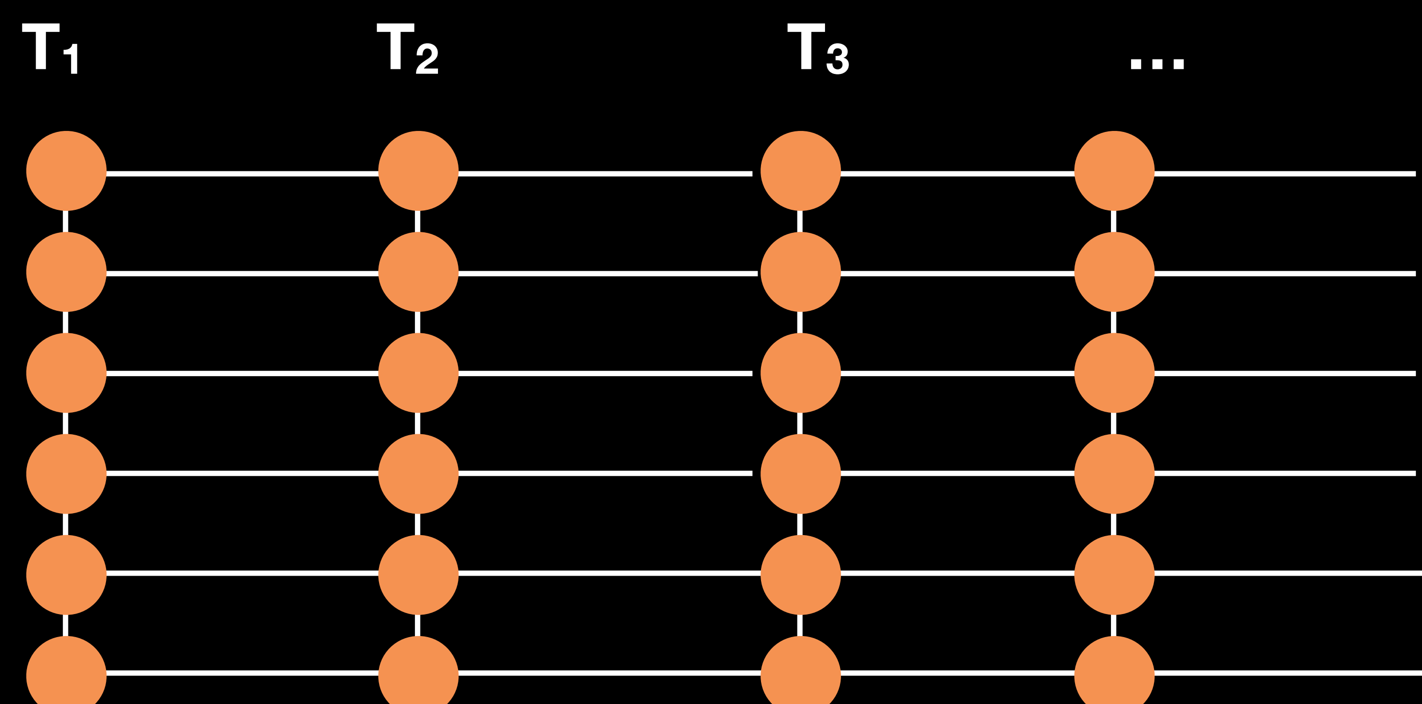
[Gignate, Charles, Krishnaswamy, Mishne, NeurIPS-2019]

Visualizing Dynamics

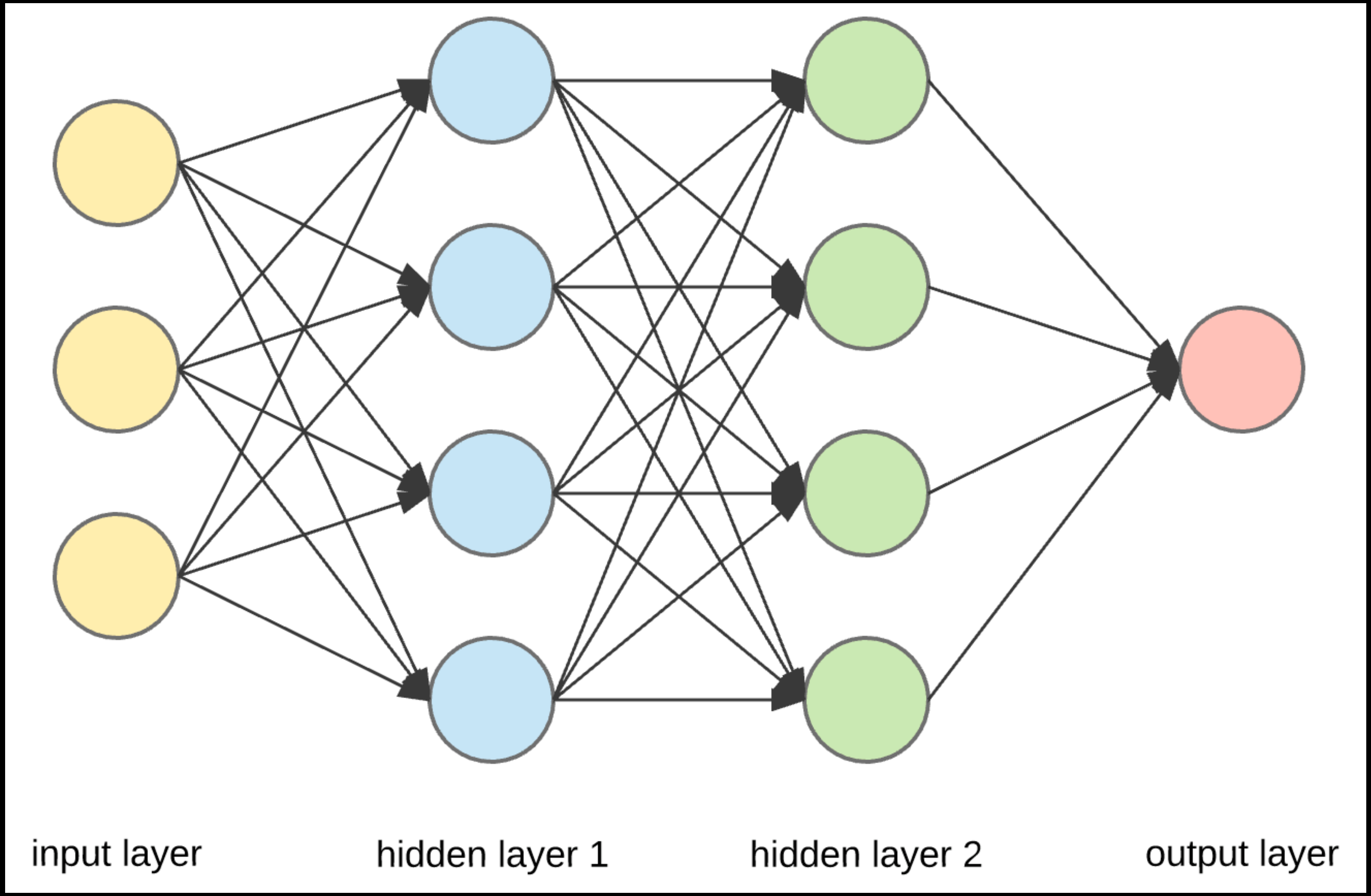




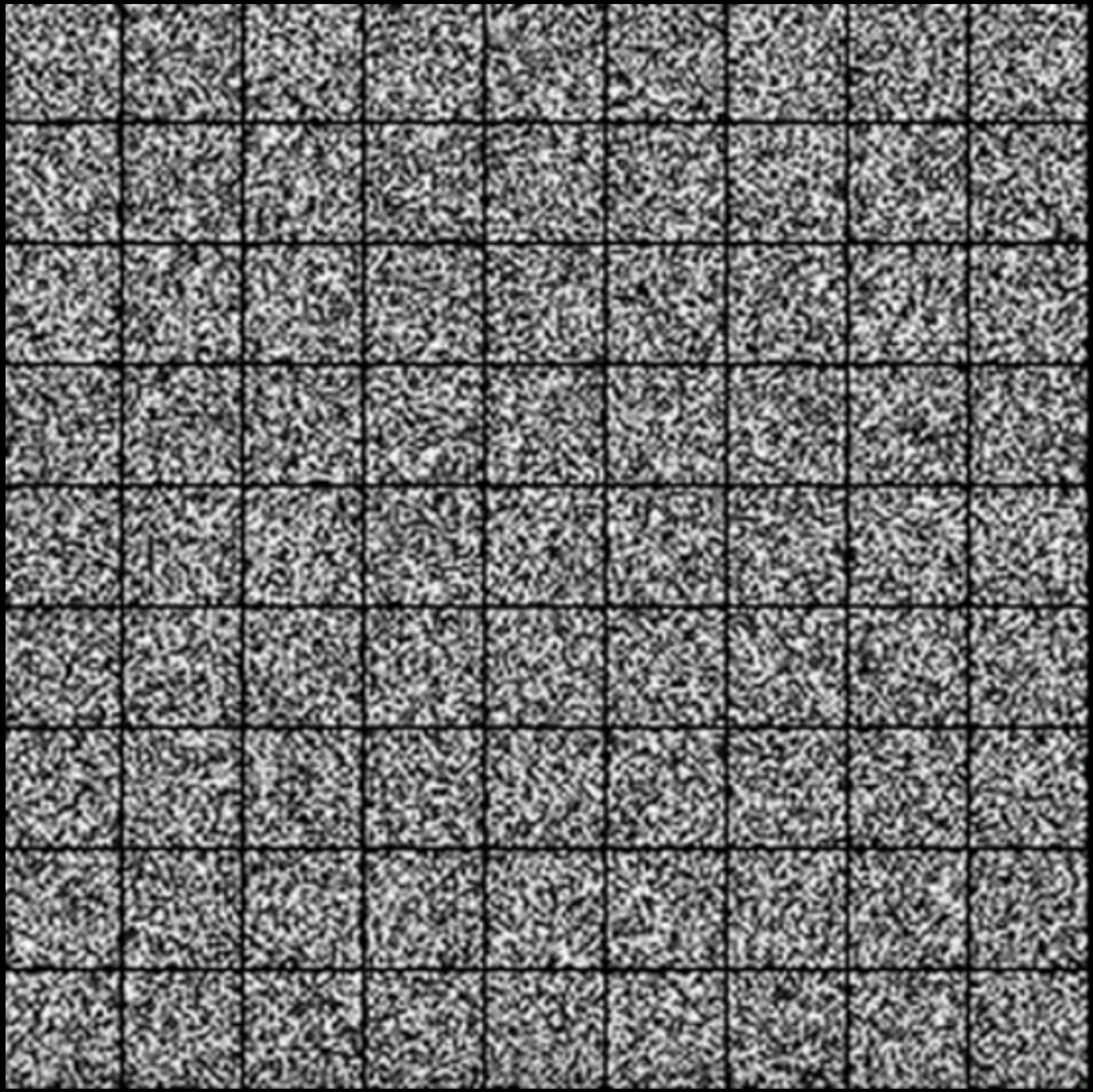
Impose structure with Multislice graph



Visualizations of learning

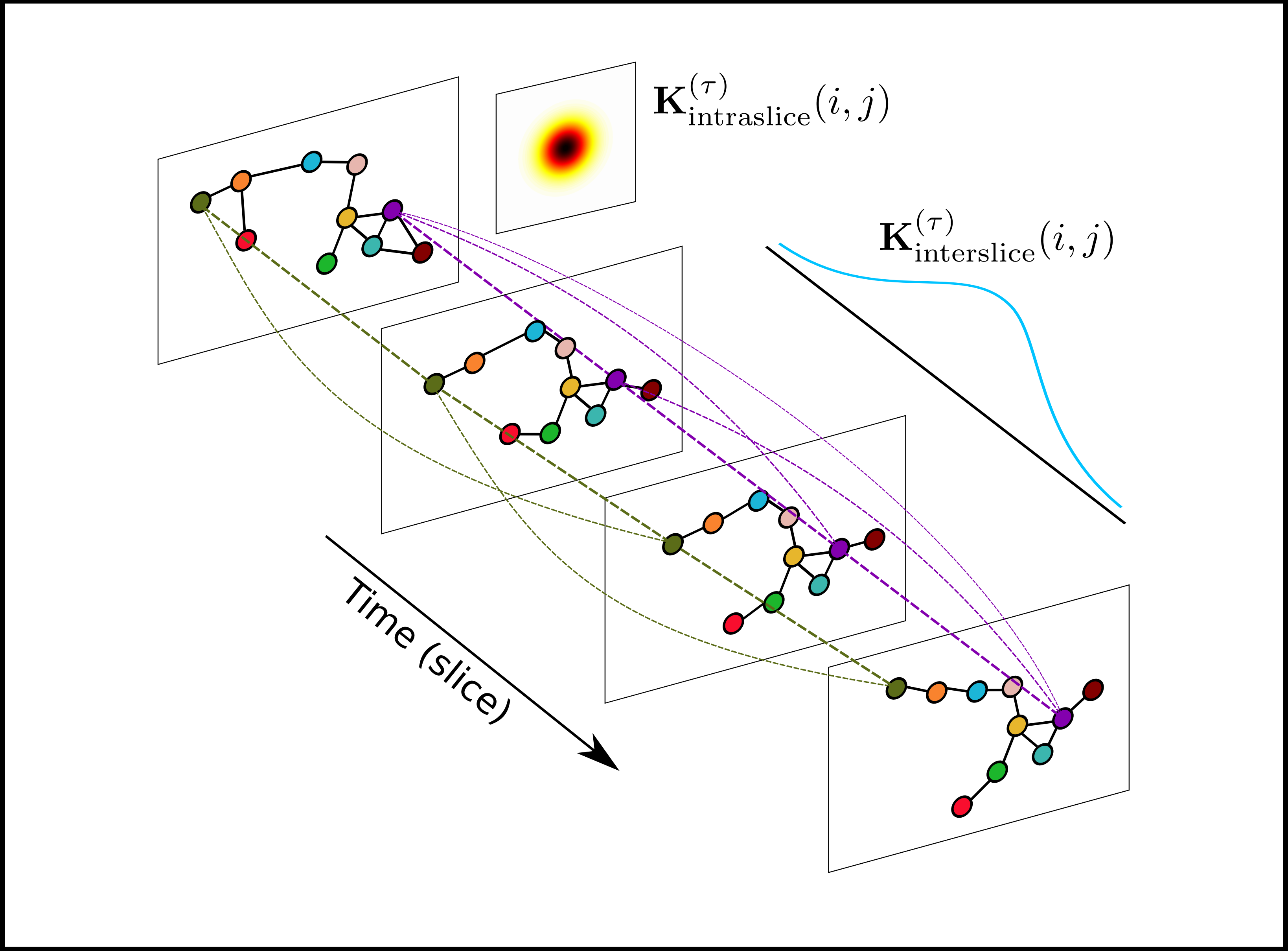


Visualizations of learning



Source: Youtube, Neural Network Weight Visualization - MNIST Dataset

Multi-slice graph

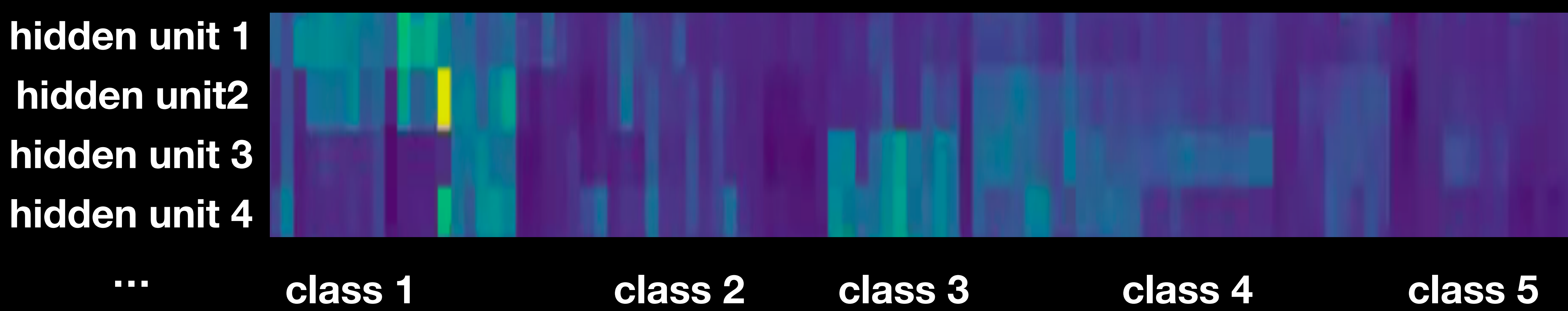


How to define similarity of hidden units within and across epochs?

Multi-slice graph

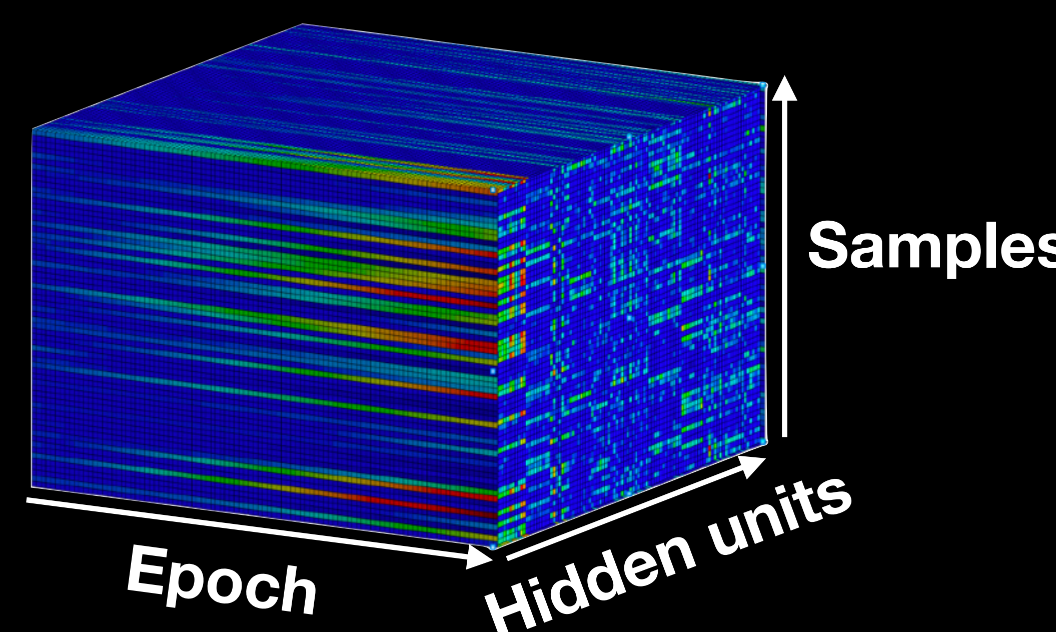
- Measure the activation of every hidden unit *i* to *p* training samples of dimension *d* at every epoch

$$F_i : \mathbb{R}^d \rightarrow \mathbb{R}$$

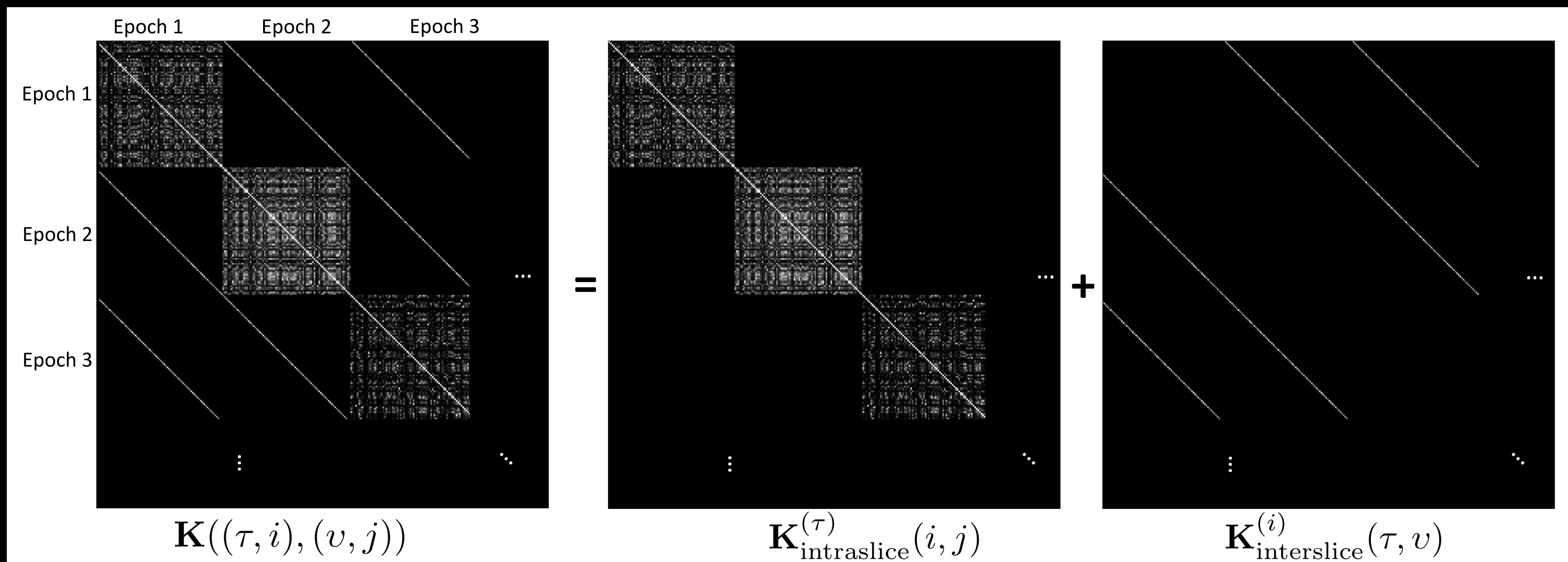


- Yields tensor $T \in \mathbb{R}^{n \times m \times p}$:

$$T(\tau, i, k) = F_i^{(\tau)}(Y_k)$$



Multi-slice Kernel



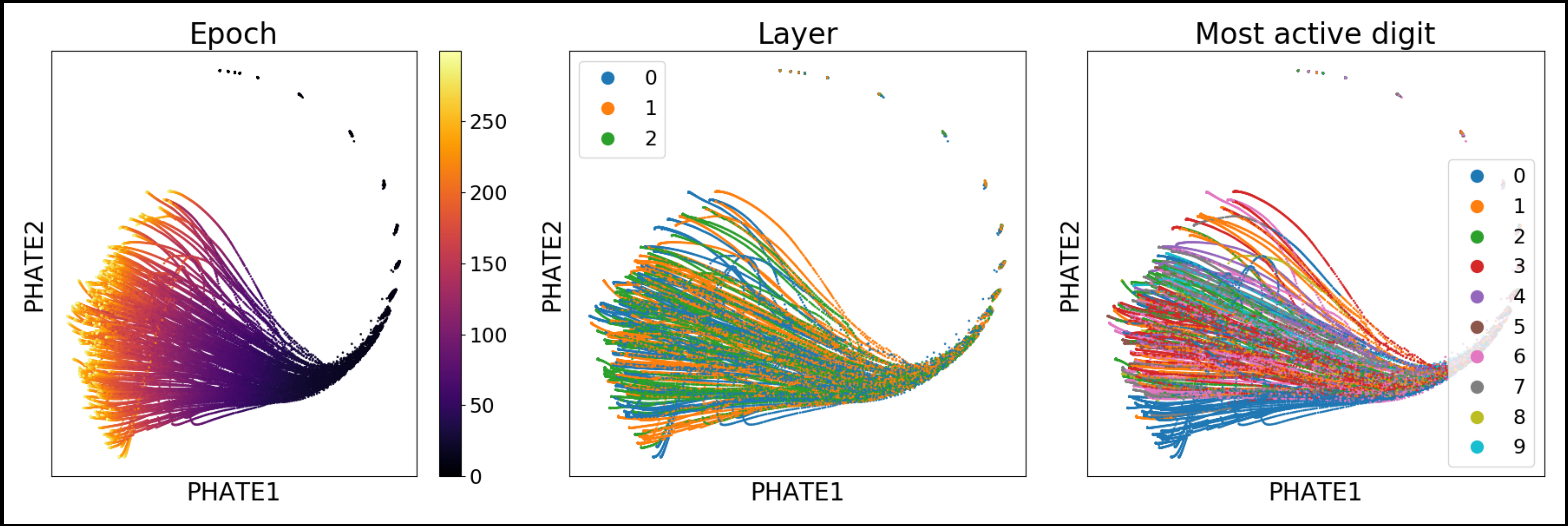
$$K^{(\tau)}_{\text{intraslice}}(i, j) = \exp\left(-\|T(\tau, i) - T(\tau, j)\|_2^{\alpha} / \sigma^{\alpha}_{(\tau, i)}\right)$$

$$K^{(i)}_{\text{interslice}}(\tau, \upsilon) = \exp\left(-\|T(\tau, i) - T(\upsilon, i)\|_2^{2} / \epsilon^{2}\right)$$

$$K((\tau, i), (\upsilon, j)) = \begin{cases} K^{(\tau)}_{\text{intraslice}}(i, j), & \text{if } \tau = \upsilon; \\ K^{(i)}_{\text{intraslice}}(\tau, \upsilon), & \text{if } i = j; \\ 0, & \text{otherwise.} \end{cases}$$

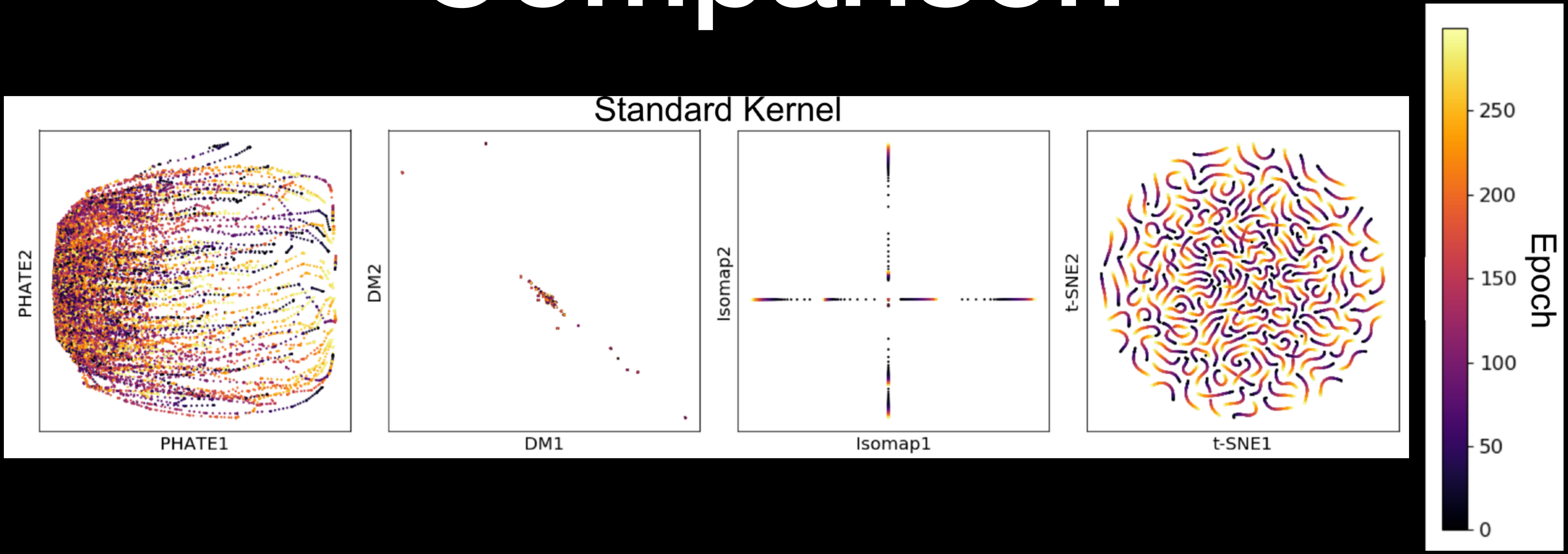
Visualizations of learning

Apply PHATE [Moon et al. 2017] to multiislice kernel

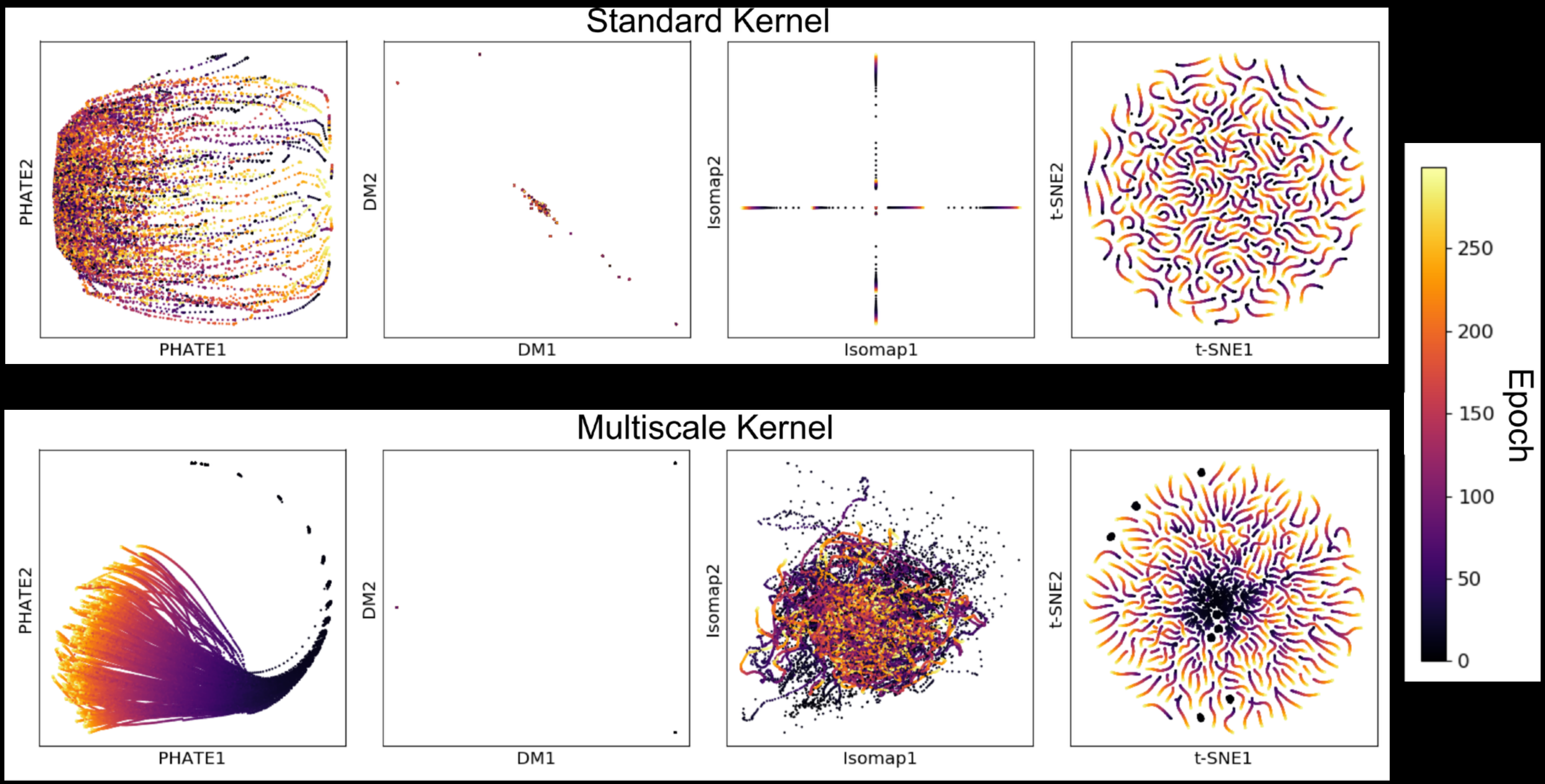


https://github.com/scottgigante/M-PHATE

Comparison

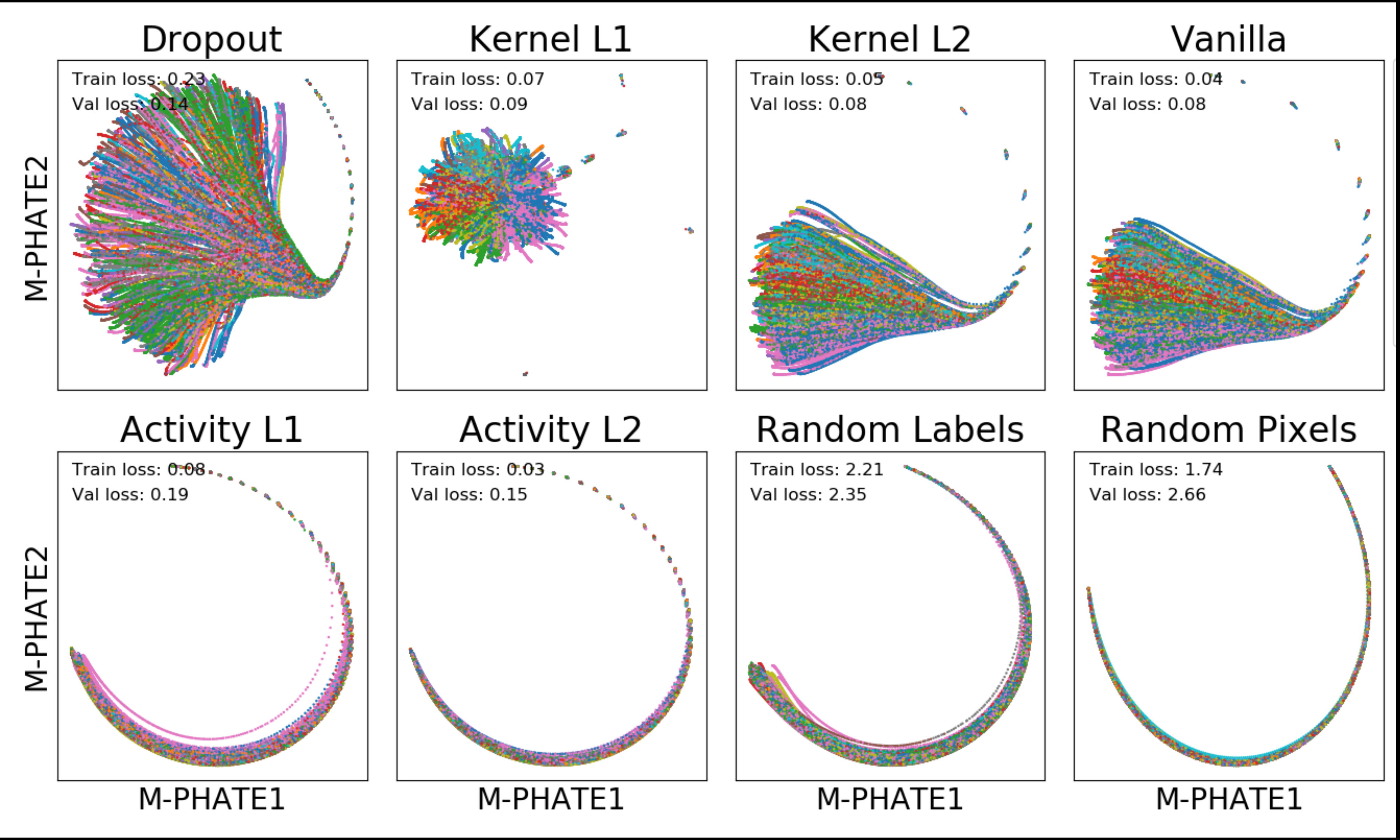

Standard Kernel
PHATE2
PHATE1
DM2
DM1
Isomap2
Isomap1
t-SNE2
t-SNE1
Epoch
250
200
150
100
50
0

Comparison



M-PHATE: multislice kernel + PHATE

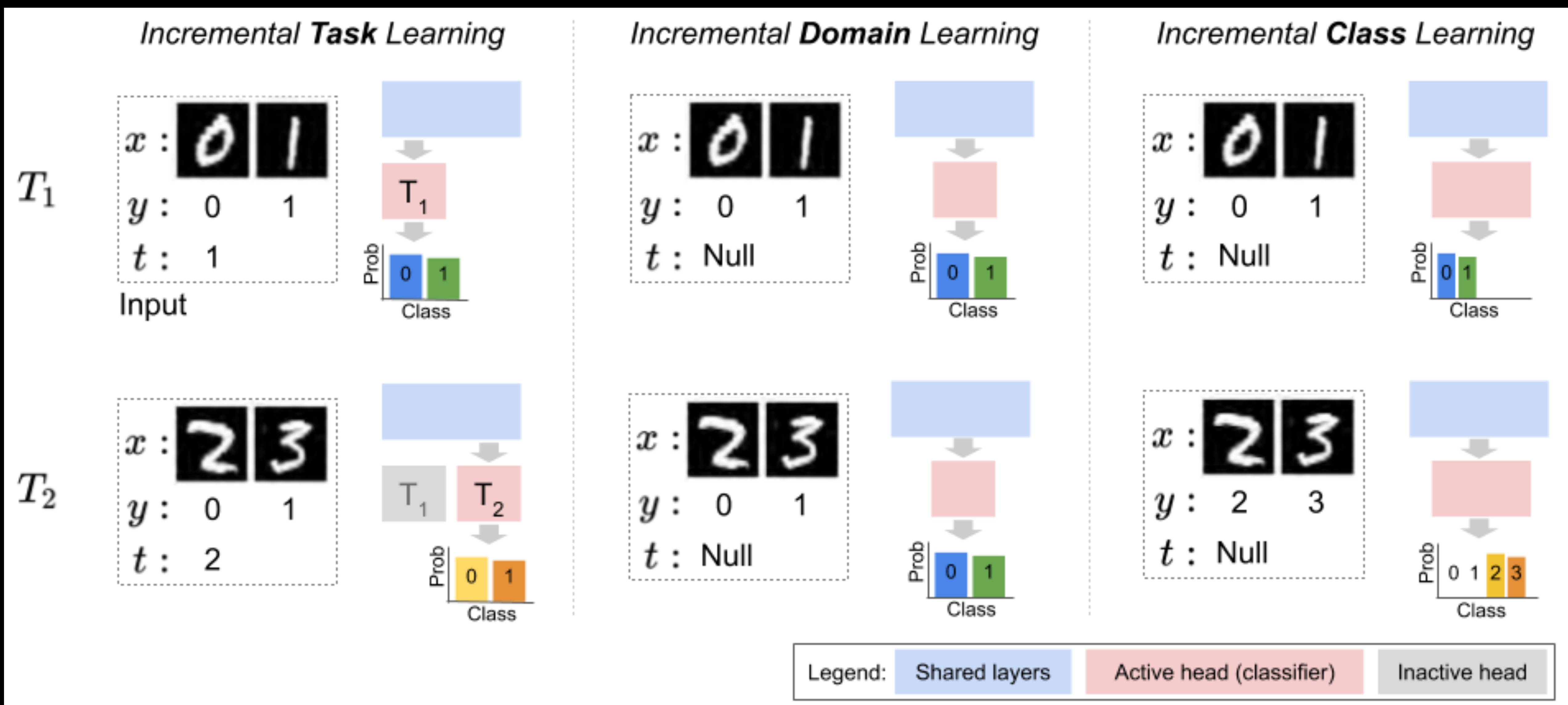
Generalization



memorization

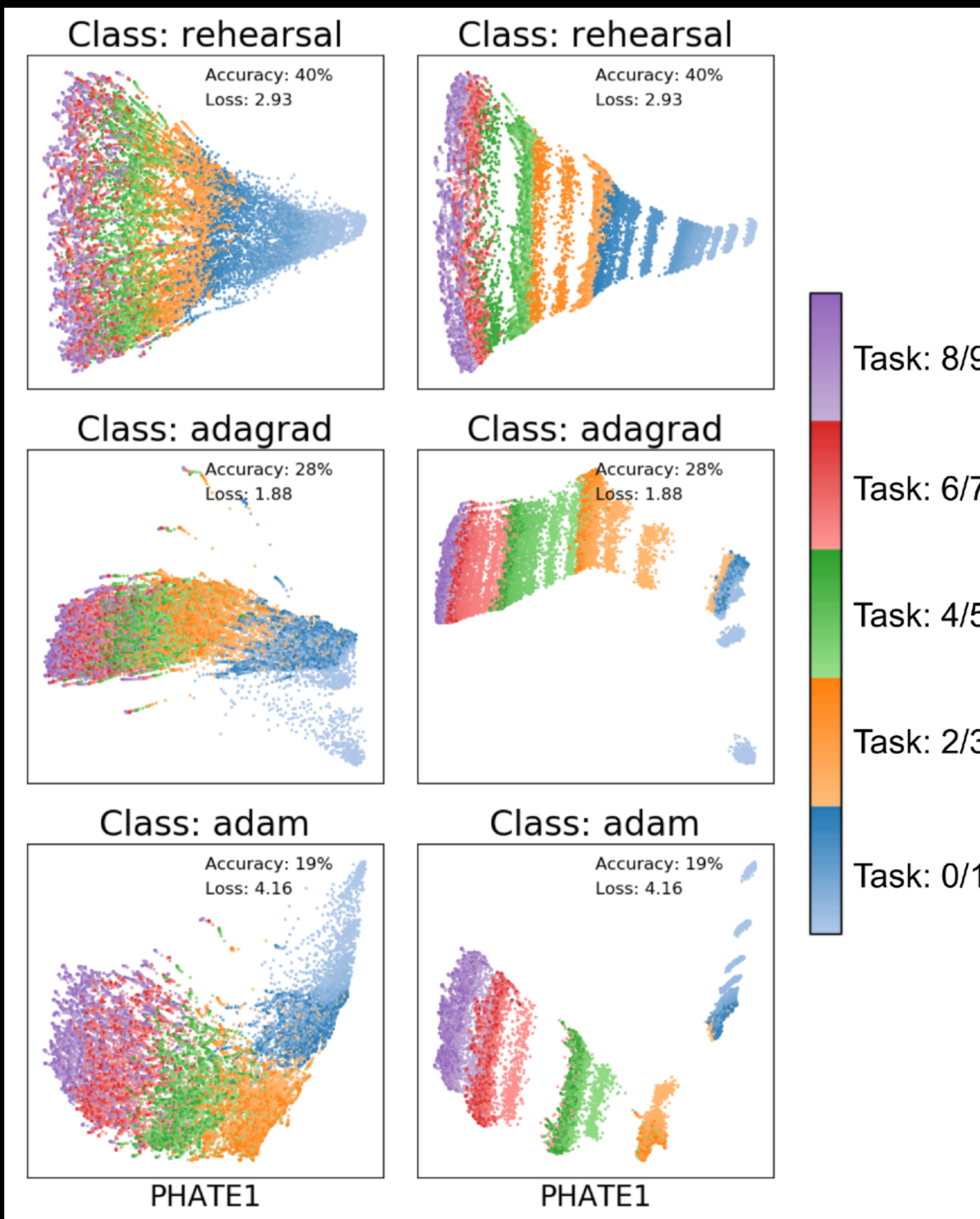
Continual learning

An ongoing challenge in artificial intelligence is in making a single model perform well on many tasks independently



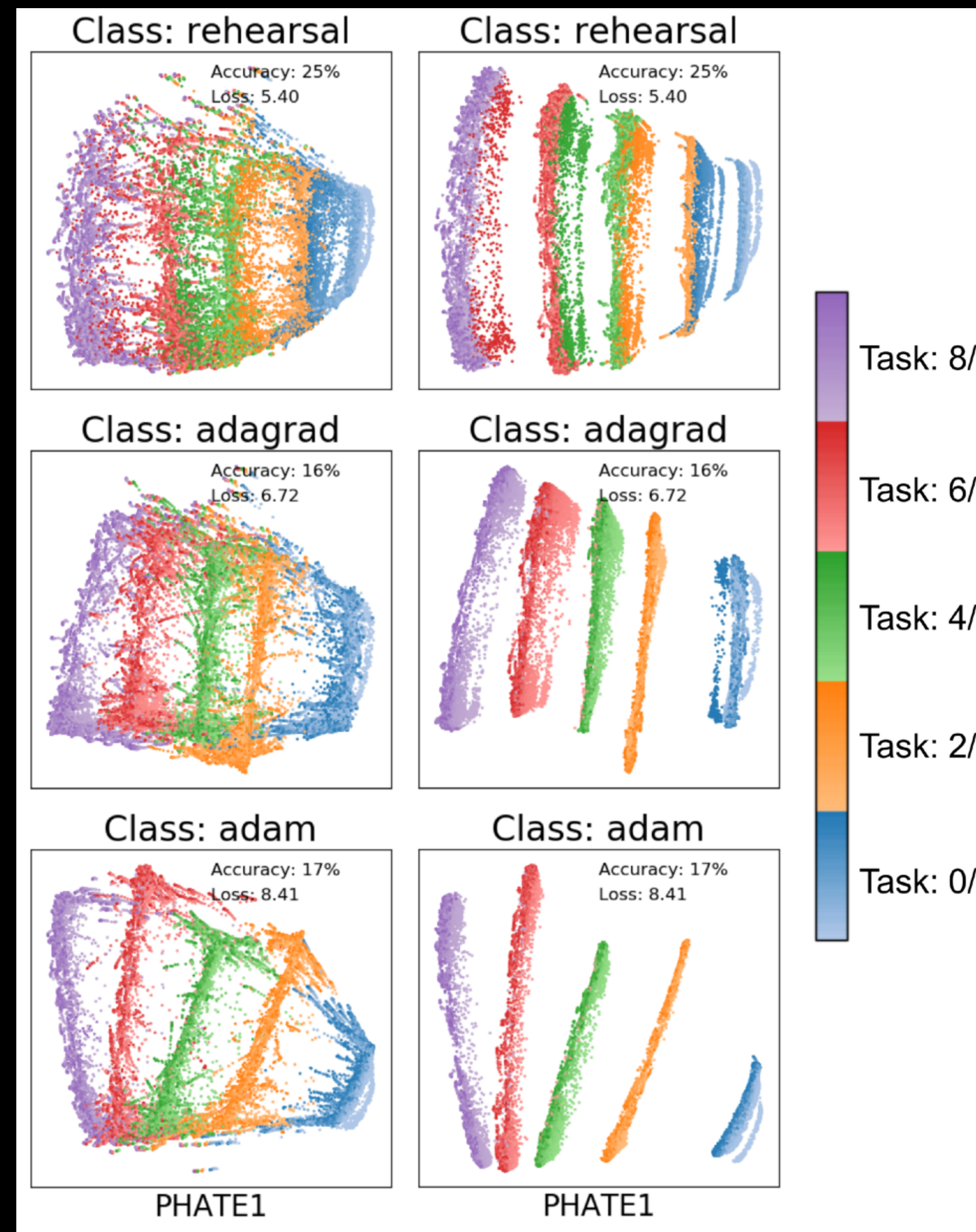
Continual learning

MNIST



Layer 1 **Layer 2**

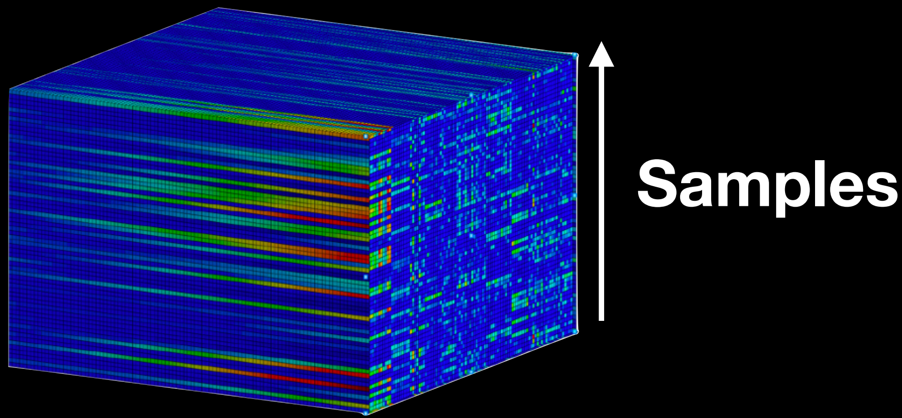
CIFAR-10

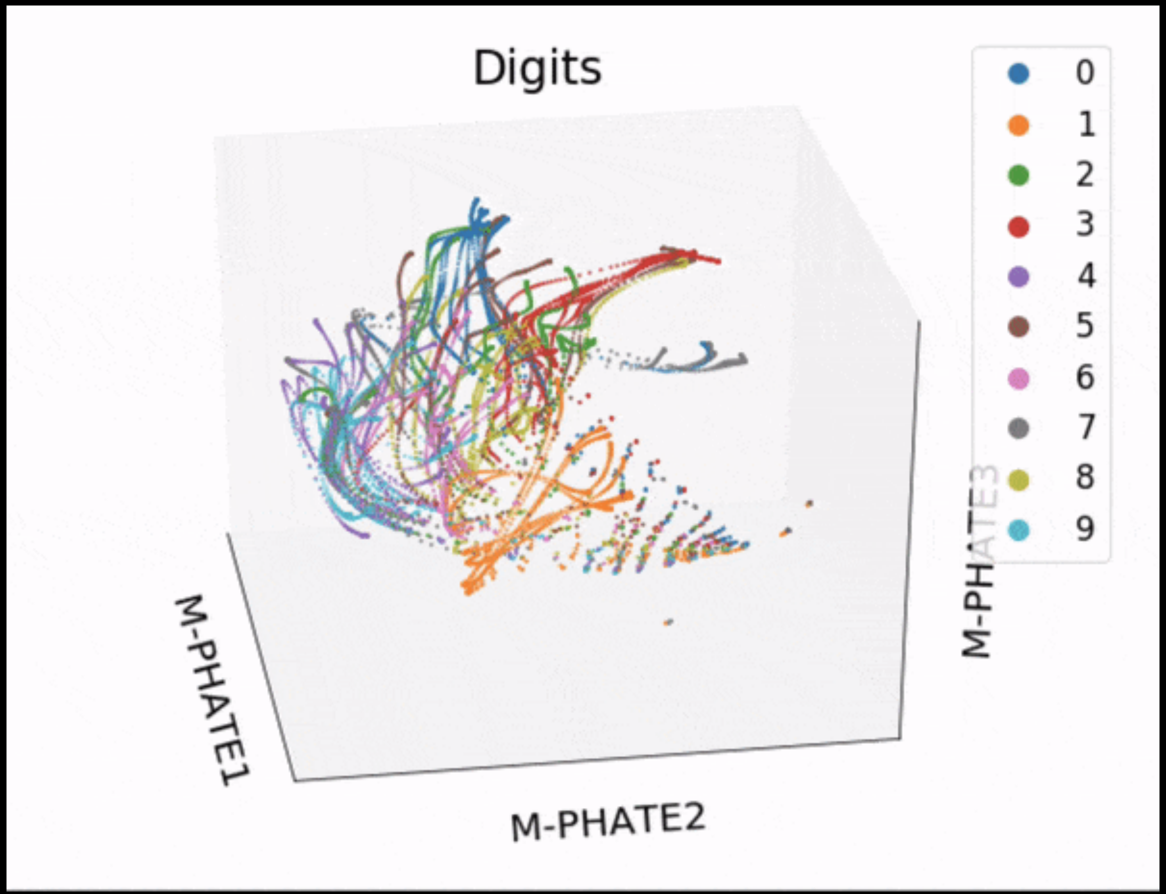


Layer 1 **Layer 2**

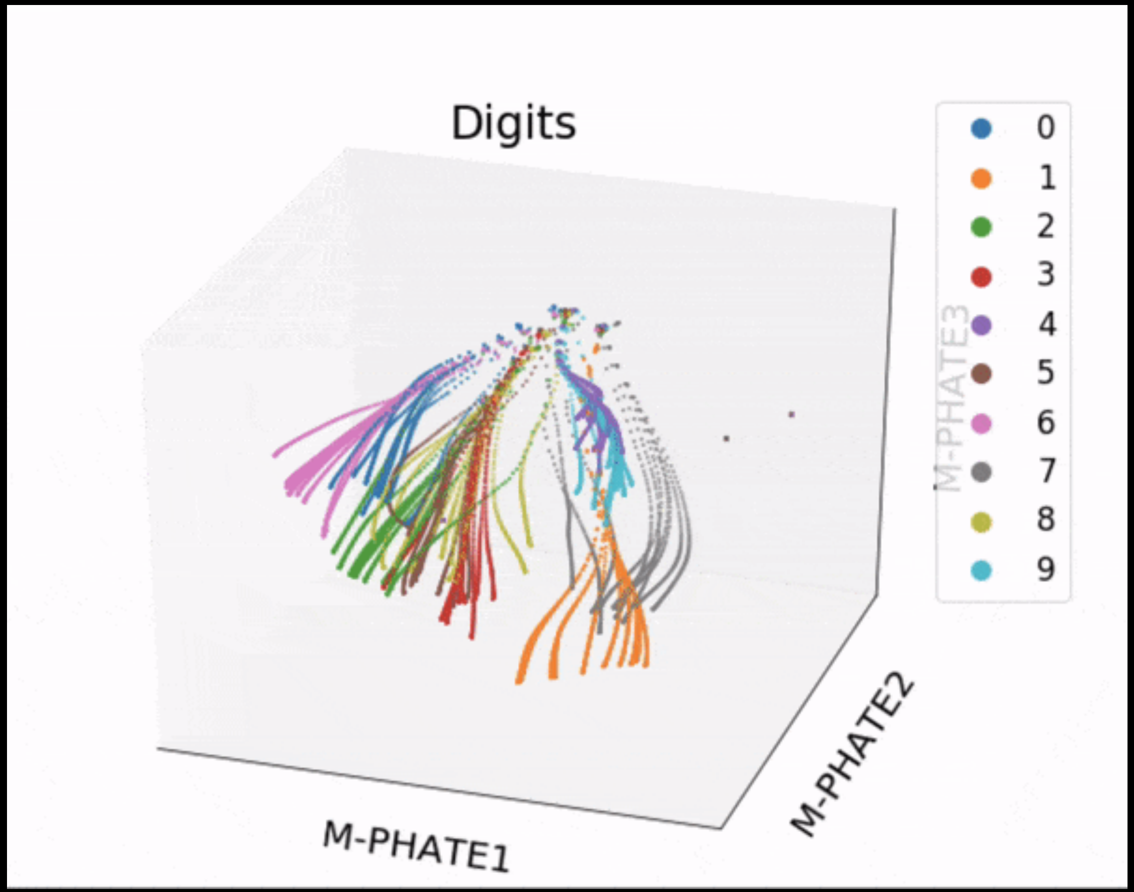
A different viewpoint

- Visualize evolution of samples as seen by the network





Autoencoder



Classifier

Acknowledgements

Yale
Raphy Coifman
Siyuan Gao
Dustin Scheinost
Scott Gigante
Smita Krishnaswamy
Reuma Gadassi Polack
Jutta Joormann

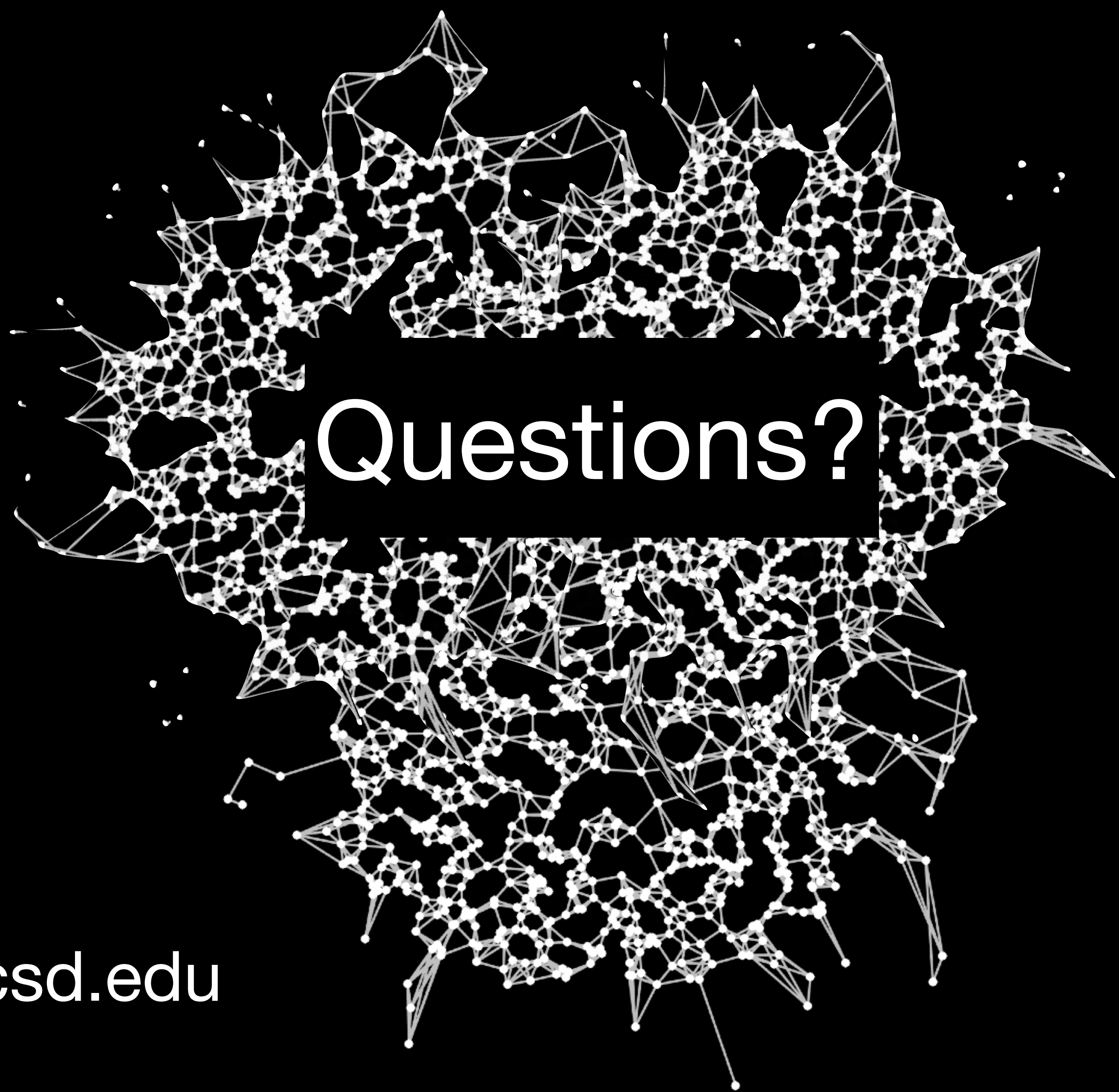
Technion
Ronen Talmon
Israel Cohen
Ron Meir
Jackie Schiller

Princeton
Adam Charles

Ghent University
Jonas Everaert

Funding: NIH

Thank you



gmishne@ucsd.edu

@gmishne