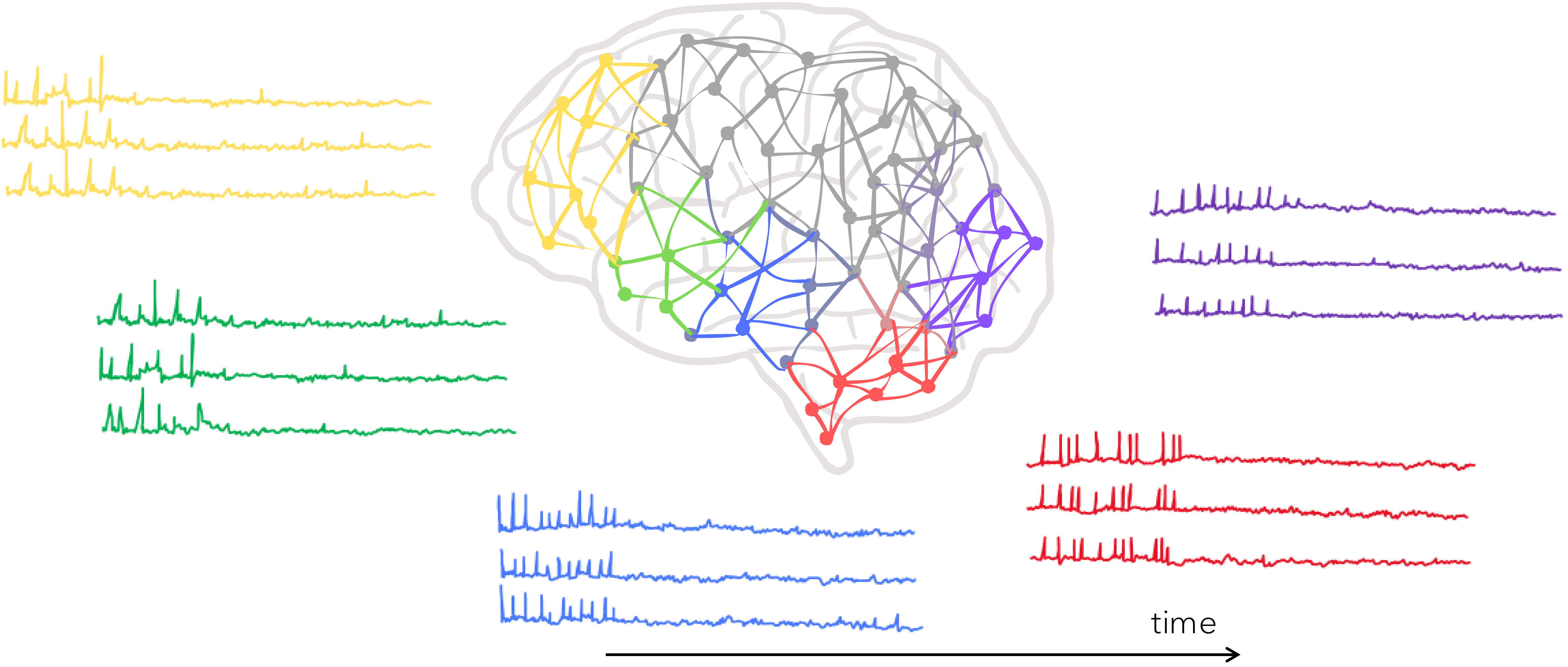
# Modeling Dynamics on a Canonical Neural Manifold in C. *elegans*

Liana Akobian, BSc

Vienna University of Technology
Faculty of Informatics (Data Science)
University of Vienna
Dept. of Neurosciences and Developmental Biologies

**Supervisors**
Univ.-Prof. Dipl.-Ing. Dr.techn. Peter Filzmoser
Univ.-Prof. Dr. Manuel Zimmer
Charles Fieseler, MSc. PhD

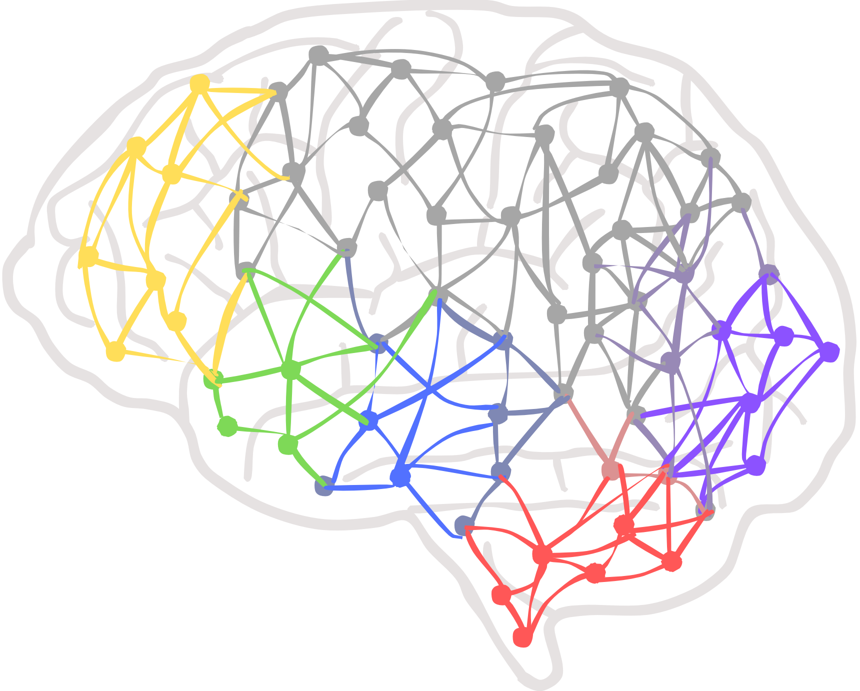
# The unit of computation is the neural population

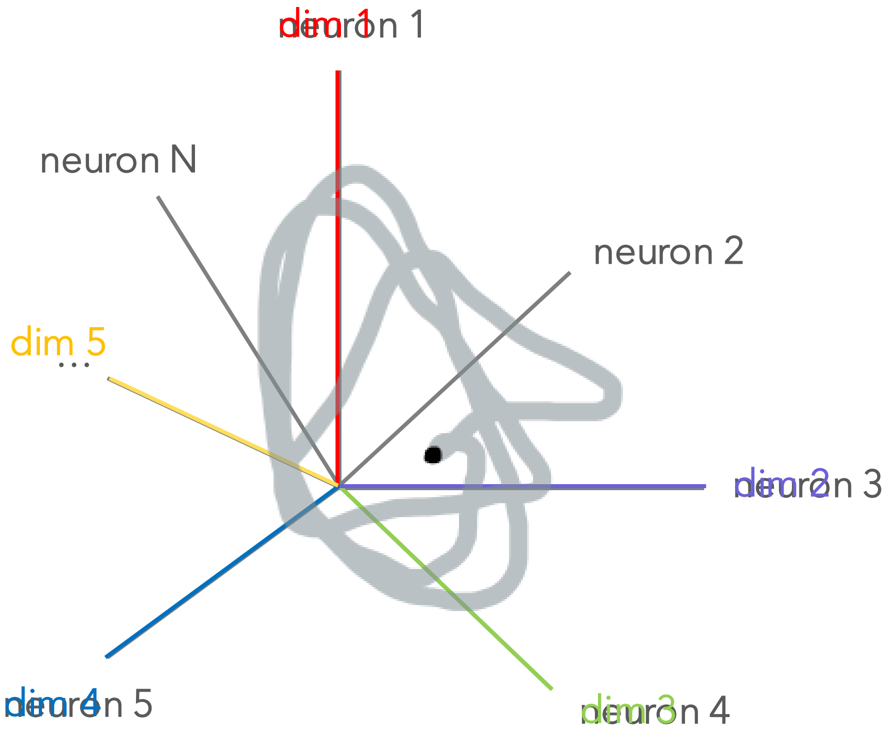



neural population =
a collection of neurons with correlated activity

neural dynamics = interaction of neurons over time

# From the population doctrine to a reduced neural state space






neural manifold

# Neural state evolution is given by dynamical functions

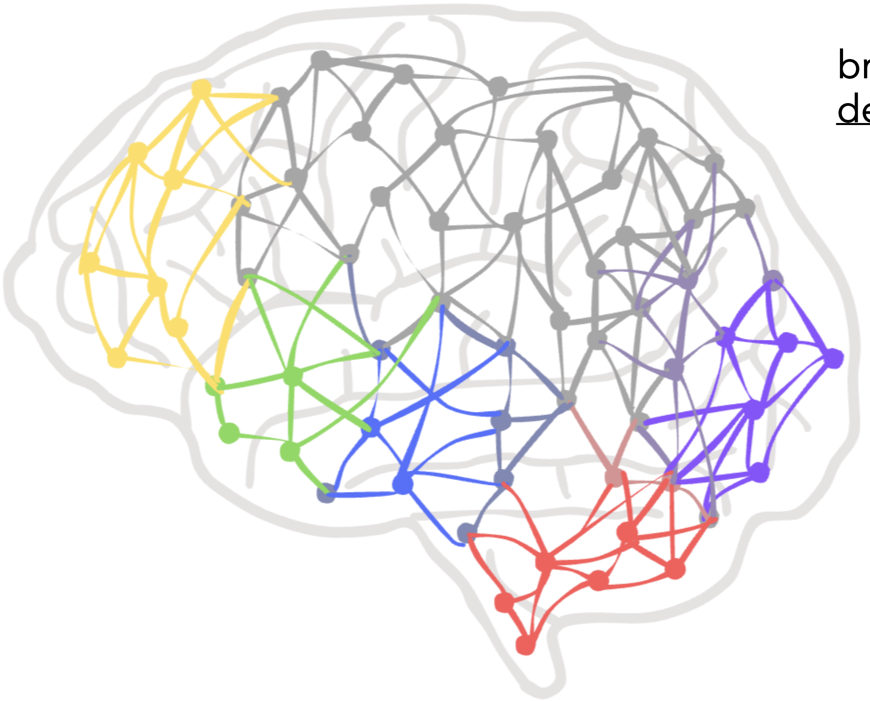
**well-studied**

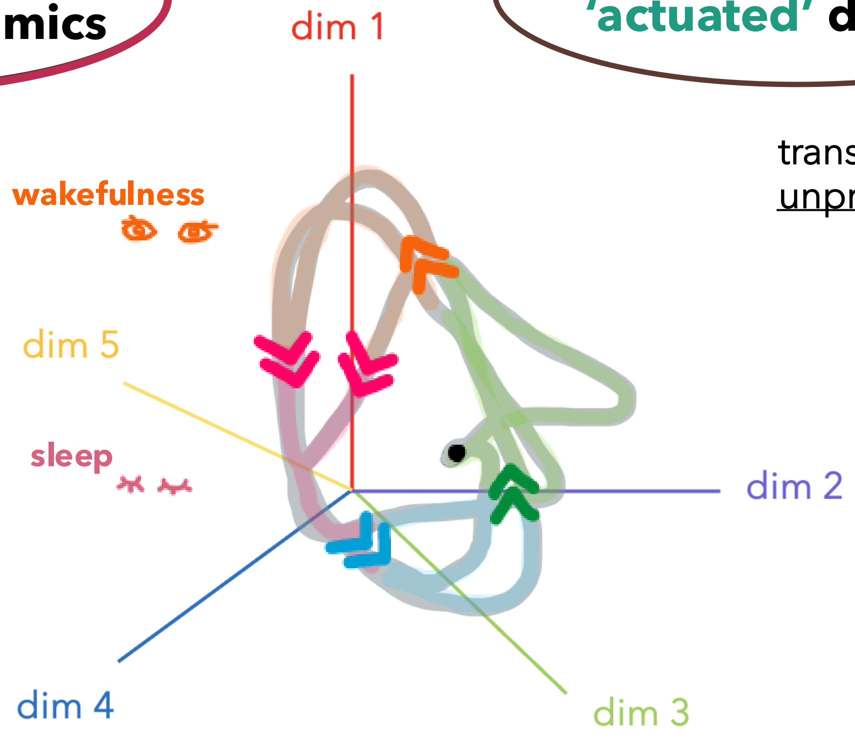
**autonomous**
**'intrinsic' dynamics**

brain is moved in a
deterministic way

**non-autonomous**
**'actuated' dynamics**

transitions are
unpredictable!






neural manifold displays
latent states

# How can we build a model of these dynamical functions?

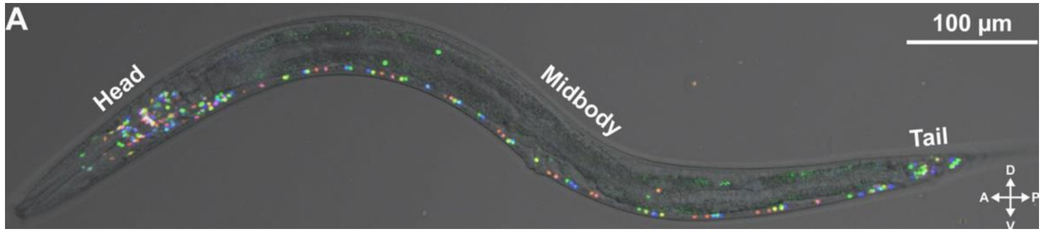
How can we capture non-autonomous, actuated dynamics alongside the autnomous ones?

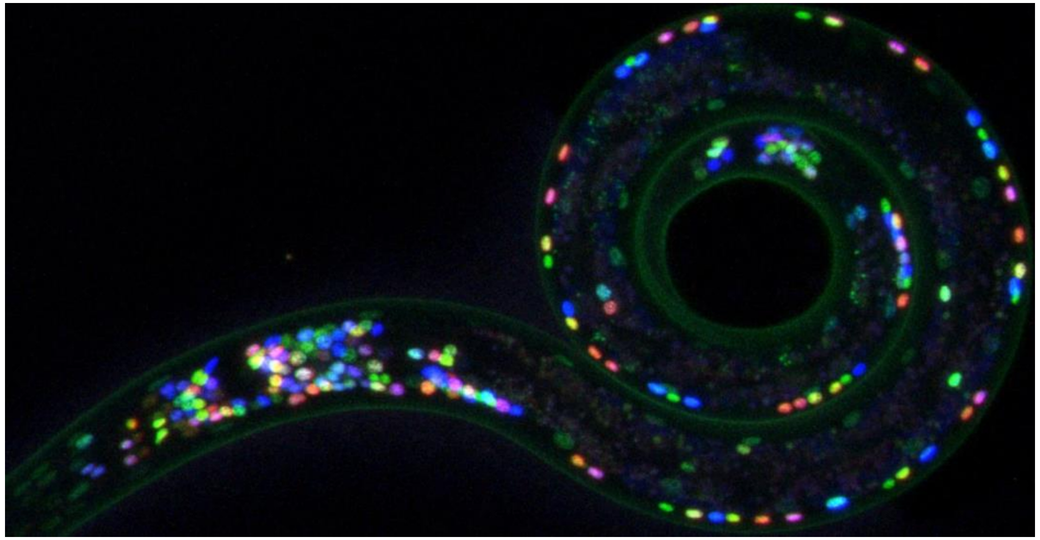
What are the steps required to create a neural manifold that is consistent across individuals?

Getting whole brain recordings is challenging!

We know a lot about C. elegans:
Connectome, Biophysical properties,..

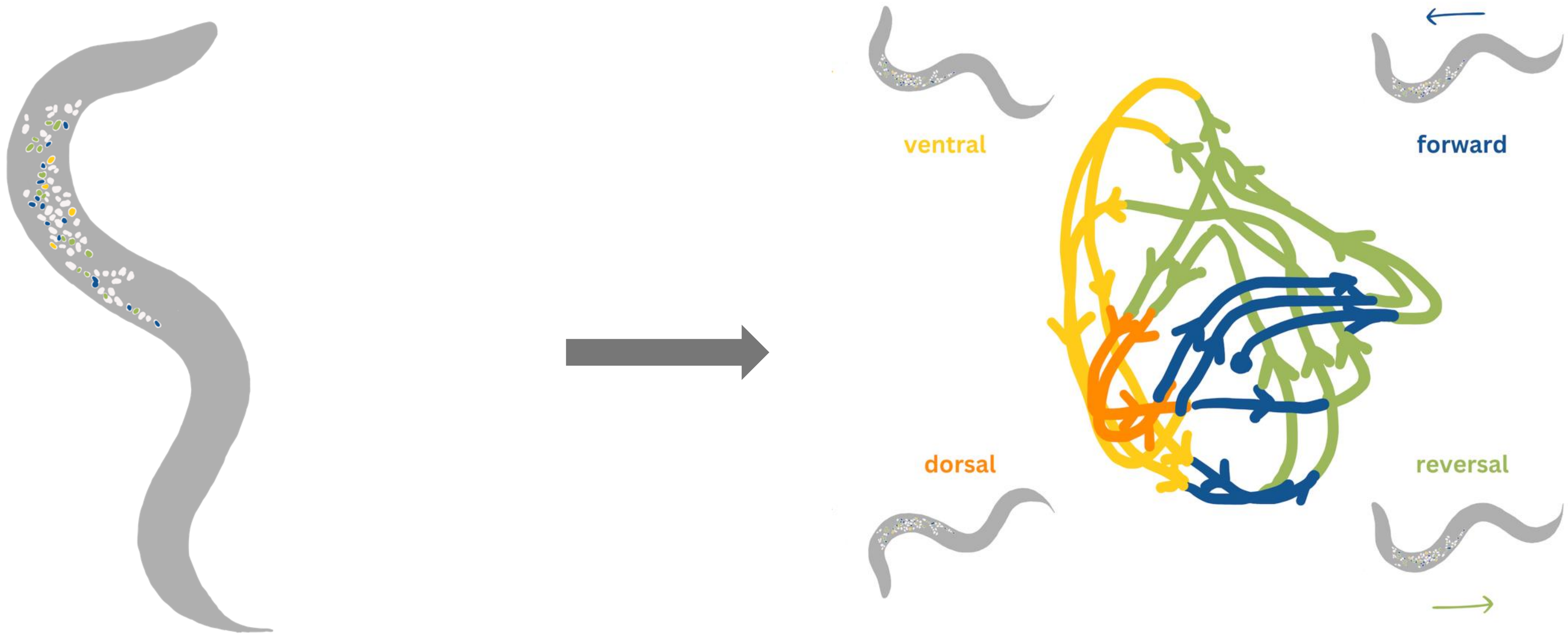
*Horbert Lab*



*Caenorhabditis elegans* = a nematode with 302 fully mapped neurons

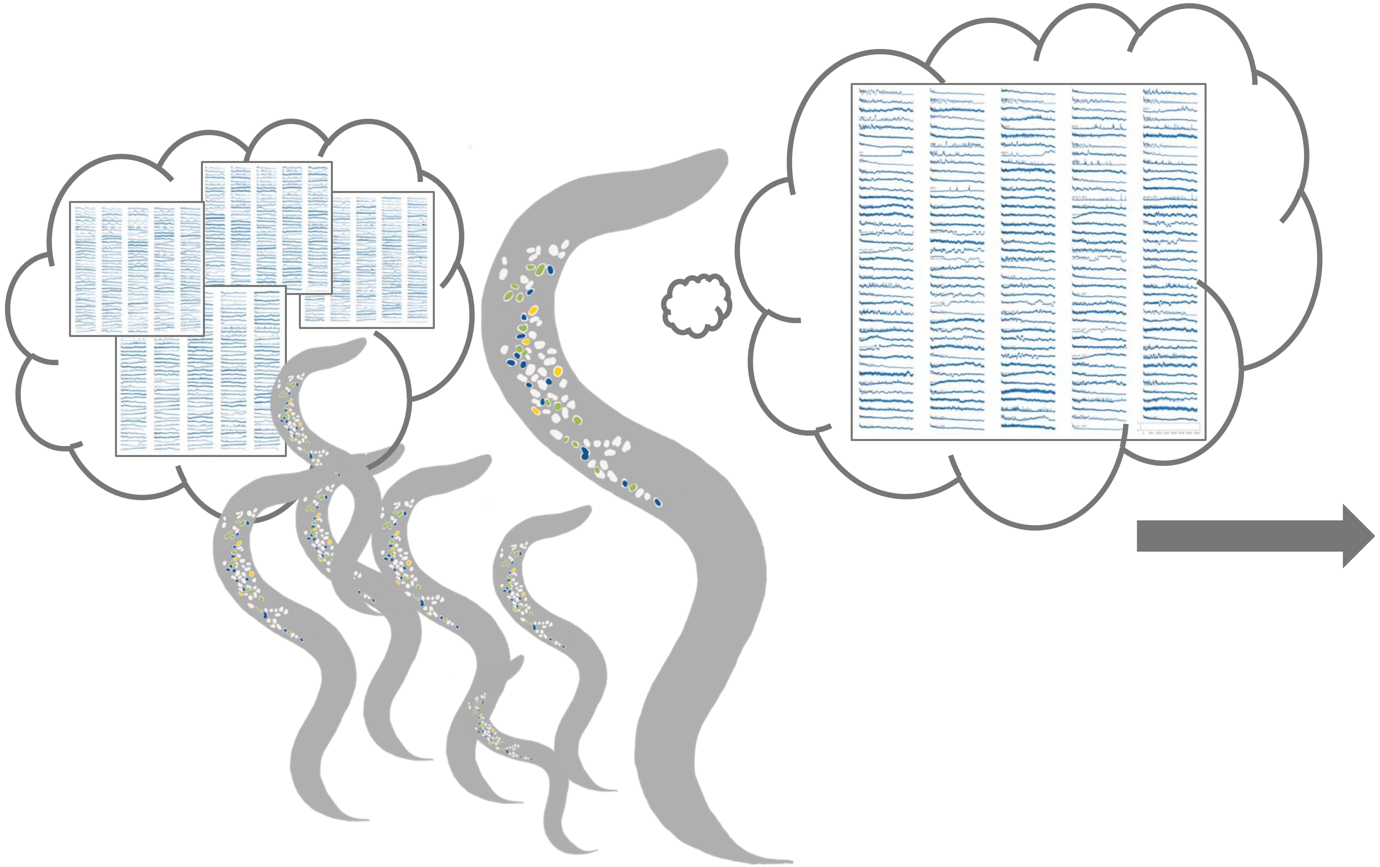
# In search of a neural manifold for C. *elegans*




Major behaviors of *C. elegans* are embedded in low-dimensional manifold structures.
*Adapted from Kato et al., 2015.*

Does this structure exhibit properties that apply 'universally' to all C. *elegans* individuals?

# What if we align whole brain recordings to construct a canonical neural manifold?





Thanks to Rebecca Kresnik and Kerem Uzel for the data!

Data Alignment opens a can of worms due to inconsistencies in..

→ frame rate
→ settings of laser power
→ number of neurons identified

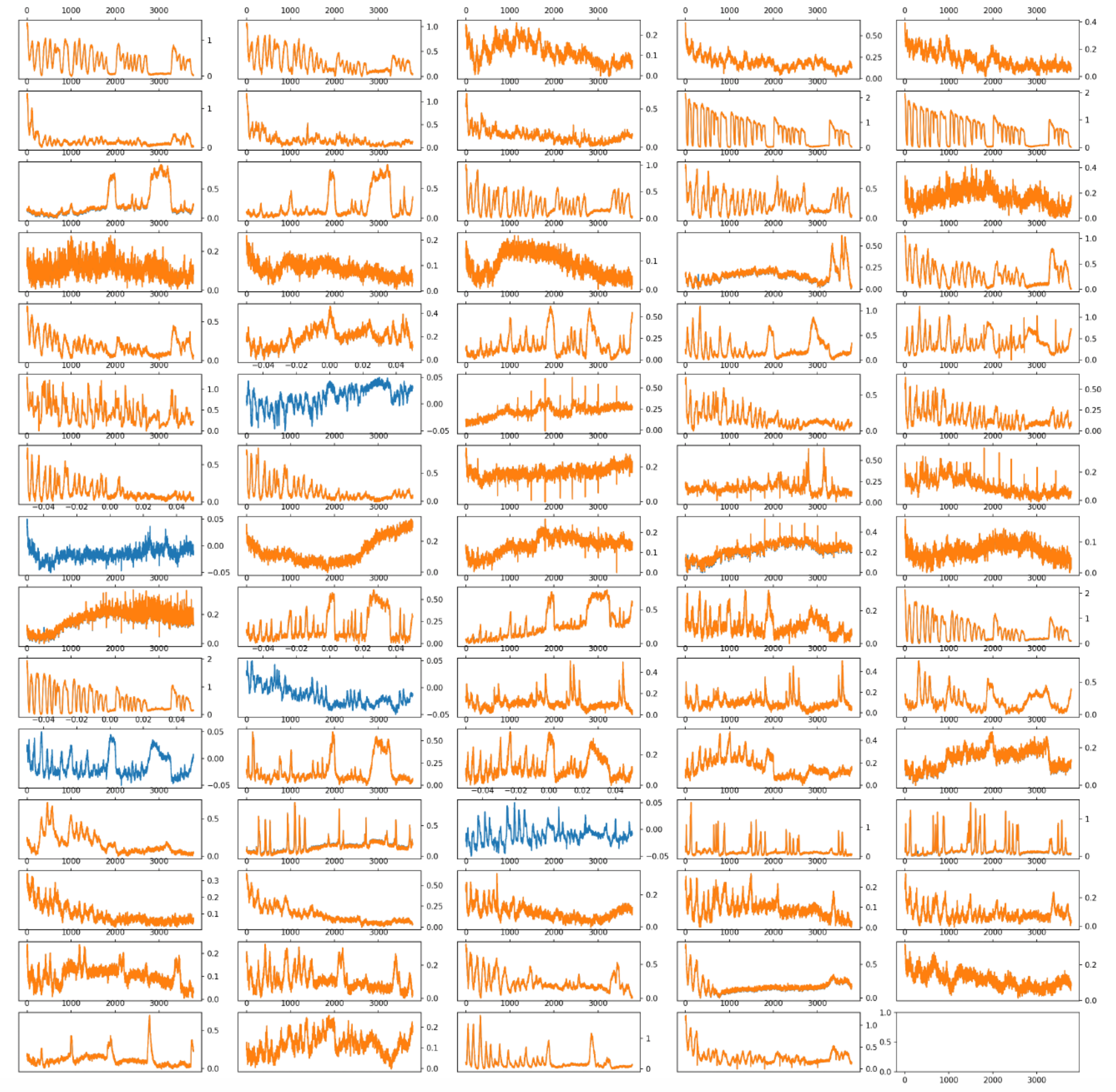
# Methods for Data Preprocessing

Up- and Downsampling

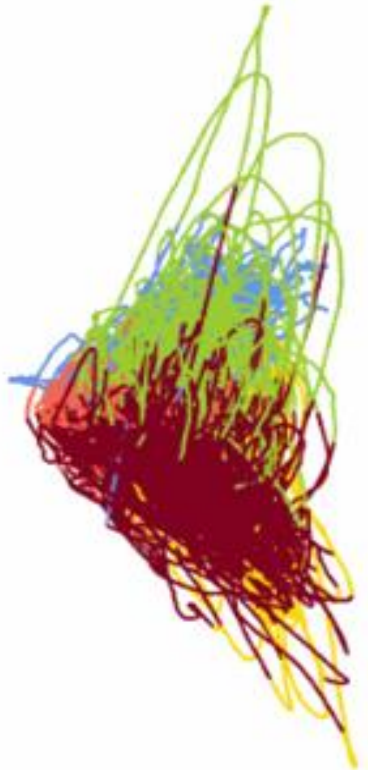
Quartile Normalization

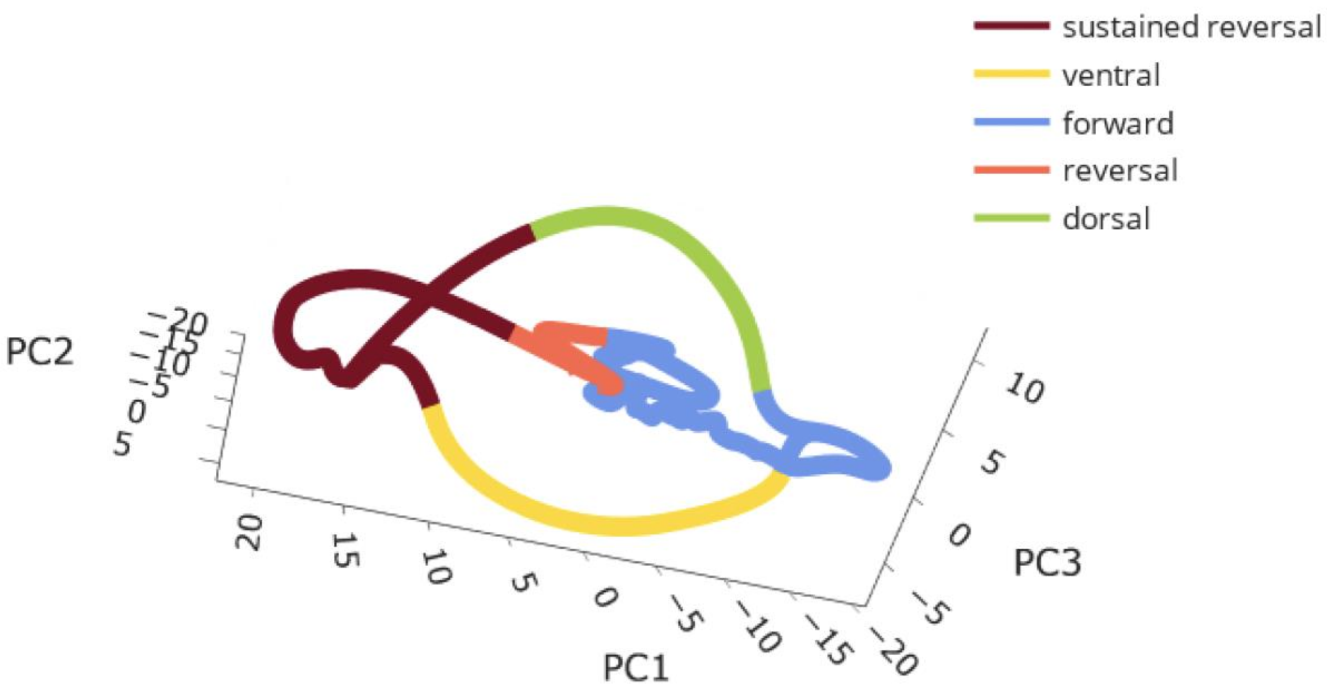
Data Imputation

**Probabilistic Principal Component Analysis**



# A PCA-based low-dimensional representation of 23 whole-brain recordings shows a manifold-like structure

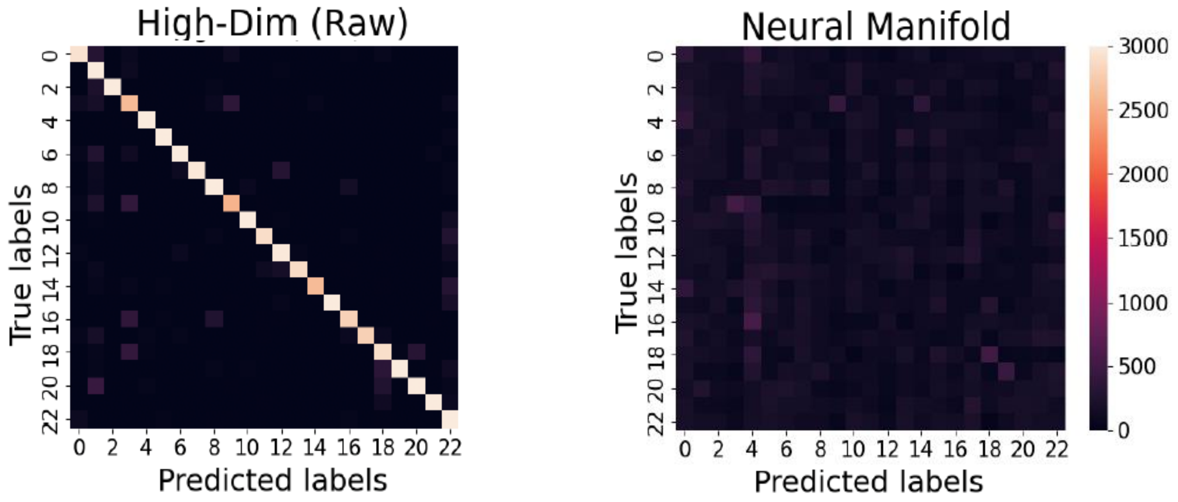





the manifold can be separated by behavior..
and should not be separated by individuality

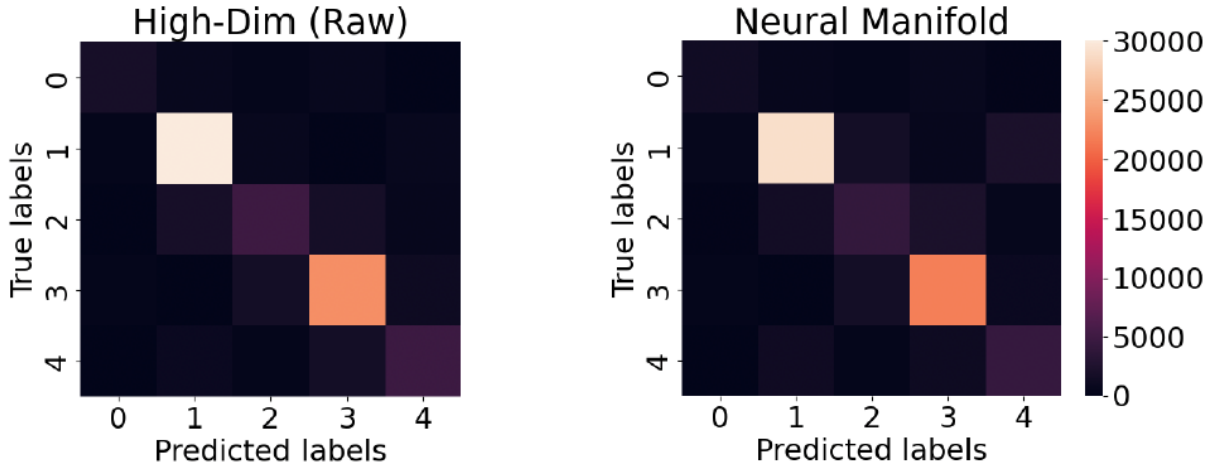
# While the neural manifold shows decreased variability caused by individuality..

dataset membership




# it still successfully separates behavior states

behavior state annotation



# Many existing models of the (non-) autonomous dynamics

**Dynamical Systems**

$$x_{t+1} = Ax_t$$

**simple**

interpretable

**rSLDS** (Linderman et al. 2016)

**DMDc** (Fieseler et al. 2020)

**We are here**

**LFADS** (Sussilo et al. 2016)

**iLQR-VAE** (Schimel et al. 2022)

**Neural Networks**

$$f\left(b + \sum_{i=1}^{n} x_i w_i\right)$$

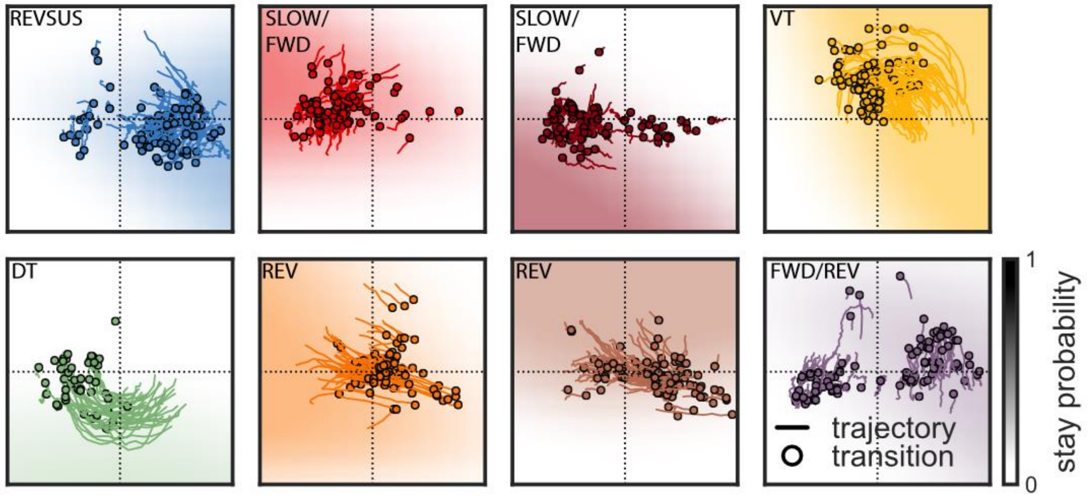
**complex**

uninterpretable

**DKF** (Krishnan et al. 2015)

# Dynamical Systems Models

latent subsystems don't overlap in time?



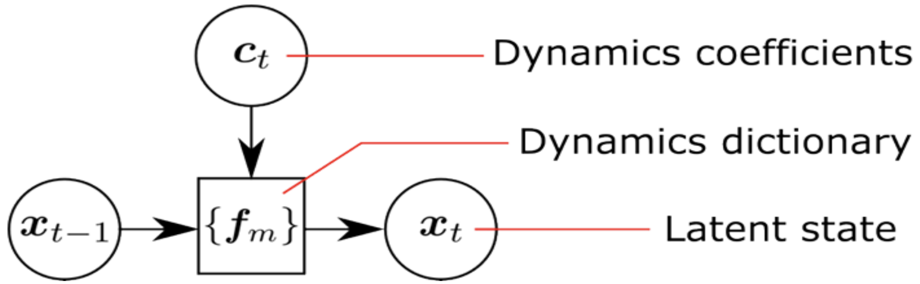

*Hierarchical recurrent state space models reveal discrete and continuous dynamics of neural activity in C. elegans, Linderman et al.*

Complex nonlinear dynamics can be broken into simpler linear dynamics

A linear dynamical system corresponds to a latent state

Mudrik et al. (2022): **neural activity at a given time point is given by a time-varying <u>linear combination of these linear dynamical systems</u>**

**Decomposed Linear Dynamical System (dLDS)**

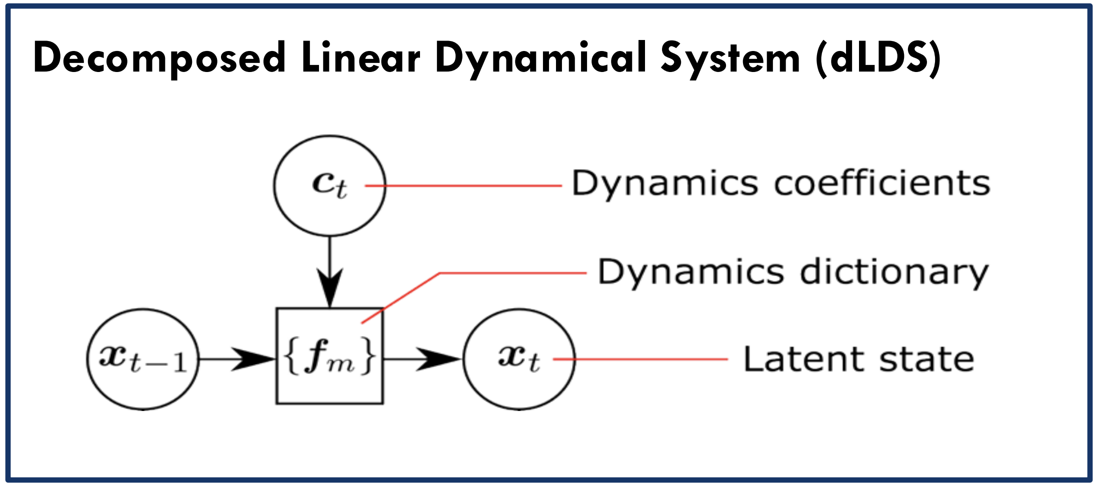


# The good and the bad of dLDS

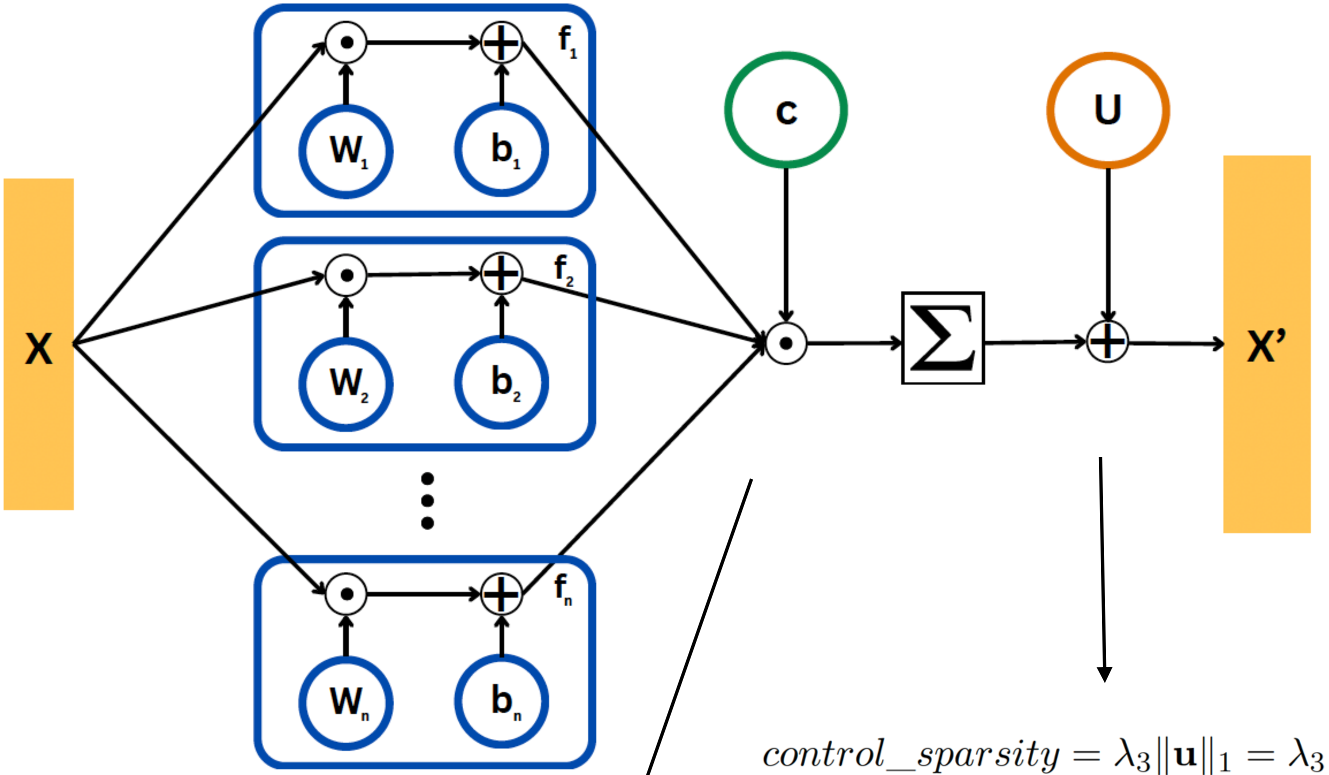
**Advantage**: interpretable model components (coefficients)

**Disadvantages**: autonomous system (does not capture actuation), iterative algorithm (slow)

**Solution:** Gradient Descent, Control Theory $x_{t+1} = Ax_t \longrightarrow x_{t+1} = Ax_t + Bu_t$



# cdLDS: Reformulating dLDS as a feed-forward Neural Network to incorporate control signal inference




$$control\_sparsity = \lambda_3 \|\mathbf{u}\|_1 = \lambda_3 \sum_{i=1}^{d} |u_i|$$

$$sparsity = \lambda_1 \|\mathbf{c}\|_1 = \lambda_1 \sum_{i=1}^{d} |c_i|$$

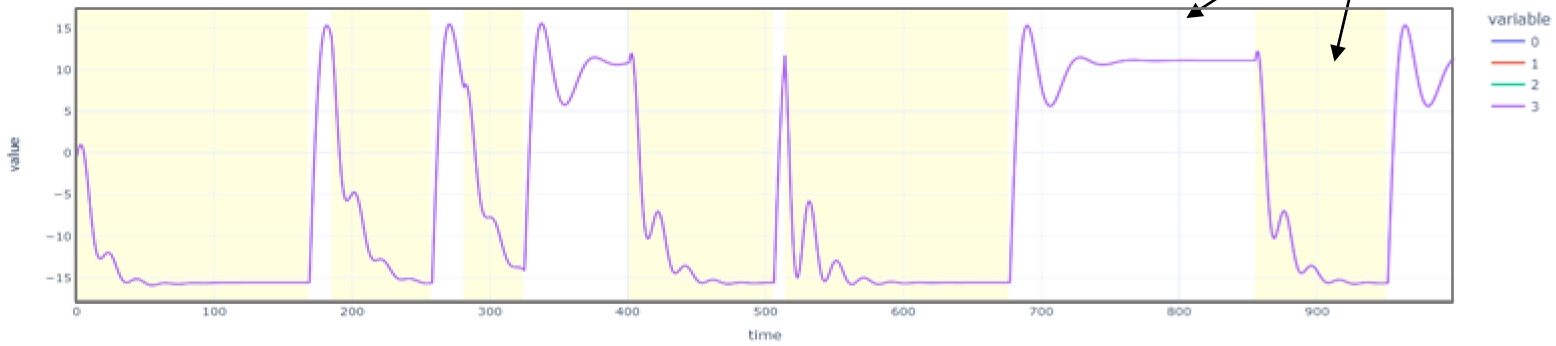
$$smoothness = \lambda_2 \|(\mathbf{c}_{t+1} - \mathbf{c}_t)\|_2 = \lambda_2 \sqrt{\sum_{i=1}^{d} (c_{i+1} - c_i)^2}$$

$$MSE = \frac{1}{d} \sum_{i=1}^{d} (x_{i+1} - \hat{x}_{i+1})^2$$

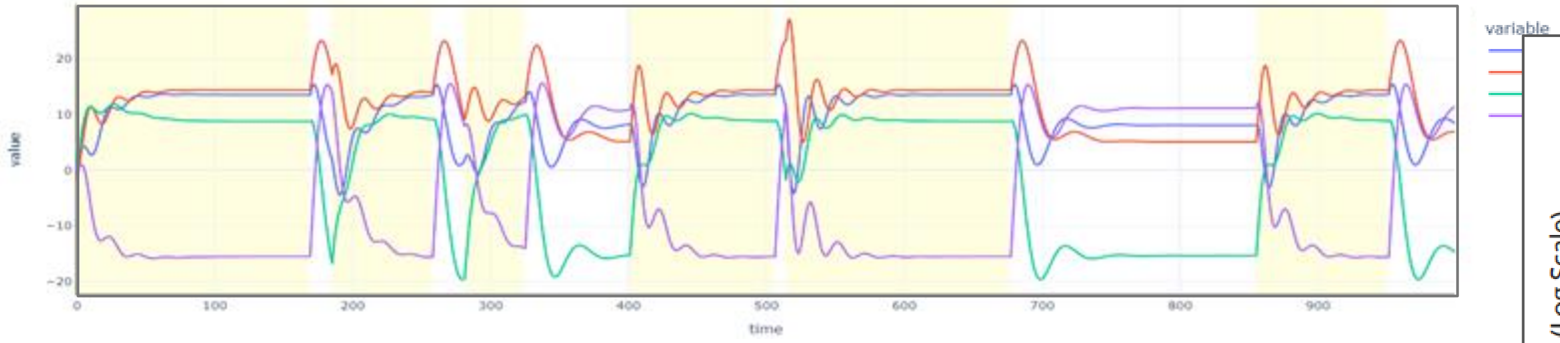
# cdLDS – Model Validation/Testing

different dynamics from 2 systems

1. Generate synthetic data



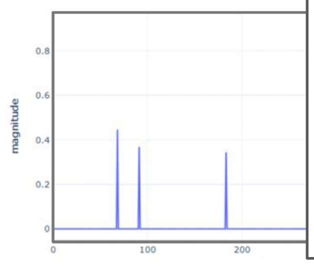

Synthetic Data from a two state system

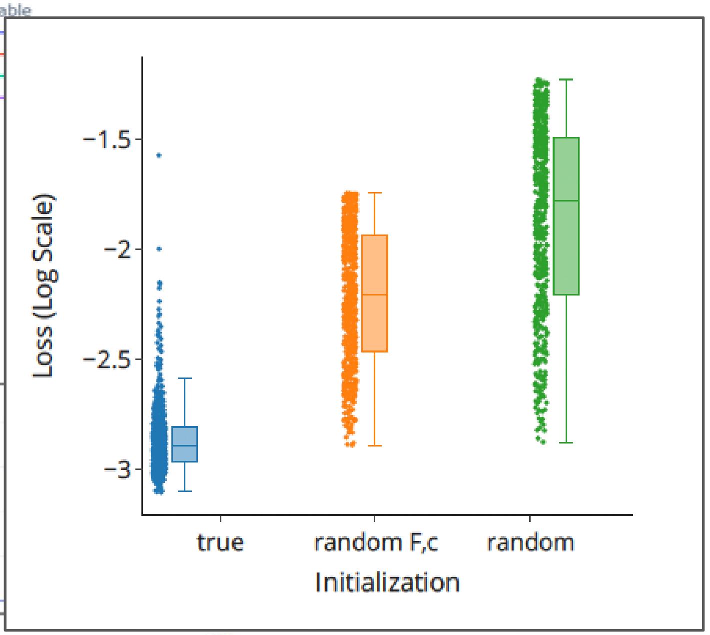



2. Initialize Model components with Ground Truth

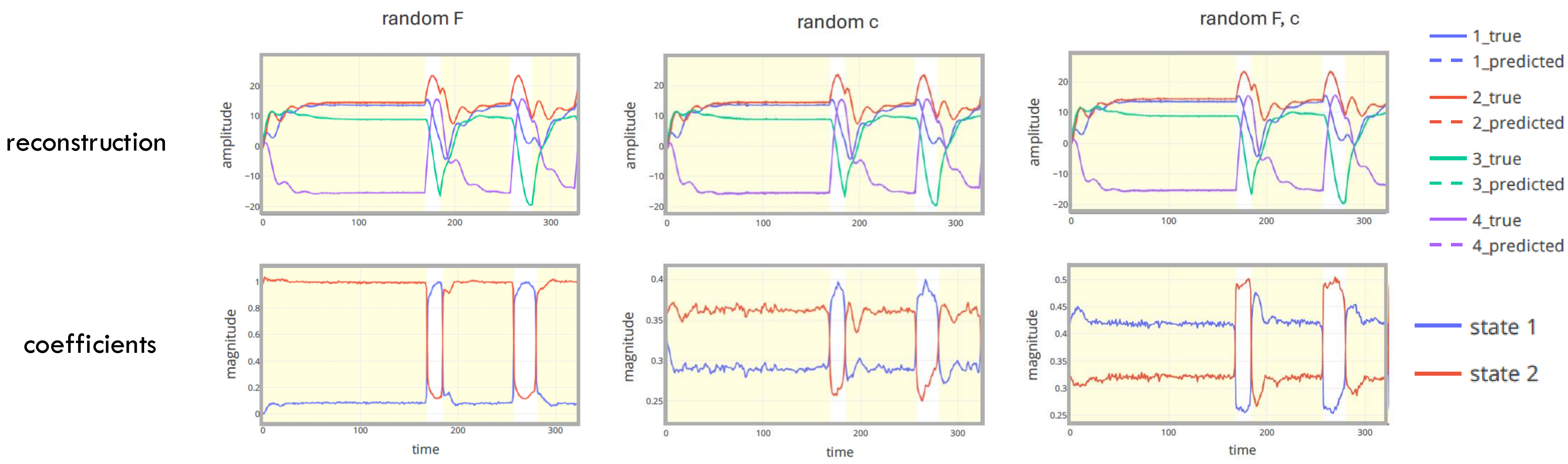
3. Sweep over hyperparameters

Control Signals



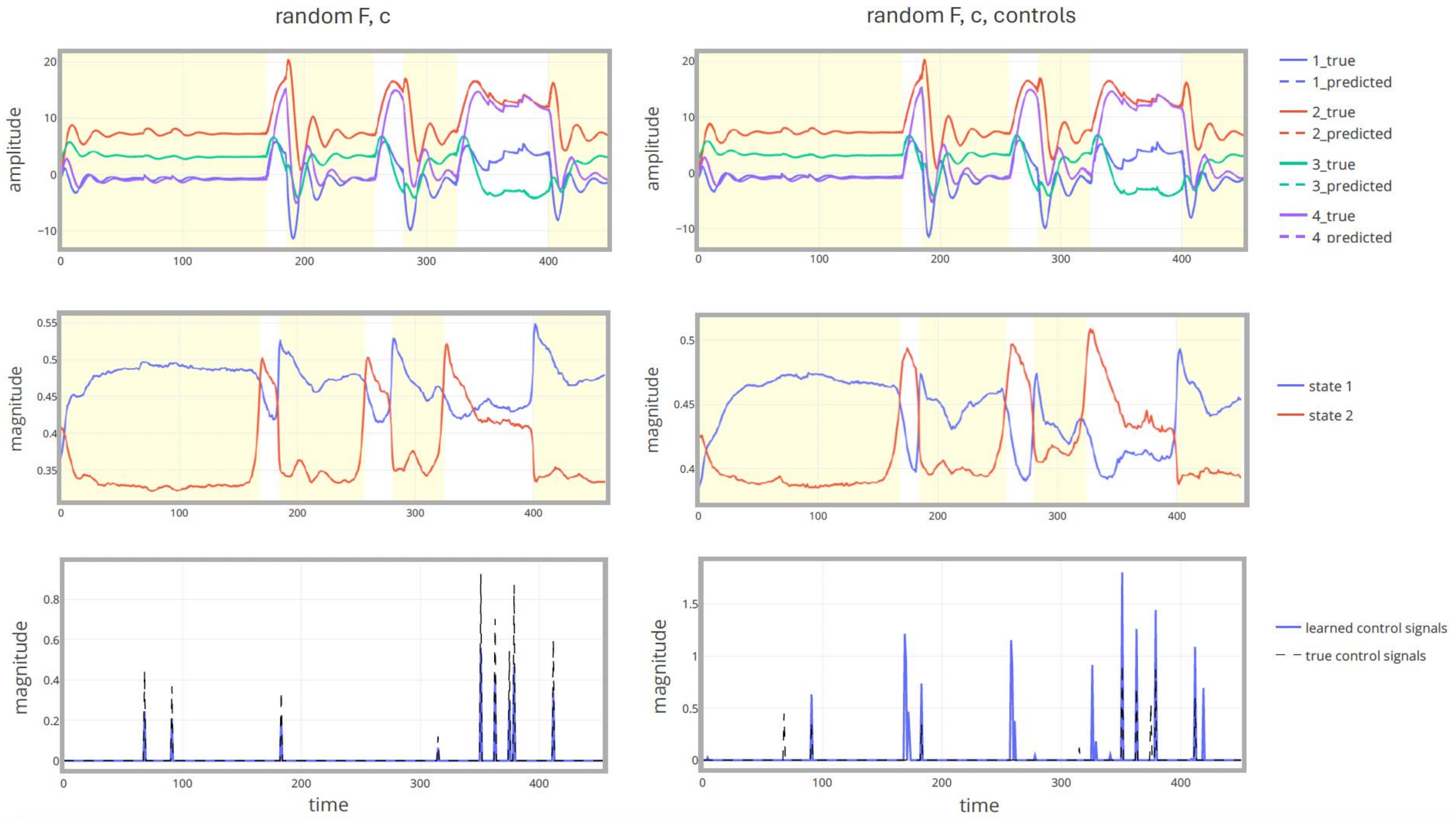



# dLDS accurately infers model coefficients in synthetic data



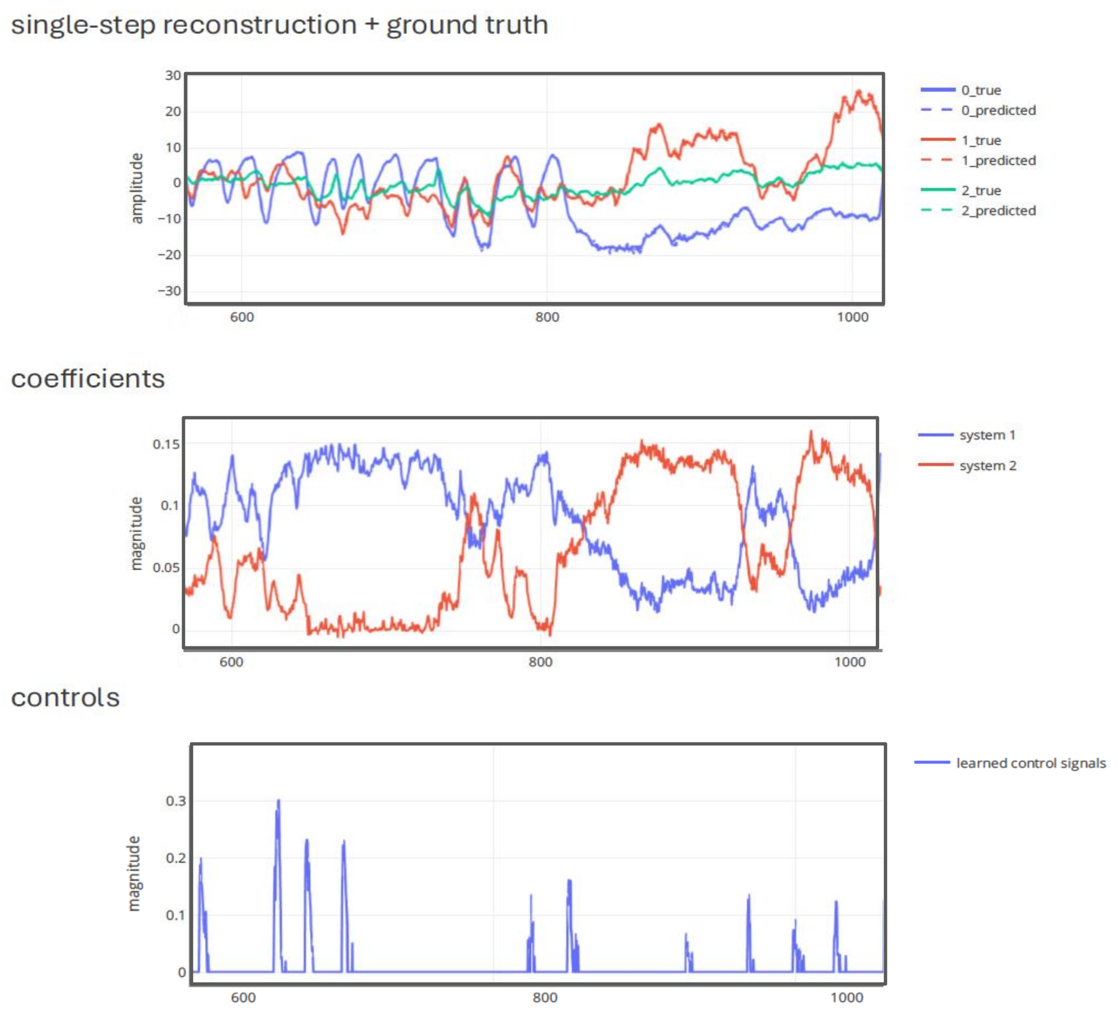

dLDS with true components applied to toy data retains the true coefficients.

# cdLDS disambiguates control signals from the coefficients in synthetic data




cdLDS applied to toy data yields coefficients and control signals close to their true counterparts.

# cdLDS applied to the canonical neural manifold consistently separates coefficients and control signals




**cdLDS with a control initialization strategy applied to a single recording projected to the canonical neural manifold.**

# Conclusion and Outlook

**canonical neural manifold**
→ a low-dimensional manifold structure from 23 *C. elegans* individuals
→ a *C. elegans* latent state space (that is already used for modeling)
→ a pipeline of preprocessing steps to align neural activity data

**controlled decomposed linear dynamical systems (cdLDS)**
→ PoC, disambiguation intrinsic and actuated dynamics on the neural manifold
→ a method to learn control signals
→ a dLDS model that is much faster than the original algorithm

Future Work?

# Acknowledgements

universität wien

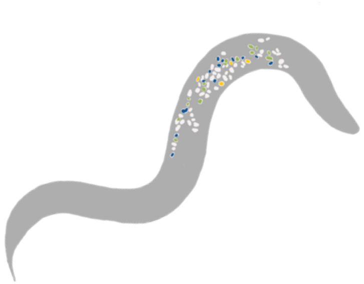
TU WIEN

LiSC
Life Science
Computer Cluster

**TU intern evaluators:**
Peter Filzmoser
Christian Huemer
Thomas Gärtner

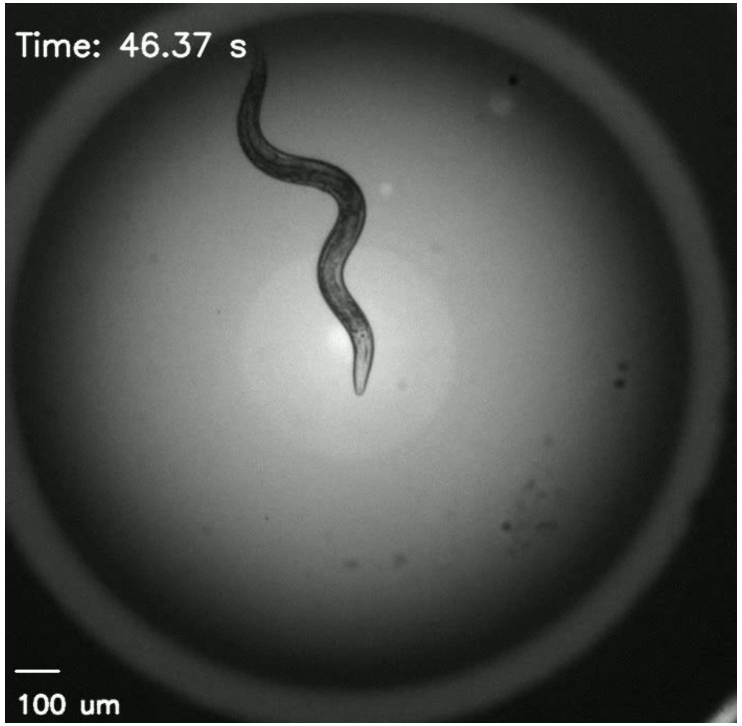
**Univie Supervisors:**
Manuel Zimmer
Charles Fieseler

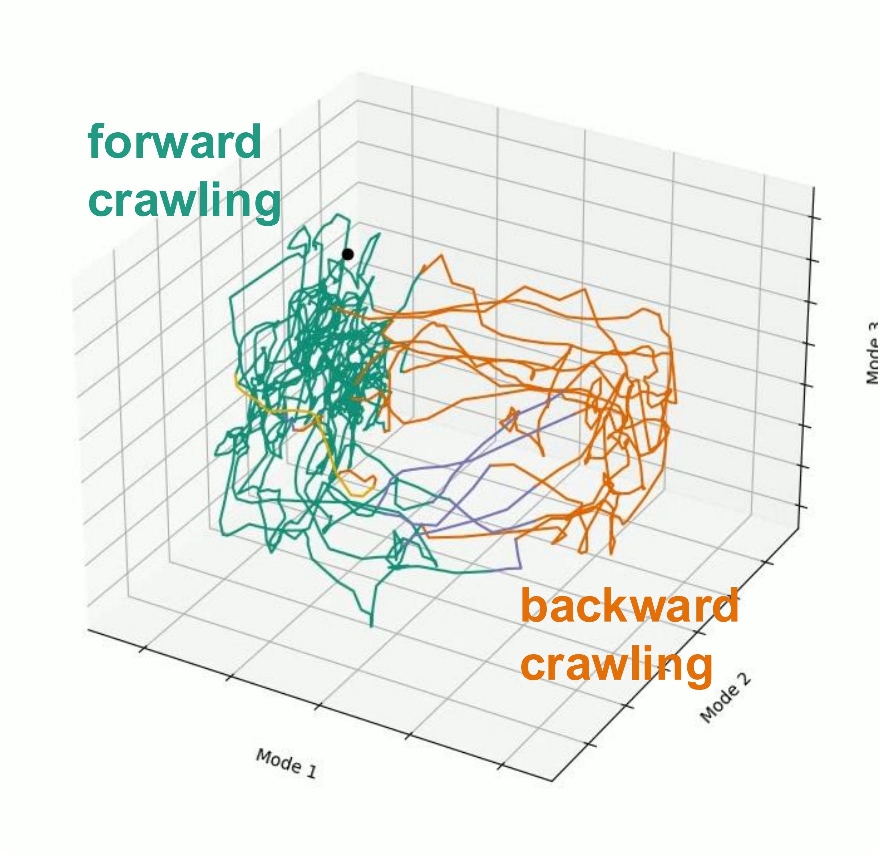
**Data acquired by:**
Rebecca Kresnik
Kerem Uzel



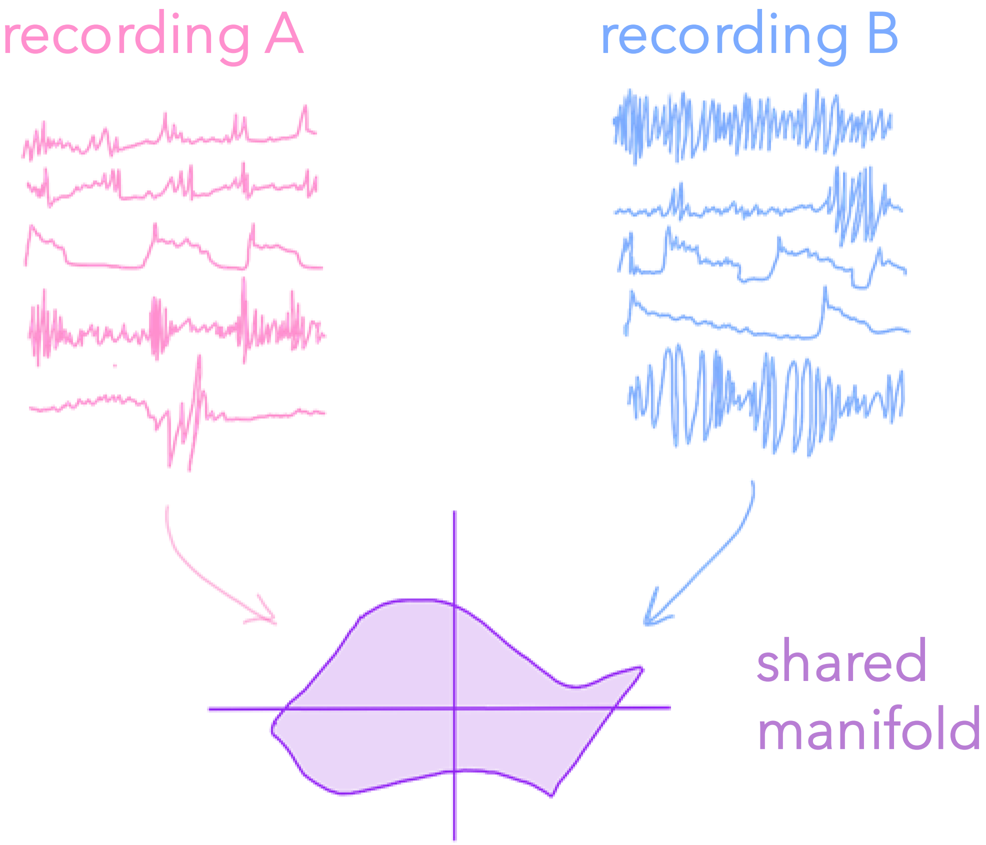
Thanks to current and past members of the Zimmer Lab for the support & a wonderful year:
Itamar Lev, Jalaja Madhusudhanan, Lukas Hille, Richard Latham, Fabio Näecht, Stephanie Eder, Maria Böhm, Matthias Kaltenböck, Eva Gratzl, Chanuka Fernando, Benjamin Schaar, Hannah Brenner, Laxmi Yadav, Mrinnanda Bhattacharya, Wenxin Li (Charlotte), Paul Pichler, Pedro Augusto, Charles C.H. Cohen, Samuel Zidaric, Eleftheria Giagnisi, Katharina Sweis, Verena Steinkasser, Belen Martinez-Asenjo, Norbert John Msaki

# An *elegant* organism to study whole brain dynamics



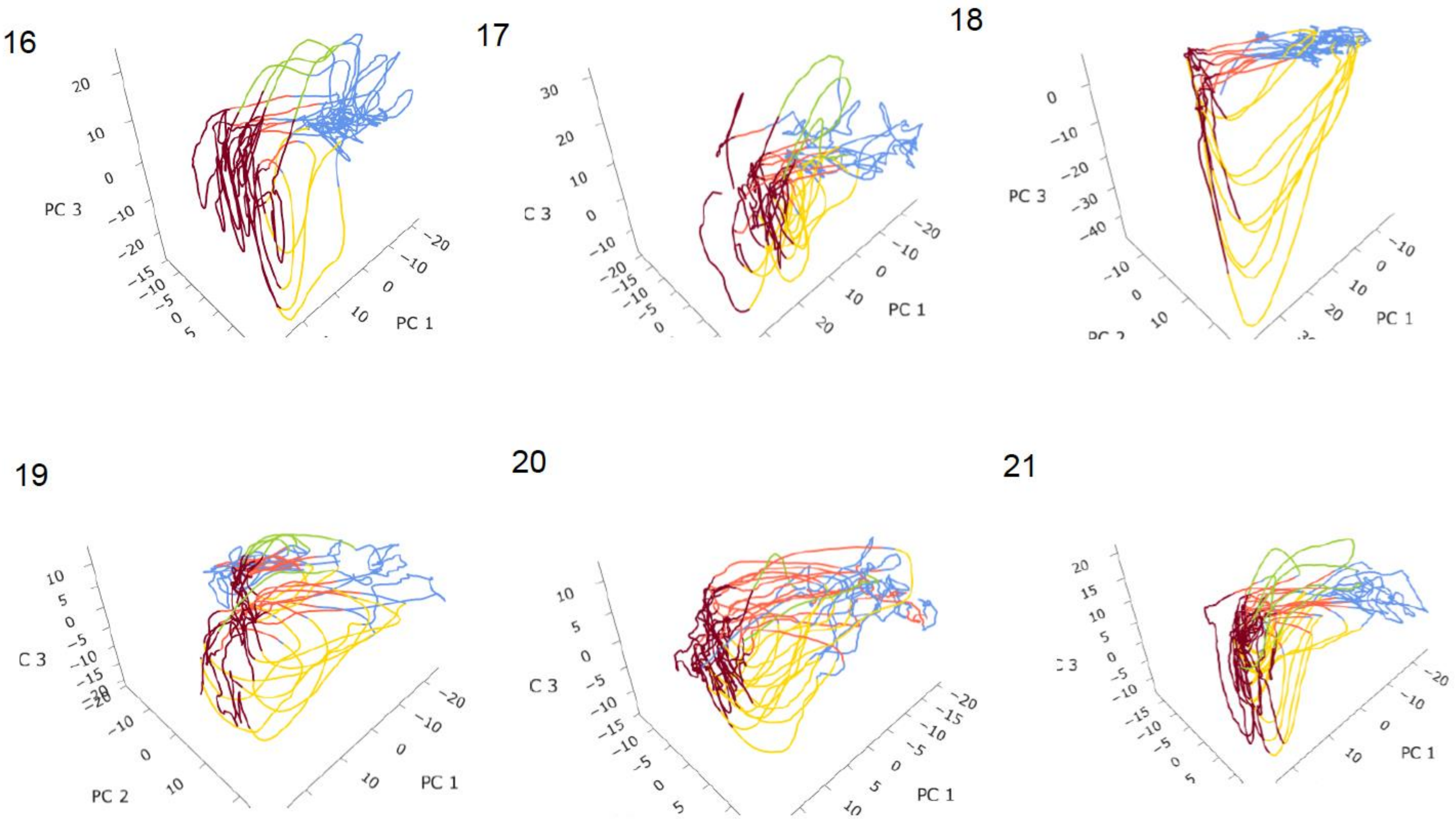



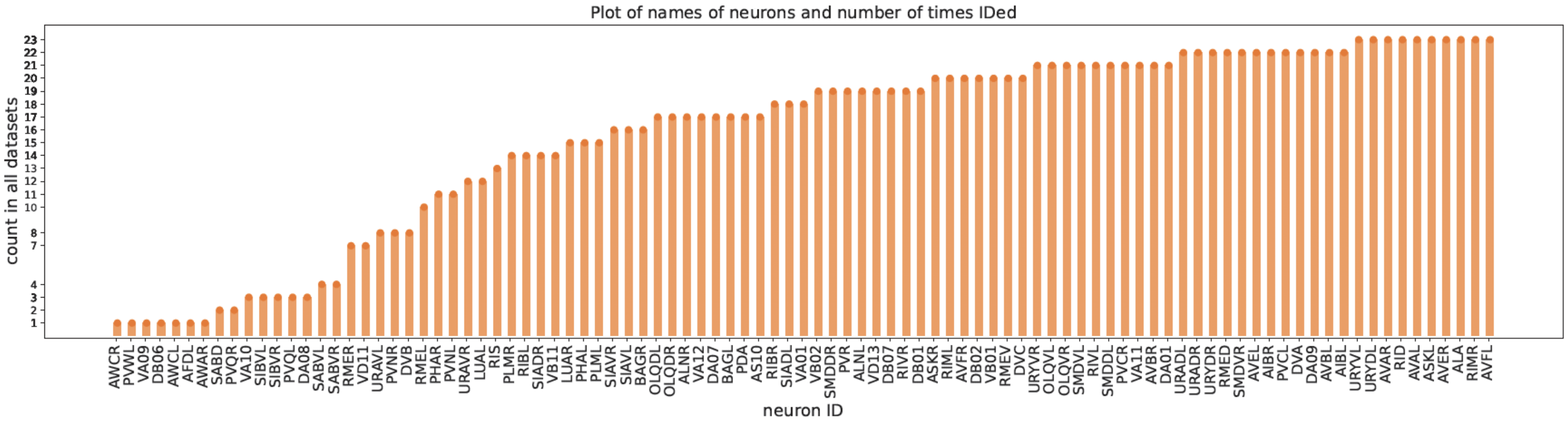

*Video and Visualization Source: Charles Fieseler (UniVie)*



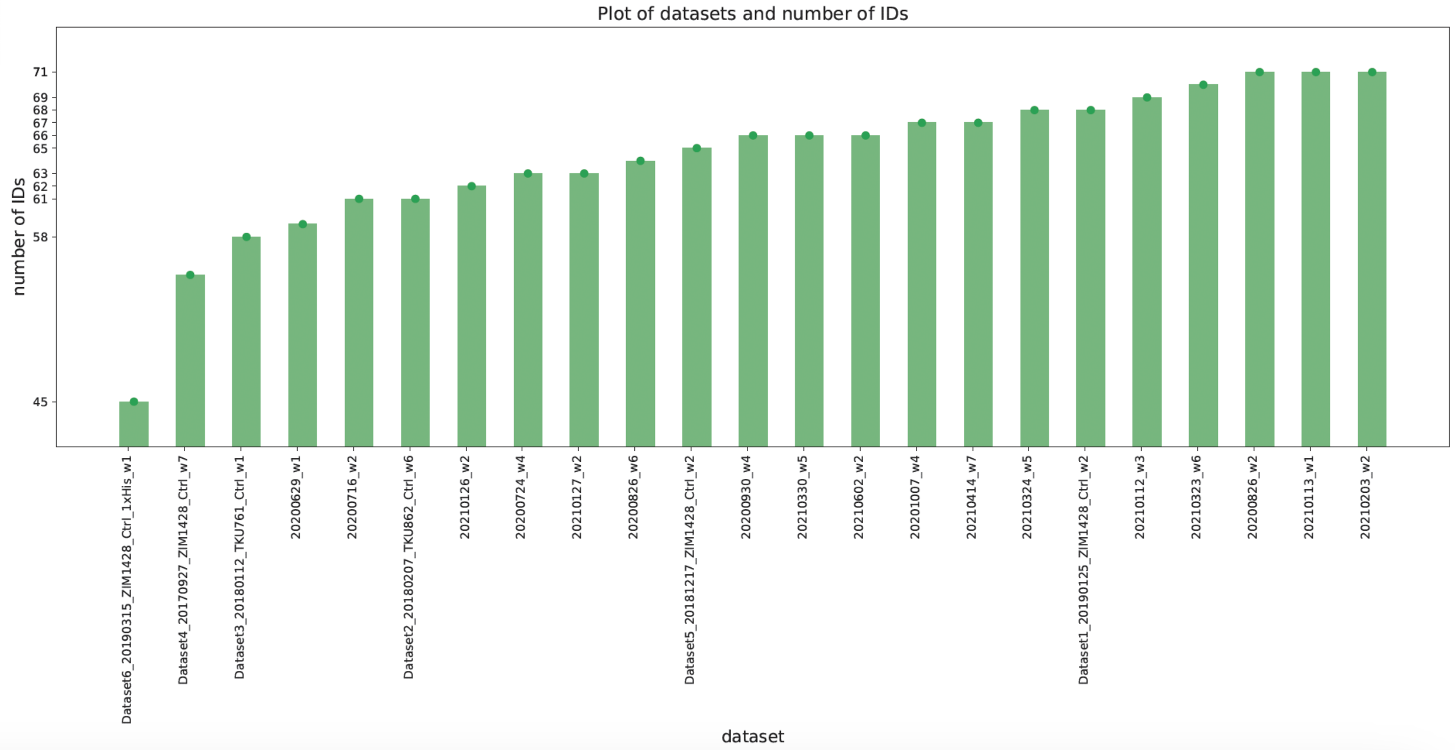

Given the inconsistencies across recordings, what preprocessing steps are required to align neural activity data of individual C. elegans in order to obtain a canonical manifold?

# Individual data sets projected to shared PC space



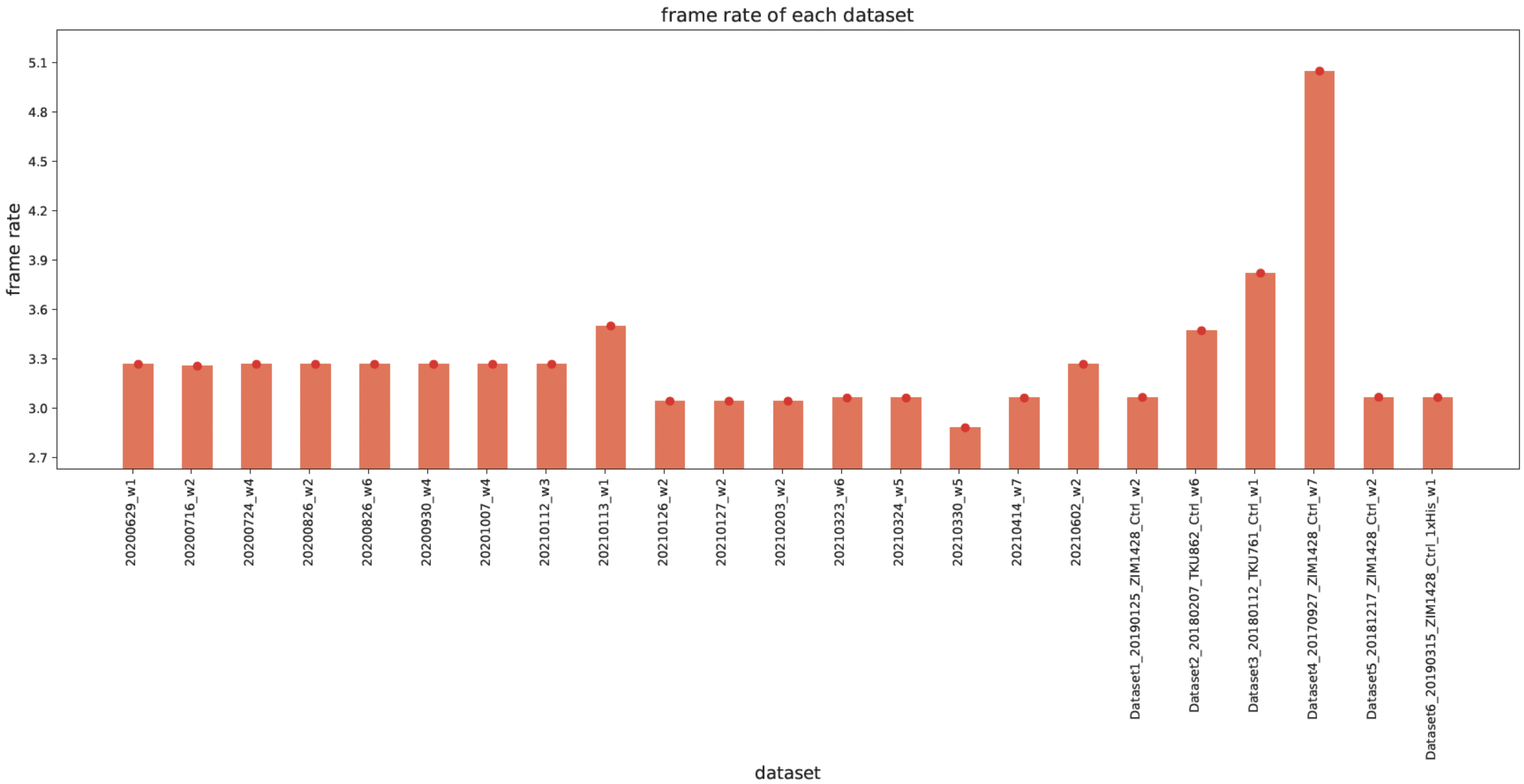



**Number of datasets in which each neuron is identified**

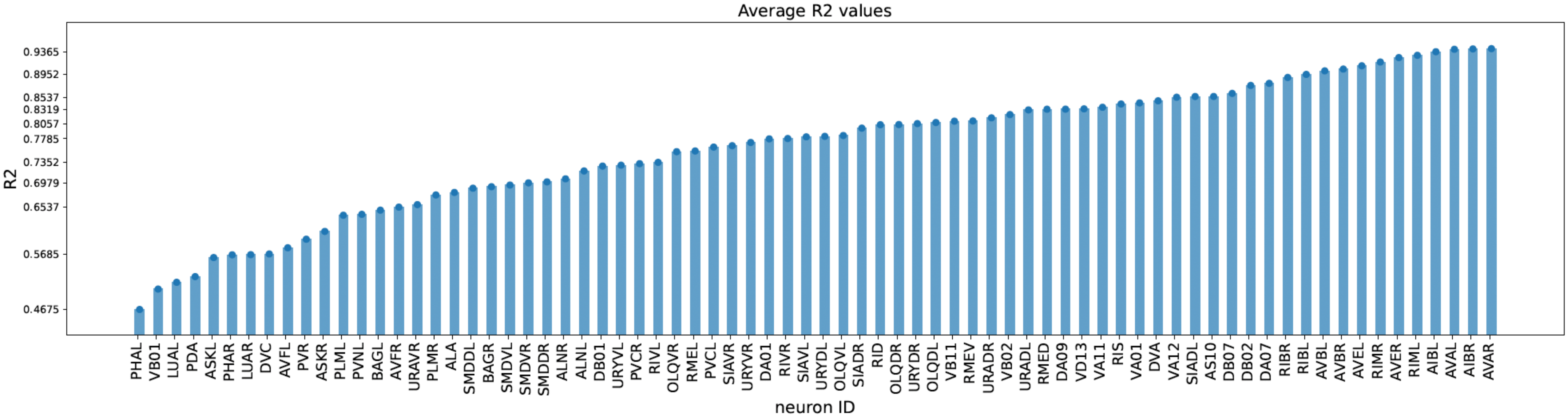



**Number of neurons that have been identified in each dataset**

# The frame rate varies across recordings

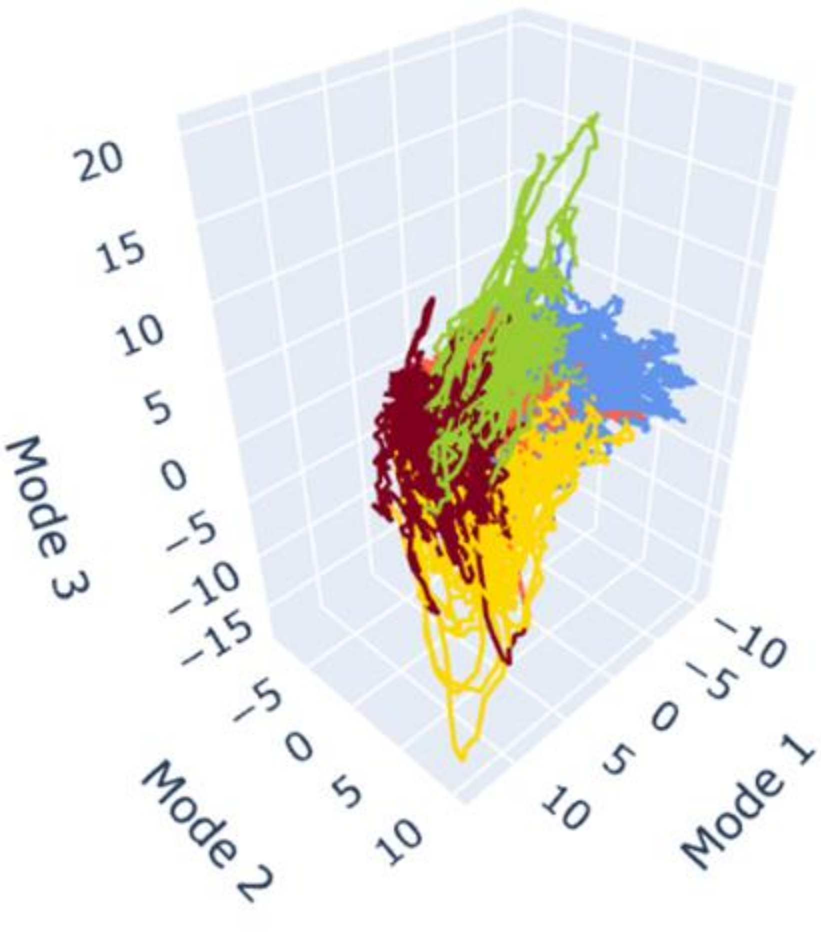


# PLSR determines a lack of uniqueness in most neurons (PPCA check)

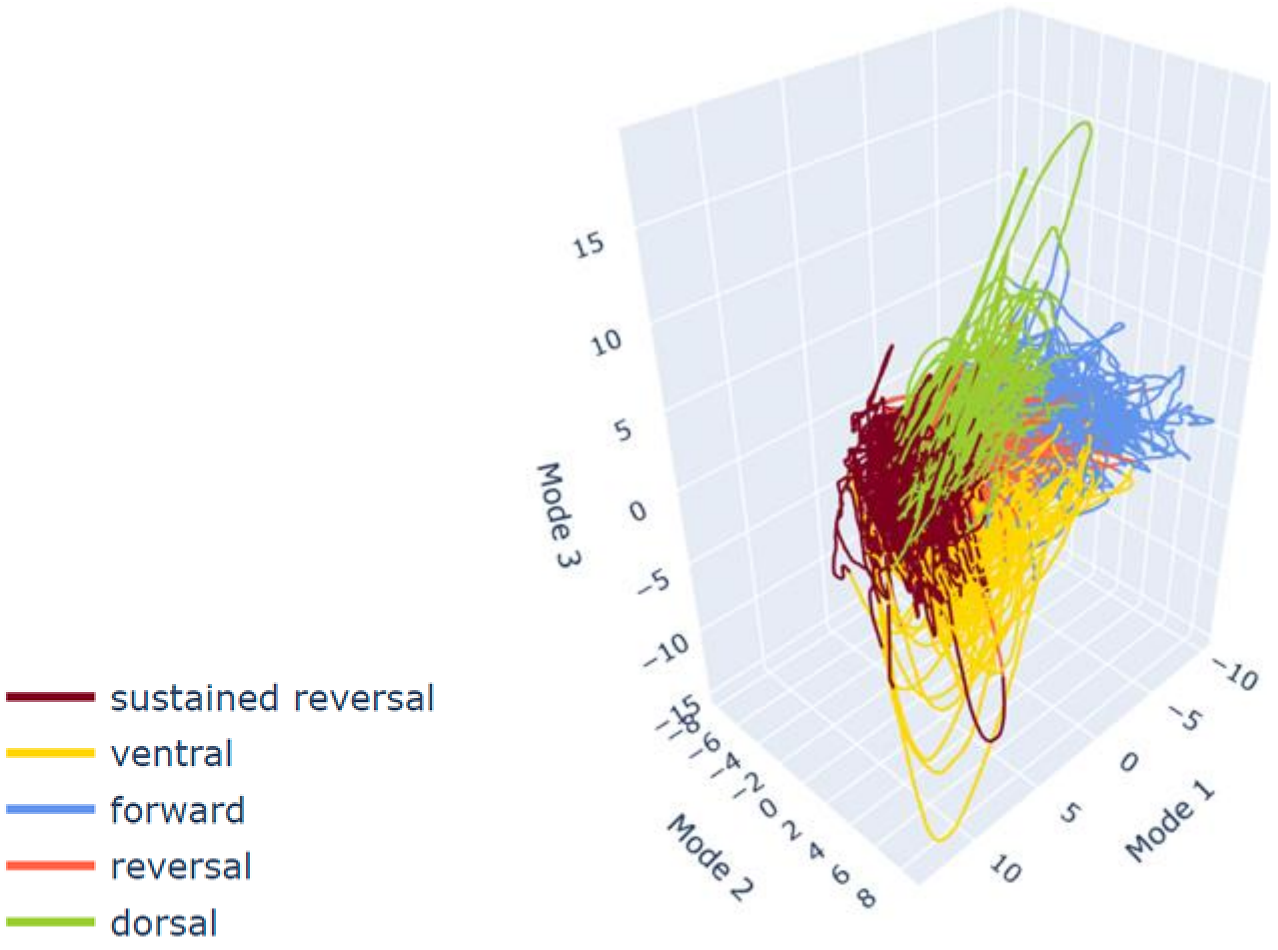


# Preprocessing steps that worked

WITHOUT TIME DELAY EMBEDDING

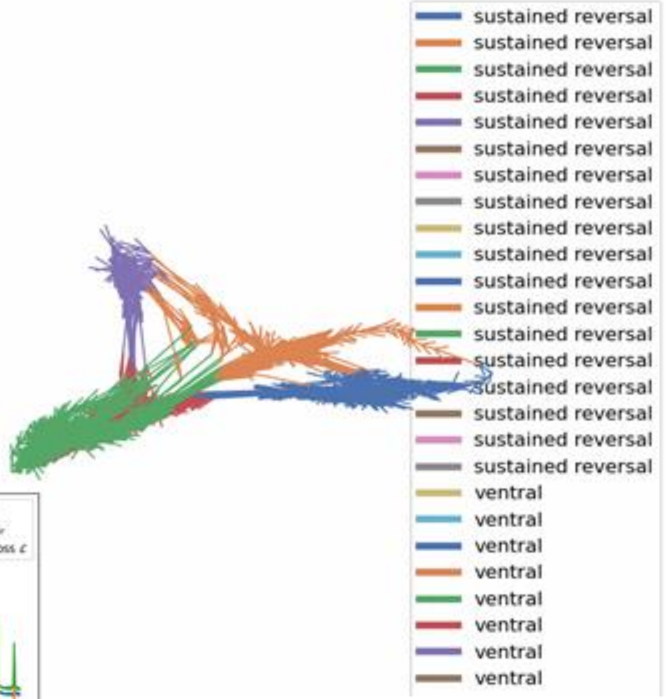


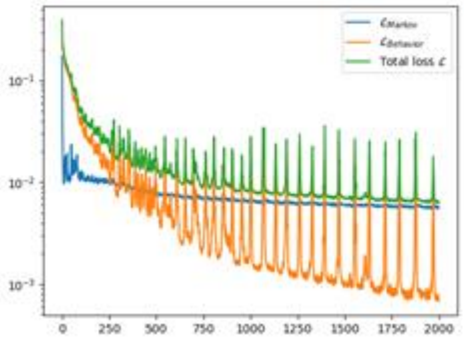

WITH TIME DELAY EMBEDDING



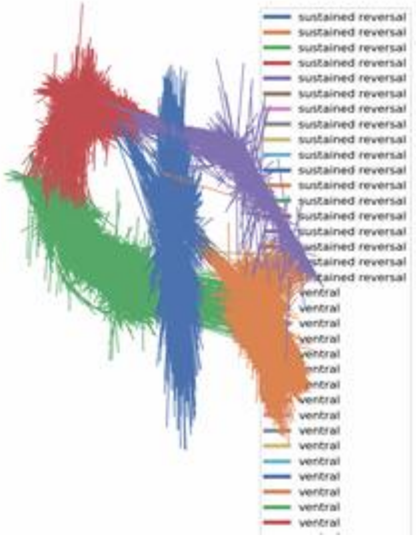
# BunDLe-Net

dataset 1

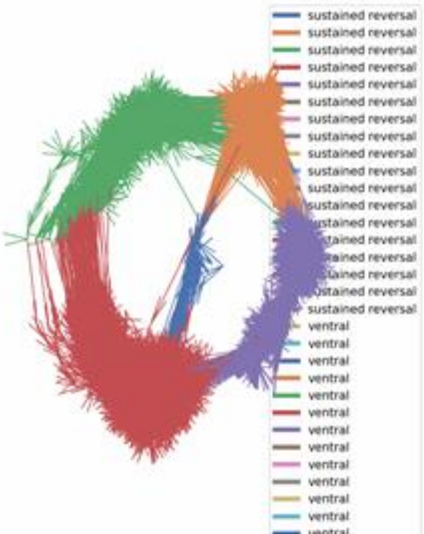



datasets up to 7



datasets up to 8



# CLUSTERING I

→ use k-means to cluster data points into 23 clusters and check the similarity between the predicted clusters and the true dataset clusters
→ evaluate with adjusted mutual information score*

$$AMI(U, V) = [MI(U, V) - E(MI(U, V))] / [avg(H(U), H(V)) - E(MI(U, V))]$$

| AMI | High-Dim (Raw) | Low-Dim (Raw) | Neural Manifold |
|---|---|---|---|
| All states | 0.332 | 0.277 | 0.219 |

after applying the preprocessing steps there is a huge decrease in similarity between the predicted clustering and the dataset membership

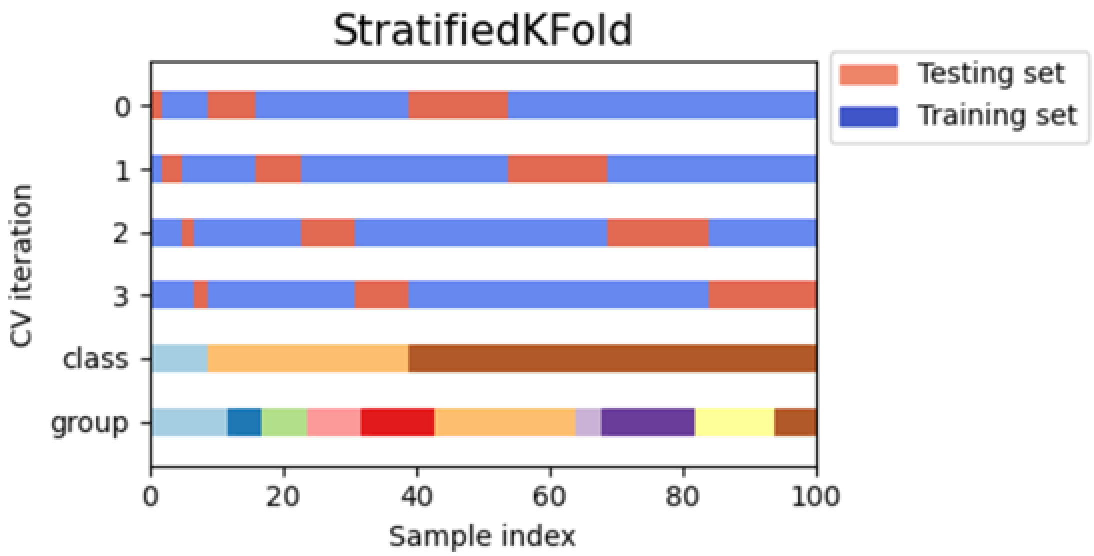
*where 1.0 == identical

# CLASSIFICATION I

→ use **simple linear model** to classify time points

→ dataset membership. If model performs poorly then this might indicate that there is not much difference between the datasets

→ **train-test split** based on **StratifiedKFold**



# CLASSIFICATION I

→ use SVC = Support Vector Machine (Classifier)
→ dataset membership

| Metric | High-Dim (Raw) | High-Dim | Low-Dim (Raw) | Neural Manifold |
|---|---|---|---|---|
| accuracy | 0.872 | 0.481 | 0.192 | 0.058 |
| precision | 0.905 | 0.522 | 0.181 | 0.061 |
| recall | 0.872 | 0.480 | 0.192 | 0.058 |
| f1 | 0.869 | 0.475 | 0.166 | 0.058 |

after preprocessing, all metric scores decrease dramatically, meaning that observations from different recordings are not distinguishable = variability might not be explained by dataset

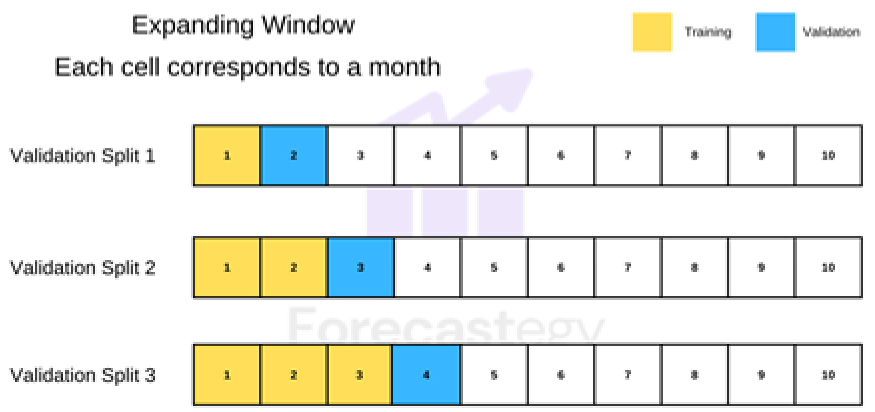
# CLASSIFICATION II

→ use **simple linear model** to classify time points

→ dataset membership. If model performs poorly then this might indicate that there is not much difference between the datasets

→ behaviour state. If model performs poorly then this might indicate that there is no clear separation between the states

→ **train-test split** based on '**expanding window cross-validation**'




`sklearn.model_selection.`TimeSeriesSplit

**image from forecastegy*

# CLASSIFICATION II

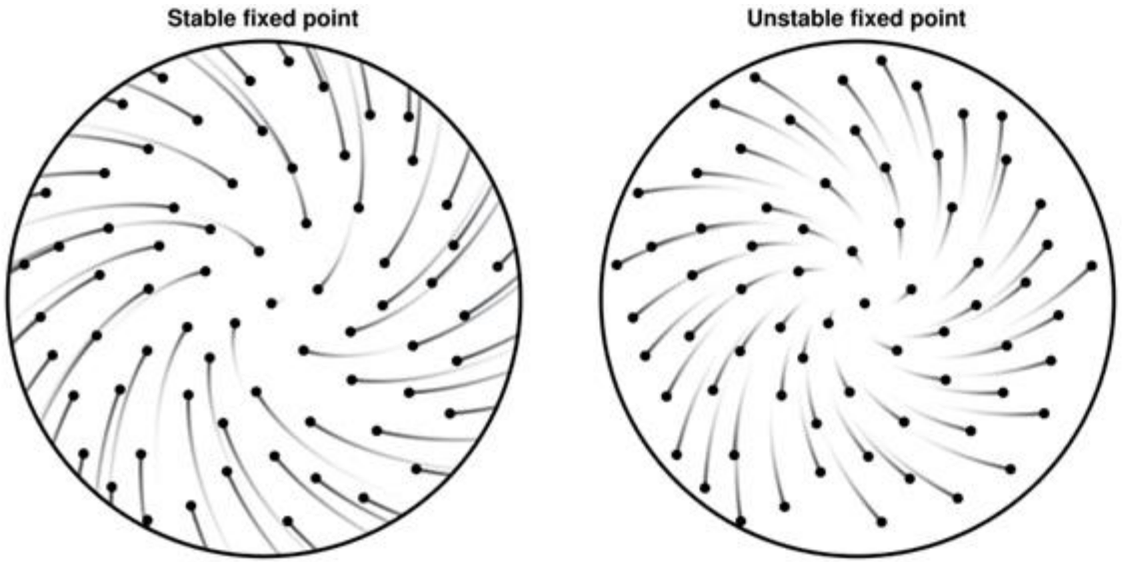
→ use SVC = Support Vector Machine (Classifier)
→ behaviour states

| Metric | High-Dim (Raw) | High-Dim | Low-Dim (Raw) | Neural Manifold |
|---|---|---|---|---|
| Accuracy | 0.732 | 0.901 | 0.809 | 0.760 |
| Precision | 0.777 | 0.901 | 0.78 | 0.799 |
| Recall | 0.732 | 0.901 | 0.809 | 0.760 |
| F1 | 0.723 | 0.899 | 0.780 | 0.759 |

# Synthetic Data to reflect switches between linear dynamical systems

$$x_{t+1} = \sum \begin{matrix} 0 \times A_1 x_t \\ 1 \times A_2 x_t \\ \vdots \\ 0 \times A_n x_t \end{matrix}$$

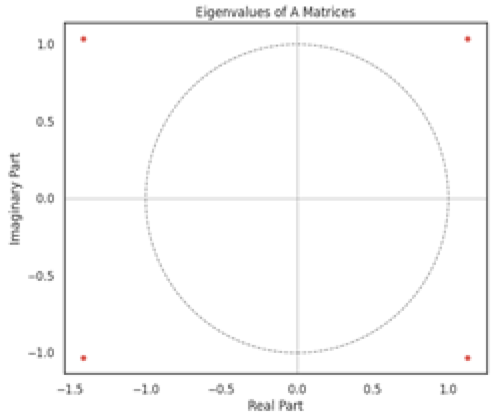
choice of eigenvalues determines stability of the system

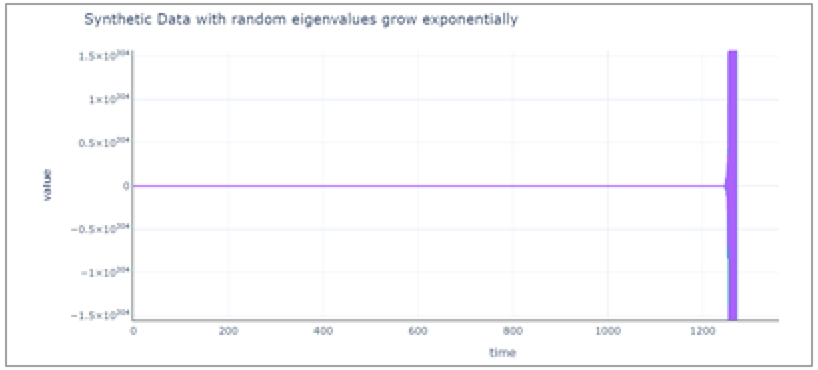



animation by Jacopo Bertolotti

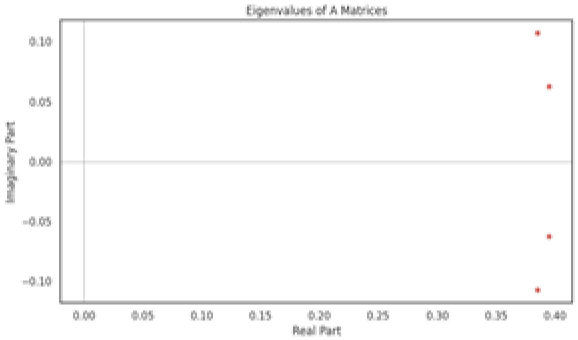
# Synthetic Data to reflect switches between linear dynamical systems

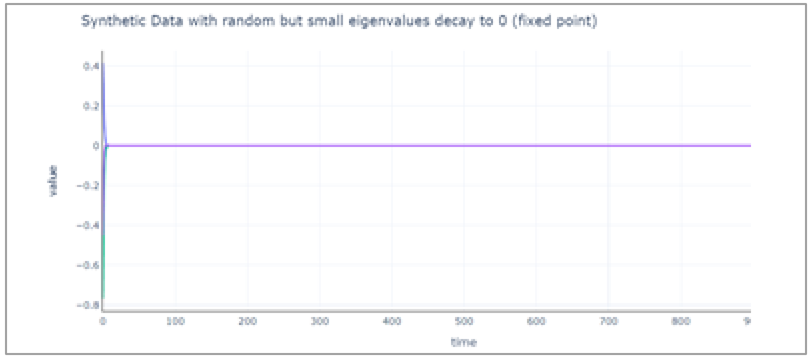
**λ > 1**



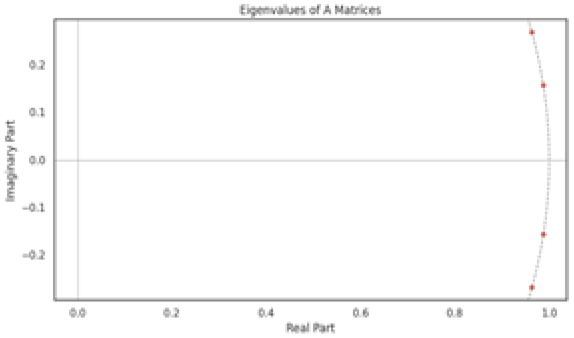



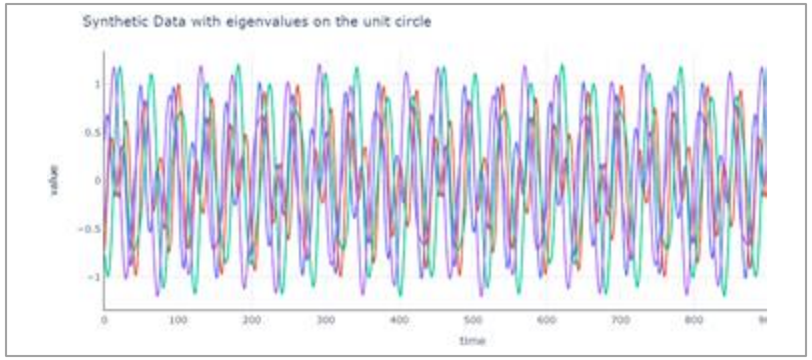

**λ < 1**



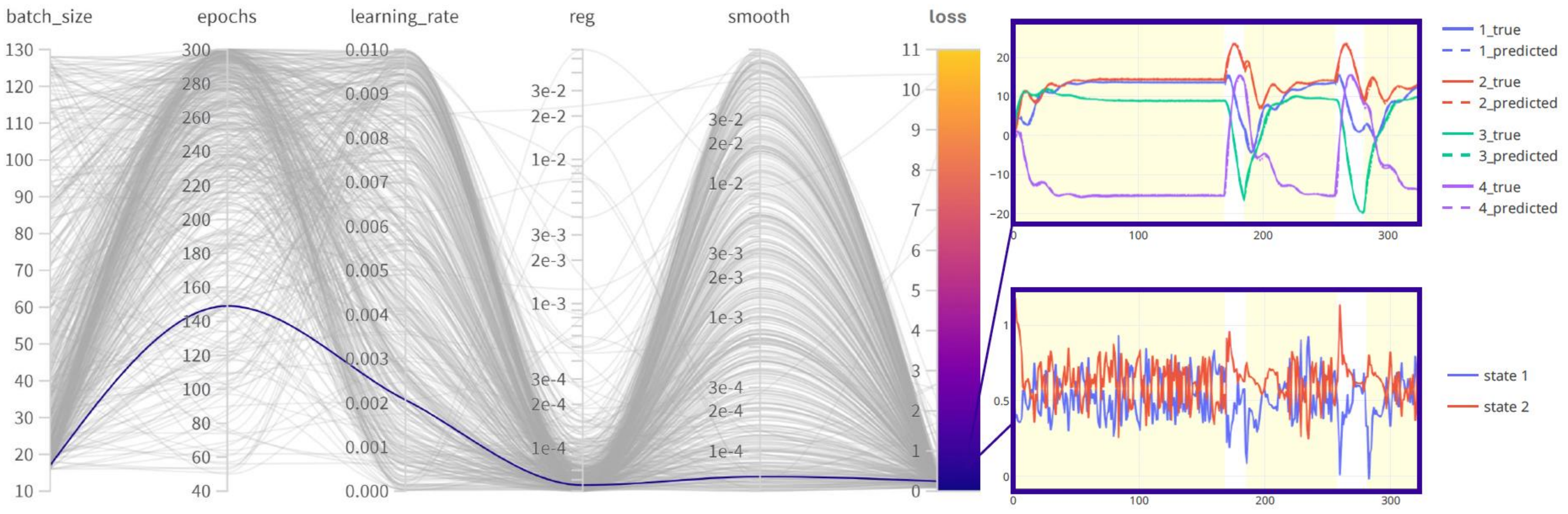




**λ = 1**

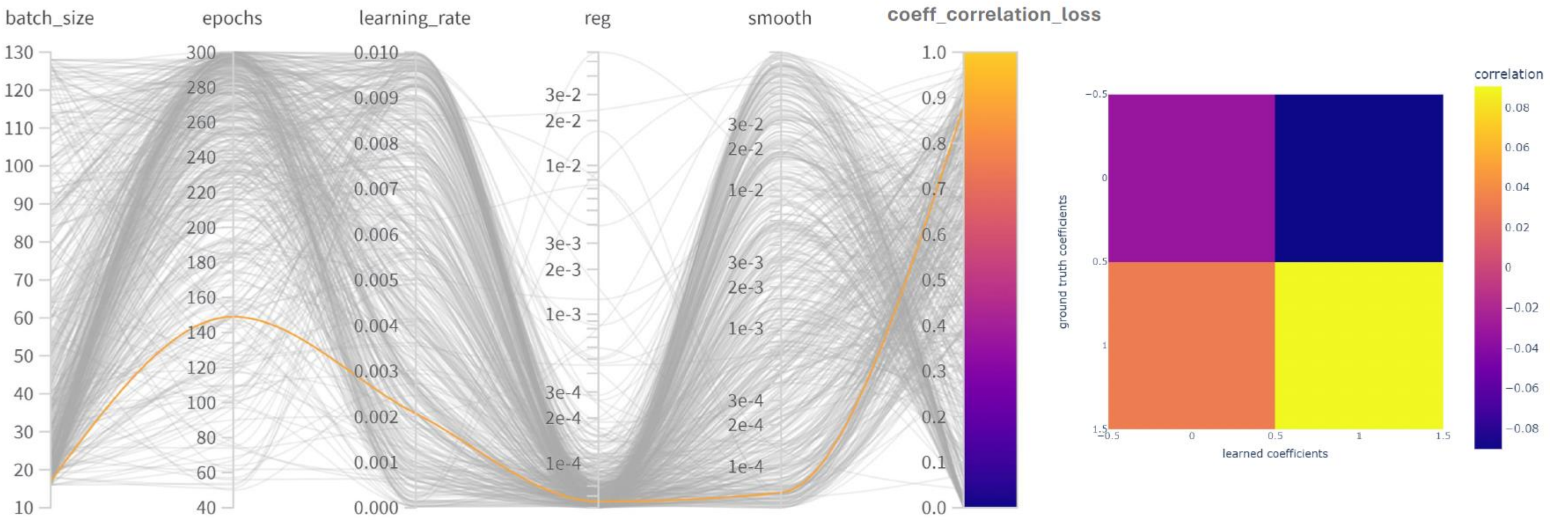


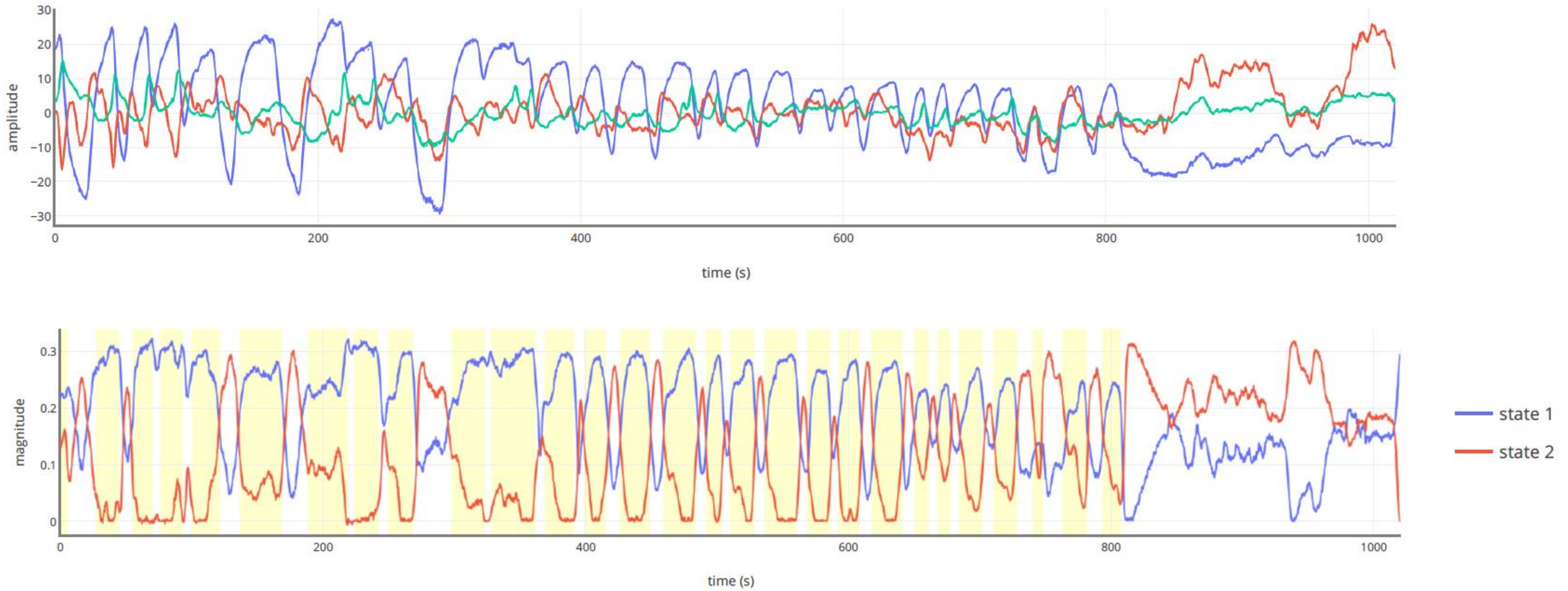
# A low dLDS loss does not imply interpretable coefficients..




## and indicates the need for an additional evaluation metric

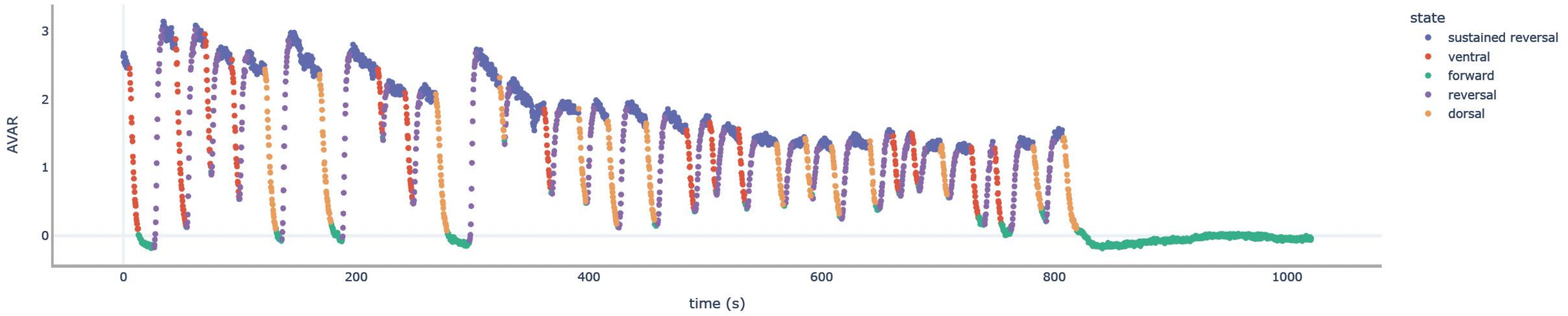


# Neural Manifold latent states correspond to behavior states




**dLDS applied to a single recording projected to the Canonical Neural Manifold.** The yellow highlights indicate where the worm is in a reversal state, as opposed to the forward state, which remains unhighlighted.

# Intrinsic vs. actuated dynamics in C. elegans




Activity of reversal driver AVA and correlated neurons show ‘pirouettes’. What is from a behaviour perspective a period of frequent turns might be the system failing to stabilize.

This could be explained by a computational model that can infer actuated dynamics.

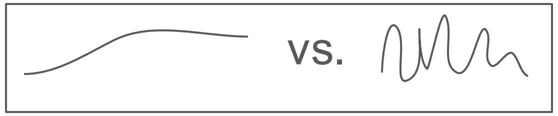
# Reformulating dLDS as a feed-forward Neural Network

## Why?

- Additional components like the controls can be integrated more easily
- Lower runtime for equal or better performance
- State of the Art
- MLOps can be leveraged for monitoring computational experiments

# What can go wrong?

- **Bad Hyperparameters**
  - # of epochs
  - batch size
  - learning rate
- **Bad Regularization/Loss Functions**
  - smoothness over time-varying coefficients
  - mean squared error




$$\mathrm{MSE} = \overset{\text{Mean}}{\boxed{\frac{1}{n}\sum_{i=1}^{n}}}\,(\overset{\text{Error}}{\boxed{Y_i - \hat{Y}_i}})^{\overset{\text{Squared}}{\boxed{2}}}$$

$$\widehat{\boldsymbol{x}}_t, \widehat{\boldsymbol{c}}_t = \arg\min_{\boldsymbol{x}_t, \boldsymbol{c}_t} \left[ \|\boldsymbol{y}_t - \boldsymbol{D}\boldsymbol{x}_t\|_2^2 + \lambda_0 \left\|\boldsymbol{x}_t - \widetilde{\boldsymbol{F}}_t \boldsymbol{c}_t\right\|_2^2 + \lambda_1 \|\boldsymbol{x}_t\|_1 + \lambda_2 \|\boldsymbol{c}_t\|_1 + \lambda_3 \|\boldsymbol{c}_t - \widehat{\boldsymbol{c}}_{t-1}\|_2^2 \right]$$

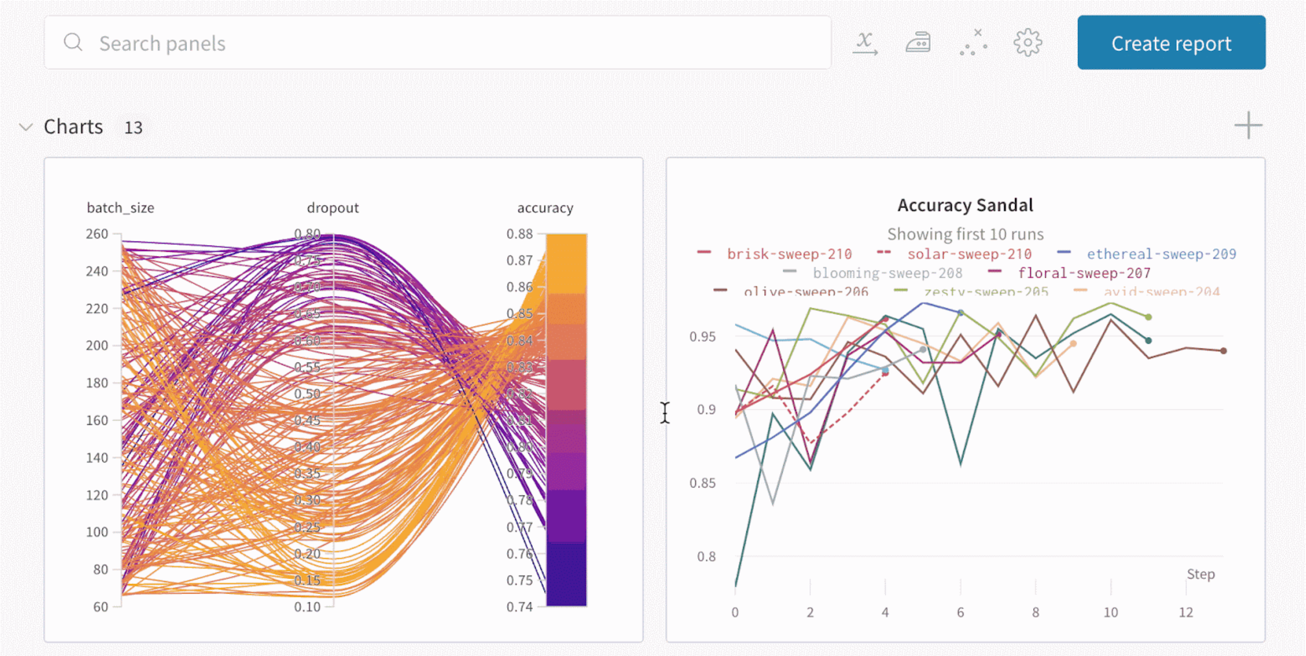
*formula from https://suboptimal.wiki/explanation/mse/*

# dLDS Experimental Conditions

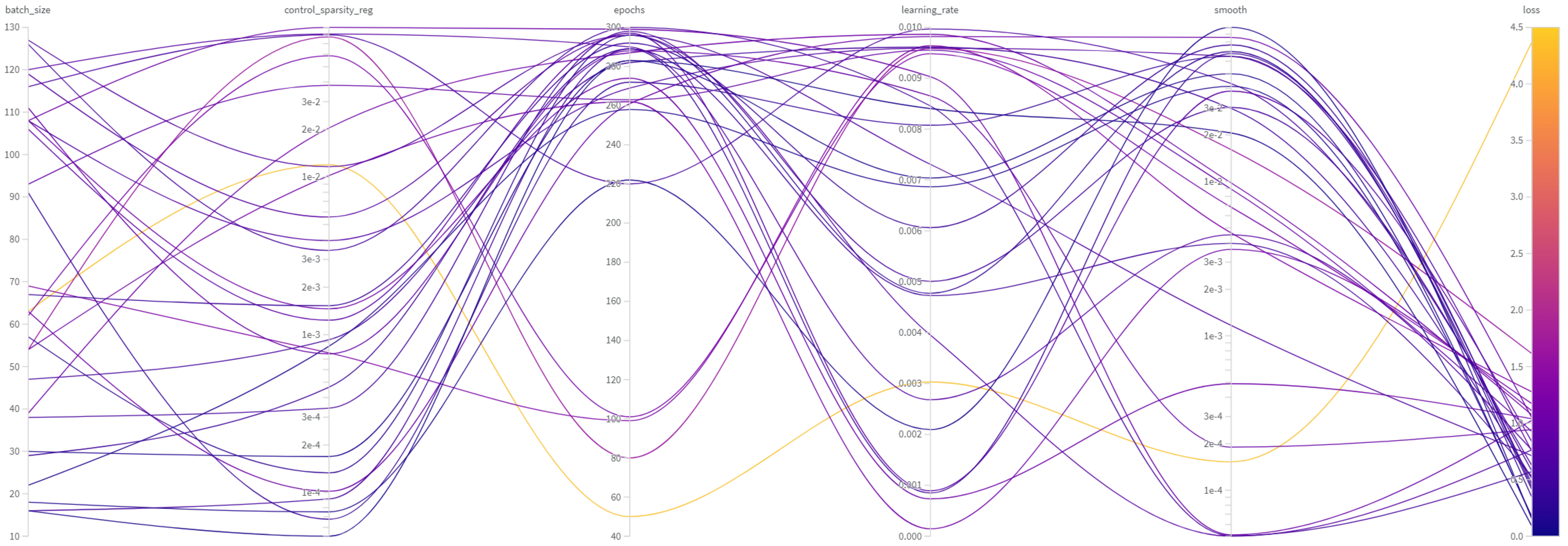
1. Ground Truth Initialization
2. True F Initialization
3. True c Initialization
4. Random Initialization

per condition we can run a sweep over hyperparameters with:

Weights & Biases

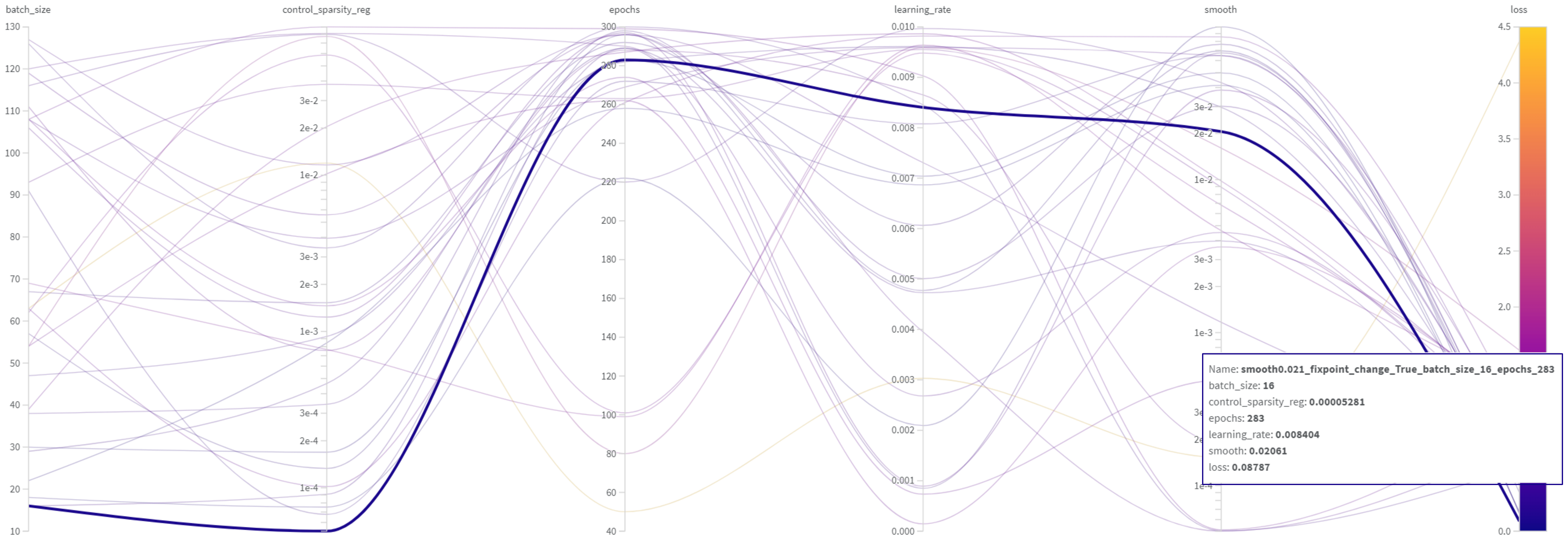


# dLDS Experimental Conditions - Example




Weights & Biases

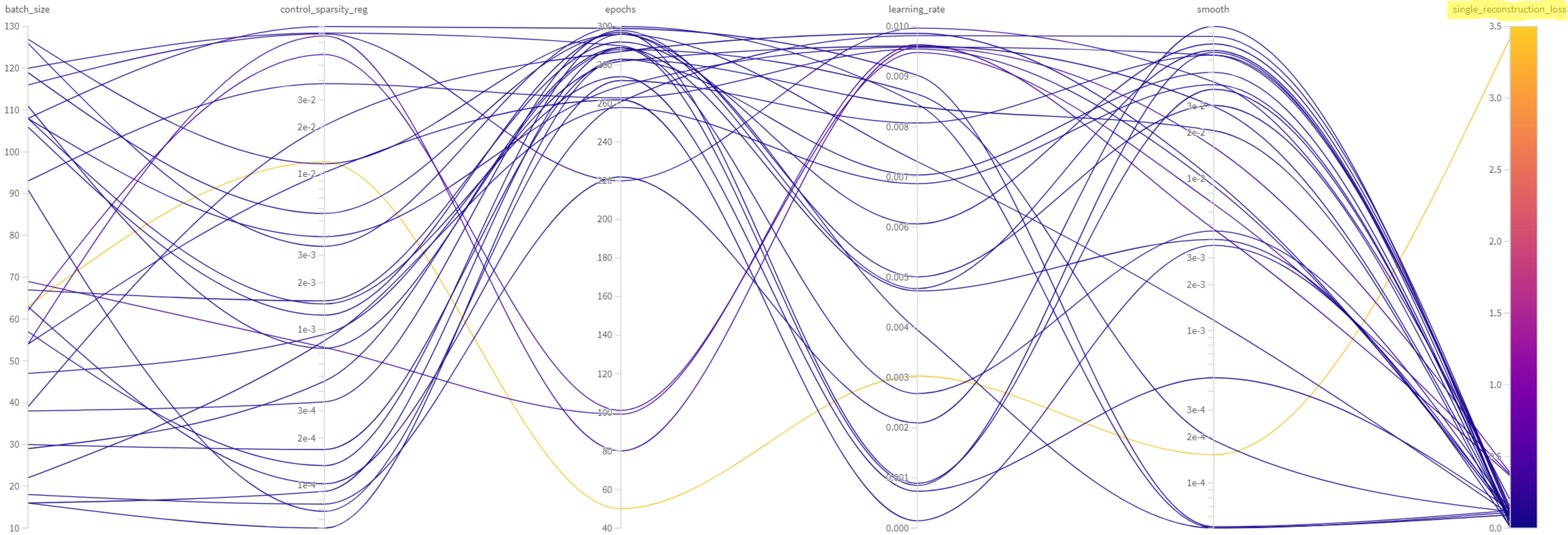
# dLDS Experimental Conditions - Example




Weights & Biases

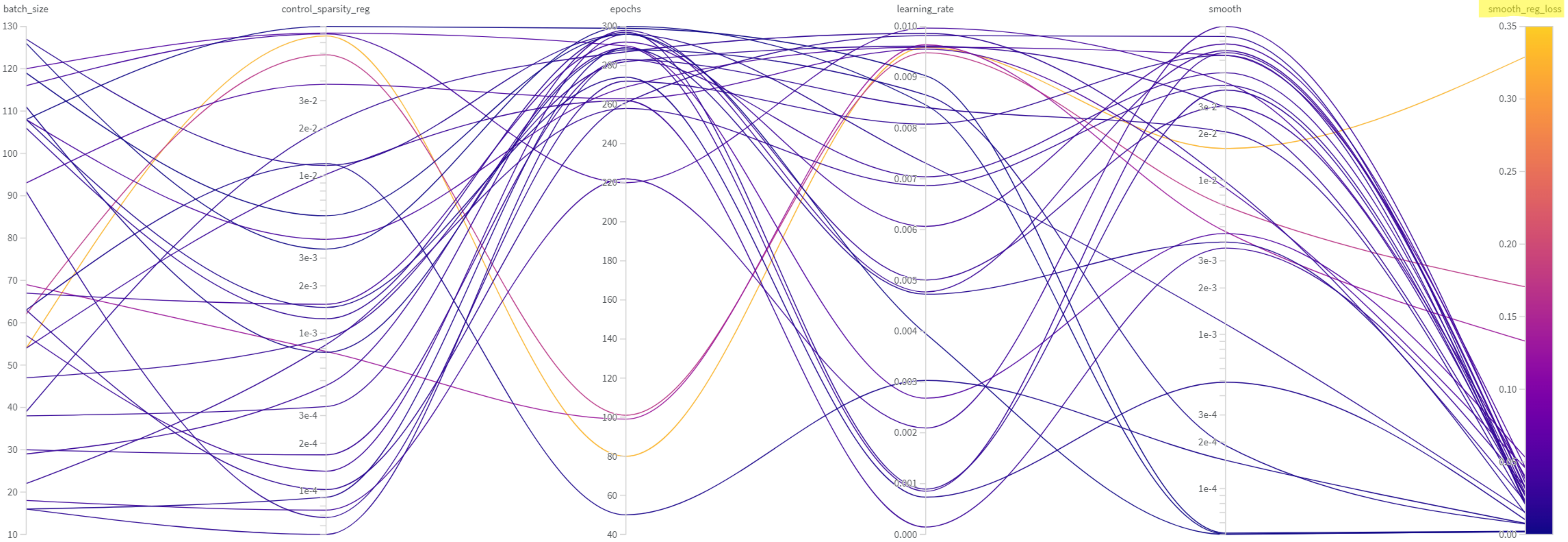
# dLDS Experimental Conditions - Example

$$\text{MSE} = \underbrace{\frac{1}{n}\sum_{i=1}^{n}}_{\text{Mean}}\underbrace{(Y_i - \hat{Y}_i)}_{\text{Error}}{}^{\overbrace{2}^{\text{Squared}}}$$



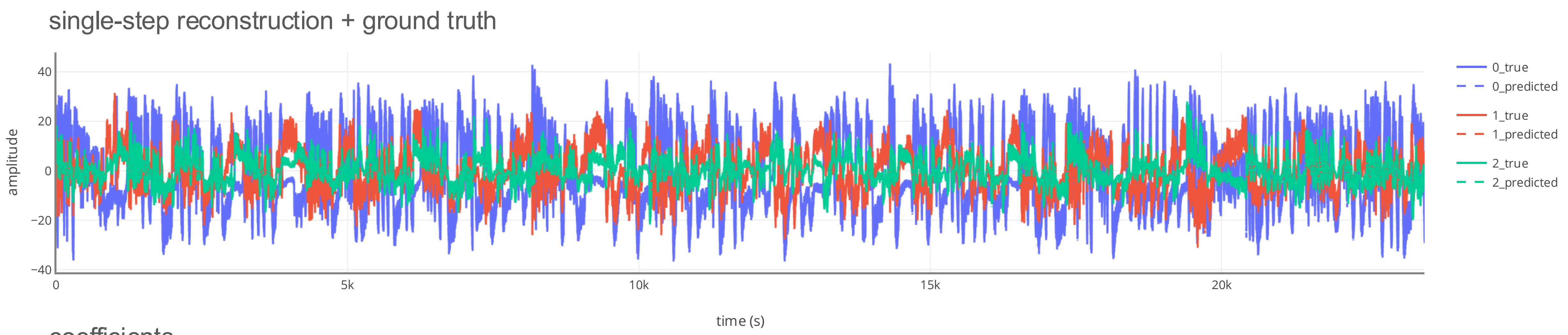

Weights & Biases

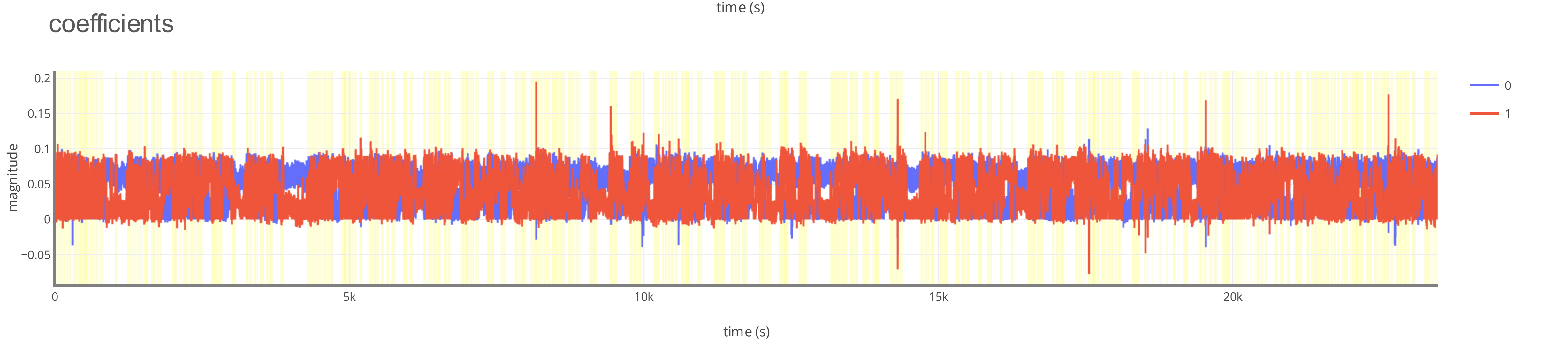
# dLDS Experimental Conditions - Example

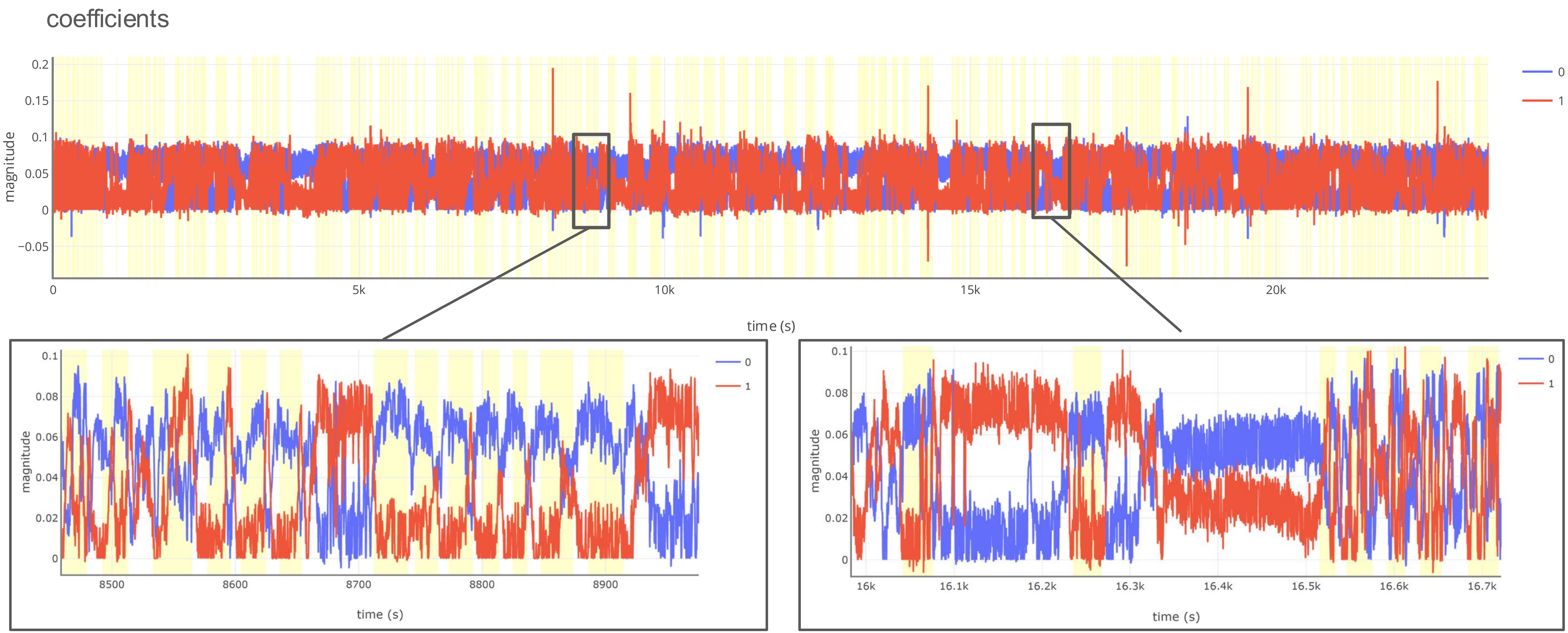

Weights & Biases

# dLDS applied to All Whole Brain Recordings projected to PC space



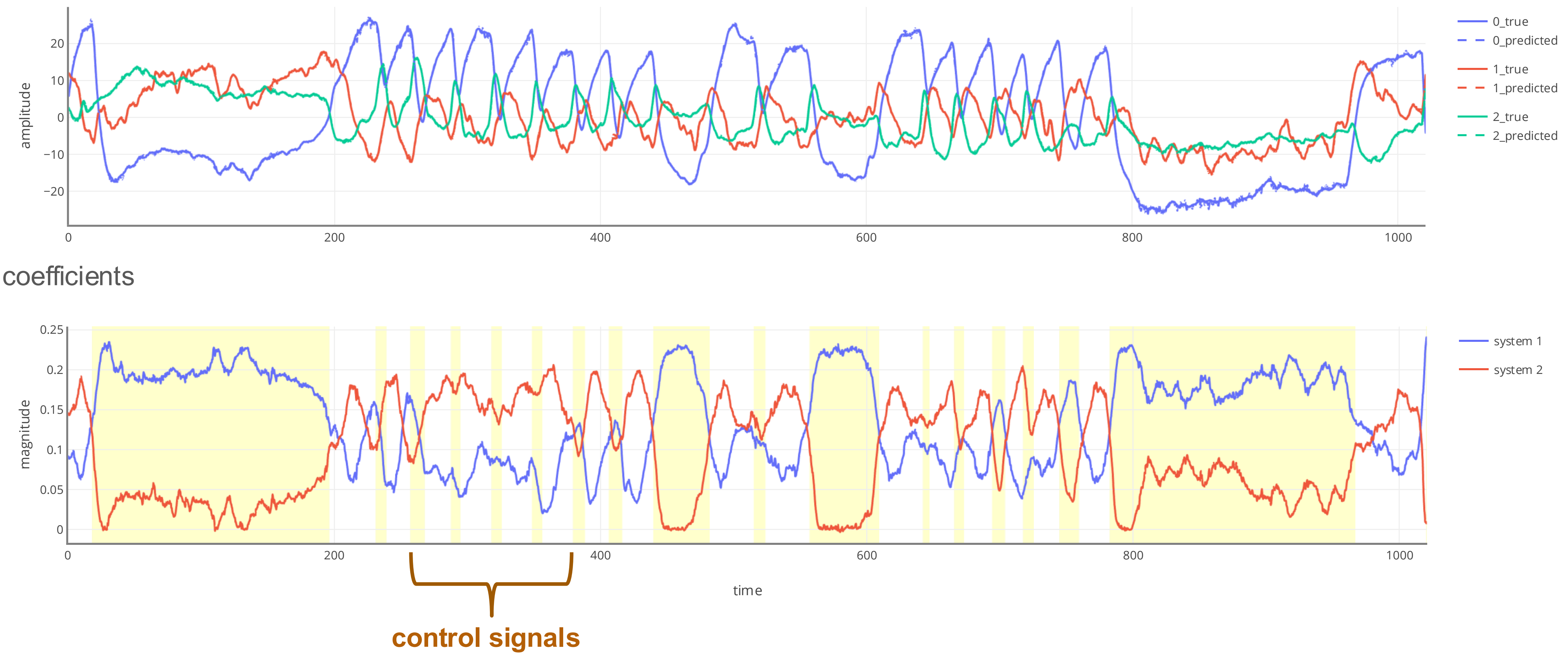



# dLDS applied to All Whole Brain Recordings projected to PC space



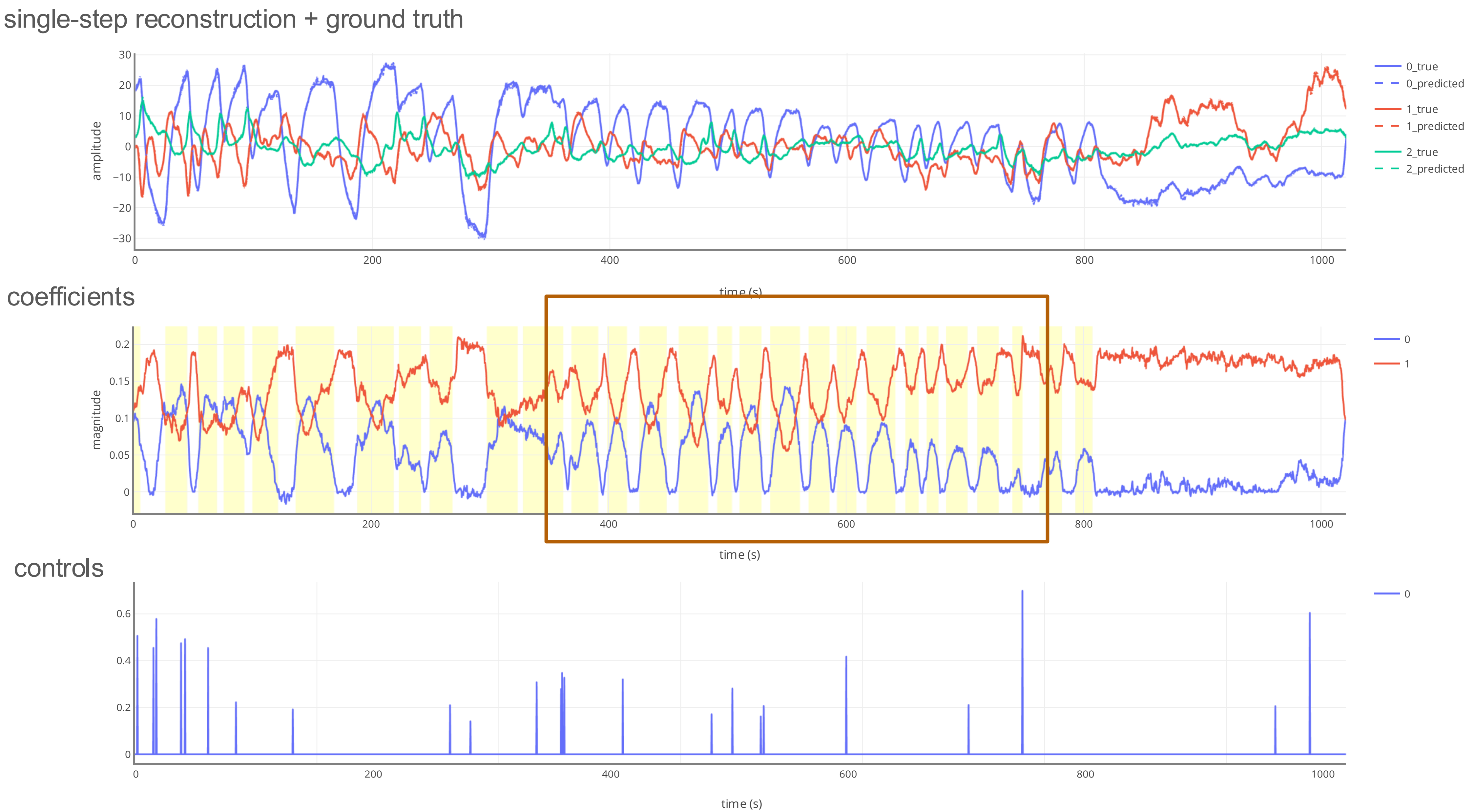
# dLDS applied to Single Whole Brain Recording projected to PC space

single-step reconstruction + ground truth




entire data fitted to PCA, DS2 projected onto that space

# cdLDS applied to Single Whole Brain Recording proj. to Neural Manifold



# Control Signal Initializations

## Potential Control Initialization: Short State Intervals