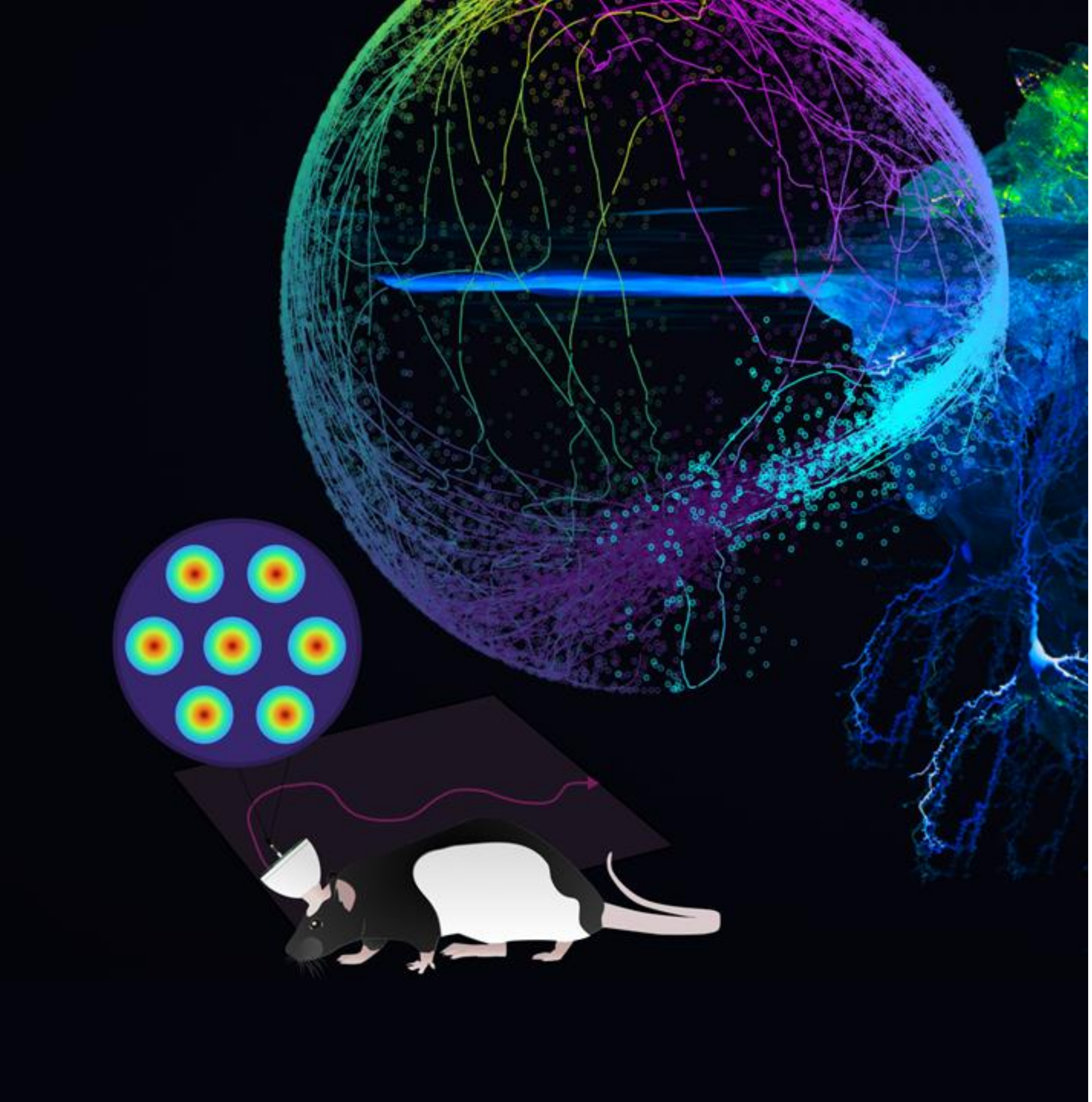
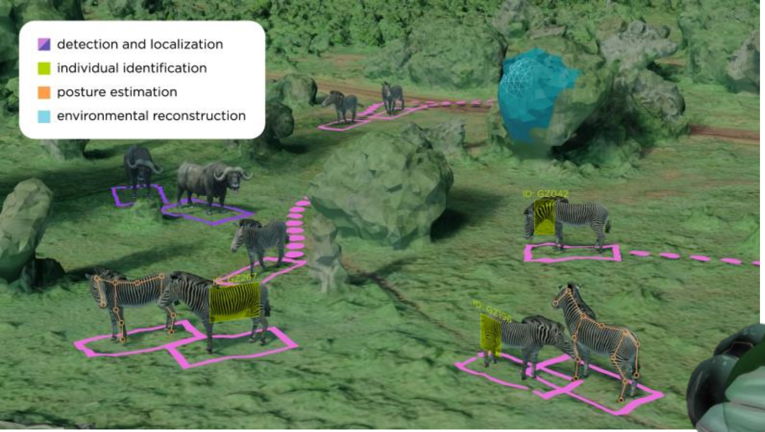
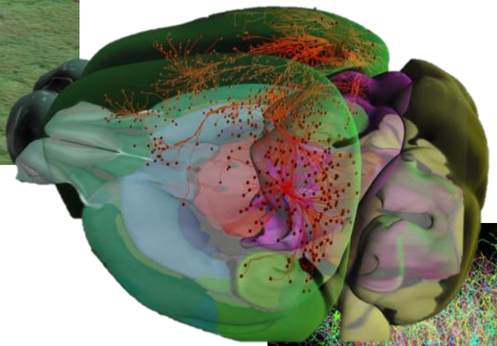


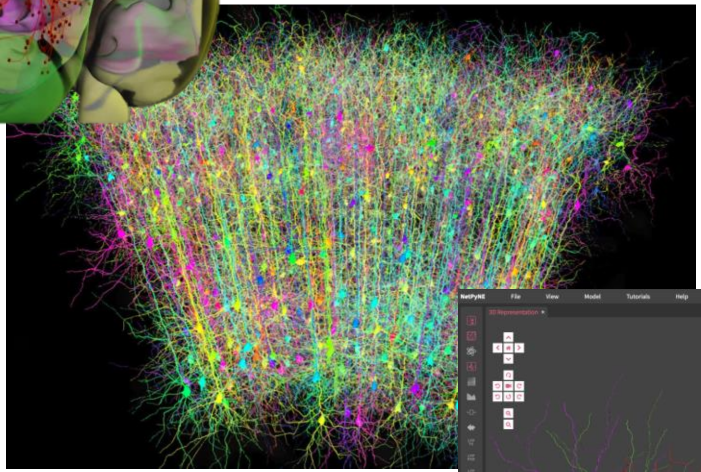
Neural data analysis

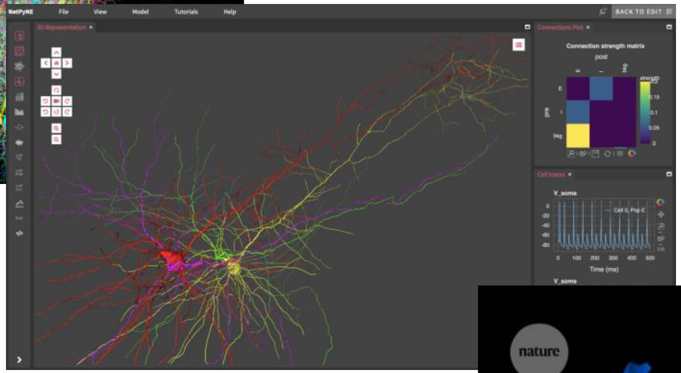
Mackenzie Mathis, PhD

From genes to behavior through the lens of systems neuroscience





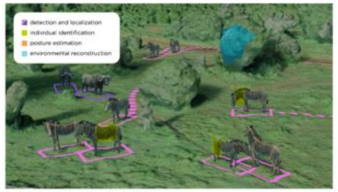


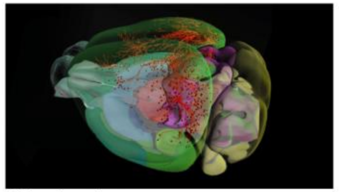


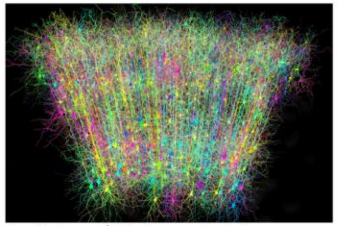
"At this level of analysis, neuroscientists study how different neural circuits analyze sensory information, form perceptions of the external world, make decisions, and execute movements."

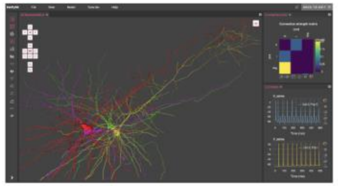
EPFL

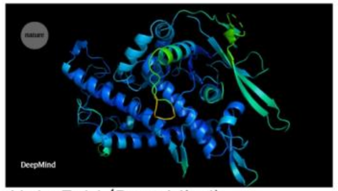
How can we study this?


Tuia et al. 2022


Allen Institute

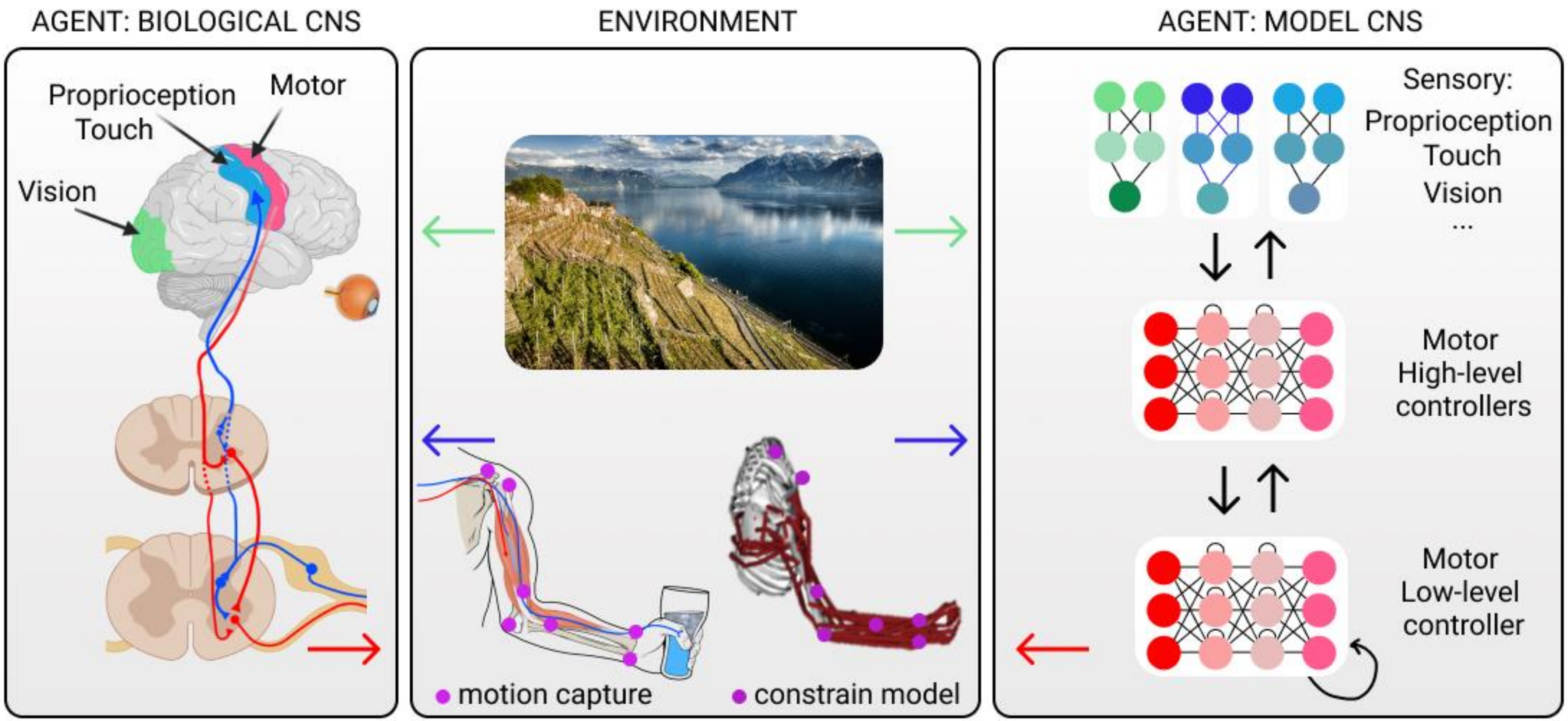

Brainbow (Litchman Lab)


NetPyNE


AlphaFold (DeepMind)

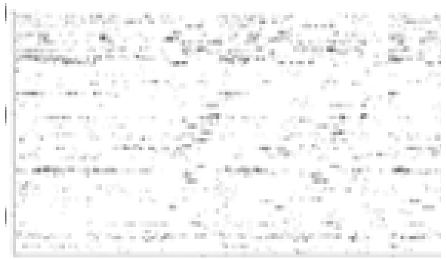
	Experimentally-derived data	Data-driven & Theory-driven models
Environment		**multi-modal ML models**
Behavior	**video recordings, EMGs**	**computer vision & RL**
Brain	**functional imaging (2P, 1P, fMRI, EEG)**	**network neuroscience**
system (circuits)		**RNNs, transformers**
		GLMs, latent variable models
Neurons (& glia)	**pharmacology, patch clamping**	**Hodgkin-Huxley LIF (NEURON, nengo, etc)**
Synapses		**ODEs, LIF**
Genes & molecules, proteins	**RNAseq, proteomics**	**AlphaFold, gene-regulatory networks**

Modeling adaptive behavior



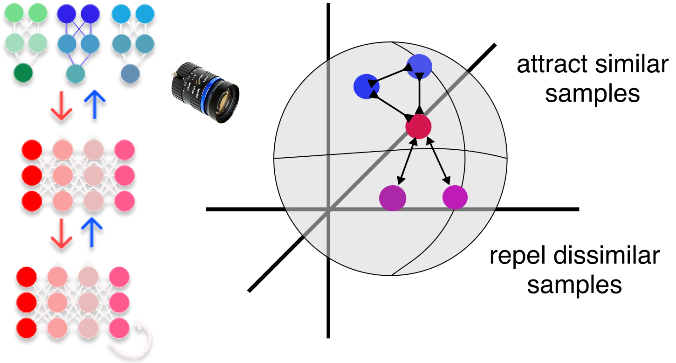
Hausmann et al. Current Opinion in Neurobiology 2021

Data-driven modeling



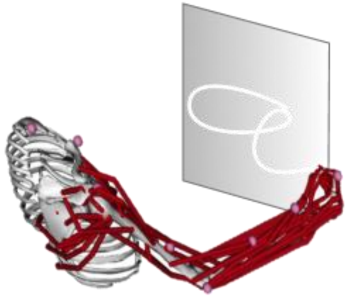
Record from neural data during a behavioral task

GLMs
.
.
.
ANNs



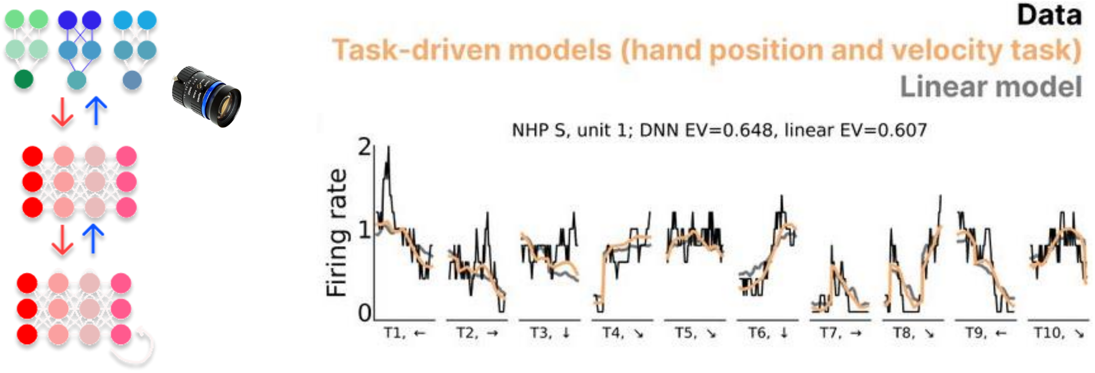
Joint models that describe neural variance & representations

Task-driven modelling



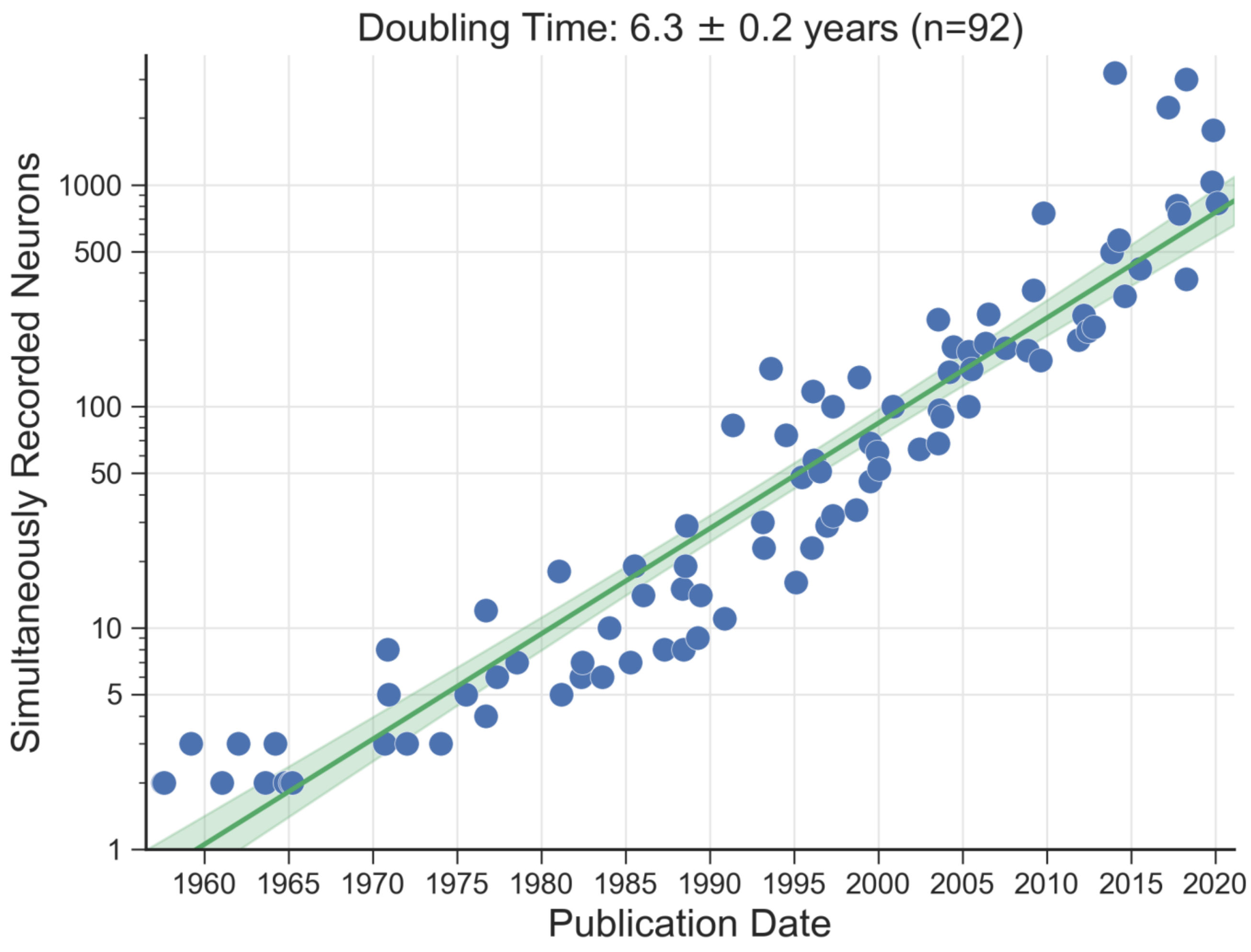
Constrain ANN based on behavioral task to test hypotheses about a system

Sandbrink et al. 2023 eLife



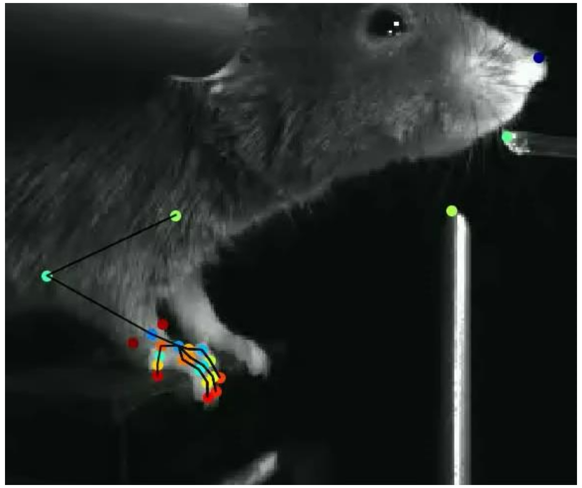
A. Mathis Lab (Marin Vargas et al. in press Cell)

NN models that describe neural variance & computationally constrain system

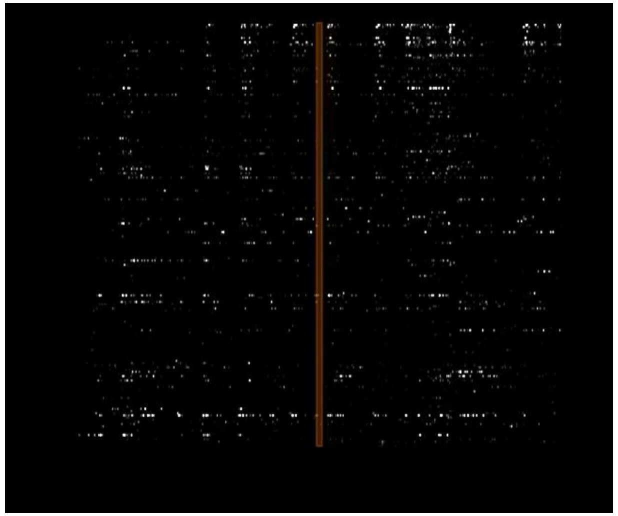


We are now able to record thousands of neurons simultaneously

- Stevenson and Kording (2011), updated in 2022



Keypoint tracking with DeepLabCut

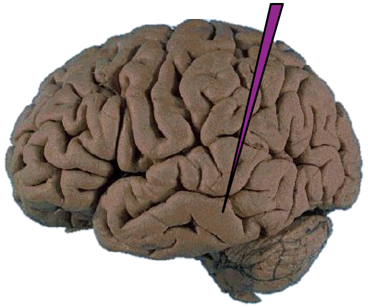


CaImAn: CNN based feature extraction + deconvolution

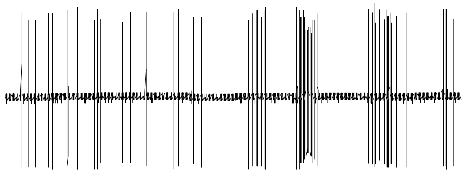
Large-scale behavioral & neural recordings call for new methods to link neural dynamics and behavior

What information is our brain trying to encode & decode?

Stimulus



Spikes



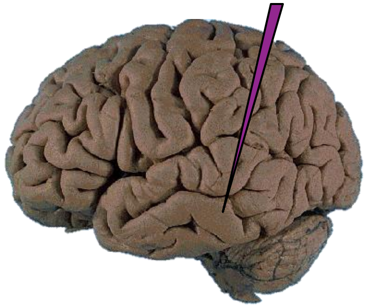
- How do neurons (**K**) respond to a certain stimulus (**x**)?
- Our brain needs to determine what is going on in the real world from patterns of spikes.

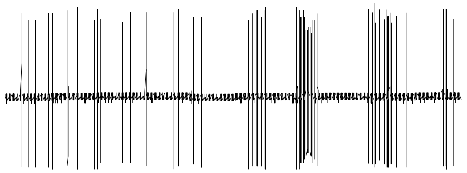
- We mathematically model this as P(**K|x**), where the neural response of population **K** to a stimulus (or event) **x.**
- **K** is a vector representing the activity of *N* neurons, and each entry represents, e.g., the number of spikes in some time bin or the rate response of that particular neuron.

What information is our brain trying to encode & decode?

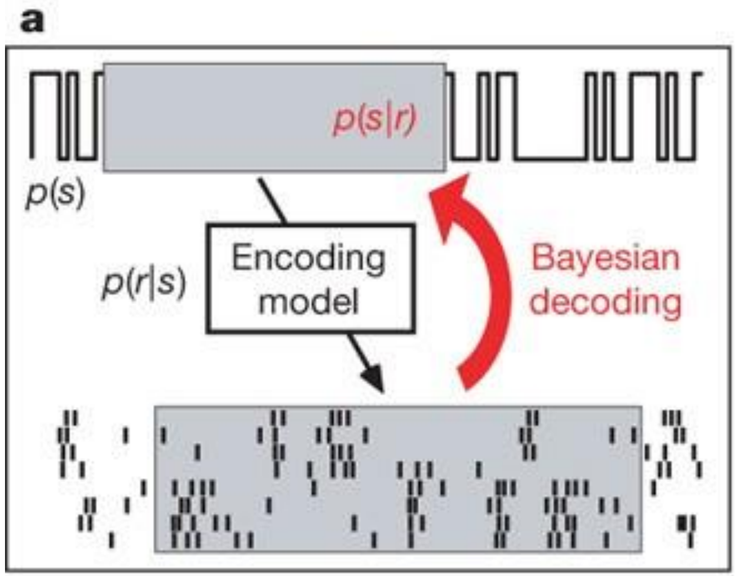
Stimulus

Spikes



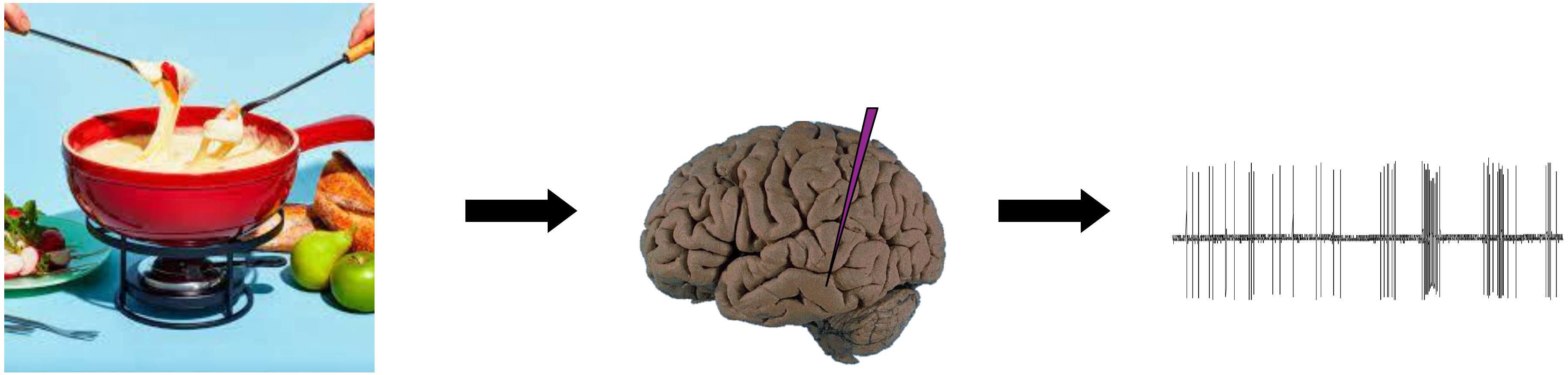


- Our brain needs to determine what is going on in the real world from patterns of spikes.



What information is our brain trying to encode?

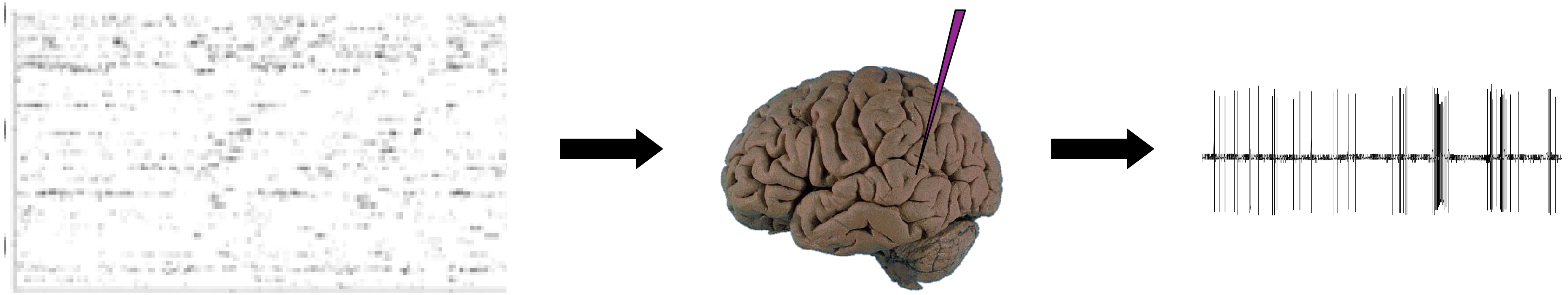
Stimulus Spikes



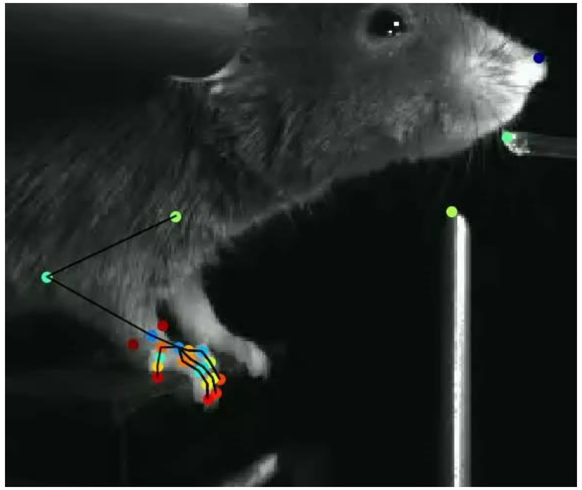
- Pixels on a screen (Pillow et al. 2008)

- *Events, like behavior: moving through an environment with arm or body*

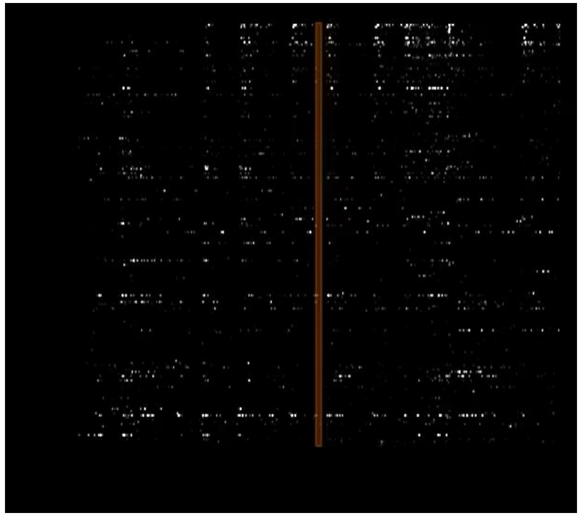
- **other neurons!**



Mapping behavioral actions to neural computations



behavioral monitoring



neural recordings in vivo

When recording from just 20 neurons, there are over one million possible instantaneous ON/OFF patterns of spiking (2^{20}) for a small time bin; this number grows to one billion for 30 neurons (2^{30})

In reality, this is much smaller due to neural connectivity constrains, but estimating all pattern frequencies reliably from typical recordings is impossible even for small populations!

Fully observed statistical models are tools that allow us to approximate the firing distribution with reasonable constraints …

Adapted from C. Hurwitz 2021 Current Opinion Neuro

Computational methods for mapping (K) neural activity to a certain stimulus (x)

Table 1

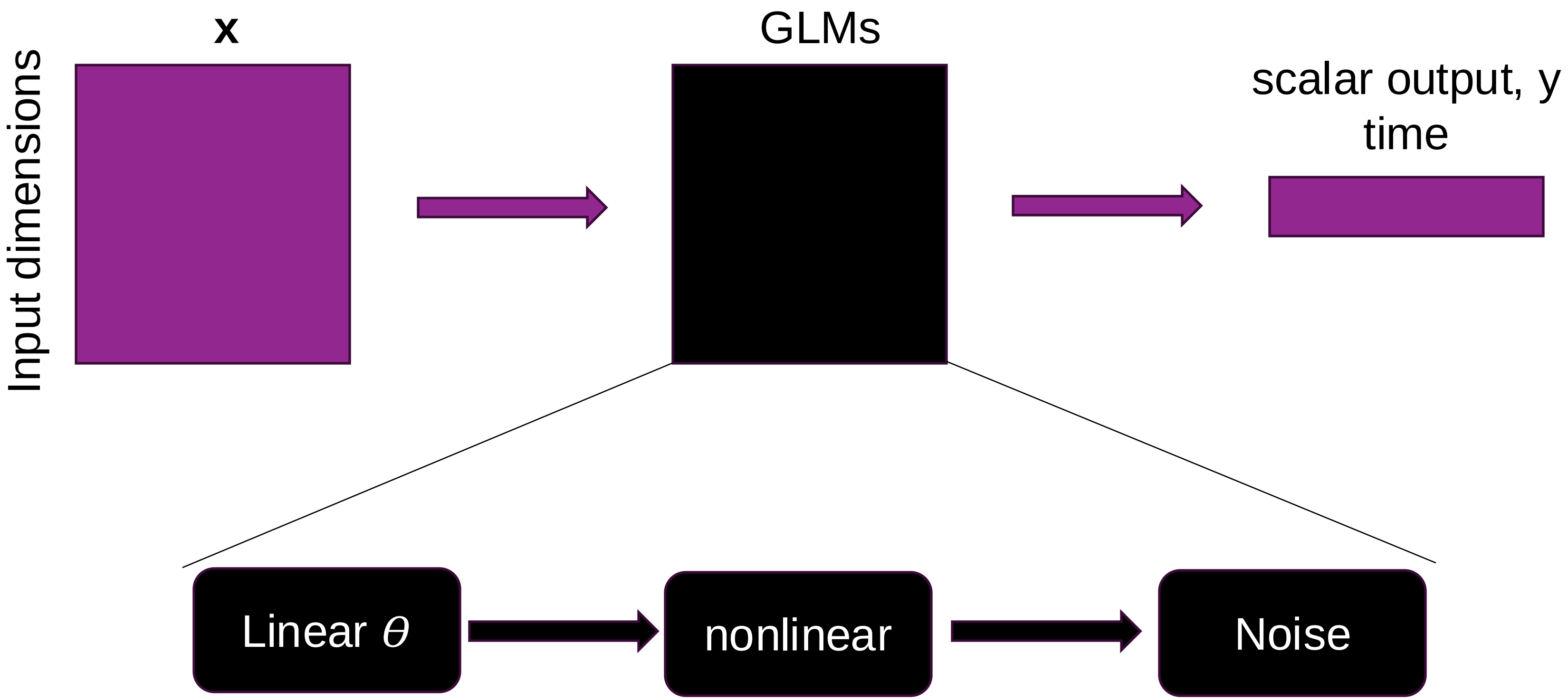
Fully observed models: A table (adapted from the study by O'Donnell et al. [50]) characterizing model properties and limitations. Here, *N* is the number of neurons and *D* is the number of coefficients per interaction term, such as filter sizes for GLM or number of parameters for parametric copula families. Sampling is possible from all of these models.

Model	References	Number of parameters	Closed-form pattern probabilities?	Fit for large *N*?
Dichotomized Gaussian	Amari et al. [2] Macke et al. [42]	$\sim N^2$	No	Yes
Pairwise MaxEnt	Schneidman et al. [79] Shlens et al. [82]	$\sim N^2$	No[a]	Difficult
Tractable MaxEnt	Tkačik et al. [87] Gardella et al. [24] O'Donnell et al. [50]	$\sim N^2$	Yes[a]	Yes
GLM	Pillow et al. [60]	$\sim DN^2$	No	Difficult
Vine Copula	Aas et al. [1] Onken and Panzeri [53]	$\sim DN^2$	Yes	Yes

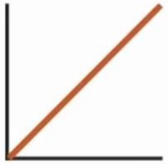
[a] For a more detailed comparison of MaxEnt models, see Table 1 in Ref. [77].

Adapted from C. Hurwitz 2021 Current Opinion Neuro

Generalized Linear Models



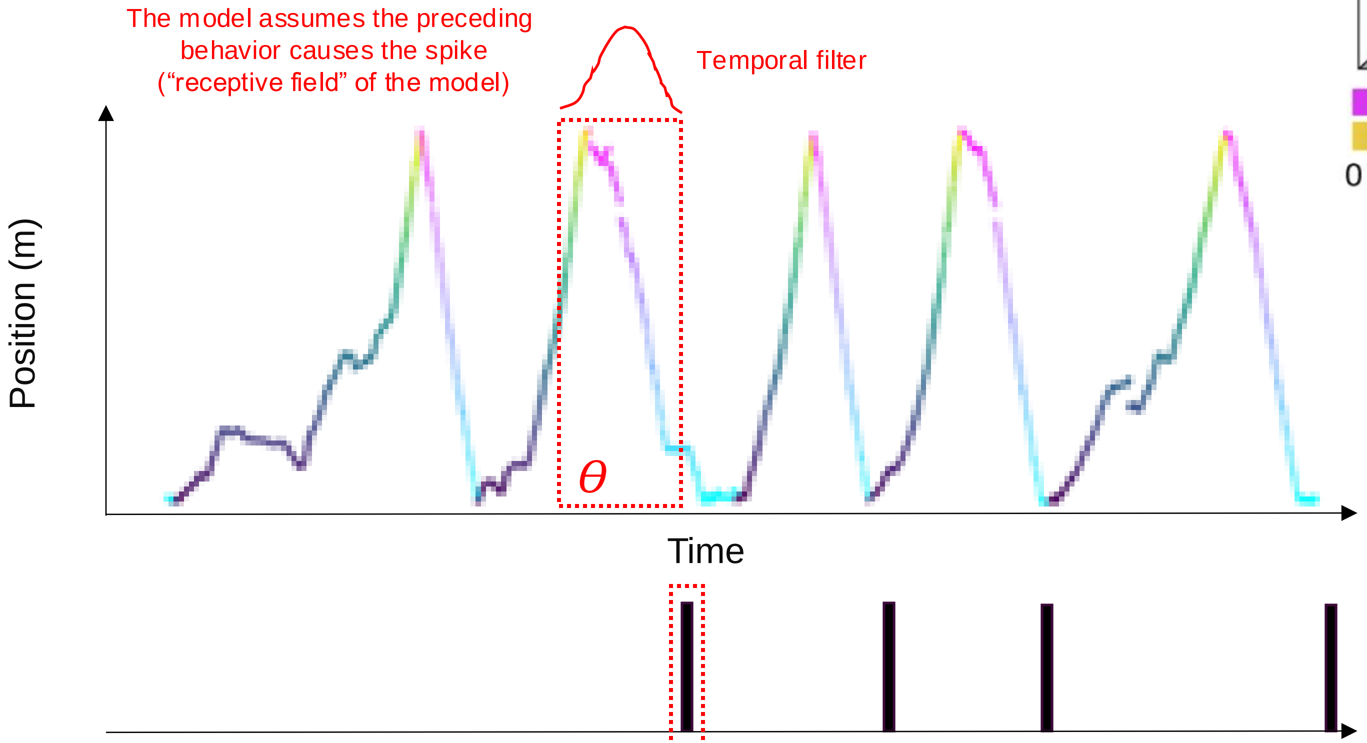
Generalized Linear Models: many variants!

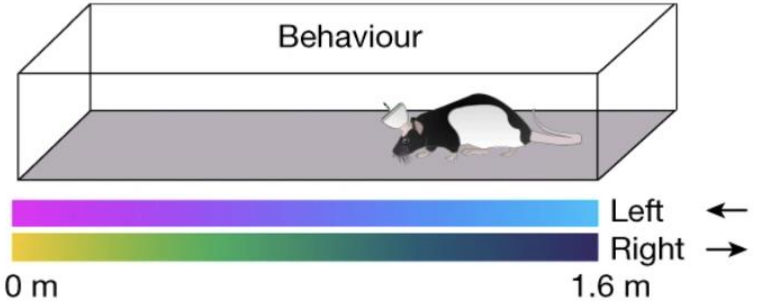
Model Name	Likelihood	Nonlinearity	Output type
 Linear regression	Gaussian $P(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp - \frac{(y-\mu)^2}{2\sigma^2}$	identity $\mu = \boldsymbol{\theta}^\top \mathbf{x}$	real values
 Poisson GLM	Poisson $P(y) = \frac{\lambda^y \exp(-\lambda)}{y!}$	exponential $\lambda = \exp\left(\boldsymbol{\theta}^\top \mathbf{x}\right)$	discrete counts 0,1,2,3…
 Logistic regression	Bernoulli $P(y) = p^y (1-p)^{1-y}$	logistic $p = f\left(\boldsymbol{\theta}^\top \mathbf{x}\right)$	binary 0,1

Adapted from Dr. Cristina Savin and neuromatch academy

Logistic regression

Stimulus (**x**), here behavior of the animal





Data: Grosmark et al. 2016 Science

$$p_t = f\left(\sum_i \theta_i \; x_{t-i}\right)$$

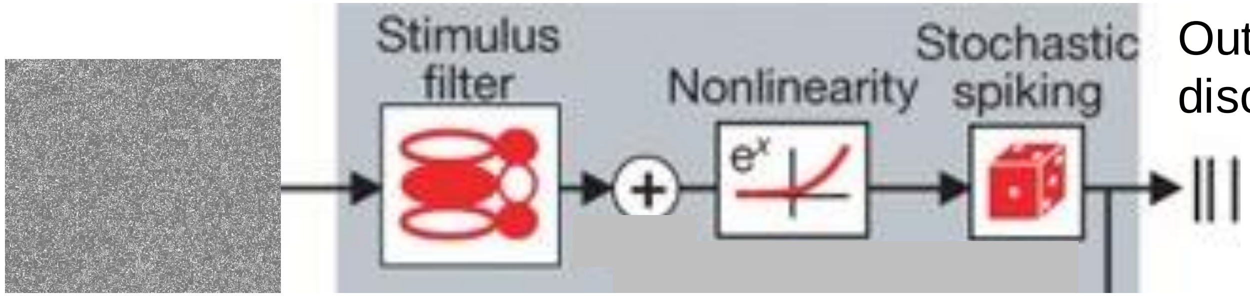
Neuron response

0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1

Poisson GLM

θ

$$P(y_t = y) = \frac{\lambda^y e^{-\lambda}}{y!}$$



Output:
discrete spikes

- **In the model, each neuron's input is described by a set of linear filters:**
  - a stimulus filter, or spatial receptive field (θ)

$$p_t = exp\left(\sum_i \theta_i \; x_{t-i}\right)$$

- image adapted from JW Pillow *et al. Nature*, 1-5 (2008) doi:10.1038/nature07140

Details: Poisson GLM

Poisson Distribution: Single Event

- Probability of events y_t at time *t*
- Formula:

$$P(y_t = y) = \frac{\lambda^y e^{-\lambda}}{y!}$$

- λ: rate parameter (average number of events)
- y!: factorial of y (number of events)

Likelihood: All Data Points

- Joint probability as the product of individual probabilities
- Formula

$$P(y_1 : T) = \prod_t P(y_t)$$

- Assumes independence between data points

Simplify!!

Log Likelihood

$$\log \mathcal{L} = \sum_t \log P(y_t)$$

Link Function and Predictors

- λ linked to predictors x_t
- Formula:

$$\lambda = \exp(\theta^T x_t)$$

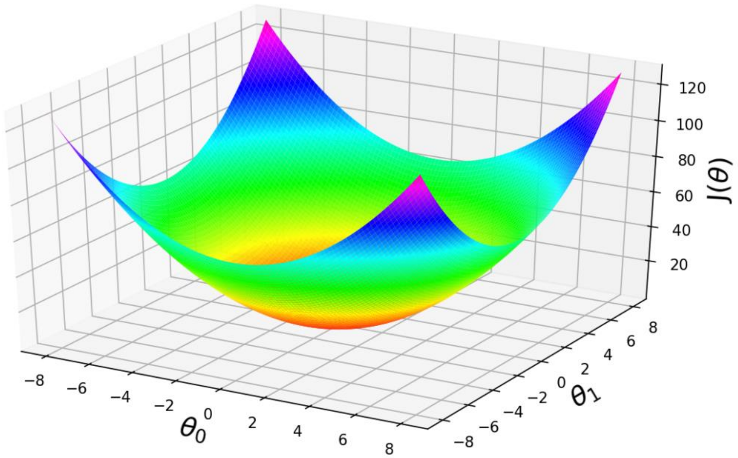
- θ : model parameters (as a vector)
- x_t : predictors vector

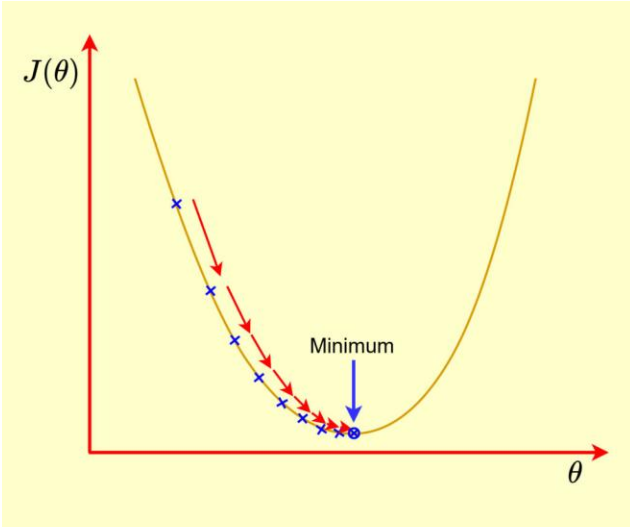
Details: fitting the Poisson GLM

Fit the model to find parameters that maximize the log-likelihood:

$$\boldsymbol{\theta}^* = \text{argmax}_\theta \log \mathcal{L}(\boldsymbol{\theta})$$

Since the space is concave (a nice feature), in practice, we invert it so we can use gradient descent



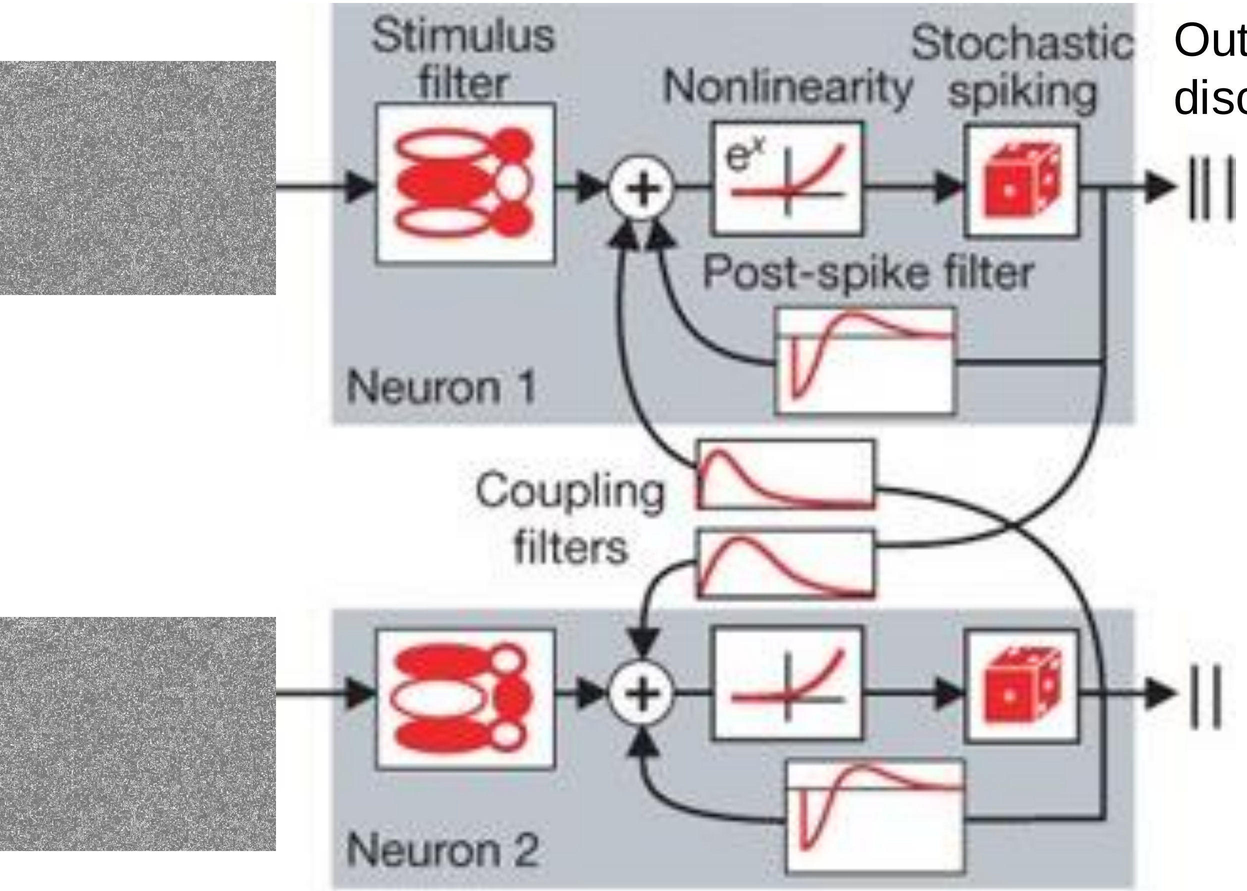


Images from: https://machinelearningspace.com/a-comprehensive-guide-to-gradient-descent-algorithm/

Poisson GLM

θ

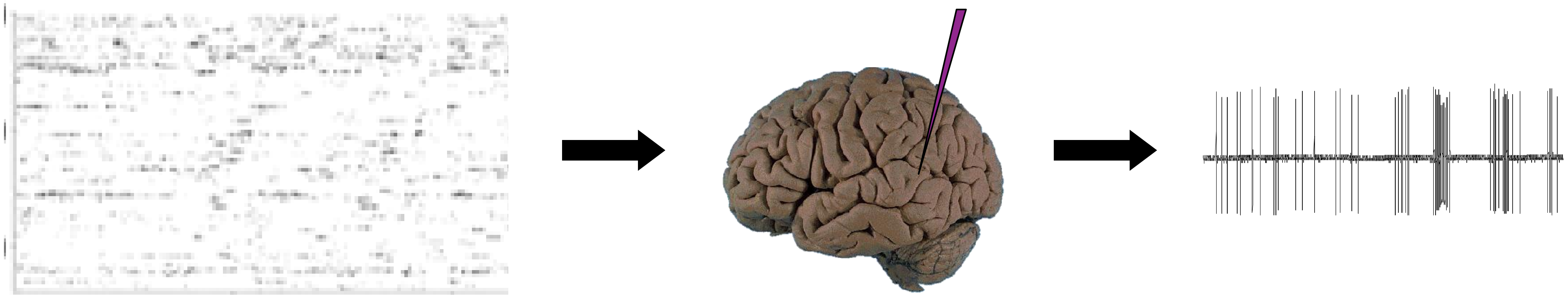
$$P(y_t = y) = \frac{\lambda^y e^{-\lambda}}{y!}$$

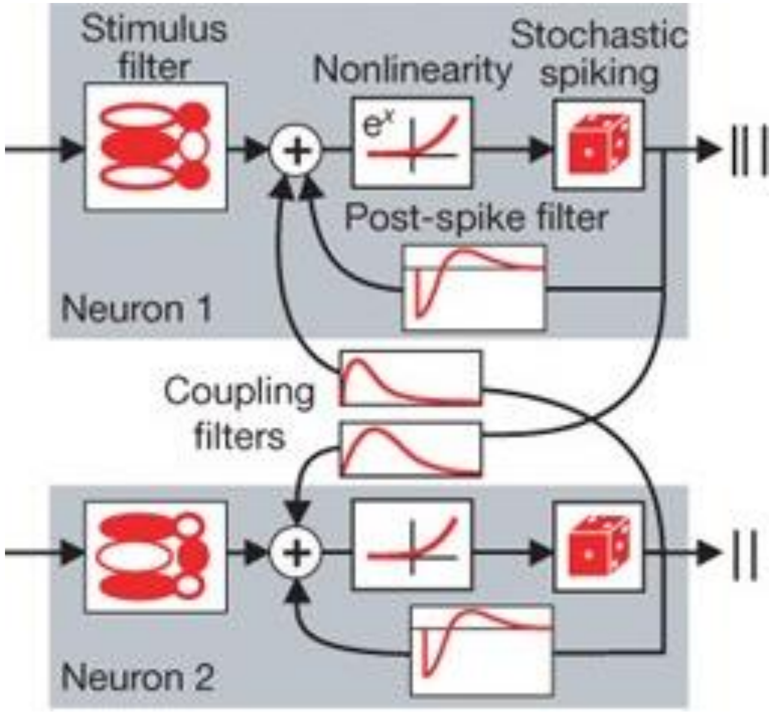


Output:
discrete spikes

- **In the model, each neuron's input is described by a set of linear filters:**
  - a stimulus filter, or spatio-**temporal** receptive field;
  - a post-spike filter, which captures dependencies on spike-train history (for example, refractoriness, burstiness and adaptation);
  - a set of coupling filters, which capture dependencies on the recent spiking of other cells

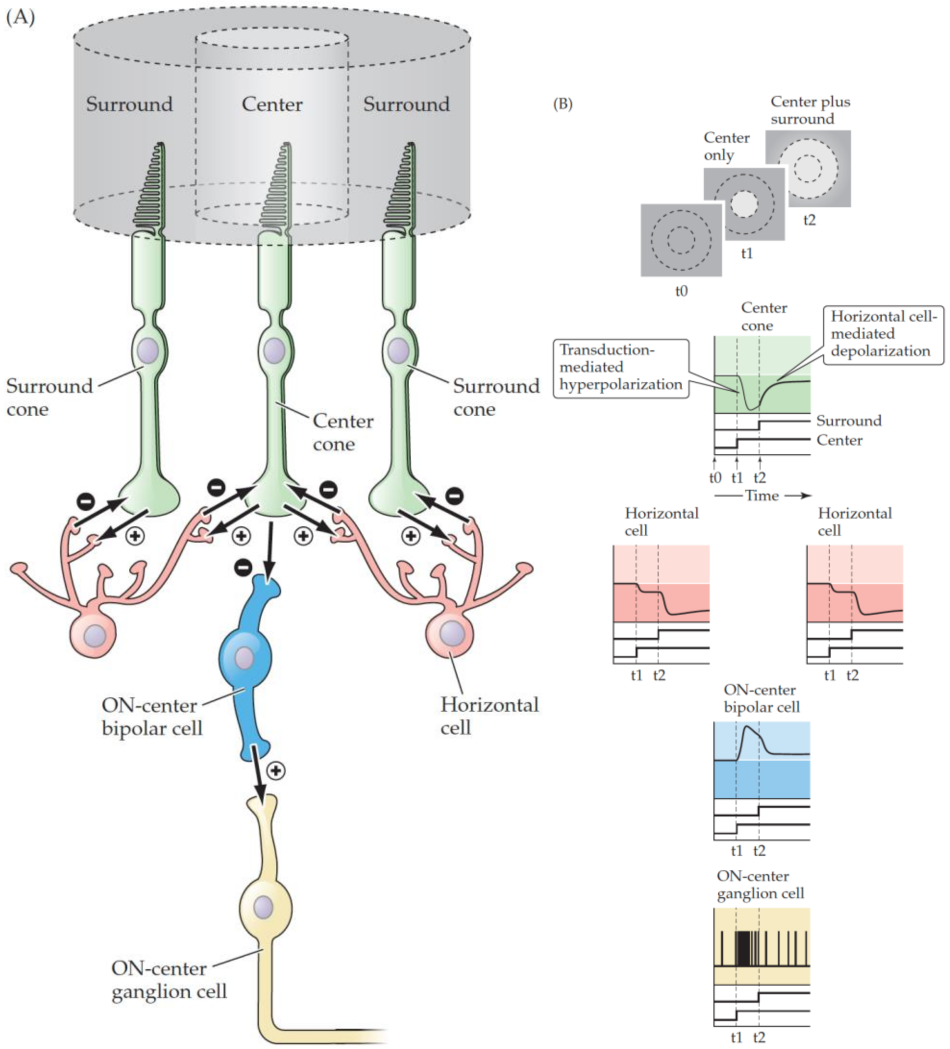
- JW Pillow *et al. Nature*, 1-5 (2008) doi:10.1038/nature07140





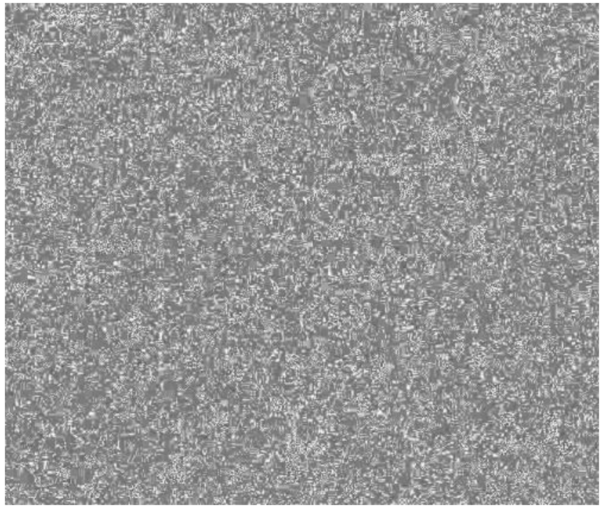
This model allows for accounting for “other” neurons activity patterns

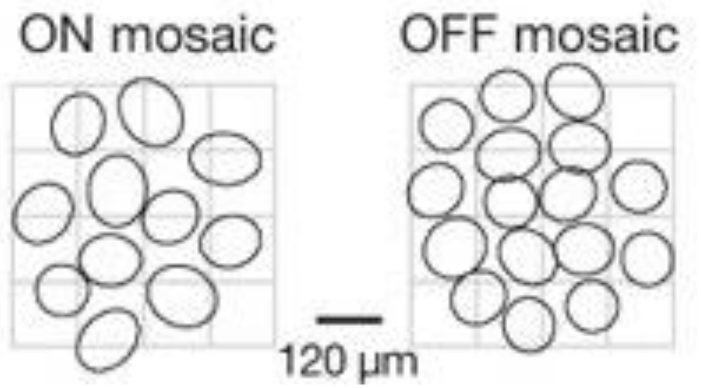
Experimental set up: the retina ganglion cell



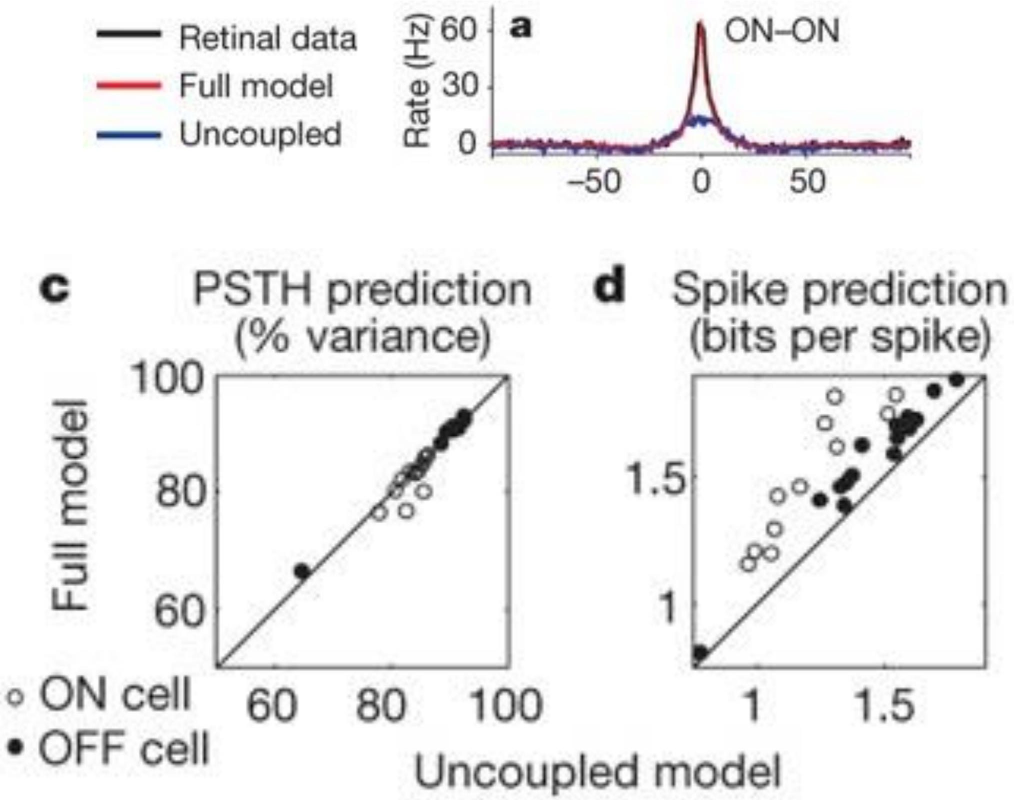
- Purves, Figure 11.21

In Pillow et al 2008, they stimulate RGCs with white noise:

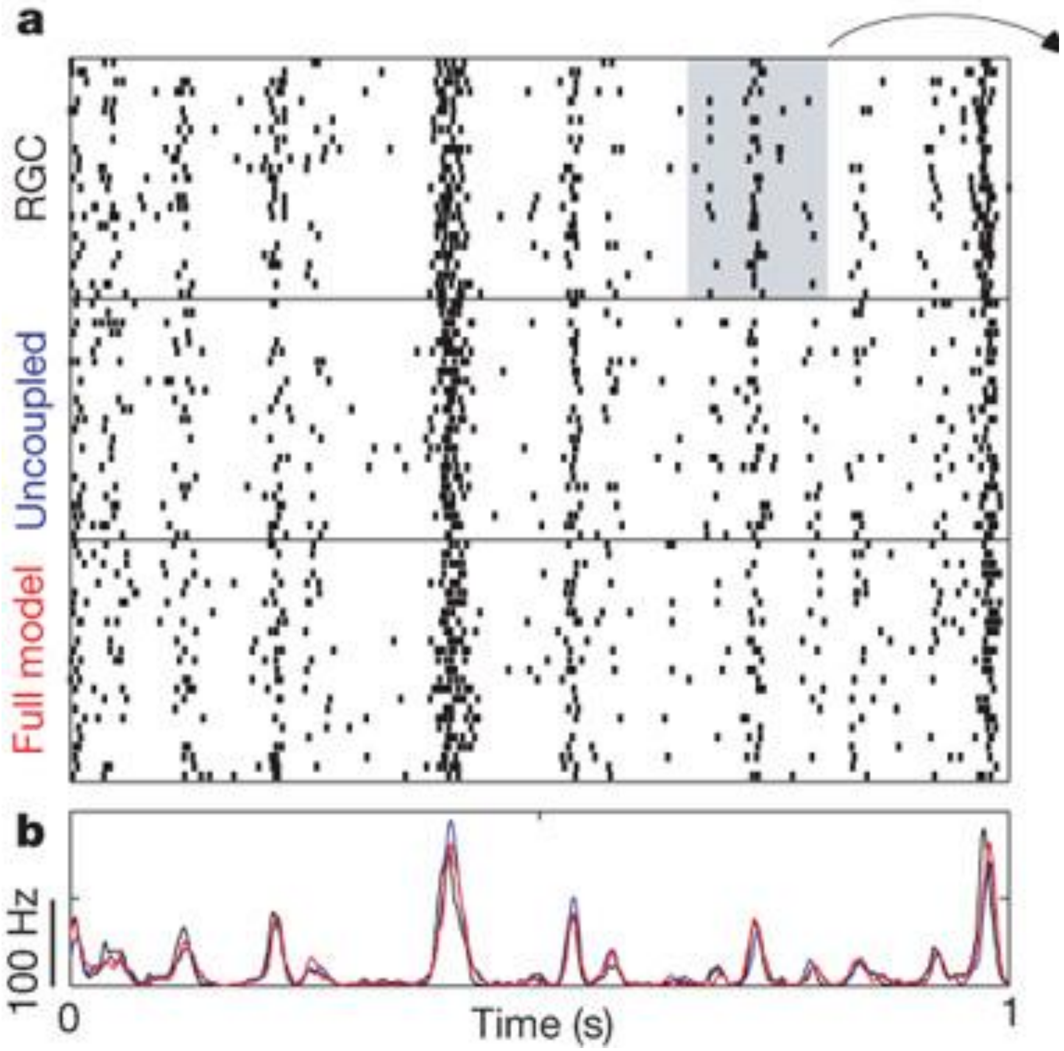




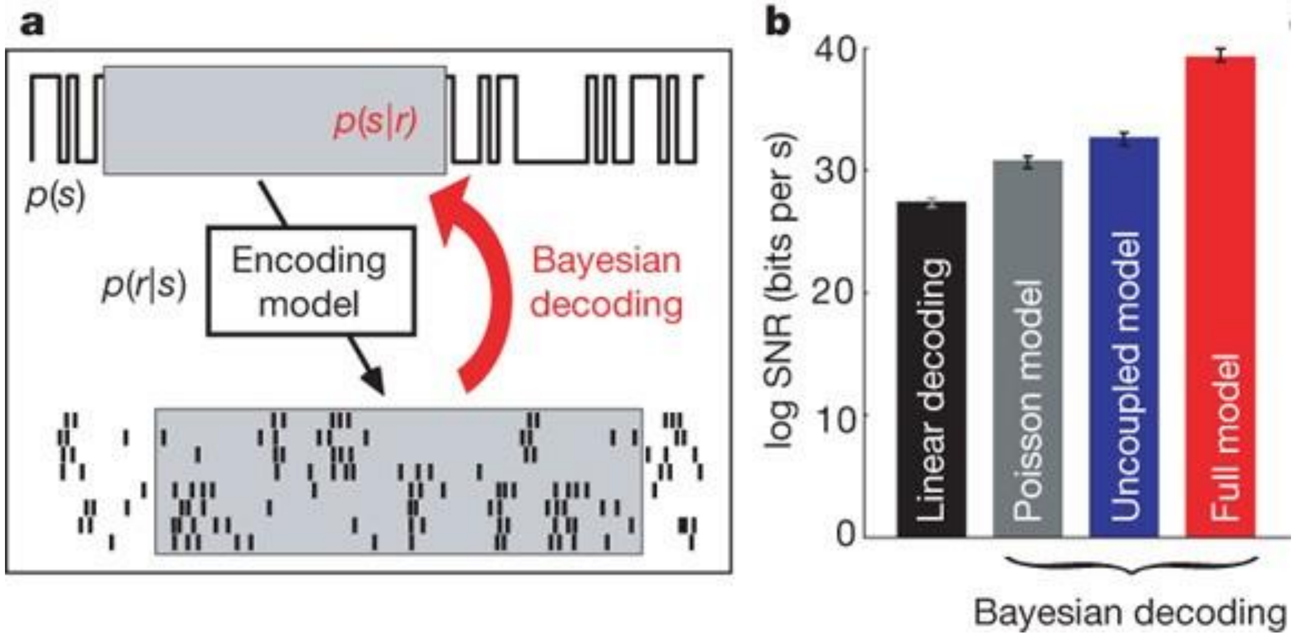
GLMs in action: Pillow et al. 2008



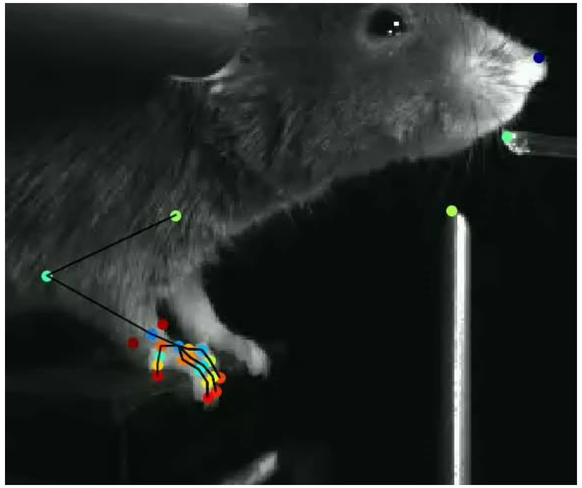
PSTH →



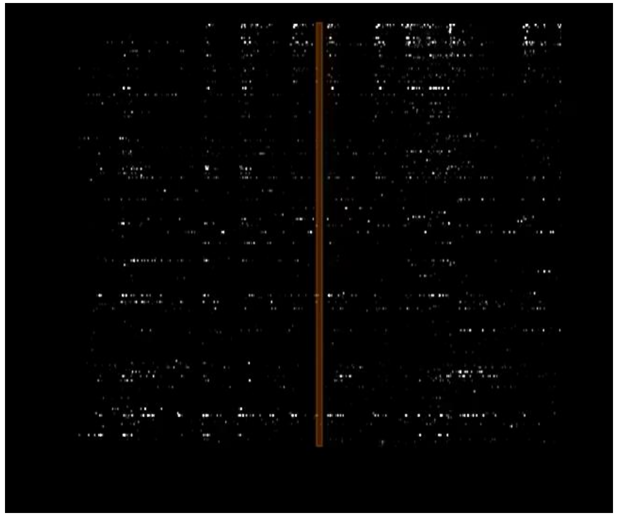
GLMs in action: Pillow et al. 2008



GLMs with coupling filters were shown to capture 40% more visual information from the retina than optimal linear decoding, indicating that GLMs can model additional details in the activity that are relevant for representing the stimulus!



Keypoint tracking with DeepLabCut



CaImAn: CNN based feature extraction + deconvolution

Large-scale behavioral & neural recordings call for new methods to link neural dynamics and behavior

Summary, part 1

- **Neural encoding** and **neural decoding** are fundamental descriptions of neural (coding) processing and data analysis.

- A fundamental goal is: how much information does **K** have about **x**

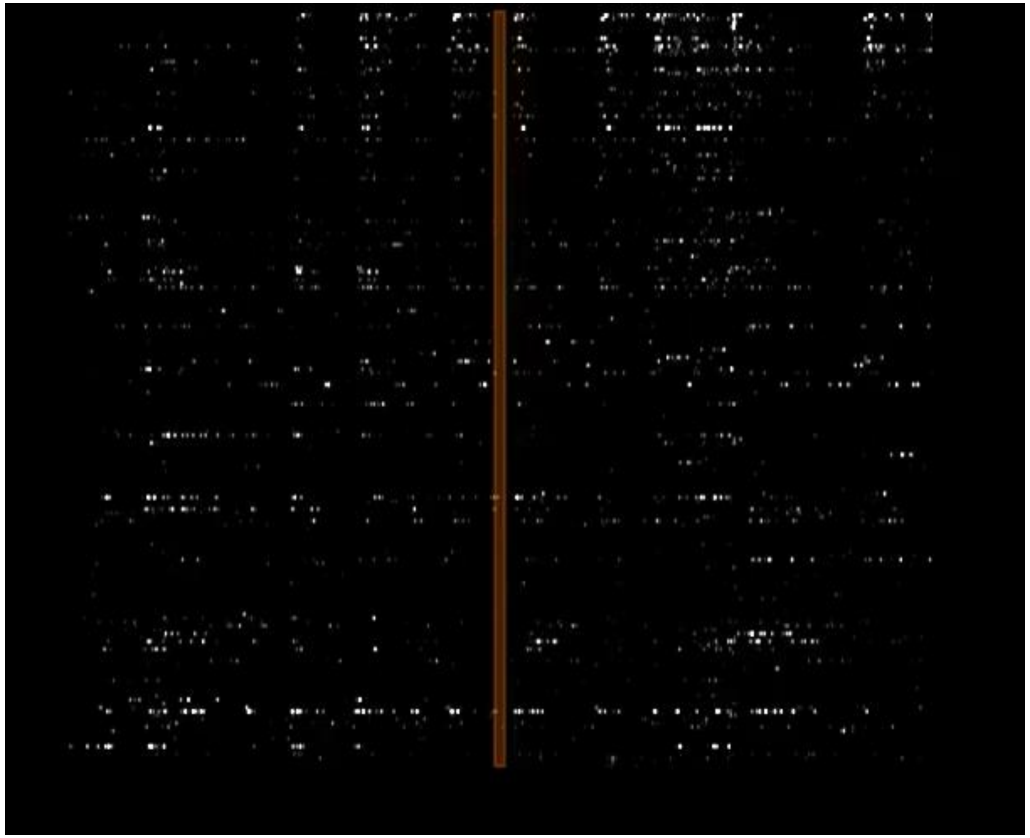
- We mathematically model this as P(K|x), where the neural response of population **K** to a stimulus (or event) **x**. **K** is a vector representing the activity of *N* neurons, and each entry represents, e.g., the number of spikes in some time bin or the rate response of that particular neuron.

- **Generalized Linear Models** (GLMs) are very attractive for both individual neurons and populations, yet assume **linear** ***θ*** dynamics (careful: despite having a nonlinear parameter).

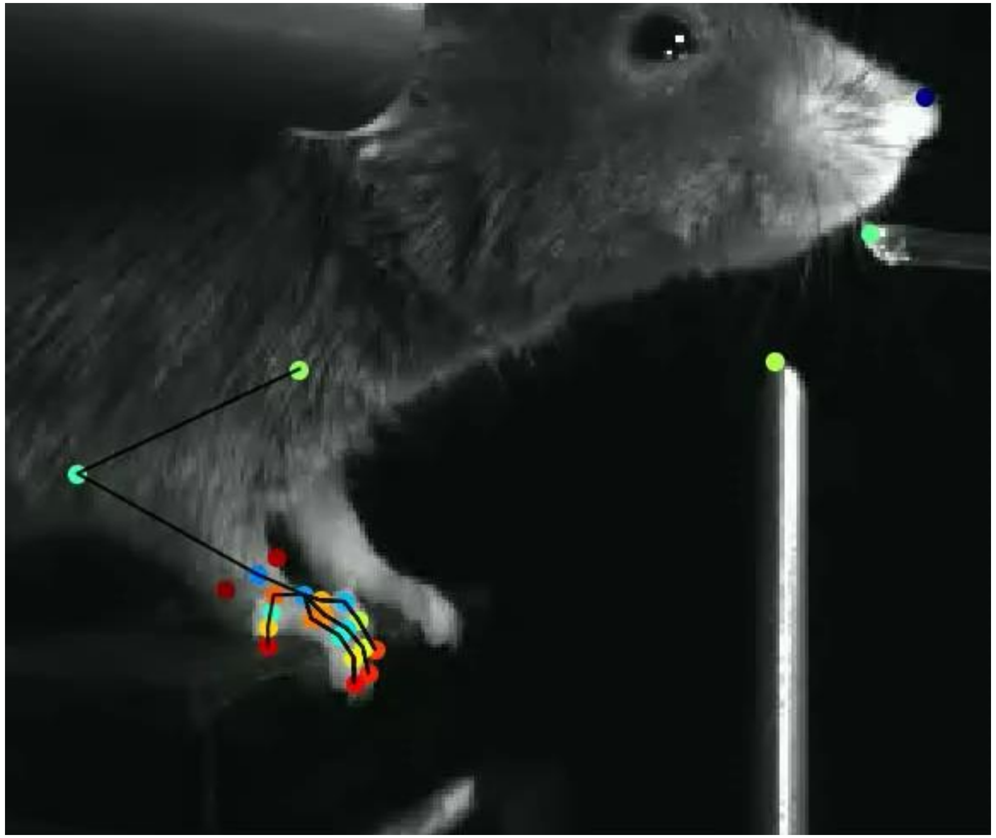
From GLMs to more powerful models ...

- In P-GLMs, external inputs, inputs from other neurons, and each neuron's spiking history are weighted, summed up, and transformed by a rate function, which drives a stochastic process to model spiking activity.
- For typical rate functions, fitting P-GLMs is a convex optimization problem that can be solved efficiently
- **Despite their flexibility, GLMs cannot model networks with neurons that perform non-linear integration of multiple external inputs….**

Mapping behavioral actions to (nonlinear) neural computations



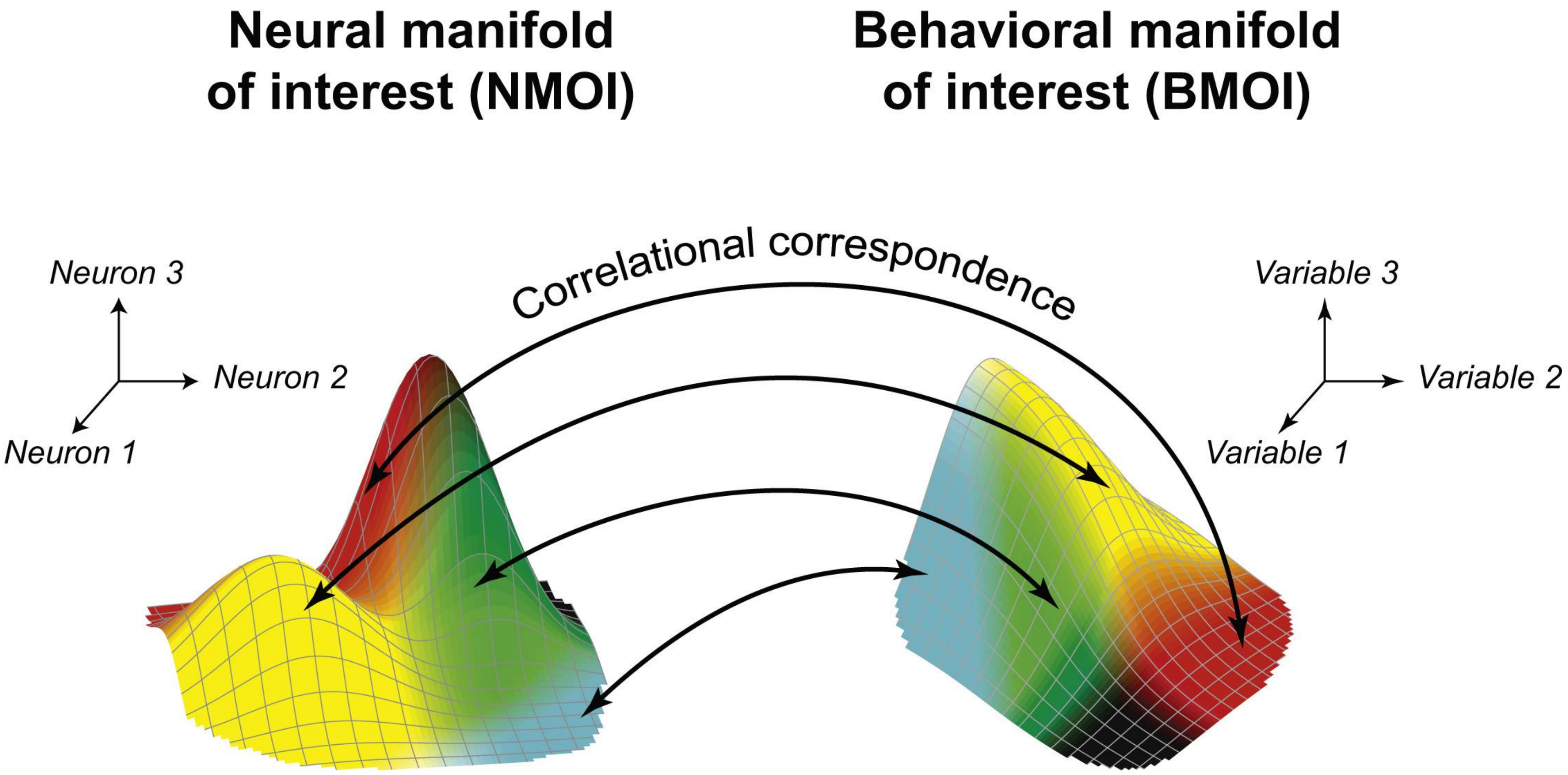
CaImAn: CNN based feature extraction + deconvolution



Keypoint tracking with DeepLabCut

Mathis Lab Data

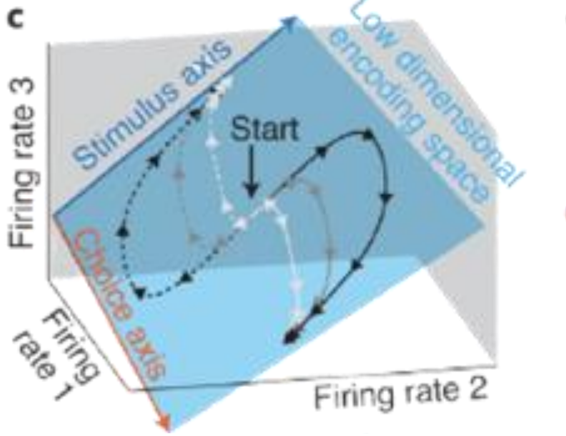
Manifolds (embeddings) for measuring neural trajectories



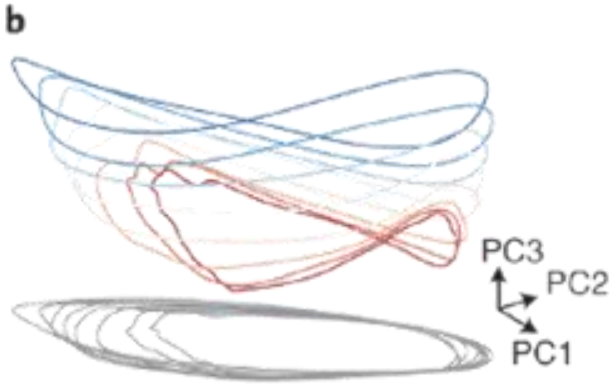
Mehrdad Jazayeri and Arash Afraz Neuron 2017

Population analysis can reveal core principles of neural coding

Urai et al Nat Neurosci 2022



Behaviorally relevant neural variance within a small number of dimensions

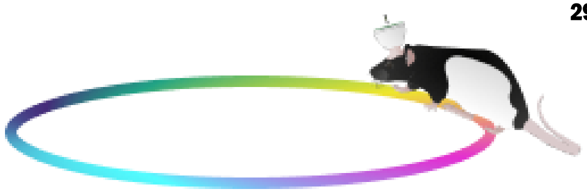


Neural computations at population dynamics but not in single-neuron firing rates

Chaudhuri, R. et al. Nat Neurosci 2019

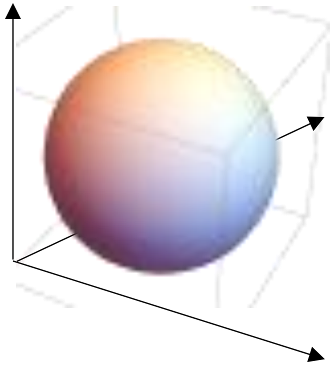
Intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep

1



1

2



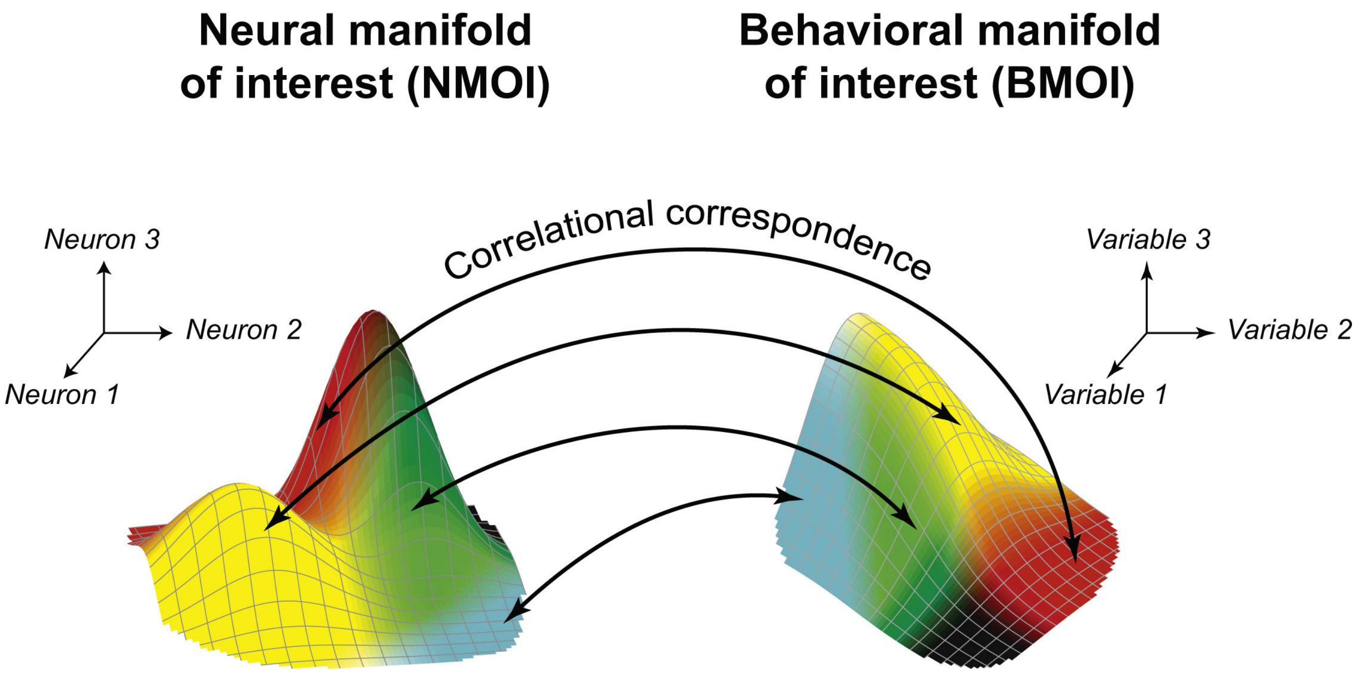
Intrinsic Dimension

Inspired by Jazayeri & Ostojic Current Opinion Neurobio 2021

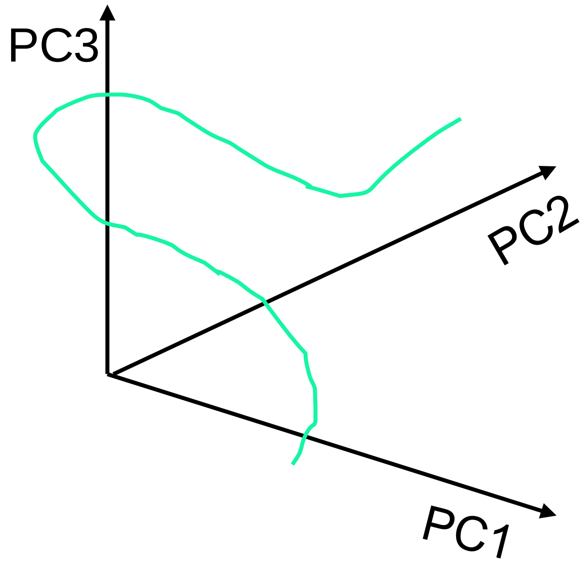
See also: Mante et al Nature (2013), Churchland et al Nature (2012), Sadtler et al Nature (2014), Elsayed & Cunningham Nat Neurosci (2017)

Manifolds for measuring neural trajectories

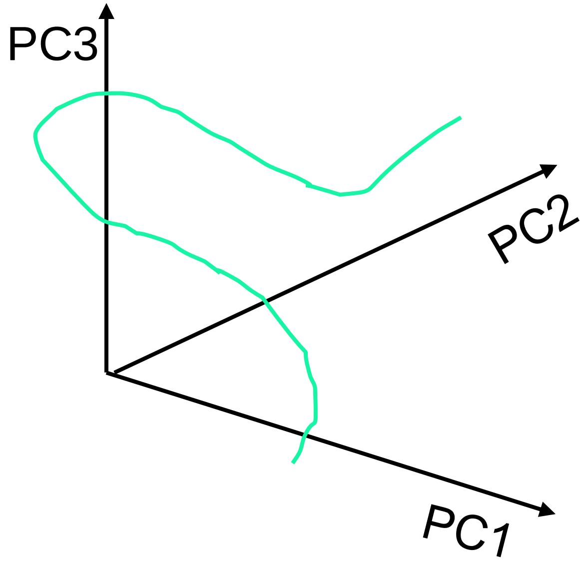
Linear (PCA) methods for measuring neural trajectories

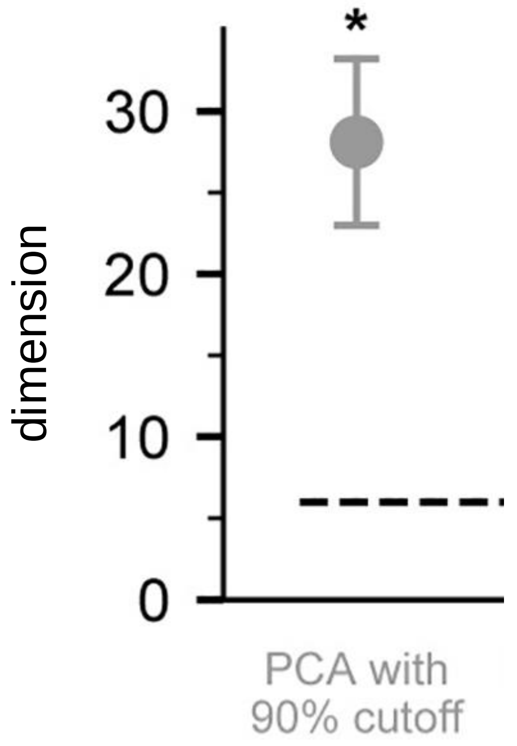


Mehrdad Jazayeri and Arash Afraz Neuron 2017



Linear (PCA) methods for measuring neural trajectories





Altan E, Solla SA, Miller LE, Perreault EJ. PLoS Comput Biol. 2021

EPFL

Statistical models for capturing the statistical structure of large-scale neural populations

	Model	Trajectories (Dynamics)	Mapping Function ^	Observation	Single-trial[a]
	PCA/FA [68]	Static	Linear	Gaussian	No
	dPCA [100]	Static	Linear	Gaussian	No
	jPCA [13]	Linear[b]	Linear	Gaussian	No
	LDS [84]	Linear	Linear	Gaussian	Yes
	PLDS [43]	Linear	Linear	Poisson	Yes
	PSID [75]	Linear	Linear	Gaussian[c]	Yes
*	PfLDS [23]	Linear	Neural Network	Poisson	Yes
	SLDS [56]	Switching Linear	Linear	Gaussian	Yes
	RSLDS [41,97]	Recurrent Switching Linear	Linear	Gaussian	Yes
	PLRNN-SSM [101]	Piecewise-linear RNN	Linear	Gaussian	Yes
	LFADS [55]	RNN	Linear	Poisson	Yes
	GP-RNN [80]	RNN	GP	Poisson/Gaussian	Yes
	GPFA [11]	GP	Linear	Gaussian	Yes
	GPFADS [72]	GP[d]	Linear	Gaussian	Yes
	vLGP [95]	GP	Linear	Poisson[e]	Yes
	P-GPLVM [93]	GP	GP	Poisson	Yes
*	piVAE	GIN Flow	Non-linear	Poisson	Yes

- ^ Mapping back to spikes

Adapted from C. Hurwitz 2021 Current Opinion Neuro

Nonlinear embeddings via linear dynamical system (LDS)

Dynamics of *n* neurons are modulated by LDS w/ *m*-dim latent state (z) that evolves:

$$\mathbf{z}_{r1} \sim \mathcal{N}(\mu_1, \mathbf{Q}_1)$$
$$\mathbf{z}_{r(t+1)}|\mathbf{z}_{rt} \sim \mathcal{N}(\mathbf{A}\mathbf{z}_{rt}, \mathbf{Q}),$$

A = linear dynamics matrix (m x m)
Q_1 = covariance of initial states
Q = Gaussian noise

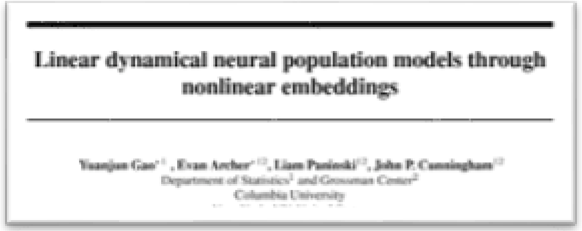
Observation model:

$$x_{rti}|\mathbf{z}_{rt} \sim \mathcal{P}_\lambda\left(\lambda_{rti} = [f(\mathbf{z}_{rt})]_i\right).$$

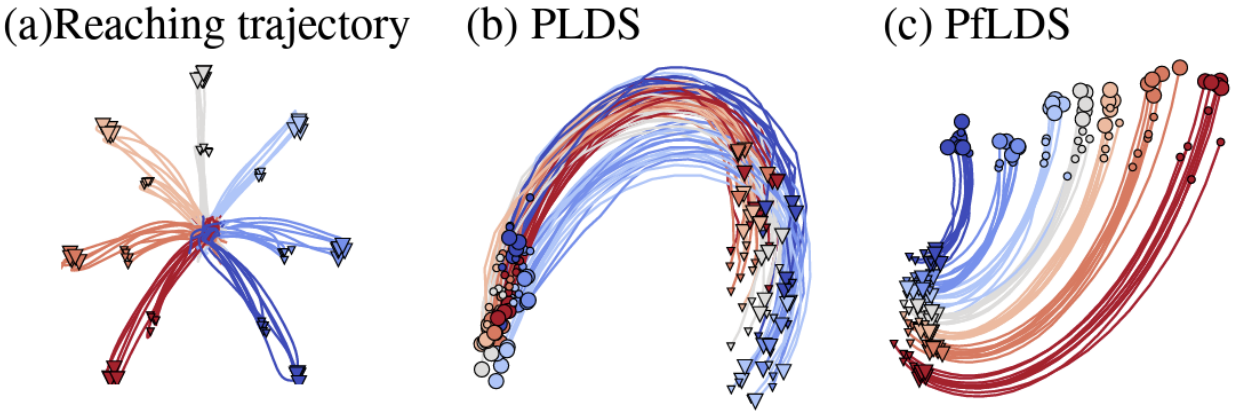
where $[f(\mathbf{z}_{rt})]_i$ is the i^{th} element of a deterministic "rate" function $f(\mathbf{z}_{rt}) : \mathbb{R}^m \to \mathbb{R}^n$, and $\mathcal{P}_\lambda(\lambda)$ is a noise model with parameter λ.

fLDS: exchange observation model such that each neuron as a separate nonlinear dep. on latent variable:

$$x_{rti}|\mathbf{z}_{rt} \sim \mathcal{P}_\lambda\left(\lambda_{rti} = [f_\psi(\mathbf{z}_{rt})]_i\right),$$

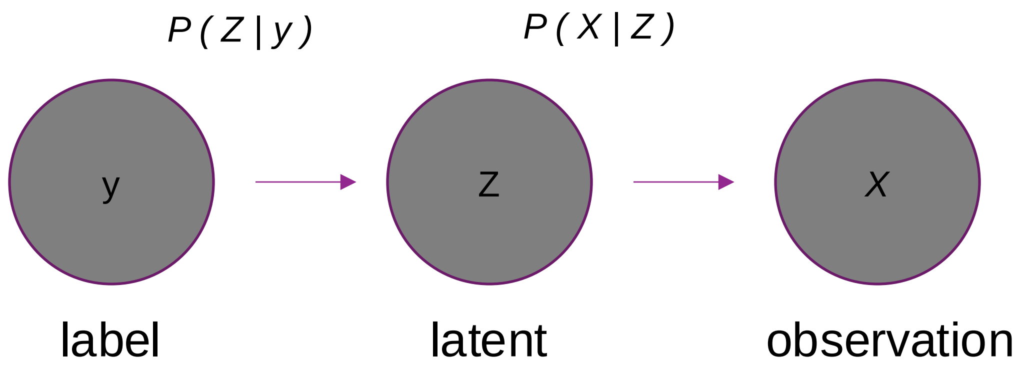

Linear dynamical neural population models through nonlinear embeddings

Yuanjun Gao, Evan Archer, Liam Paninski, John P. Cunningham
Department of Statistics and Grossman Center
Columbia University



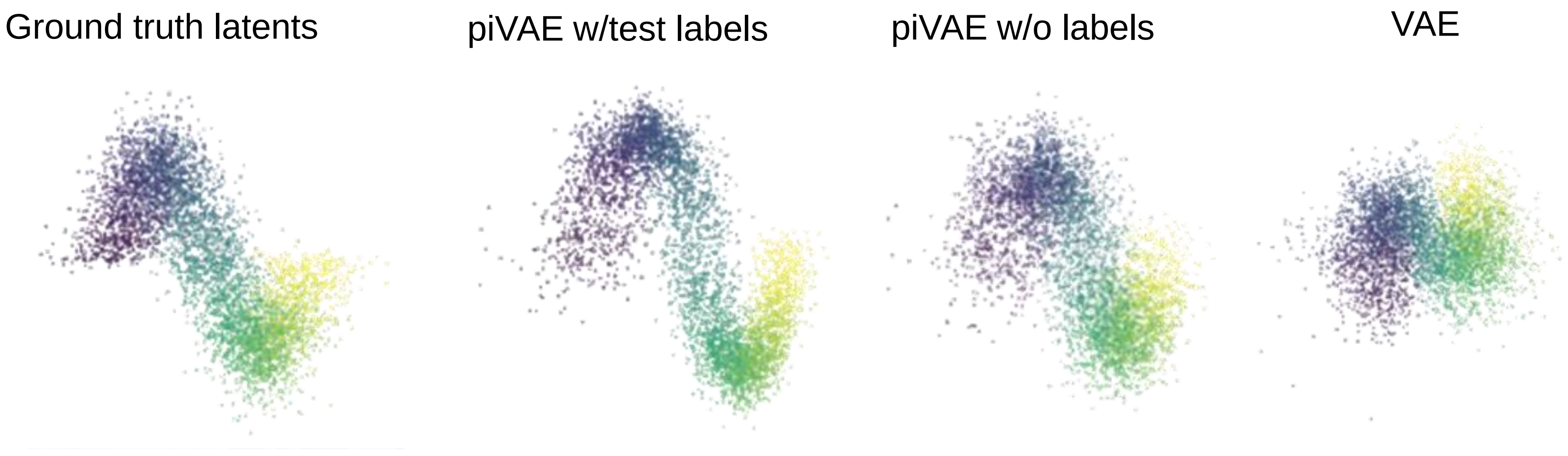
Better nonlinear embeddings by conditioning on the latent space

Label-based variational
auto encoder – piVAE: Zhou & Wei NeurIPS 2020



piVAE outperformed:

- LFADS (Pandarinath 2019)
- dPCA (Kobak 2016)
- PCA
- **pfLDS (Gao 2016)**



Limitations:

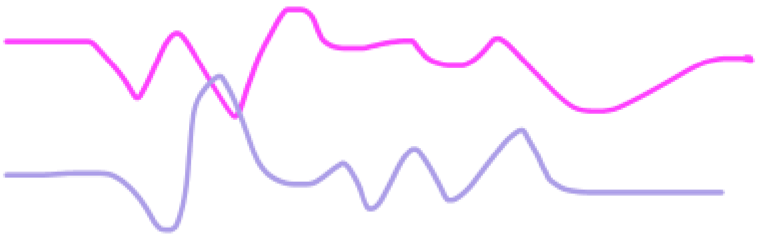
- **No identifiably guarantee***
- Restrictive assumptions on the generative model (Poisson)

*Not consistent across animals, cannot combine across datasets/can't separate the latents

Identifiable non-linear ICA: the problem setting....

Latent (hidden) underlying brain-state factors (z)

Observable neural data



. . .

$\mathbf{Z} = (z_{1,}\, z_{2,}\, \ldots.z_n)$

$\mathbf{X} = (x_{1,}\, x_{2,}\, \ldots.x_n)$

- Non-linear ICA attempts to find non-linear components such that they correspond to a well-defined generative model (Hyvärinen et al., 2001; Jutten et al., 2010).

- The aim is to recover the inverse function **g** as well as the independent components **z** based on observations of x alone.

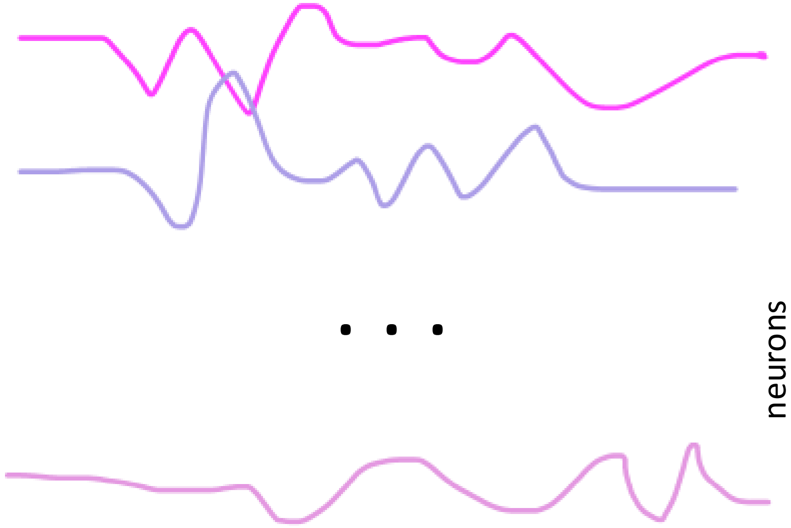
$$\mathbf{z} \xrightarrow{\text{Mixing function}} \mathbf{x} = \mathbf{g}(z)$$

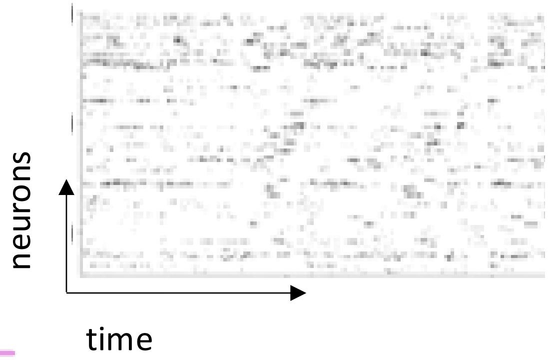
- Identifiable: “the sources can be separated”- Aapo Hyvärinen

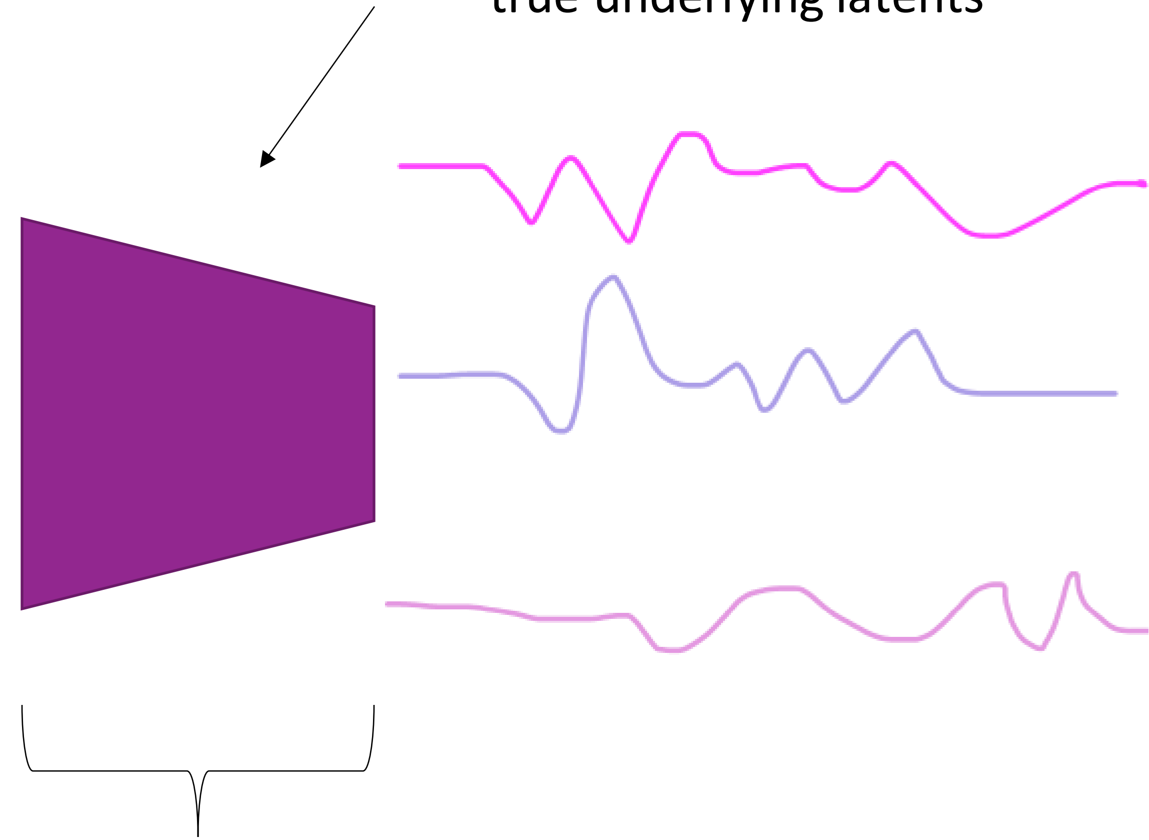
Latent (hidden) underlying brain-state factors (z)

Observable neural data

What we want: to <u>learn</u> the true underlying latents





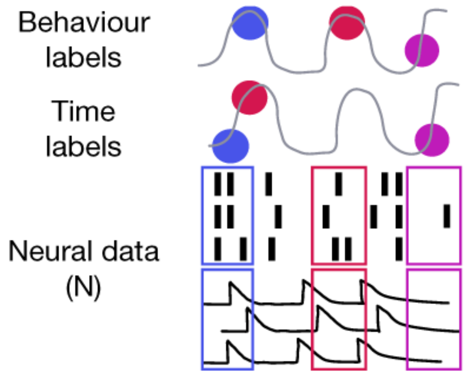


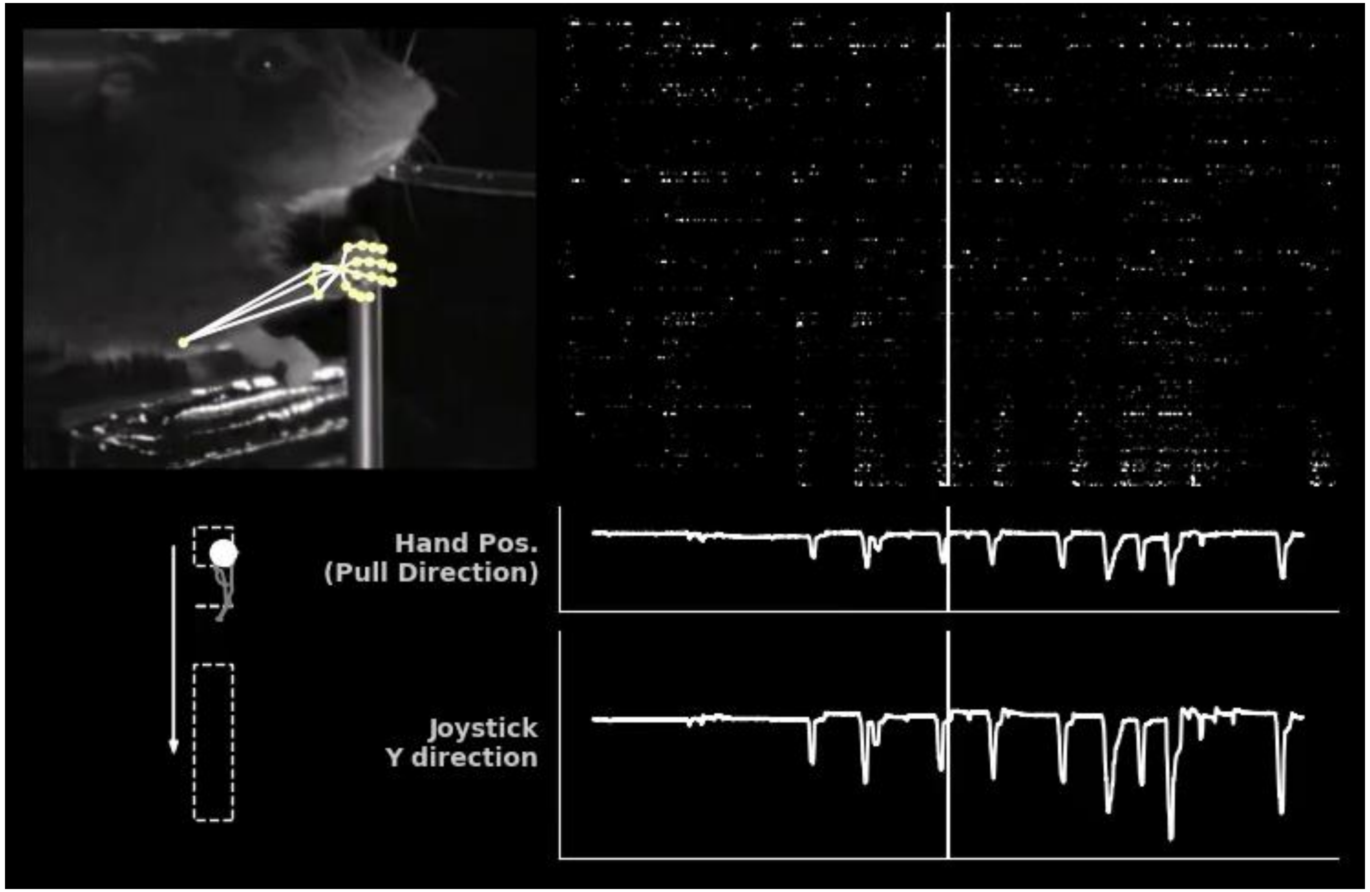
$\mathbf{z}$ → $x = \mathbf{g}(z)$

Mixing function

This can also be thought of as dimensionality reduction

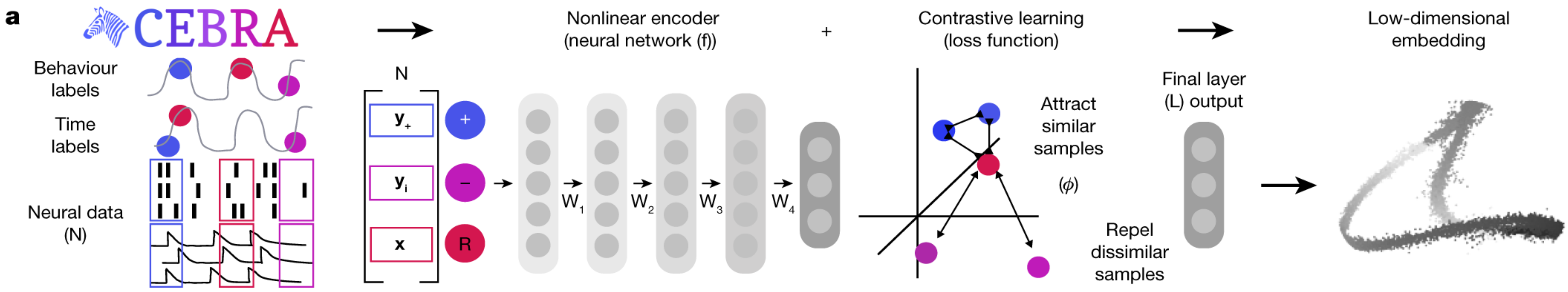
We can leverage paired behavior + neural data to learn latent embeddings





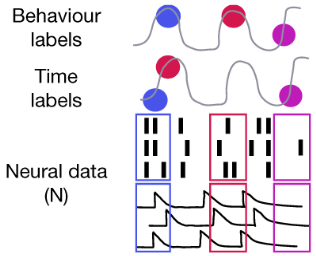
3D kinematics, reward, reward history, learning rate, trial timing, sensory inputs ….

CEBRA : an algorithm for joint modeling of auxiliary & times series data

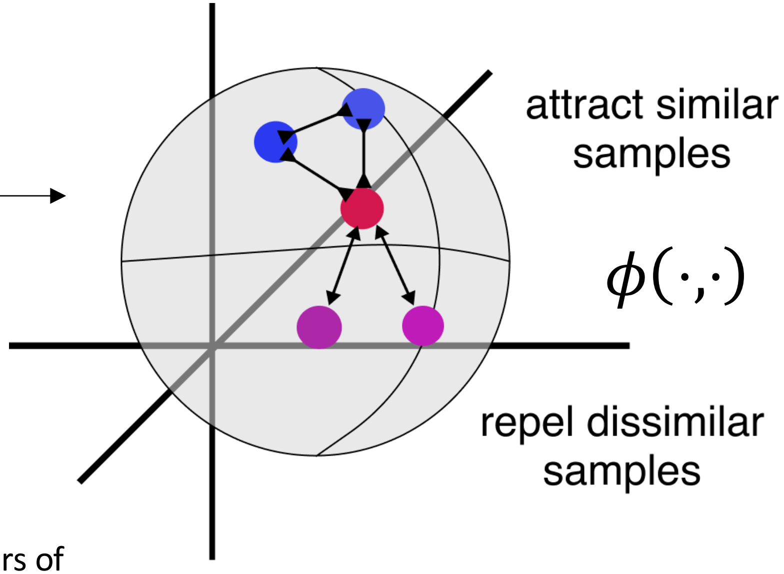


Schneider, Lee, Mathis Nature 2023

CEBRA : an algorithm for joint modeling of auxiliary & times series data



p pairs of neural data



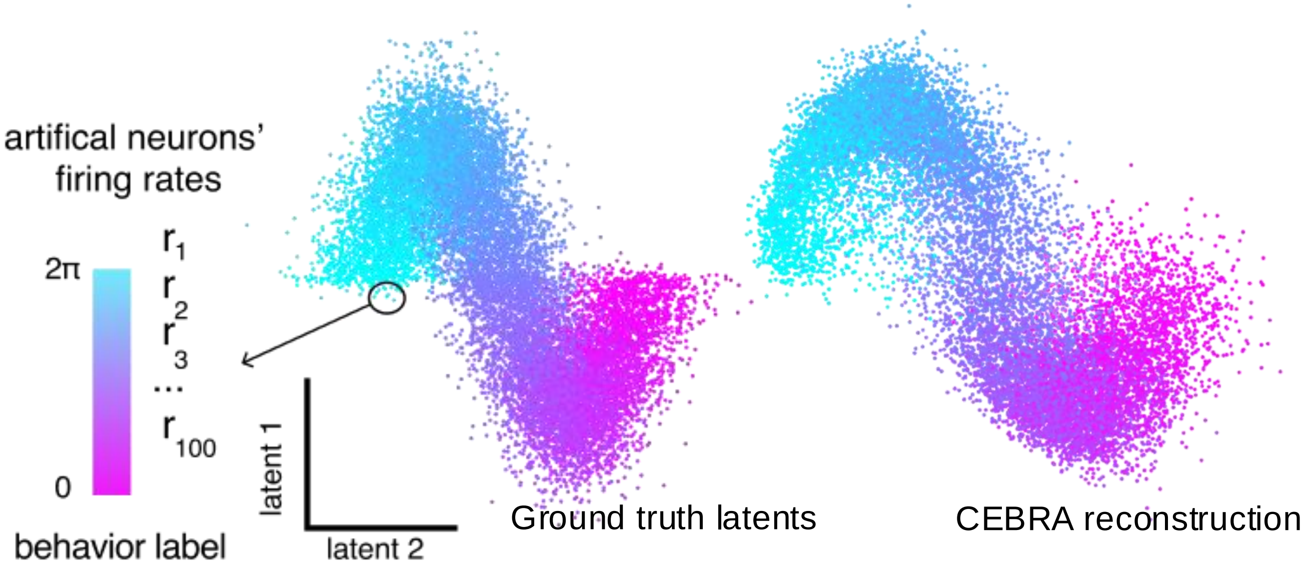
$$\mathop{\mathbb{E}}_{\substack{\mathbf{x}\sim p(\mathbf{x})\ \mathbf{y}_+\sim p(\mathbf{y}|\,\mathbf{x}) \\ \mathbf{y}_1,\ldots,\mathbf{y}_n\sim q(\mathbf{y}|\mathbf{x})}} \left[-\underbrace{\psi(\mathbf{x}, \mathbf{y}_+)} + \log \sum_{i=1}^{n} e^{\psi\underbrace{(\mathbf{x}, \mathbf{y}_i)}} \right]$$

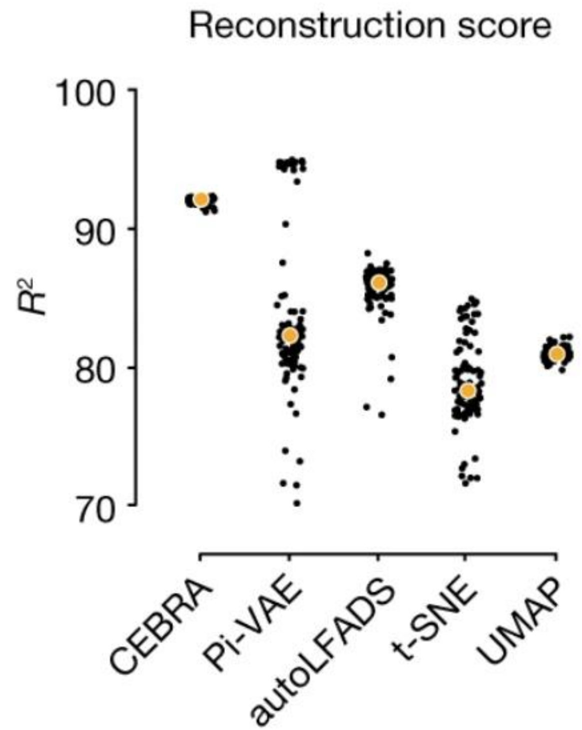
q pairs of neural data

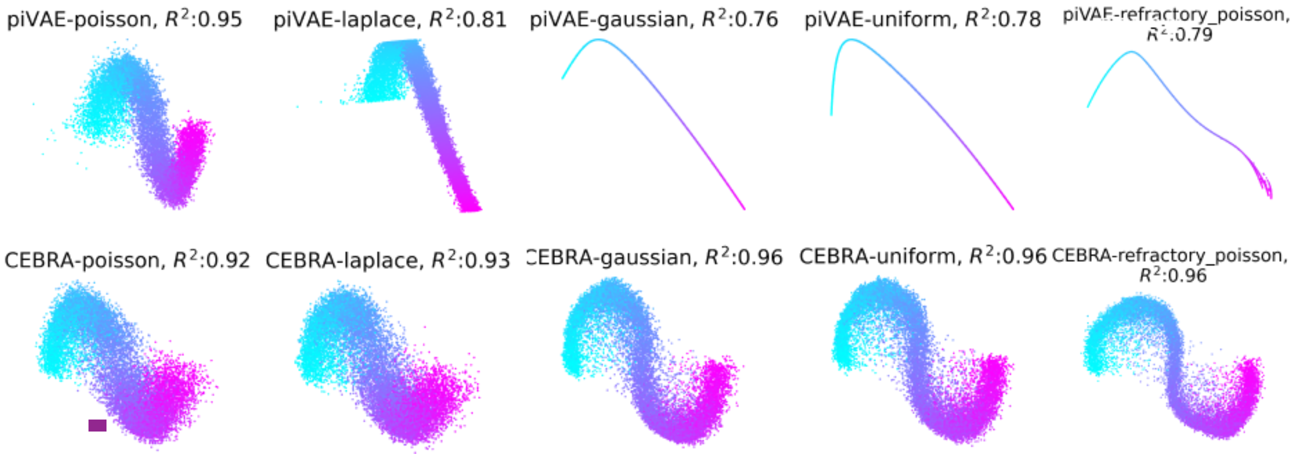
p = Positive distribution

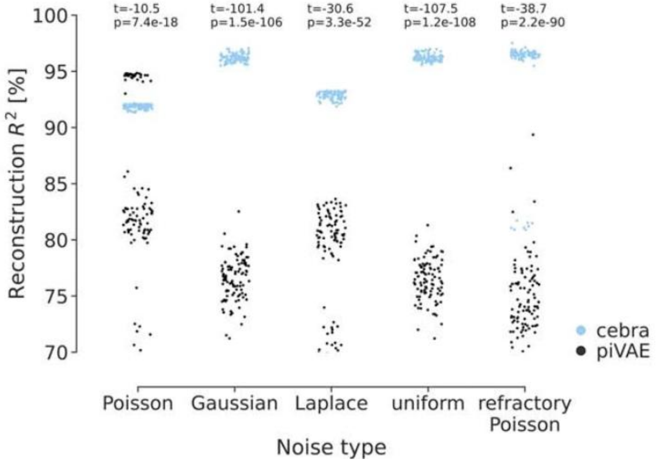
q = Negative distribution

CEBRA outperforms UMAP, tSNE, and generative models in an equivalent benchmarking setting



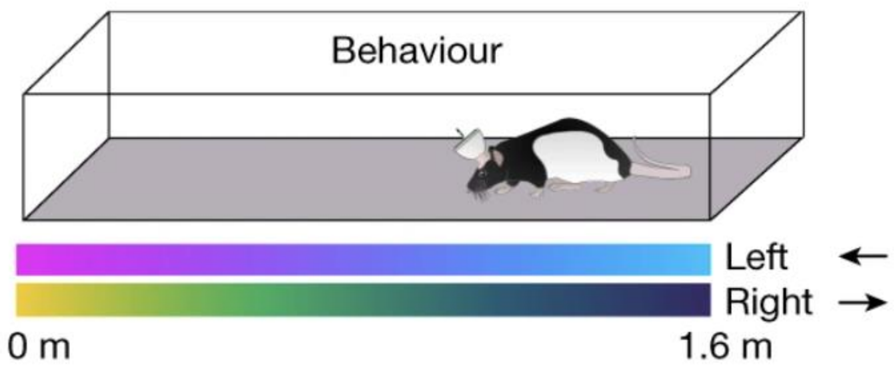


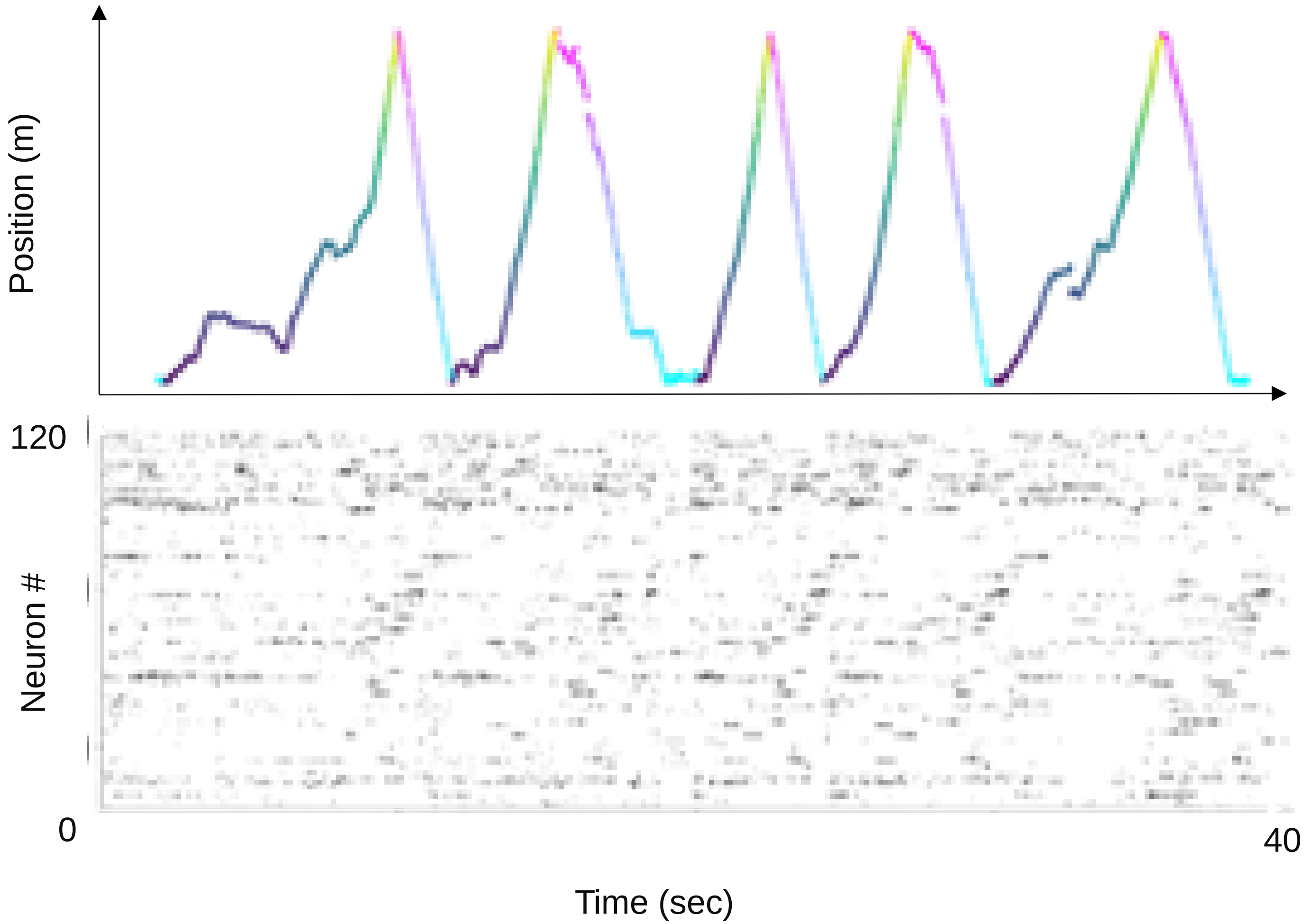




Highly consistent
& high performance

High performance and consistent latent embeddings: test-case, the CA1 map of space....

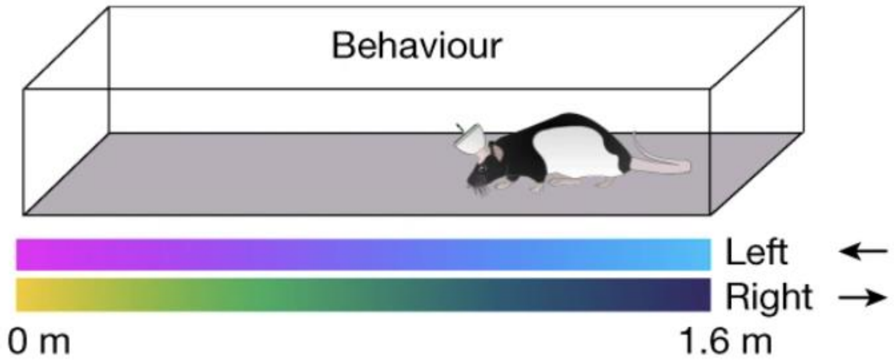


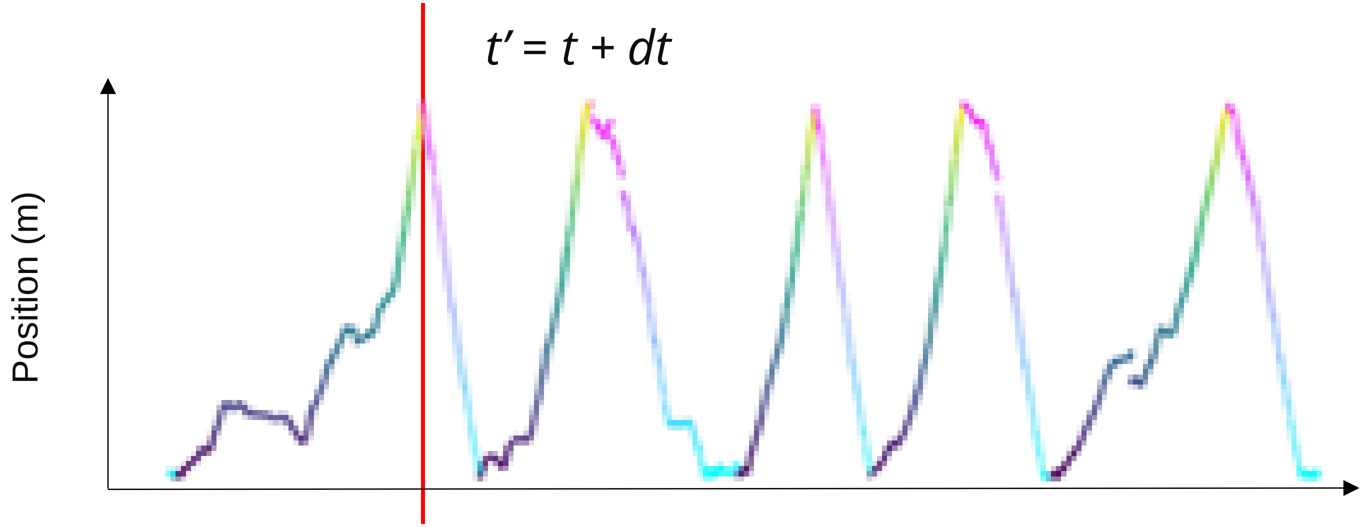


- Data: Grosmark et al. 2016 Science

How to choose the positive/negative distributions to obtain embeddings?

- **Use time: nearby samples form a positive pair**
  - Assumption: Interesting factors vary across time

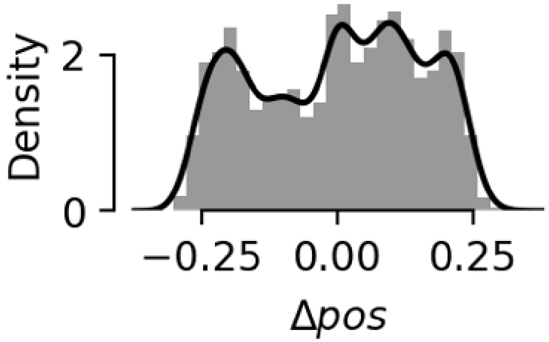


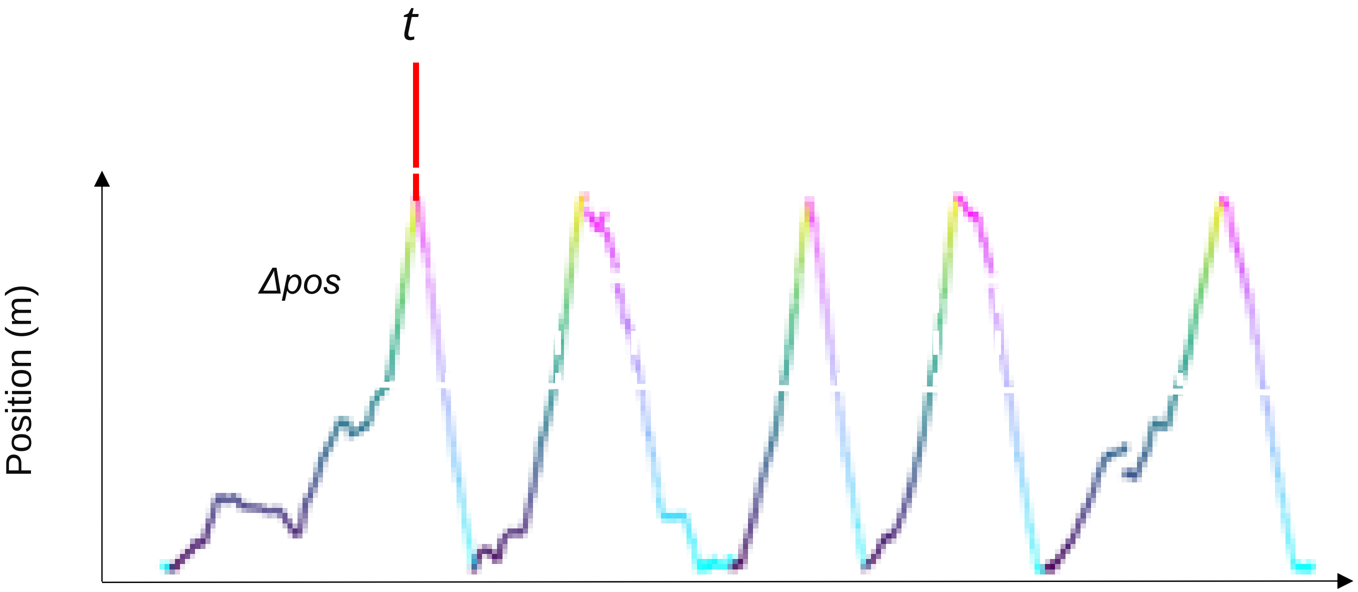


Self-supervised sampling based on time difference

How to choose the positive/negative distributions to obtain embeddings?

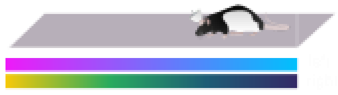
- **Build an empirical distribution of time differences**
  → *Behavior conditional distribution is independent of current location*

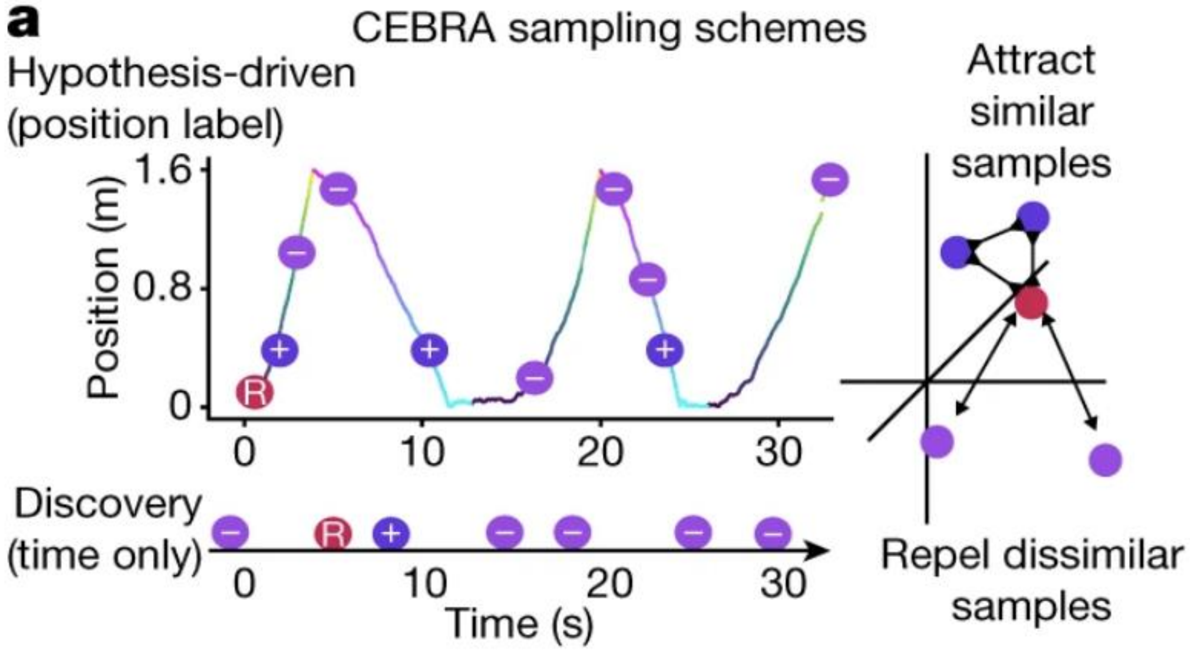




sampling based on empirical behavior differences

Discover-driven and hypothesis-based embeddings



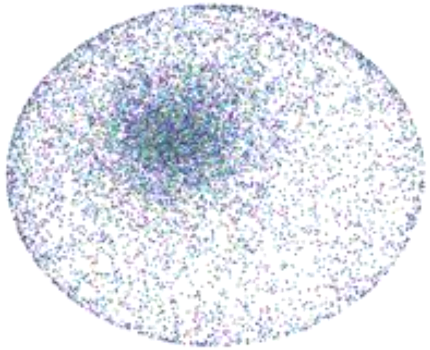


Erroneous: keep structure of neural data, behavior-label shuffled

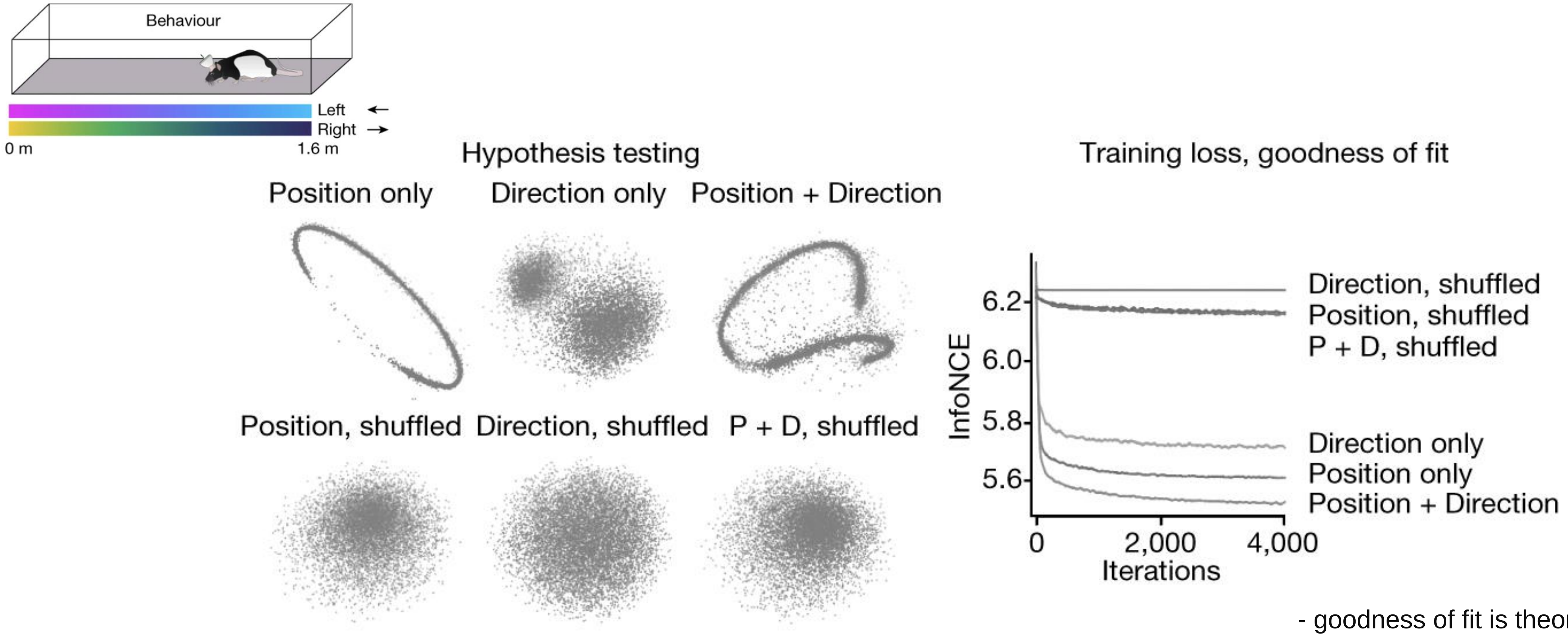
Discovery-driven SSL

Hypothesis-guided SSL

Shuffle Control

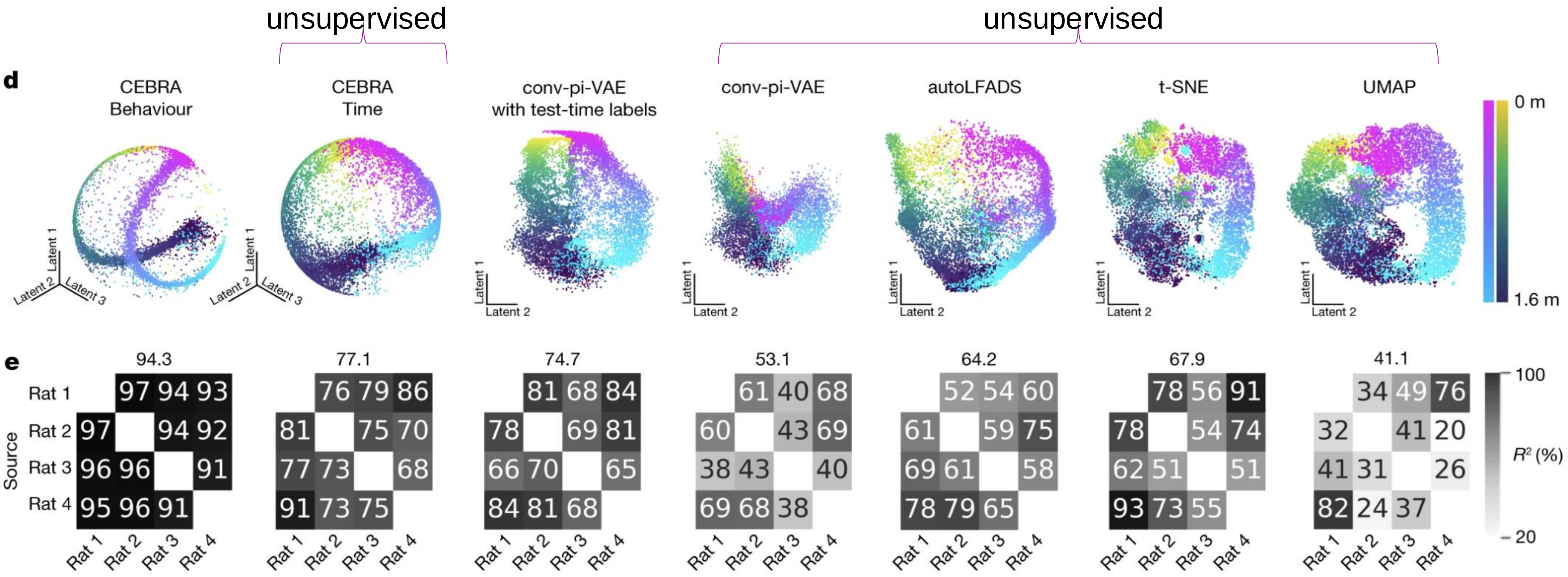


Hypothesis-guided latent embeddings

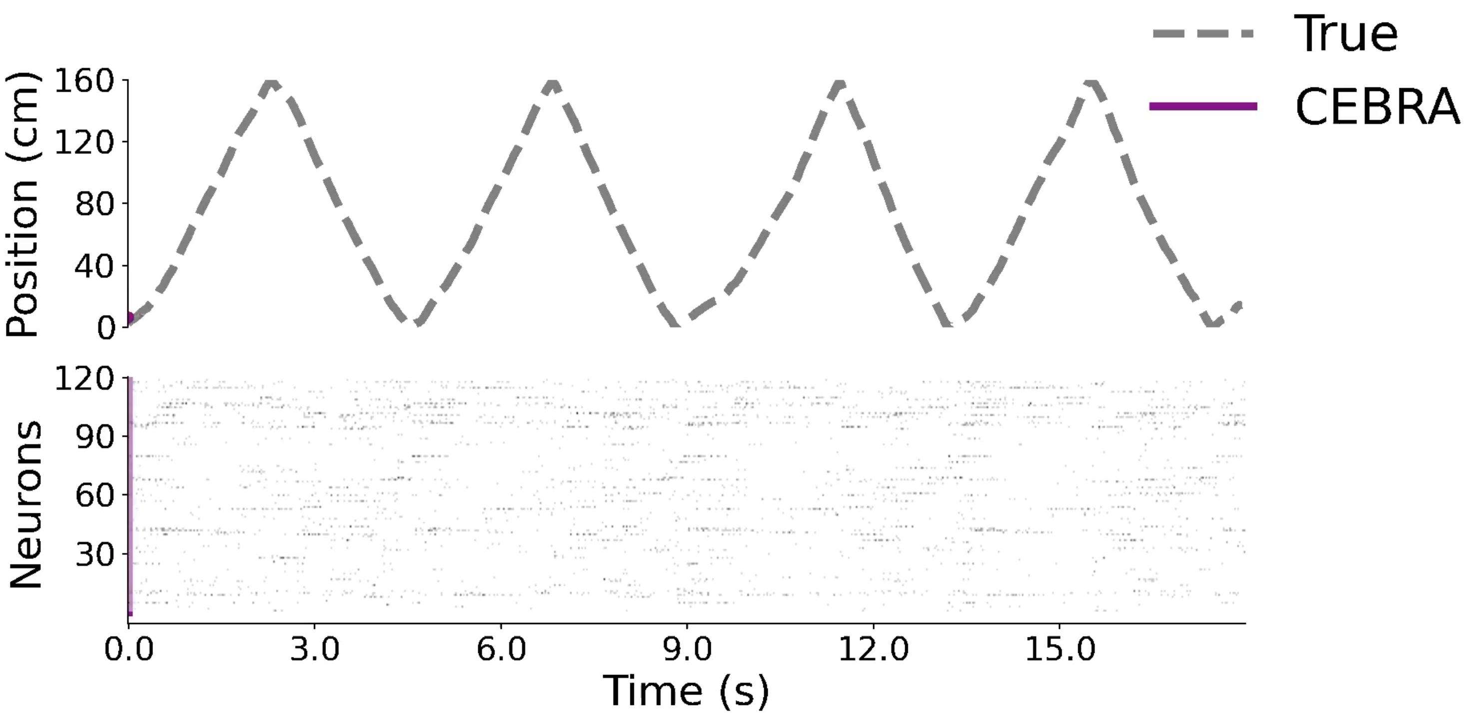


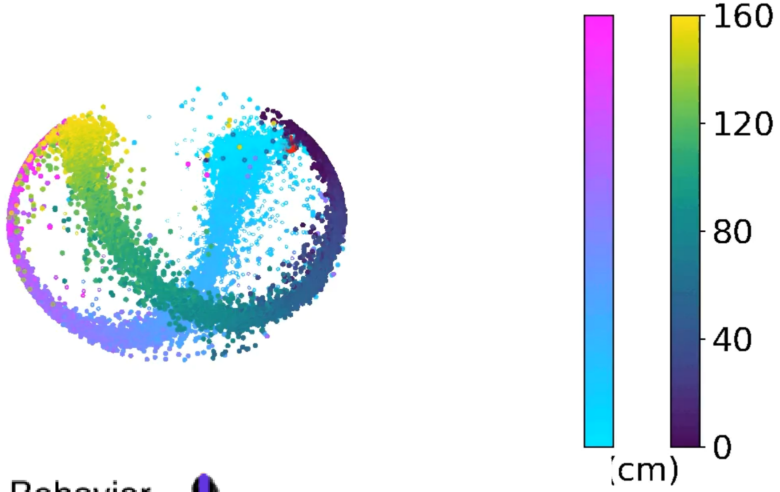
- goodness of fit is theoretically defined

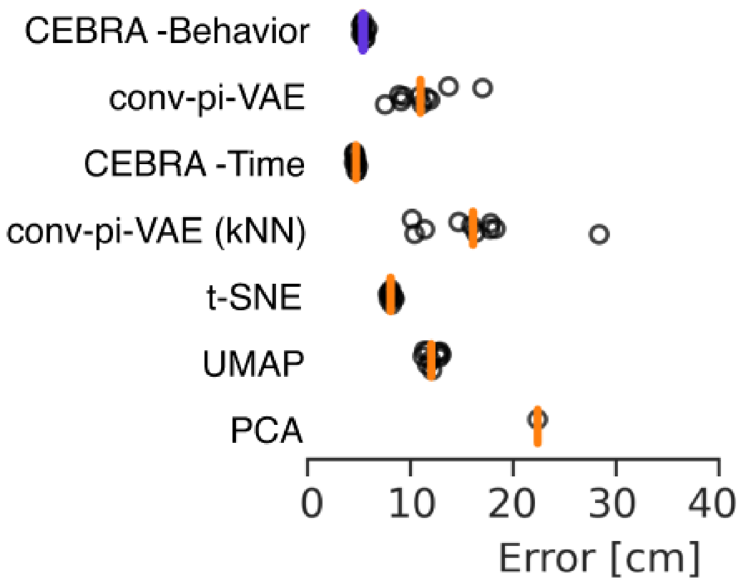
CEBRA yields highly consistent embeddings across different animals performing the same task



High performance decoding of space from hippocampus

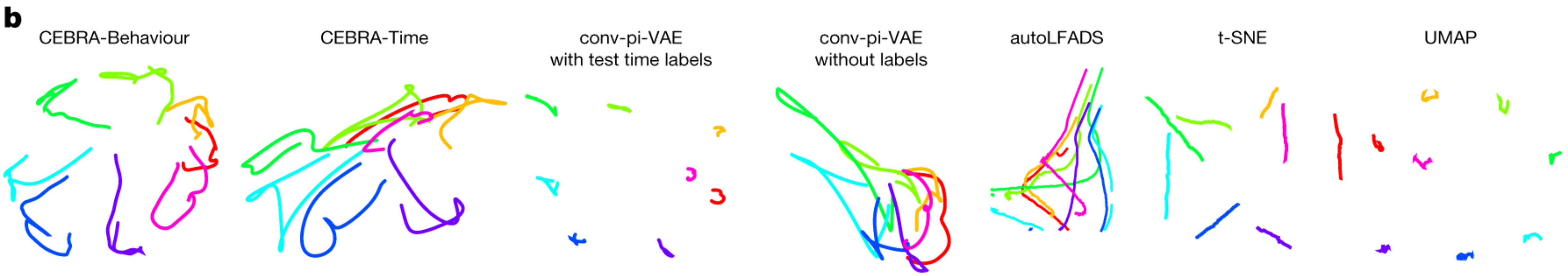






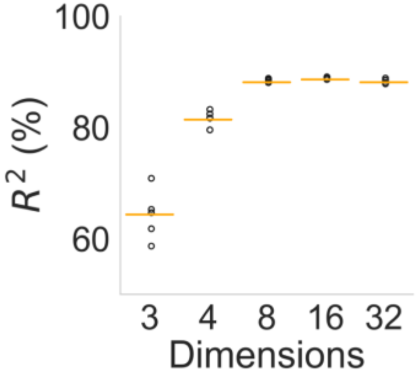
Somatosensory cortex: embeddings reveal positional and A vs. P dynamics

Data:Chowdhury et al 2020 eLife

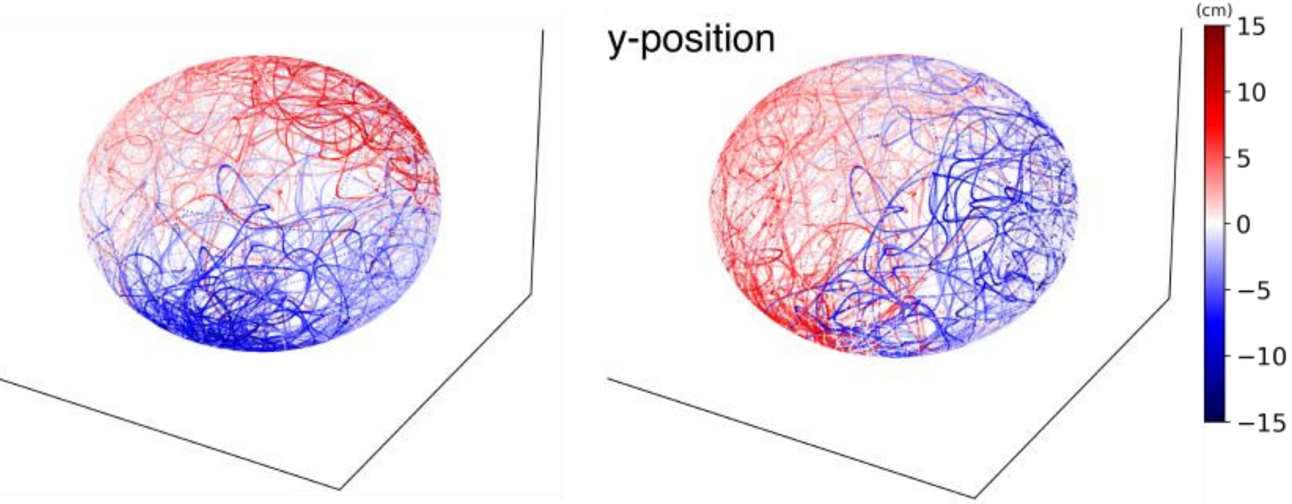


Somatosensory cortex: embeddings reveal positional and A vs. P dynamics

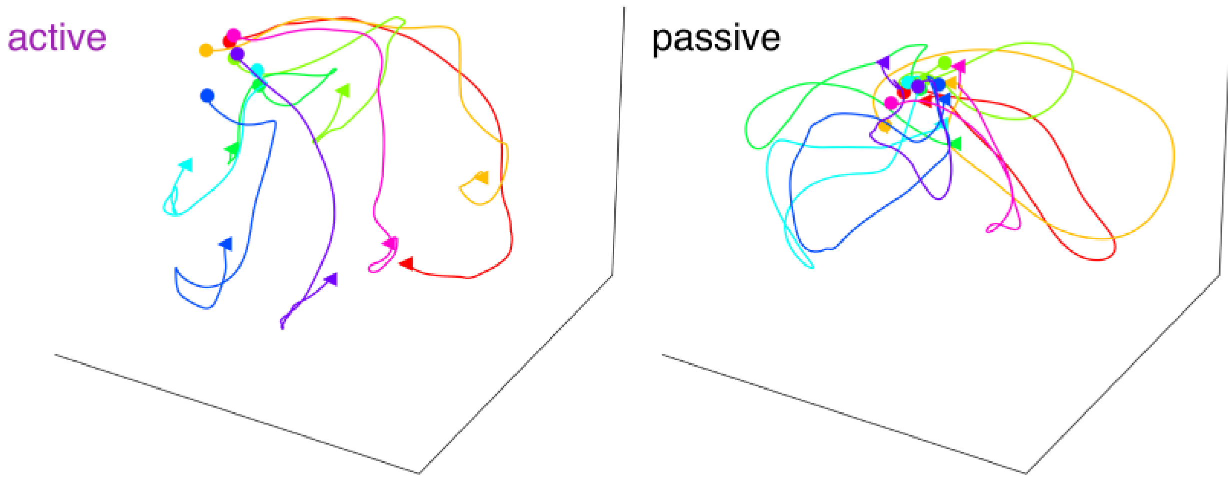
Data:Chowdhury et al 2020 eLife



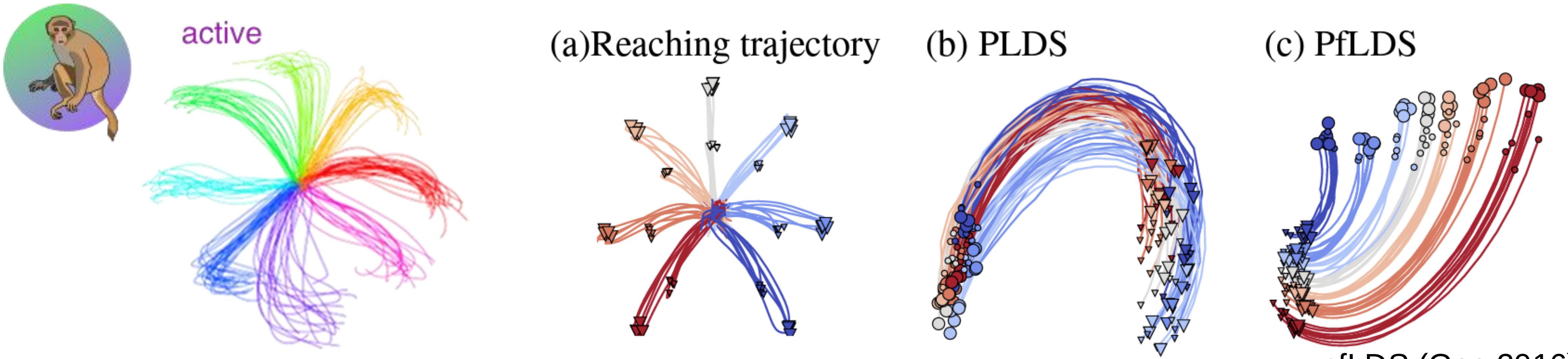
CEBRA-Time: no explicit positional information used



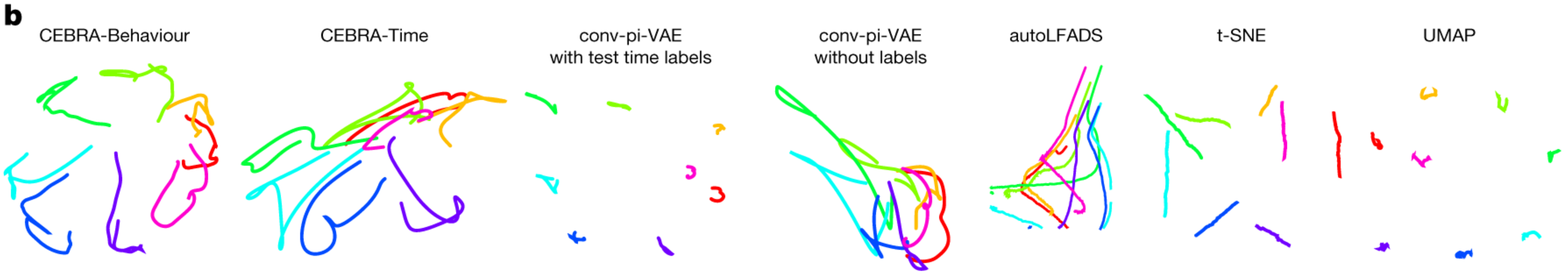
CEBRA-Behavior: labels are continuous position



CEBRA can also provide an interpretable embedding space visualization of the data



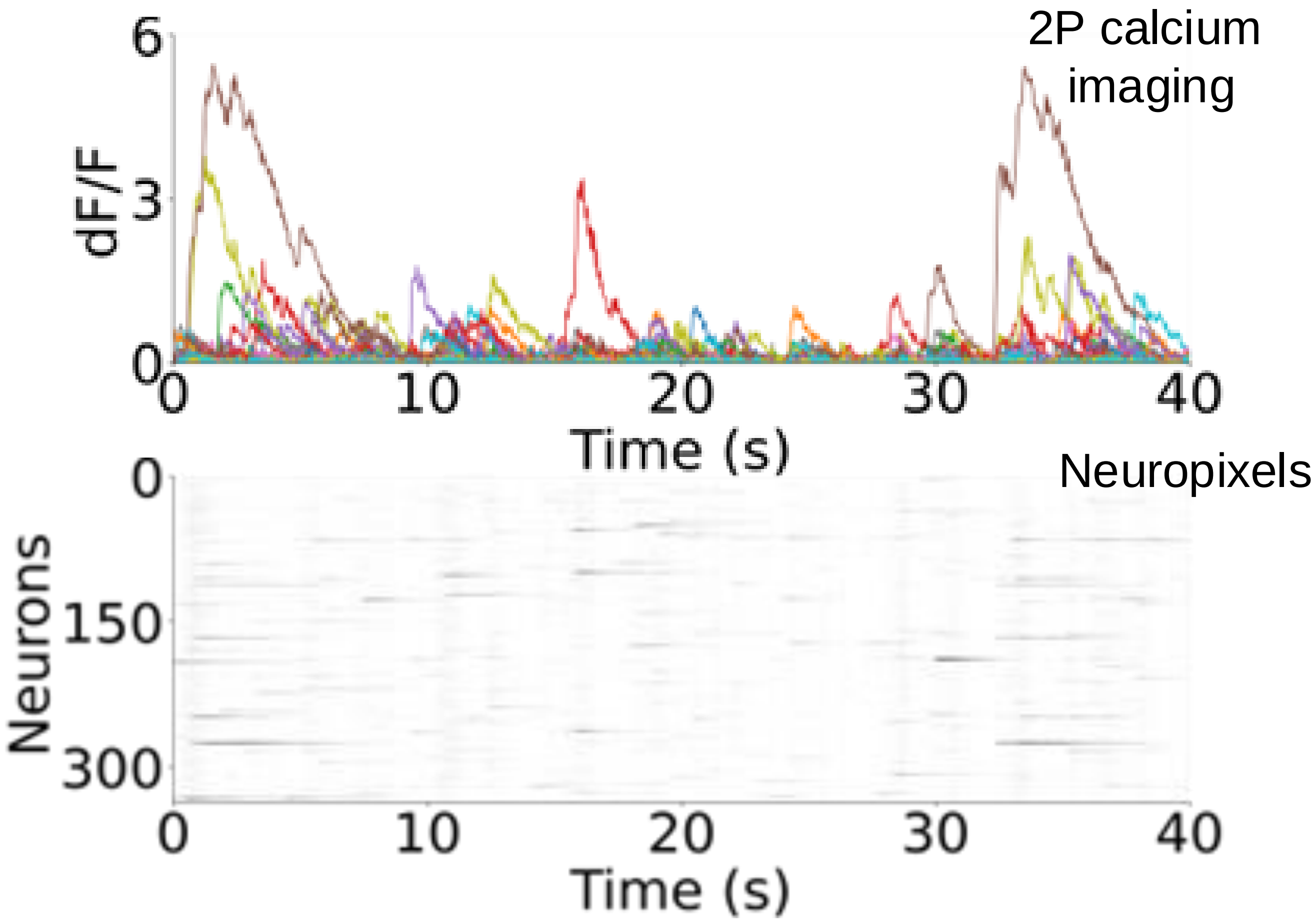
pfLDS (Gao 2016)



- CEBRA (Schneider, Lee, Mathis 2013)

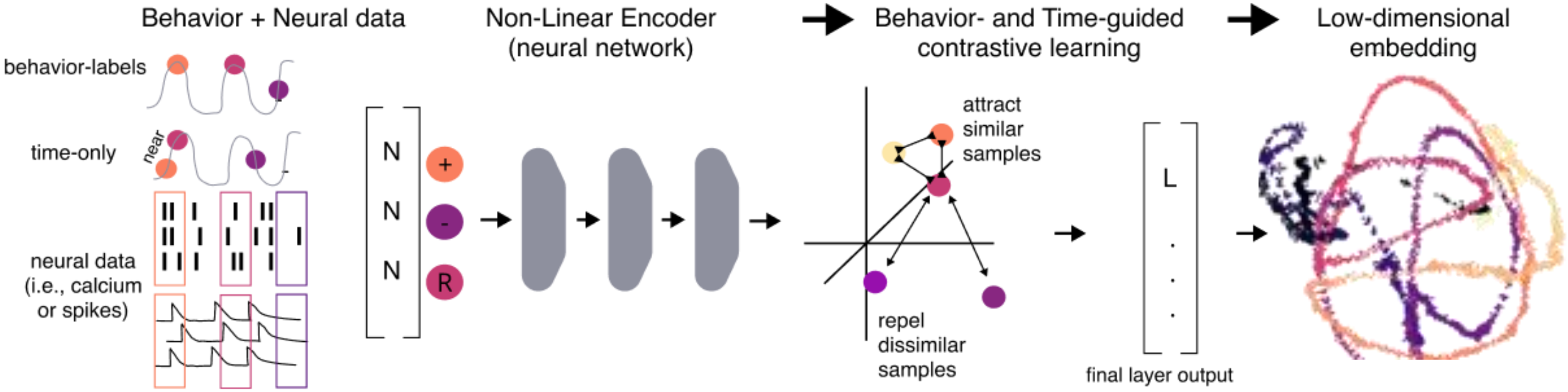
Using CEBRA to unover differences in neural dynamics across the visual system

Learnable latent embeddings from V1: mapping naturalistic video to latent spaces across animals, sessions, modalities ...



Data: de Vries et al. (2020)
Deitch, Rubin, and Ziv (2021)
- Siegle et al. (2021)

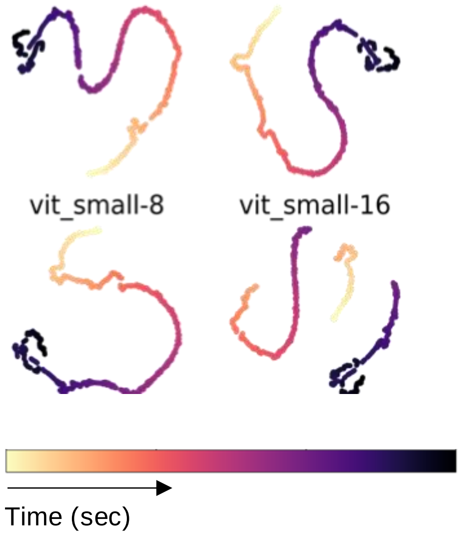
Sensory input as "behavior" labels: leveraging visual latent features



Pre-trained DINO

Caron et al 2021 arxiv

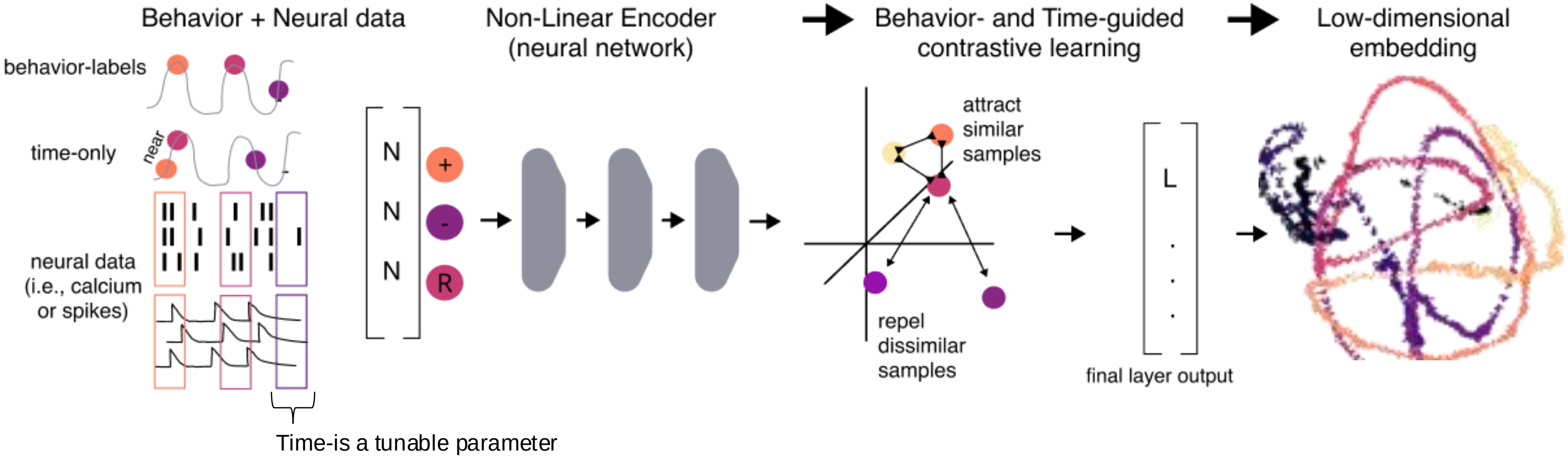
Emerging Properties in Self-Supervised Vision Transformers



- Can we find evidence for movie-feature representations in V1 using CEBRA-Behavior, CEBRA-Time?
- Can we produce highly similar embeddings from either Neuropixels or GCaMP6 data?
- Can we leverage across data-input type?
- Interpretability: can we decode the natural movie frames from the neural latent embeddings?

EPFL

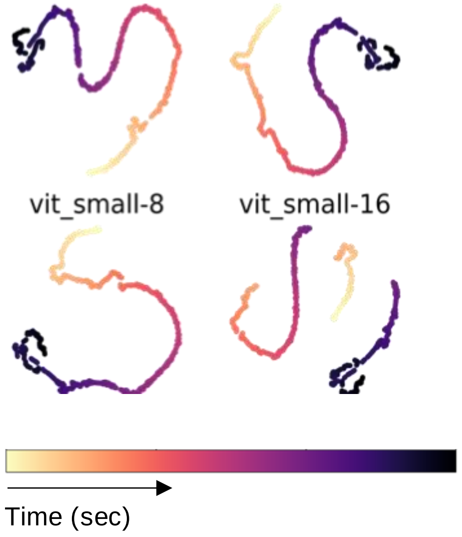
Sensory input as "behavior" labels: leveraging visual latent features



Pre-trained DINO

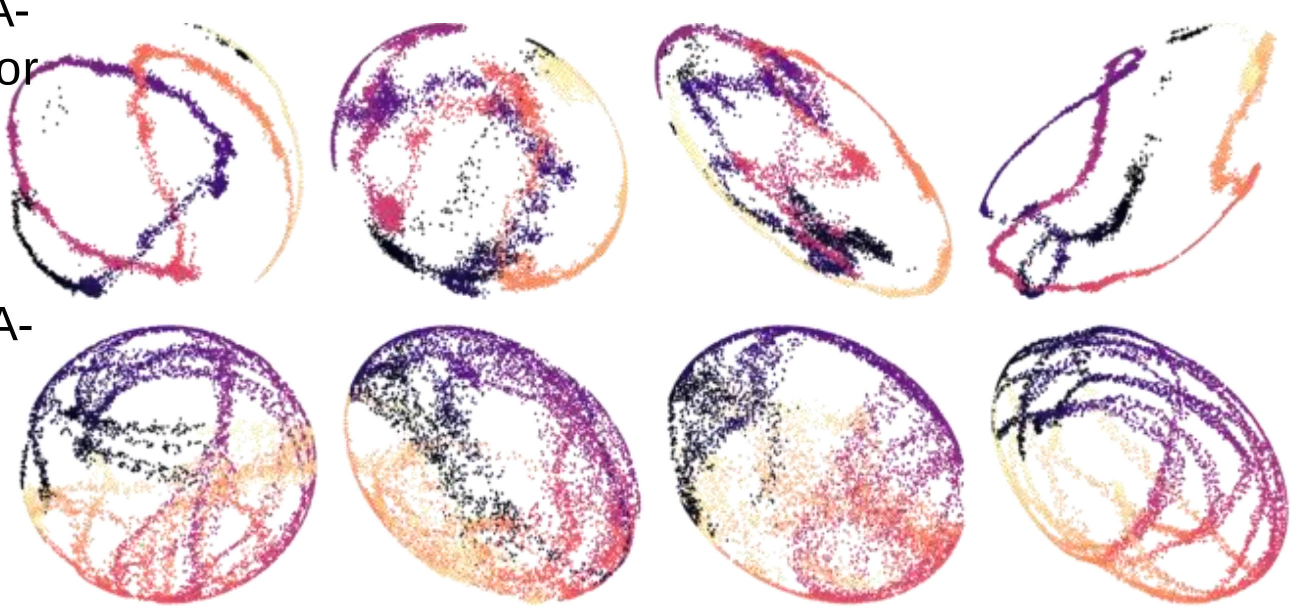
Caron et al 2021 arxiv

Emerging Properties in Self-Supervised Vision Transformers



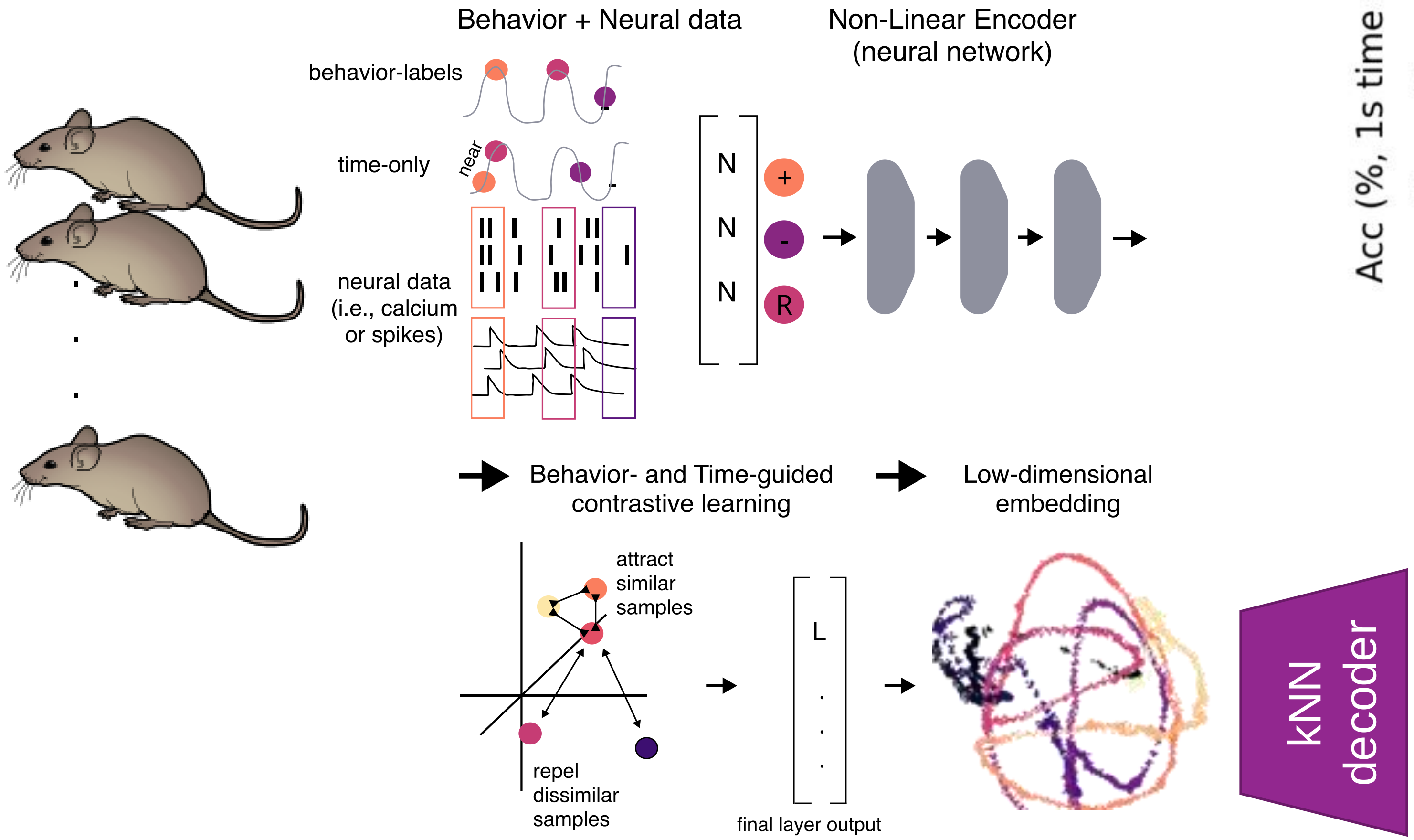
CEBRA-Behavior

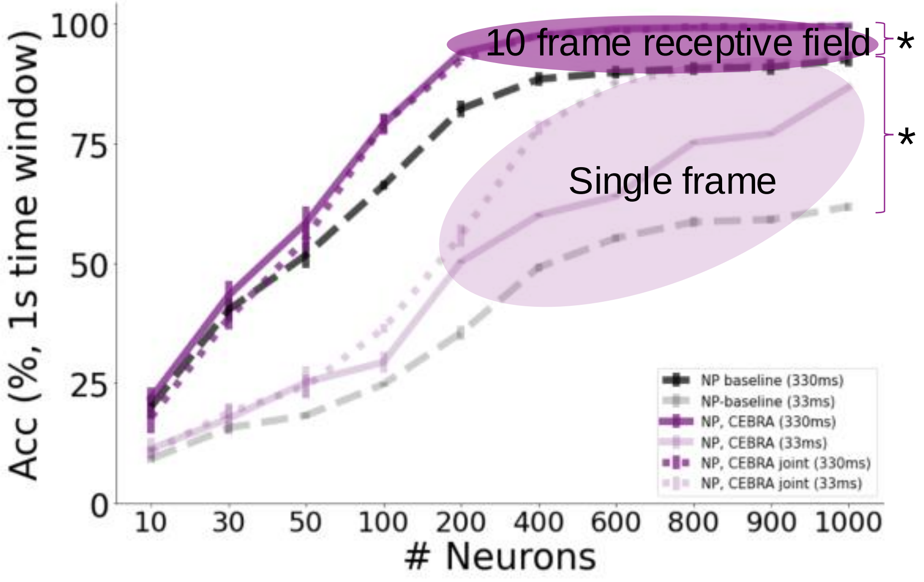
CEBRA-Time

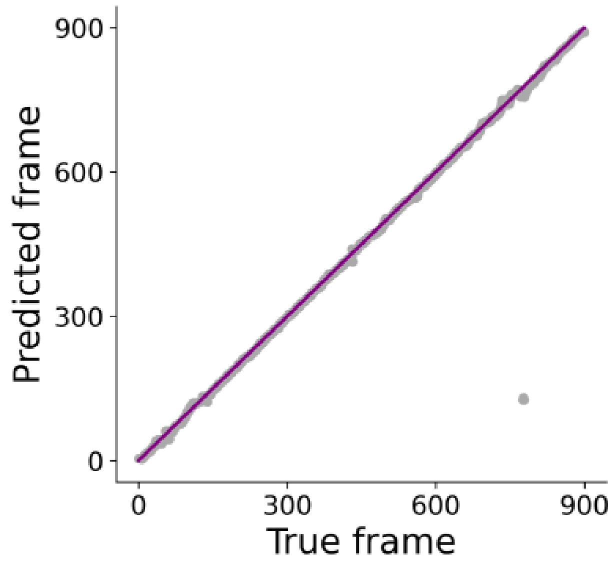


Improved decoding performance from CEBRA-based latent embedding

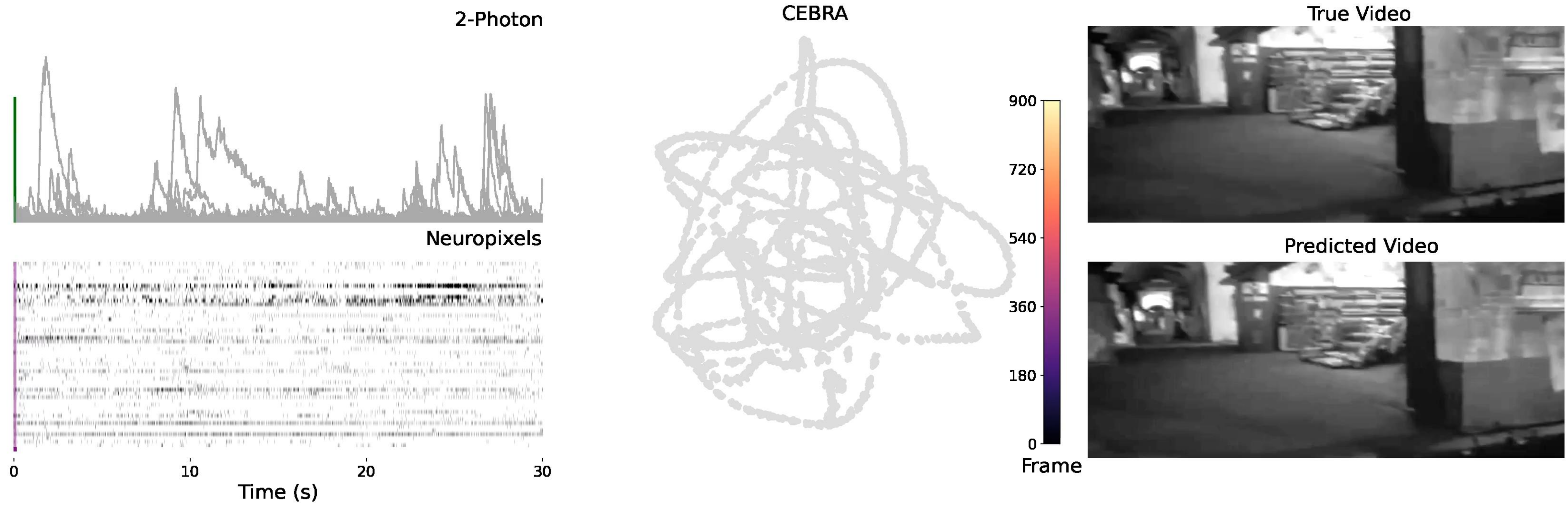
Pseudo-mouse decoding framework:







High performance decoding with CEBRA



EPFL

Summary

- **Neural encoding** and **neural decoding** are fundamental descriptions of neural (coding) processing and data analysis.
- A fundamental goal is: how much information does **K** have about **x**
- We mathematically model this as P(K|x), where the neural response of population **K** to a stimulus (or event) **x**. **K** is a vector representing the activity of *N* neurons, and each entry represents, e.g., the number of spikes in some time bin or the rate response of that particular neuron.
- **Generalized Linear Models** (GLMs) are very attractive for both individual neurons and populations, yet assume **linear** θ dynamics (careful: despite having a nonlinear parameter).
- Modern hardware advances continue to push the upper limit on the # of neurons we can record, and therefore we need new mathematical tools for understanding neural coding.
- Manifold of behavioral and neural data hypothesis comes into play…
- Two large classes of approaching modeling a system: data-driven or hypothesis (task)--driven
- Modern methods for mapping the statistical properties of neurons to a stimulus/behavior are fully-observable models and latent variable models.
- Latent variable models infer hidden (i.e., latent) variables that capture the underlying structure of the observed data through a joint probability distribution.
- VAEs and contrastive learning approach to neural analysis; contrastive learning (CEBRA) has highly attractive properties like combining across datasets and producing consistent latent embeddings.