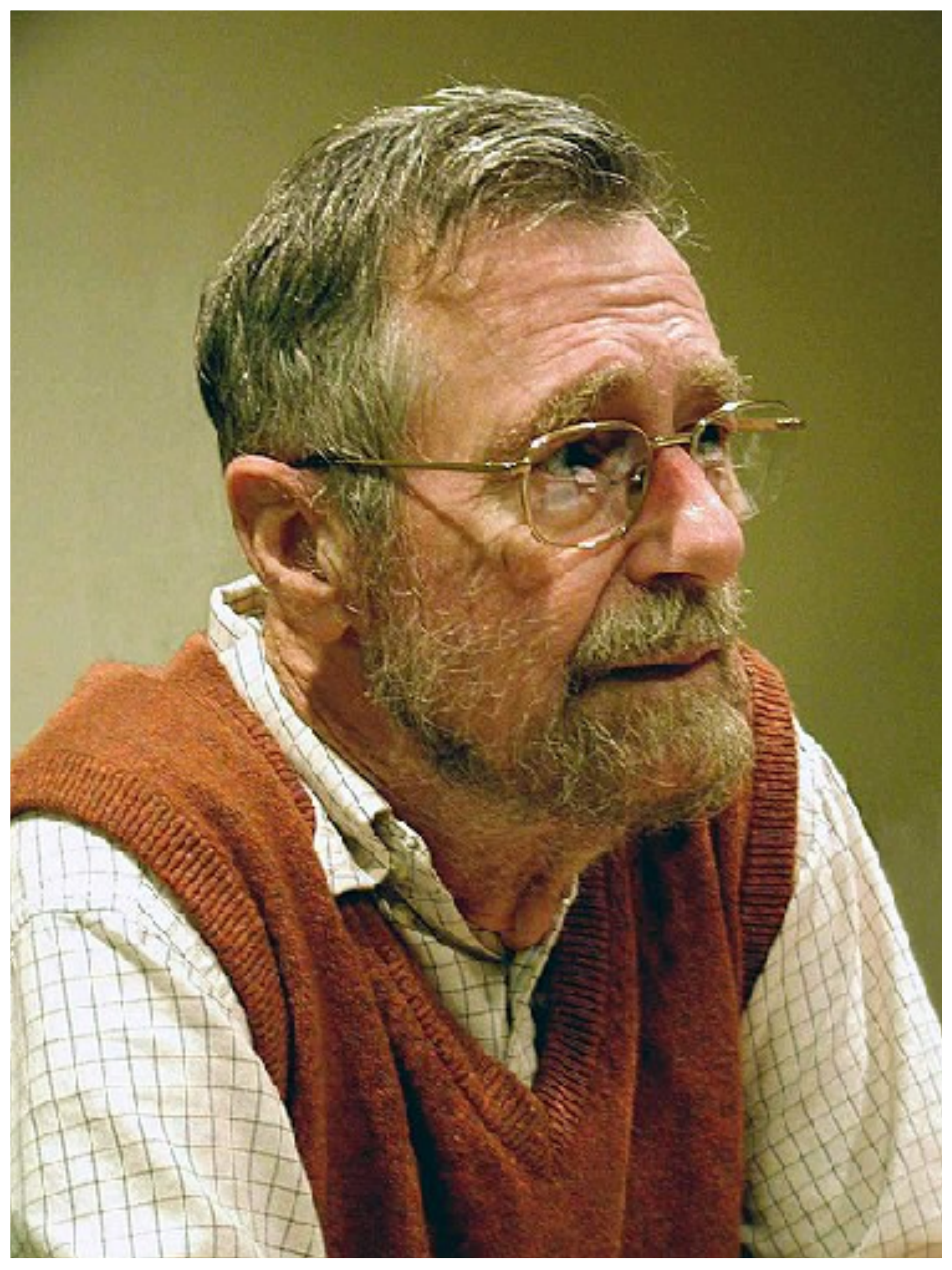
State Space Models for Natural and Artificial Intelligence

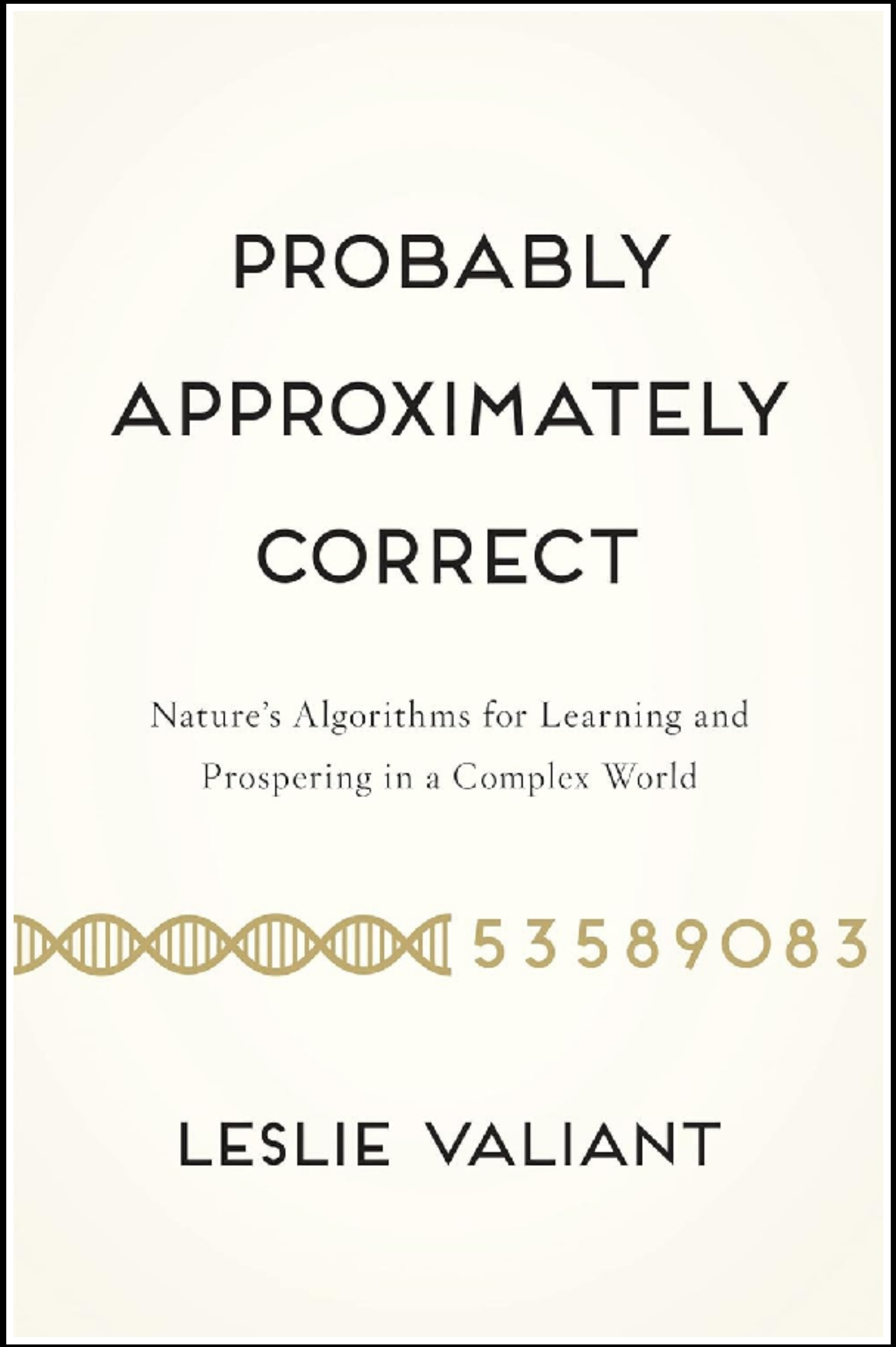
Scott Linderman
Department of Statistics
Wu Tsai Neurosciences Institute
Stanford University

Leslie Valiant

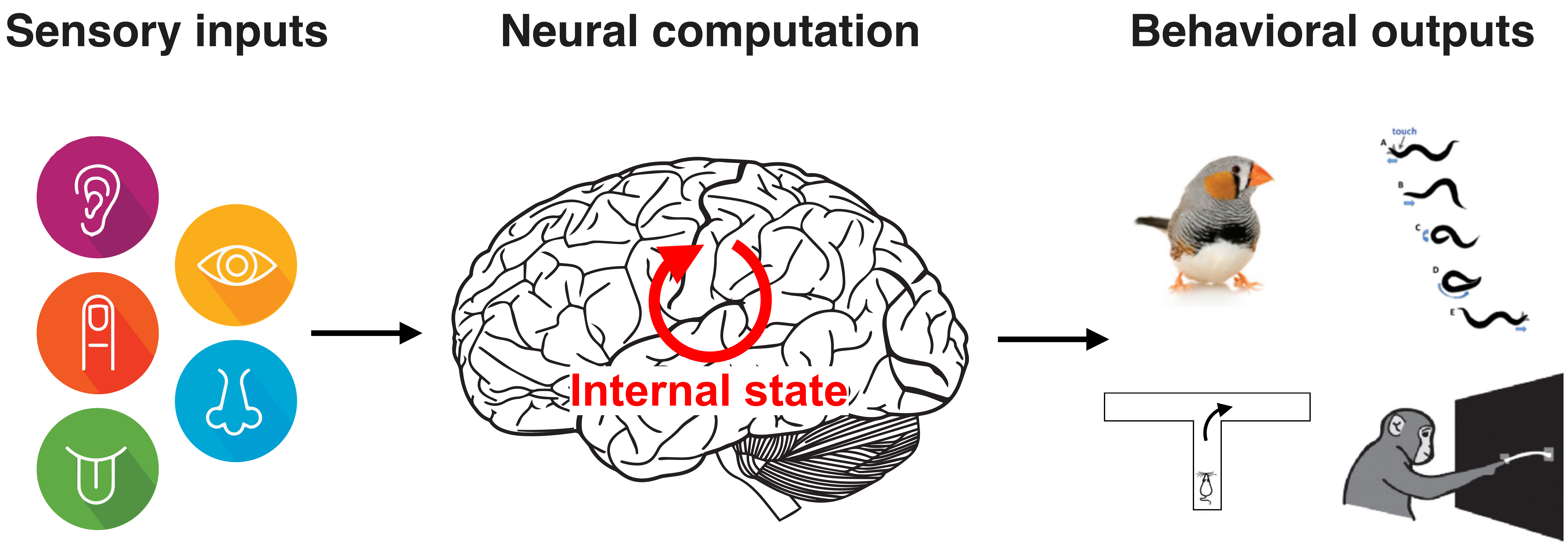
Why not study I?



Edsger Dijkstra



Intelligent systems must maintain **internal states** to produce appropriate behavioral outputs

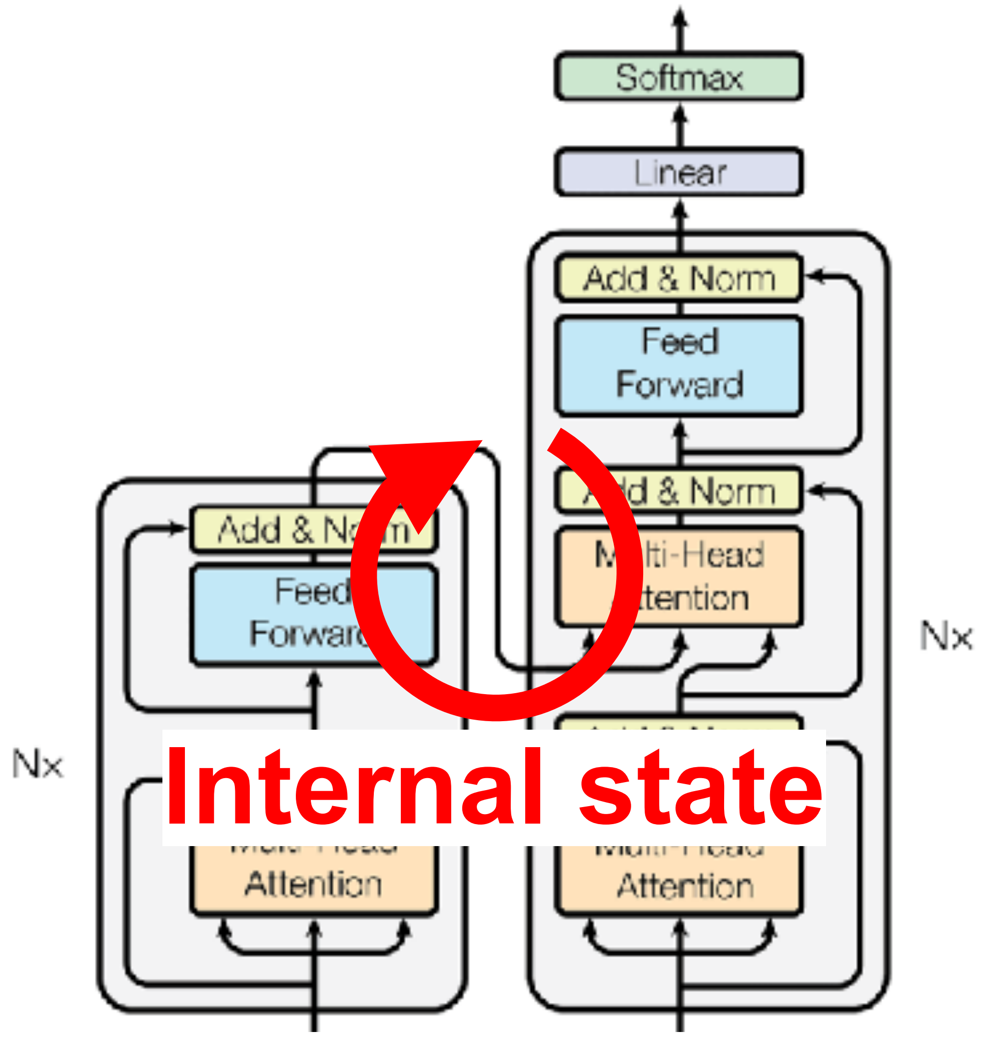


Intelligent systems must maintain **internal states** to produce appropriate behavioral outputs

Sensory inputs

Please write a Kalman filter in Python using JAX

In silico computation



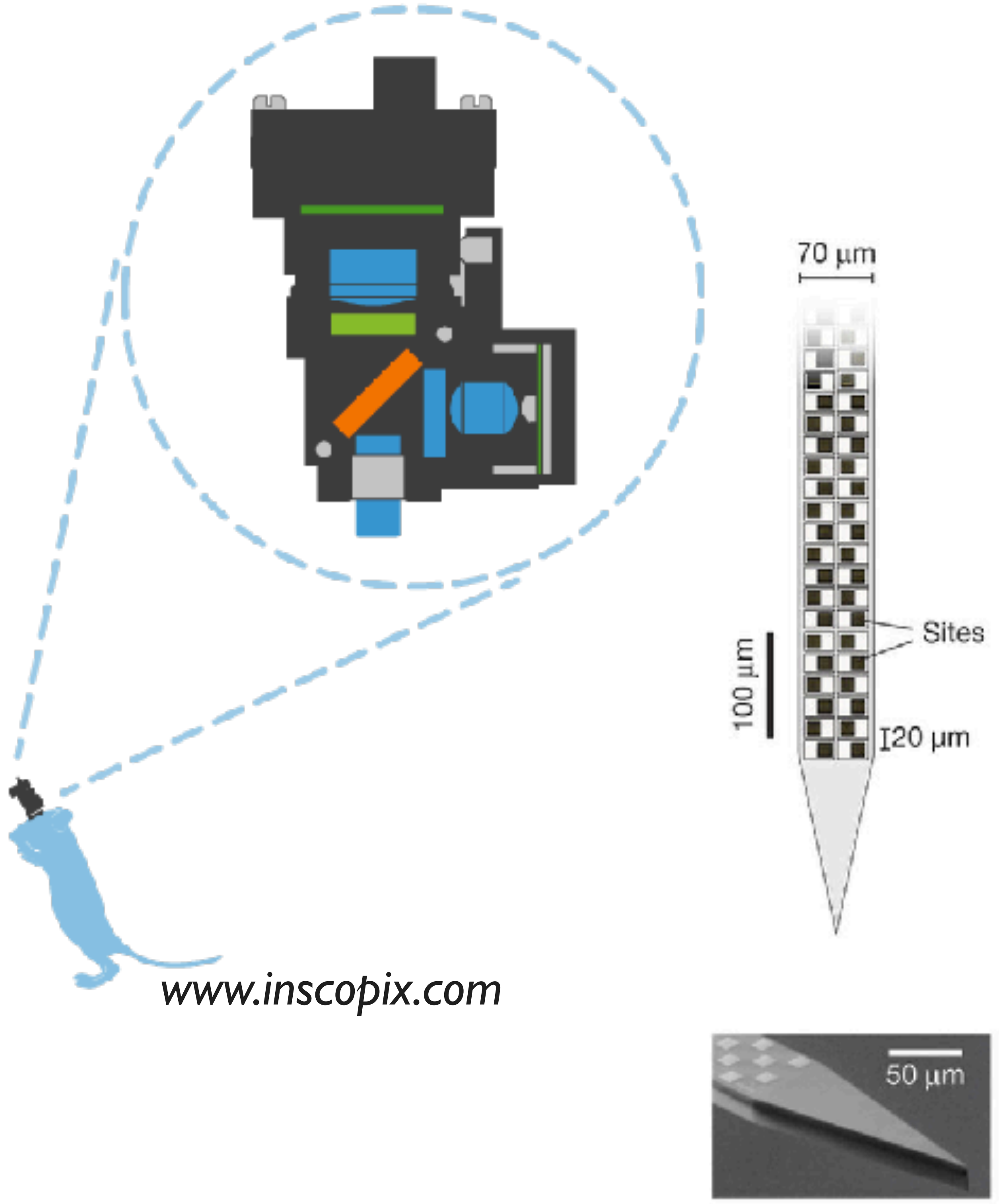
Behavioral outputs

python Copy code

```python
import jax
import jax.numpy as jnp

# Kalman Filter Step Function
def kalman_filter_step(state, params):
    # Unpack the state
    x, P = state

    # Unpack the parameters
    F, H, Q, R, z = params

    # Predict step
    x_pred = jnp.dot(F, x)
    P_pred = jnp.dot(F, jnp.dot(P, F.T)) + Q

    # Update step
    y = z - jnp.dot(H, x_pred)
    S = jnp.dot(H, jnp.dot(P_pred, H.T)) + R
    K = jnp.dot(P_pred, jnp.dot(H.T, jnp.linalg.inv(S)))
```

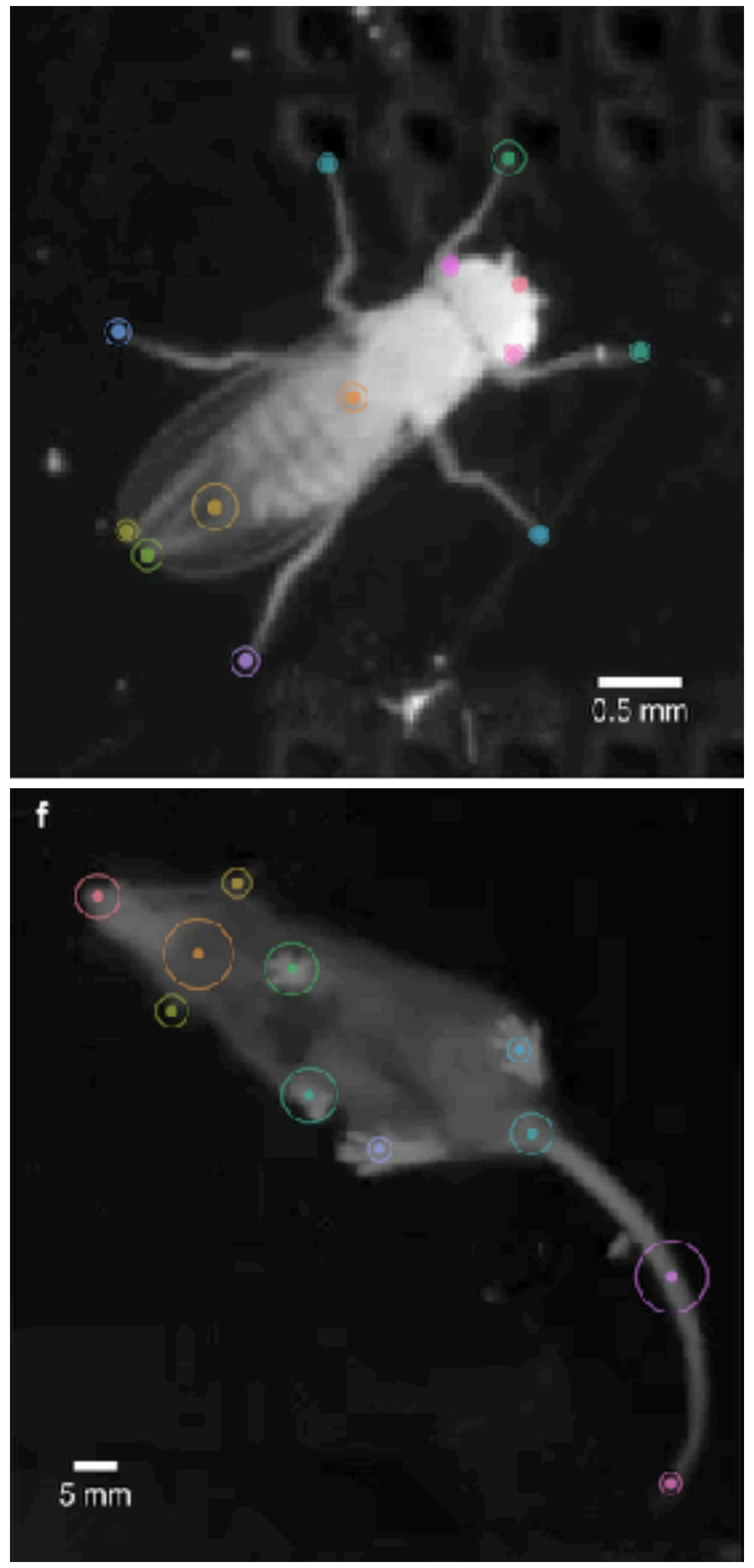
Exciting advances in neural and behavioral recording technologies

Large-scale neuronal measurements in freely behaving animals



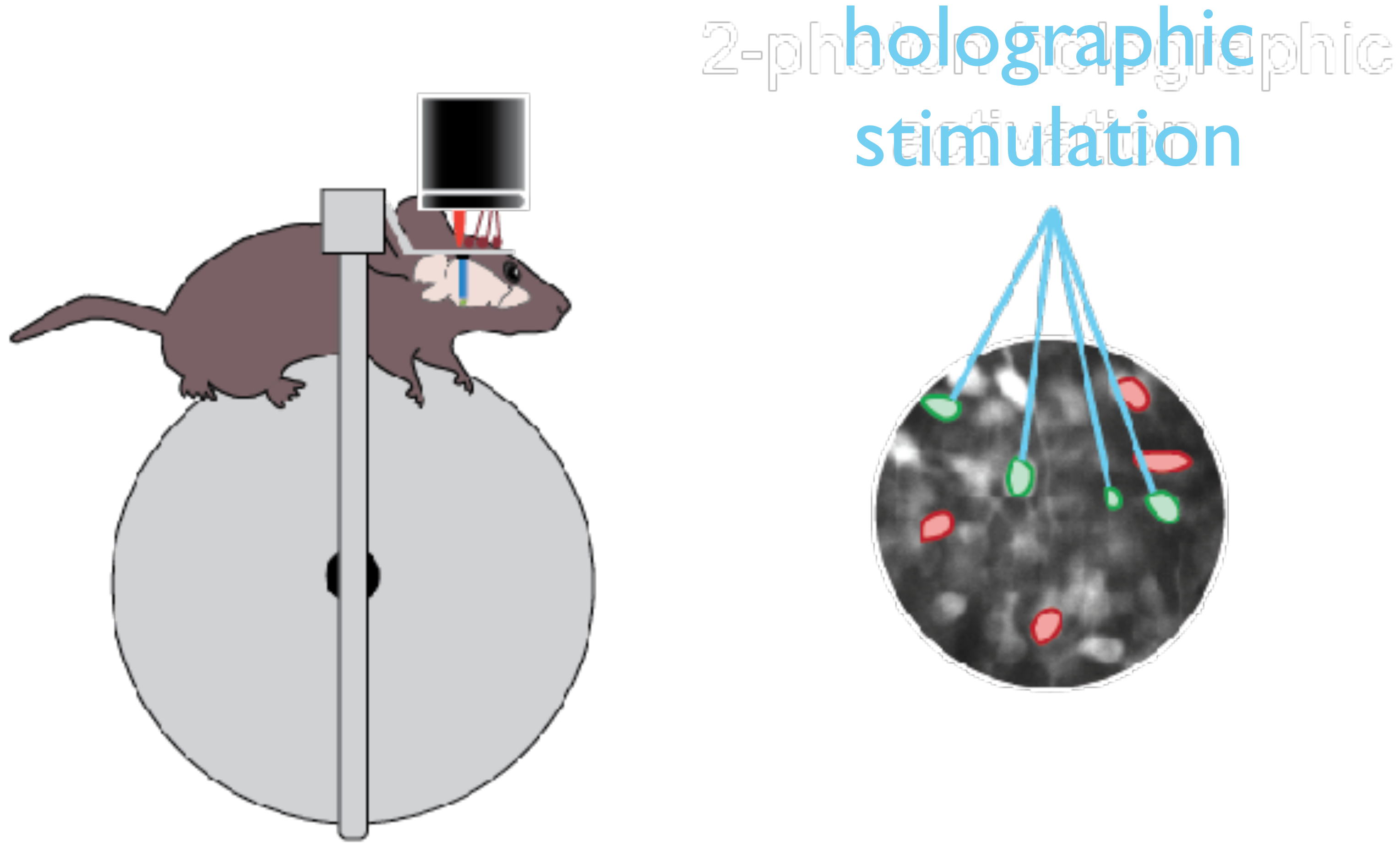
Jun, Steinmetz et al (2017).

Automated tracking and computer vision for quantifying natural behavior



Pereira et al (2022)

Causal perturbations with closed-loop, patterned optogenetic stimulation



Exciting advances in machine learning models and computational hardware

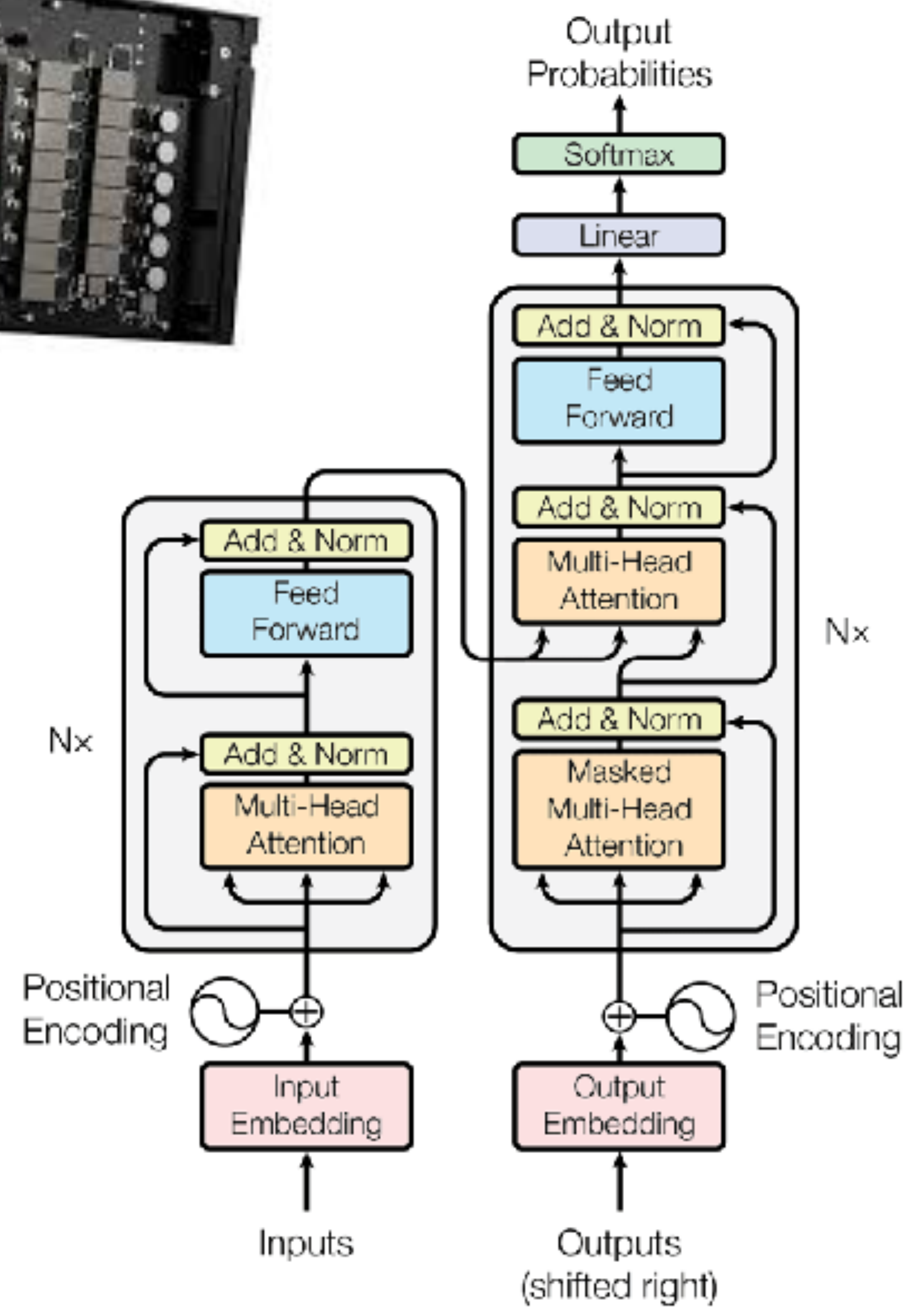
Nobel Physics Prize Awarded for Pioneering A.I. Research by 2 Scientists

With work on machine learning that uses artificial neural networks, John J. Hopfield and Geoffrey E. Hinton “showed a

Nobel Prize in Chemistry Goes to 3 Scientists for Predicting and Creating Proteins

The Nobel, awarded to David Baker of the University of Washington and Demis Hassabis and John M. Jumper of Google

nVidia H100 GPU



Transformer architecture

Outline

1. *Natural Intelligence: Internal states and attractor dynamics in the hypothalamus*
2. *Artificial Intelligence: Deep state space models for sequence-to-sequence modeling*

Outline

1. *Natural Intelligence: Internal states and attractor dynamics in the hypothalamus*

    - *<u>Question</u>: How does the brain represent and maintain internal states?*

    - *<u>Method</u>: Recurrent switching linear dynamical systems (rSLDS)*

    - *<u>Results</u>: Intrinsic line attractor dynamics in the hypothalamus encode an aggressive internal state*

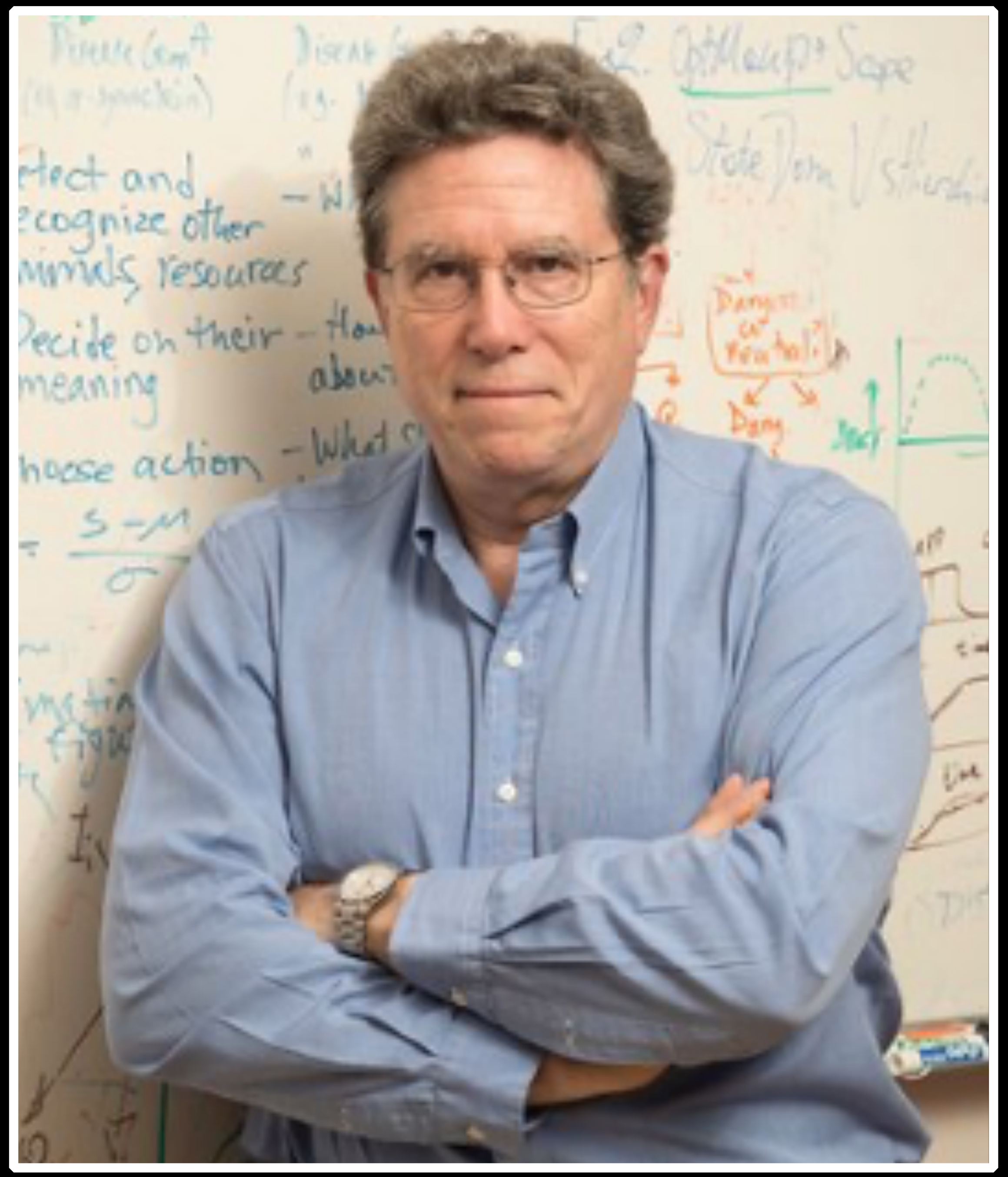
    - *<u>Extension</u>: Smoothly interpolating between states in an rSLDS*

2. *Artificial Intelligence: Deep state space models for sequence-to-sequence modeling*

Collaborators

Adi Nair

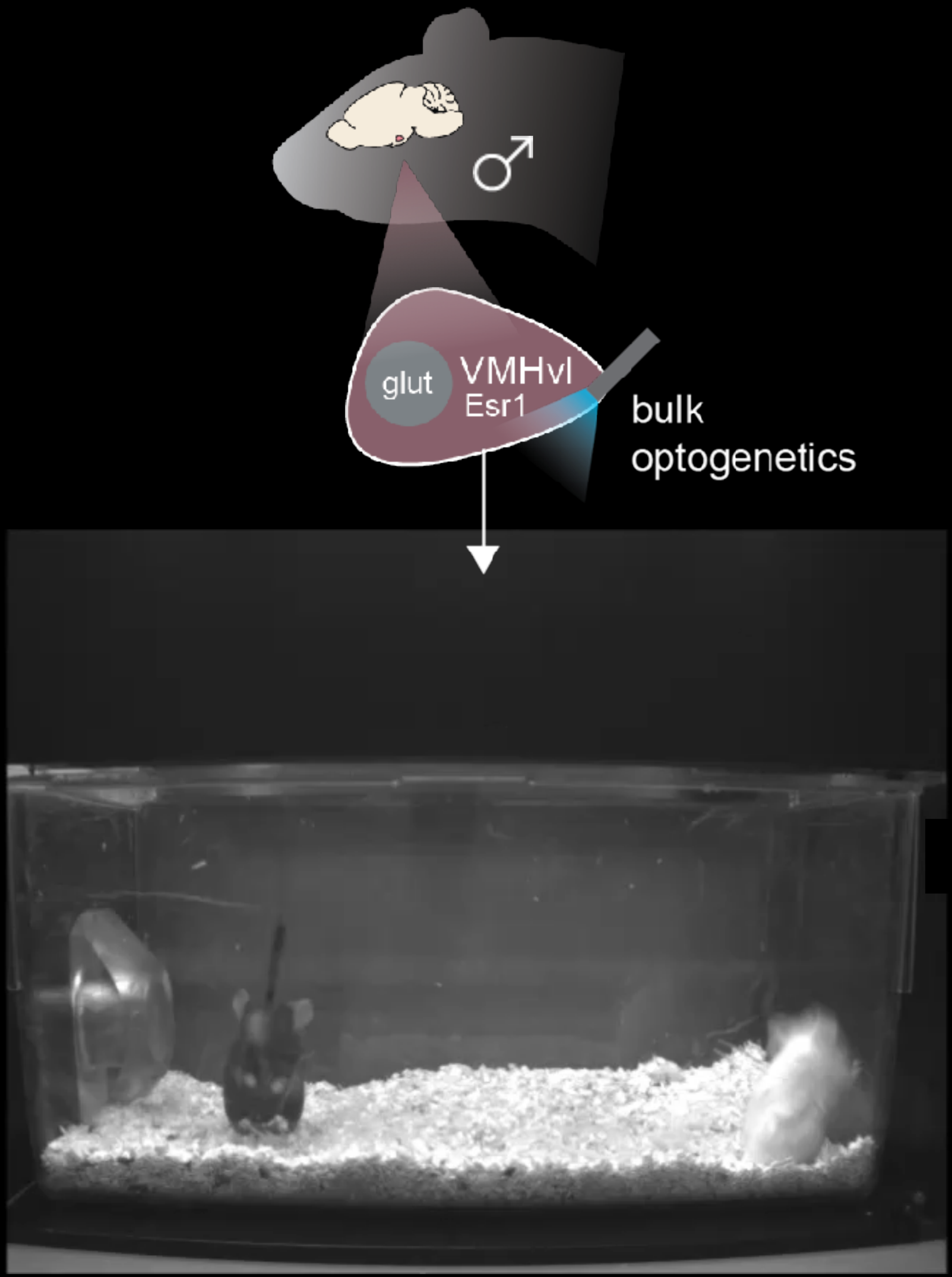
David Anderson



Ann Kennedy

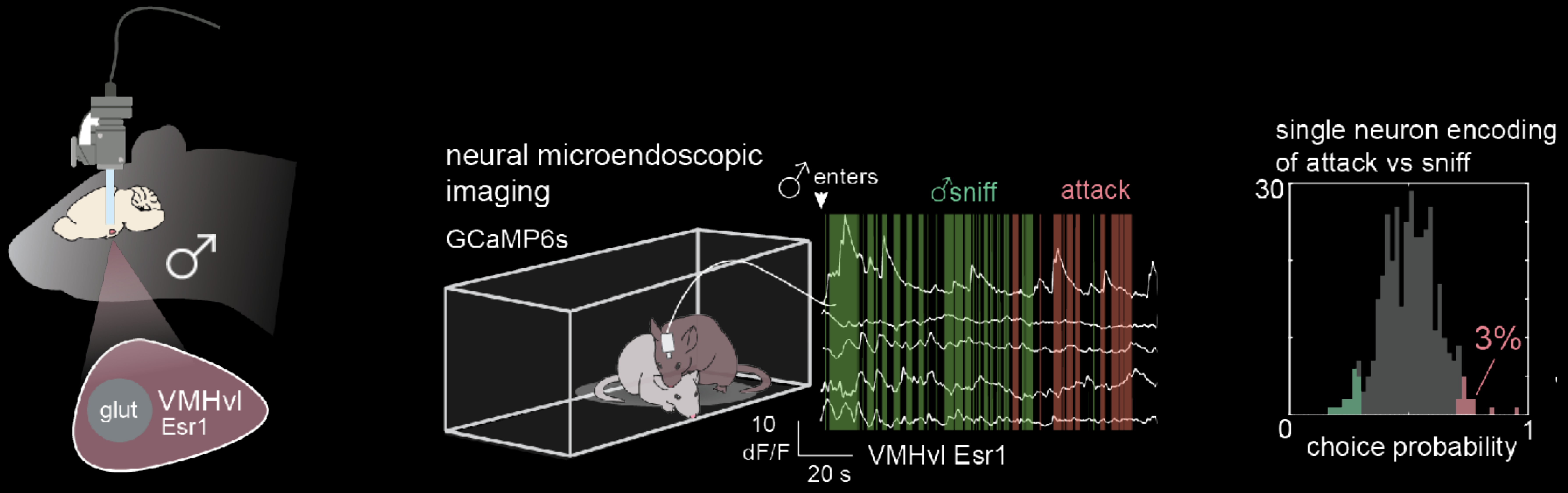
Adi helped prepare some of the slides that follow.

Optogenetic activation of neurons in the hypothalamus elicits attack behavior



Lee et al. (Nature, 2014)

Miniscope imaging in VMHvl during spontaneous aggression shows mixed selectivity



Most neurons in VMHvl are tuned to intruder sex and are active during both sniffing and attack.

Remedios, Kennedy et al. (Nature, 2019)
Karigo et al (Nature, 2021)

Miniscope imaging in VMHvl during spontaneous aggression shows mixed selectivity

Hypothesis

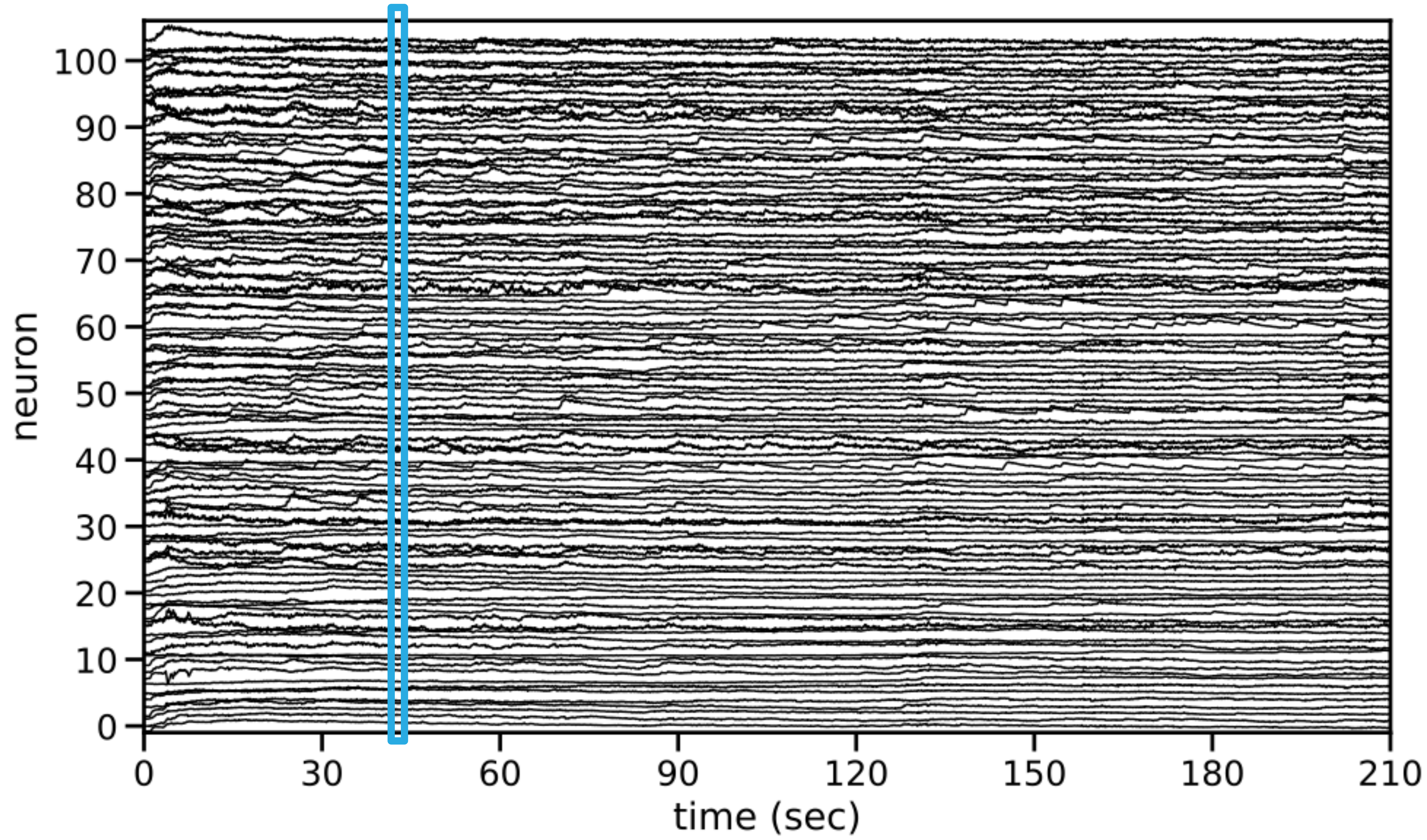
An **internal state of aggressiveness** is encoded in the **collective activity of neurons** in the VMHvl.

Most … attack.

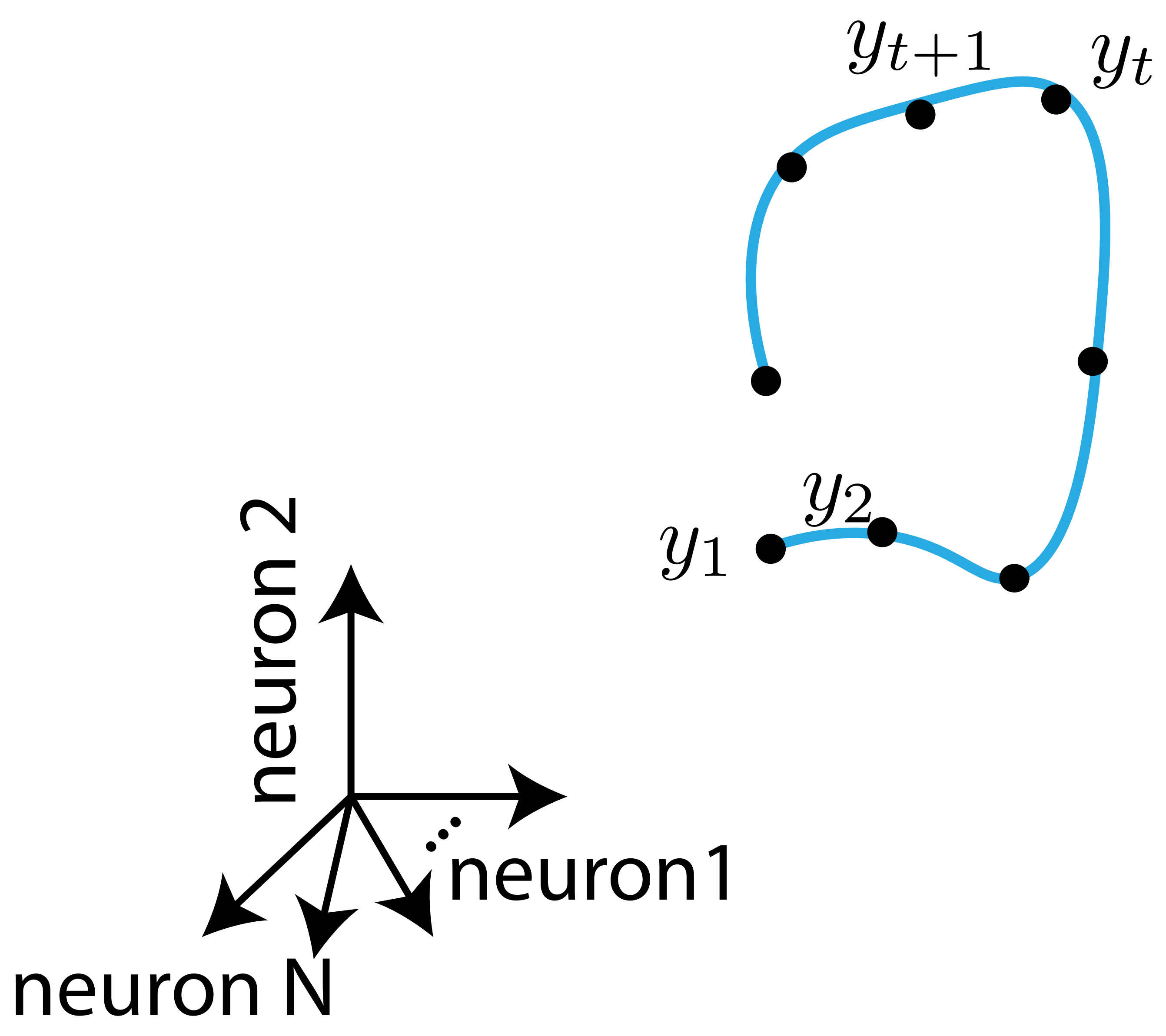
Remedios, Kennedy et al. (Nature, 2019)
Karigo et al (Nature, 2021)

Formalizing this hypothesis with a probabilistic model

$\boldsymbol{y}_t \in \mathbb{R}^N$: neural population activity at time t

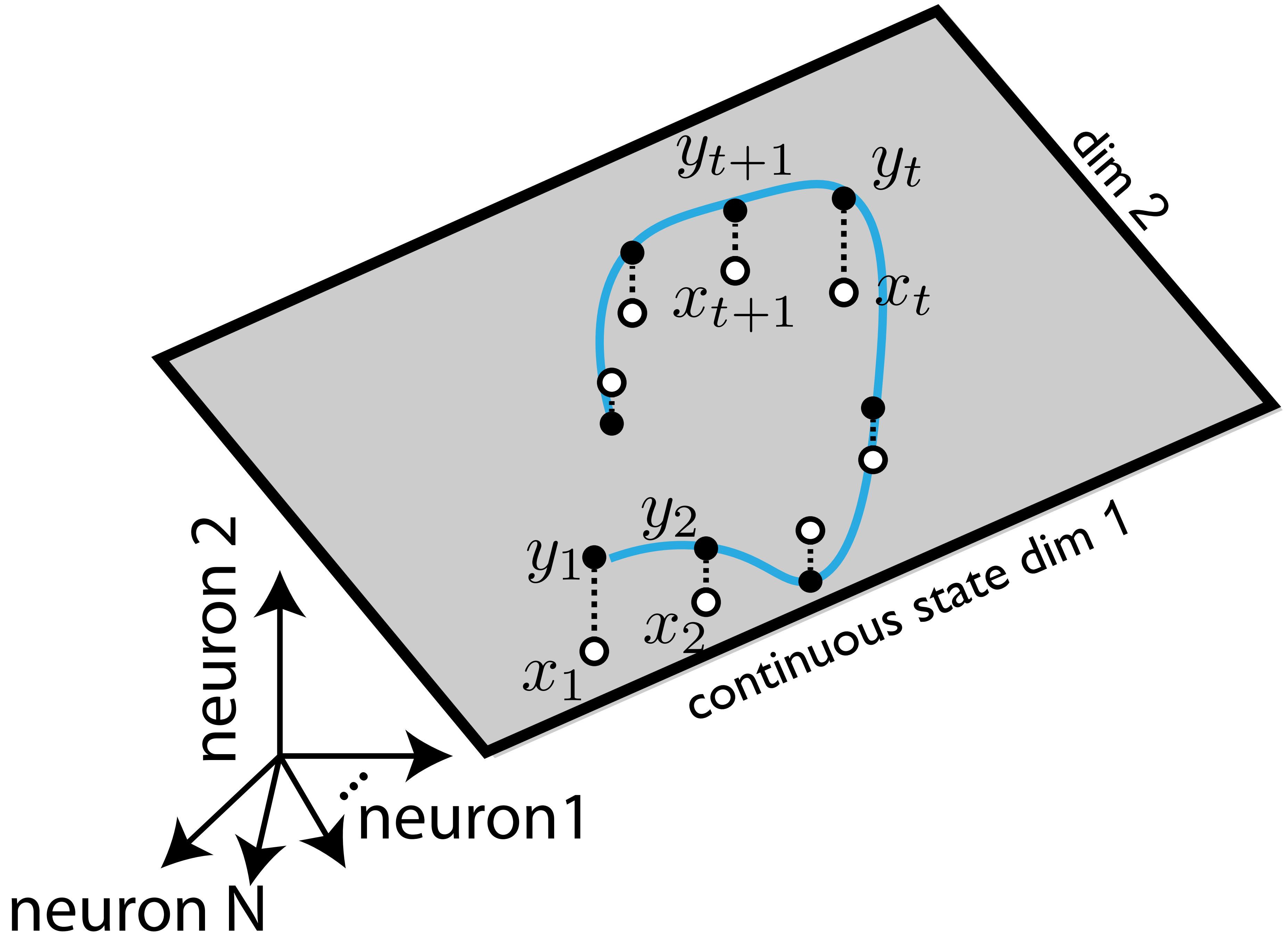


activity traces a trajectory through neural state space



Low-dimensional structure in neural data

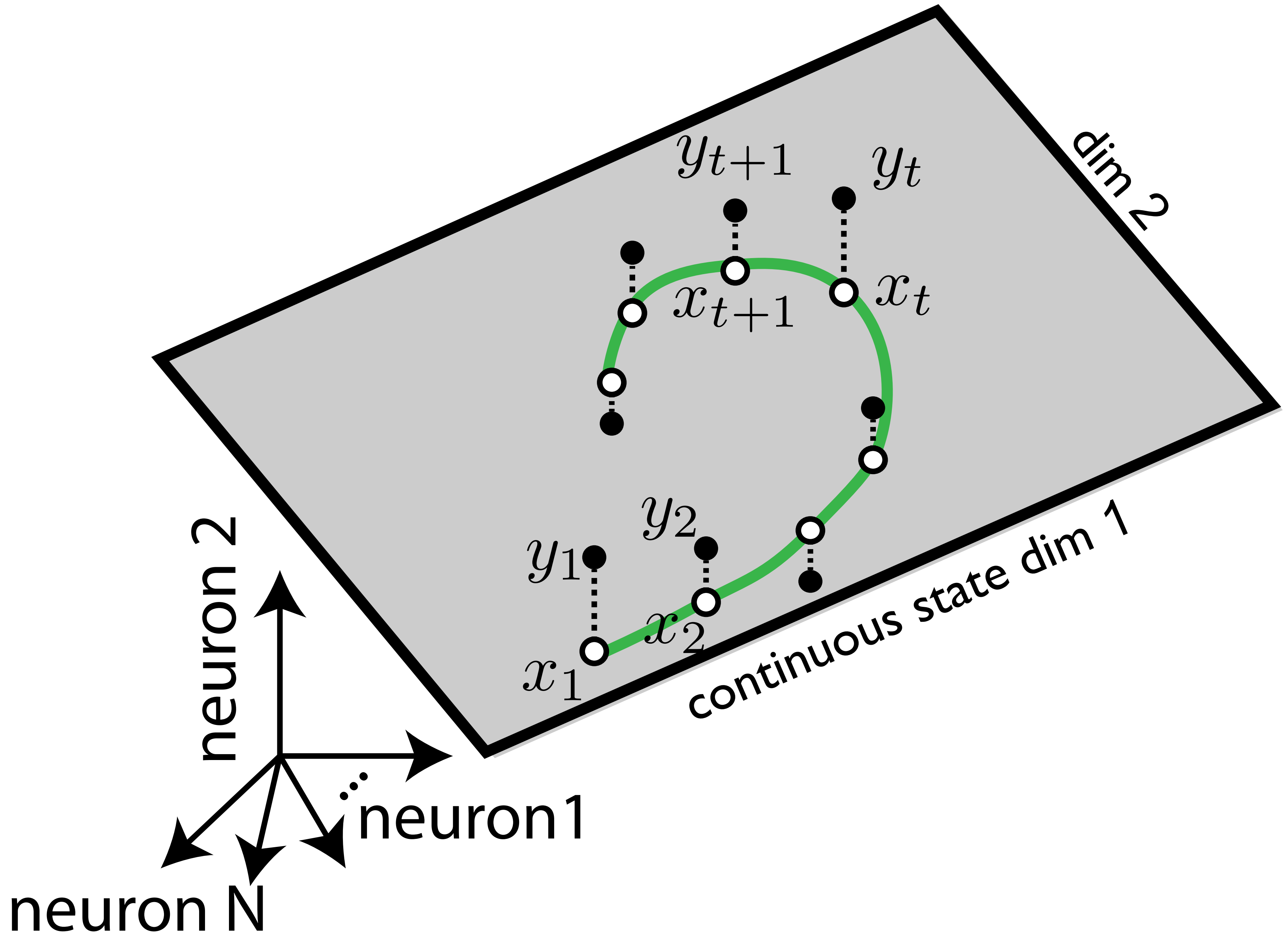
If collective activity encodes a low-dimensional state (e.g., "aggressiveness"), the data should lie near a low-dimensional manifold.



$\boldsymbol{x}_t \in \mathbb{R}^D$: *continuous latent state* (i.e., manifold coordinate)

Low-dimensional structure in neural data

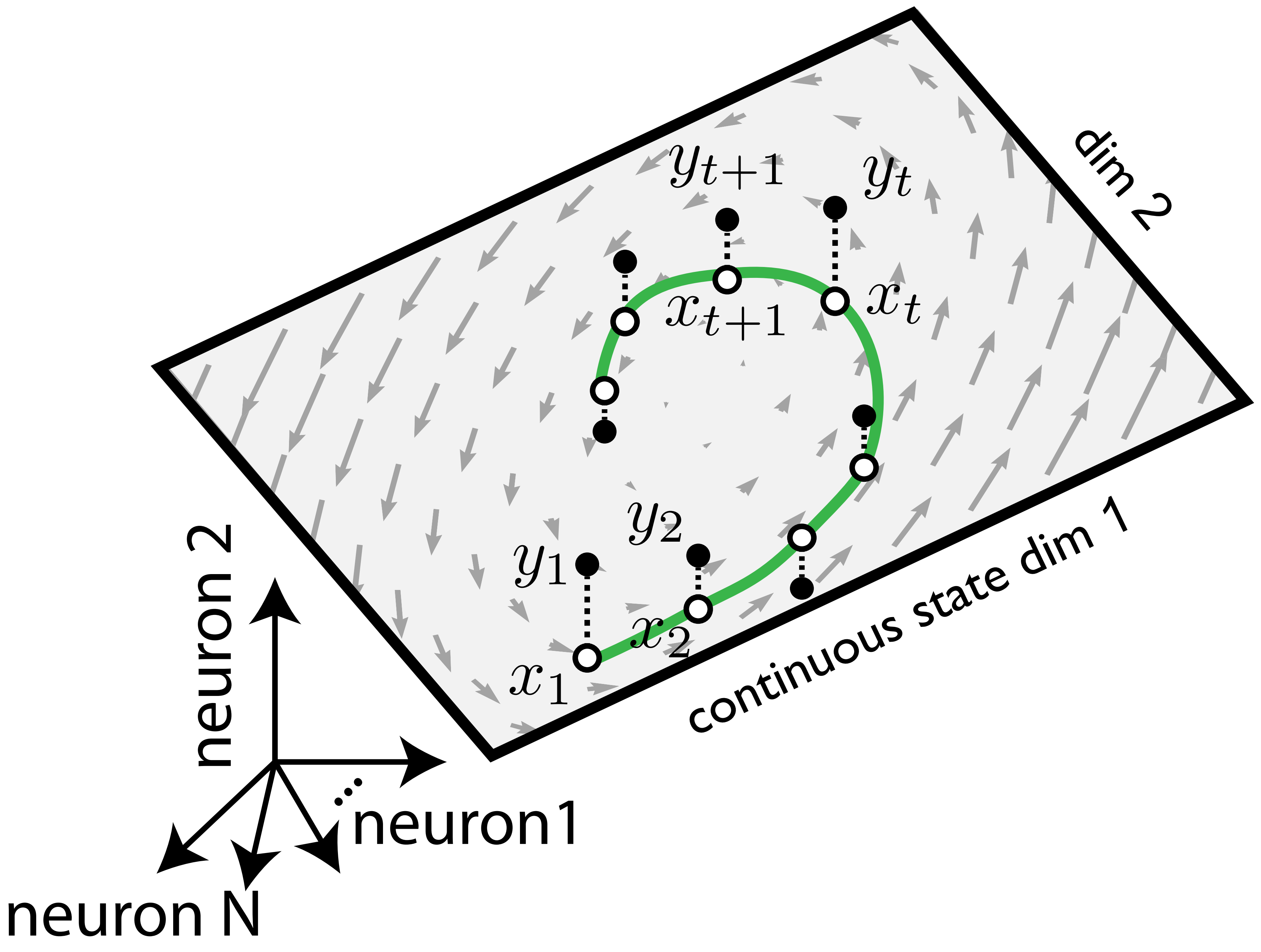
We think of neural activity as a noisy observation of a trajectory on the low-d manifold.



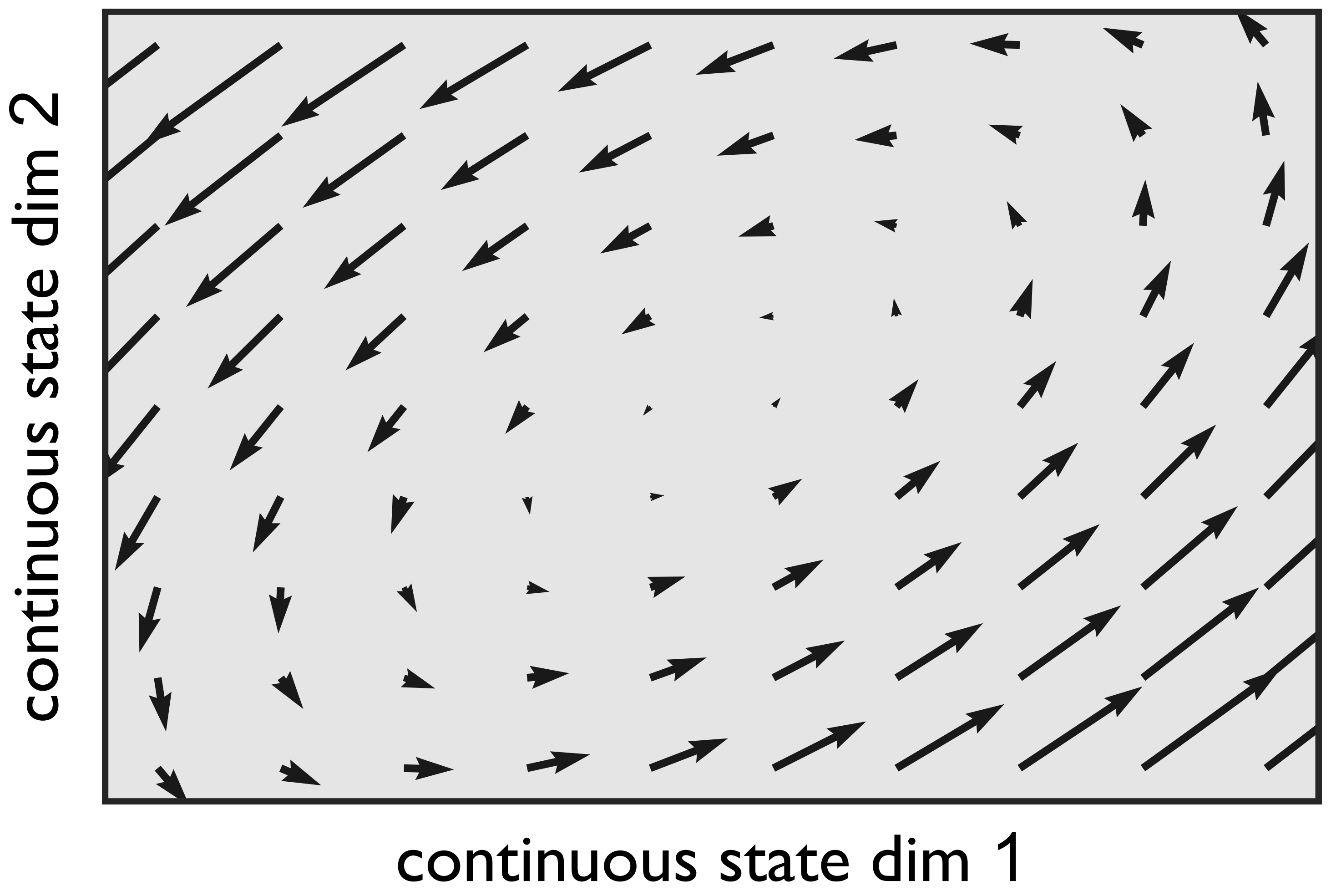
$\boldsymbol{x}_t \in \mathbb{R}^D$: *continuous latent state* (i.e., manifold coordinate)

Low-dimensional structure in neural data

We want to learn the dynamics that govern how trajectories unfold.

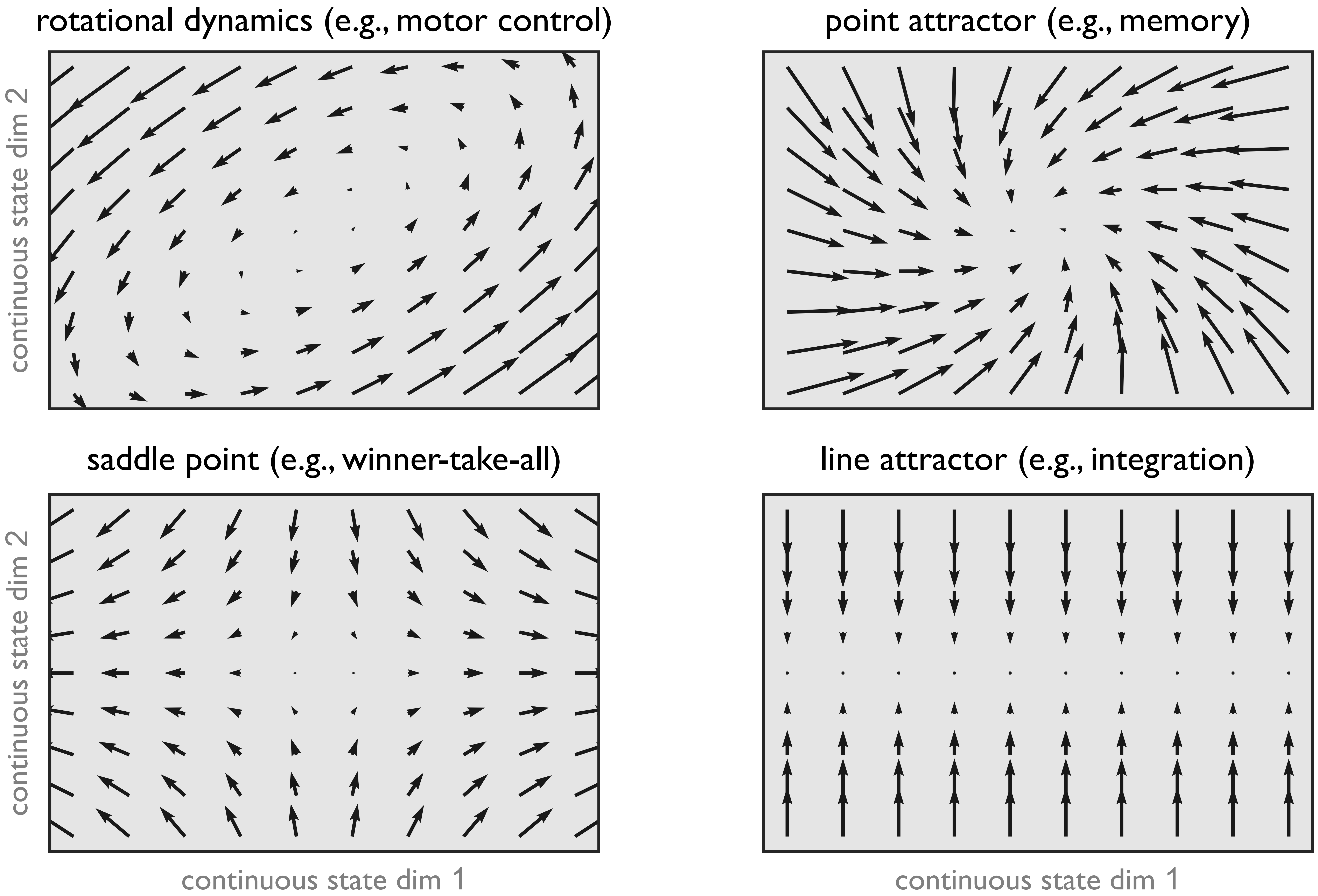


$f : \mathbb{R}^D \mapsto \mathbb{R}^D$: dynamics function



Computation through neural dynamics

Dynamical motifs are hypothesized to underlie various forms of neural computation.

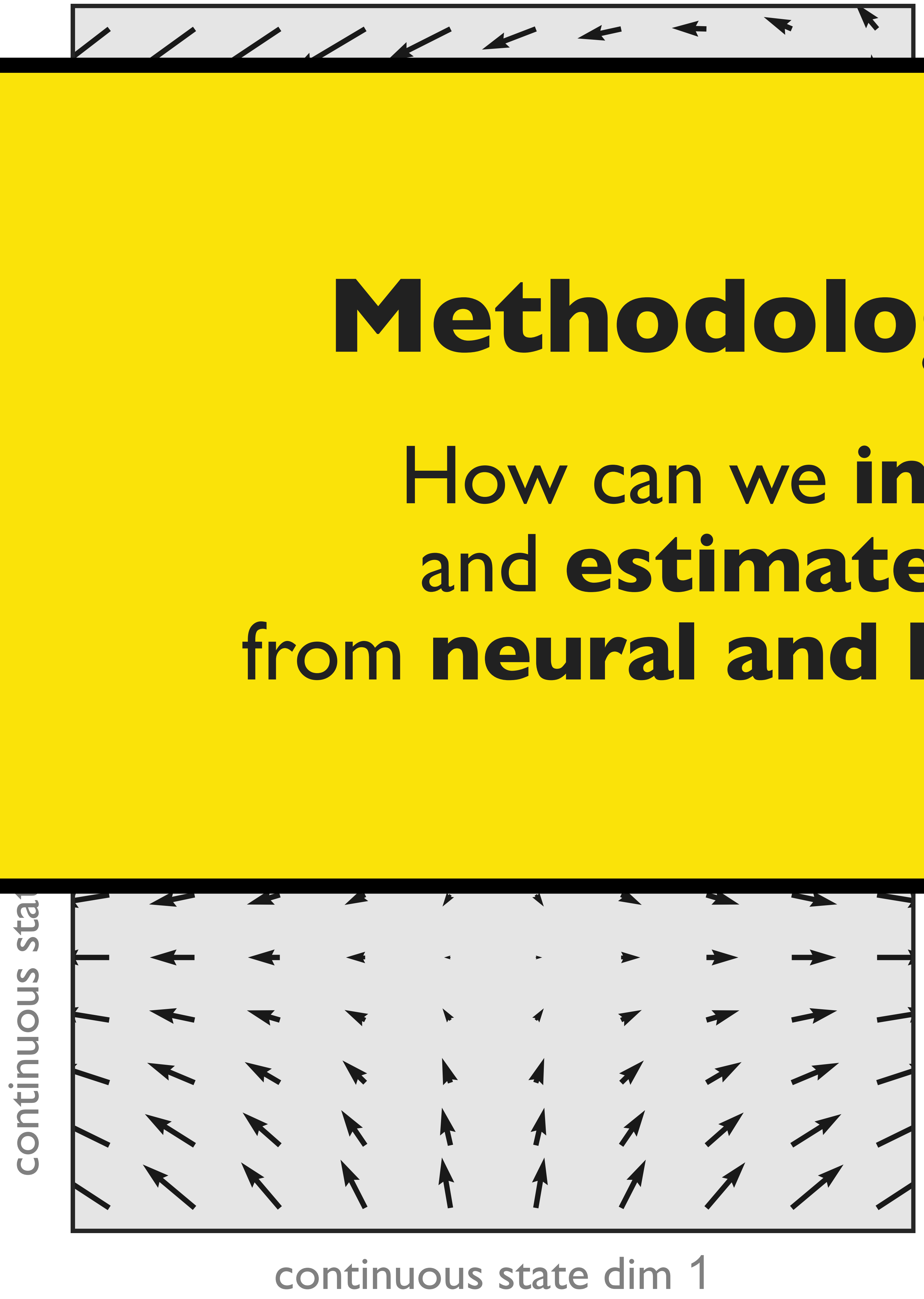


Adapted from Vyas et al. (2020)

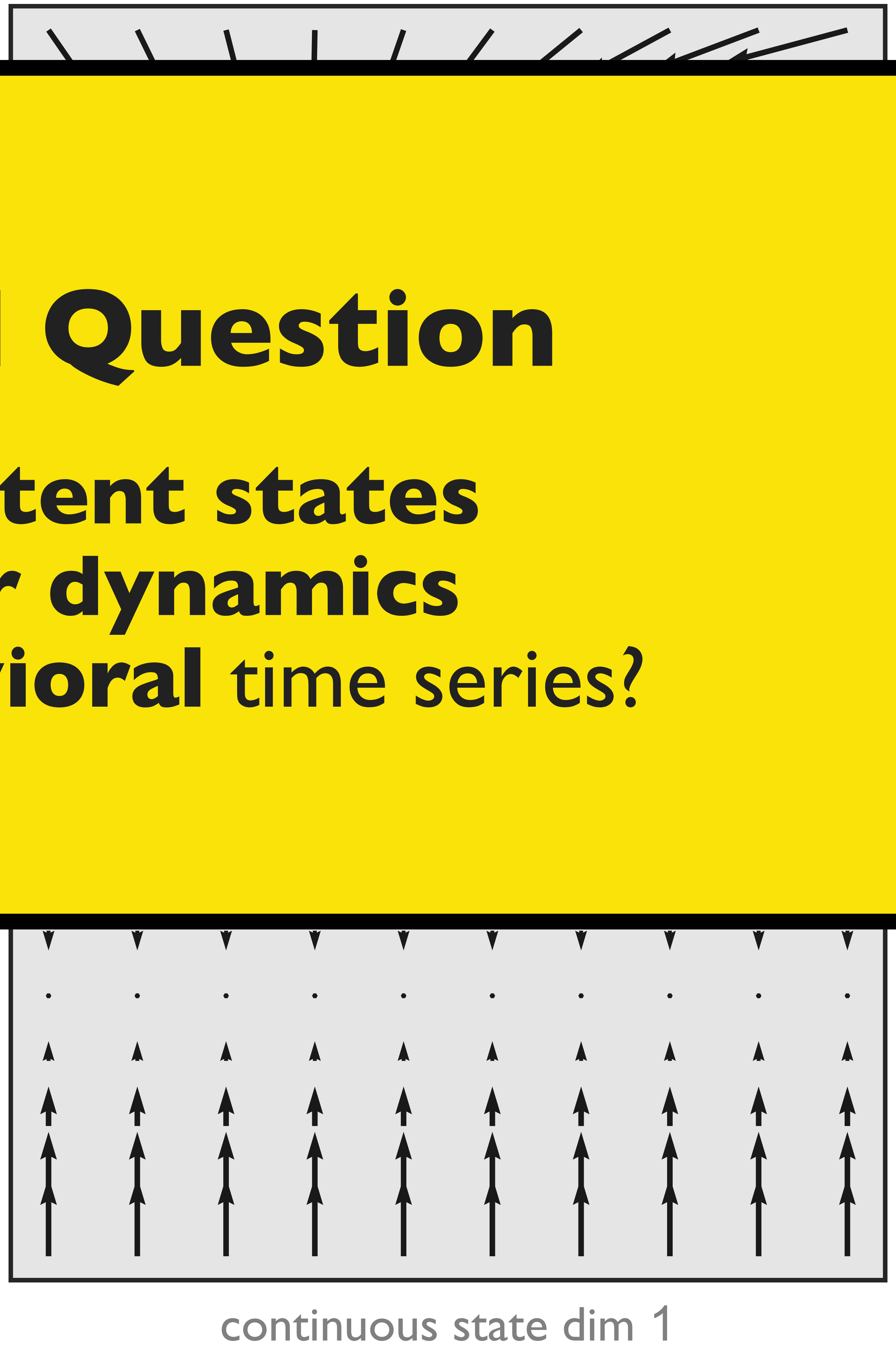
Computation through neural dynamics

Dynamical motifs are hypothesized to underlie various forms of neural computation.

rotational dynamics (e.g., motor control)



point attractor (e.g., memory)

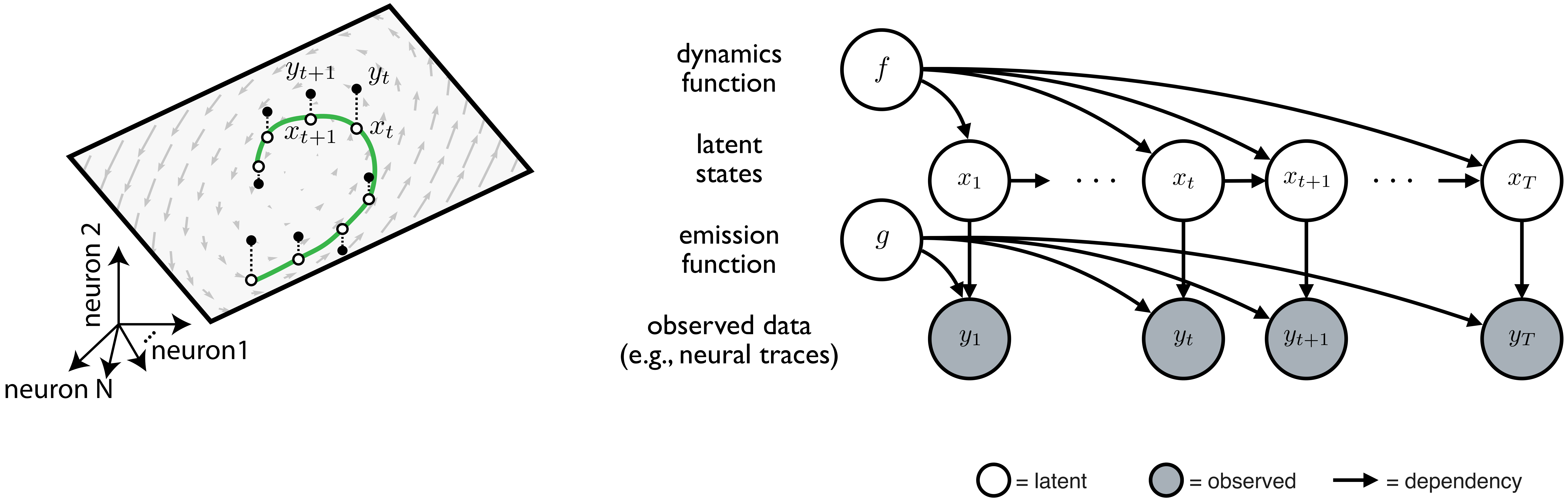


Methodological Question

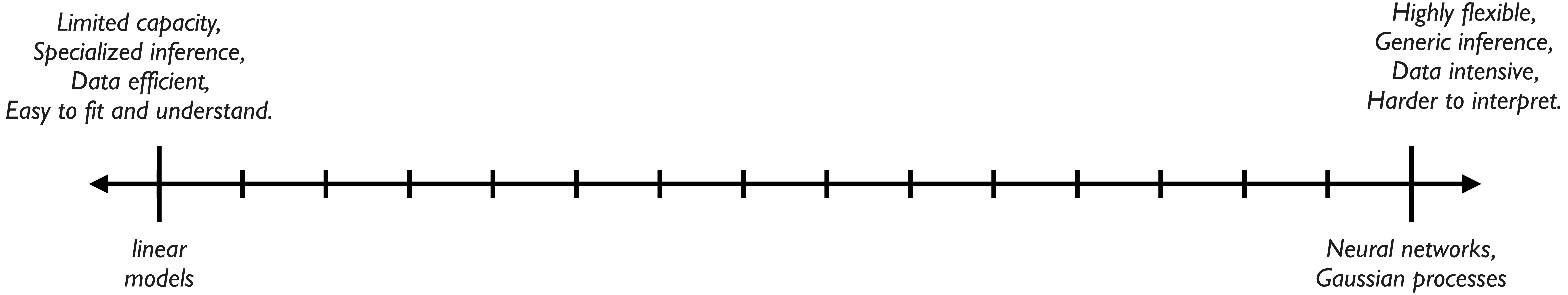
How can we **infer latent states**
and **estimate their dynamics**
from **neural and behavioral** time series?

Adapted from Vyas et al. (2020)

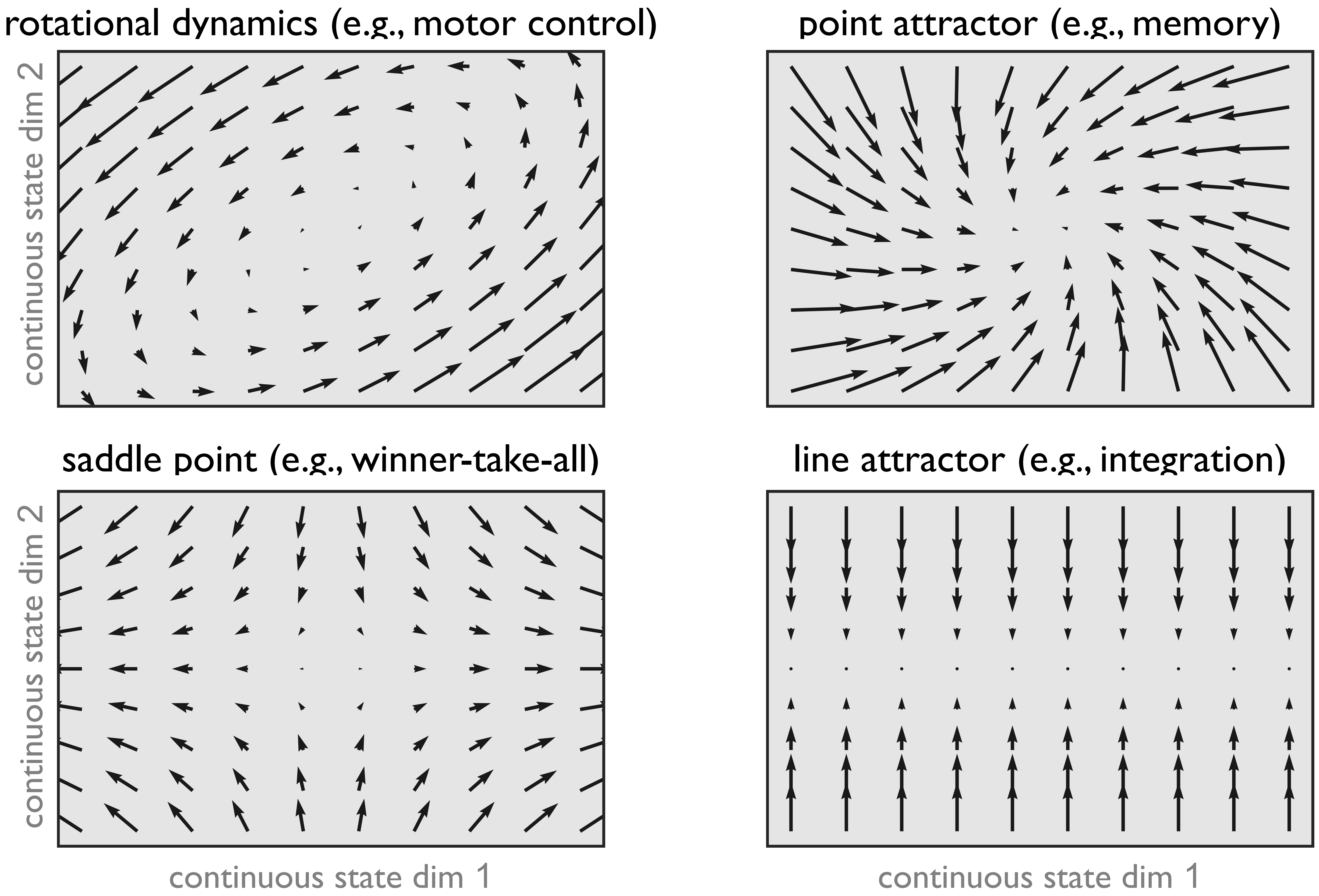
Probabilistic state space models



A spectrum of dynamics models



What can linear models do?



A lot! E.g., the motifs from before were all linear models, $f(x) = Ax + b$.

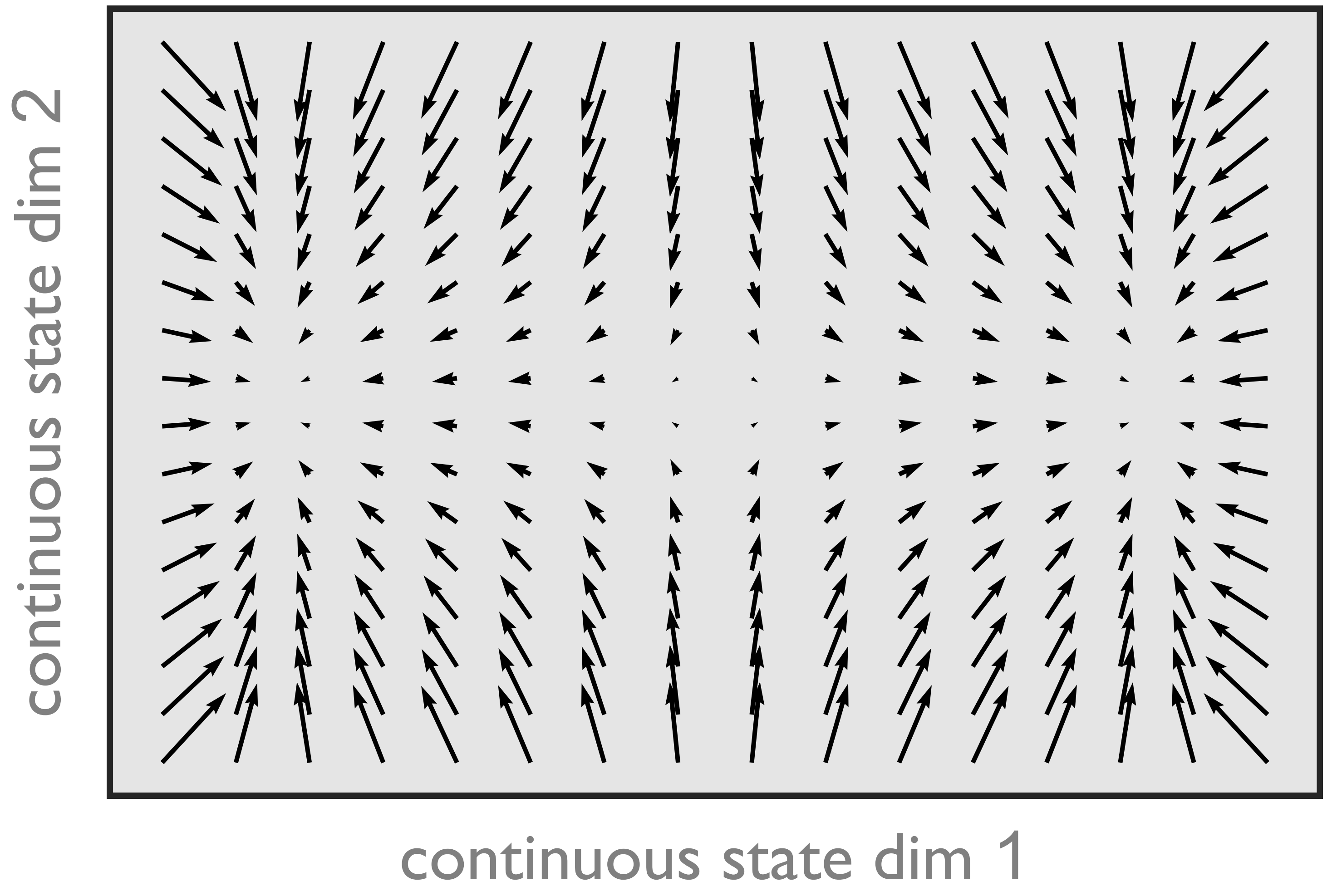
Moreover, linear systems are interpretable.

We can find analytical solutions for:
- fixed points
- stationary distribution (w/ Gaussian noise)
- dynamics along eigenmodes
- optimal control

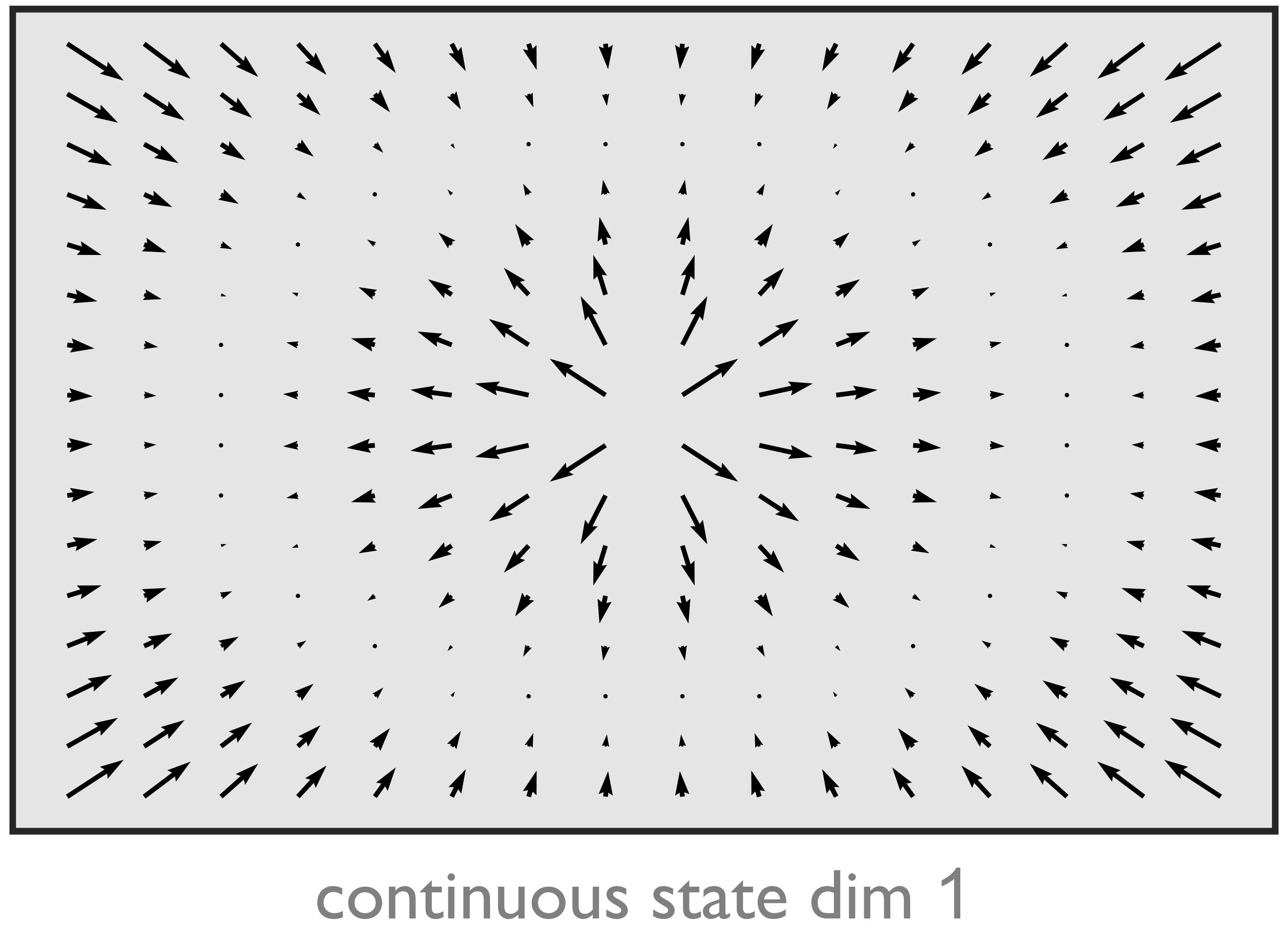
What **can't** linear models do?

Still, most computations require nonlinear dynamics.

bistability (e.g., decision making)



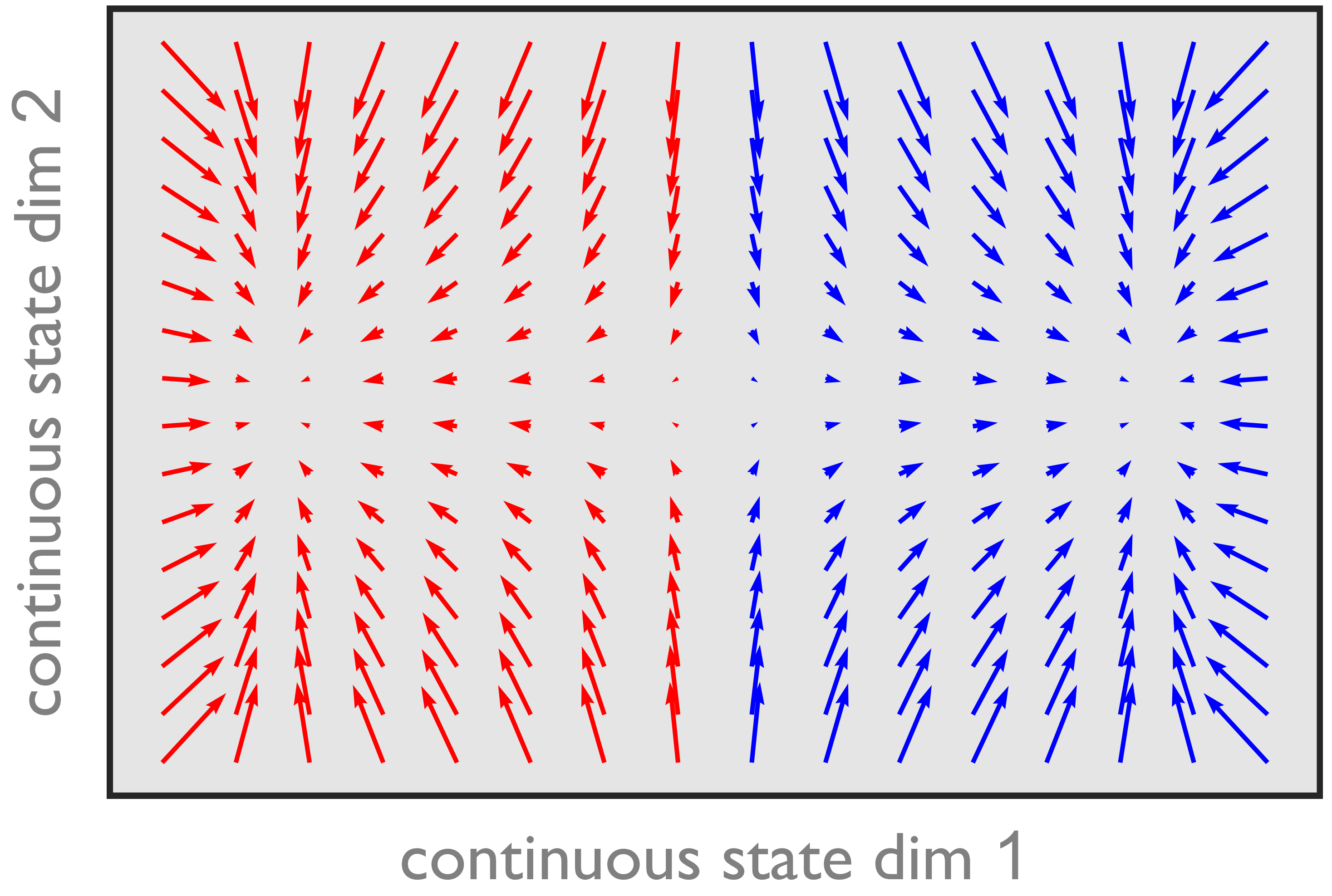
ring attractor (e.g., head direction)



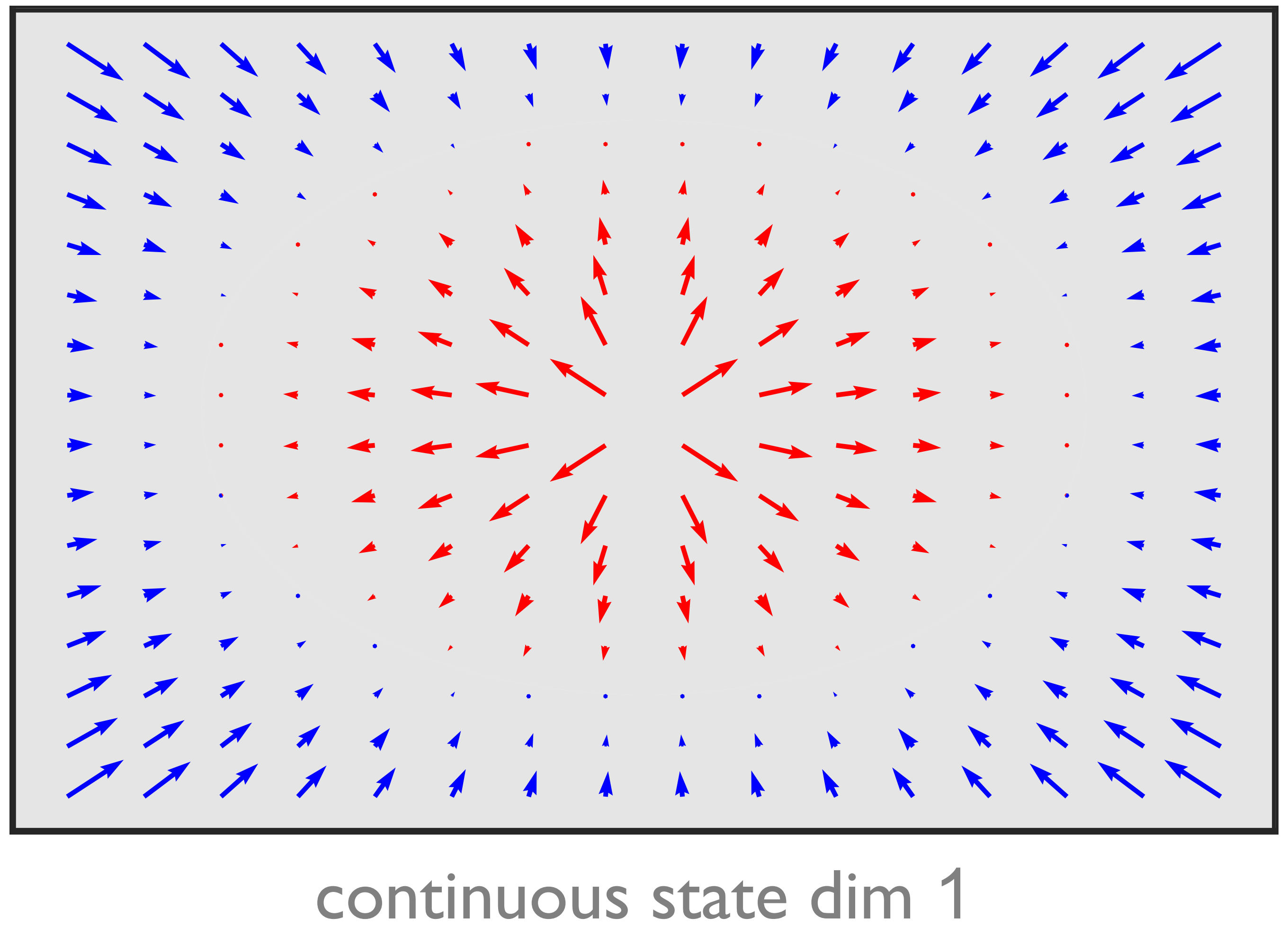
Key idea: nonlinear dynamics can often be approximated as piecewise-linear

Indeed, that's often how we analyze nonlinear dynamical systems!

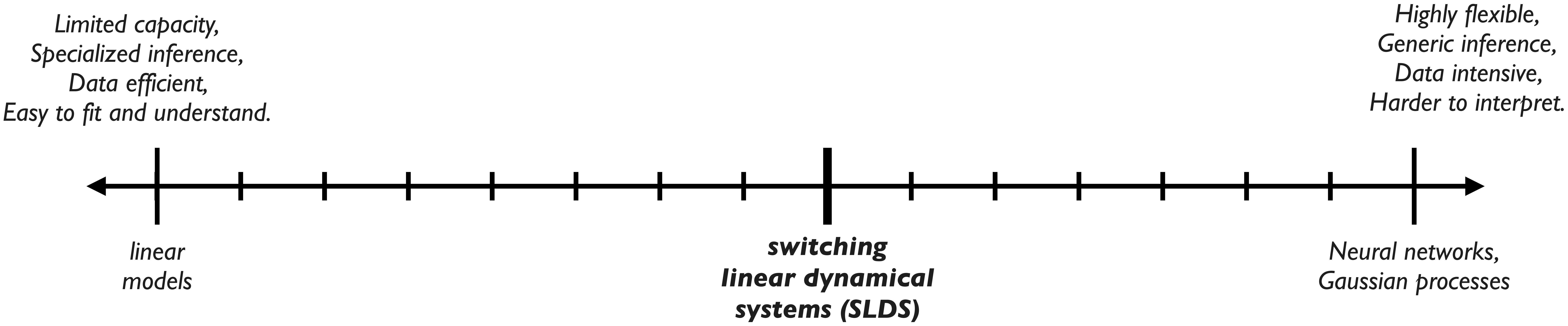
bistability (e.g., decision making)



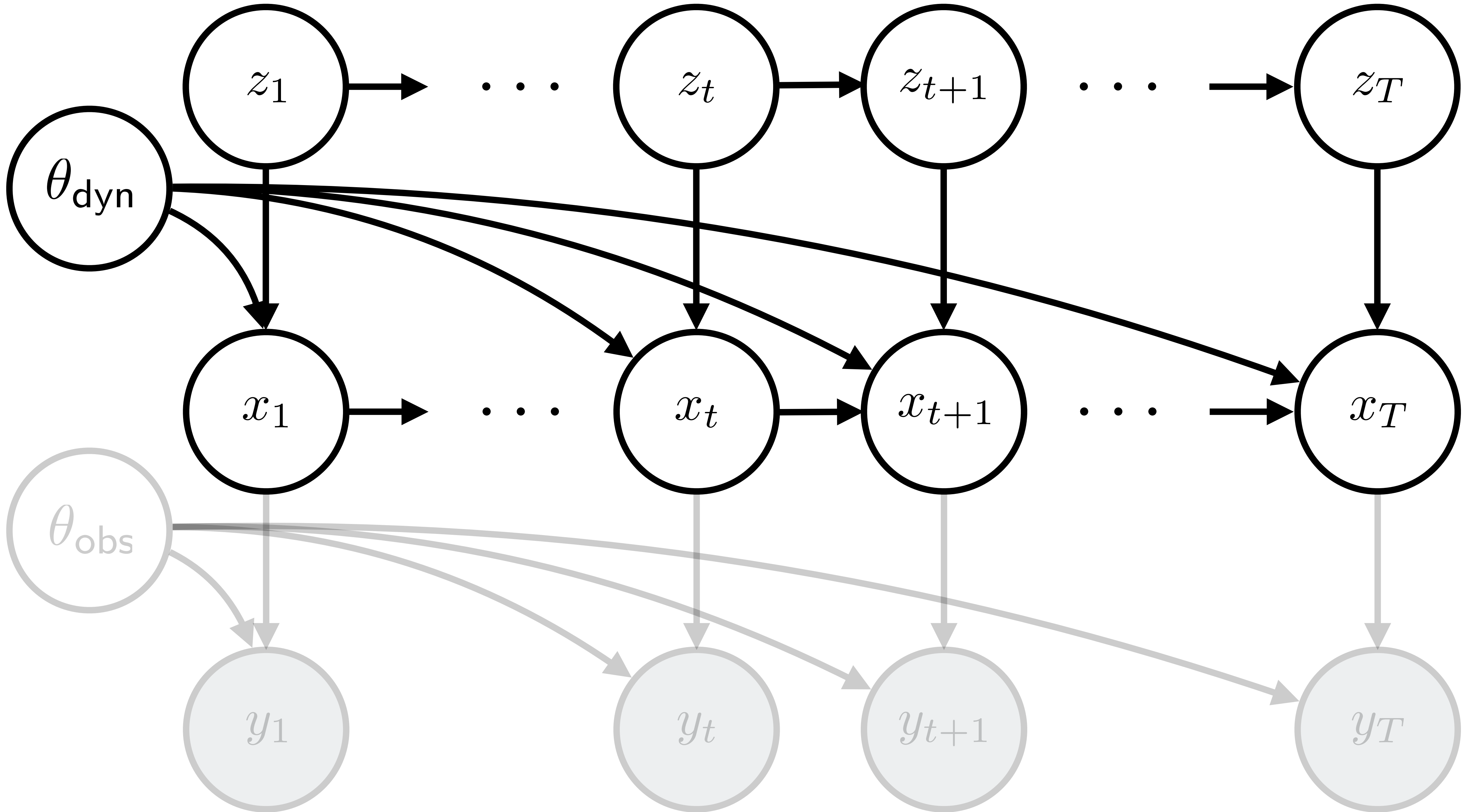
ring attractor (e.g., head direction)



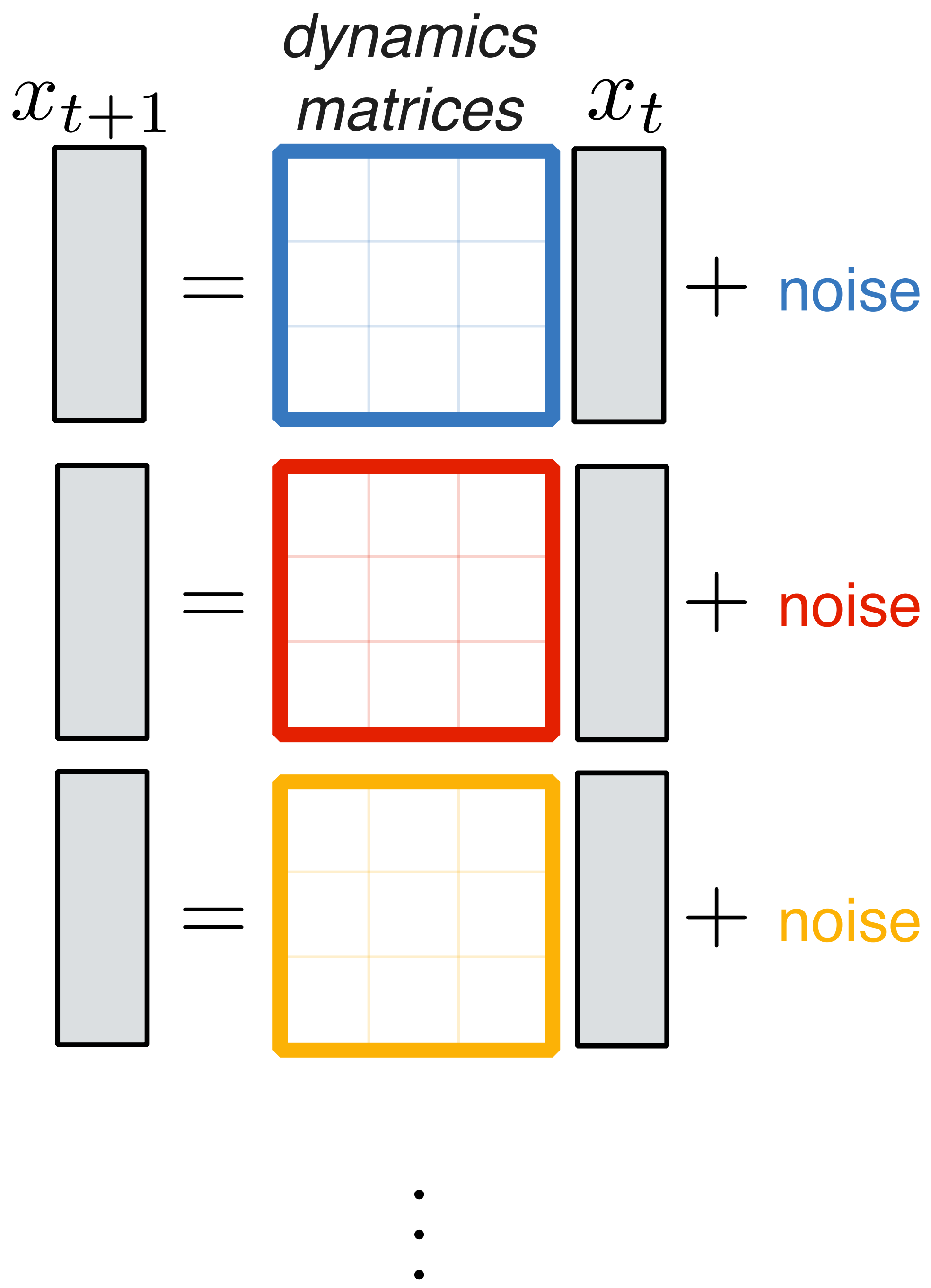
A spectrum of dynamics models



Switching linear dynamical systems (SLDS)

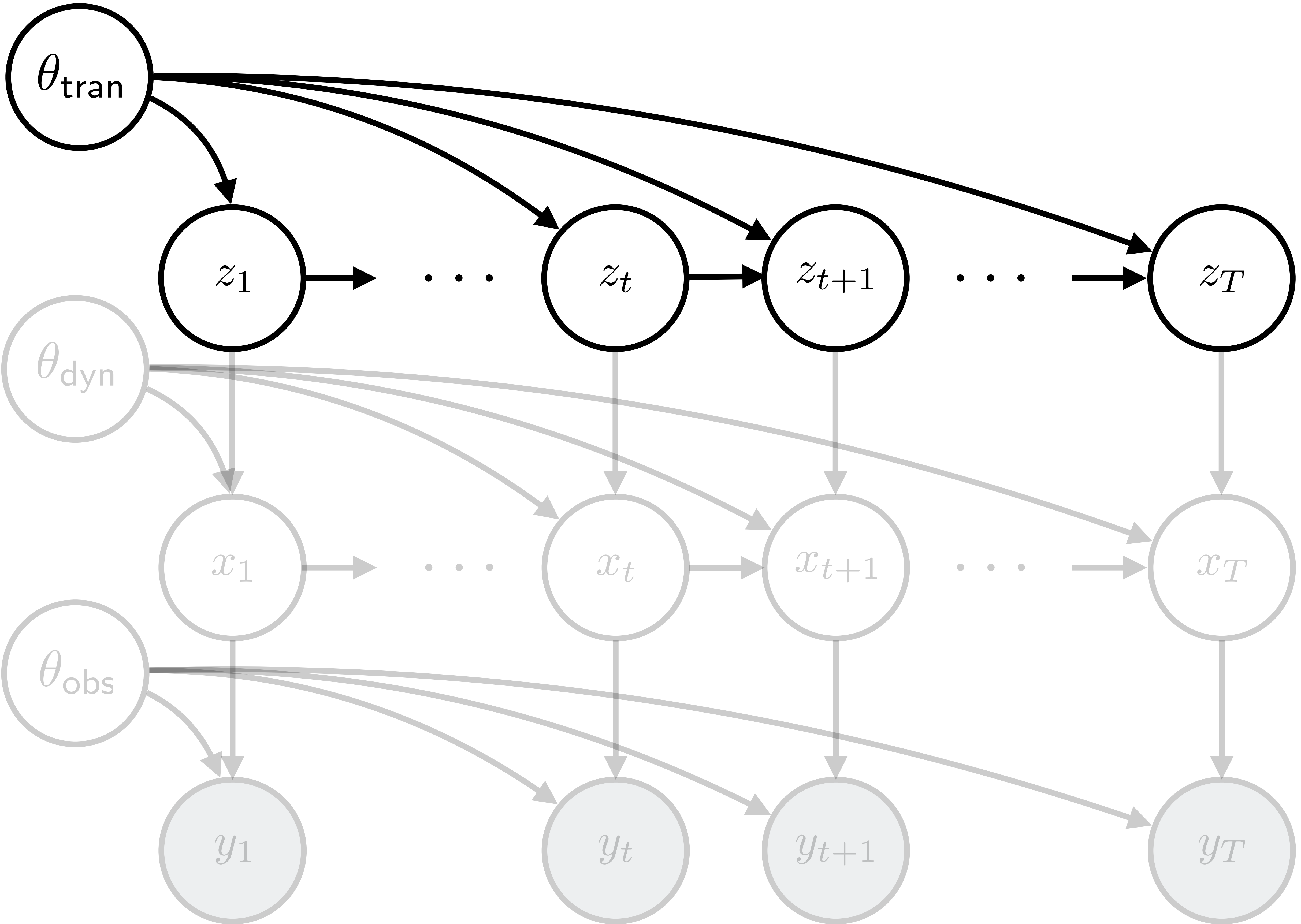


Different **linear dynamics**
in each discrete state

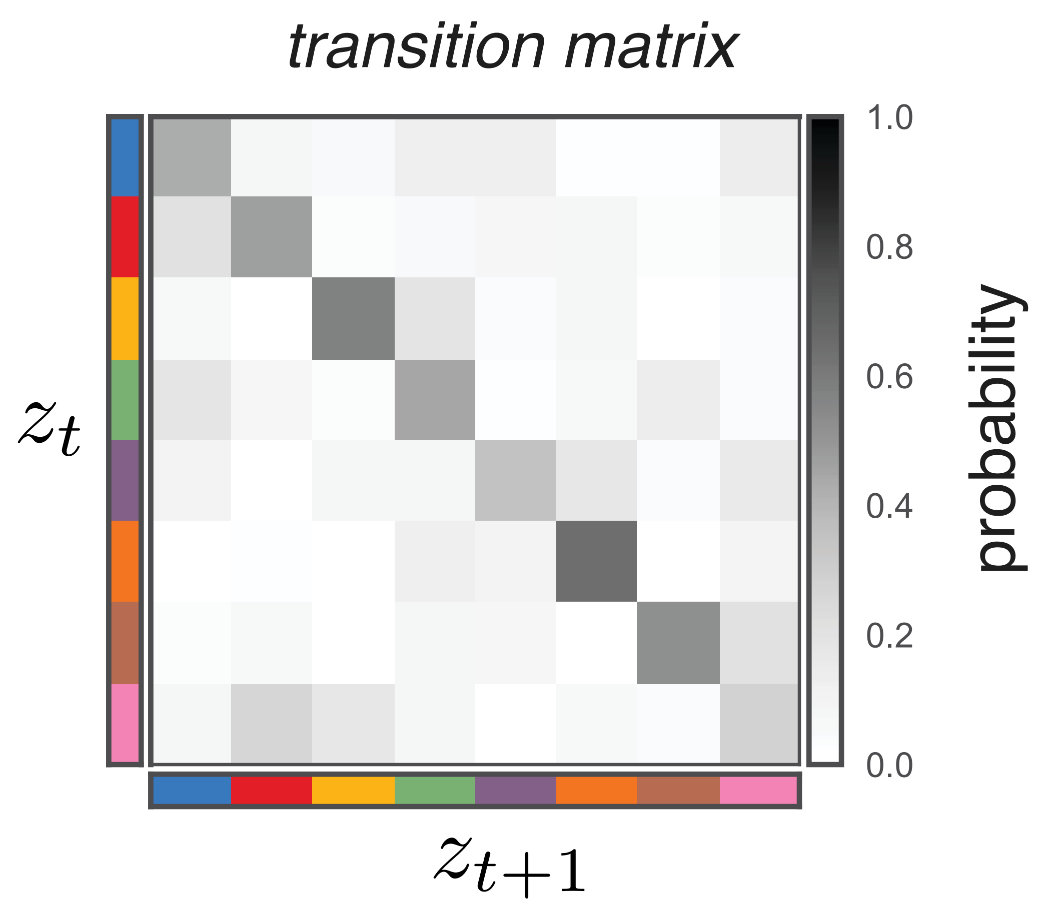


Ackerson and Fu (1970)
Chang and Athans (1978)
Hamilton (1990)
Ghahramani and Hinton (1996)
Murphy (1998)
Fox et al (2009)

Switching linear dynamical systems (SLDS)

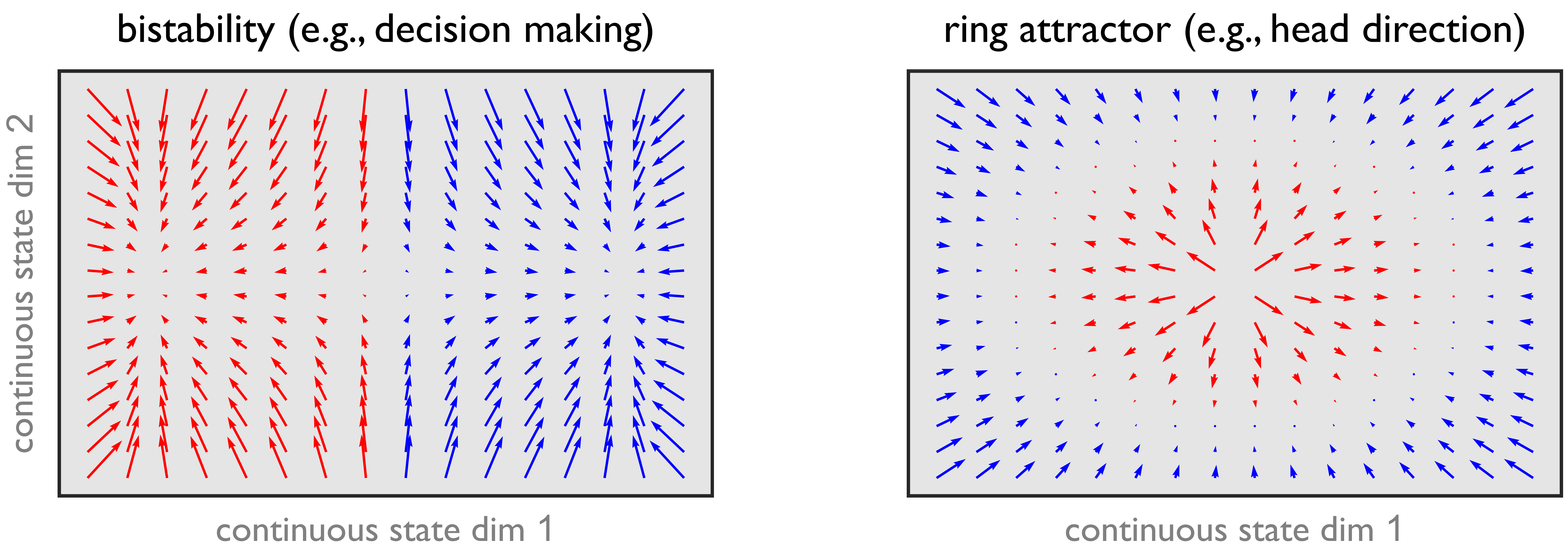


State-dependent
switching probabilities

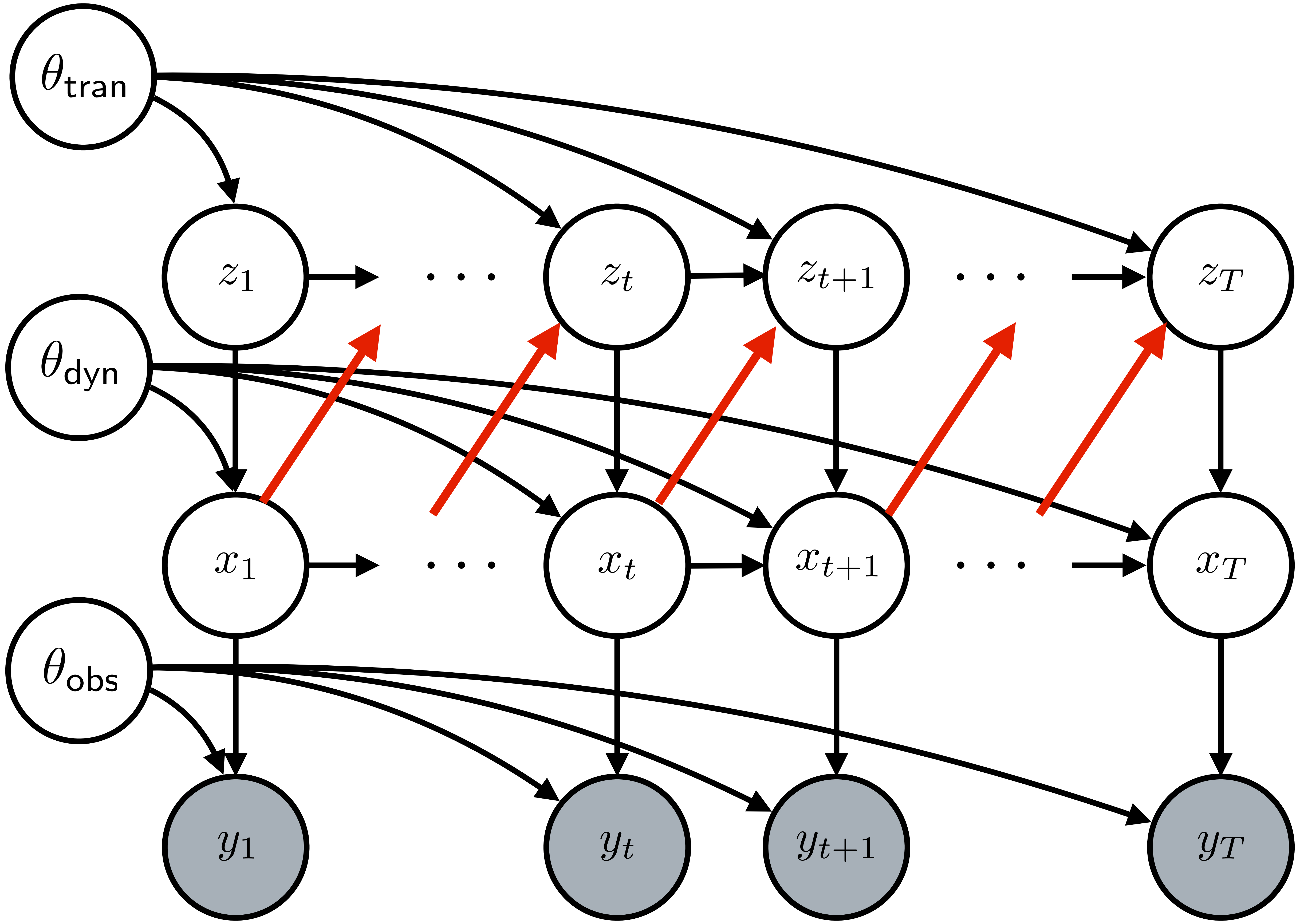


Ackerson and Fu (1970)
Chang and Athans (1978)
Hamilton (1990)
Ghahramani and Hinton (1996)
Murphy (1998)
Fox et al (2009)

Problem: in an SLDS, discrete state transitions are independent of location!

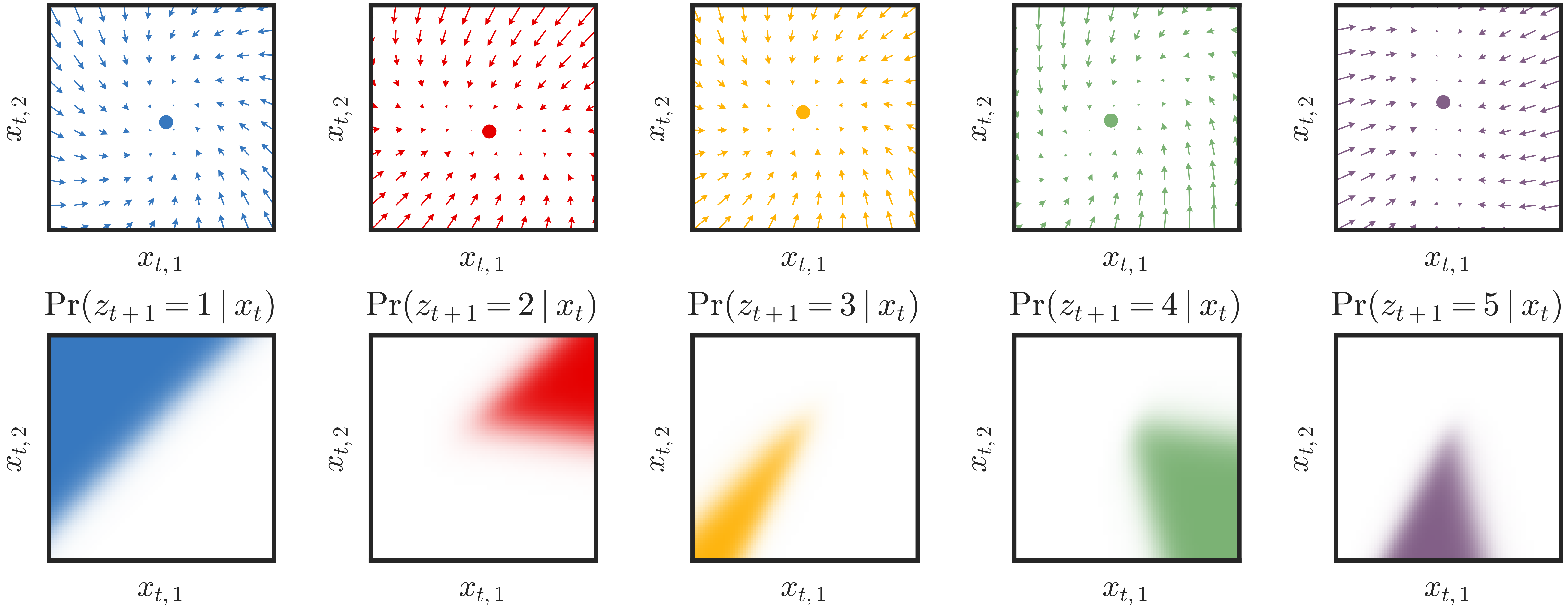


Recurrent switching linear dynamical systems (**r**SLDS)



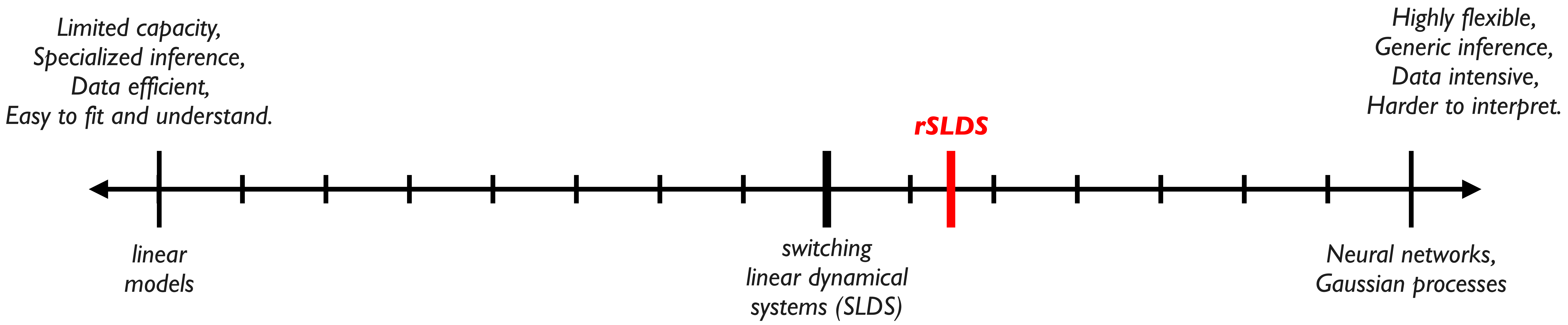
Linderman et al. (AISTATS, 2017)
Zoltowski, Pillow, & Linderman (2020)
…and now many more

Recurrent SLDS partition continuous state space into regions with linear dynamics

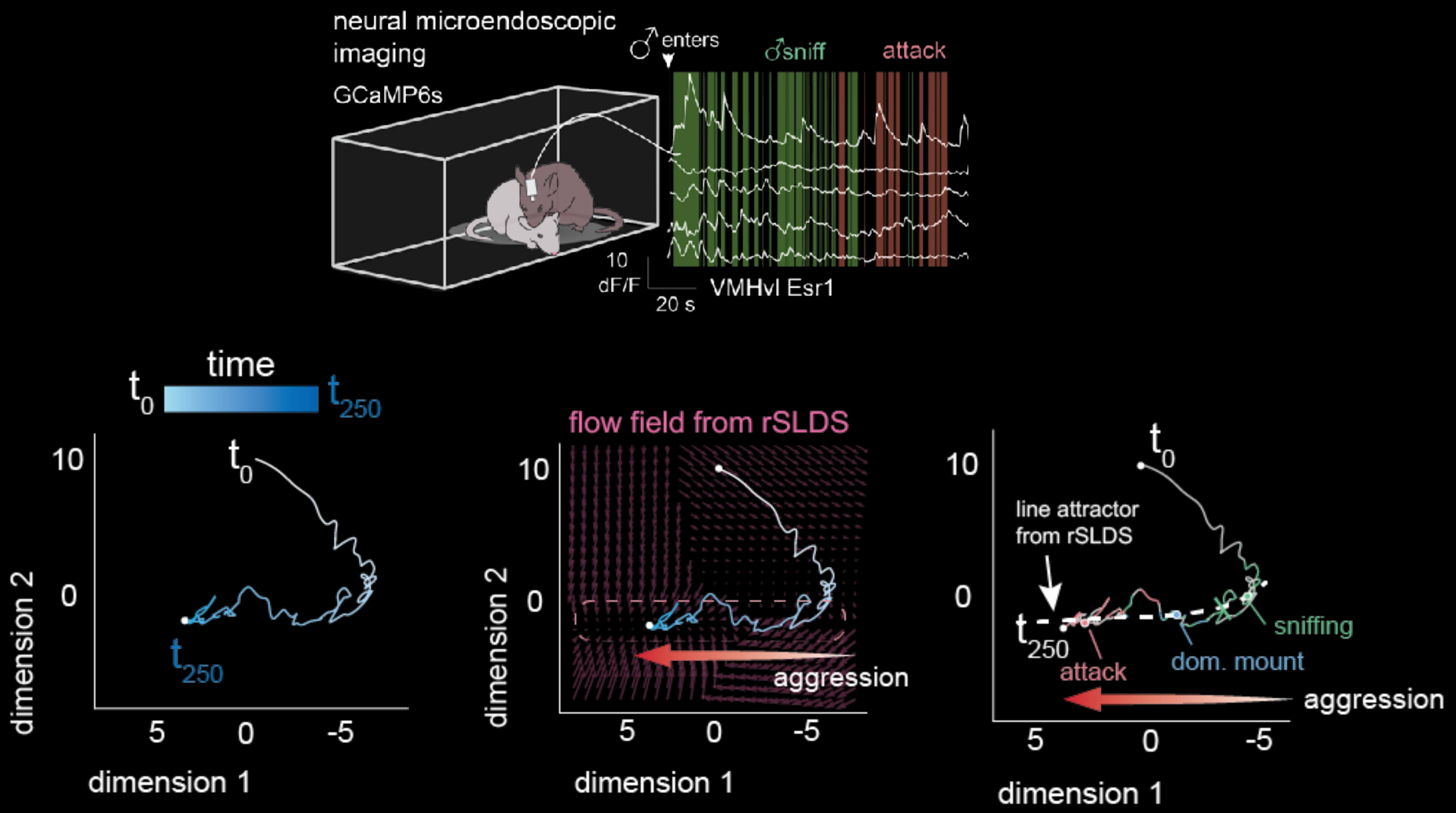


Linderman et al. (AISTATS, 2017)
Zoltowski, Pillow, & Linderman (2020)
…and now many more

A spectrum of dynamics models

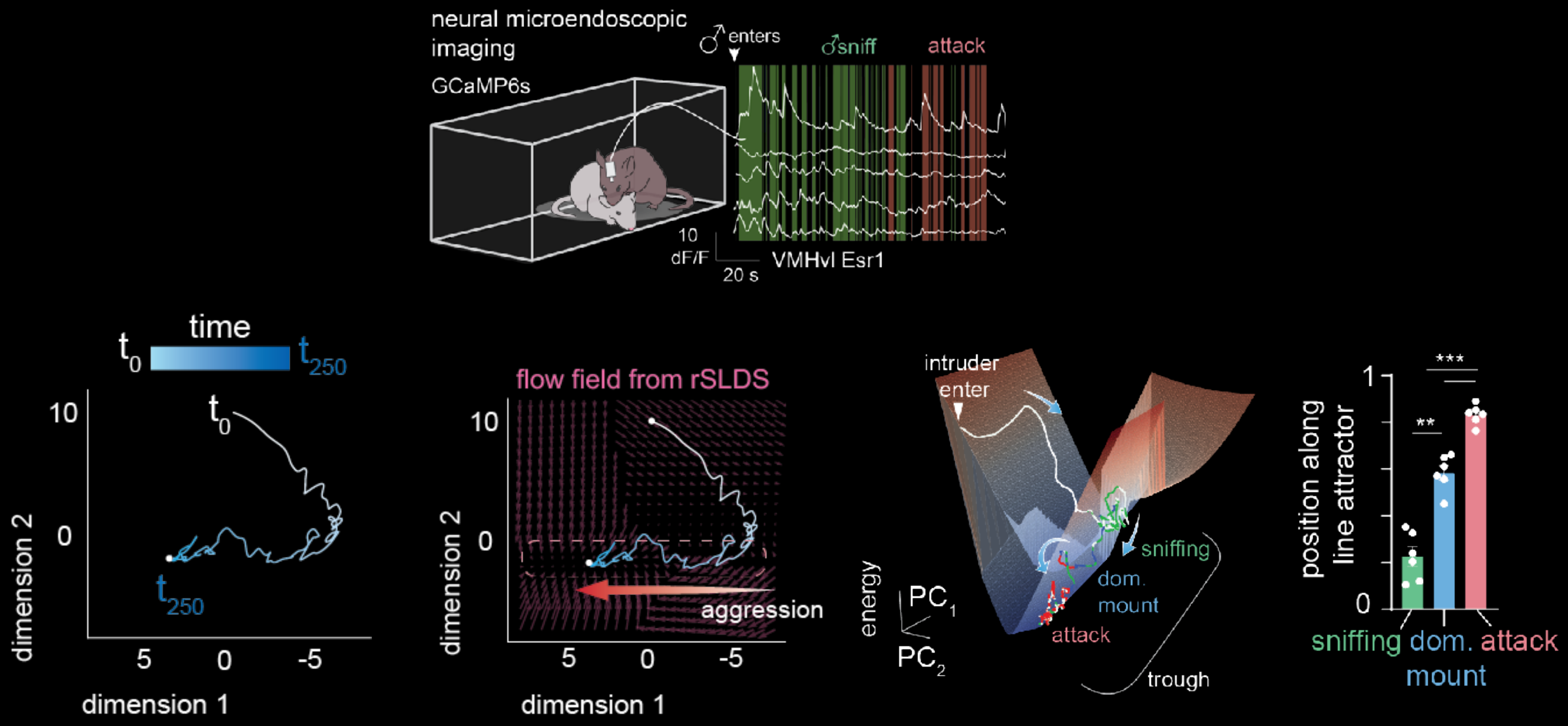


rSLDS analysis reveals line attractor-like dynamics in VMHvl



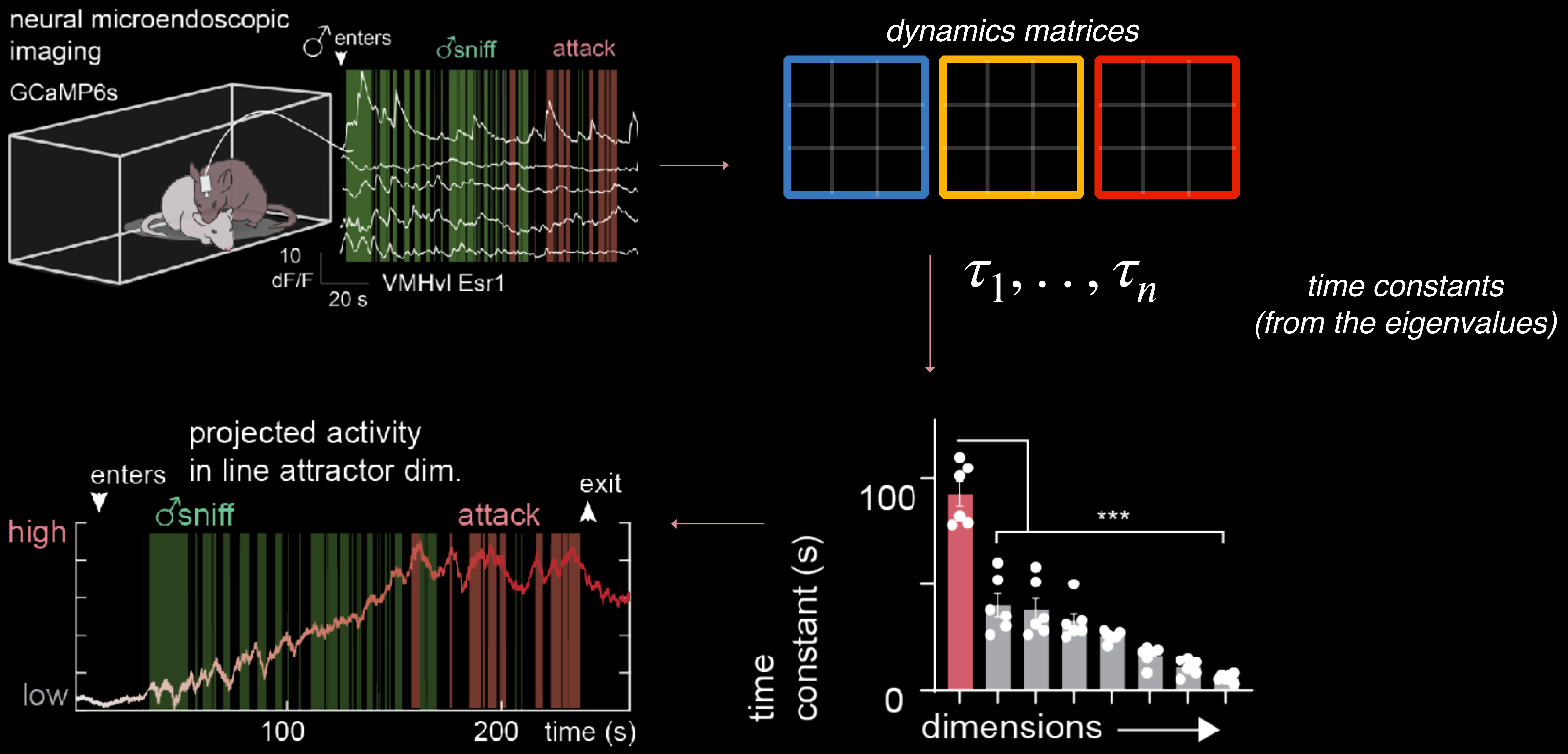
Nair et al. (Cell, 2023)

rSLDS analysis reveals line attractor-like dynamics in VMHvl



Nair et al. (Cell, 2023)

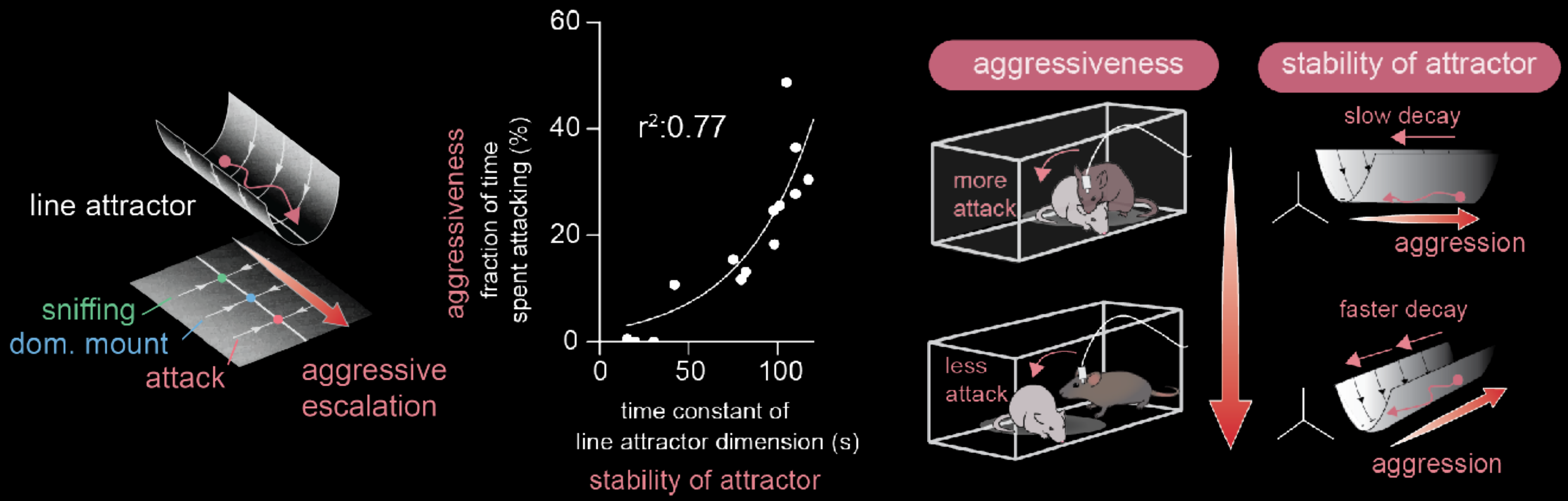
rSLDS analysis reveals line attractor-like dynamics in VMHvl



Importantly, this is not true of all hypothalamic nuclei, e.g., MPOA.

Nair et al. (Cell, 2023)

Dynamical systems explain individual differences in aggressiveness

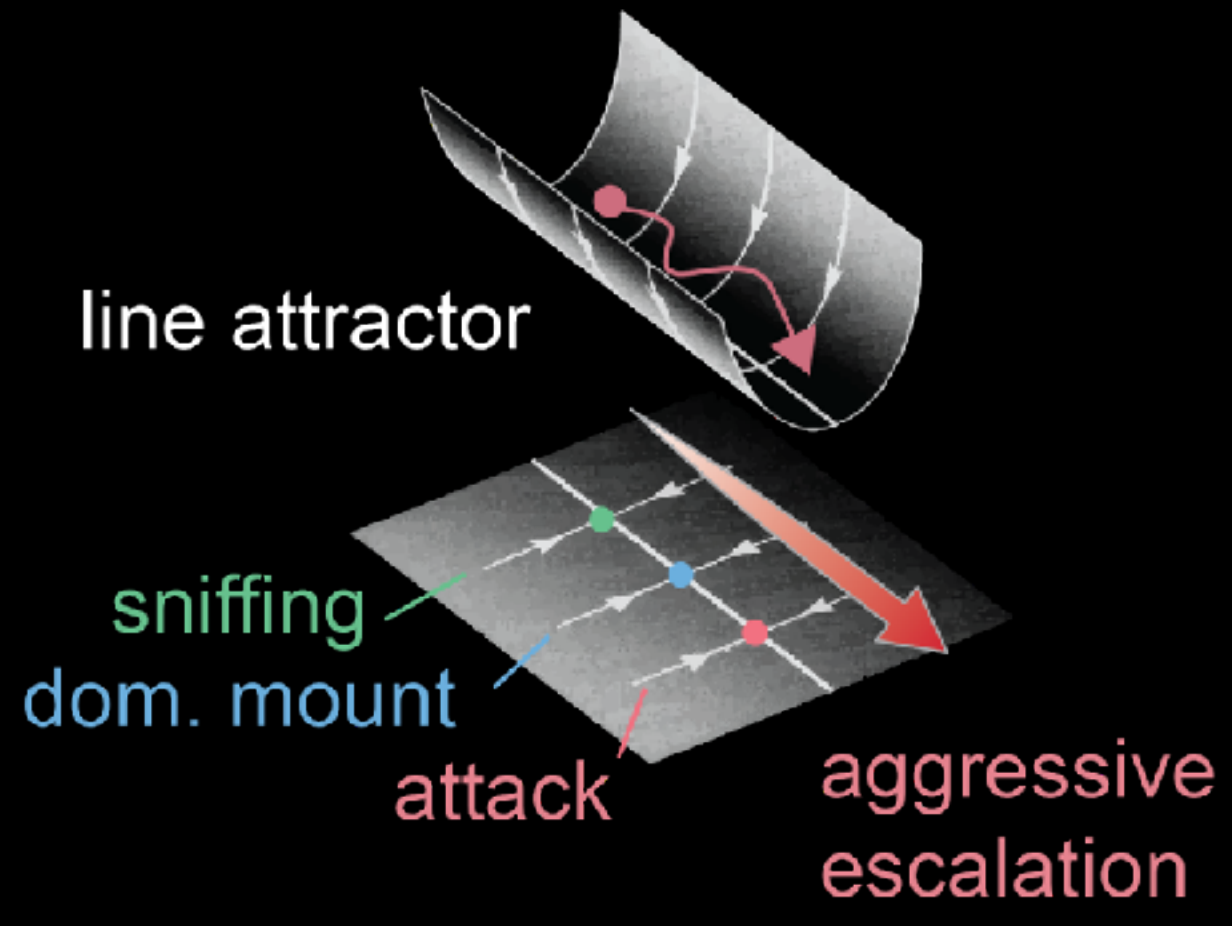


the stability of the attractor is enhanced in mice that are more aggressive

Nair et al. (Cell, 2023)

Are these dynamics intrinsic to VMHvl or a read-out of an upstream region?

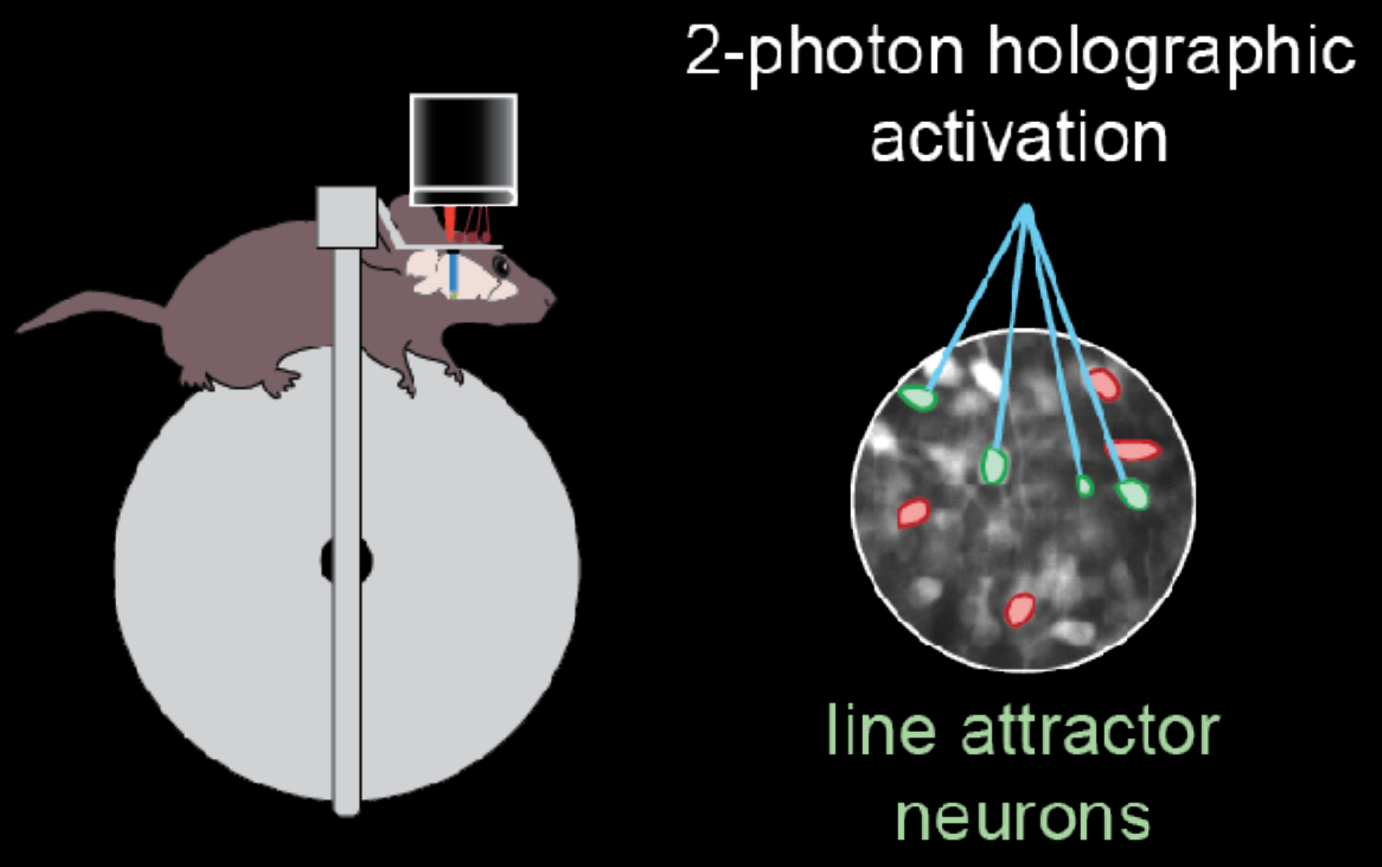
No study has causally demonstrated the existence of <u>intrinsic</u> line attractor dynamics in mammals.



Amit Vinograd

Vinograd, Nair et al. (Nature, 2024)

How can we gain access to the line attractor for perturbation?

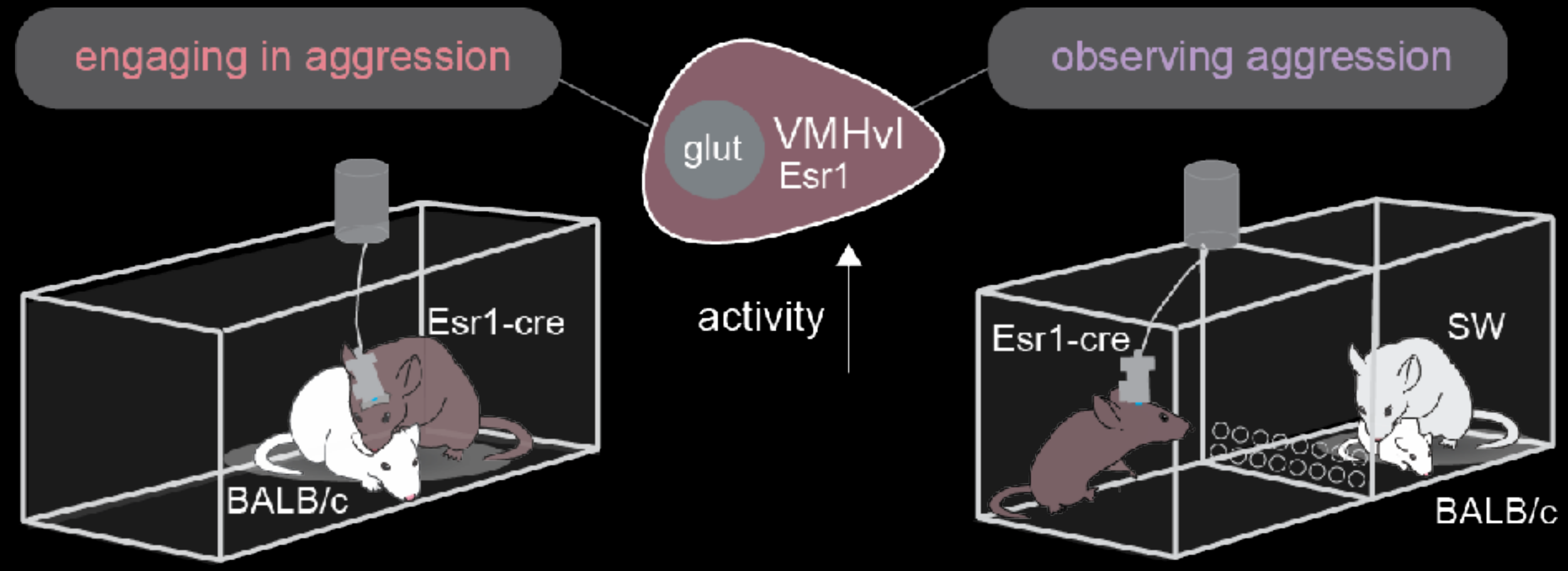


"dream experiment"

Unfortunately, head-fixation results in loss of attack behavior.

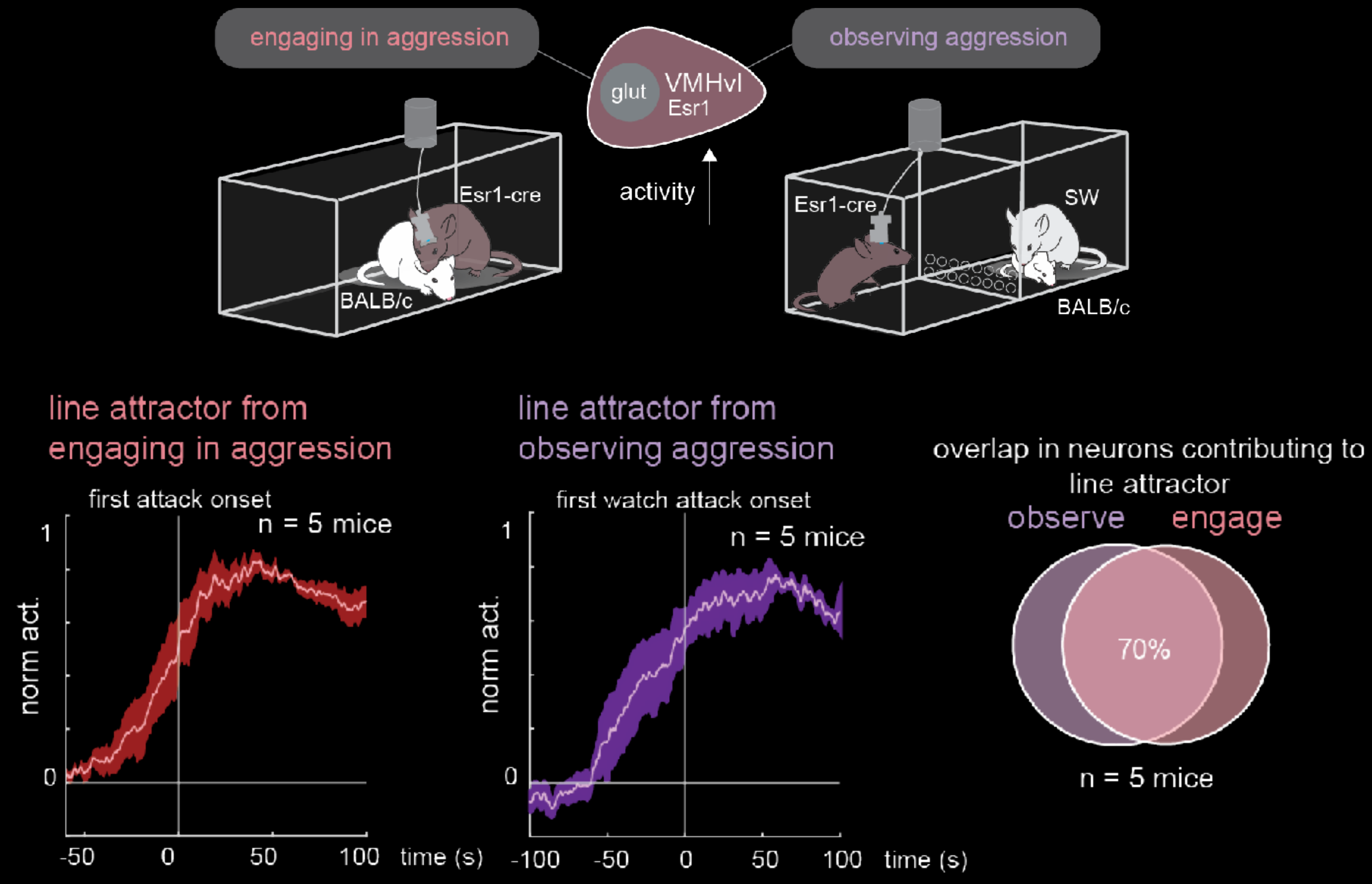
Vinograd, Nair et al. (Nature, 2024)

How can we gain access to the line attractor for perturbation?



VMHvl-Esr1 neurons are also active during observation of aggression
(Yang et al., Cell 2023)

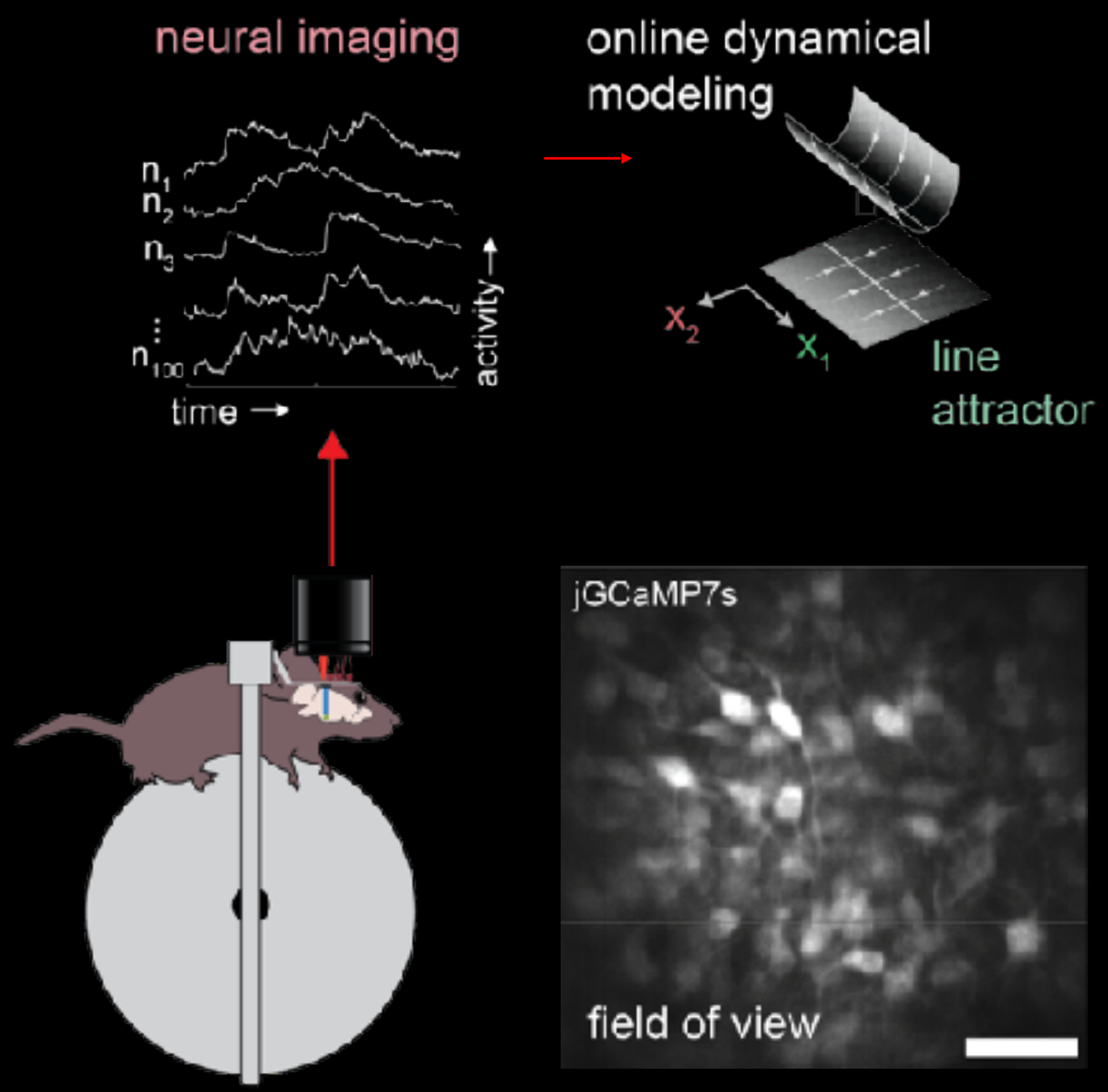
How can we gain access to the line attractor for perturbation?

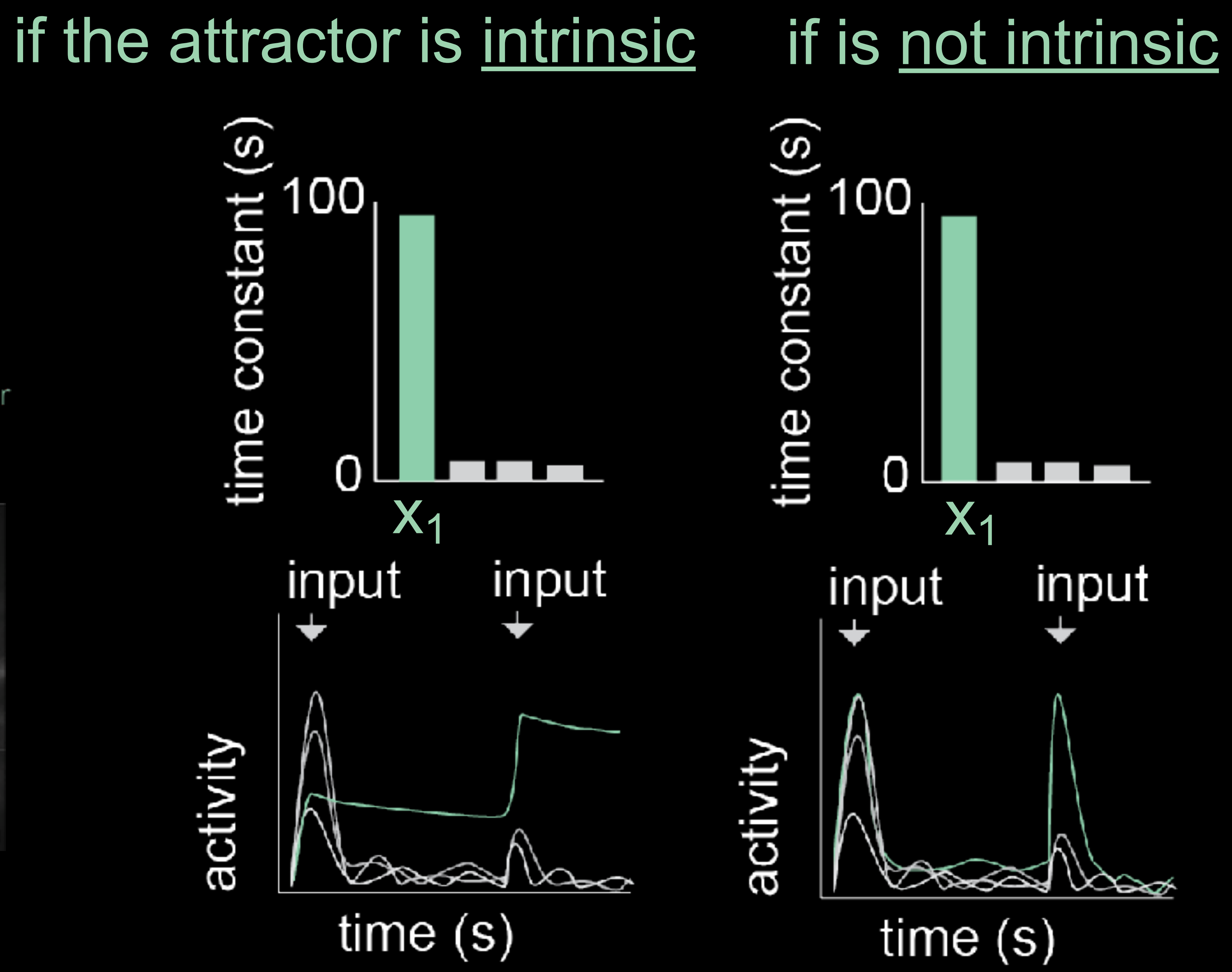


VMHvl-Esr1 neurons are show line attractor dynamics during observation of aggression

Vinograd, Nair et al. (Nature, 2024)

Closed-loop perturbation of dynamics in VMHvl

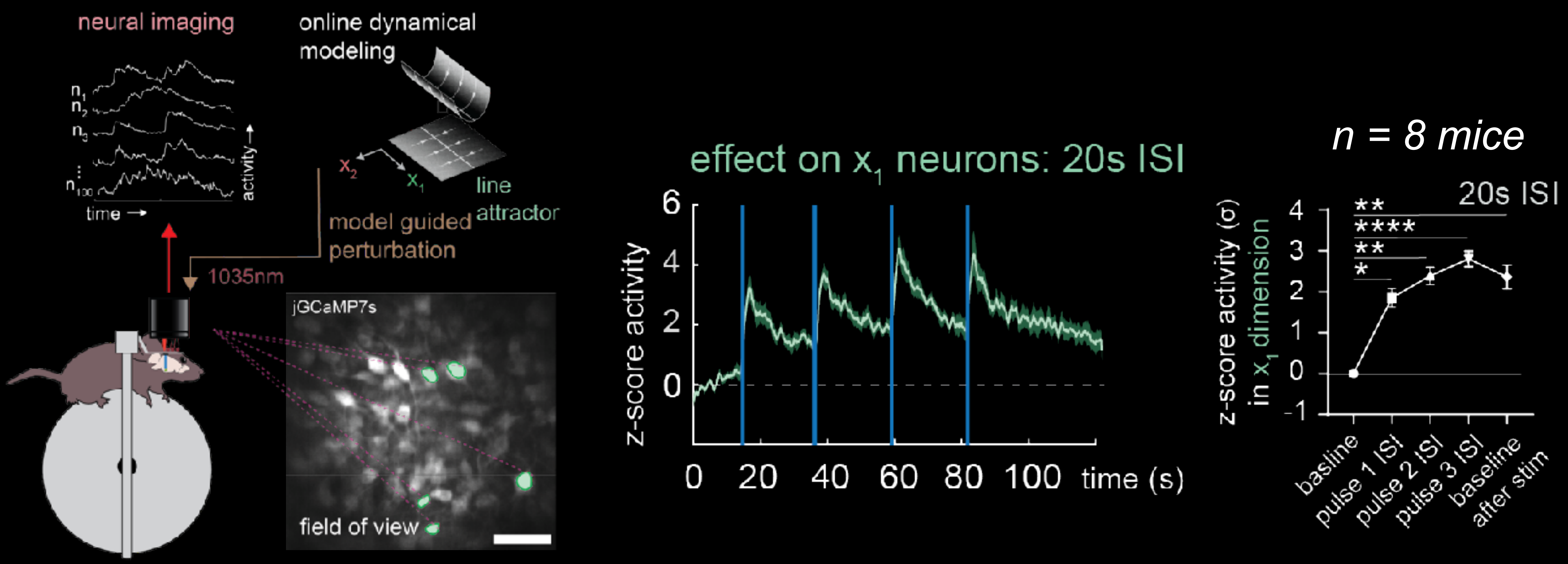




Activation of x_1 neurons should lead to integration if the line attractor is intrinsic.

Vinograd, Nair et al. (Nature, 2024)

Closed-loop perturbation of dynamics in VMHvl

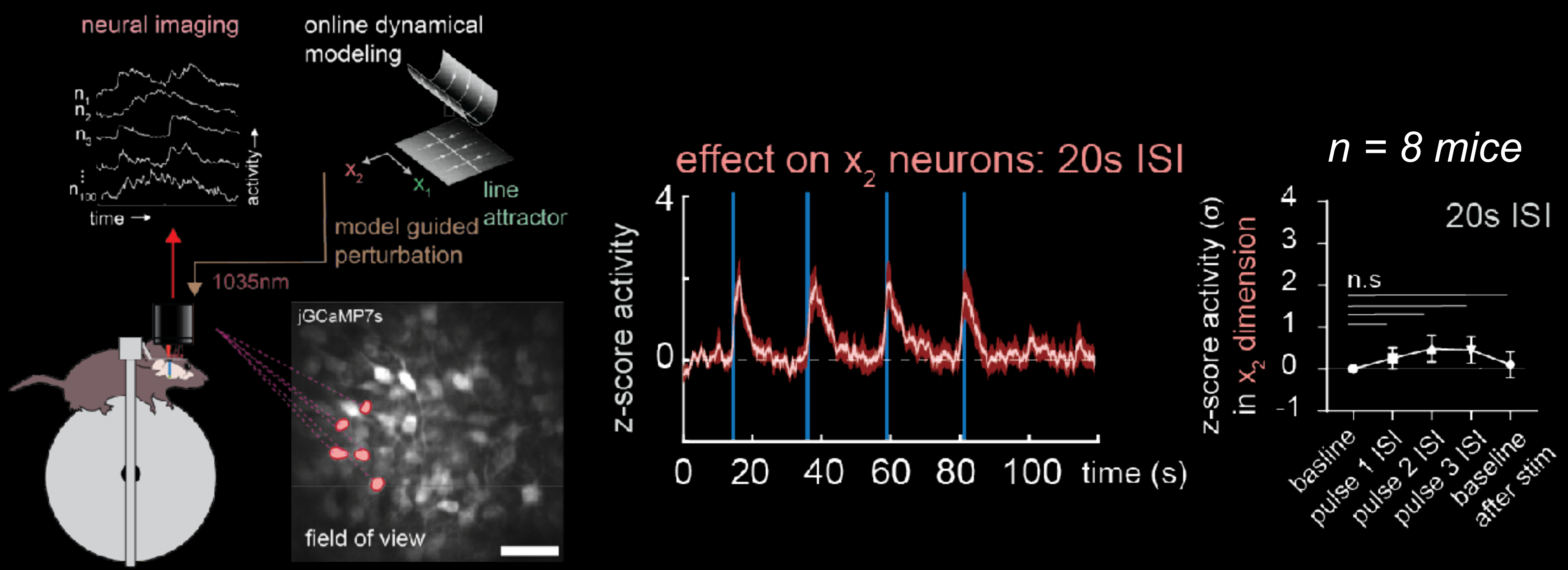


Holographic on-manifold activation* of line-attractor aligned x_1 neurons leads to integration.

**Note: this requires fitting an rSLDS online, during the session, to design the perturbation.*

Vinograd, Nair et al. (Nature, 2024)

Closed-loop perturbation of dynamics in VMHvl

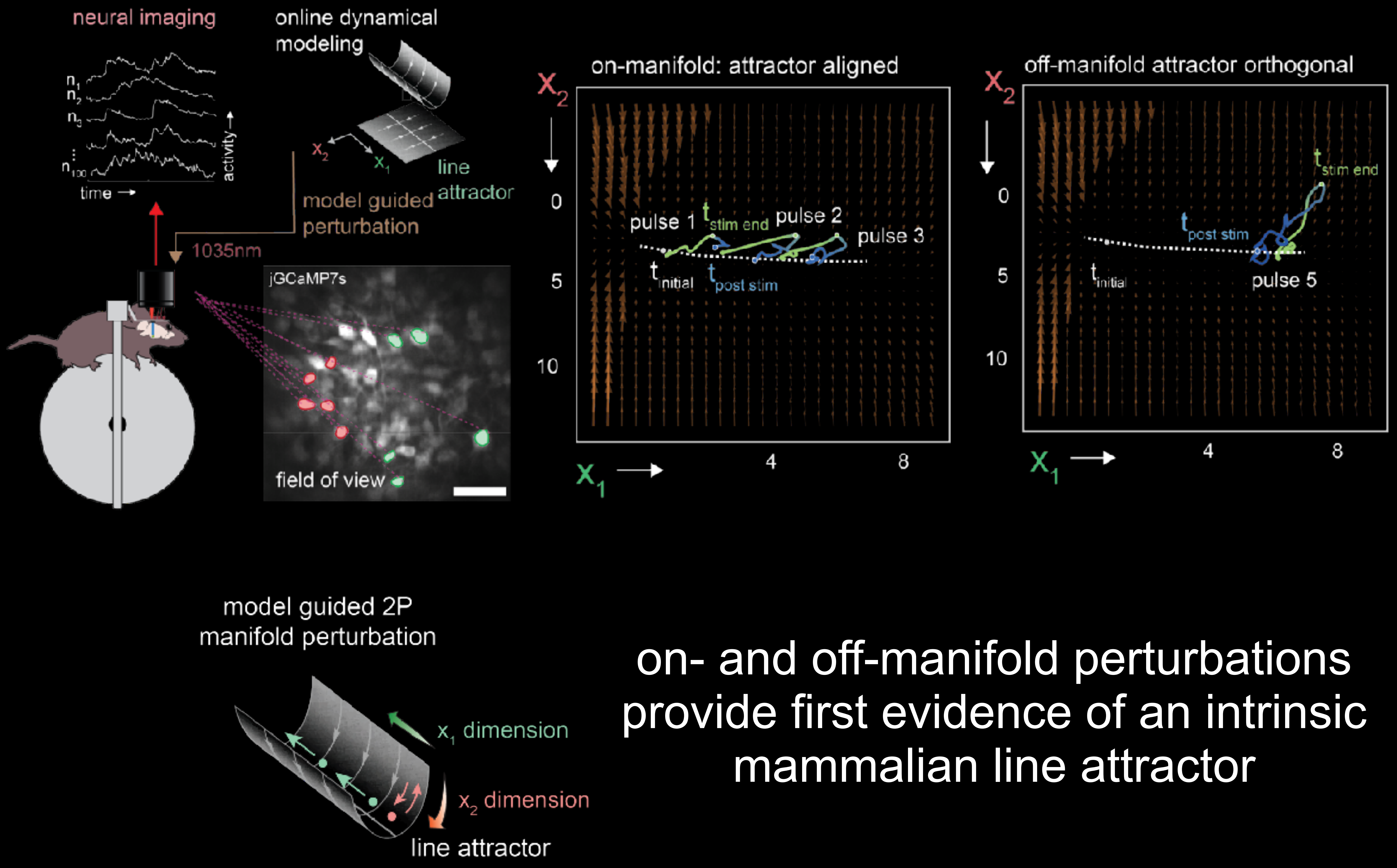


Holographic off-manifold activation of line-orthogonal x_2 neurons <u>does not</u> lead to integration.

Note: this requires fitting an rSLDS online, during the session, to design the perturbation.

Vinograd, Nair et al. (Nature, 2024)

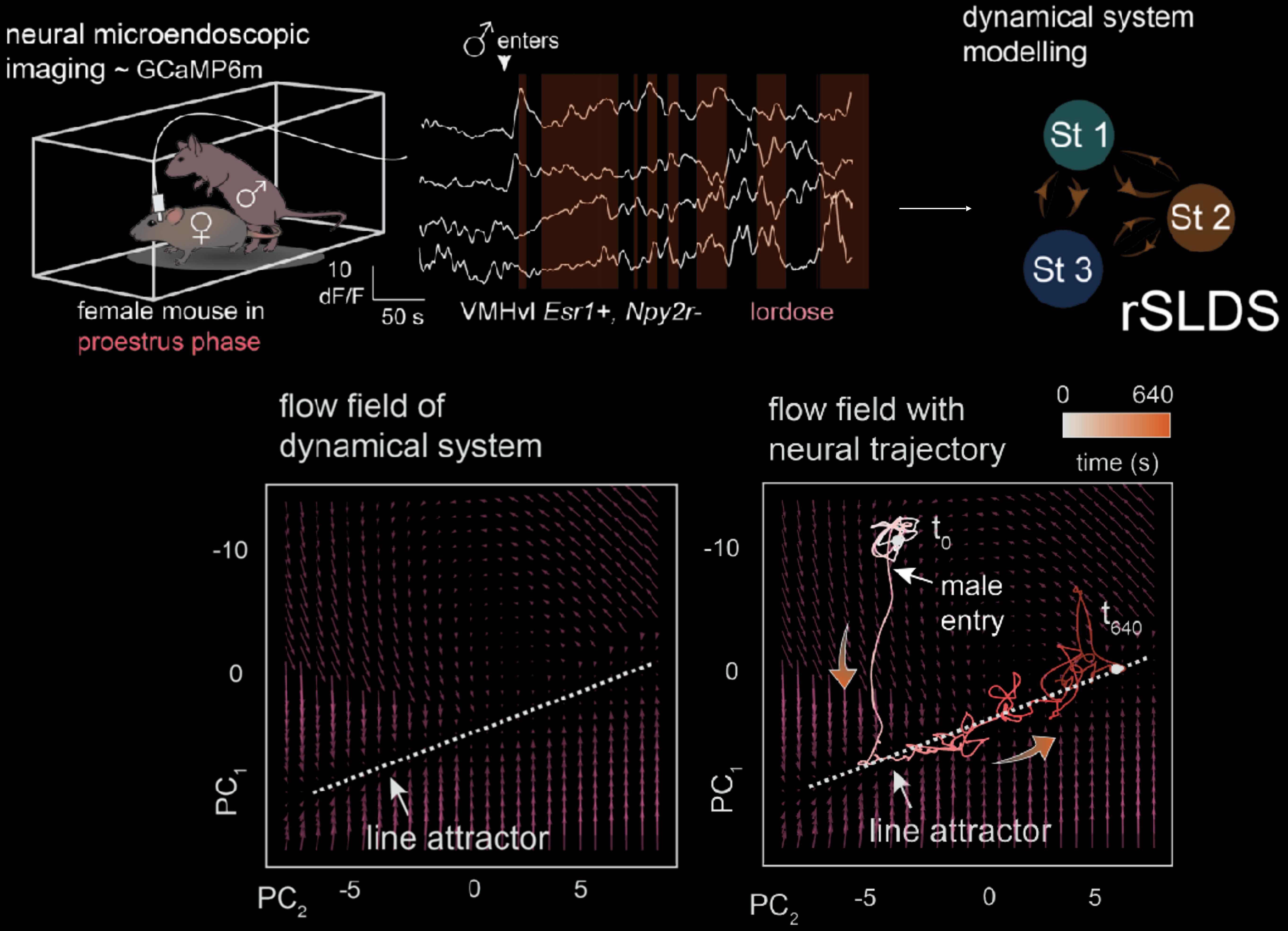
Closed-loop perturbation of dynamics in VMHvl



on- and off-manifold perturbations provide first evidence of an intrinsic mammalian line attractor

Vinograd, Nair et al. (Nature, 2024)

Do these attractor dynamics generalize to other internal state computations?

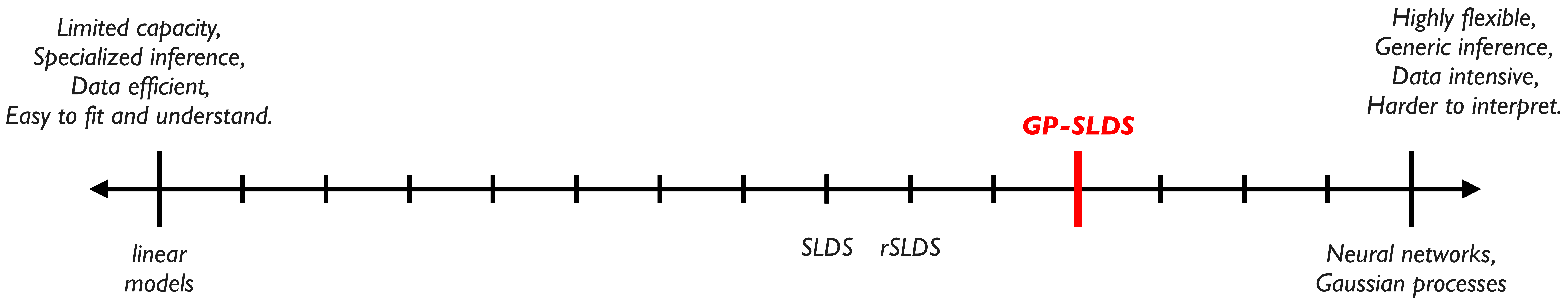


VMHvl shows attractor dynamics in female mice during mating, but only in proestrus.

Liu, Nair et al. (Nature, 2024)

This collaboration has inspired new methodological work to address limitations of rSLDS

Key idea: parameterize $f(x)$ as a Gaussian process with a novel kernel to produce smoothly switching linear dynamics.



Amber Hu

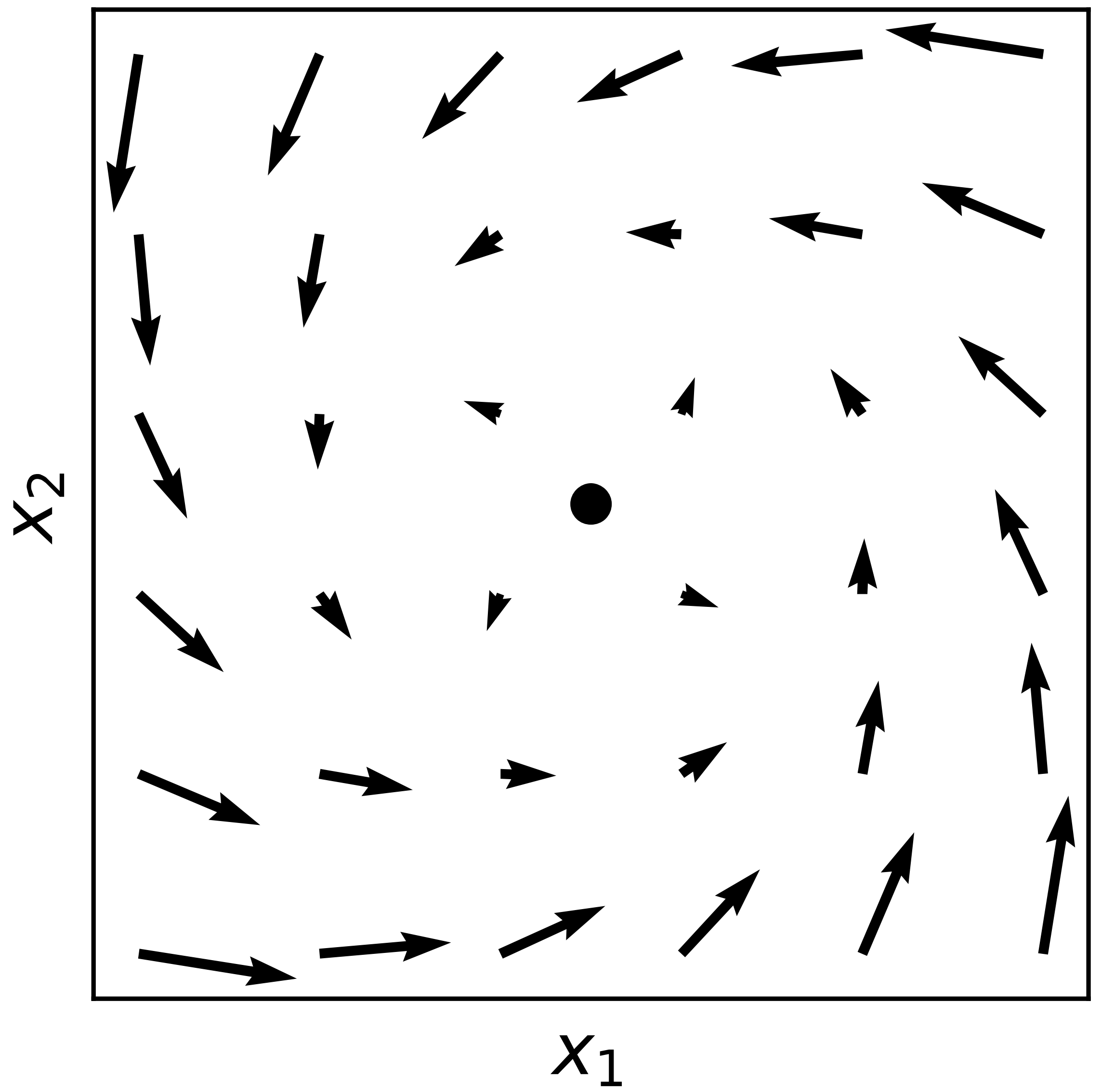
David Zoltowski

Lea Duncker

Hu et al. (NeurIPS, 2024)

Gaussian Process Stochastic Differential Equation (GP-SDE) Model

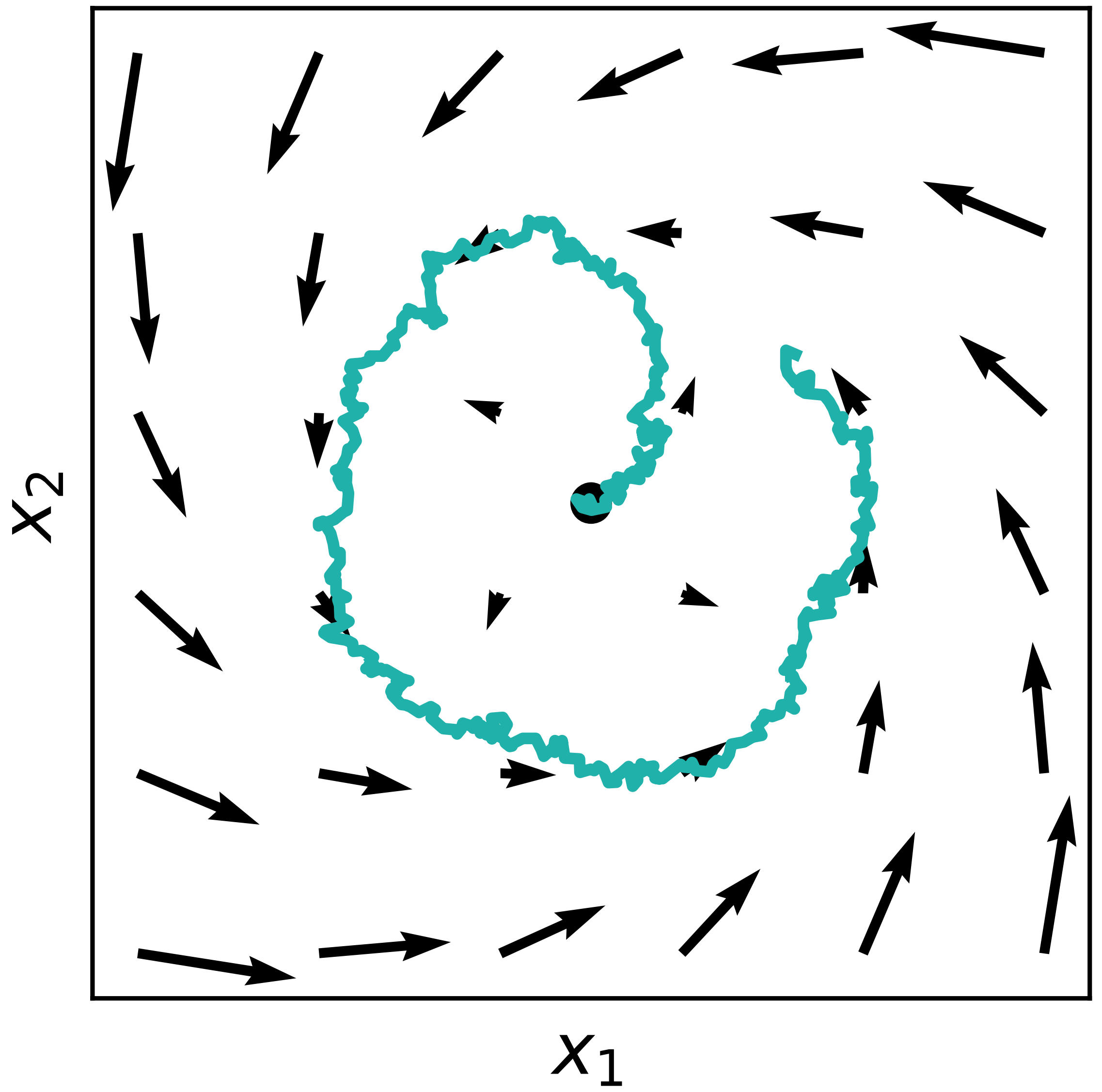
GP dynamics flow field



$$f \sim \mathrm{GP}(0, K_{ss}(\cdot, \cdot))$$

We propose a novel GP kernel to produce smoothly switching linear dynamics

latent SDE trajectory

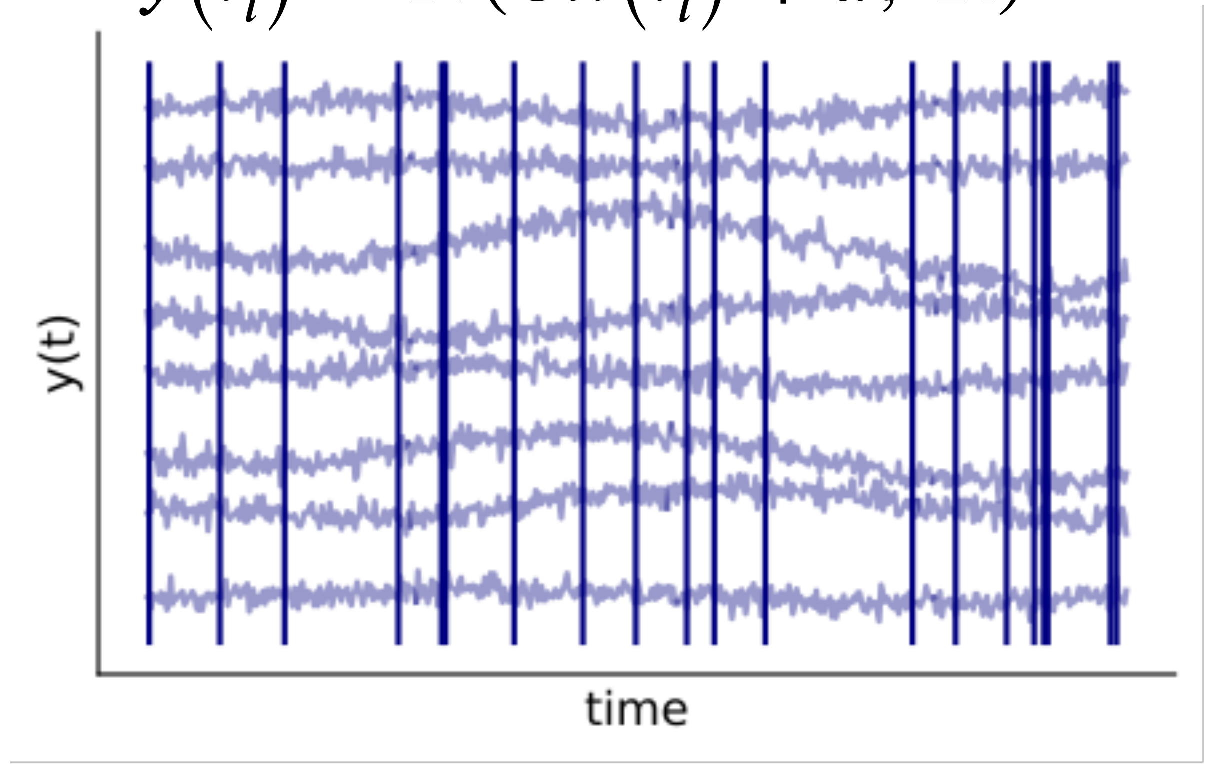


$$d\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x})dt + \sqrt{\boldsymbol{\Sigma}}\mathrm{d}\mathbf{w}$$

affine mapping to high-d space

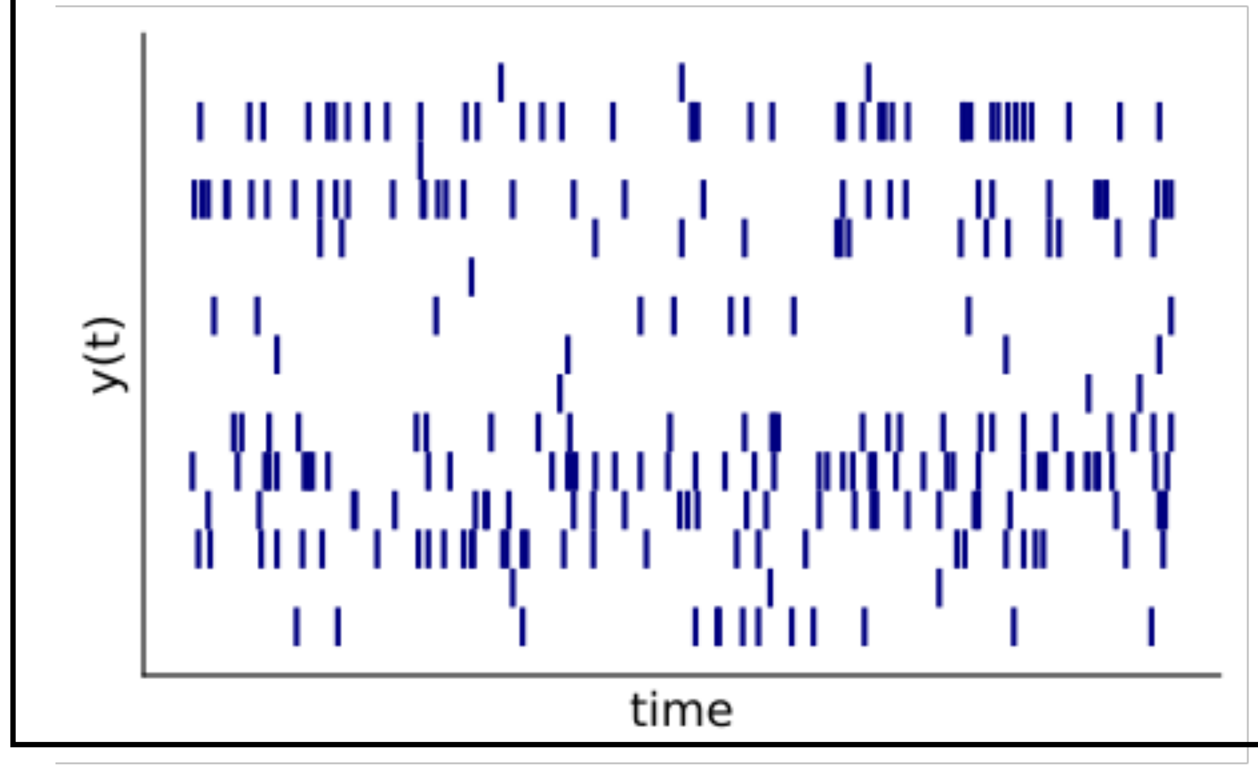
irregularly sampled Gaussian observations (e.g. calcium imaging)

$$y(t_i) \sim N(Cx(t_i) + d,\ R)$$



point process observations (e.g. spiking data)

$$y(t) \sim PP(\exp(Cx(t) + d))$$



Frigola et al. (NIPS, 2014)
Duncker et al. (ICML, 2019)
Course and Nair (Nature, 2023)

A Gaussian Process prior for **linear** functions

Consider the following model for **random linear functions** $f : \mathbb{R}^d \mapsto \mathbb{R}$**:**

$$f(\mathbf{x}) = \mathbf{w}^\top(\mathbf{x} - \mathbf{c})$$ intercept

$$\mathbf{w} \sim \mathrm{N}(\mathbf{0}, \mathbf{M})$$ slope covariance

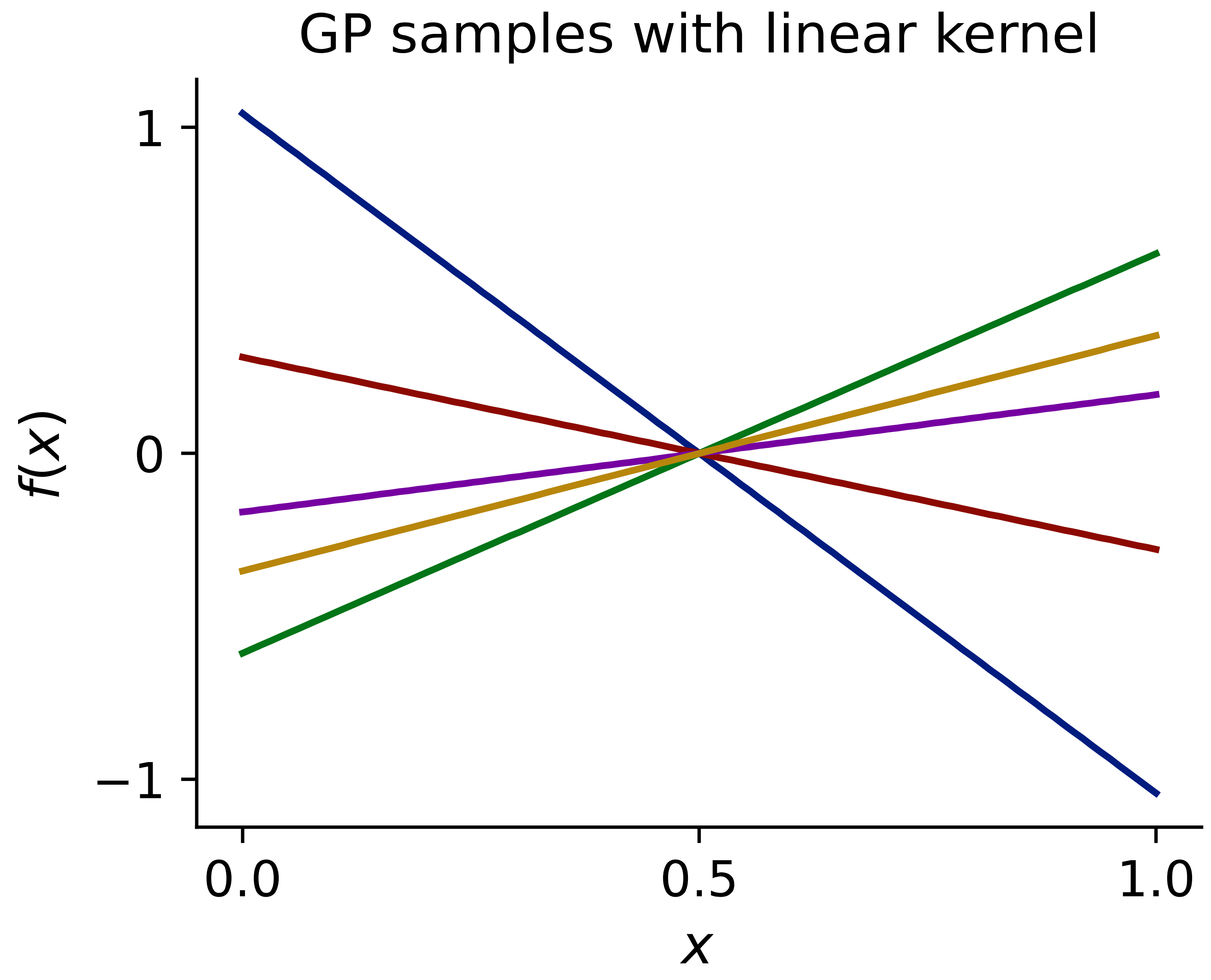
Let $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^\top$. Marginally,

$$\mathbf{f} \sim \mathrm{N}(\mathbf{0}, \mathbf{K})$$

$$\mathbf{K} = \mathbf{\Phi}\mathbf{M}\mathbf{\Phi}^\top \qquad \mathbf{\Phi} = \begin{bmatrix} \mathbf{x}_1 - \mathbf{c} \\ \vdots \\ \mathbf{x}_n - \mathbf{c} \end{bmatrix}$$

This is a **Gaussian process** with a **linear kernel!**

$$K_{\text{lin}}(\mathbf{x}, \mathbf{x}') = (\mathbf{x} - \mathbf{c})^\top \mathbf{M}(\mathbf{x}' - \mathbf{c})$$



Kernel hyperparameters:
$c = 0.5, \; M = 1$

A Gaussian process model for **linear <u>dynamics</u> functions**

We can use GPs to define a prior on **linear dynamics functions,** $f : \mathbb{R}^d \mapsto \mathbb{R}^d$

(Note: now it has a d-dimensional output.)

Approach: model each output dimension as an independent GP,

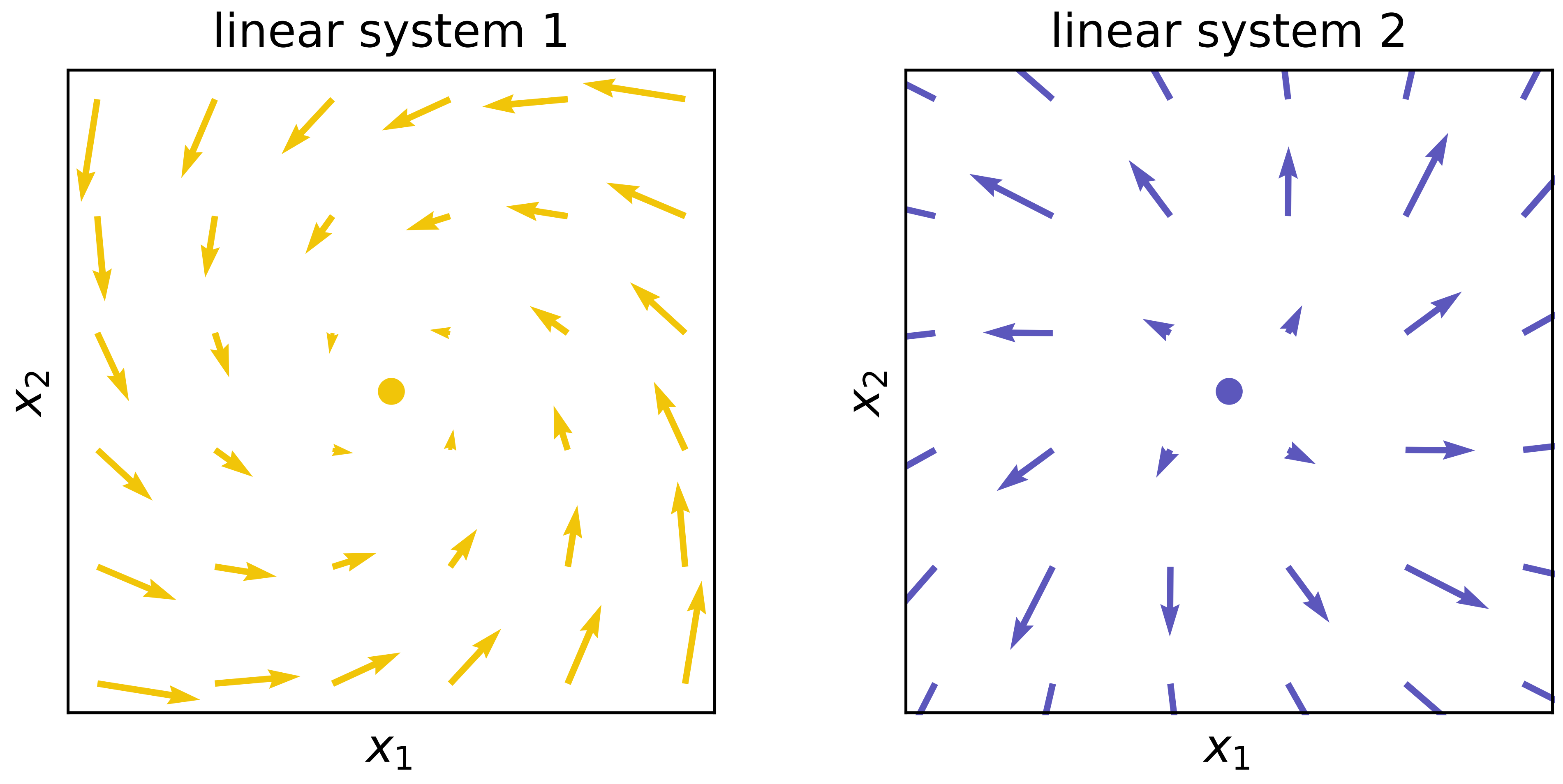
$$f_j \overset{\text{iid}}{\sim} \mathrm{GP}(0, K_{\text{lin}}(\cdot, \cdot)) \quad \text{for } j = 1, \ldots, d$$

Assume the GPs share the same kernel hyper-parameters. Then,

$$f(\mathbf{c}) = \mathbf{0} \quad \text{(a.s.)}$$

I.e., the intercept hyper-parameter defines the **fixed point** of the dynamics function.

Two samples of random linear dynamics functions:



A Gaussian process model for **piecewise constant functions**

Let $(\mathcal{A}_1, \ldots, \mathcal{A}_K)$ be a **partition** of $\mathbb{R}^d$.

Define a **one-hot** feature vector,

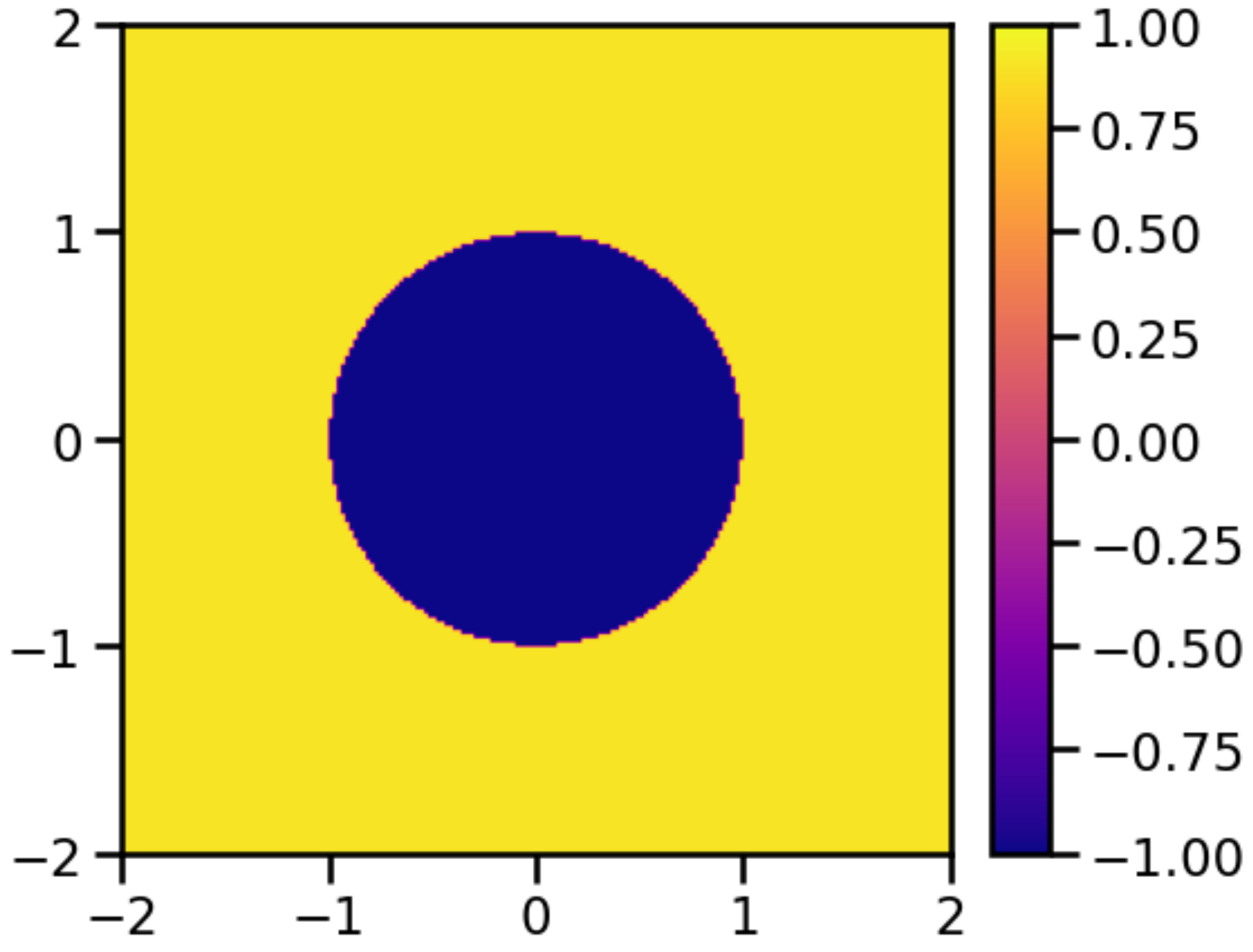
$$\pi(\mathbf{x}) = (\mathbb{I}[\mathbf{x} \in \mathcal{A}_1], \ldots, \mathbb{I}[\mathbf{x} \in \mathcal{A}_K])^\top$$

The inner product defines of features defines a **kernel**,

$$K_\pi(\mathbf{x}, \mathbf{x}') = \pi(\mathbf{x})^\top \pi(\mathbf{x}')$$

Samples $f \sim \mathrm{GP}(0, K_\pi(\cdot, \cdot))$ yield **piecewise constant functions.**

Example draw from GP with p.c. kernel



$$\mathcal{A}_1 : \left\{ \mathbf{x} \in \mathbb{R}^2 : x_1^2 + x_2^2 < 1 \right\}$$

A Gaussian process that **smoothly interpolates** between piecewise constant functions

Let $(\mathcal{A}_1, \ldots, \mathcal{A}_K)$ be a **partition** of $\mathbb{R}^d$.

Define a **one-hot** feature vector,

$$\pi(\mathbf{x}) = (\mathbb{I}[\mathbf{x} \in \mathcal{A}_1], \ldots, \mathbb{I}[\mathbf{x} \in \mathcal{A}_K])^\top$$

The inner product defines of features defines a **kernel**,

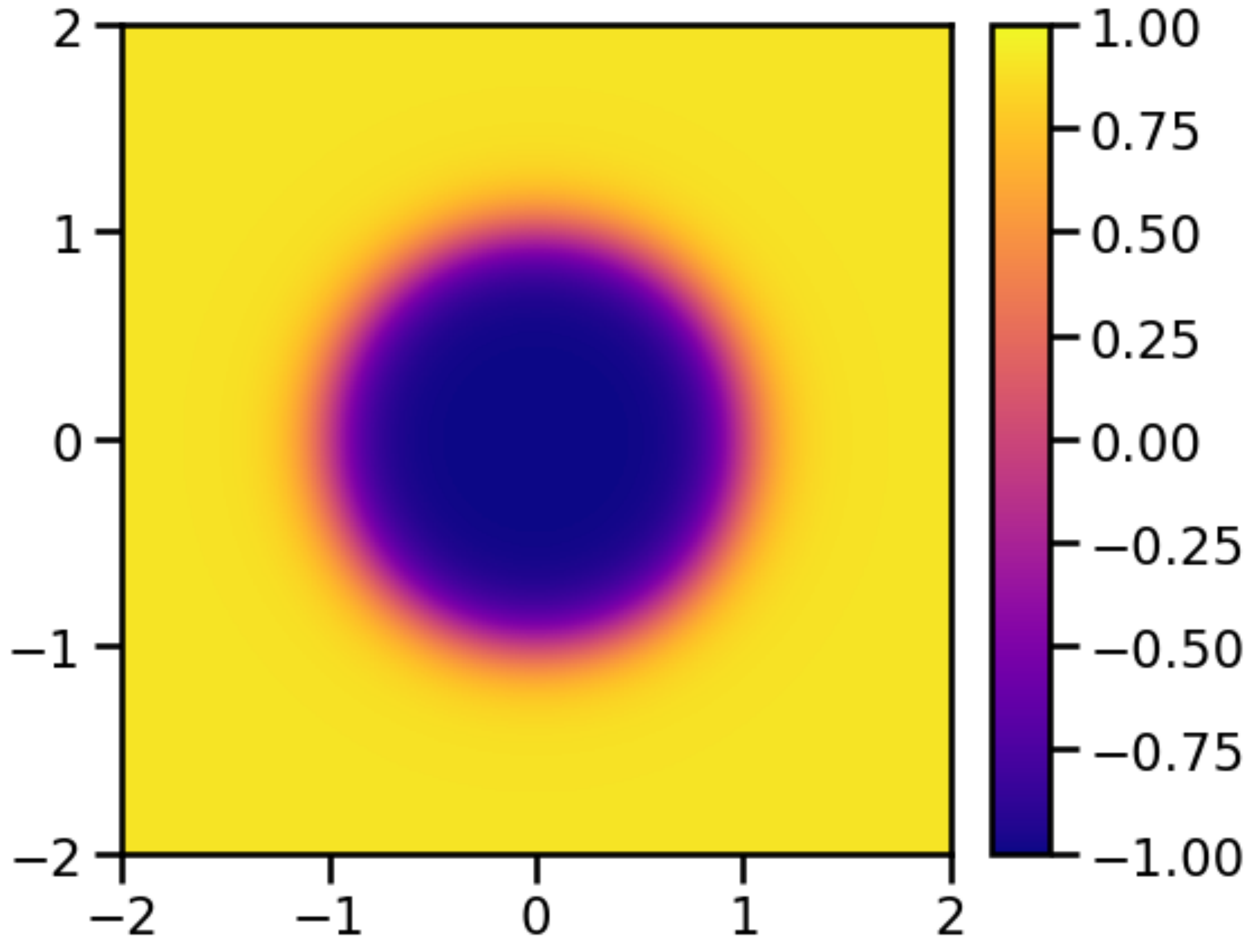
$$K_\pi(\mathbf{x}, \mathbf{x}') = \pi(\mathbf{x})^\top \pi(\mathbf{x}')$$

Samples $f \sim \mathrm{GP}(0, K_\pi(\cdot, \cdot))$ yield **piecewise constant functions.**

More generally, suppose $\pi(\mathbf{x}) \in \Delta_{K-1}$ varies smoothly.

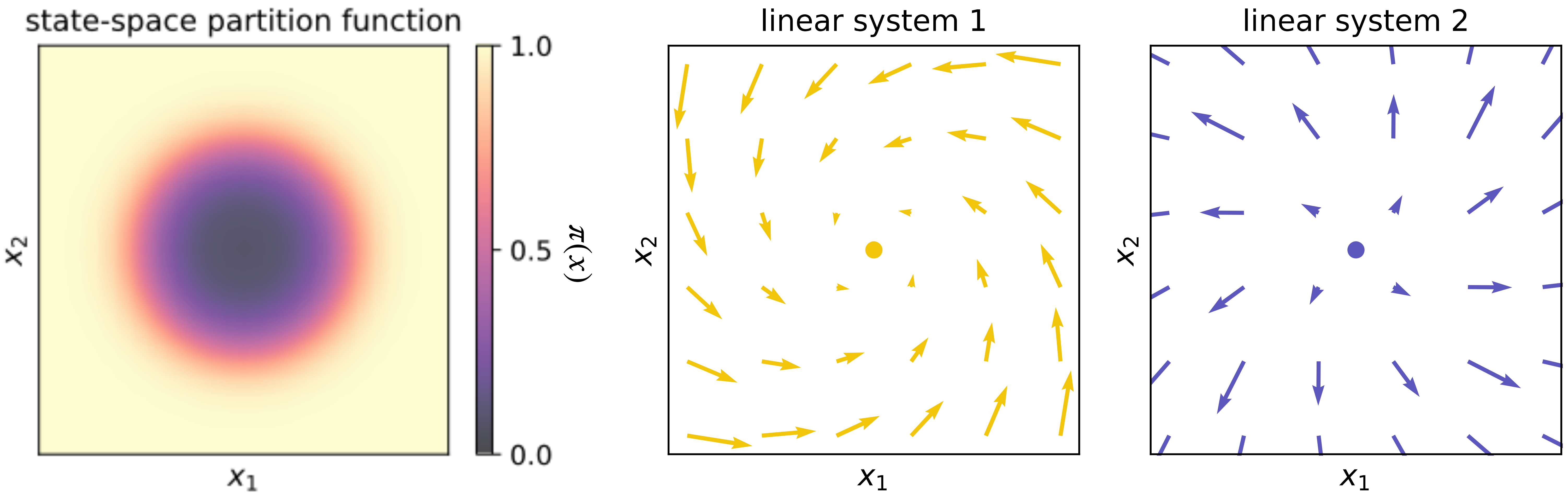
Then samples from the GP **smoothly interpolate** between **piecewise constant** functions.

Example draw from GP with p.c. kernel



$$\pi_1(\mathbf{x}) = \sigma((x_1^2 + x_2^2 - 1)/\tau)$$

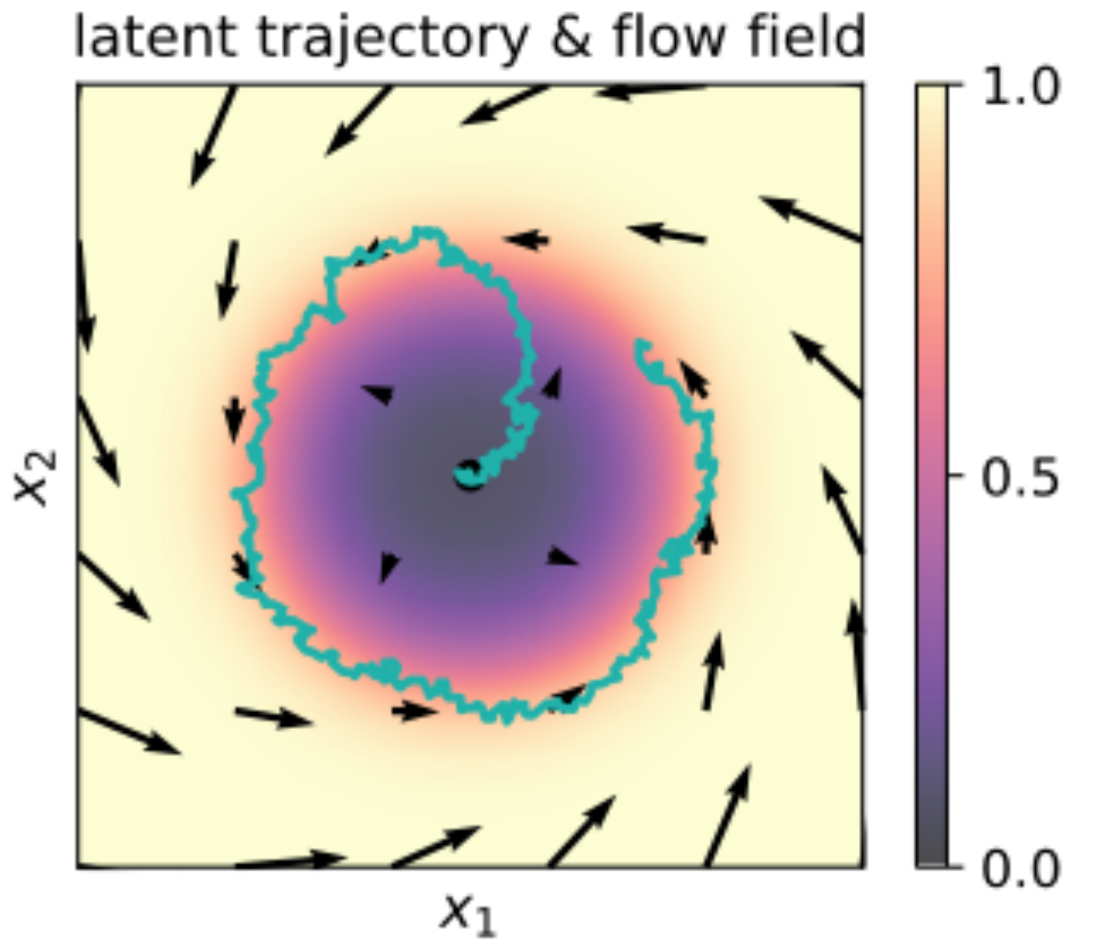
A Gaussian process kernel for **smoothly switching linear dynamics**



$\pi(x)$ $\quad K_{\text{lin}}(\cdot,\cdot)$ $\quad K_{\text{lin}}(\cdot,\cdot)$

$f_j \sim \mathrm{GP}(0, K_{\text{ss}}(\cdot,\cdot))$

$dx = f(x)dt + \sqrt{\Sigma}\mathrm{d}\mathbf{w}$



$$K_{\text{ss}}(\mathbf{x},\mathbf{x}') = \sum_{k=1}^{K} \pi_k(\mathbf{x})\pi_k(\mathbf{x}') K_{\text{lin}}^{(k)}(\mathbf{x},\mathbf{x}')$$

Probabilities that linear system k is active at x and x'

Kernel for linear system k

<u>Kernel has many nice features:</u>

✓ Piecewise linear components

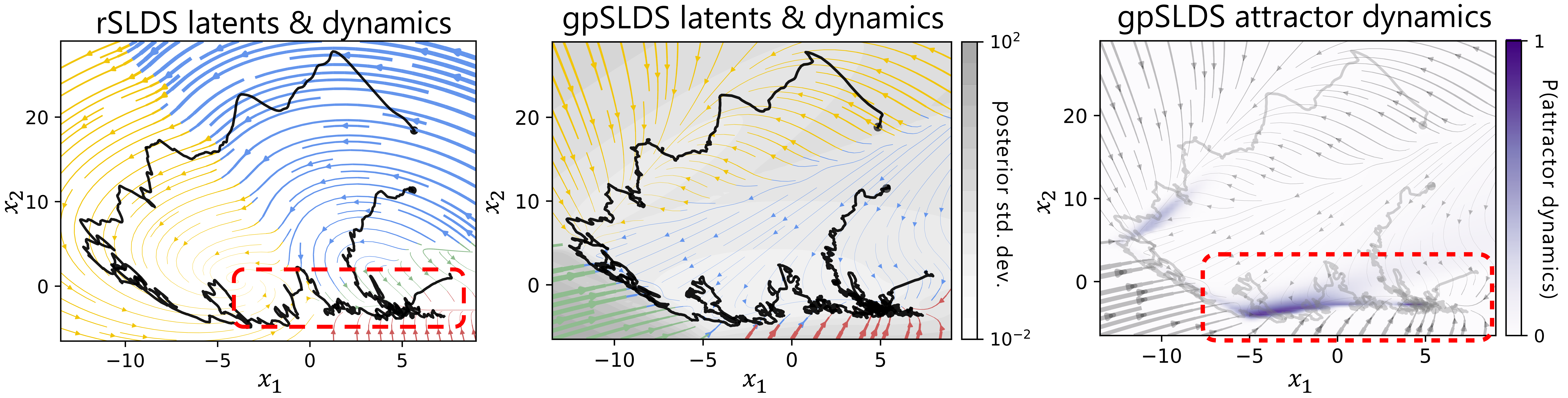
✓ $\pi(\cdot)$ can be a nonlinear function (polynomial logistic regression, neural network, etc.)

✓ Enforces smooth dynamics at boundaries

✓ Allows estimates of posterior uncertainty

Hu et al. (NeurIPS, 2024)

GP-SLDS provides uncertainty estimates for the dynamics and posterior probability of attractors.



Revisiting the VMHvl neural population activity from Nair et al. (2023) with a GP-SLDS, now we can quantify uncertainty in the latent state dynamics estimates.

Hu et al. (NeurIPS, 2024)

Outline

1. *Natural Intelligence: Internal states and attractor dynamics in the hypothalamus*
   - *Question: How does the brain represent and maintain internal states?*
   - *Method: Recurrent switching linear dynamical systems (rSLDS)*
   - *Results: Intrinsic line attractor dynamics in the hypothalamus encode an aggressive internal state*
   - *Extension: Smoothly interpolating between states in an rSLDS*
2. *Artificial Intelligence: Deep state space models for sequence-to-sequence modeling*

Outline

1. Natural Intelligence: Internal states and attractor dynamics in the hypothalamus

2. Artificial Intelligence: Deep state space models for sequence-to-sequence modeling

- *<u>Question</u>: Can simple compositions of linear systems perform more complex computations?*
- *<u>Method</u>: Simple state space layers with parallel scans (S5)*
- *<u>Results</u>: Impressive performance on long-range sequence modeling tasks*
- *<u>Extension</u>: Towards scalable and stable parallelization of nonlinear RNNs*

Acknowledgements

Jimmy Smith

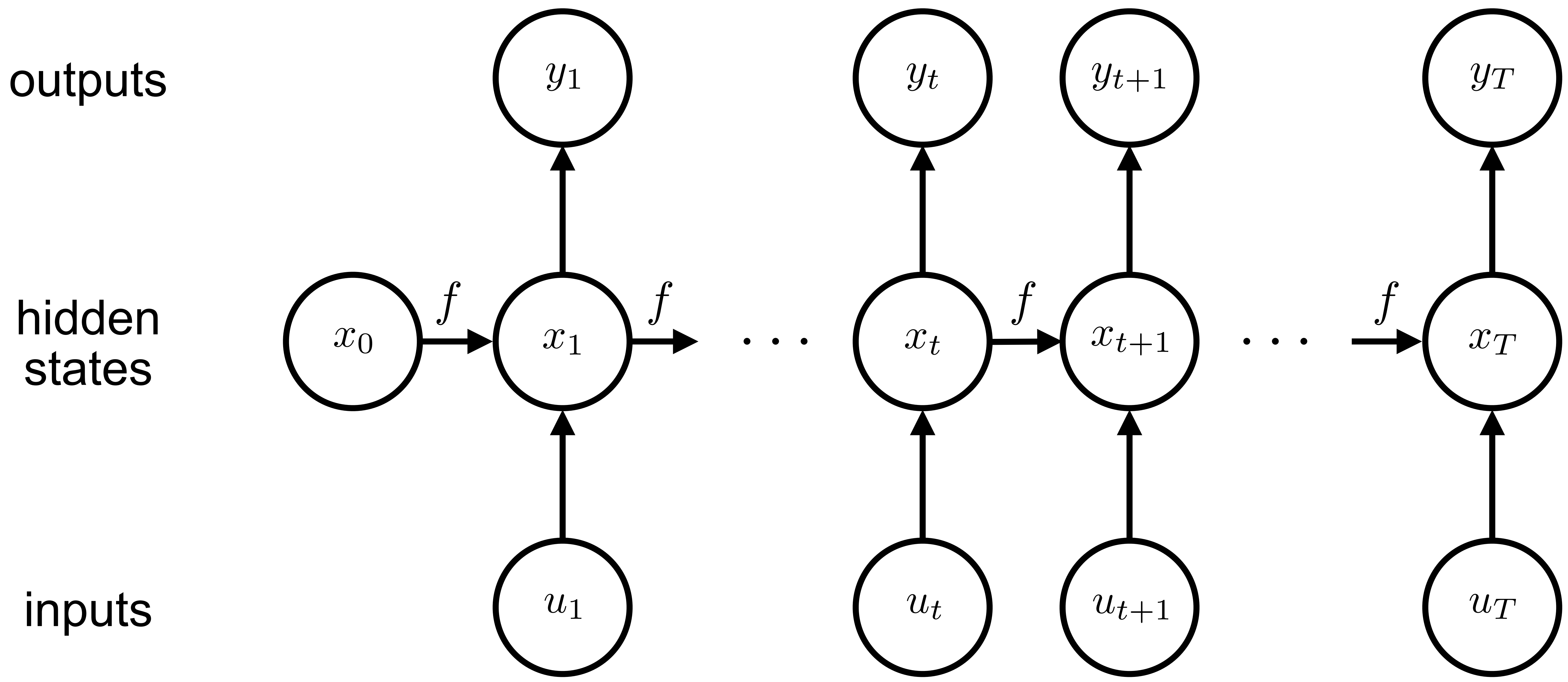
Andy Warrington

Xavier Gonzalez

Inspired by work of Albert Gu, Tri Dao, Chris Ré, and others.

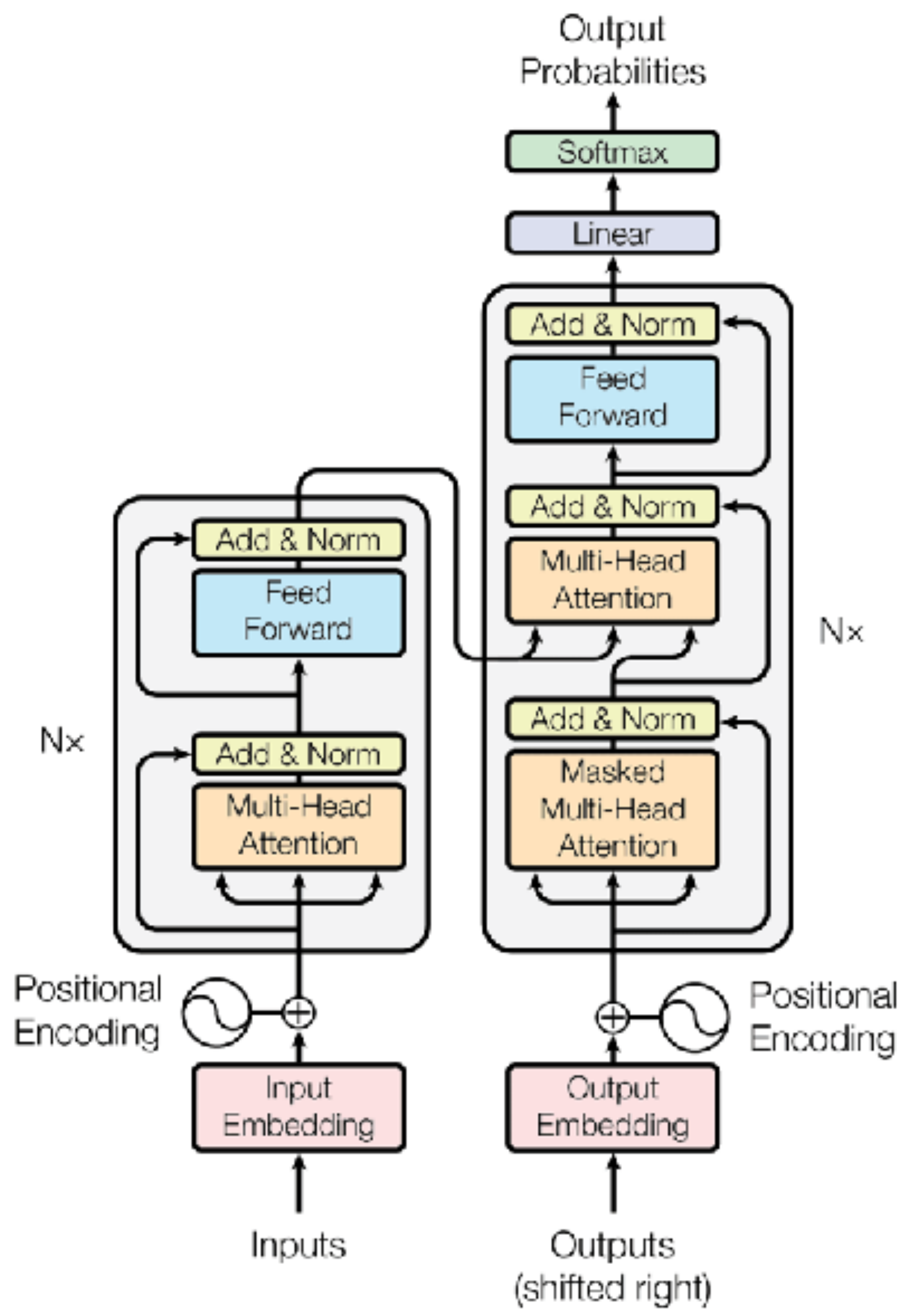
Recurrent neural networks

(Note: Everything is deterministic in this picture.)



RNNs allow **fast autoregressive generation**, but **evaluation ("inference") is inherently sequential.**

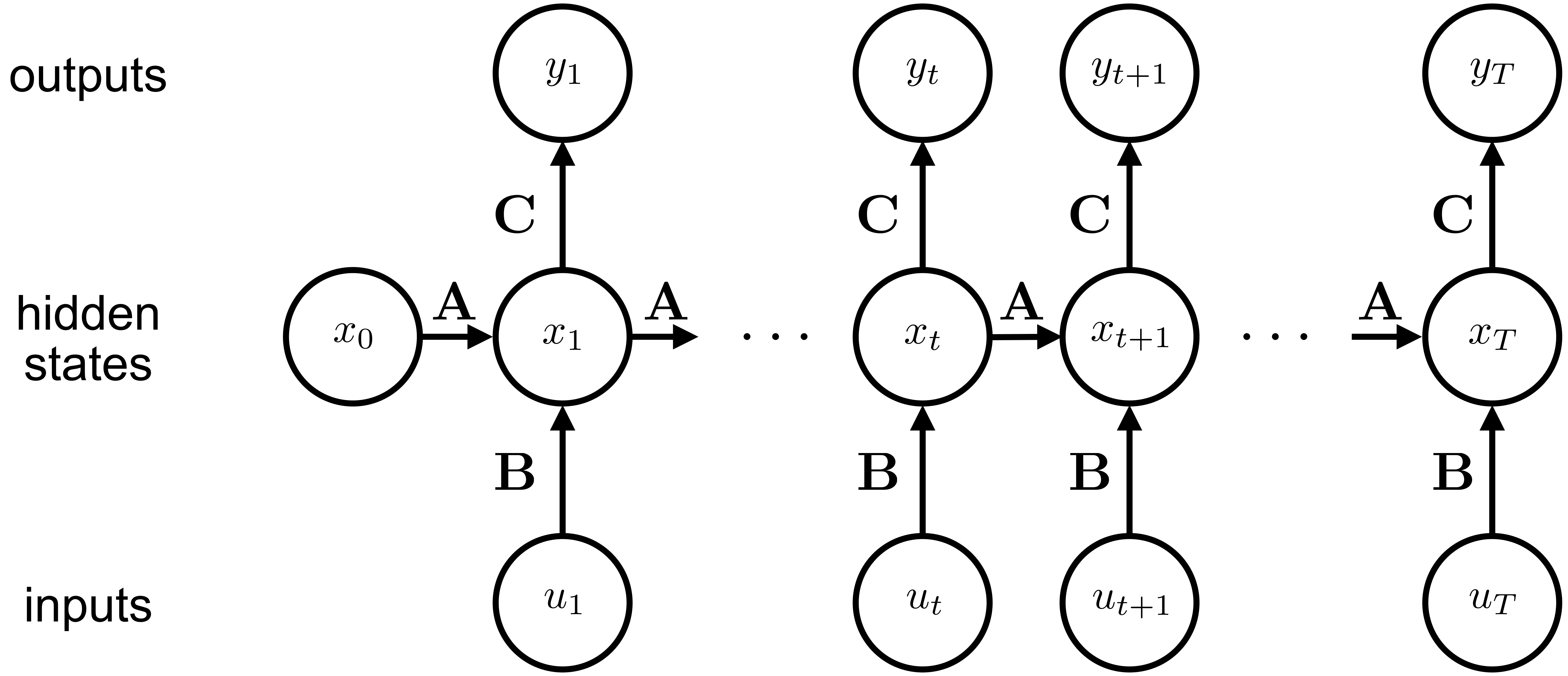
Transformers



Transformers allow **fast parallel evaluation — this** may be the biggest reason for their success.

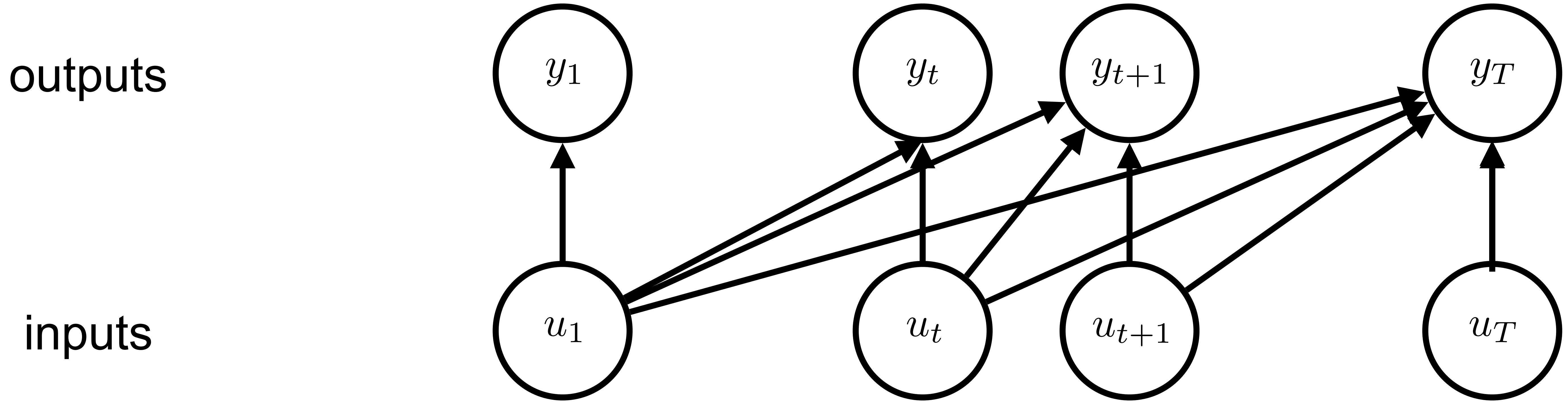
However, **autoregressive generation is costly.**

What if we restrict ourselves to **linear** recurrent neural networks?



$$x_t = \mathbf{A}x_{t-1} + \mathbf{B}u_t$$
$$y_t = \mathbf{C}x_t$$

Linear RNNs are equivalent to convolutions



$$y_t = \mathbf{C}\mathbf{A}^t\mathbf{B}u_0 + \mathbf{C}\mathbf{A}^{t-1}\mathbf{B}u_1 + \ldots \mathbf{C}\mathbf{A}\mathbf{B}u_{t-1} + \mathbf{C}\mathbf{B}u_t$$

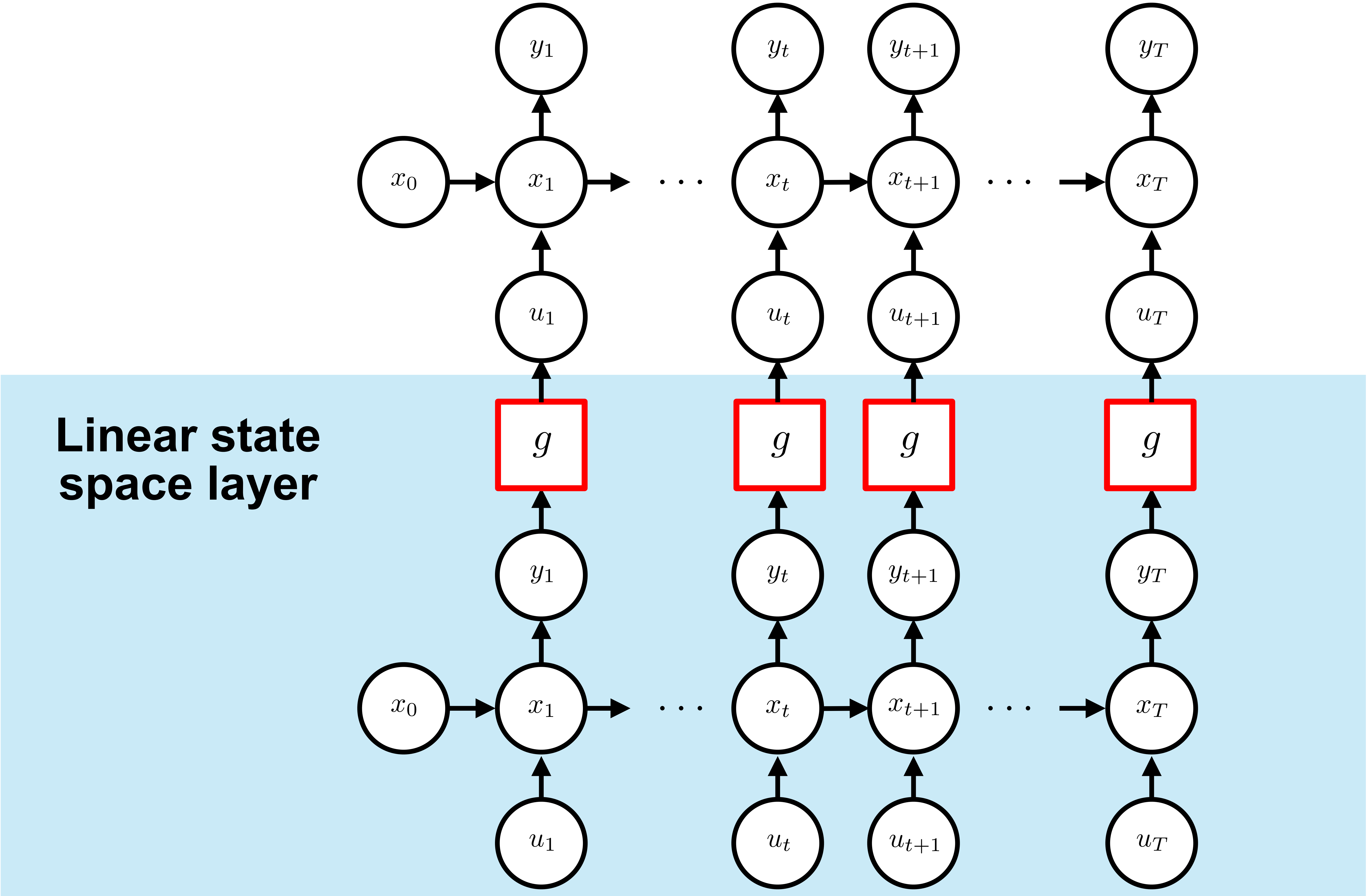
$$\Updownarrow$$

$$\mathbf{y} = \mathbf{K} \circledast \mathbf{u}$$

$$\mathbf{K} = \left(\mathbf{C}\mathbf{B}, \mathbf{C}\mathbf{A}\mathbf{B}, \ldots \mathbf{C}\mathbf{A}^{T-1}\mathbf{B}\right)$$

Linear RNNs allow **fast autoregressive generation** and **fast parallel evaluation** (via convolution)!

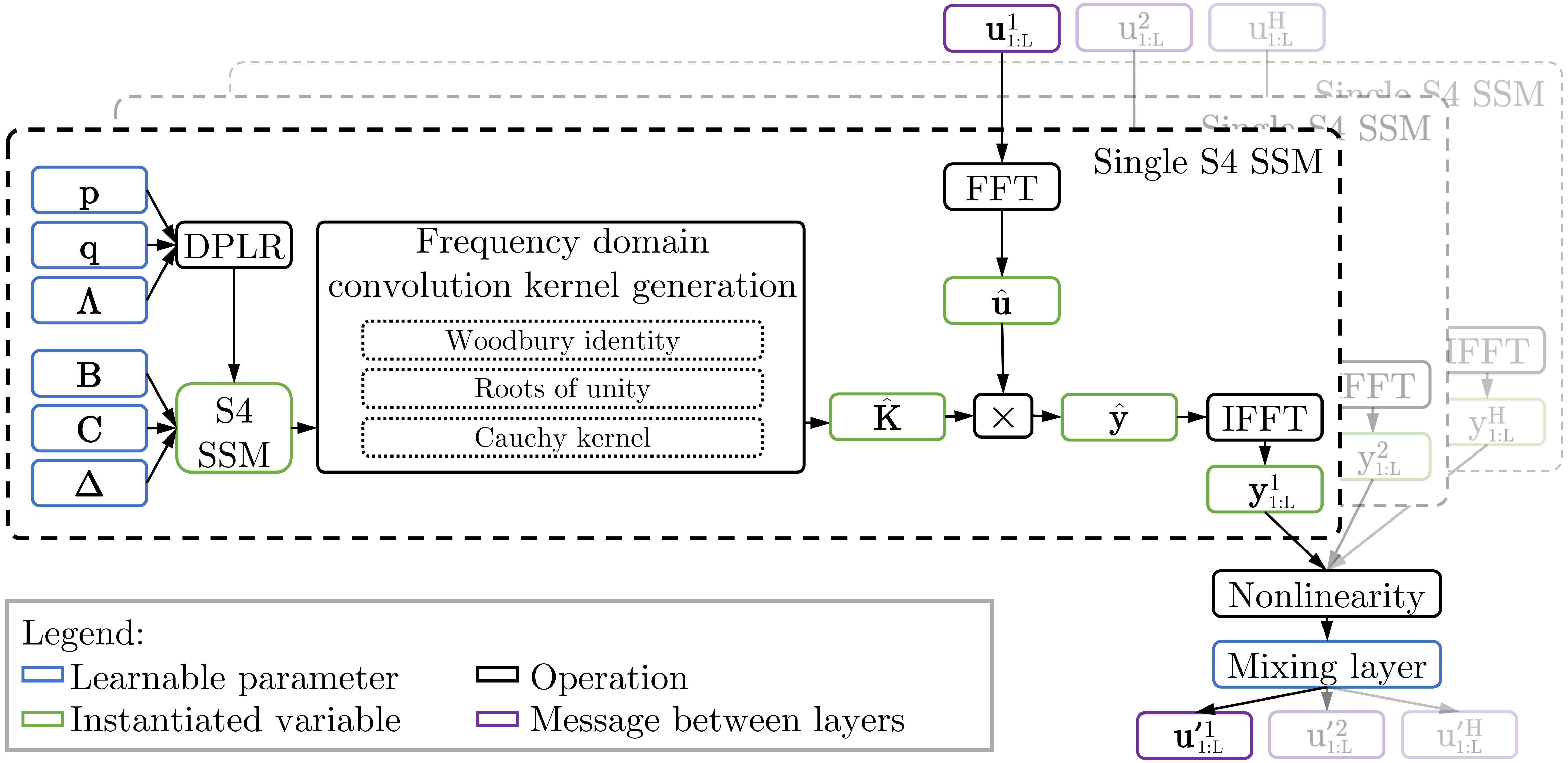
Gu et al (2021) proposed **S4**, which stacks linear state space layers with nonlinearities in between



Linear state space layers are linear in time but nonlinear in depth.

Gu et al. (2021)

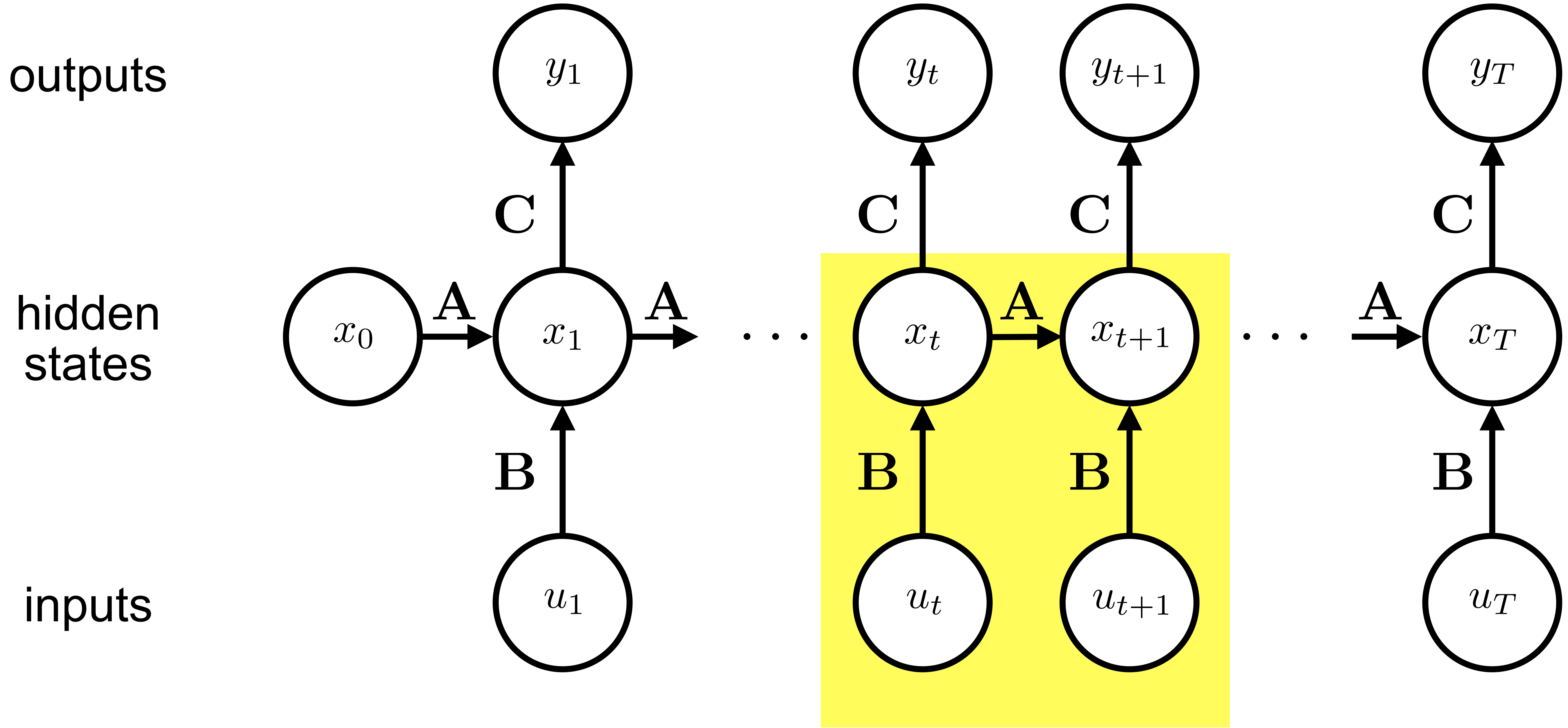
But the devil is in the details…



To efficiently compute the convolution, S4 used a bank of single-input single-output (SISO) filters and some sophisticated mathematical tricks.

Gu et al. (2021)

With **S5**, we aimed to simplify S4 by sticking in the time domain and using **parallel scans.**

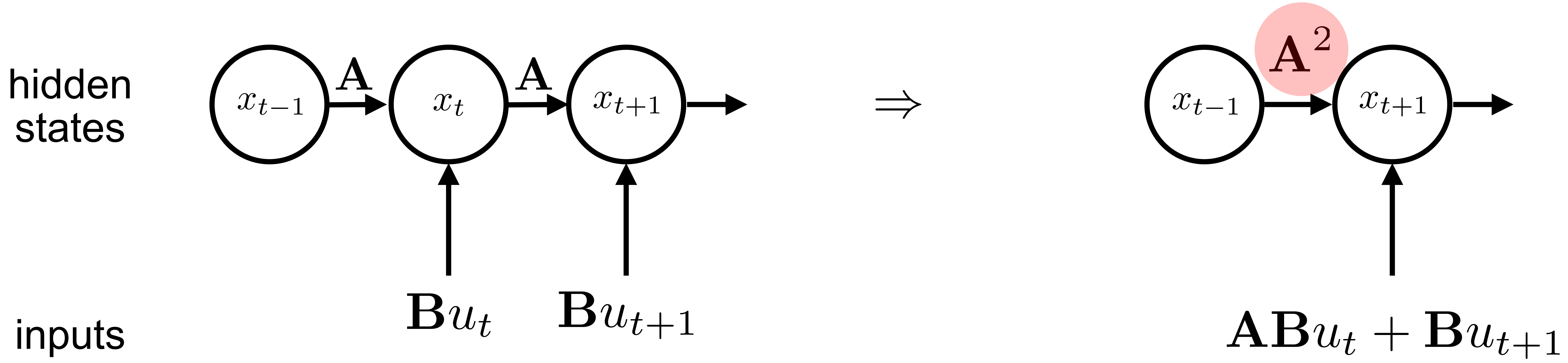


How can we parallelize the sequential evaluation?

Smith et al. (ICLR, 2023)

With **S5**, we aimed to simplify S4 by sticking in the time domain and using **parallel scans.**

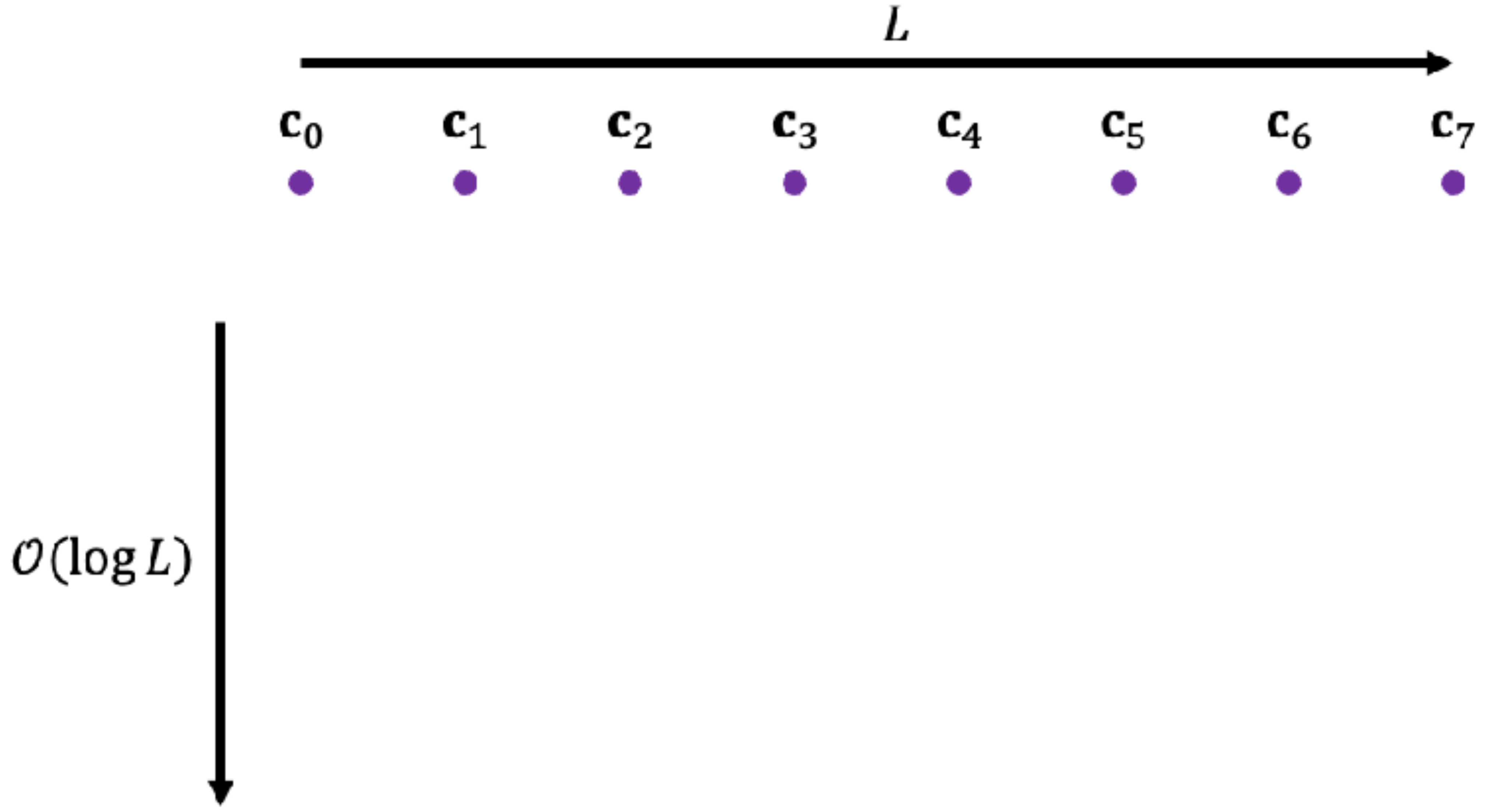
We use complex diagonal matrices to avoid cubic cost.



A pair of linear updates is still linear!

Smith et al. (ICLR, 2023)

With **S5**, we aimed to simplify S4 by sticking in the time domain and using **parallel scans.**



Applying this update in parallel and recursively allows us to compute the entire state sequence in O(log T) time on a parallel machine.

Smith et al. (ICLR, 2023)

S5 performs very well on machine learning benchmarks, including Path-X

Model (Input length)	`ListOps` (2,048)	`Text` (4,096)	`Retrieval` (4,000)	`Image` (1,024)	`Pathfinder` (1,024)	`Path-X` (16,384)	Avg.
Transformer	36.37	64.27	57.46	42.44	71.40	✗	53.66
Luna-256	37.25	64.57	79.29	47.38	77.72	✗	59.37
H-Trans.-1D	49.53	78.69	63.99	46.05	68.78	✗	61.41
CCNN	43.60	84.08	✗	88.90	91.51	✗	68.02
Mega ($\mathcal{O}(L^2)$)	**63.14**	**90.43**	<u>91.25</u>	**90.44**	**96.01**	<u>97.98</u>	**88.21**
Mega-chunk ($\mathcal{O}(L)$)	58.76	<u>90.19</u>	90.97	85.80	94.41	93.81	85.66
S4D-LegS	60.47	86.18	89.46	88.19	93.06	91.95	84.89
S4-LegS	59.60	86.82	90.90	88.65	94.20	96.35	86.09
Liquid-S4	<u>62.75</u>	89.02	91.20	<u>89.50</u>	94.8	96.66	87.32
S5	62.15	89.31	**91.40**	88.00	<u>95.33</u>	**98.58**	<u>87.46</u>

Smith et al. (ICLR, 2023)

S5 tops leaderboards for neural prediction benchmarks too!

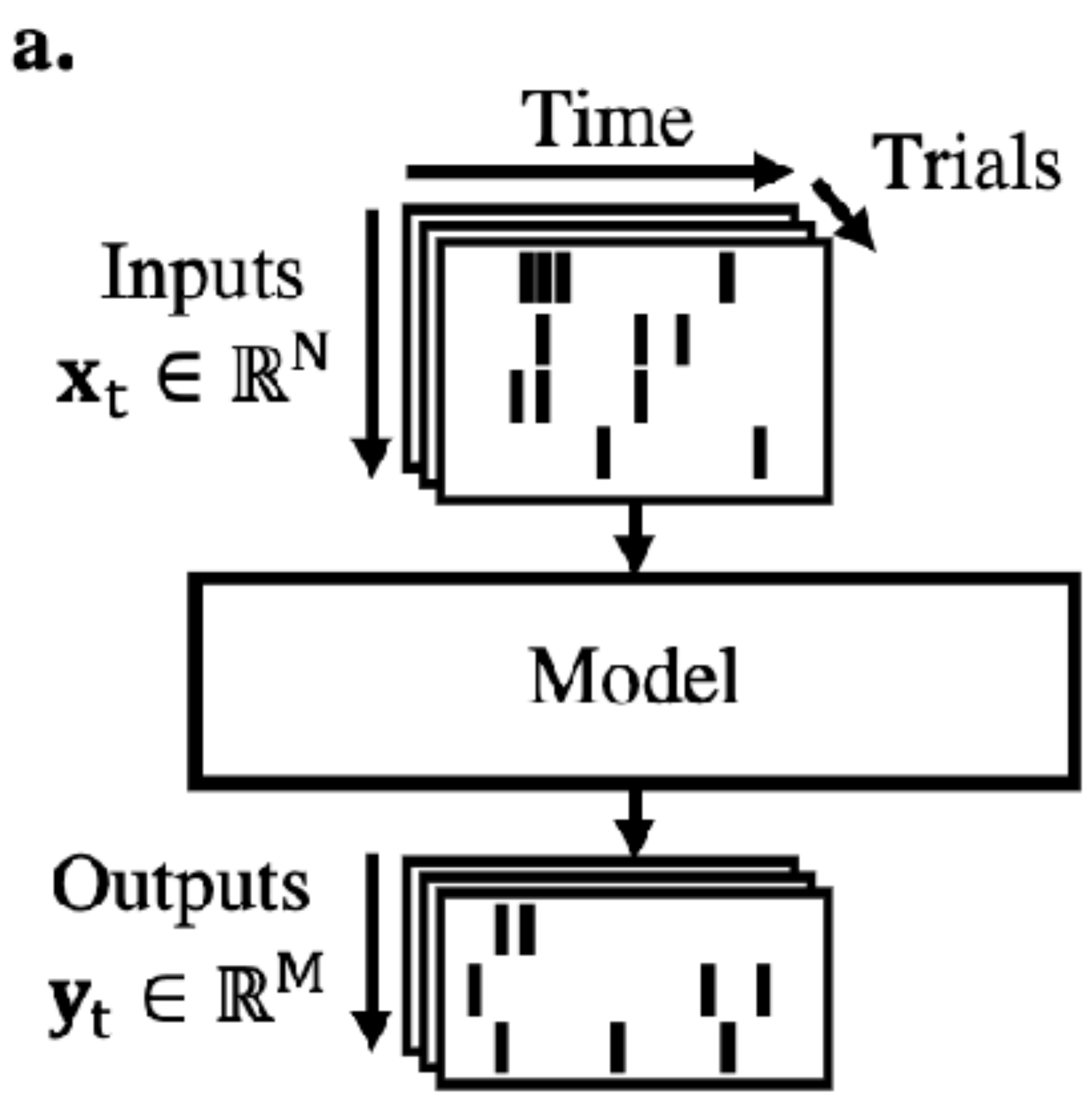


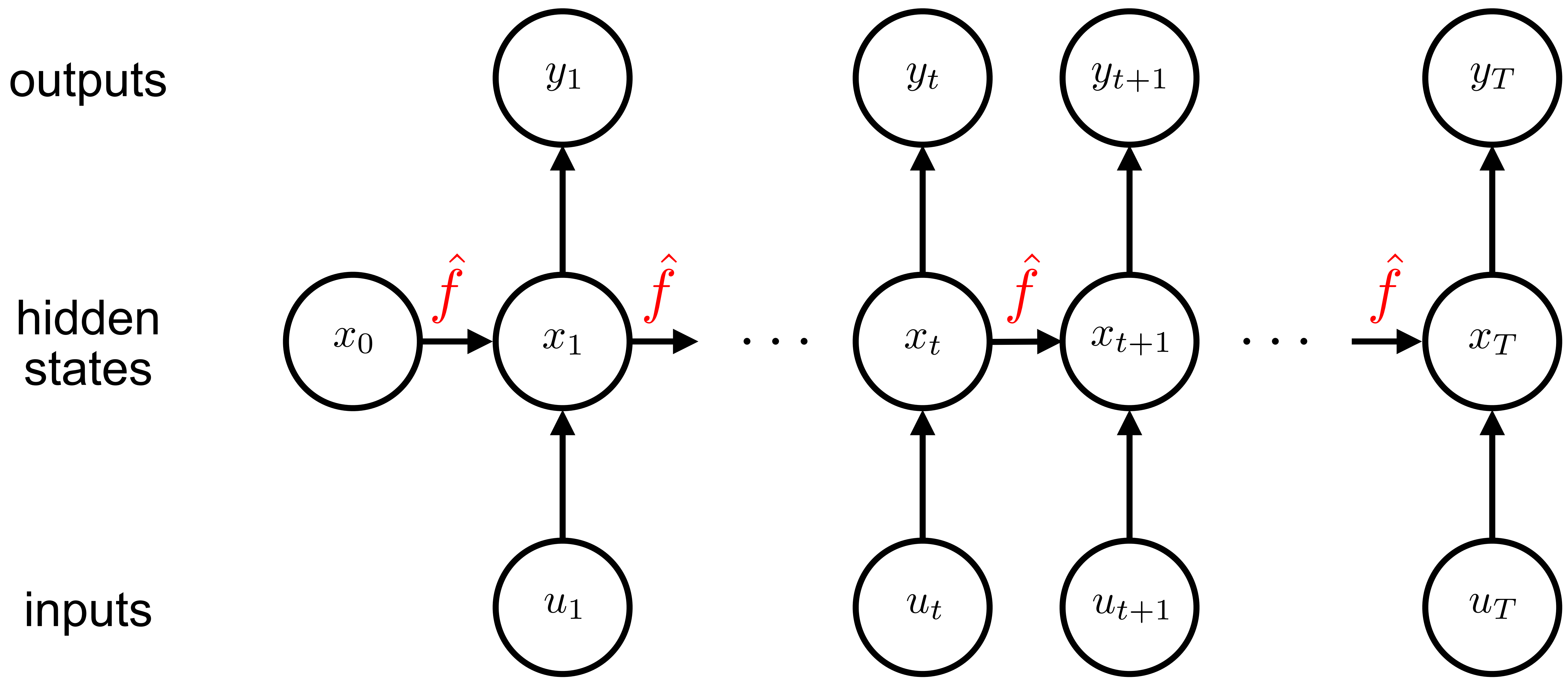
Table 1: Co-smoothing (in units of bits-per-spike) metric on MC_Maze and DMFC_RSG benchmarks [Pei *et al.*, 2021] for S5 compared to SOTA methods. Note: we exclude ensemble methods and only consider single models.

Method	MC_Maze (↑)	DMFC_RSG (↑)
S5 (Ours)	**0.3826**	**0.1981**
SSLFADS	0.3748	N/A
STNDT	0.3691	0.1859
iLQR-VAE	0.3559	N/A
Neural RoBERTa	0.3551	N/A
RNNf	0.3382	0.1781
AutoLFADS	0.3364	0.1829
MINT	0.3304	0.1821
NDT	0.3229	0.1720
SLDS	0.2249	0.1243

Unlike convolutions, the S5 parallel scan also allows for different dynamics at each time point

Gu and Dao (2023) exploited this property in **Mamba,** which uses input-dependent dynamics.

Back to RNNs… are they really so sequential?



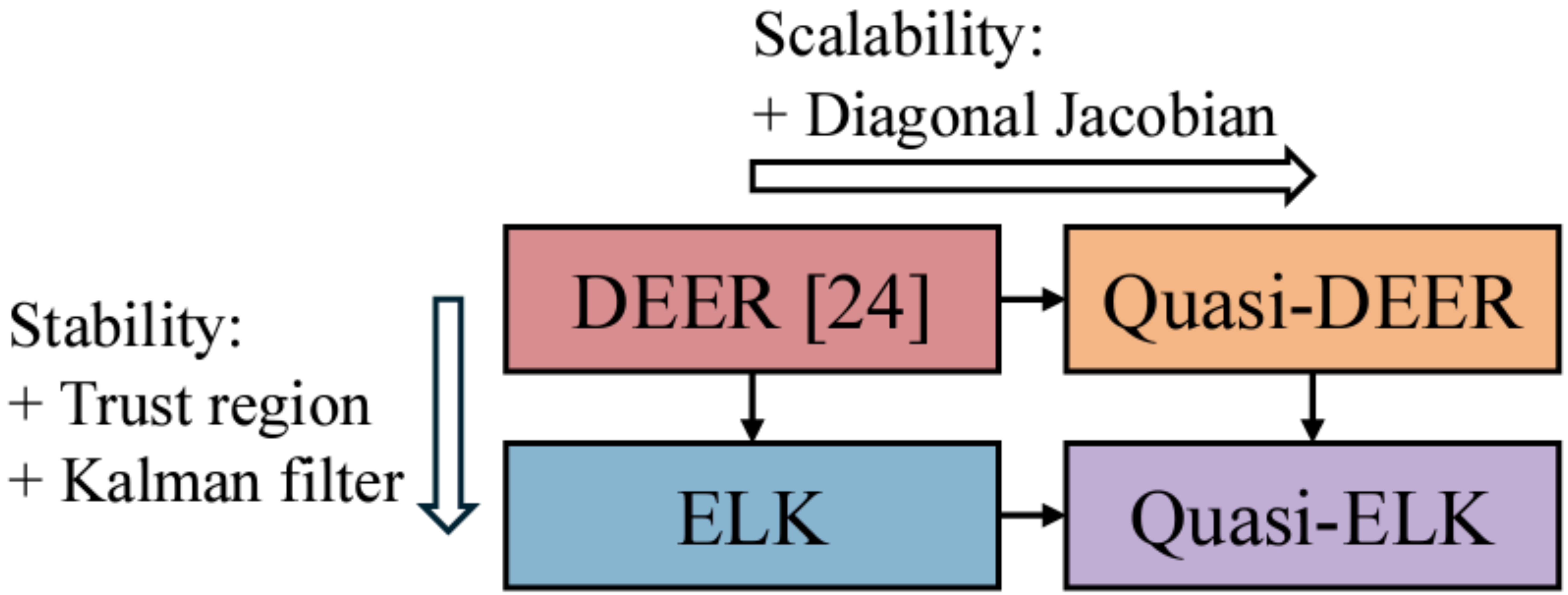
Idea: What if we linearize f around a current guess of the latent states?

$$f(x_t) \approx f(x_t^{(i)}) + \frac{\partial f}{\partial x}(x_t^{(i)})(x_t - x_t^{(i)}) \triangleq \hat{f}(x_t; x_t^{(i)})$$

Then we can use parallel scan to solve for the states, linearize again, and repeat.
This is the Gauss-Newton method. Lim et al (2024) called it **DEER**.

Lim et al (ICLR, 2024)
Gonzalez et al. (NeurIPS, 2024)

Improving on DEER with **ELK**

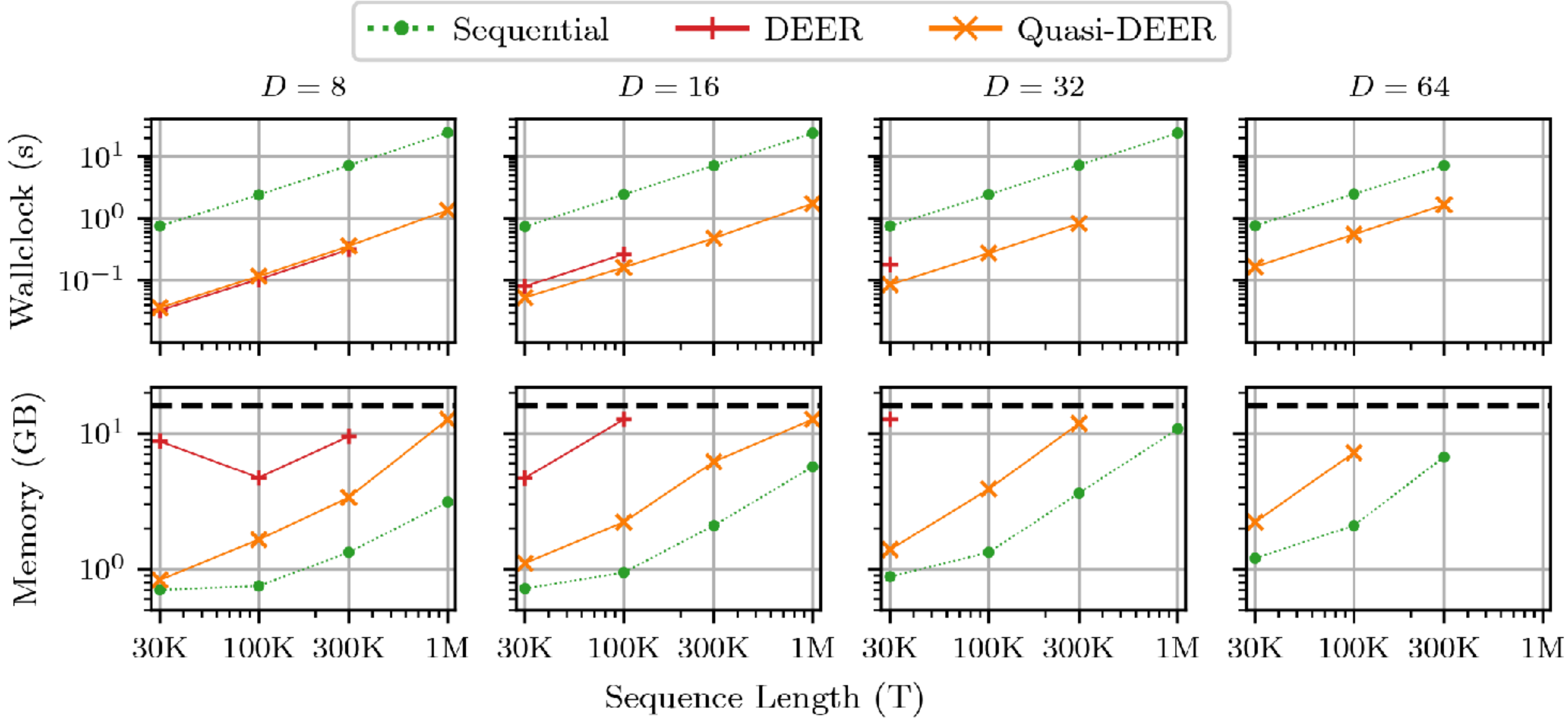


Gauss-Newton can be unstable! Levenberg-Marquardt introduces a trust region, which turns this into a Kalman filtering problem.

We call this method **ELK.**

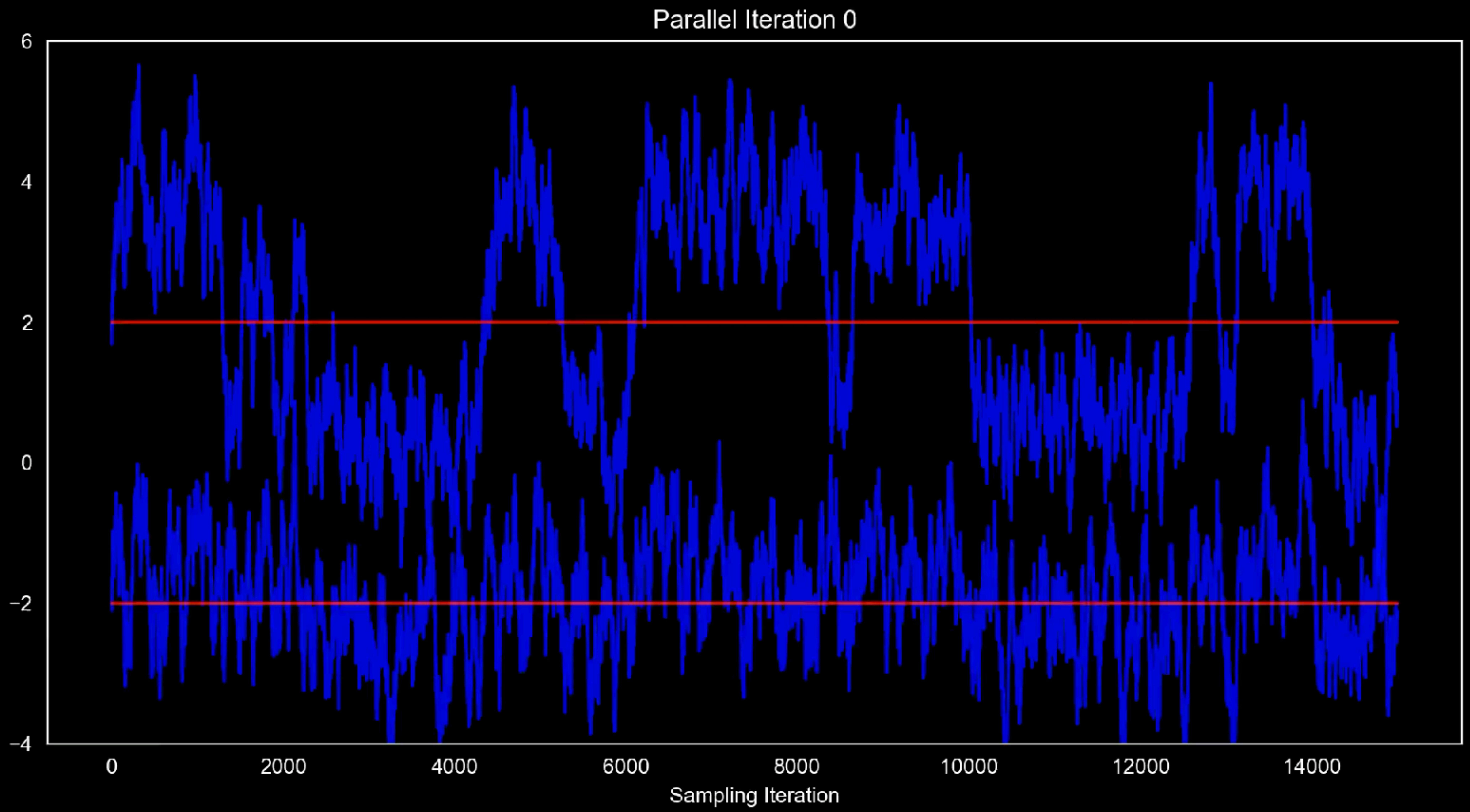
Gonzalez et al. (NeurIPS, 2024)

Quasi-DEER is faster and more memory efficient with diagonal Jacobian approximations

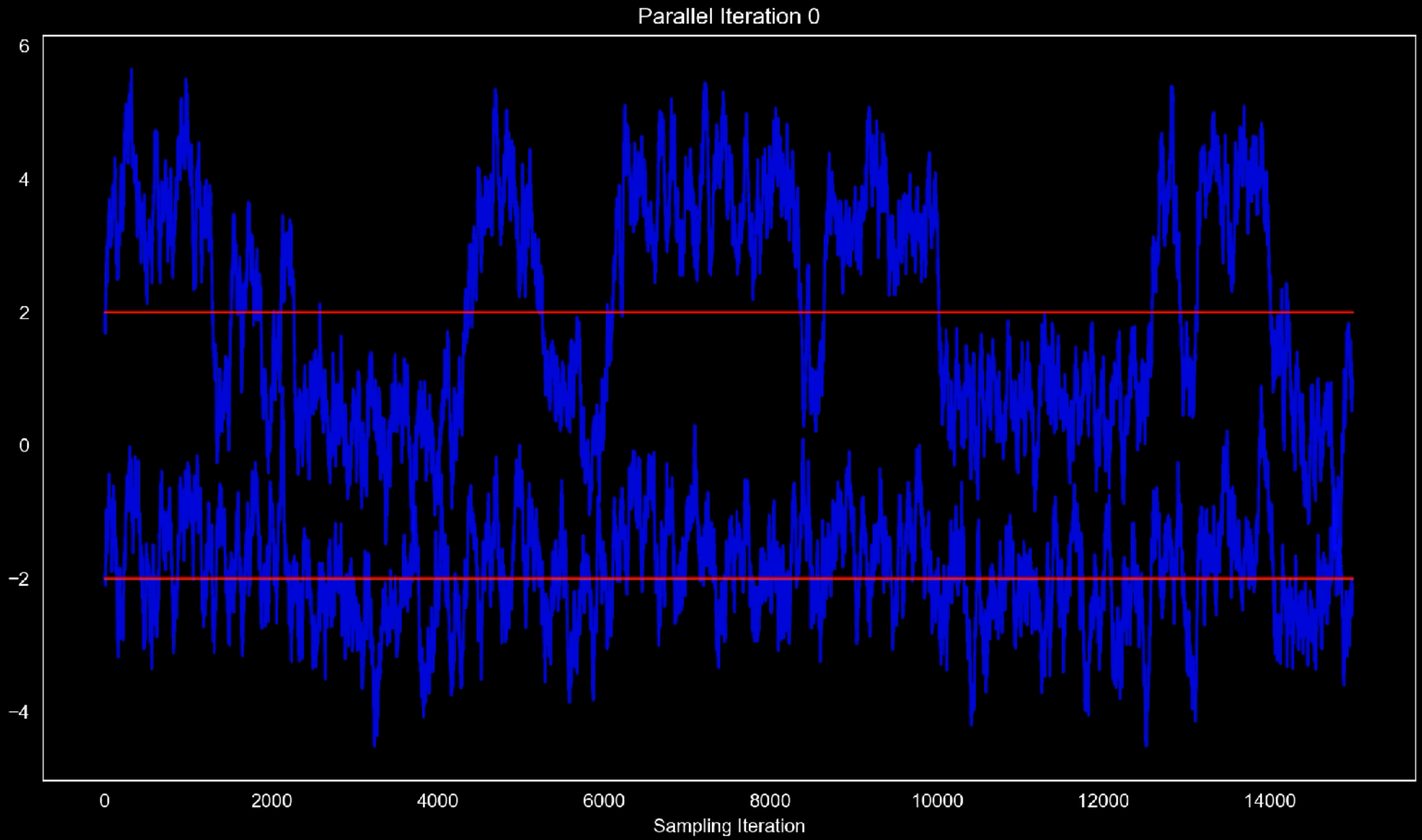


Gonzalez et al. (NeurIPS, 2024)

(Quasi-DEER) becomes numerically unstable when the linear approximation diverges



ELK addresses this limitation by introducing a trust region



Conclusions

- Whether you’re interested in biological or artificial intelligence, state space models are for you!

- With modern methods for measuring and perturbing brain activity, we can fit SSMs at scale, test their predictions, and start to elucidate underlying circuit methods.

- Perhaps surprisingly, the humble linear dynamical system is making a comeback in ML and beating benchmarks for long-range sequence modeling.

- There is lots of exciting (and very interdisciplinary) work to be done on both fronts.

Acknowledgements

Linderman Lab

Dan Biderman
Kelly Buchanan
Julia Costacurta
Xavier Gonzalez
Amber Hu
Aditi Jha
Hyun Dong Lee
Alisa Levin
Matt MacKay
Jimmy Smith
Evelyn Song
Ian Christopher Tanoh
Libby Zhang
Yixiu Zhao
David Zoltowski

Alumni

Ben Antin
Shaunak Bhandarkar
Lea Duncker
Elizabeth DuPre
Maxwell Kounga
Dieterich Lawson
Blue Sheffer
Andy Warrington
Alex Williams

Collaborators

David Anderson
Adi Nair
Ann Kennedy
Mengyu Liu
Amit Vinograd
George Mountoufaris

Bob Datta
Bernardo Sabatini
Sam Gershman
Nao Uchida
John Assad
Kanaka Rajan
Jeff Markowitz
Win Gillis
Caleb Weinreb

Anne Brunet
Karl Deisseroth
Claire Bedbrook
Ravi Nath

Liqun Luo
Drew Friedmann

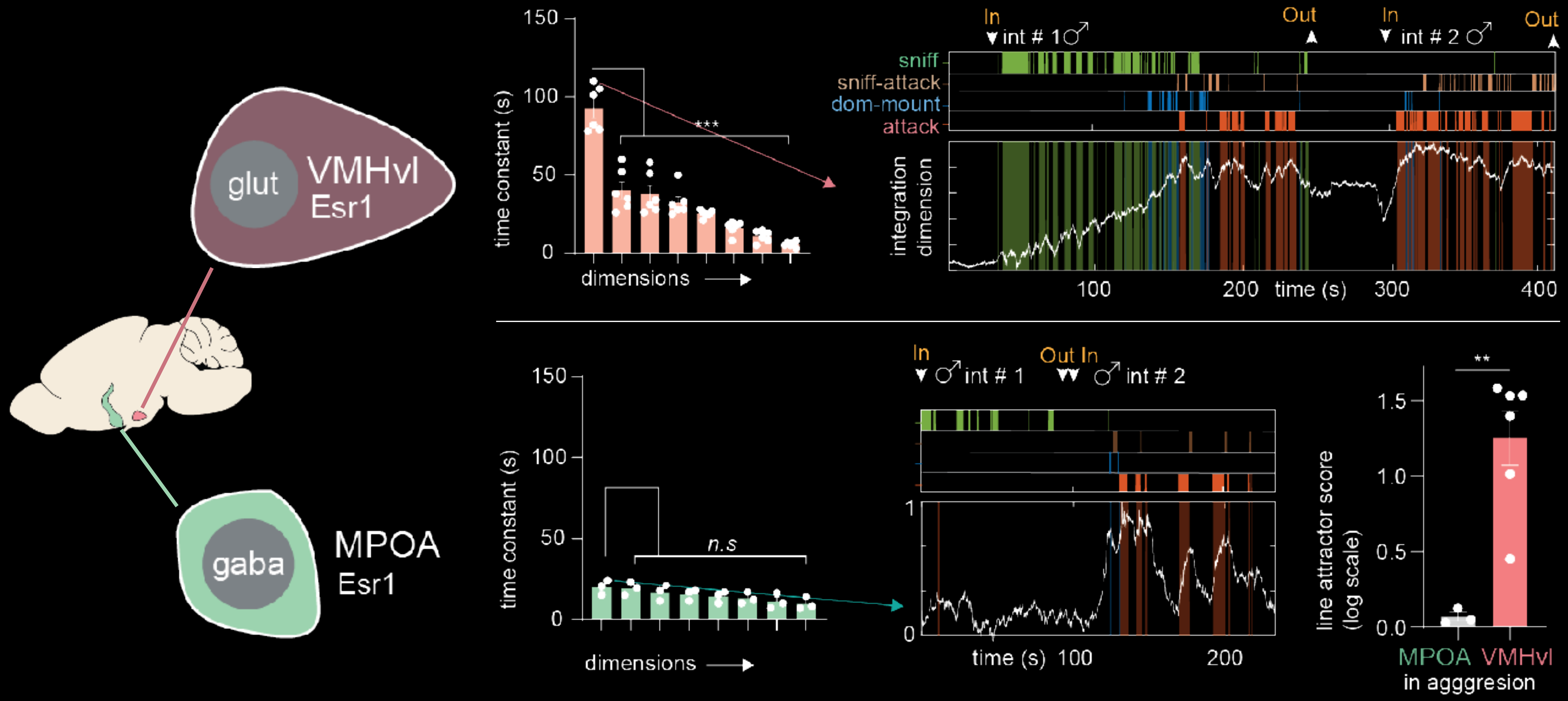
Kevin Murphy

Funding Sources

NIH BRAIN Initiative
NSF CRCNS
McKnight Foundation
Simons Foundation
Sloan Foundation

https://github.com/lindermanlab https://web.stanford.edu/~swl1/

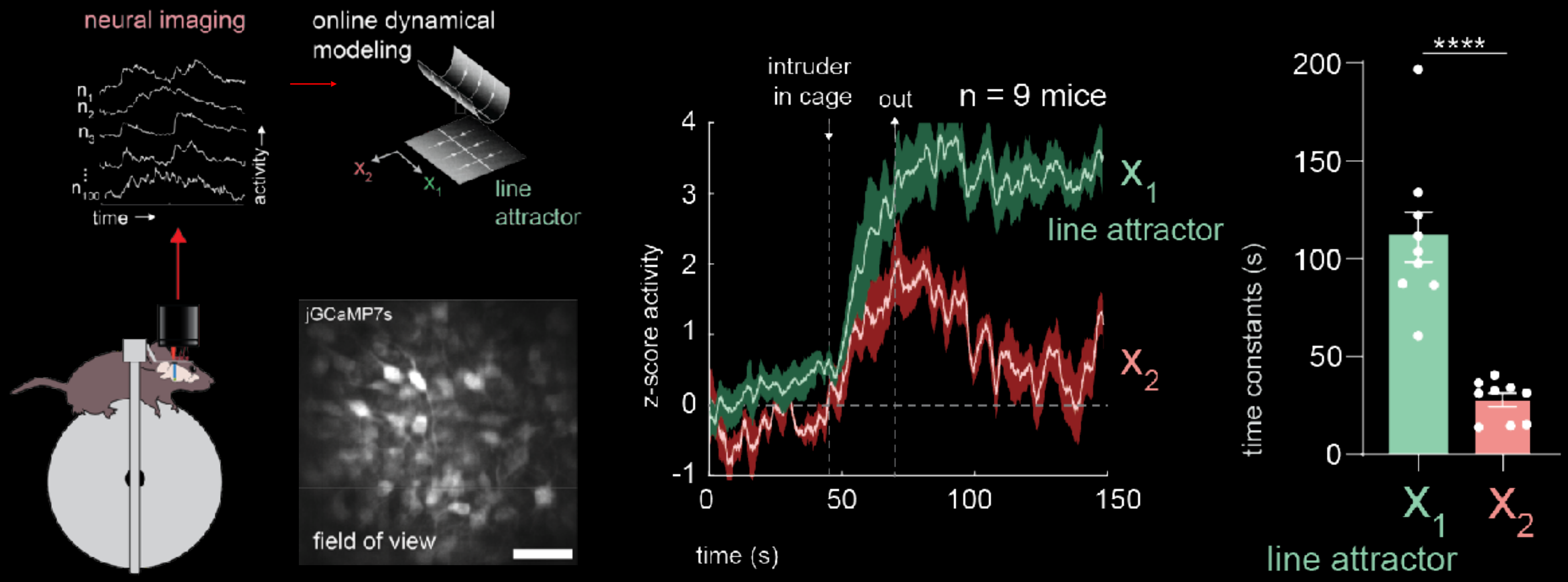
Not all hypothalamic nucleii exhibit line attractor-like dynamics



rSLDS analysis of MPOA finds faster time-locked motor behavior related dimensions

Nair et al. (Cell, 2023)

How can we gain access to the line attractor for perturbation?

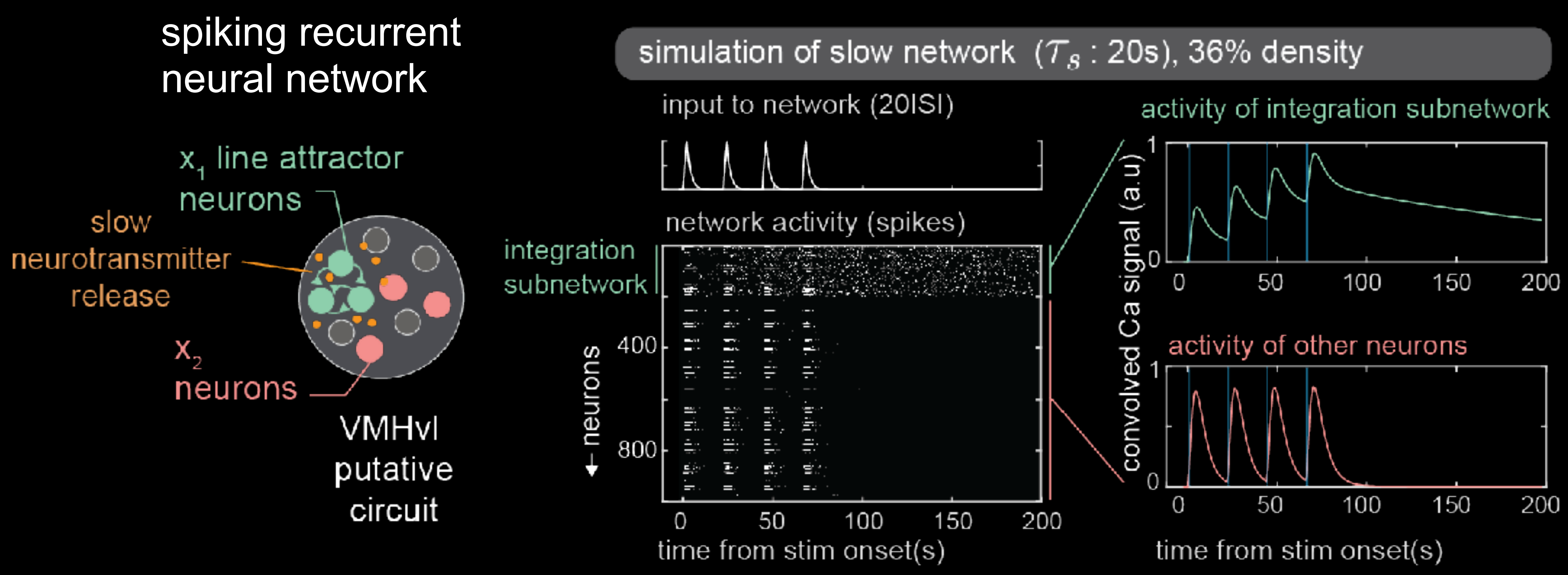


VMHvl-Esr1 neurons are show line attractor dynamics during observation of aggression

Vinograd, Nair et al. (Nature, 2024)

What mechanisms give rise to the line attractor?

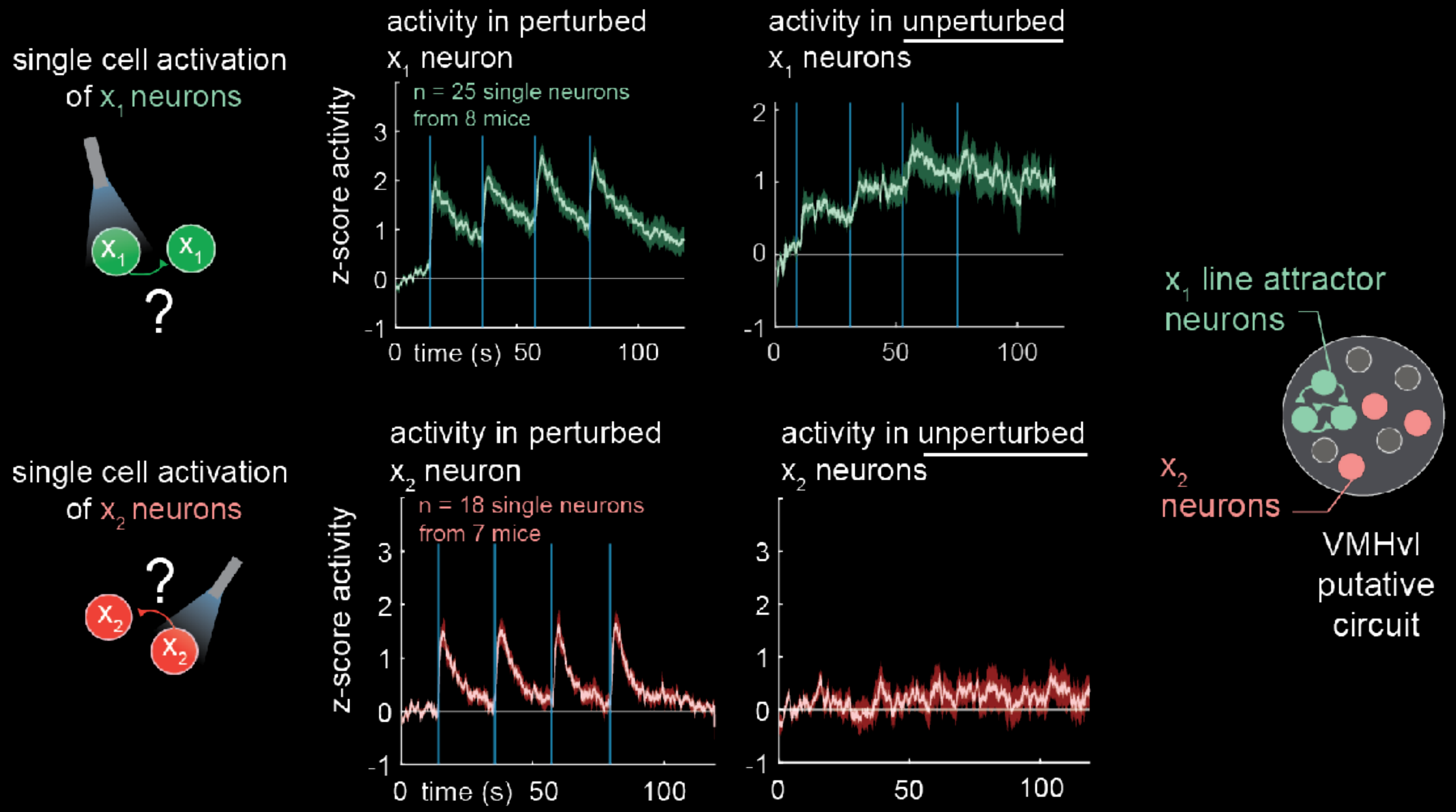
Line attractors are notoriously fragile and require precise fine-tuning.



spiking neural networks with slow neurotransmitter release can create robust line attractors with time constants seen in data

Vinograd, Nair et al. (Nature, 2024)

What mechanisms give rise to the line attractor?

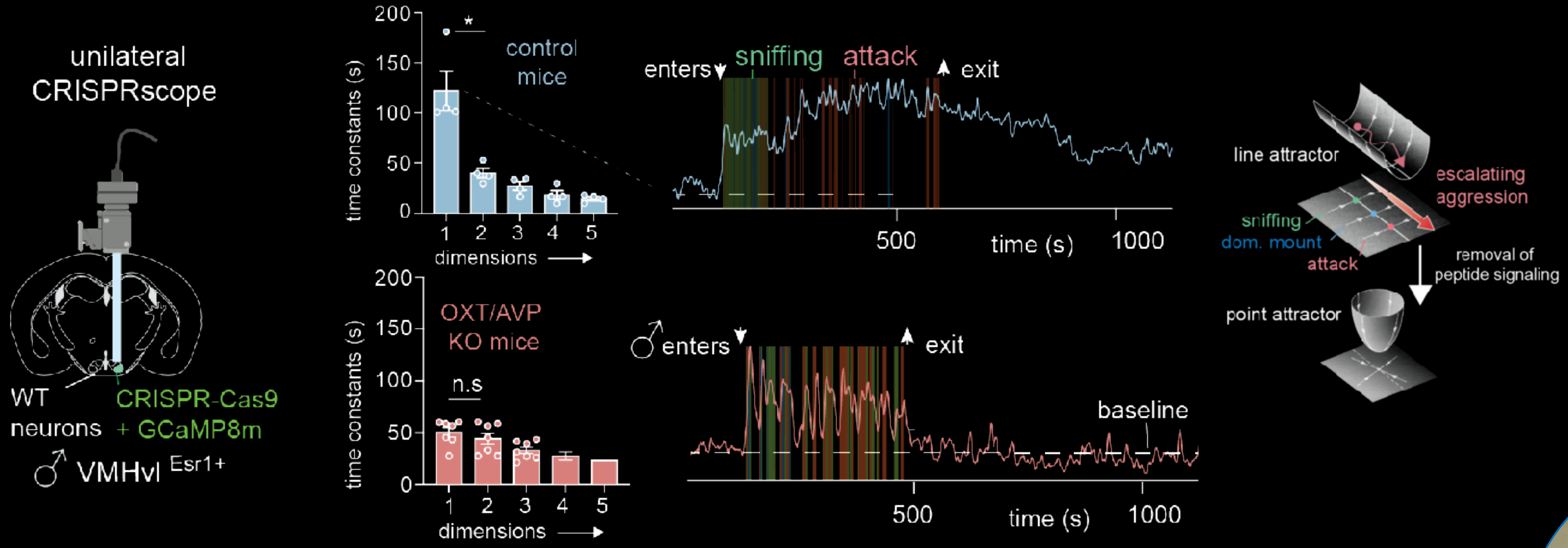


selective functional connectivity among line attractor-aligned x_1 neurons

Vinograd, Nair et al. (Nature, 2024)

What mechanisms give rise to the line attractor?

“CRISPRscope” for cell-type specific perturbation of neuropeptide receptors + neural imaging

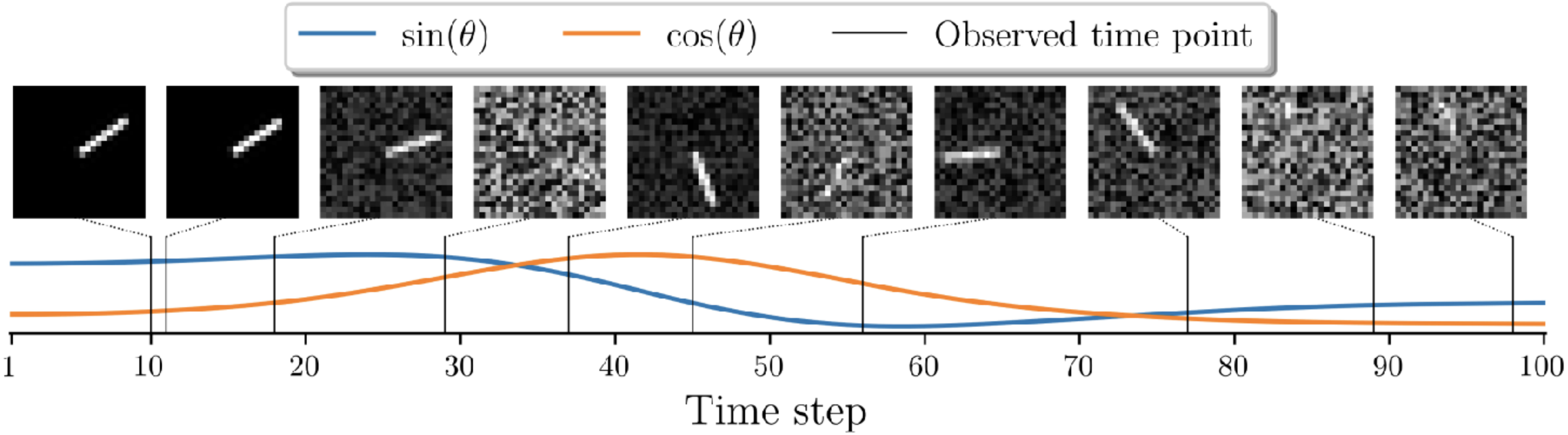


Removal of OXT/AVP receptors eliminates integration & line attractor dynamics in VMHvl.

Mountoufaris et al. (Cell, 2024)

Unlike convolutions, S5 also allows for different dynamics at each time point

With different A's for each time step, we can handle irregularly sampled, continuous time data.



Model	Relative speed	Regression MSE ($\times 10^{-3}$)
mTAND*	8.3×	65.64 ± 4.05
RKN*	1.3×	8.43 ± 0.61
RKN-Δ_t*	1.3×	5.09 ± 0.40
ODE-RNN*	0.68×	7.26 ± 0.41
CRU*	0.68×	4.63 ± 1.07
CRU (our run)	1.00×	3.81 ± 0.28
S5	**130×**	**3.38 ± 0.28**

Smith et al. (ICLR, 2023)