The Nuts and Bolts of Probabilistic State Space Models

Part I: Foundations

Scott Linderman
Stanford University

Outline

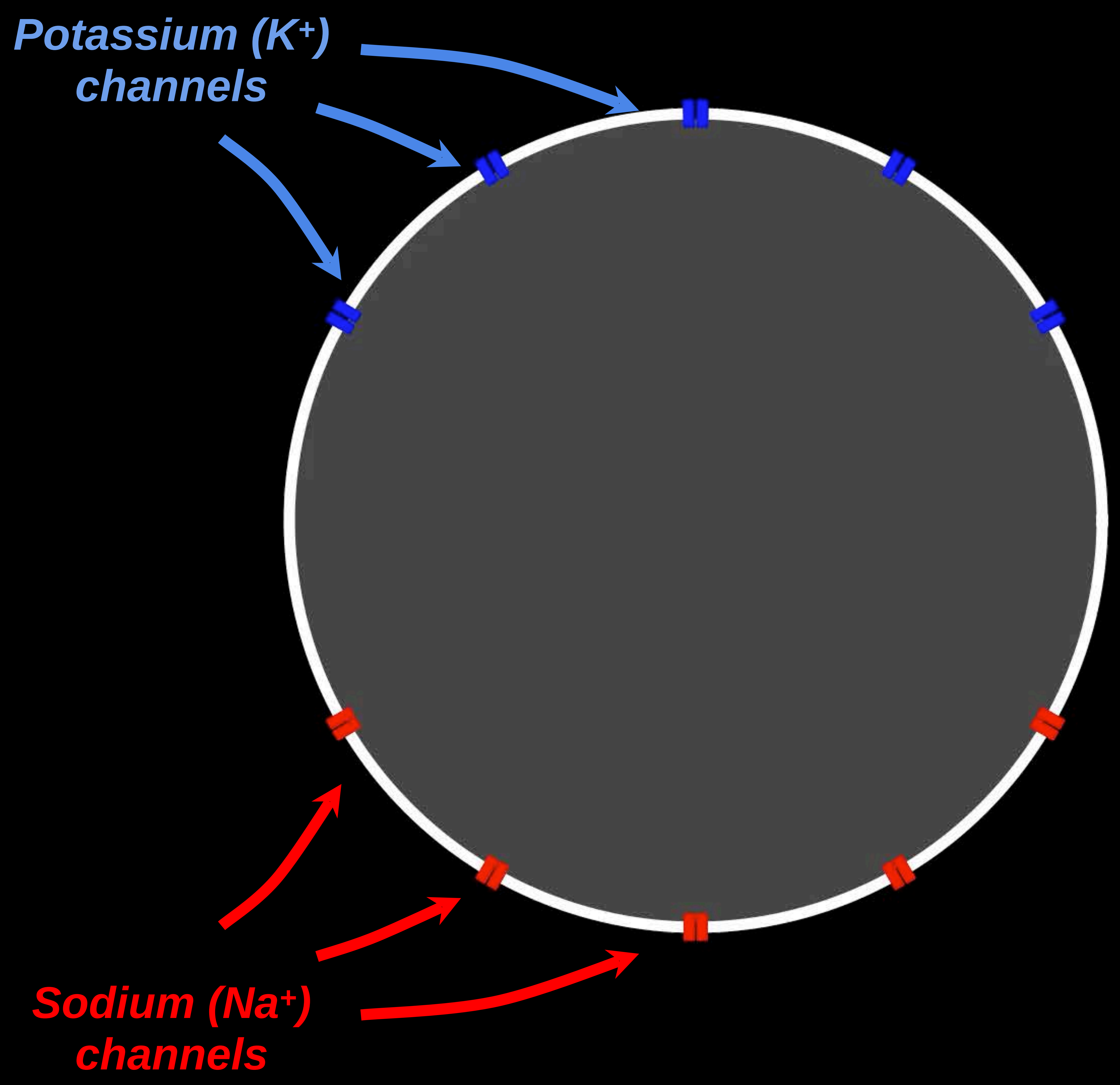
Part I: Foundations

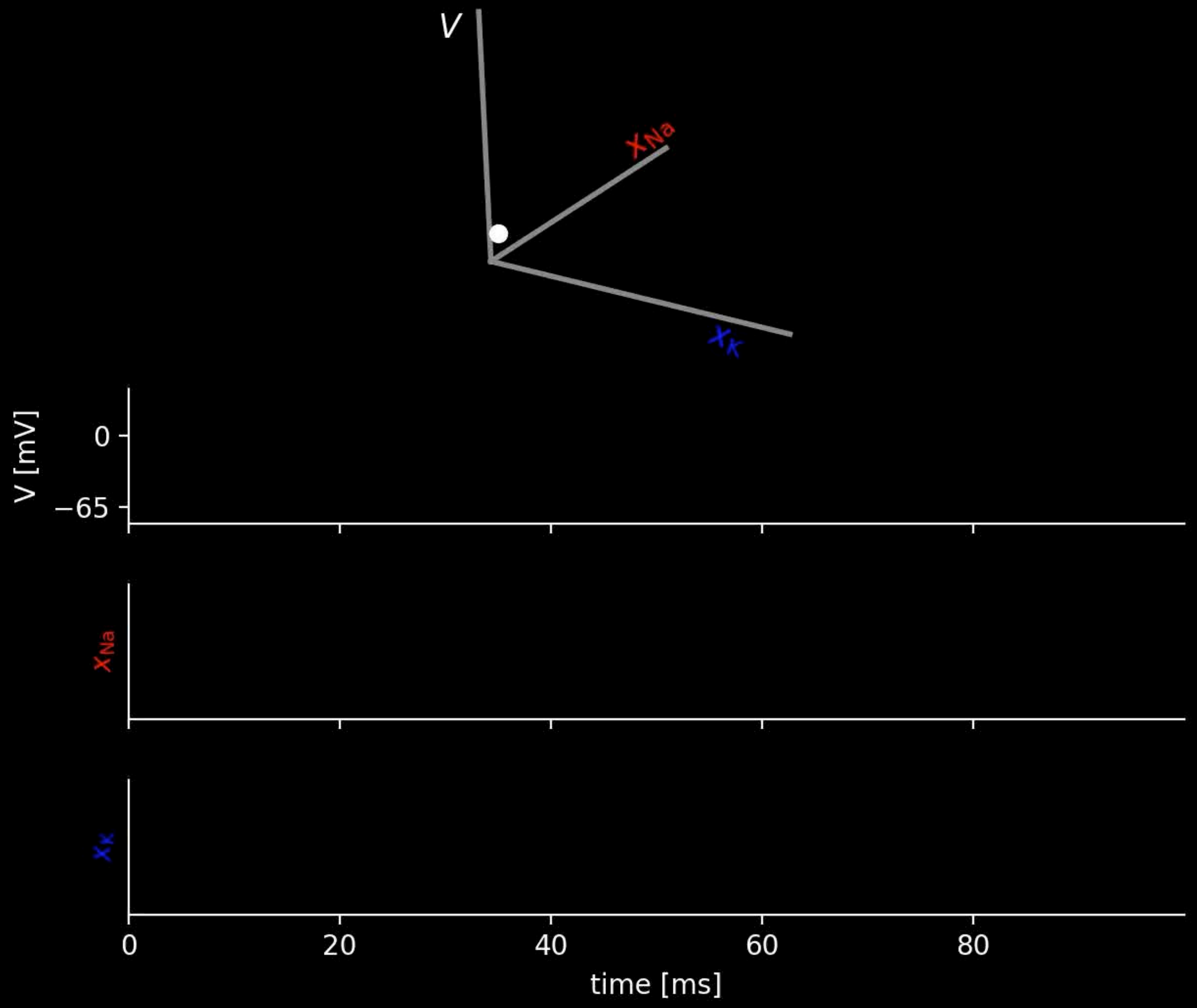
- Motivating Examples
- State Space Models (SSMs)
  - Hidden Markov Models
  - Linear Dynamical Systems
  - Nonlinear & Switching Linear Dynamical Systems
- Learning and Inference Algorithms
  - Expectation-Maximization
  - Message Passing
  - Approximate Inference (E/UKF, SMC, VI)
- Code Pointers

Part II: Trends

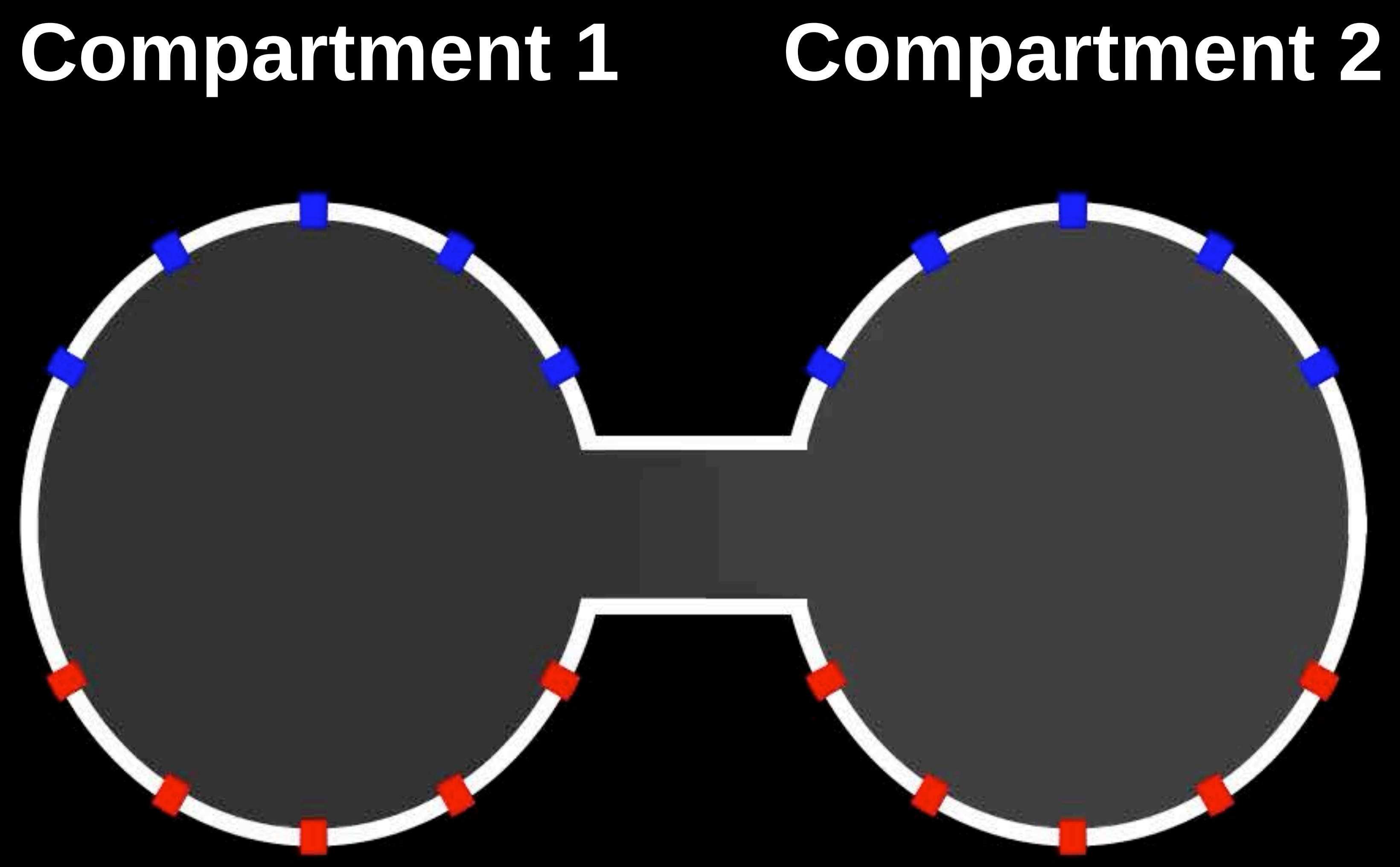
- Better Models
  - Time-Warped and Keypoint-MoSeq
  - Simple State Space Layers (S5)
- Better Algorithms
  - Variational Laplace-EM
  - Smoothing Inference with Twisted Objectives (SIXO)
  - Structured Variational Autoencoders (SVAE)

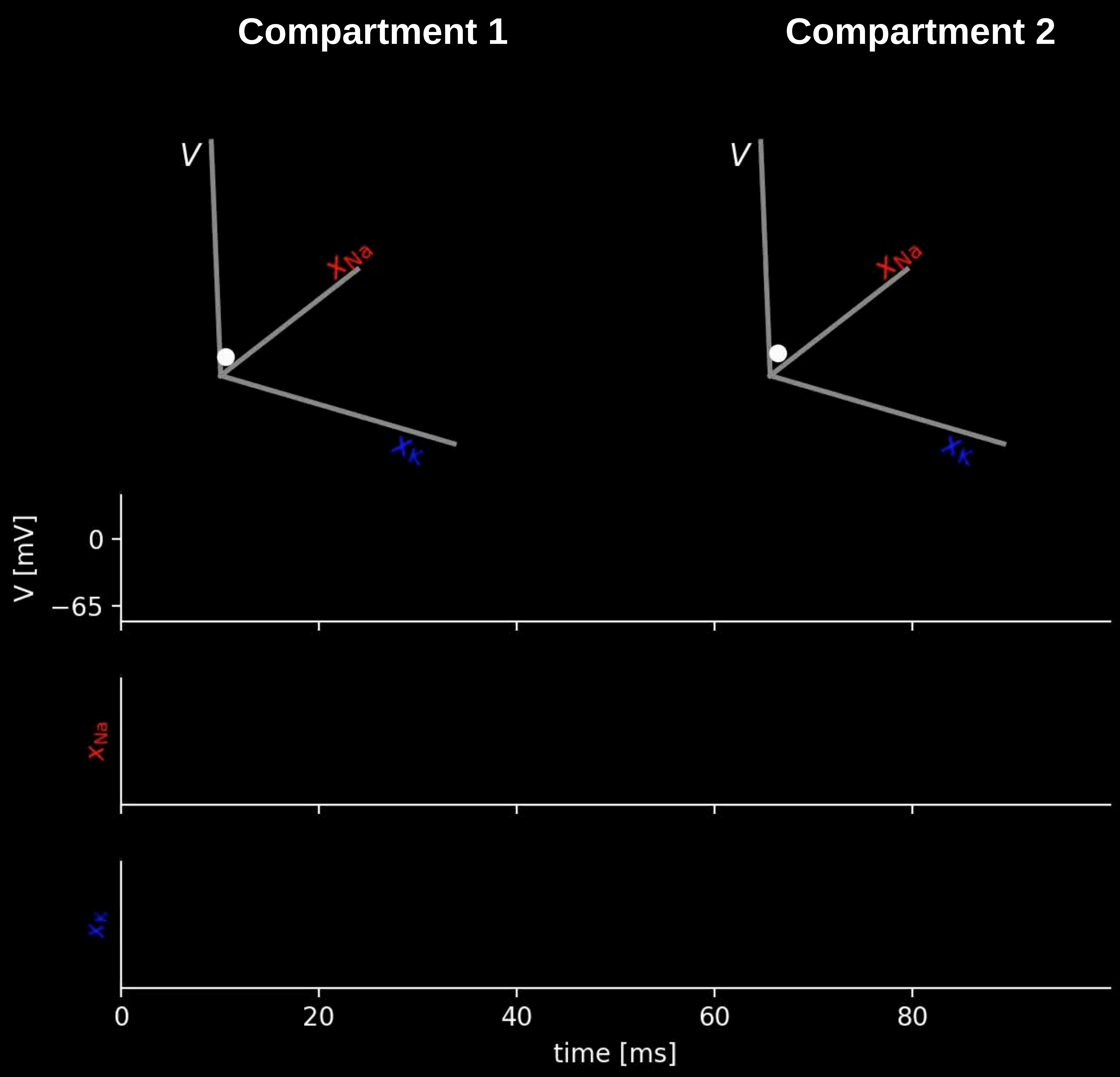
The Hodgkin-Huxley Model



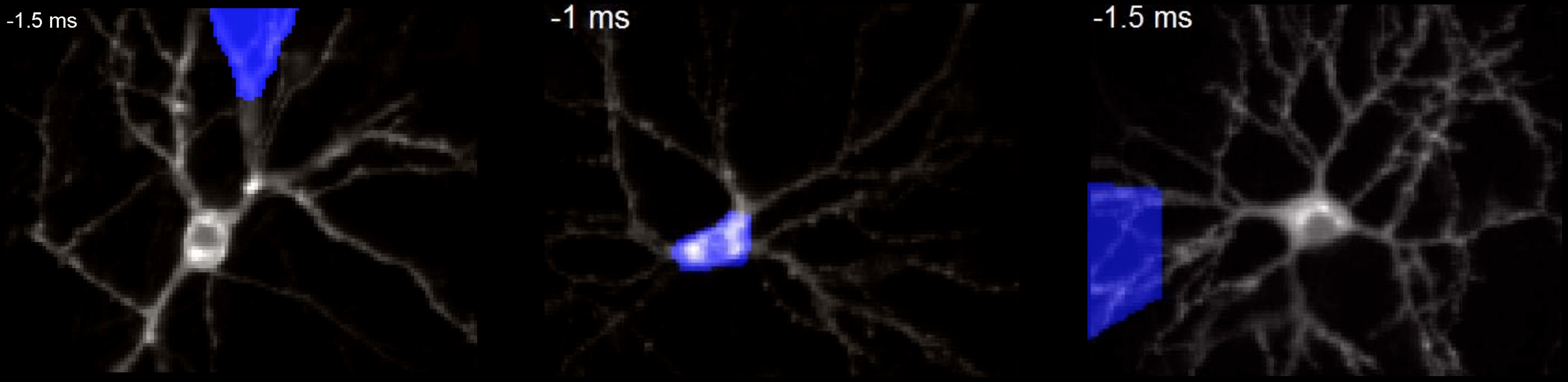


The Hodgkin-Huxley Model





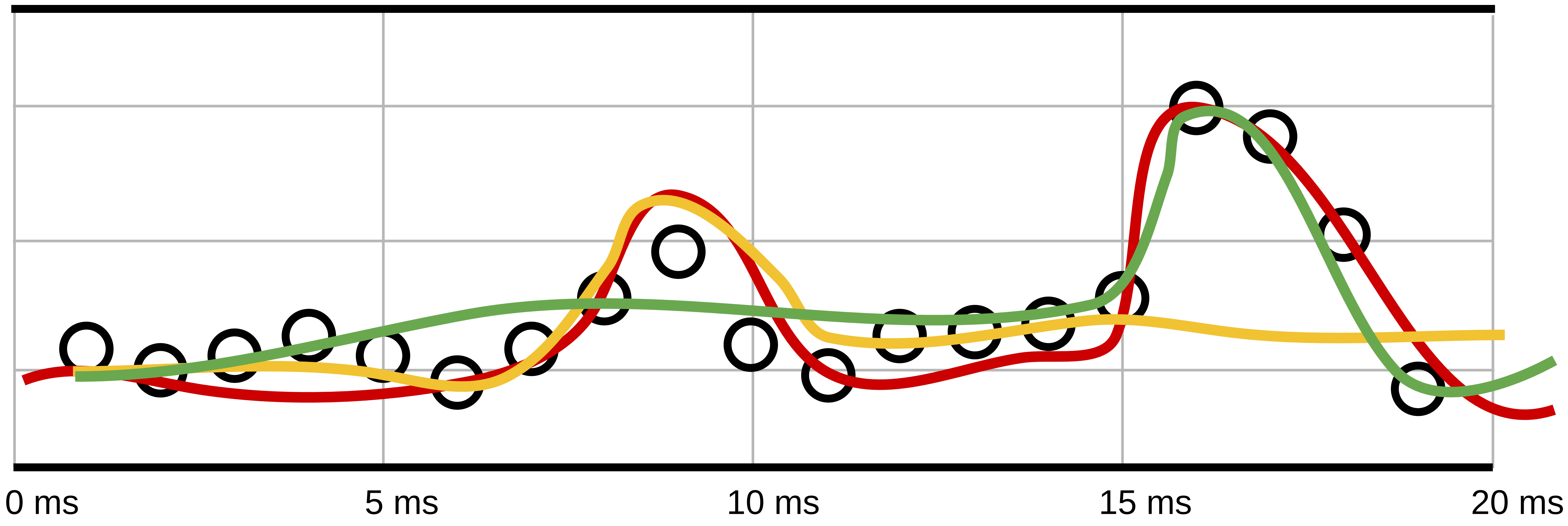
Application: Smoothing voltage imaging data



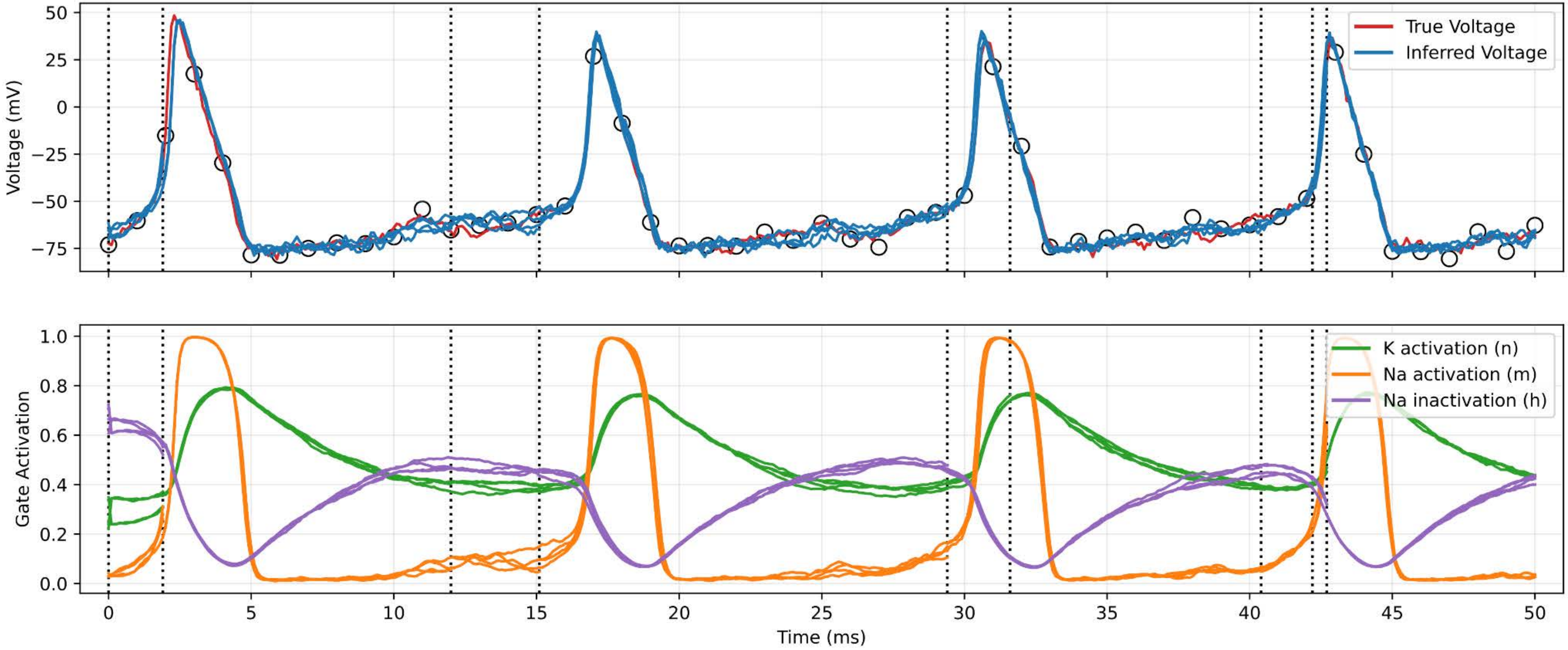
Hochbaum et al (2014)

Application: Smoothing voltage imaging data

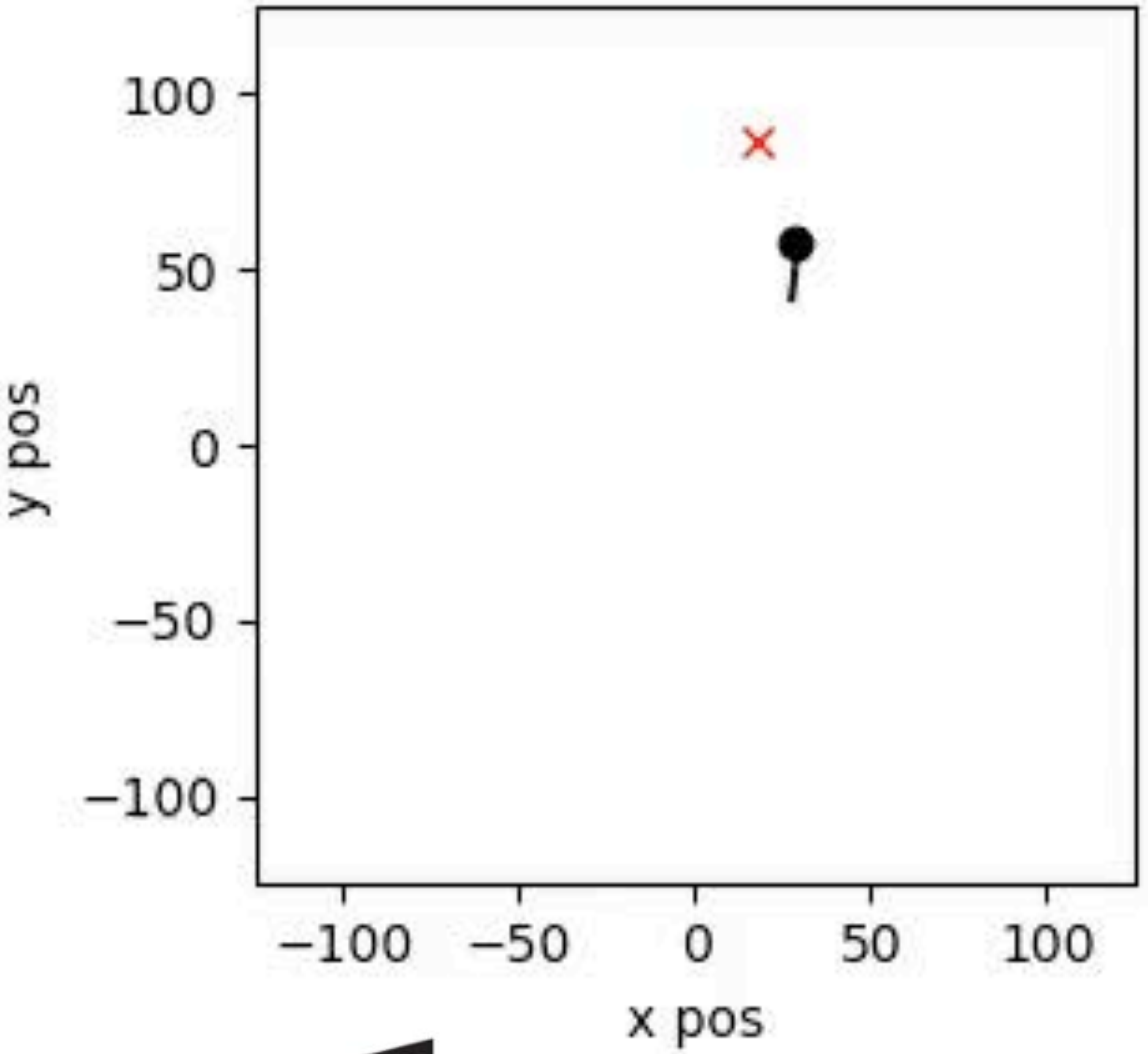
Voltage imaging data is noisy and relatively slow. Rather than simply interpolating, we can use the Hodgkin-Huxley model to smooth it.

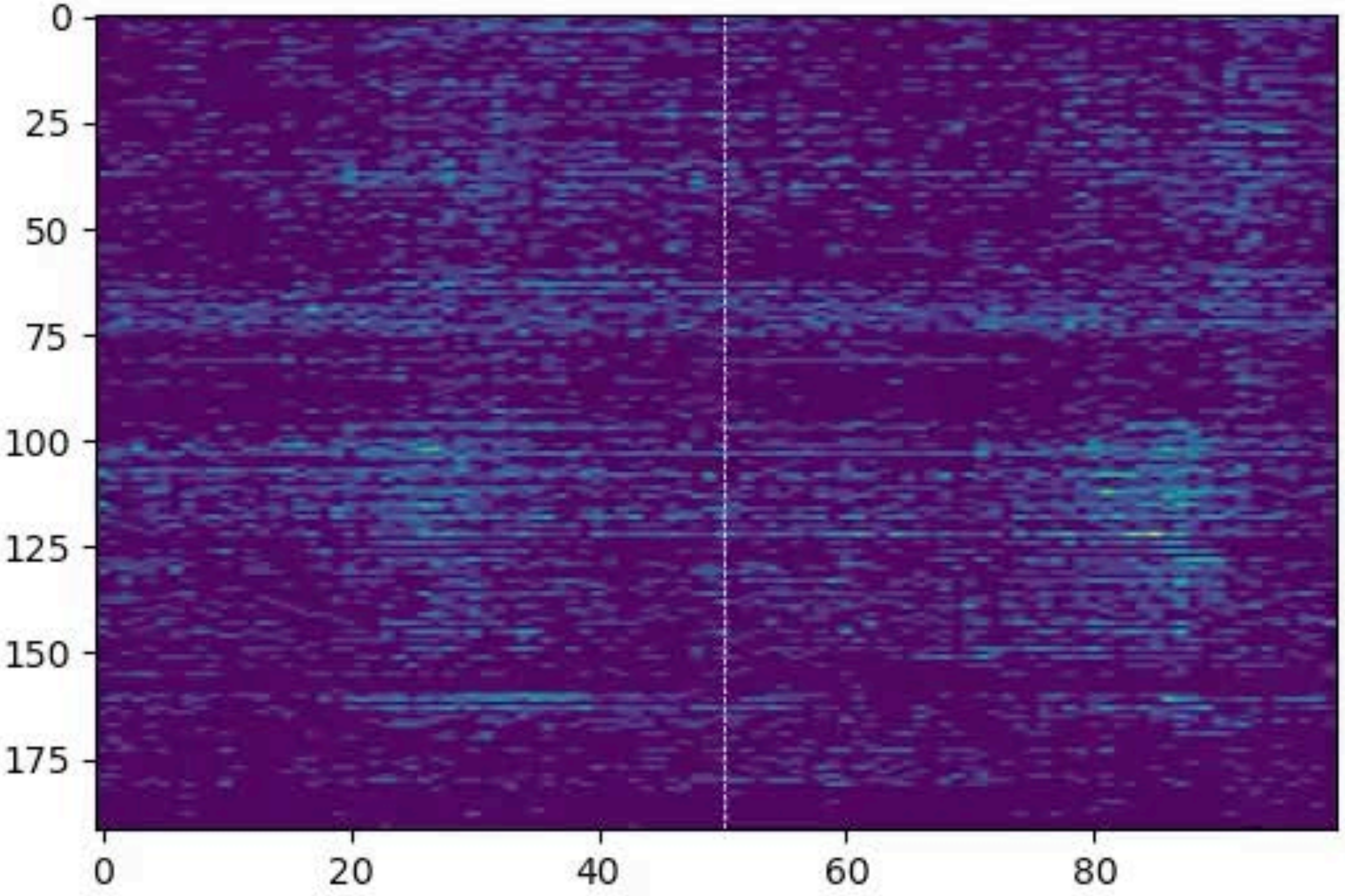


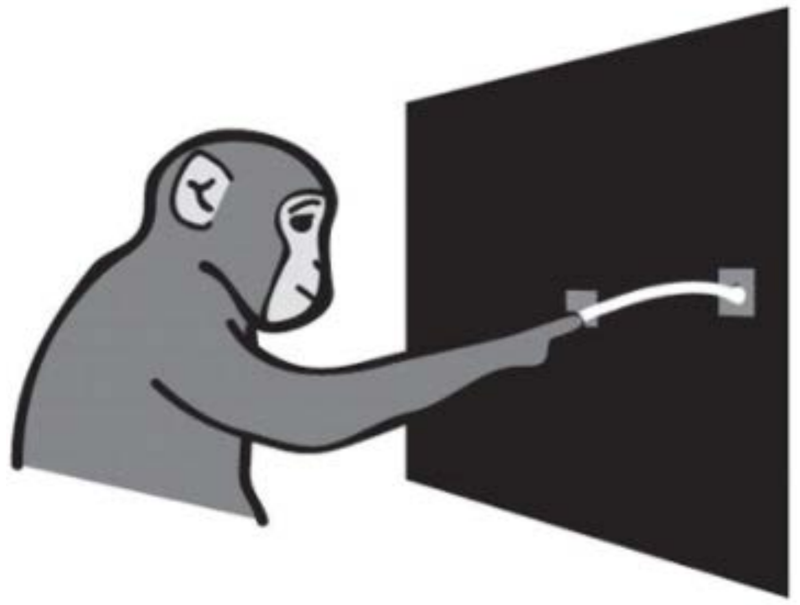
Application: Smoothing voltage imaging data



Application: Low-dimensional dynamics of neural population activity

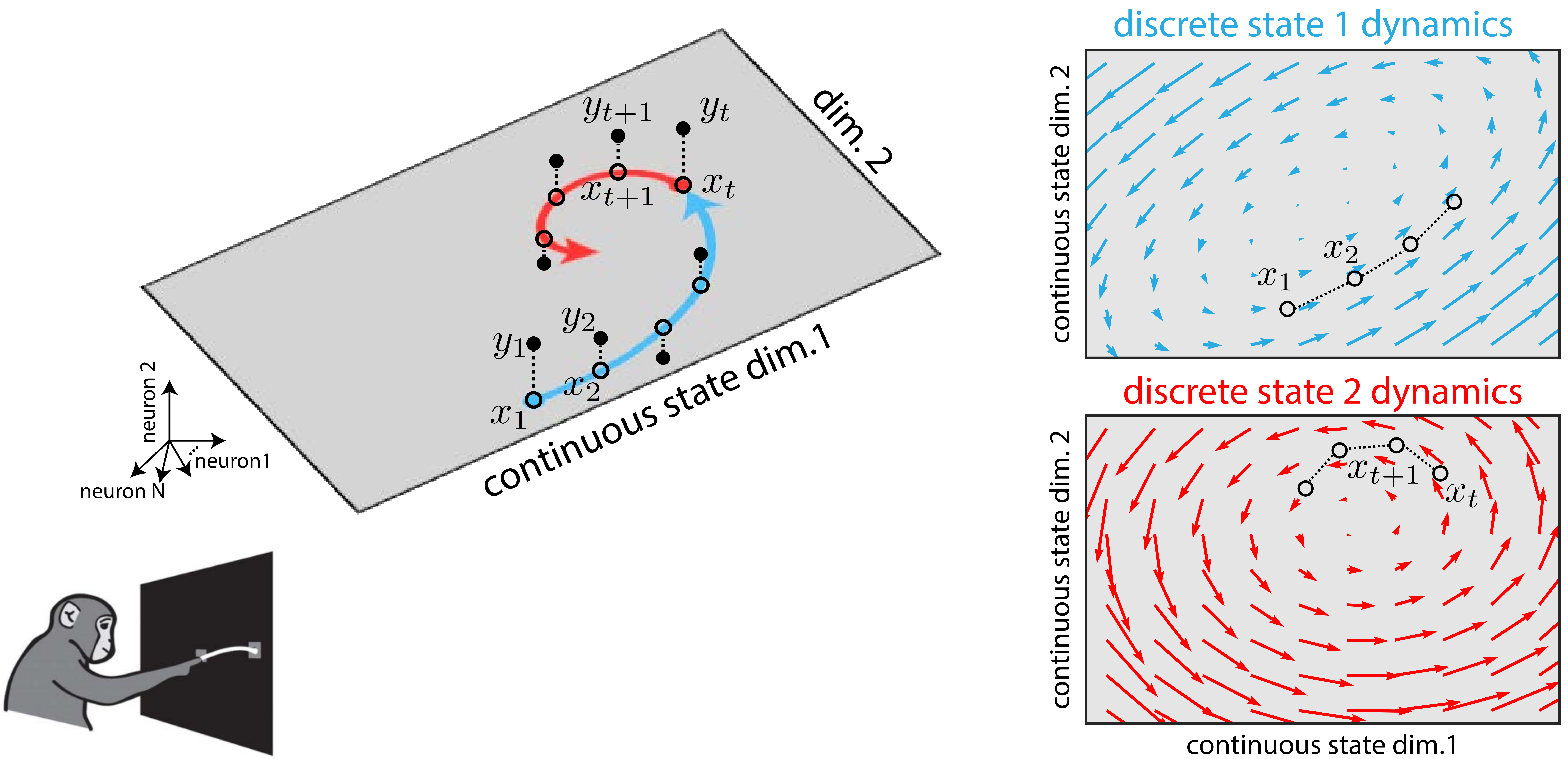




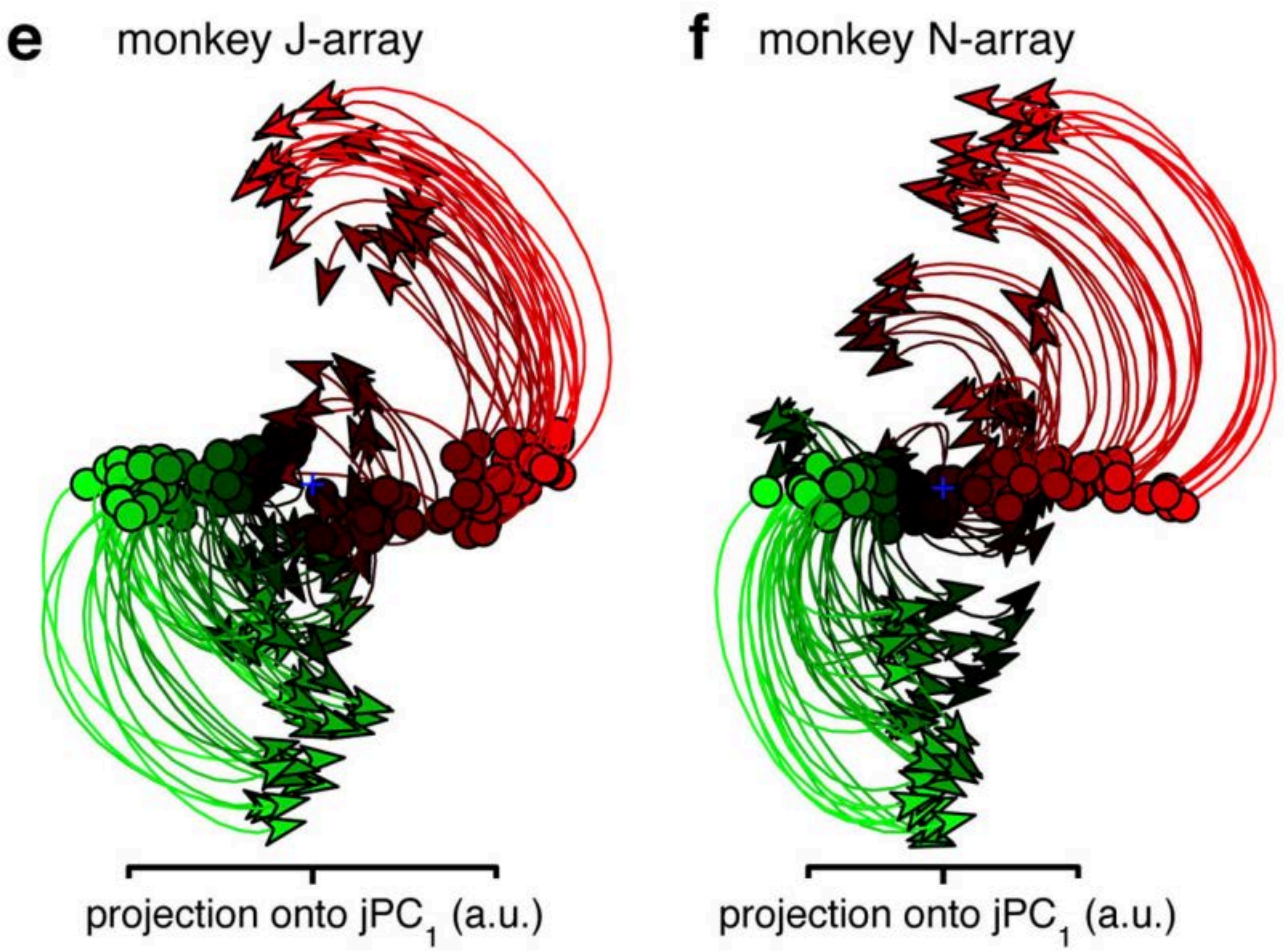


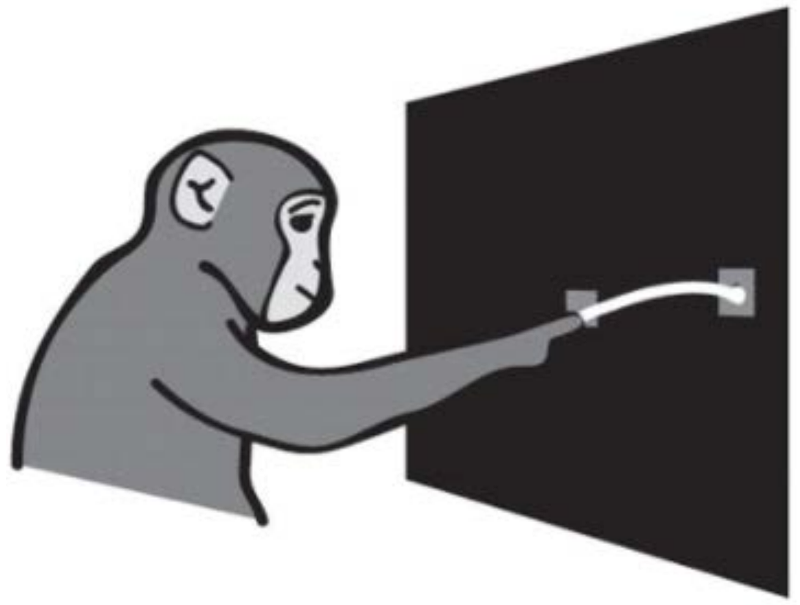
Shenoy Lab, Stanford

Application: Low-dimensional dynamics of neural population activity

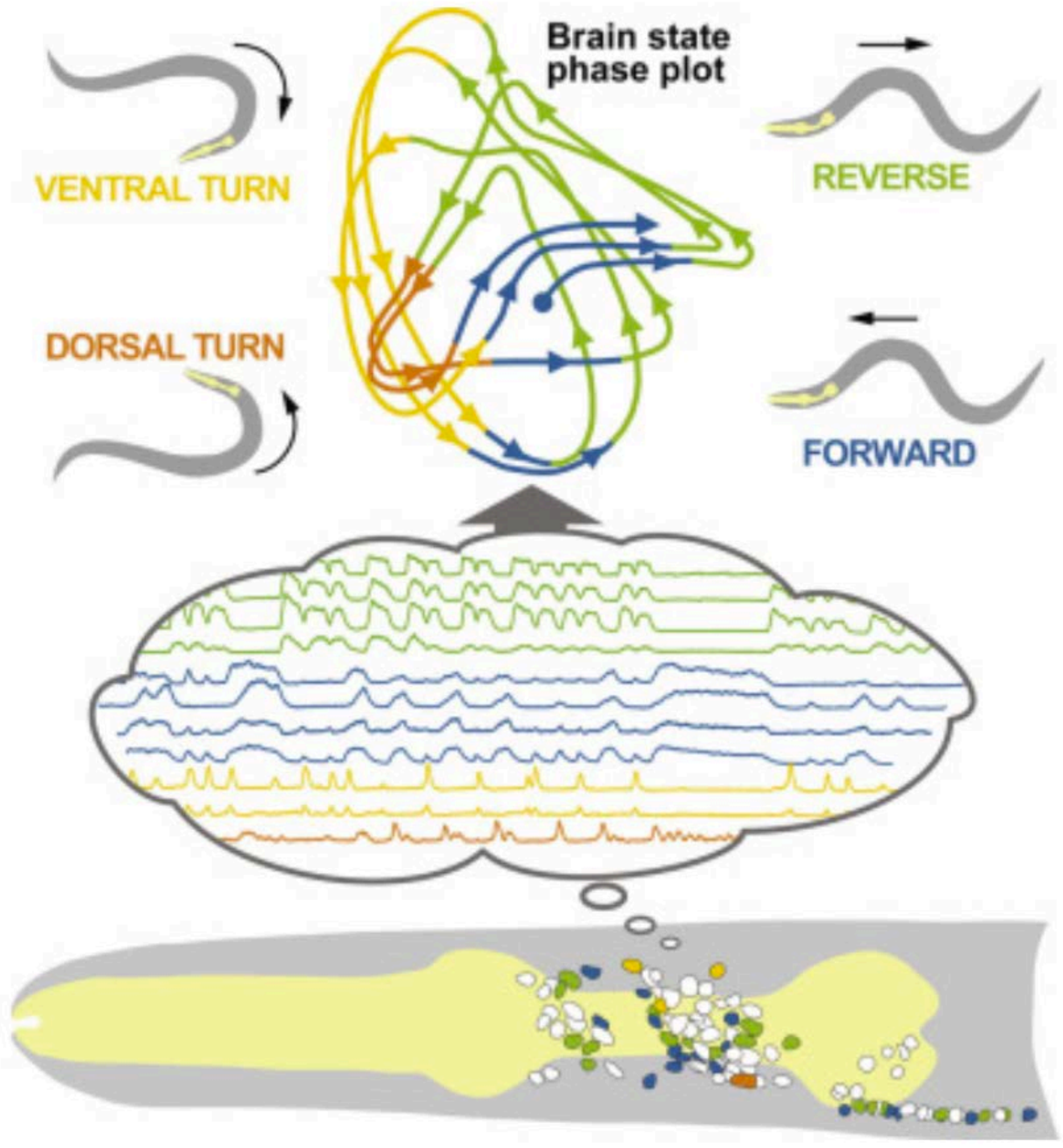


Application: Low-dimensional dynamics of neural population activity



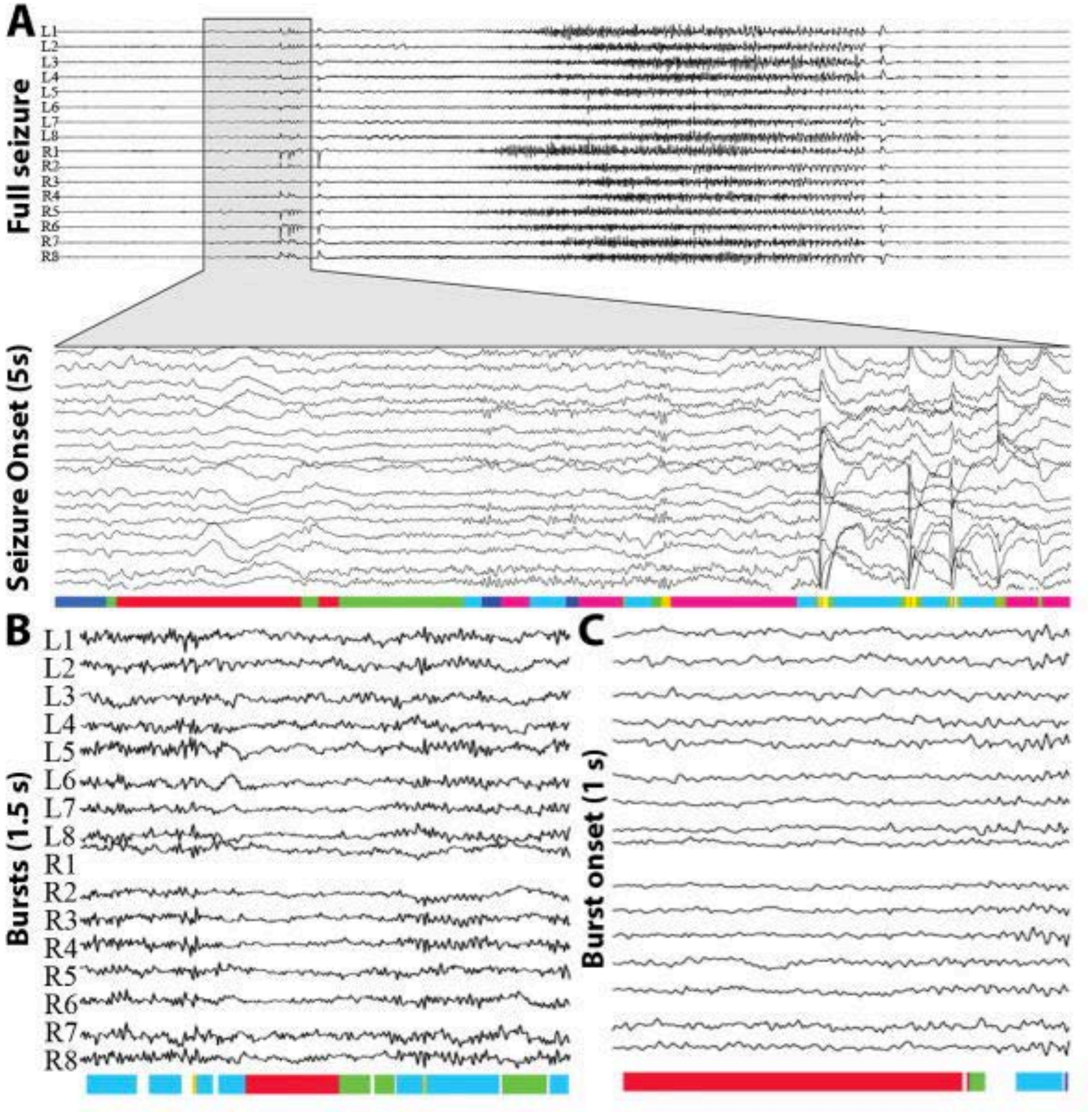


Churchland et al (2012)



Kato et al (2015)

Application: Predicting seizure onset in EEG data

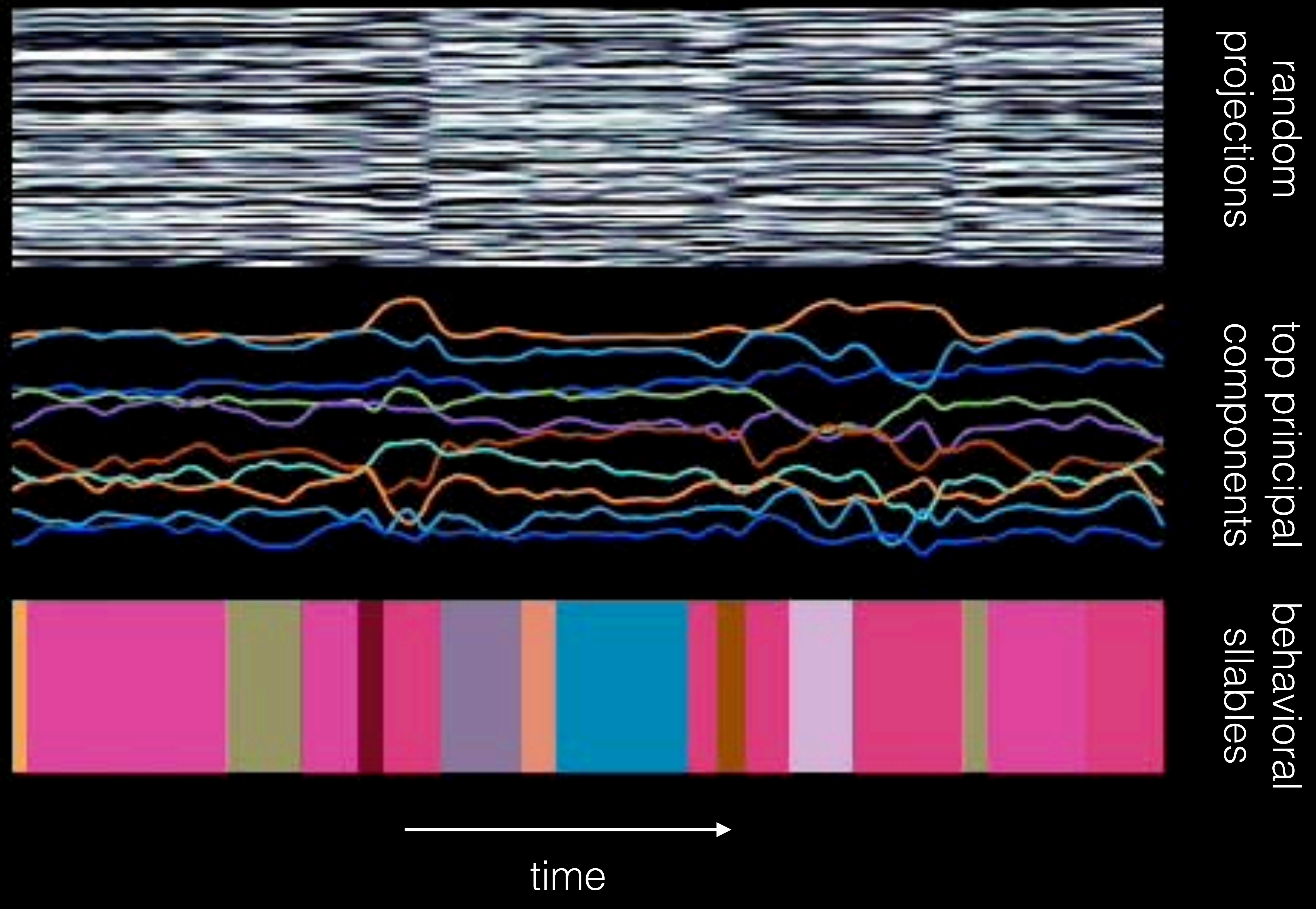


Davis et al (2016)

- EEG dynamics change in characteristic ways at the onset of a seizure.
- State space models detect seizures seconds ahead of unequivocal epileptic activity.
- Better predictions could improve real time by anti-epileptic devices.

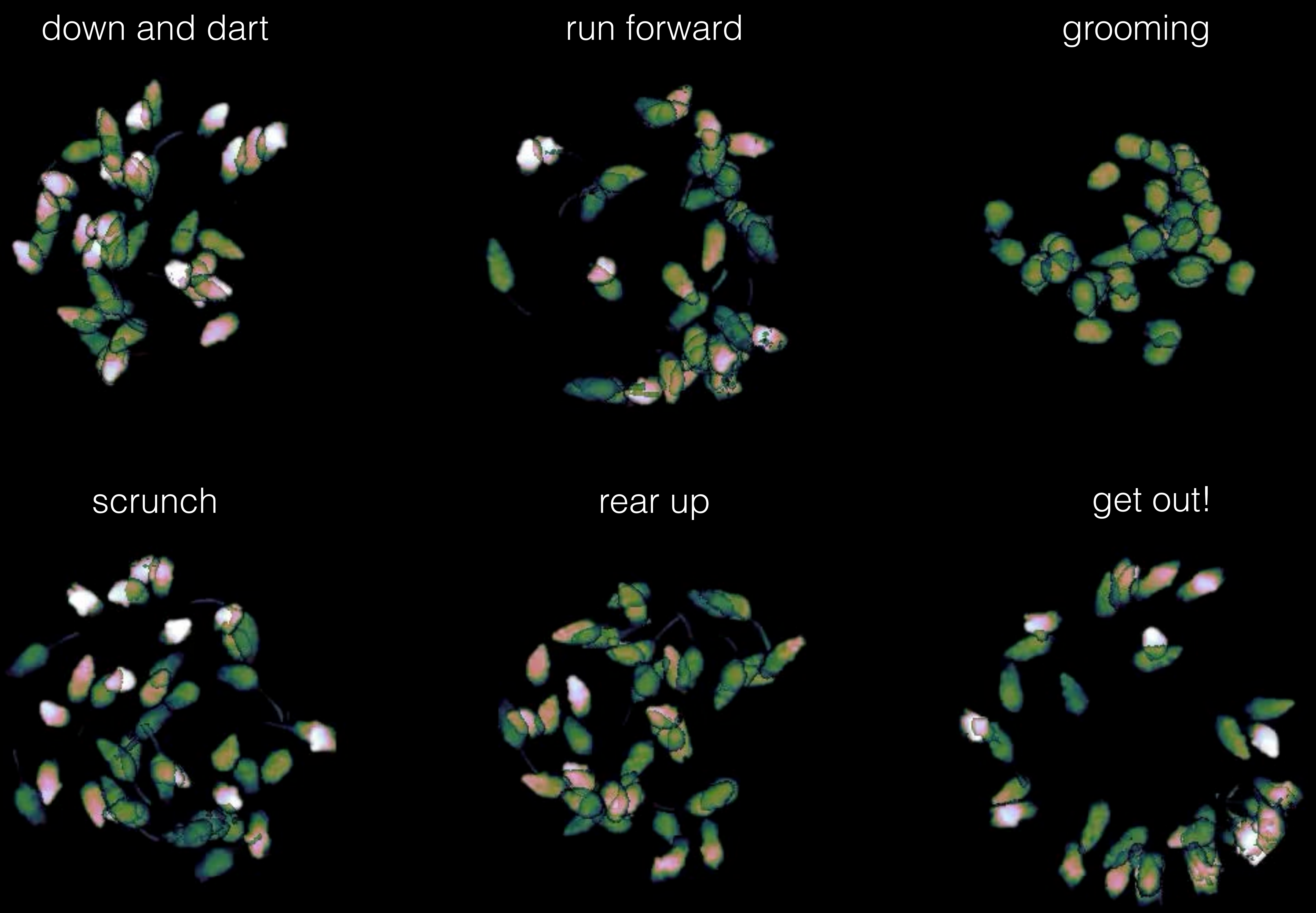
Application: Segmenting behavioral video into “syllables”

centered and aligned video



Wiltschko et al. (2015)

Application: Segmenting behavioral video into “syllables”

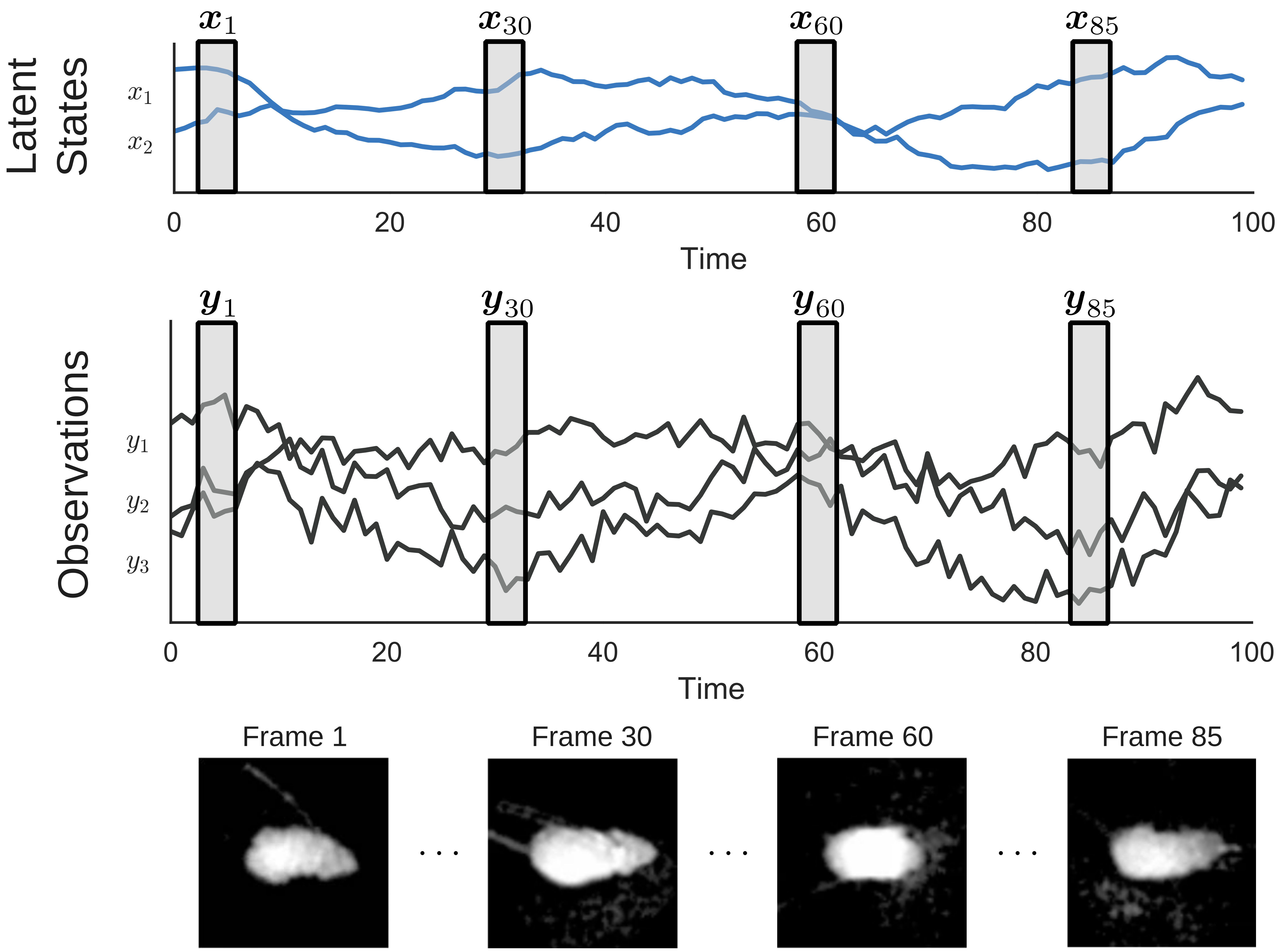


Outline

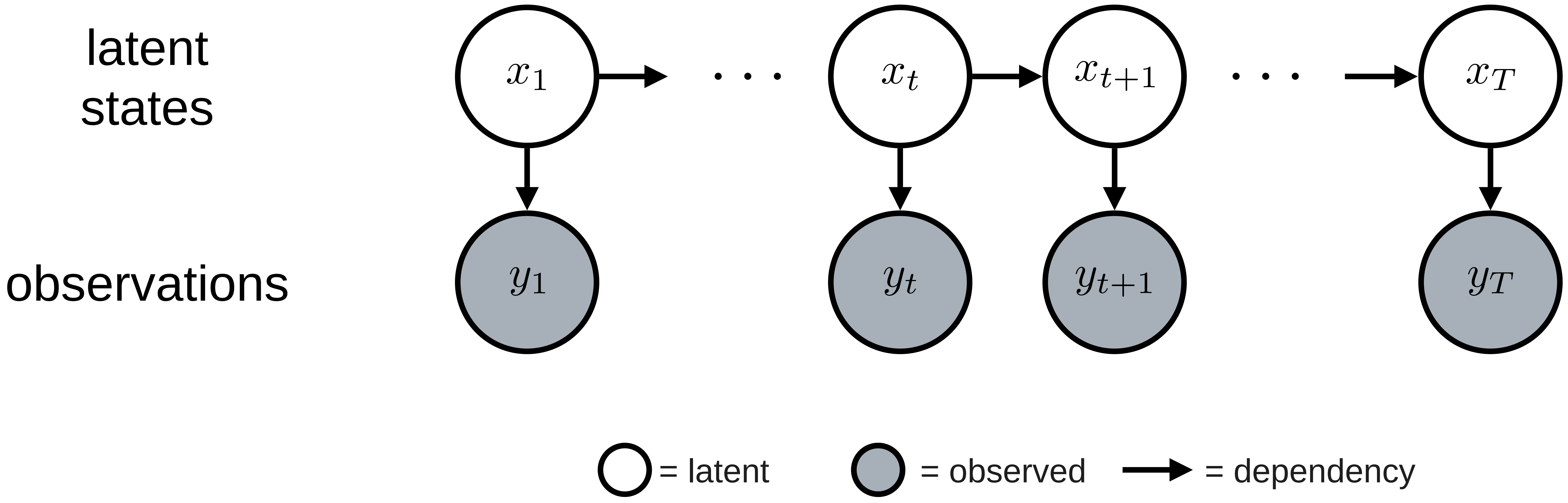
<u>**Part I: Foundations**</u>

- Motivating Examples
- **State Space Models (SSMs)**
  - Hidden Markov Models
  - Linear Dynamical Systems
  - Nonlinear & Switching Linear Dynamical Systems
- Learning and Inference Algorithms
  - Expectation-Maximization
  - Message Passing
  - Approximate Inference (E/UKF, SMC, VI)
- Code Pointers

State Space Models (SSM's)



Anatomy of a state space model



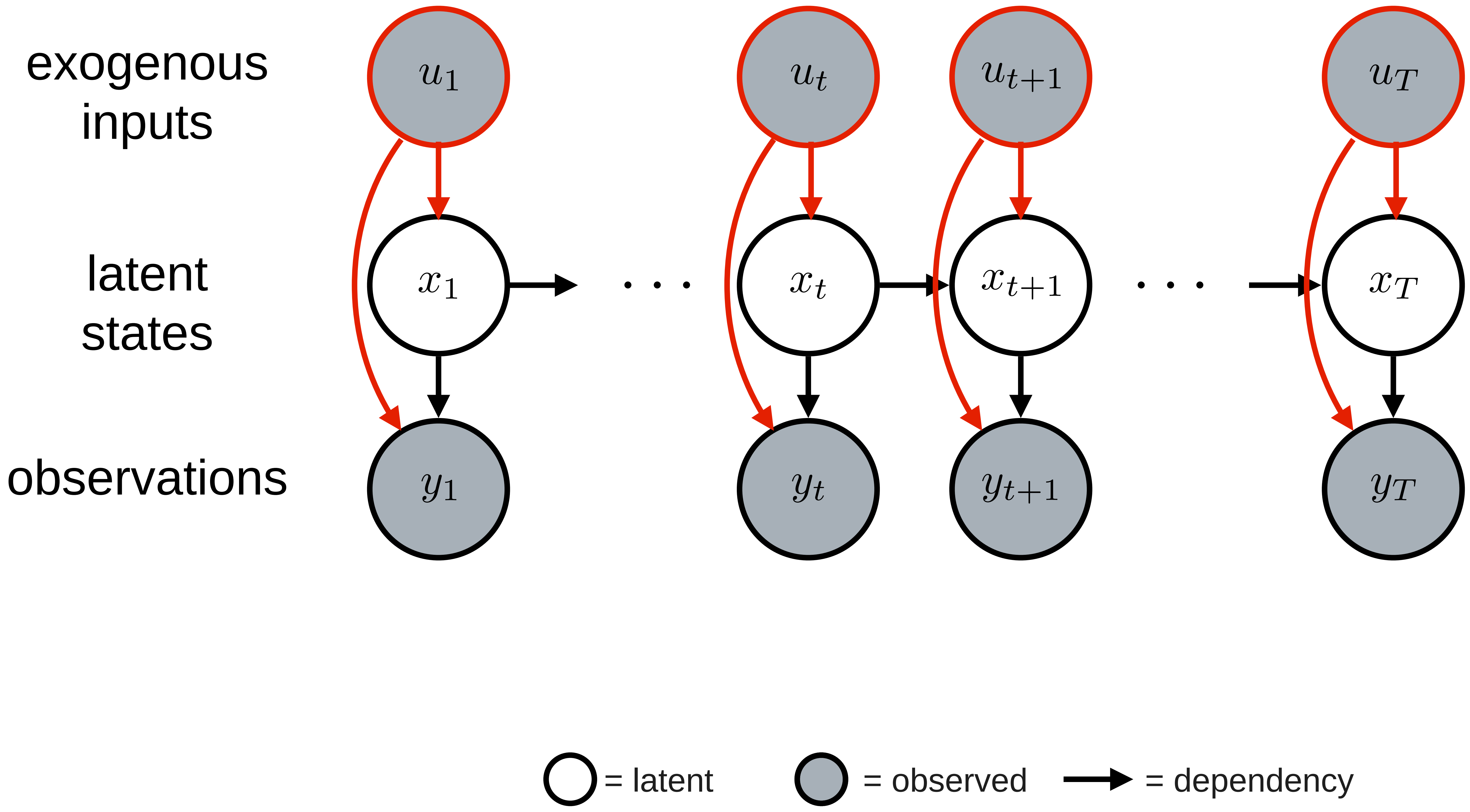
1. Dynamics:
evolution of latent state

$$x_t \sim p(x_t \mid x_{t-1})$$

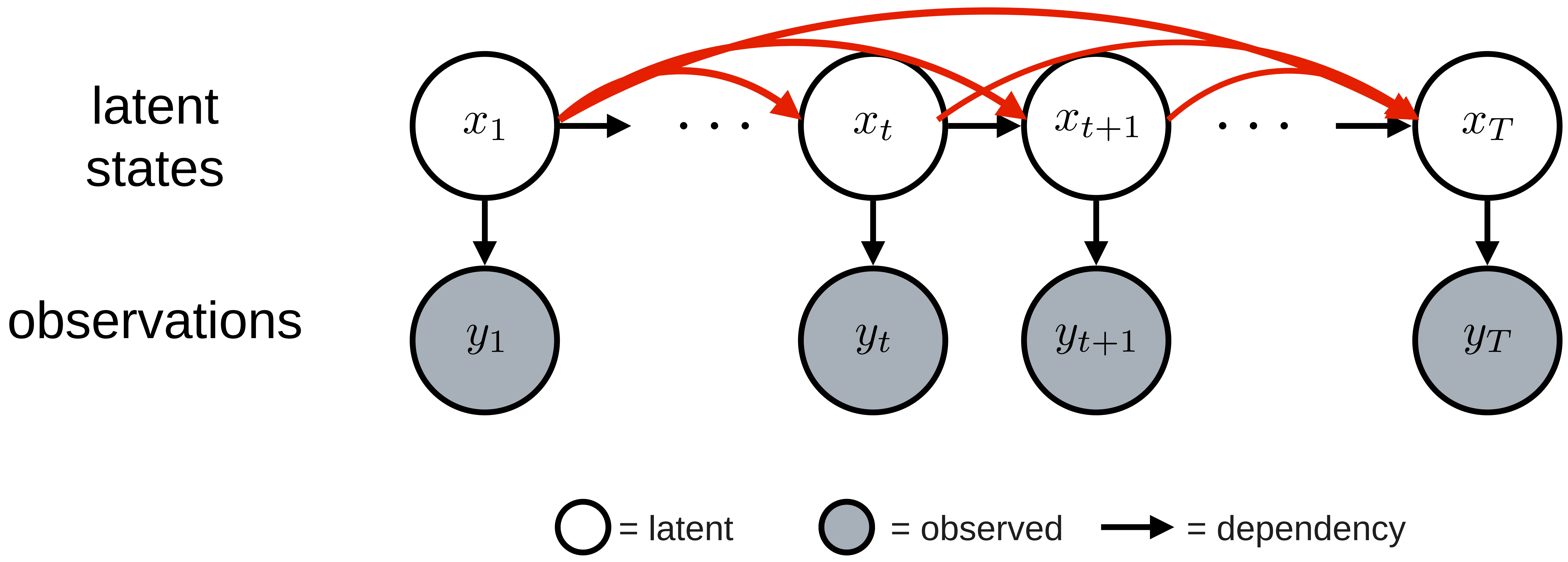
2. Observation model:
connecting states and neural observations

$$y_t \sim p(y_t \mid x_t)$$

Extensions: Exogenous inputs



Extensions: Non-Markovian dynamics



Design decisions

- Type of states and observations:
  - *Discrete, continuous, mixed?*
- Class of observation and dynamics functions:
  - *Linear vs nonlinear? Any constraints?*
- Noise distributions:
  - *Gaussian, Poisson, heavy-tailed, over-dispersed?*
- Discrete vs continuous time:
- Prior distributions; parameter sharing?

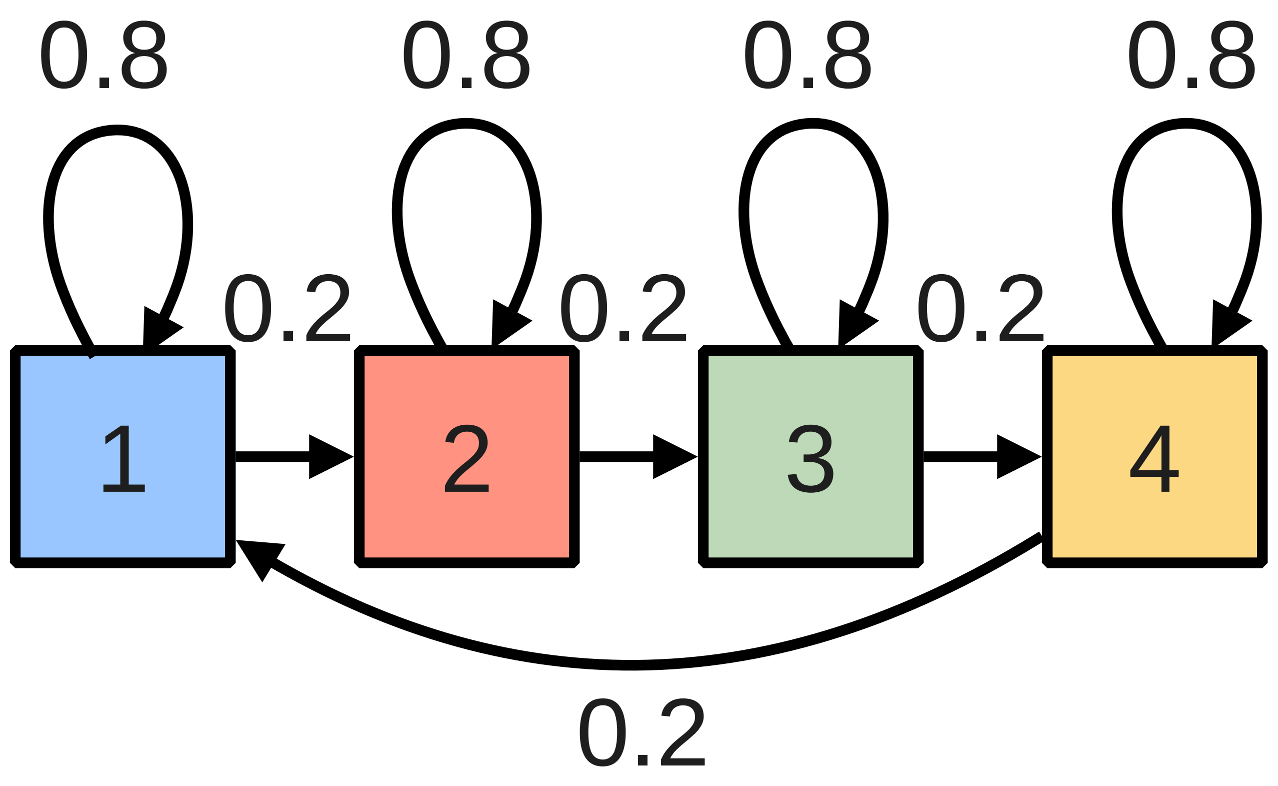
Taxonomy of state space models

Observation Model and Type

Dynamics Model and Type

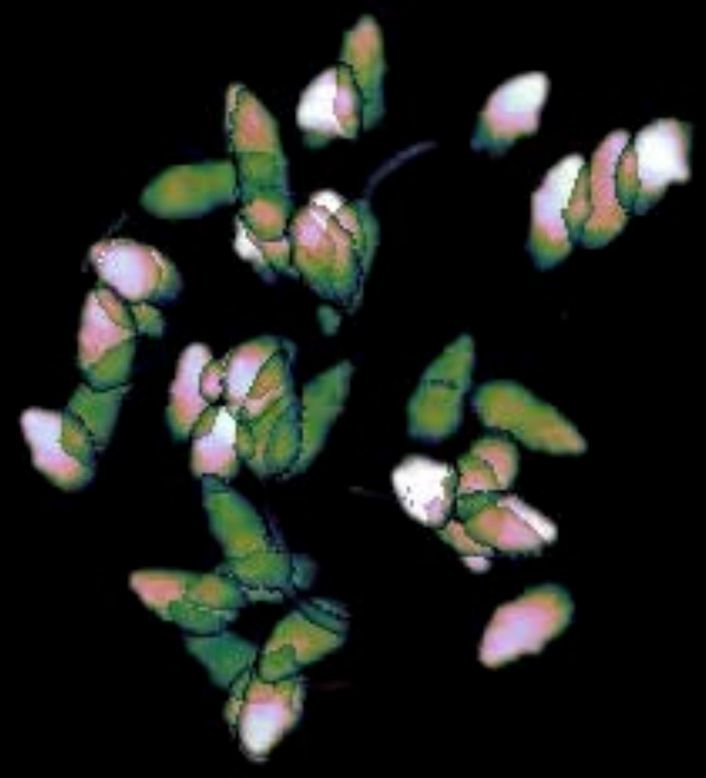
	Continuous Linear	Counts Generalized Linear	Nonlinear observation models
Discrete Linear	**HMM** *Rabiner (1989)*	**HMM** *Rabiner (1989)*	**Structured VAE** *Johnson et al (2016)*
Continuous Linear	**LDS** *Kalman (1960)*	**Poisson LDS** *Smith and Brown (2003), Paninski et al (2010), Macke et al (2011)*	**Deep PfLDS** *Archer et al (2016), Gao et al (2016)*
Mixed Switching Linear	**SLDS** *Ghahramani and Hinton (1996)* *Murphy (1998)*	**Poisson SLDS** *Petreska et al (2013)*	**Structured VAE** *Johnson et al (2016)*
Mixed Recurrent Linear	**recurrent/augmented SLDS** *Barber (2006); Pachitariu et al (2014); Linderman et al (2017); Nassar et al (2019)*	**rSLDS** *Linderman et al (2017)* *Nassar et al (2019)* *Zoltowski et al (2020)*	**Structured VAE** *Johnson et al (2016)*
Continuous Nonlinear (parametric)	**NLDS, e.g. Hodgkin-Huxley** *Ahrens, Huys, Paninski (2006)* *Huys and Paninski (2009)*	**NLDS, e.g. Hodgkin-Huxley** *Meng, Kramer, Eden (2011)*	**GPSSM, DKF, LFADS, VIND** *Frigola et al (2013) , Krishnan et al (2015), Sussillo et al (2016), Hernandez et al (2018), Pandarinath et al (2018)*
Continuous Nonlinear (smoothing)	**GPFA** *Yu, Cunningham, et al (2009)*	**vLGP** *Zhao and Park (2017)*	**GPLVM** *Wu et al (2017)*
Continuous Nonlinear (nonparametric)	**GPSSM, DKF, LFADS, VIND** *Frigola et al (2013) , Krishnan et al (2015), Sussillo et al (2016), Hernandez et al (2018)*	**GPSSM, DKF, LFADS, VIND** *Frigola et al (2013) , Krishnan et al (2015), Sussillo et al (2016), Hernandez et al (2018)*	**GPSSM, DKF, LFADS, VIND** *Frigola et al (2013) , Krishnan et al (2015), Sussillo et al (2016), Hernandez et al (2018), Pandarinath et al (2018)*

Hidden Markov Models

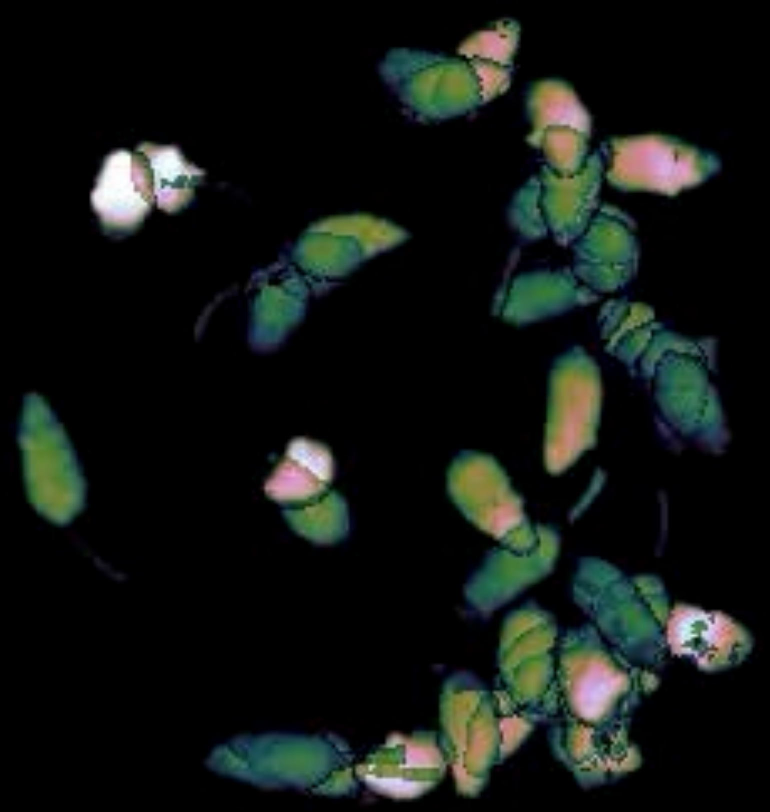


Behavioral "Syllables"

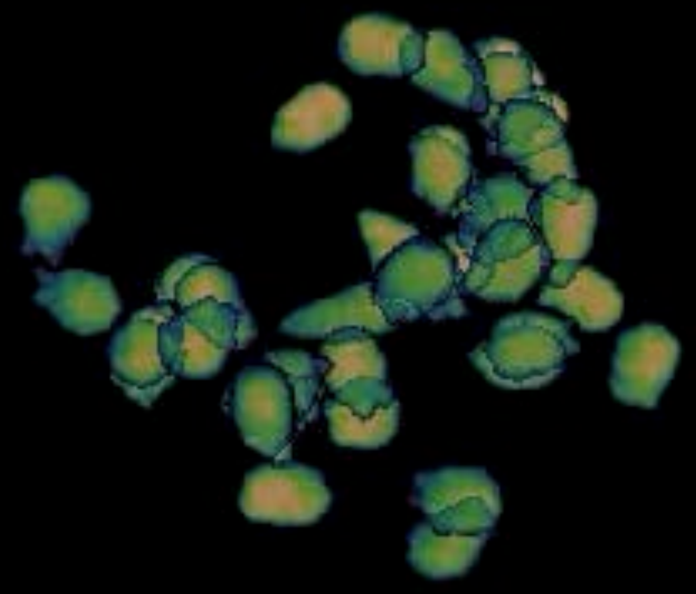
down and dart



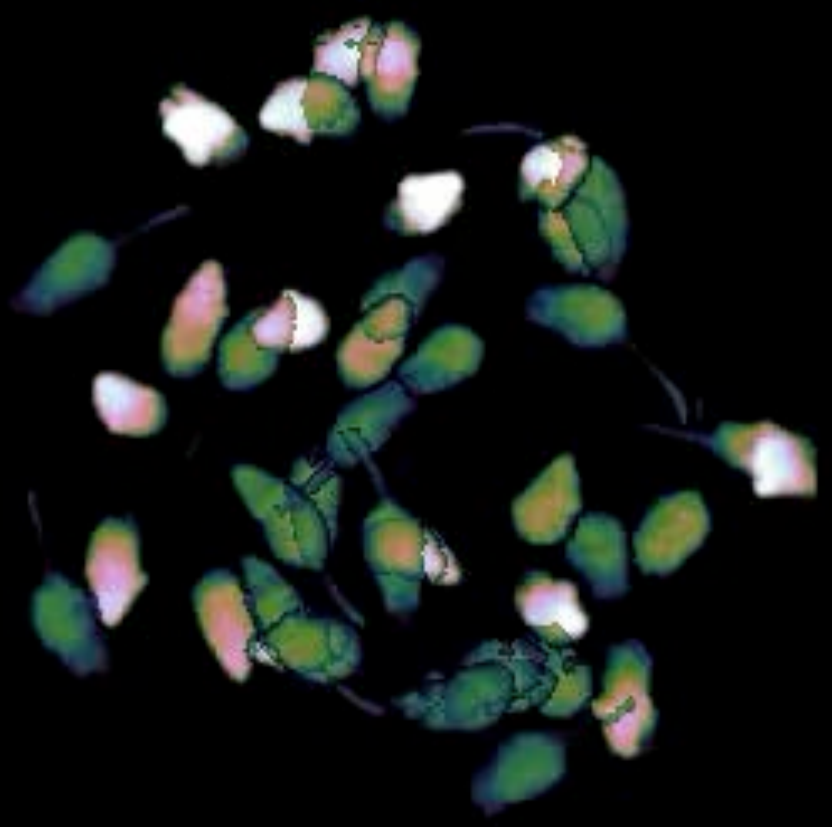
run forward



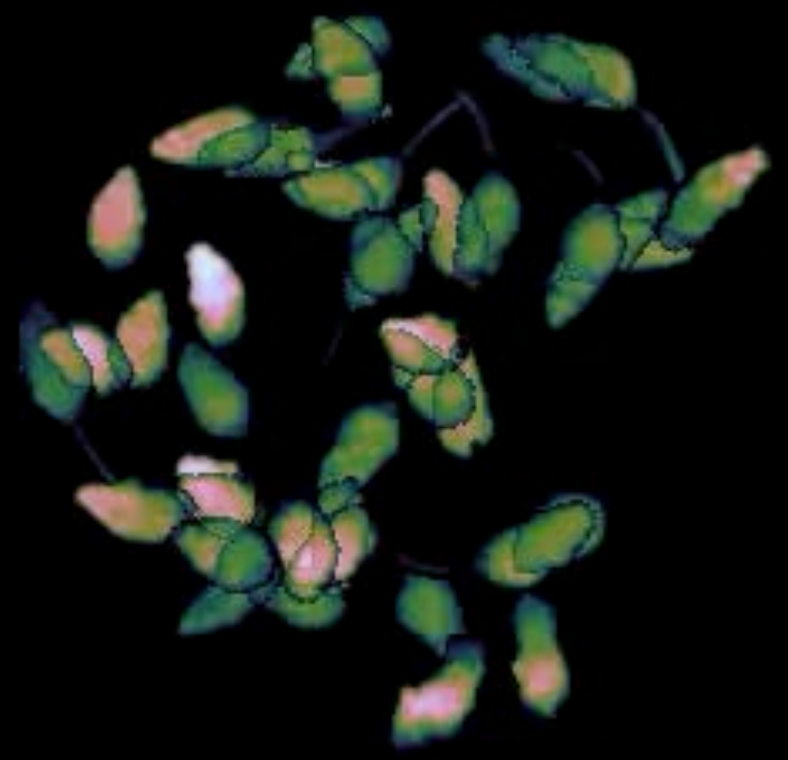
grooming



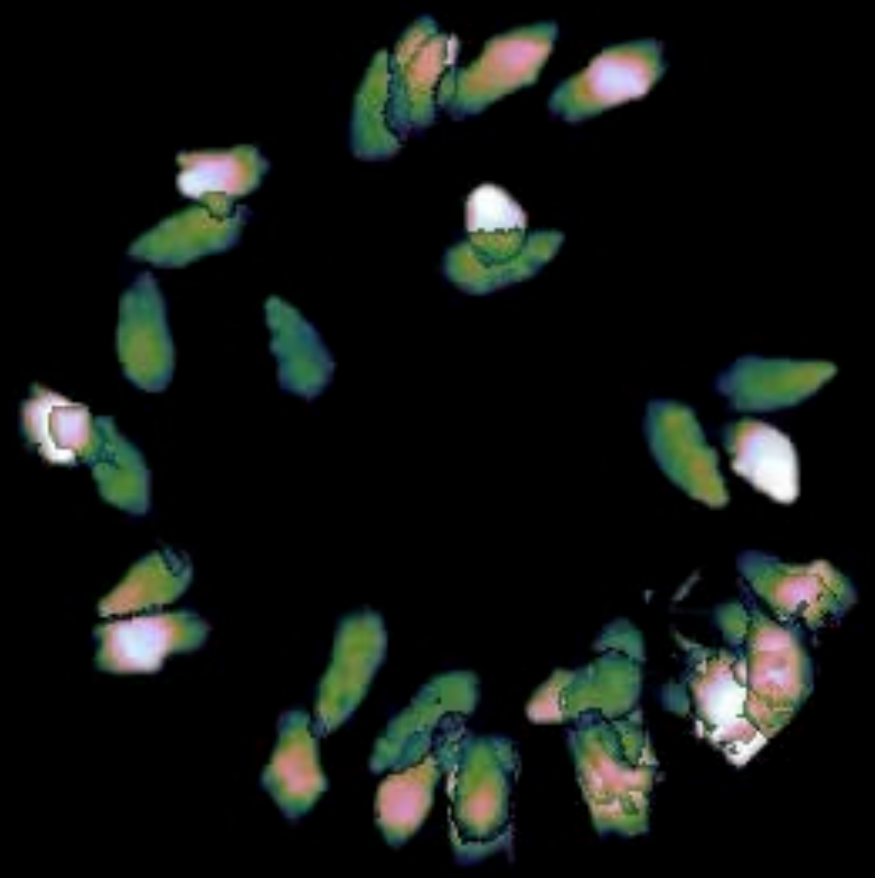
scrunch



rear up



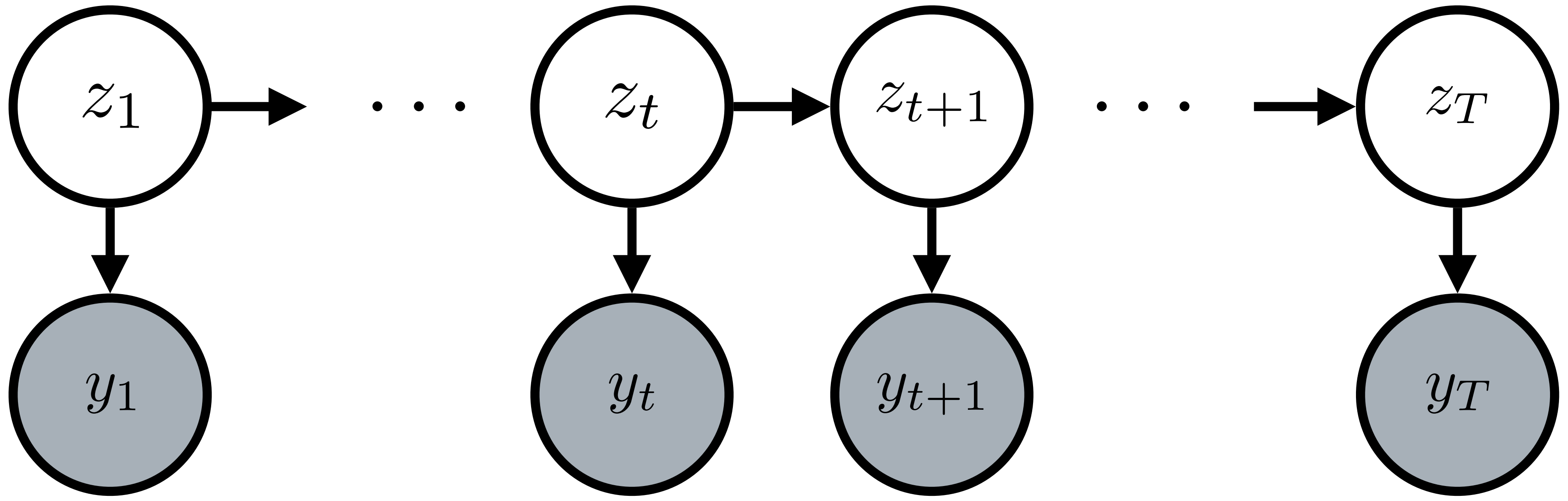
get out!



Hidden Markov Models

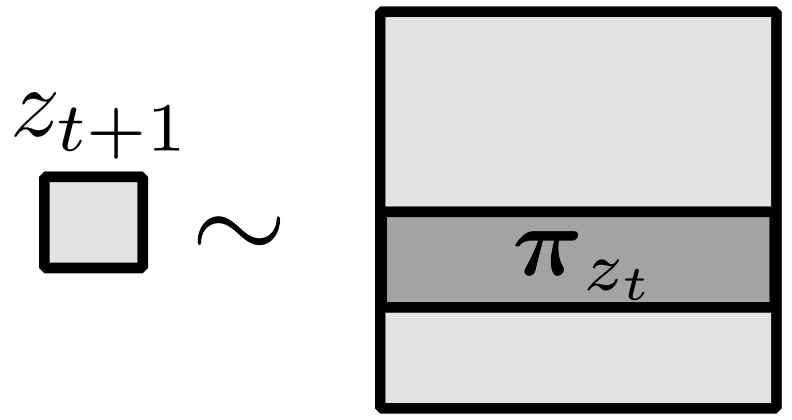
discrete states

neural observations



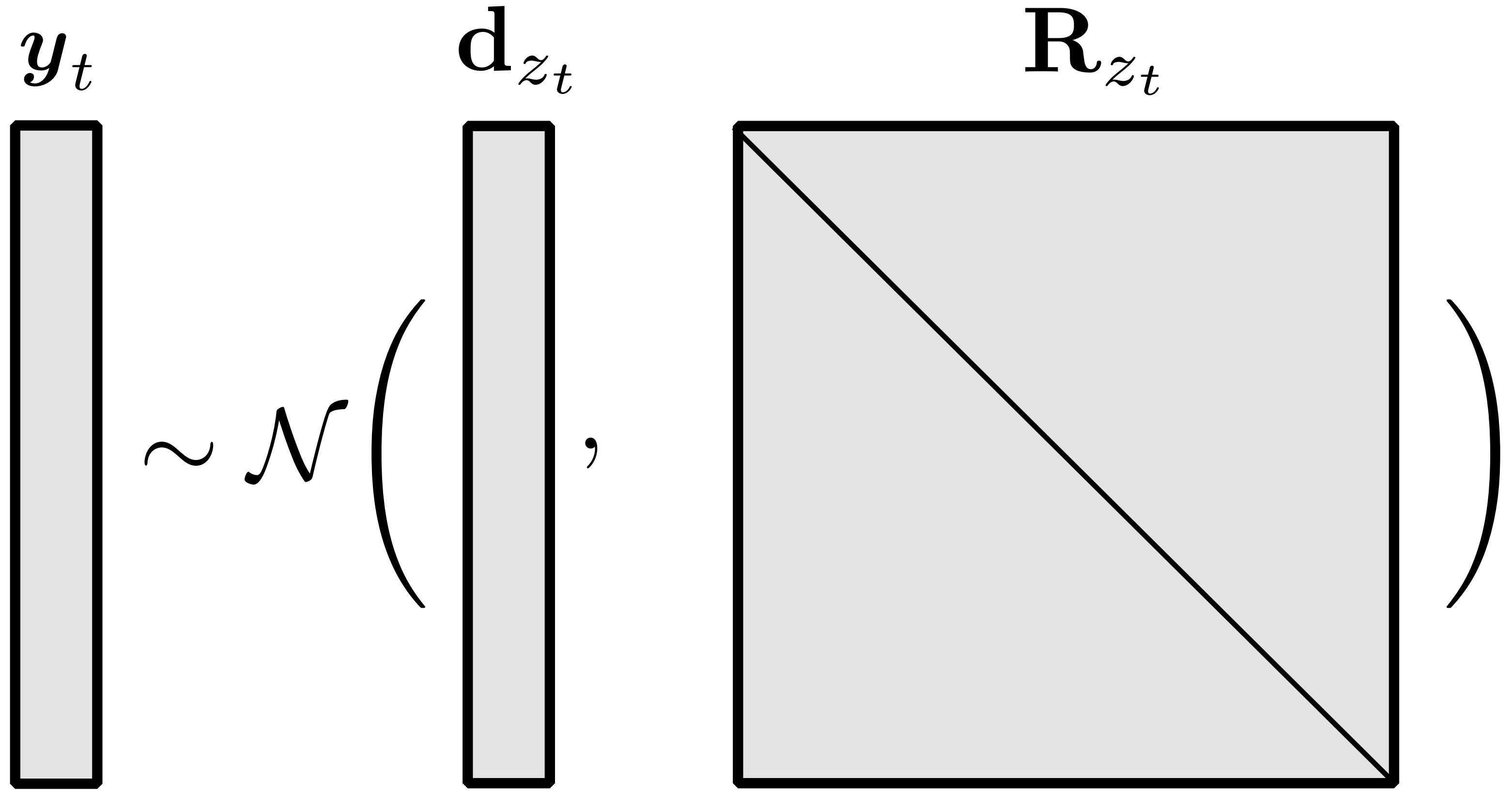
Dynamics:
transition matrix

$$z_{t+1} \mid z_t \sim \mathrm{Cat}(\pi_{z_t})$$



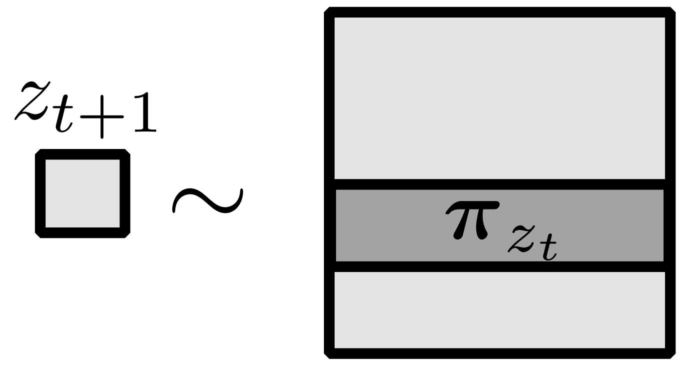
Observation model:
different parameters for each discrete state

$$y_t \sim p(y_t; \theta_{z_t})$$



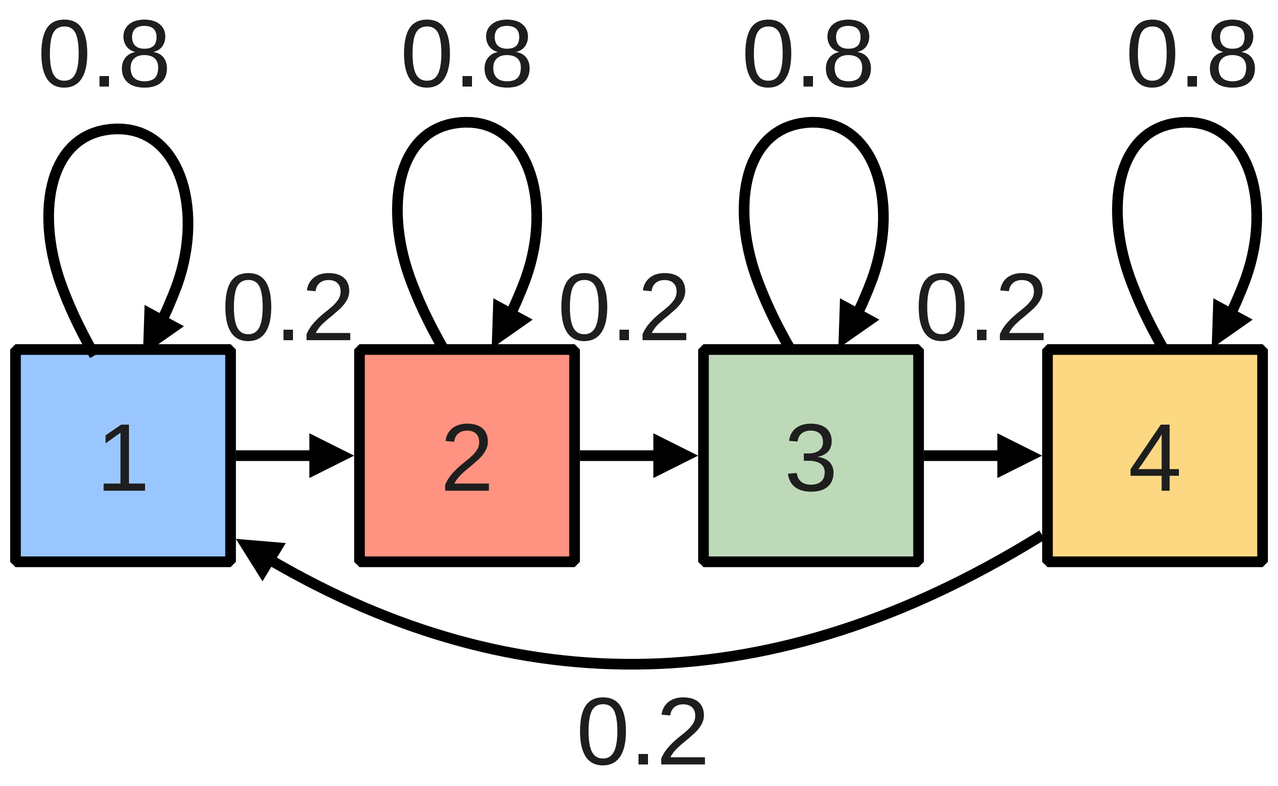
Visualizing discrete dynamics

discrete latent state dynamics

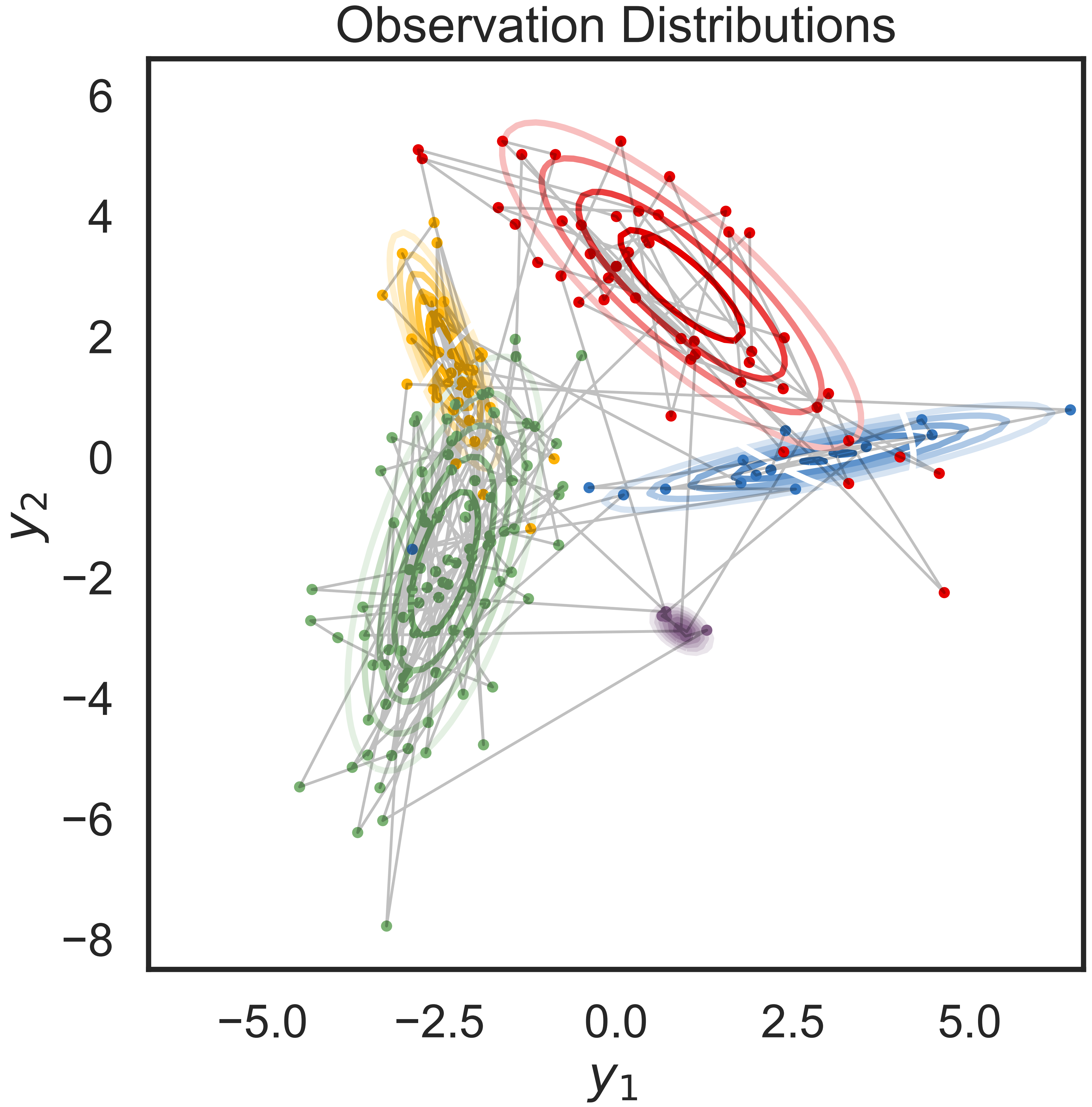


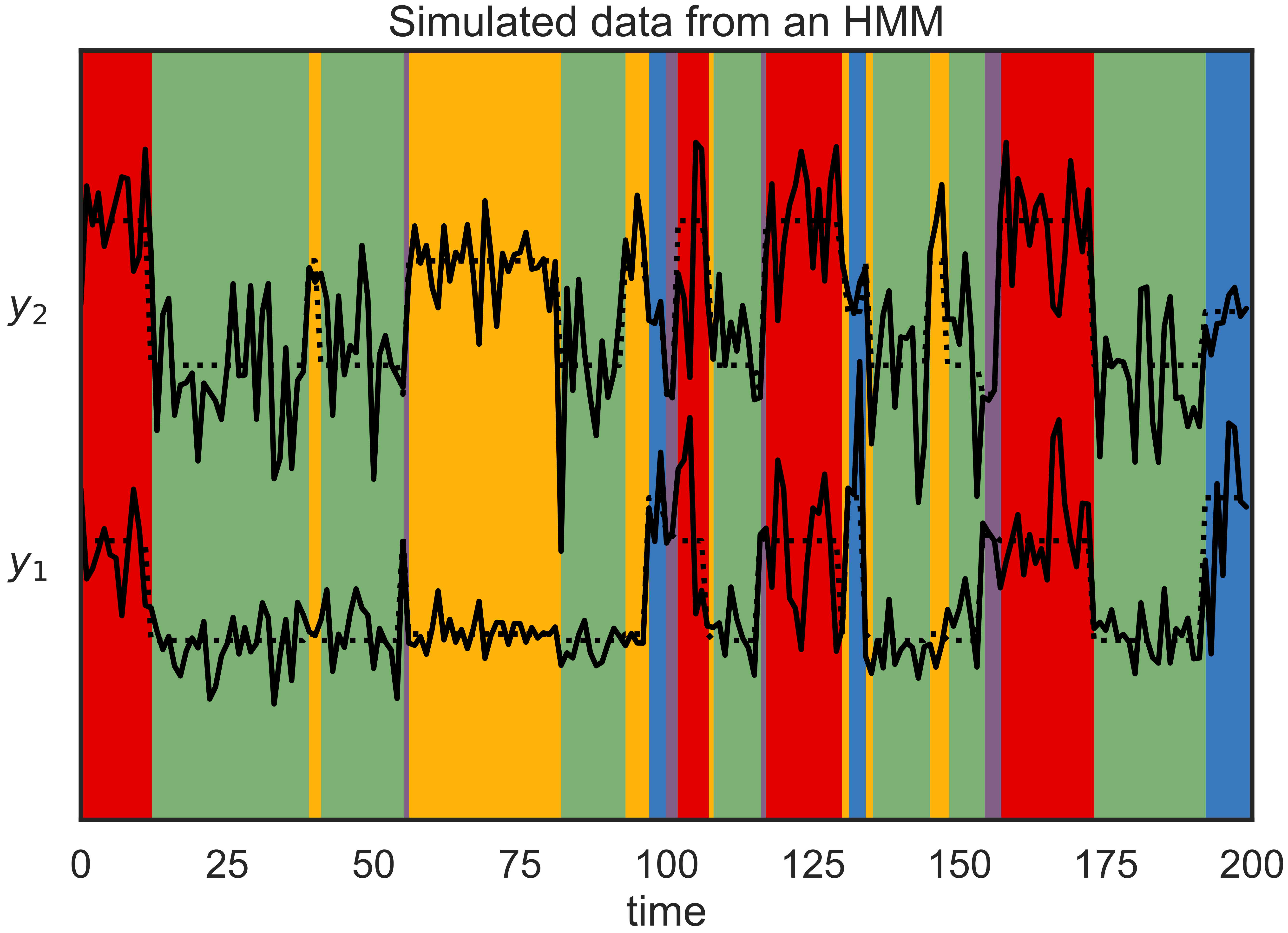
$$\boldsymbol{P} = \begin{bmatrix} - & \boldsymbol{\pi}_1 & - \\ - & \boldsymbol{\pi}_2 & - \\ & \cdots & \\ - & \boldsymbol{\pi}_K & - \end{bmatrix}$$

$$\text{e.g. } \boldsymbol{P} = \begin{bmatrix} 0.8 & 0.2 & 0.0 & 0.0 \\ 0.0 & 0.8 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.8 & 0.2 \\ 0.2 & 0.0 & 0.2 & 0.8 \end{bmatrix}$$

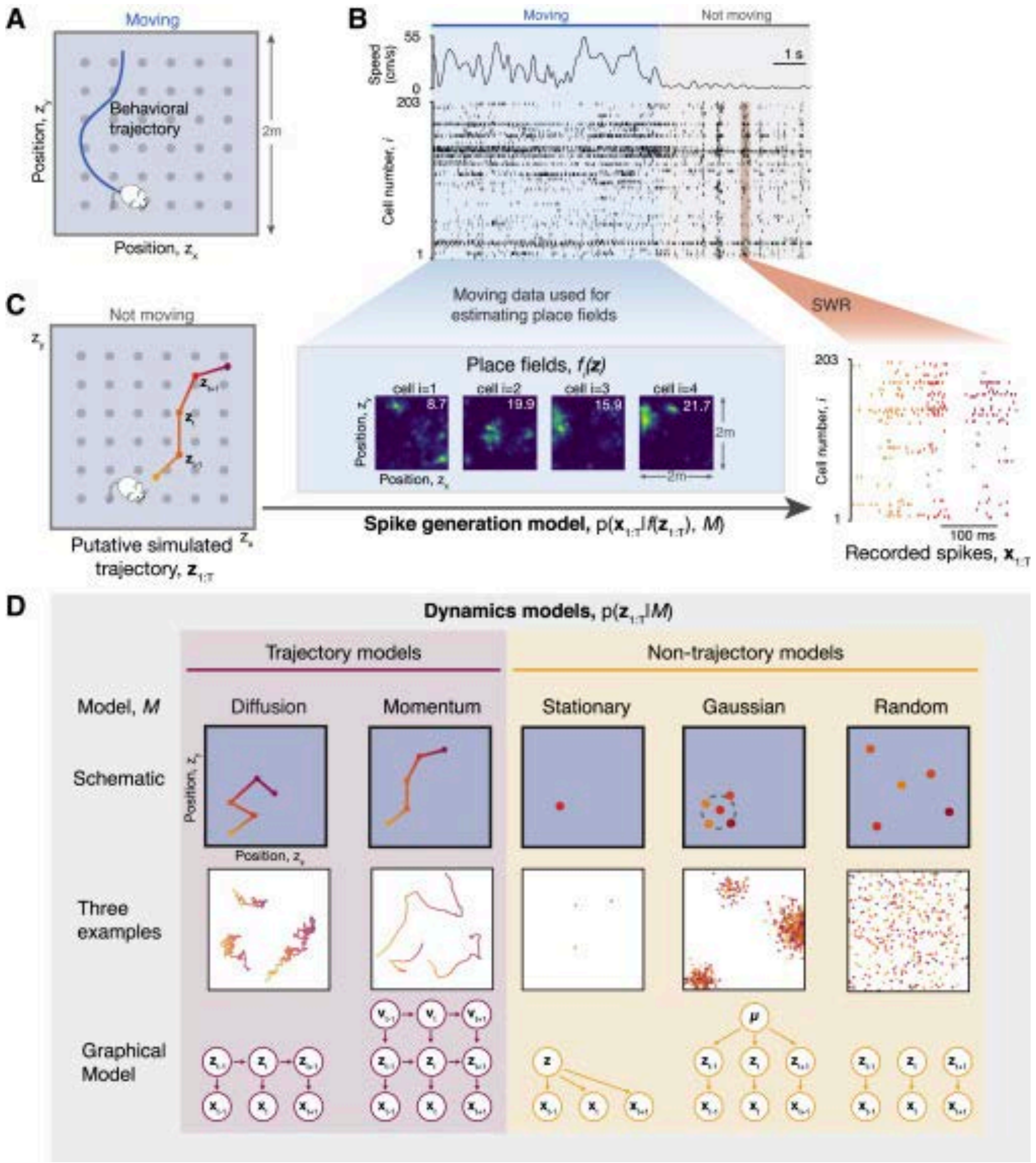


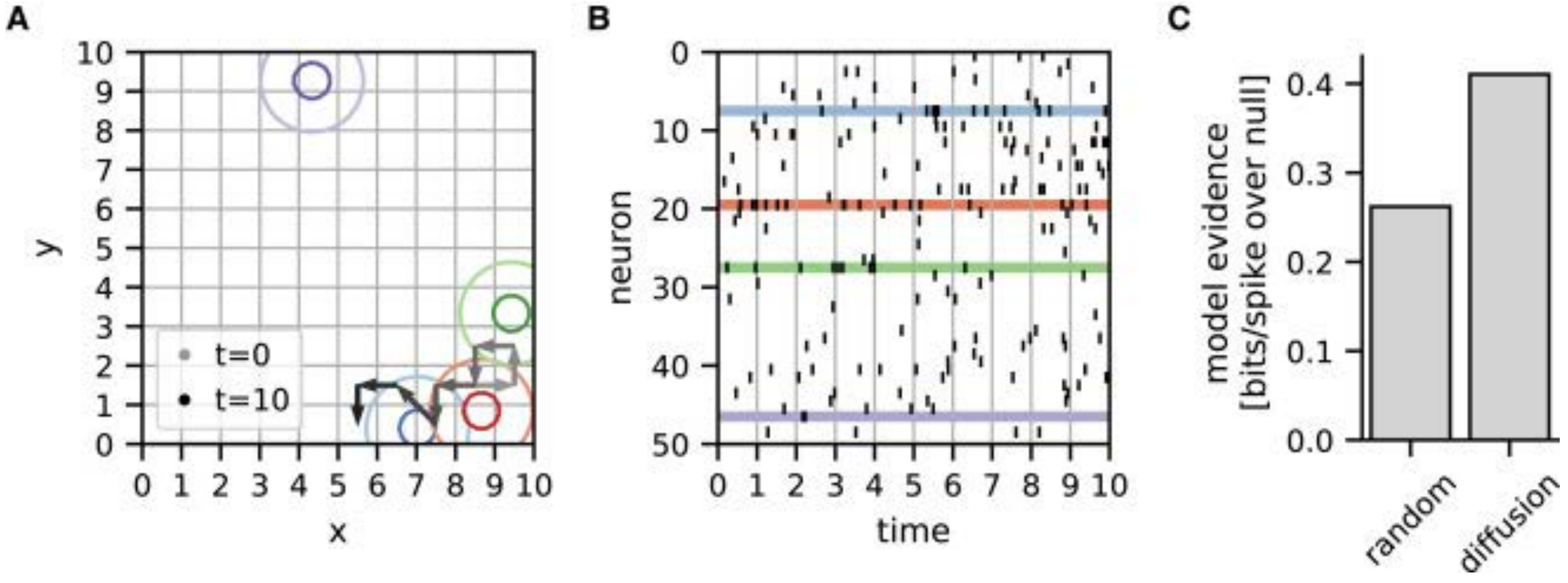
Visualization of a Gaussian HMM





HMMs for characterizing the spatiotemporal structure of SWRs



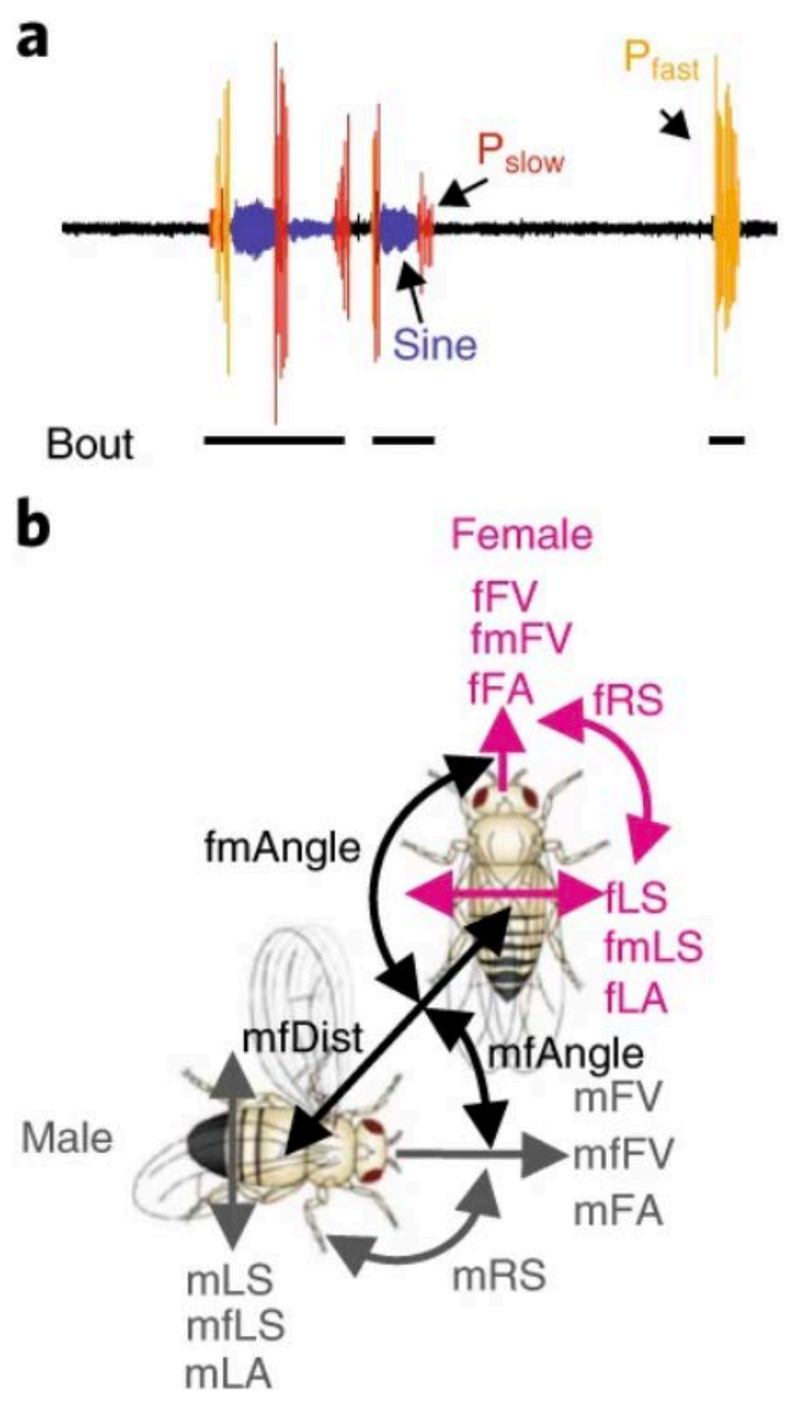


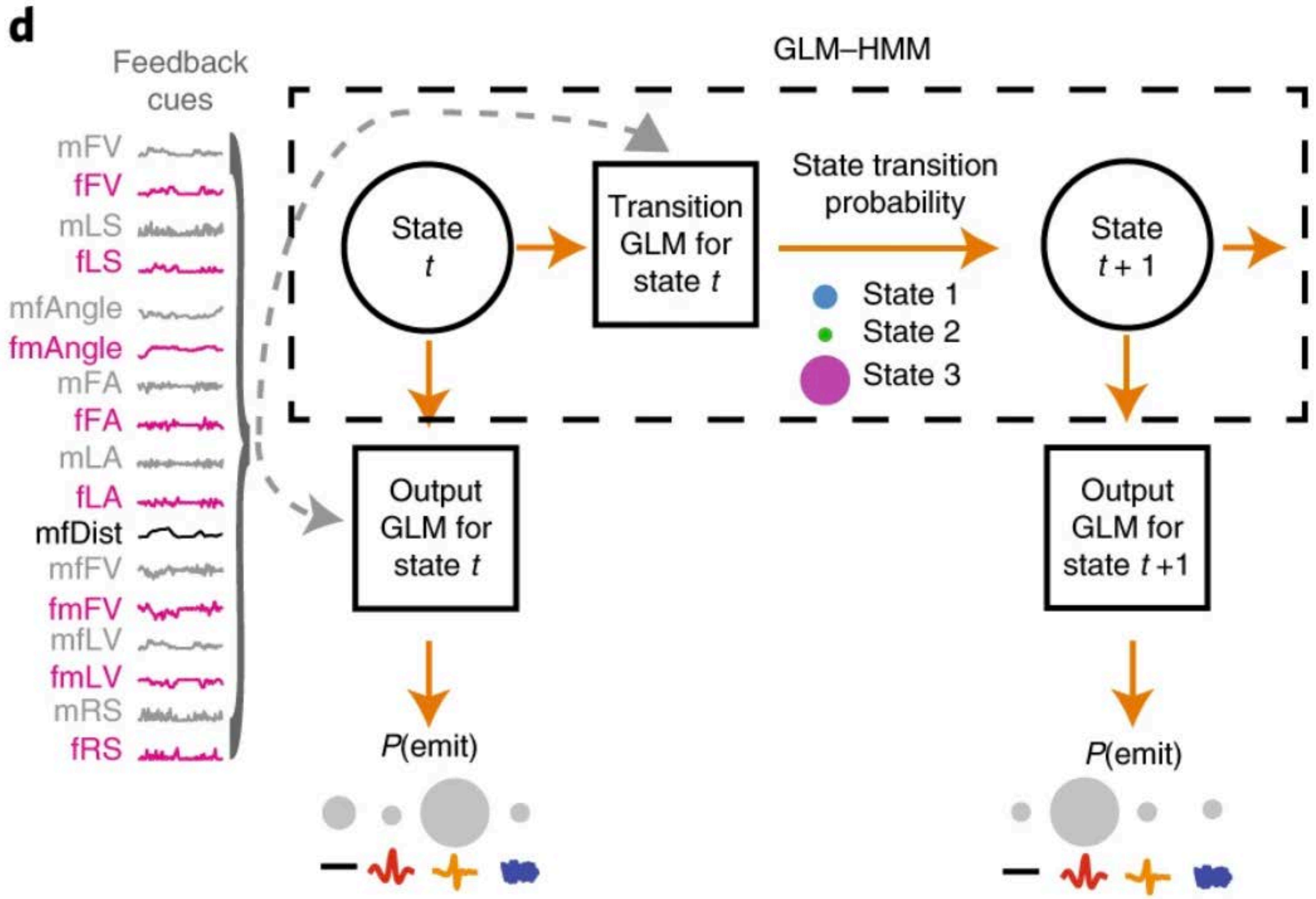
Krause & Drugowitsch (2022)

Linderman (2022)

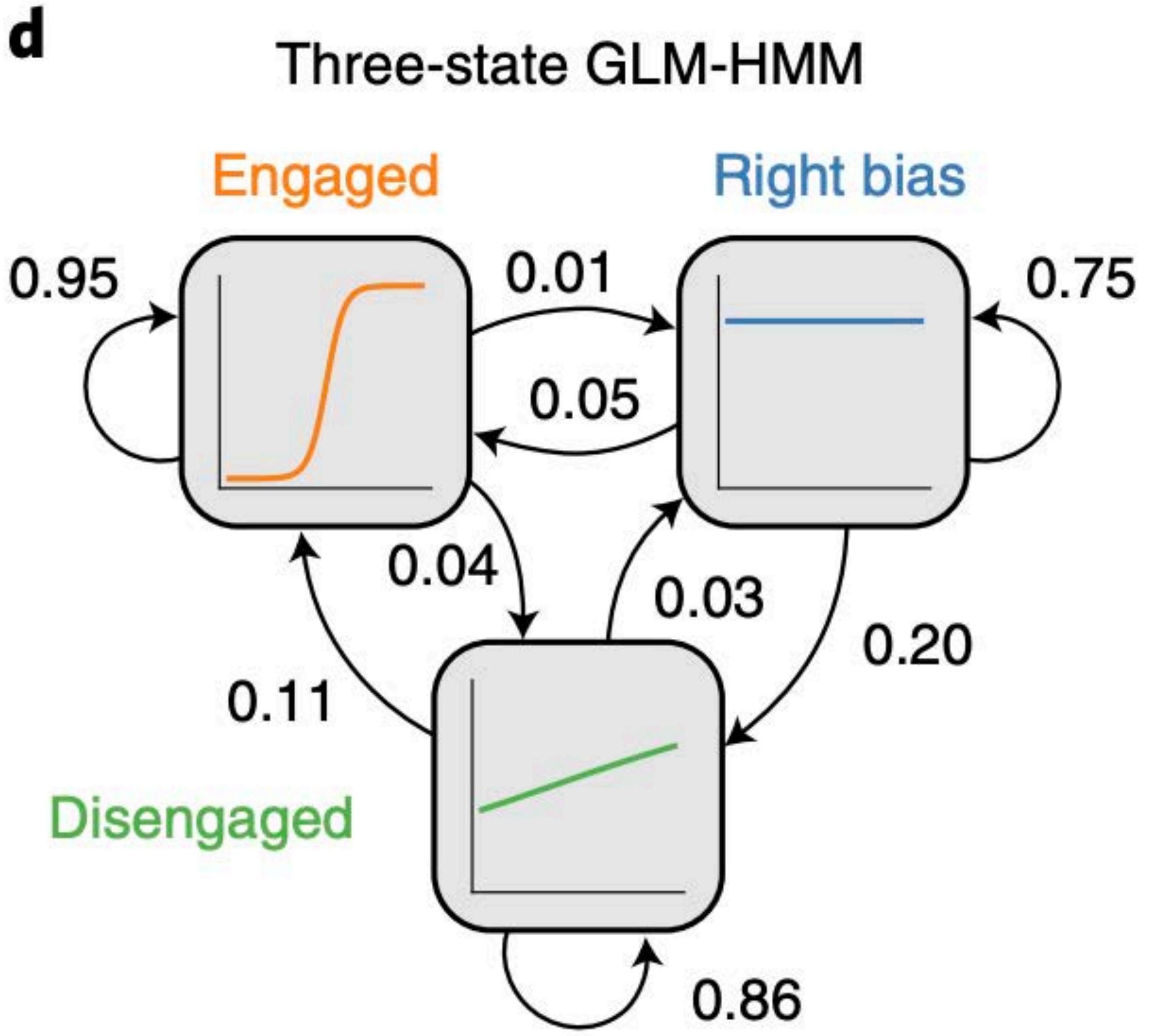
HMM-GLMs for characterizing behavior

Drosophila courtship





Perceptual decision making

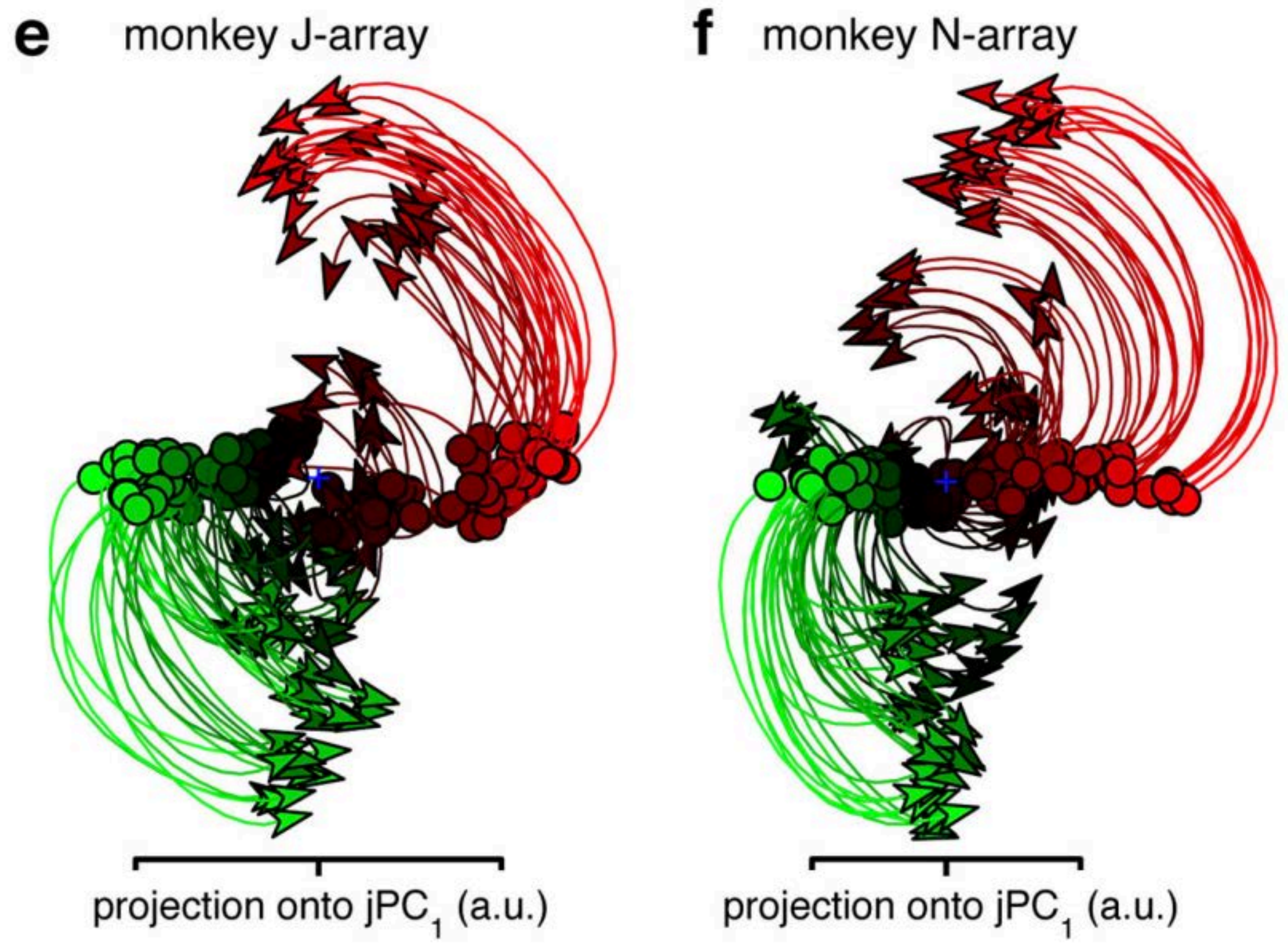


Calhoun et al (2019)

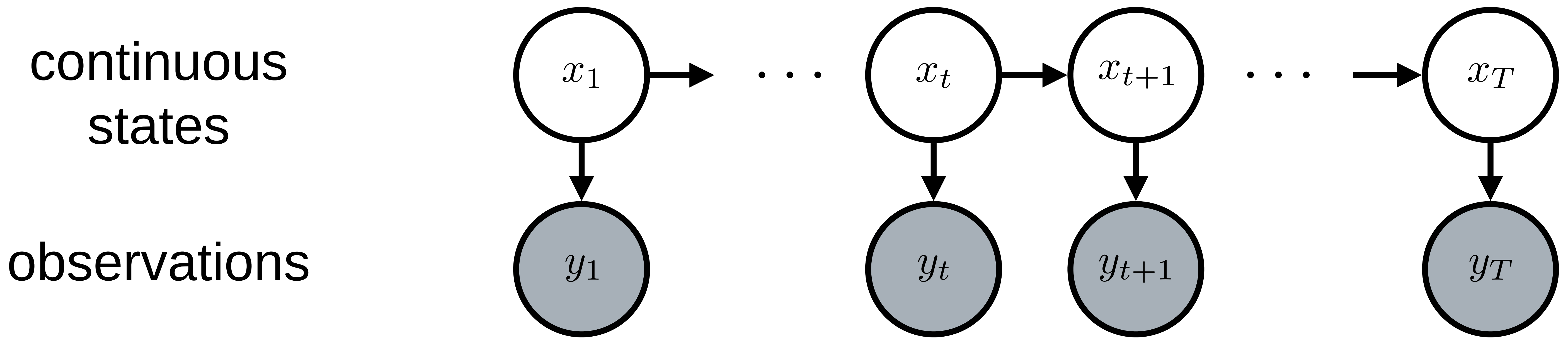
Stone et al (2022)

Ashwood et al (2022)

Linear Dynamical Systems

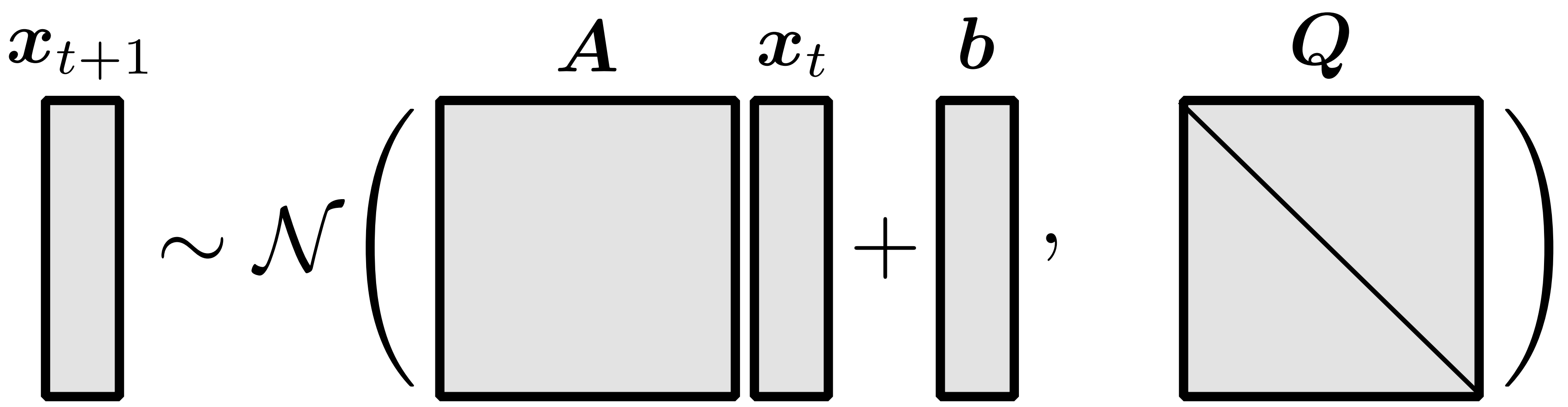


Linear dynamical systems - continuous, sequential latents



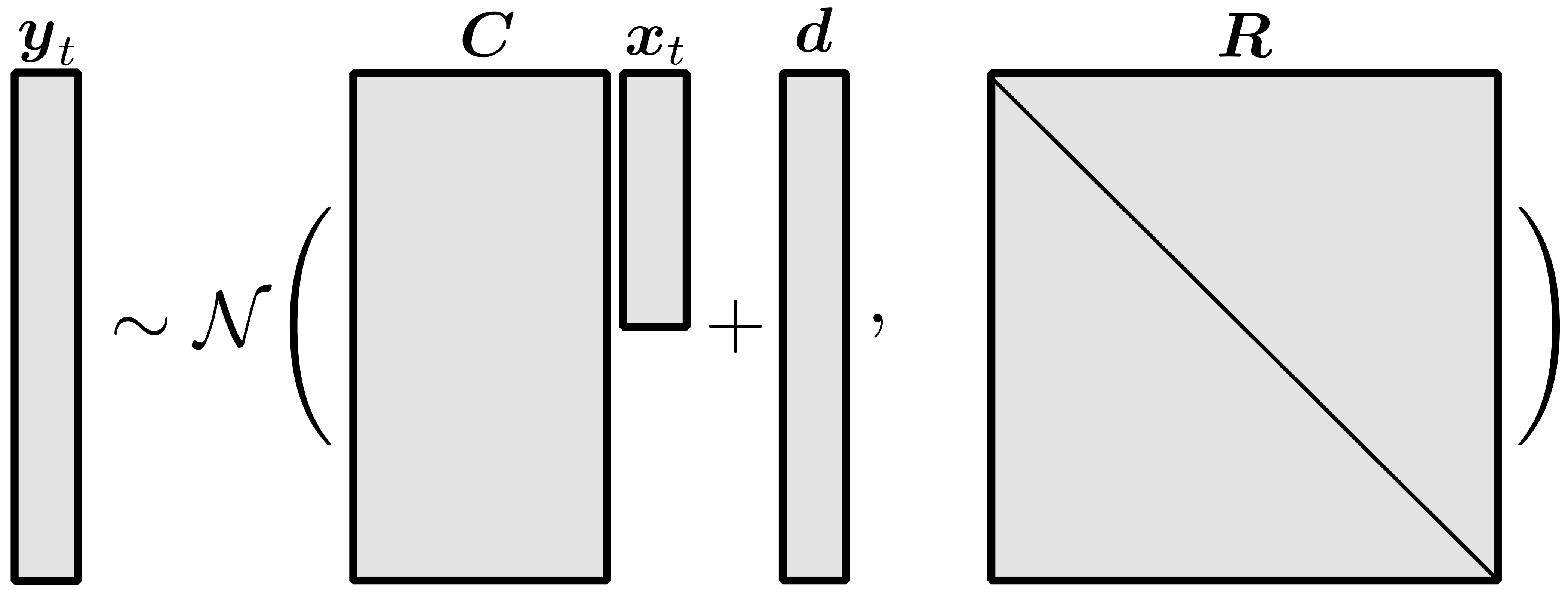
Dynamics:
Linear, Gaussian

$$x_{t+1} \mid x_t \sim \mathcal{N}(Ax_t + b, Q)$$

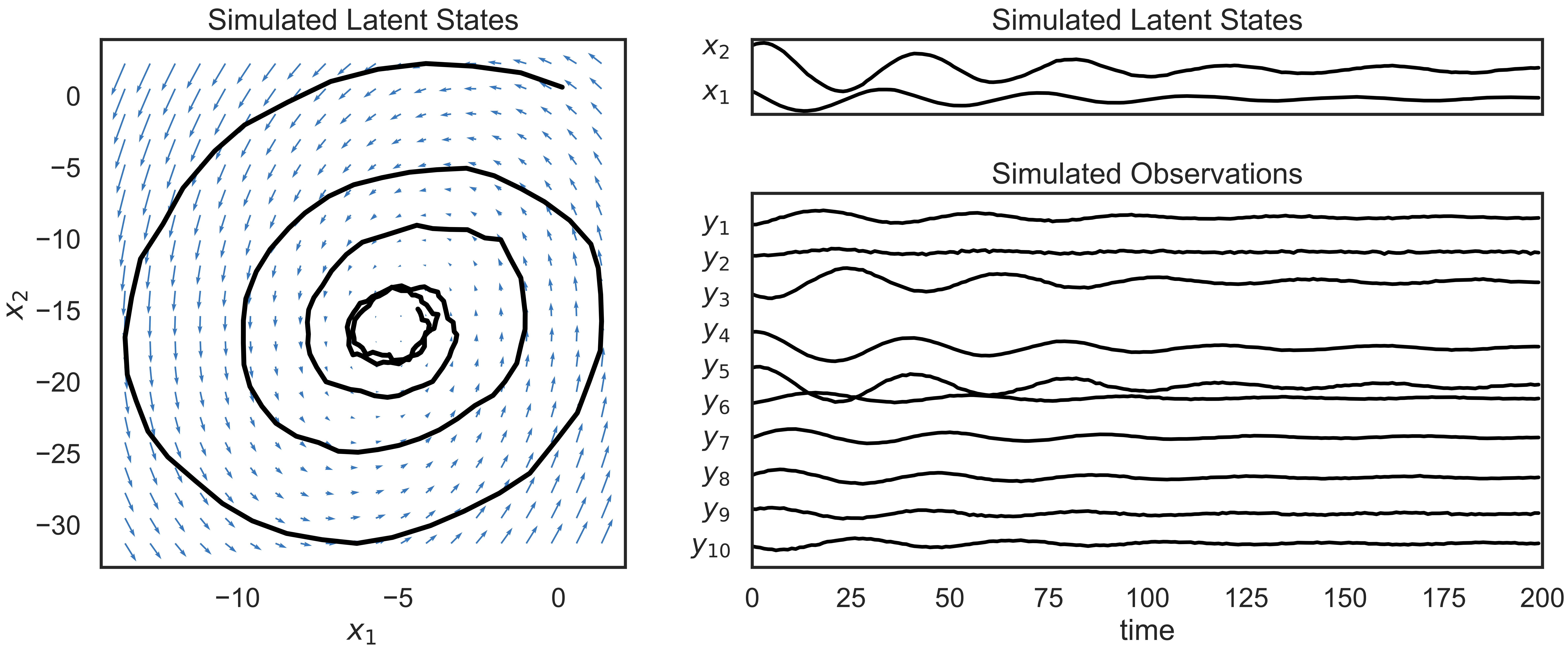


Observation model:
Generalized Linear

$$y_t \mid x_t \sim \mathcal{P}(f(Cx_t + D))$$

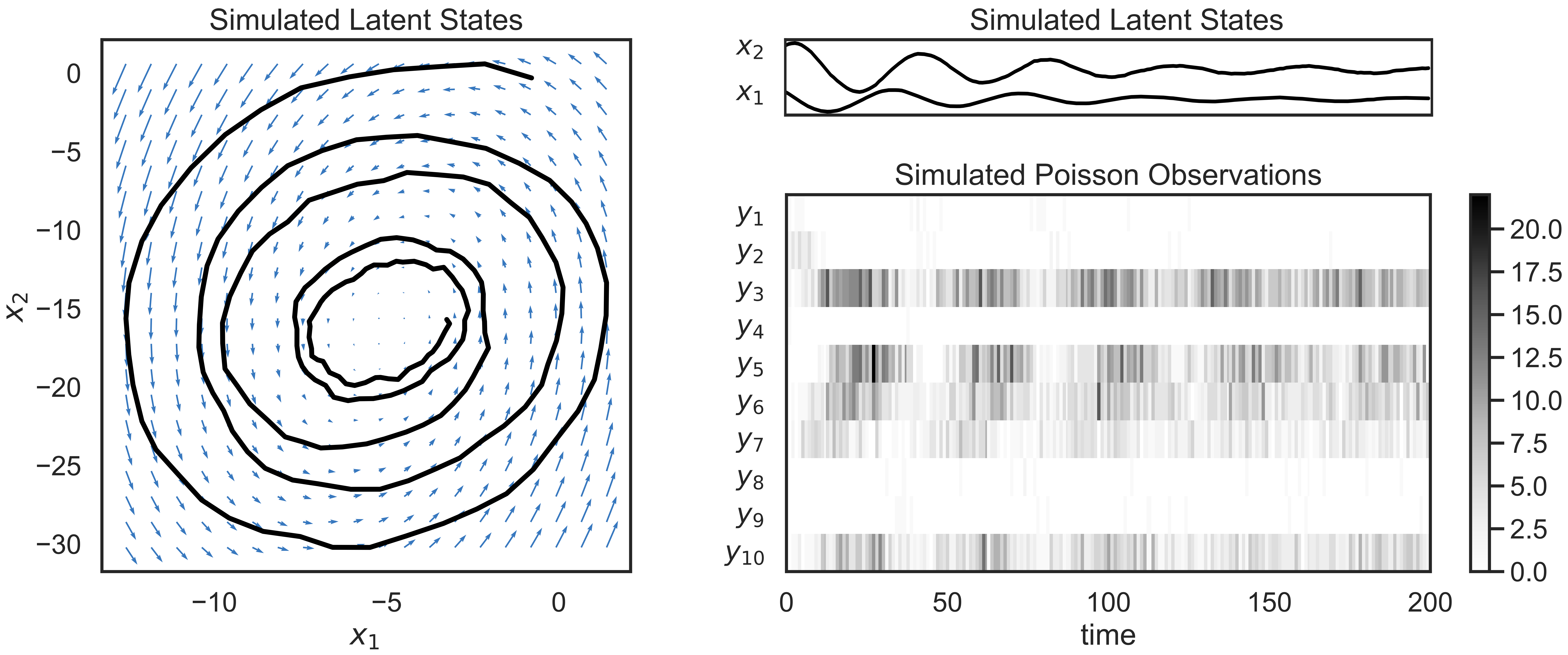


Visualizing a Gaussian LDS

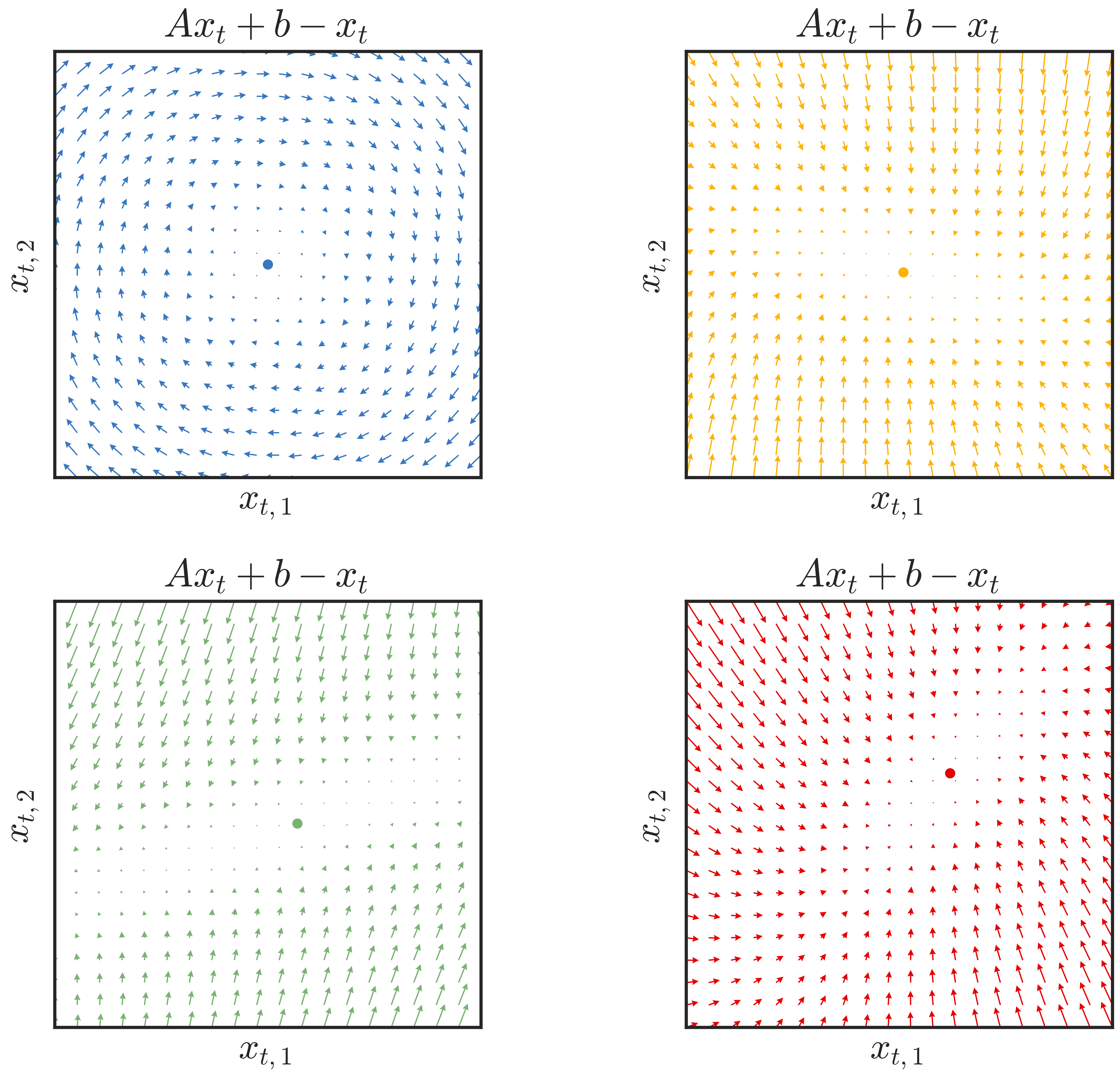


For spiking data data: spike count observations

$$y_t \mid x_t \sim \mathrm{Poisson}(\exp(Cx_t + D))$$

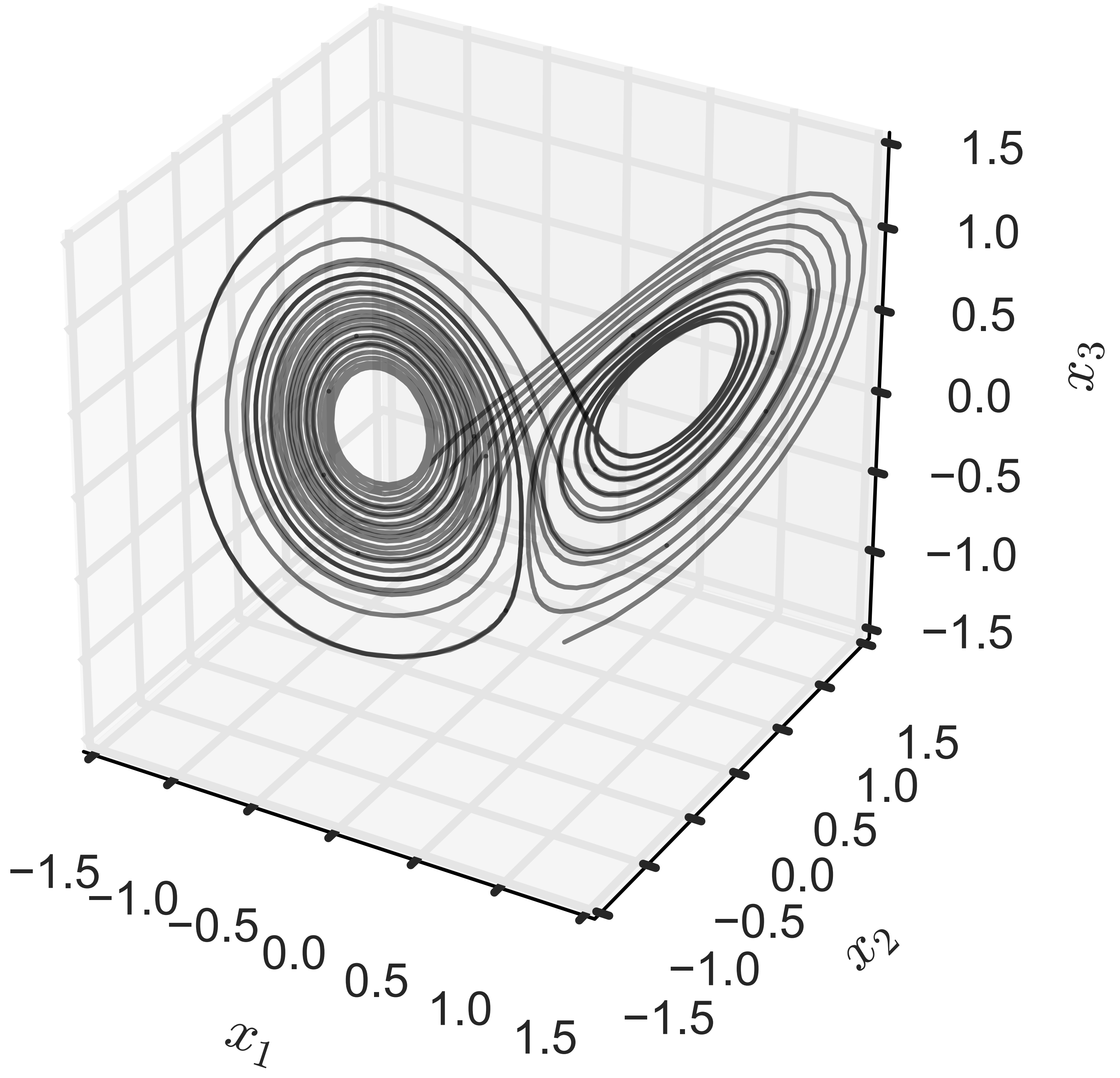


Linear dynamical systems can't do all that much...



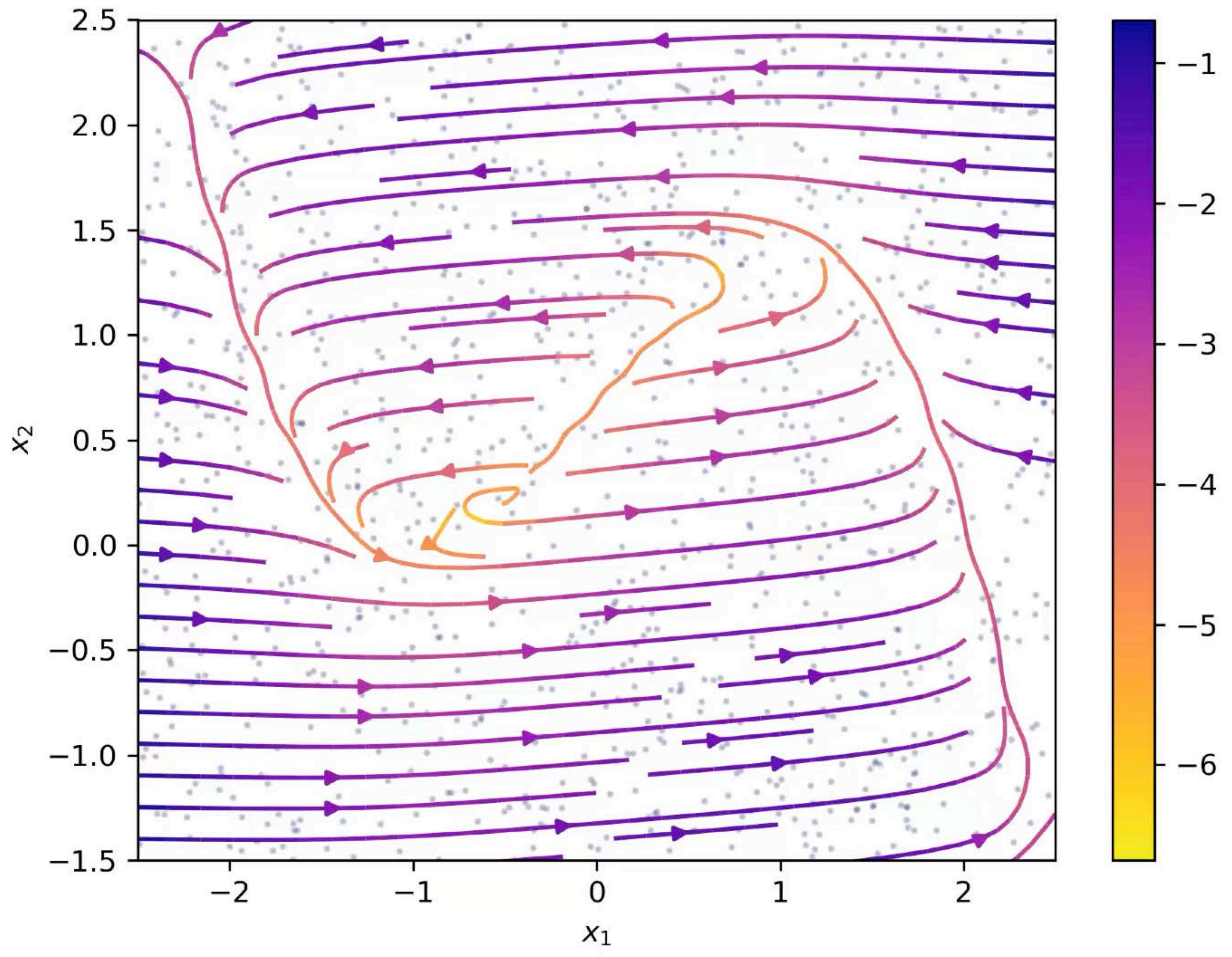
Beyond linear dynamics

Lorenz Attractor



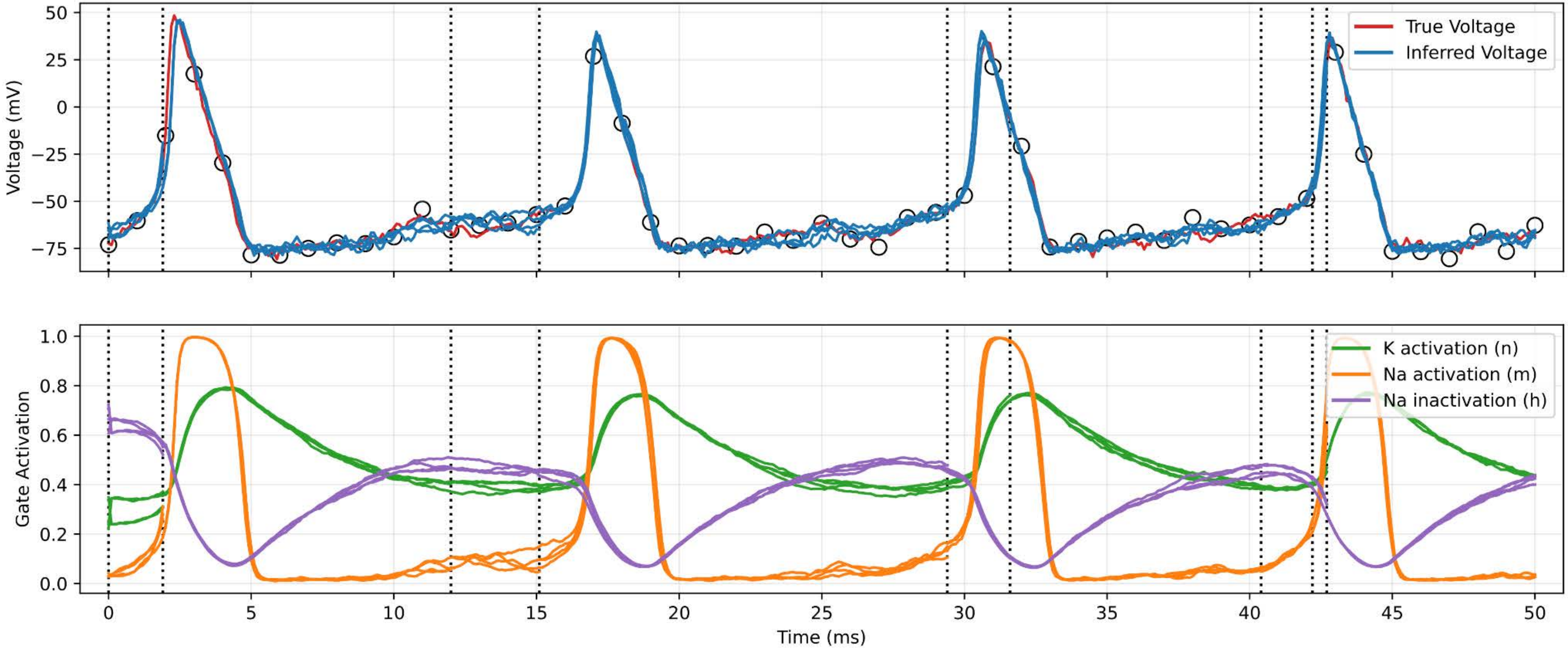
$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \begin{bmatrix} \alpha(x_2 - x_1) \\ x_1(\beta - x_3) - x_2 \\ x_1 x_2 - \gamma x_3 \end{bmatrix}$$

Fitzhugh-Nagumo Model



$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \begin{bmatrix} x_1 - x_1^3 - x_2 \\ \tau^{-1}(x_1 + a - bx_2) \end{bmatrix}$$

Application: Smoothing voltage imaging data



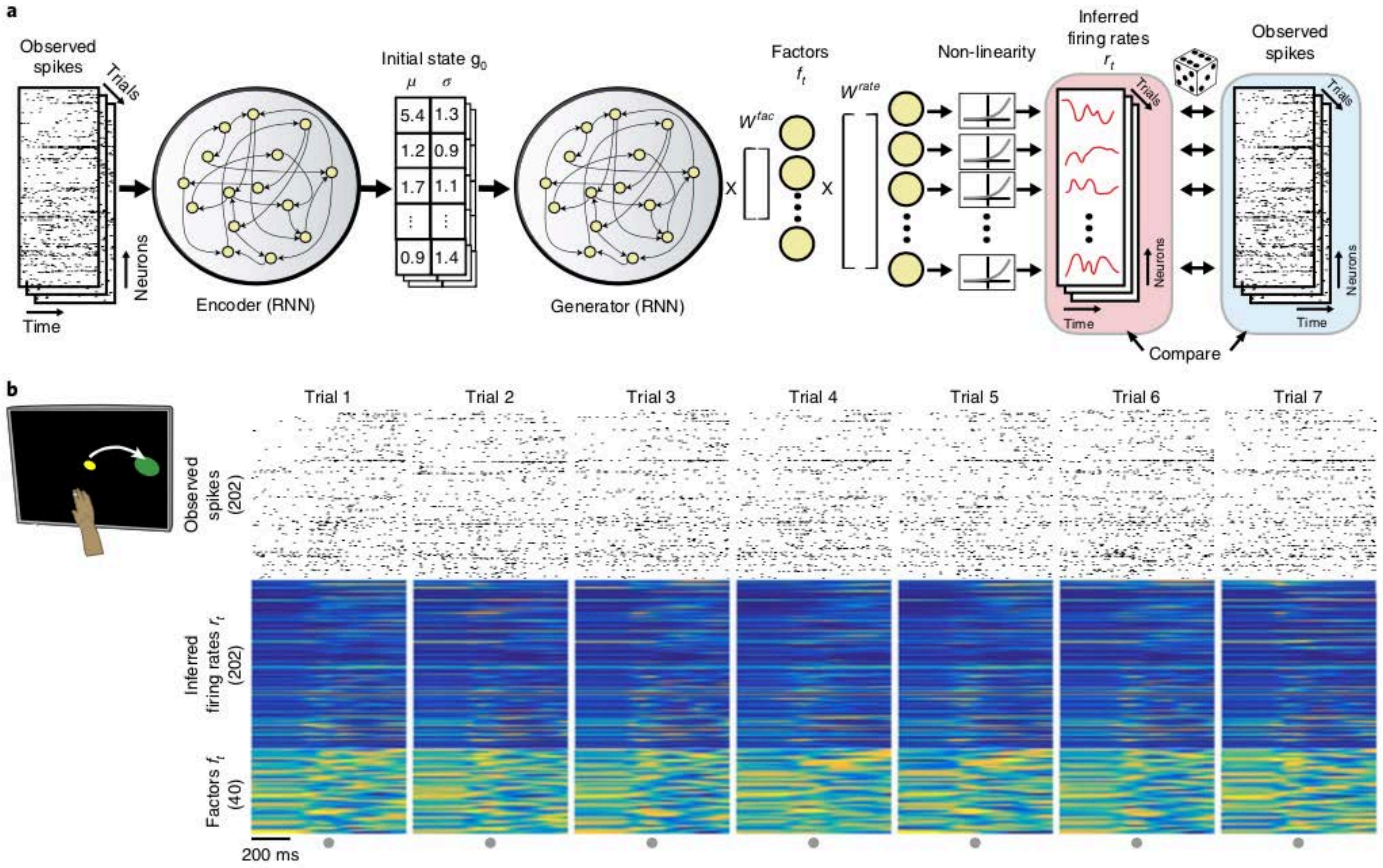
A Taxonomy of state space models

Observation Model (data type, function class, noise model)

Dynamics Model (type, function class, noise model)

	Continuous Linear Gaussian	**Discrete (Gen.) Linear Bernoulli/Poisson/etc.**	
Discrete Markovian Categorical	**HMM** *Rabiner (1989)*	**HMM** *Rabiner (1989)*	
Continuous Linear Gaussian	**LDS** *Kalman (1960)*	**Poisson LDS** *Smith and Brown (2003)* *Paninski et al (2010)* *Macke et al (2011)*	
Continuous Nonlinear Gaussian	**NLDS** *Ahrens, Huys, Paninski (2006)* *Huys and Paninski (2009)*	**NLDS** *Meng, Kramer, Eden (2011)*	

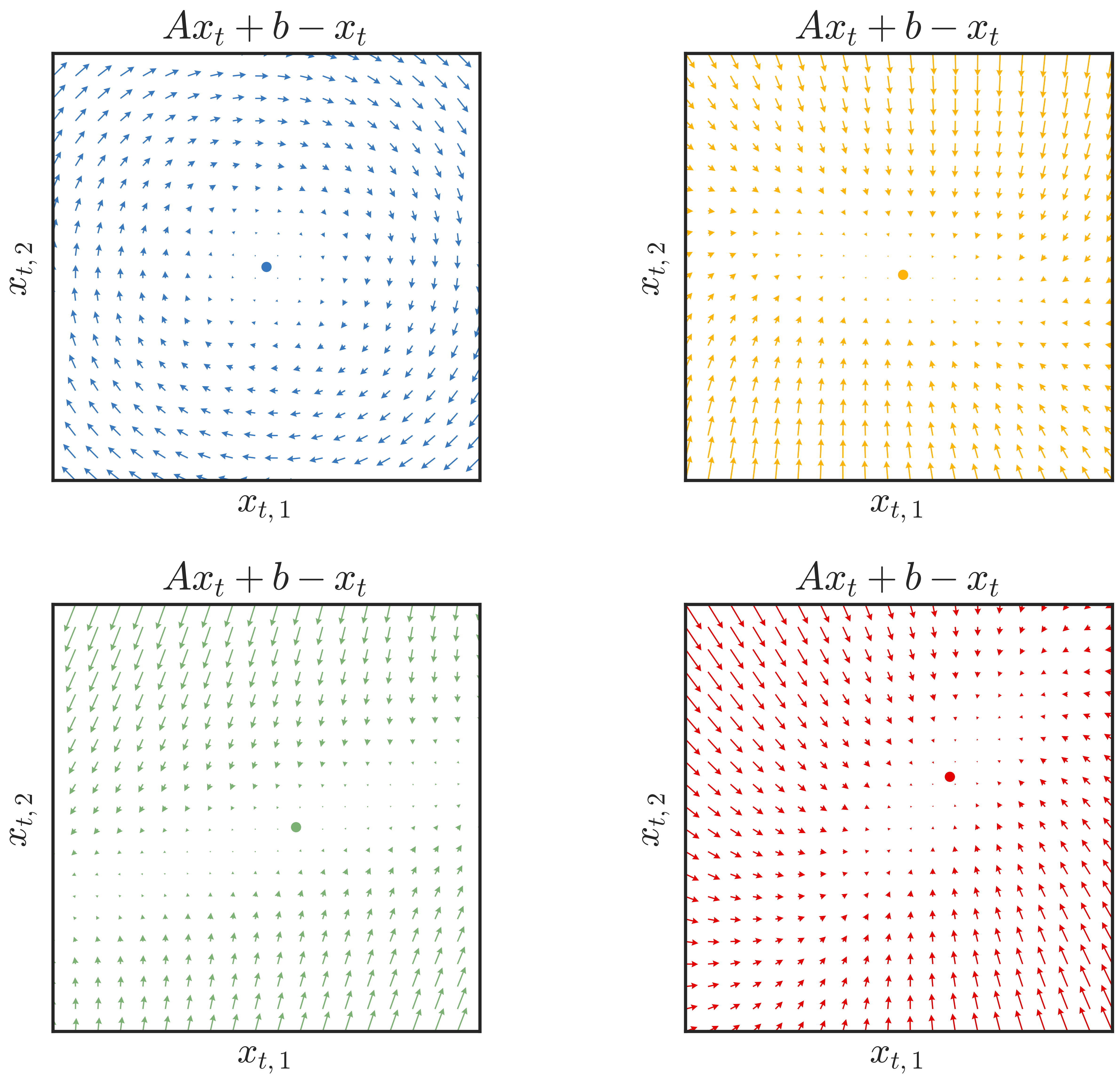
Learning nonlinear dynamical systems



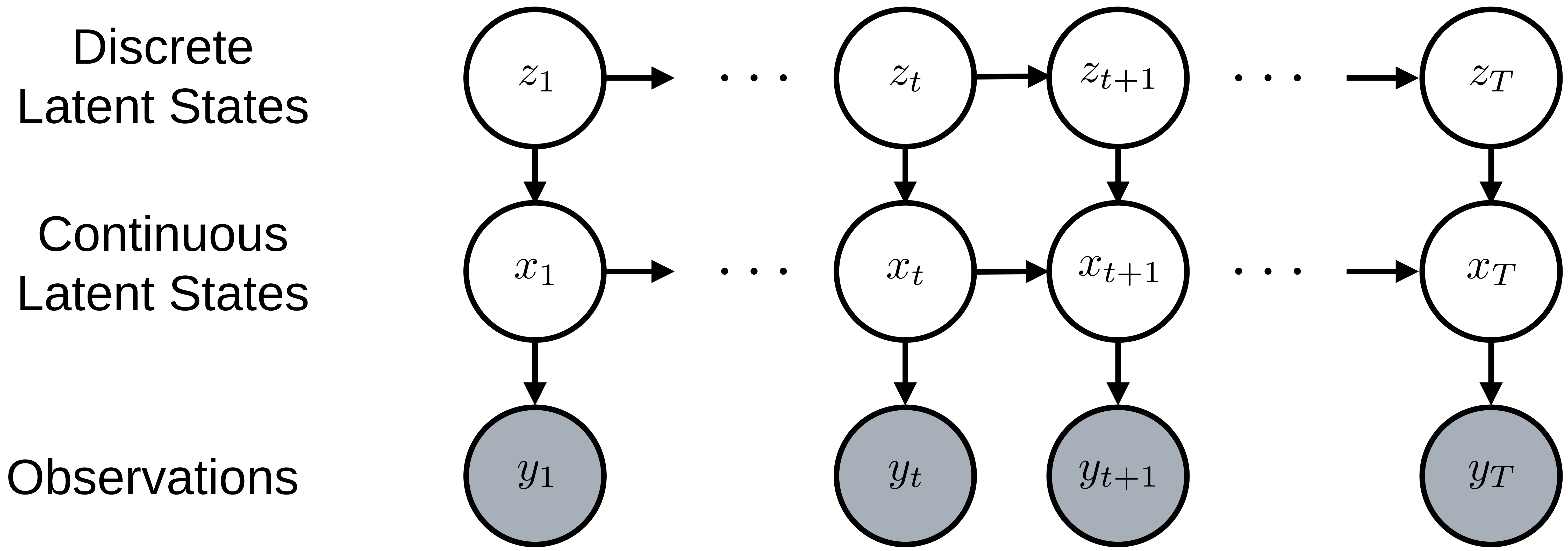
Pandarinath et al (2018)

- Specify a class of nonlinear functions, e.g. those parameterized by weights of a neural network or by a Gaussian process.
- Challenges:
  - How to choose a good function class?
  - How to fit with limited data?
  - How to interpret dynamics?
- More to come in Part II, but first…

Linear dynamical systems can't do all that much…



Switching linear dynamical systems (SLDS)



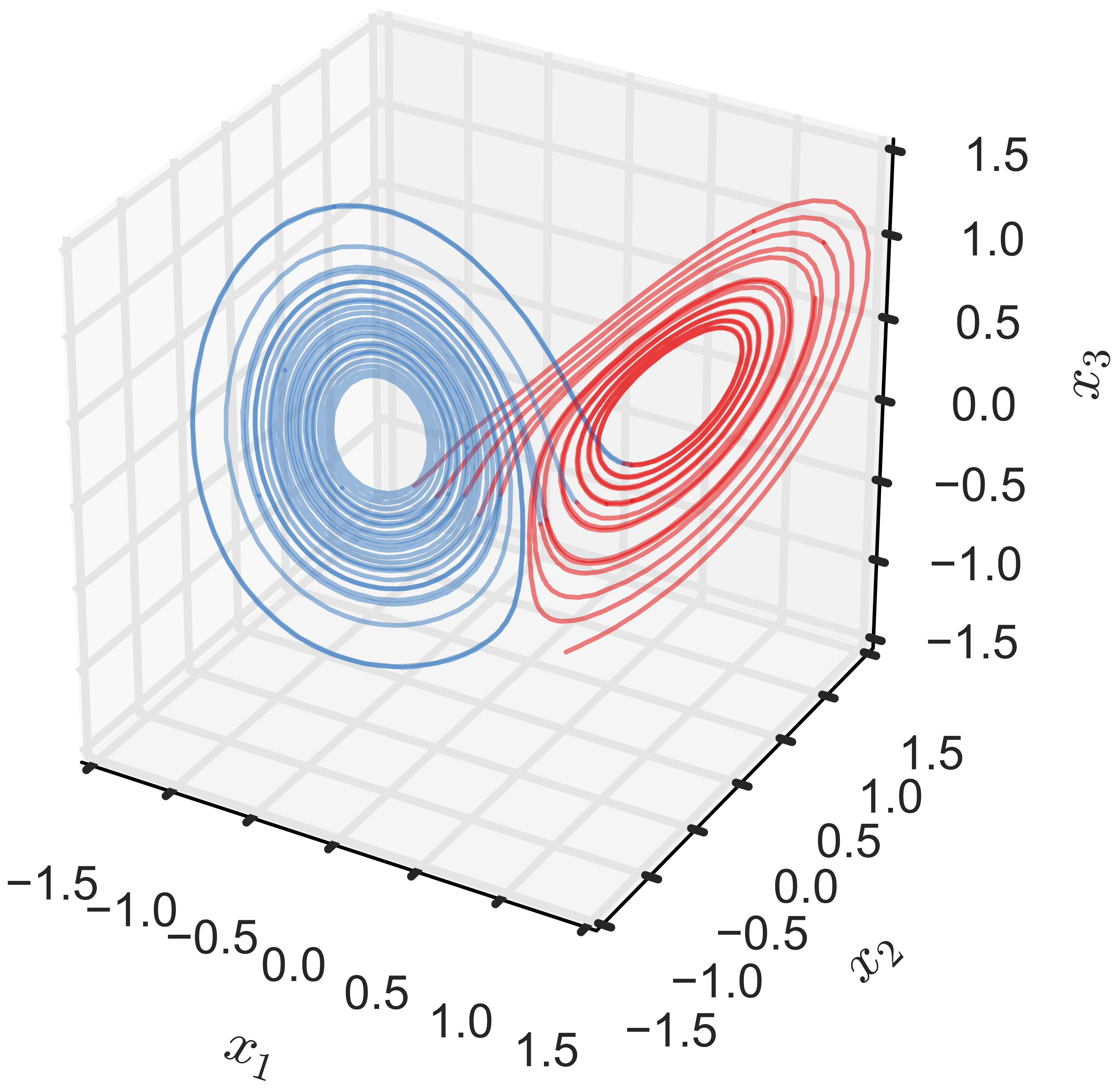
Dynamics:

$$z_{t+1} \mid z_t \sim \mathrm{Cat}(\pi_{z_t})$$
$$x_{t+1} \mid x_t, z_t \sim \mathcal{N}(A_{z_t} x_t + b_{z_t}, Q_{z_t})$$

Observation model:

$$y_t \mid x_t \sim \mathcal{P}(f(C x_t + D))$$

SLDS can approximate nonlinear dynamical systems

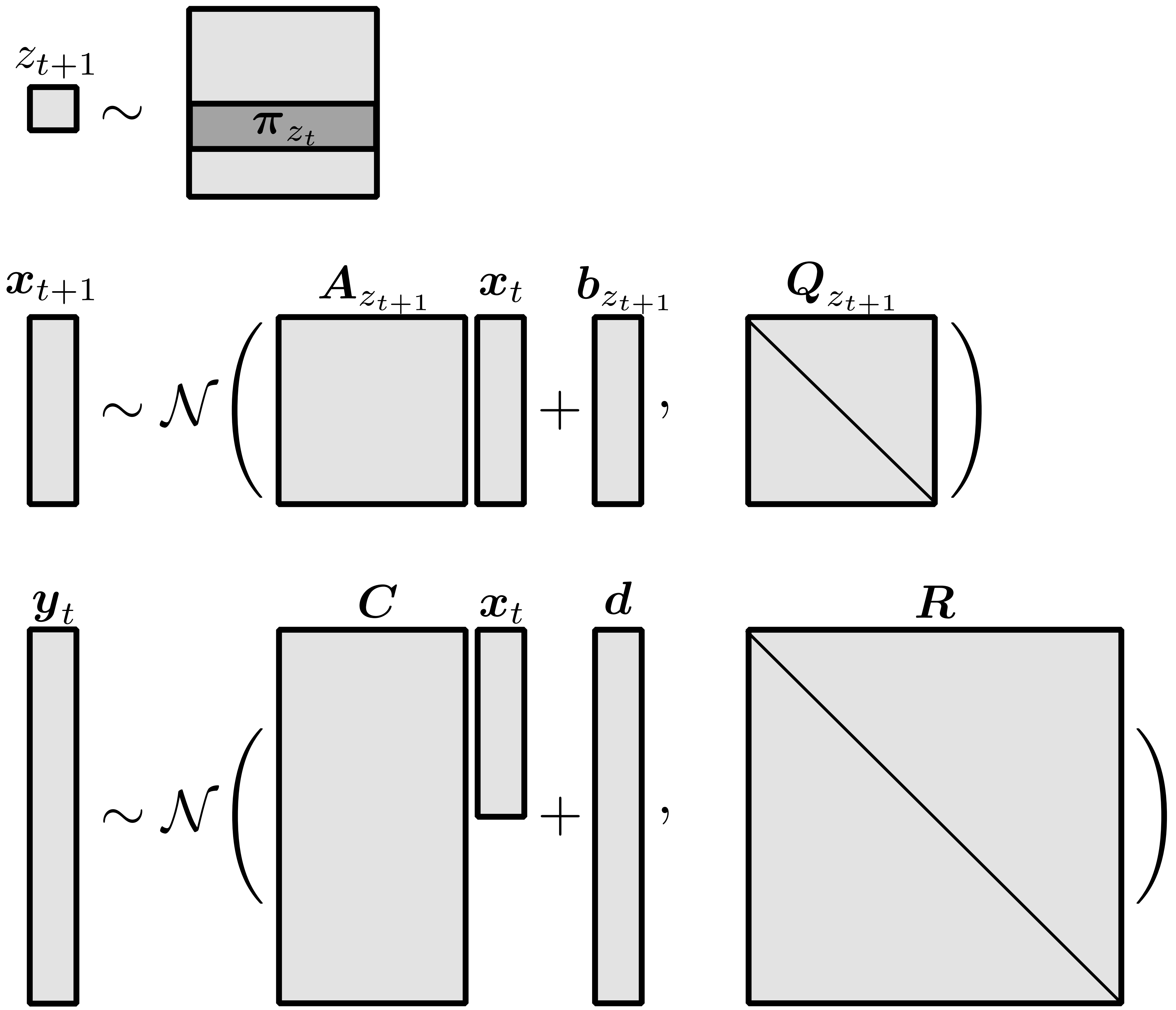


Specifying the form of the dependencies

Discrete Latent State Dynamics

Continuous Latent State Dynamics

Observation Model



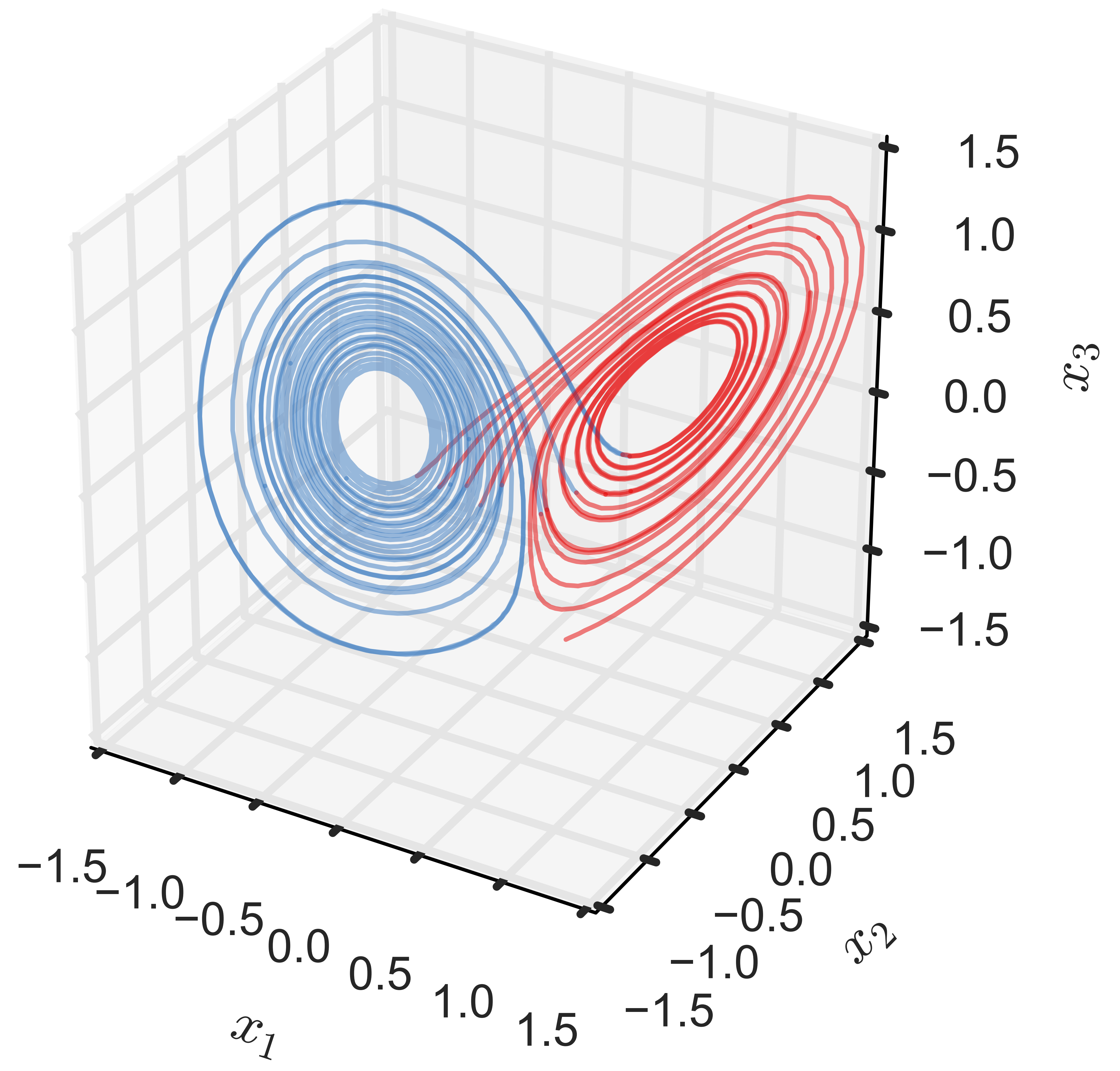
A Taxonomy of state space models

Observation Model (data type, function class, noise model)

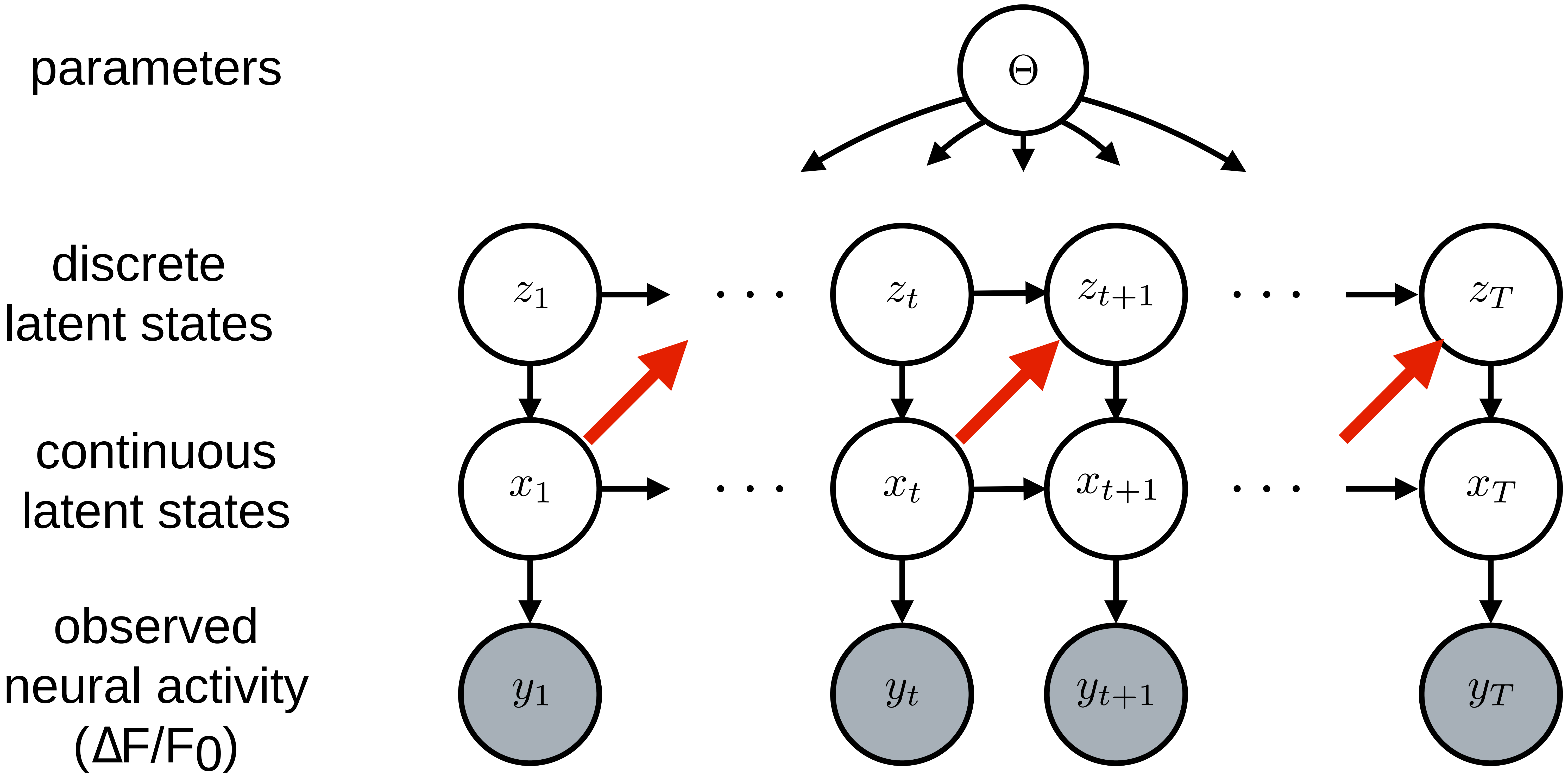
Dynamics Model (type, function class, noise model)

	Continuous Linear Gaussian	Discrete (Gen.) Linear Bernoulli/Poisson/etc.	
Discrete Markovian Categorical	**HMM** *Rabiner (1989)*	**HMM** *Rabiner (1989)*	
Continuous Linear Gaussian	**LDS** *Kalman (1960)*	**Poisson LDS** *Smith and Brown (2003), Paninski et al (2010)* *Macke et al (2011)*	
Continuous Nonlinear (parametric) Gaussian	**NLDS, e.g. Hodgkin-Huxley** *Ahrens, Huys, Paninski (2006)* *Huys and Paninski (2009)*	**NLDS, e.g. Hodgkin-Huxley** *Meng, Kramer, Eden (2011)*	
Mixed Switching Linear	**SLDS** *Ghahramani and Hinton (1996)* *Murphy (1998)*	**Poisson SLDS** *Petreska et al (2013)*	

Problem: SLDS don’t know when to switch!

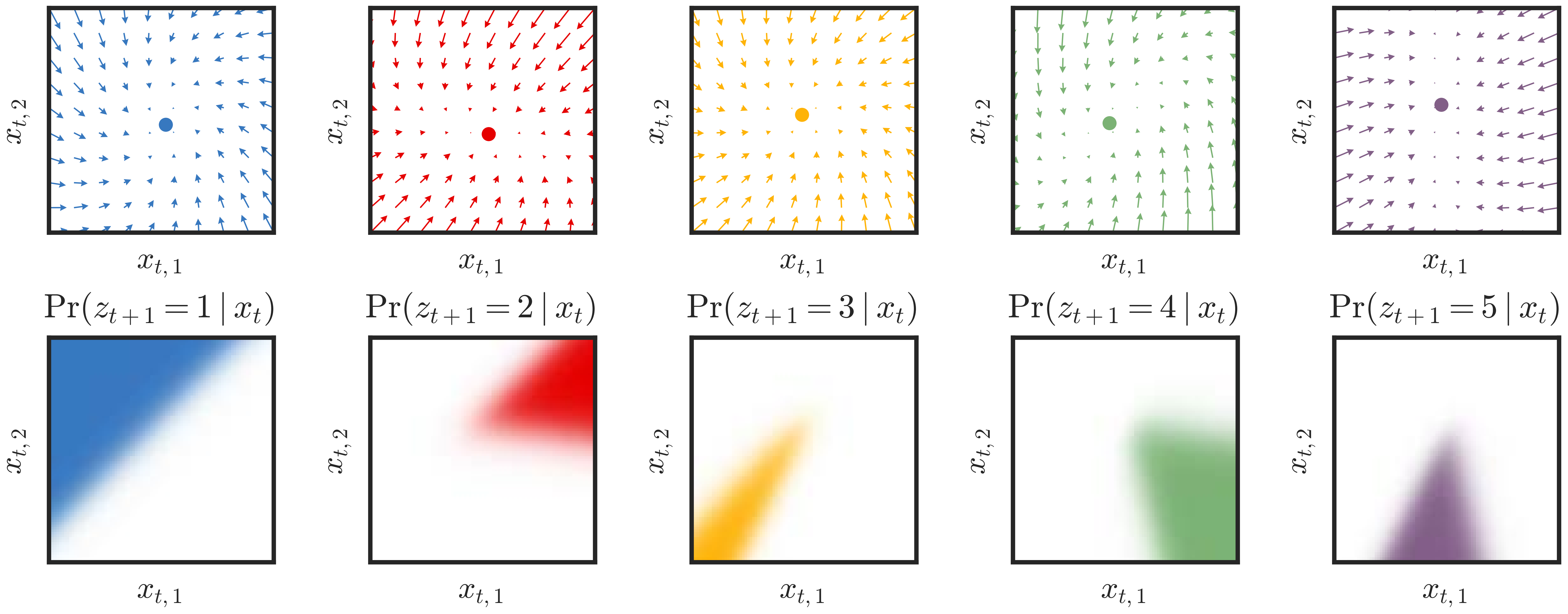


Smarter switching with “Recurrent” SLDS



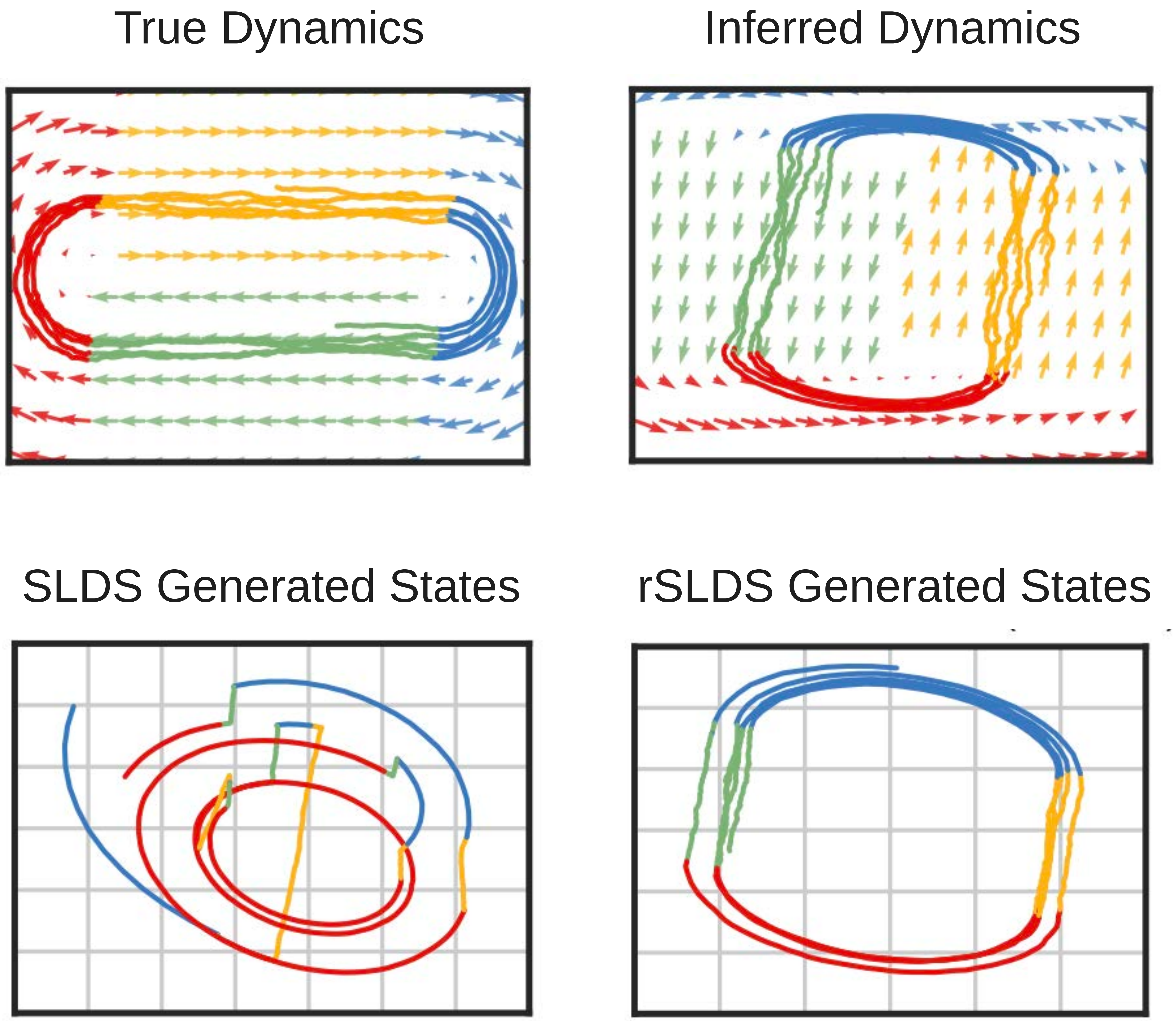
Barber (2006)
Linderman et al. (2017)
Nassar et al. (2019)

Recurrent dependencies carve up continuous space into regions with different dynamics



Barber (2006)
Linderman et al. (2017)
Nassar et al. (2019)

Recurrent switching linear dynamical systems



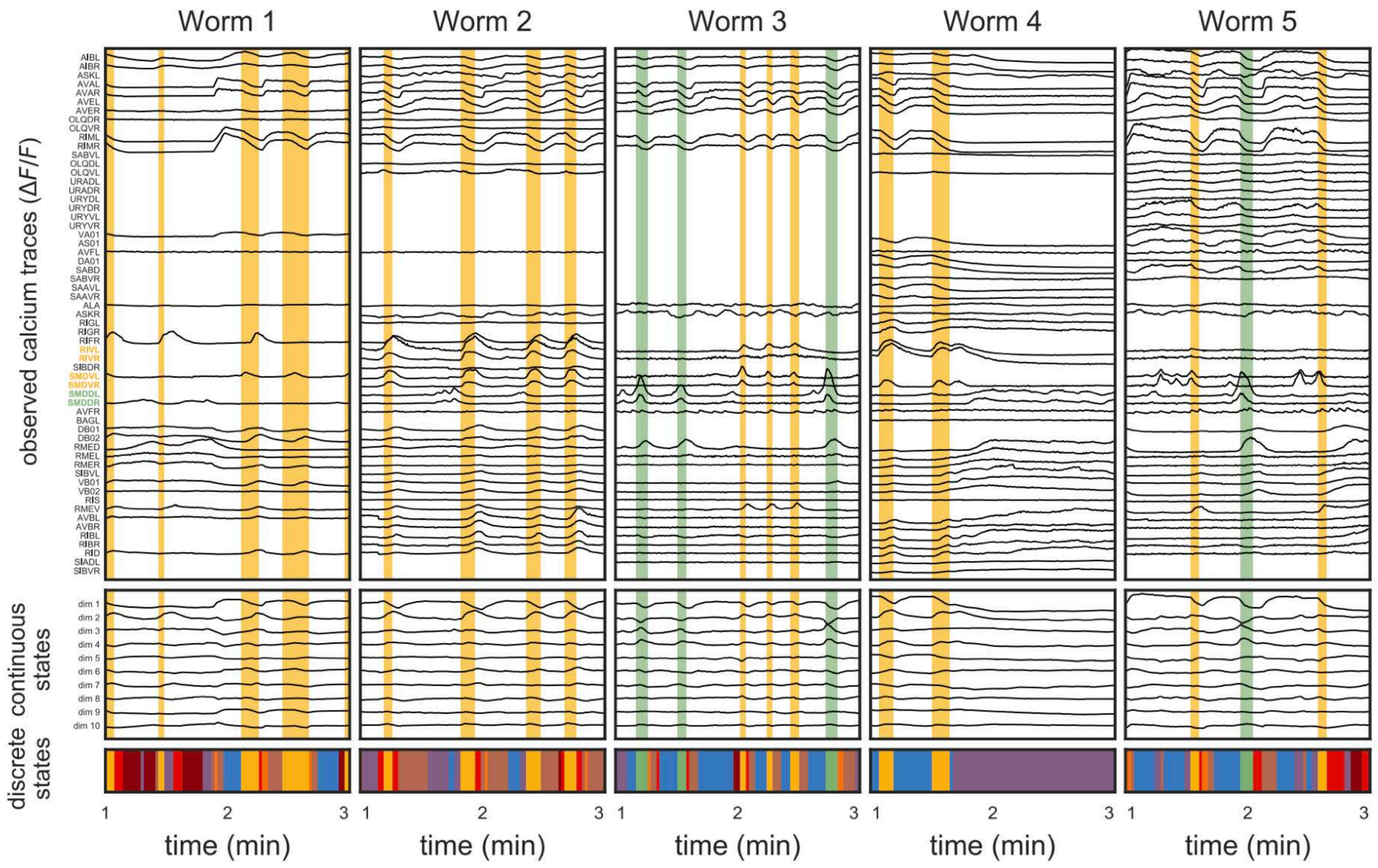
A Taxonomy of state space models

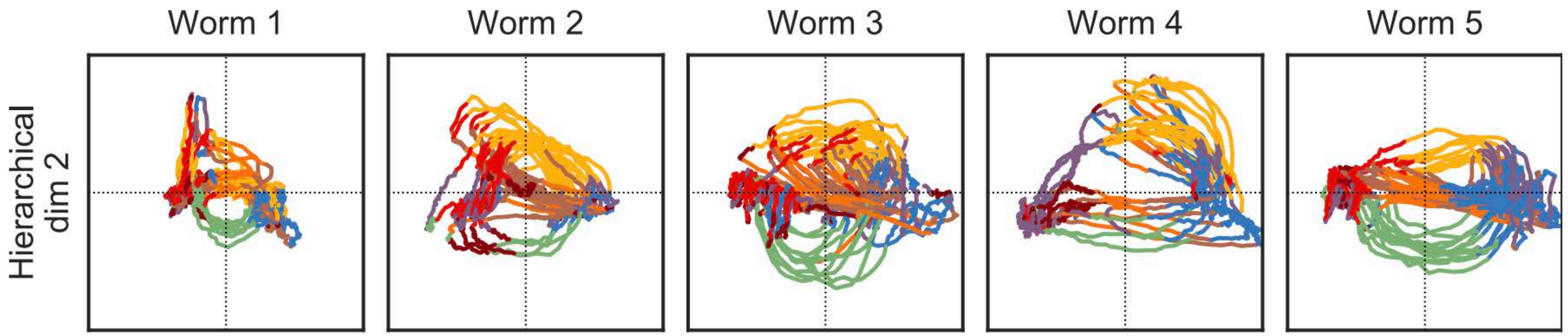
Observation Model (data type, function class, noise model)

Dynamics Model (type, function class, noise model)

	Continuous Linear Gaussian	Discrete (Gen.) Linear Bernoulli/Poisson/etc.	
Discrete Markovian Categorical	**HMM** *Rabiner (1989)*	**HMM** *Rabiner (1989)*	
Continuous Linear Gaussian	**LDS** *Kalman (1960)*	**Poisson LDS** *Smith and Brown (2003), Paninski et al (2010)* *Macke et al (2011)*	
Continuous Nonlinear (parametric) Gaussian	**NLDS, e.g. Hodgkin-Huxley** *Ahrens, Huys, Paninski (2006)* *Huys and Paninski (2009)*	**NLDS, e.g. Hodgkin-Huxley** *Meng, Kramer, Eden (2011)*	
Mixed Switching Linear	**SLDS** *Ghahramani and Hinton (1996)* *Murphy (1998)*	**Poisson SLDS** *Petreska et al (2013)*	
Mixed Recurrent Linear	**recurrent/augmented SLDS** *Barber (2006); Pachitariu et al (2014); Linderman et al (2017); Nassar et al (2019)*	**rSLDS** *Linderman et al (2017)* *Nassar et al (2019)*	

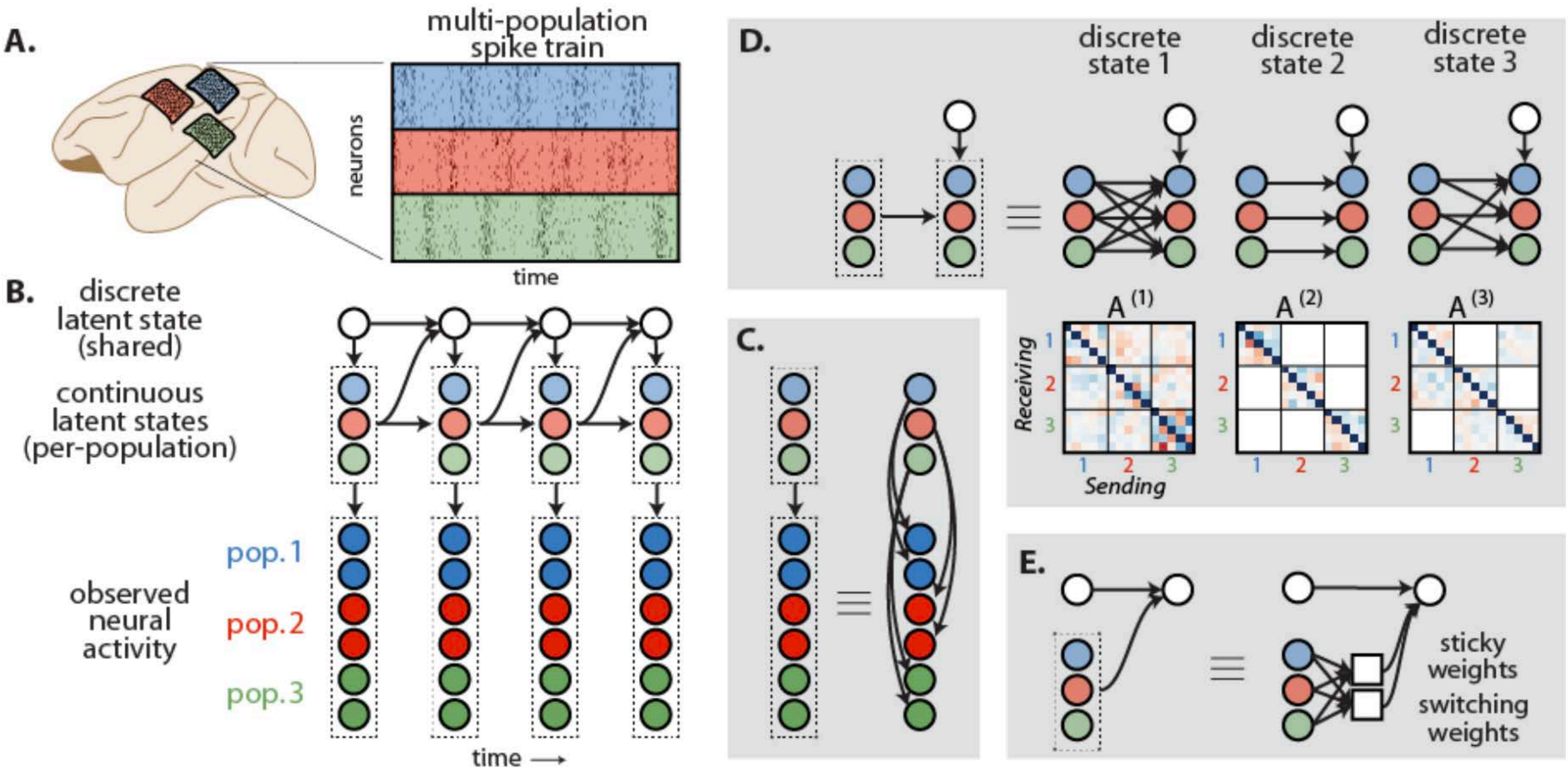
Hierarchical model uncovering states of worm dynamics





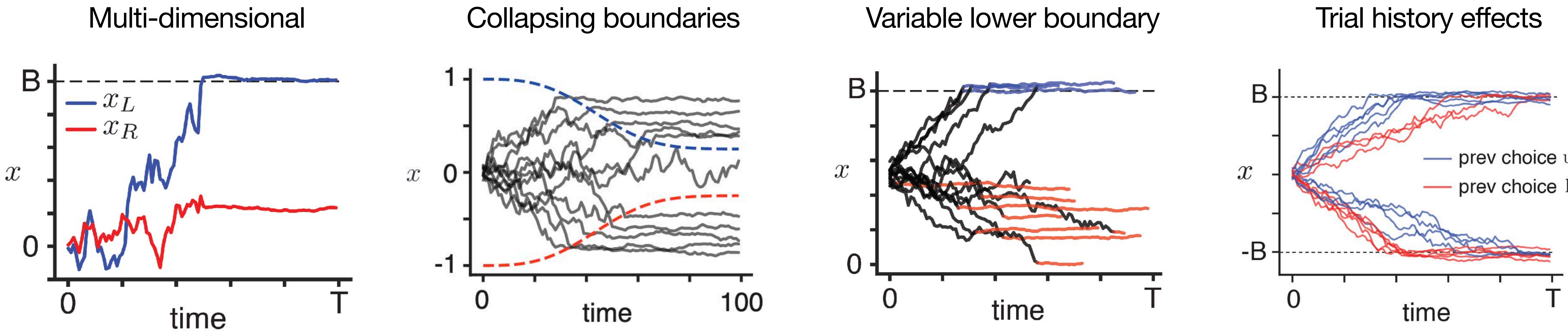
Linderman et al (2019)

Interactions between brain-regions with multi-region rSLDS



Glaser et al (2020)

Unifying and generalizing neural dynamics during decision-making

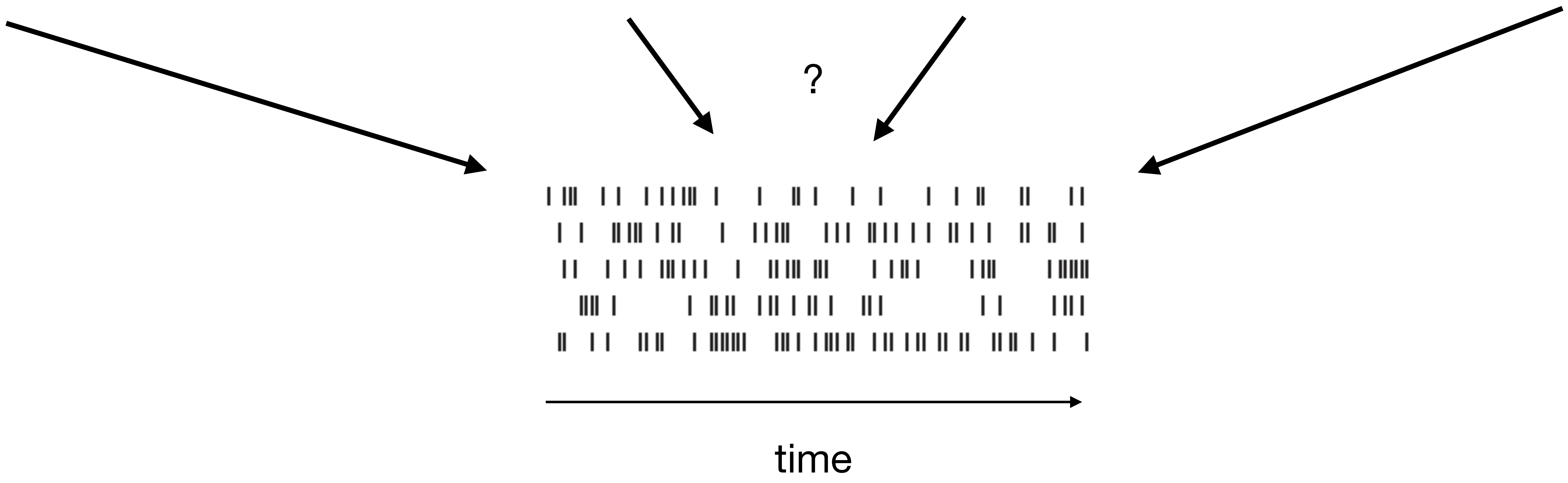


Gold and Shadlen, 2007
Churchland et al., 2008

Drugowitsch et al., 2012
Hawkins et al., 2015

Roitman and Shadlen; 2002
Gold and Shadlen; 2007

Urai et al., 2019



Zoltowski et al (2020)

flow field of VMHvl (mouse 1 - intruder 1) related to Figure 3

An approximate line attractor in the hypothalamus that encodes an aggressive internal state

Aditya Nair, Tomomi Karigo, Bin Yang, Scott Linderman
David J Anderson* & Ann Kennedy*

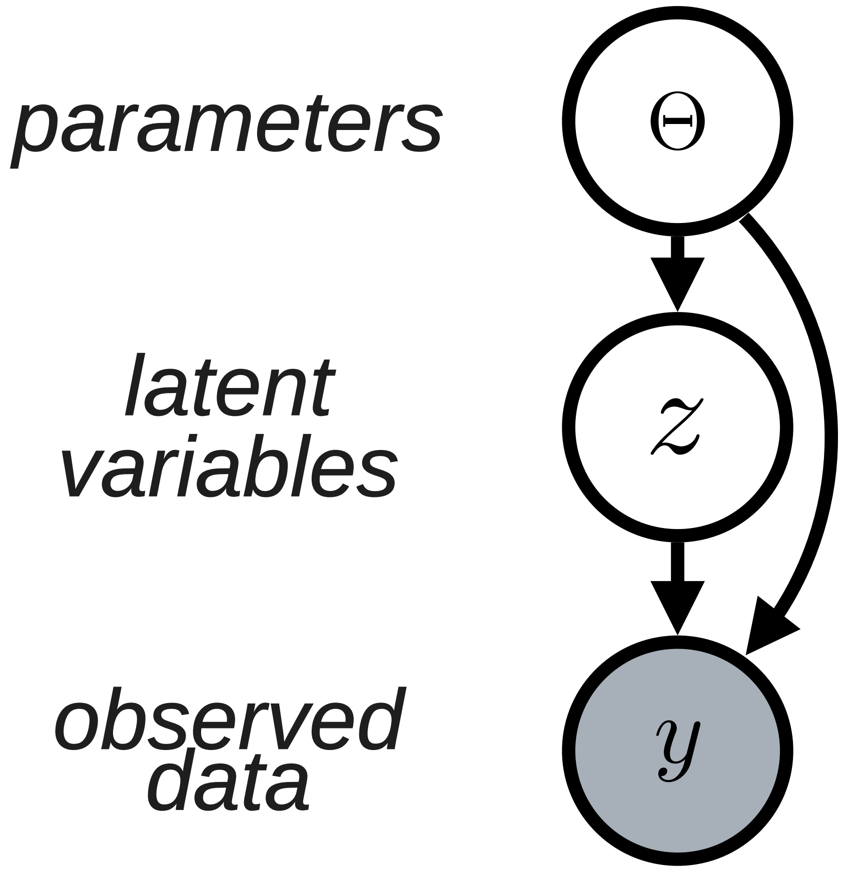
Outline

<u>**Part I: Foundations**</u>

- Motivating Examples
- State Space Models (SSMs)
  - Hidden Markov Models
  - Linear Dynamical Systems
  - Nonlinear & Switching Linear Dynamical Systems
- **Learning and Inference Algorithms**
  - Expectation-Maximization
  - Message Passing
  - Approximate Inference (E/UKF, SMC, VI)
- Code Pointers

Bayesian Learning and Inference Challenges

Simpler model,
same problem:



Learning Goal: find parameters that maximize *marginal likelihood*:

$$\Theta^\star = \arg\max p(y; \Theta)$$
$$= \arg\max \int p(y, z; \Theta)\,\mathrm{d}z$$

Inference Goal: approximate the *posterior distribution* of latent variables:

$$p(z \mid y; \Theta) = \frac{p(y, z; \Theta)}{p(y; \Theta)}$$
$$= \frac{p(y, z; \Theta)}{\int p(y, z; \Theta)\,\mathrm{d}z}$$

Evaluate **posterior expectations** of interest:

- Expected latent states (smoothing):

$$\mathbb{E}_{p(z|y;\Theta)}\left[z_t\right]$$

- Probability of being in a discrete state (smoothing):

$$\mathbb{E}_{p(z|y;\Theta)}\left[\mathbb{I}[z_t = k]\right]$$

- Second moments (covariances):

$$\mathbb{E}_{p(z|y;\Theta)}\left[z_t z_{t+1}^\mathsf{T}\right]$$

- Expected observations (reconstruction):

$$\mathbb{E}_{p(z|y;\Theta)}\left[g(z_t)\right]$$

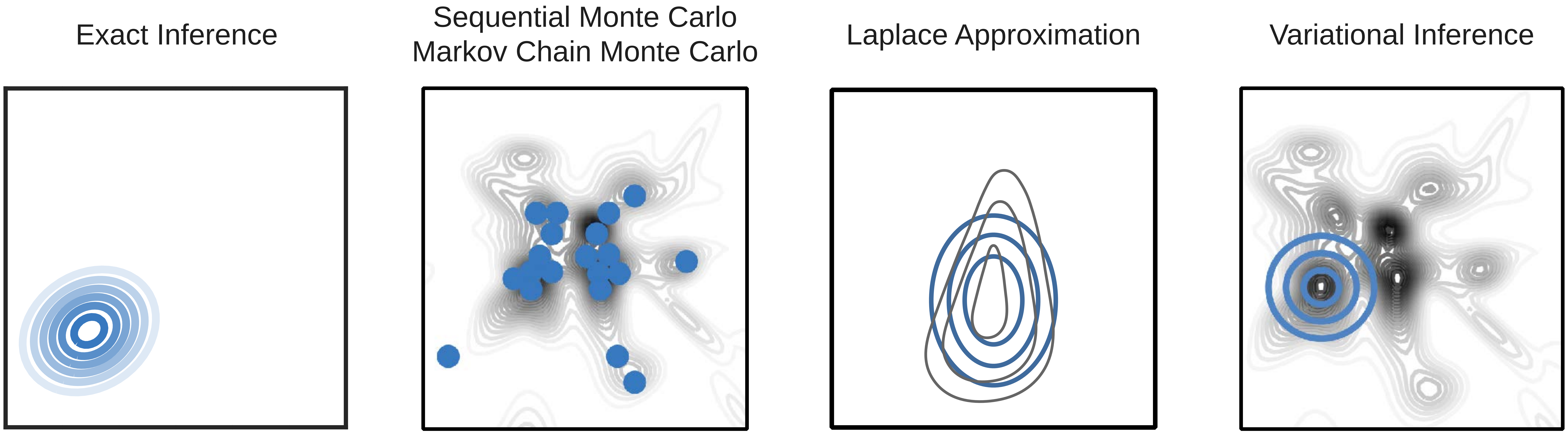
- Future observations (prediction):

$$\mathbb{E}_{p(z|y;\Theta)}\left[g(f(z_T))\right]$$

- Expected log joint probability:

$$\mathbb{E}_{p(z|y;\Theta)}\left[\log p(z, y; \Theta')\right]$$

Methods of approximate Bayesian inference



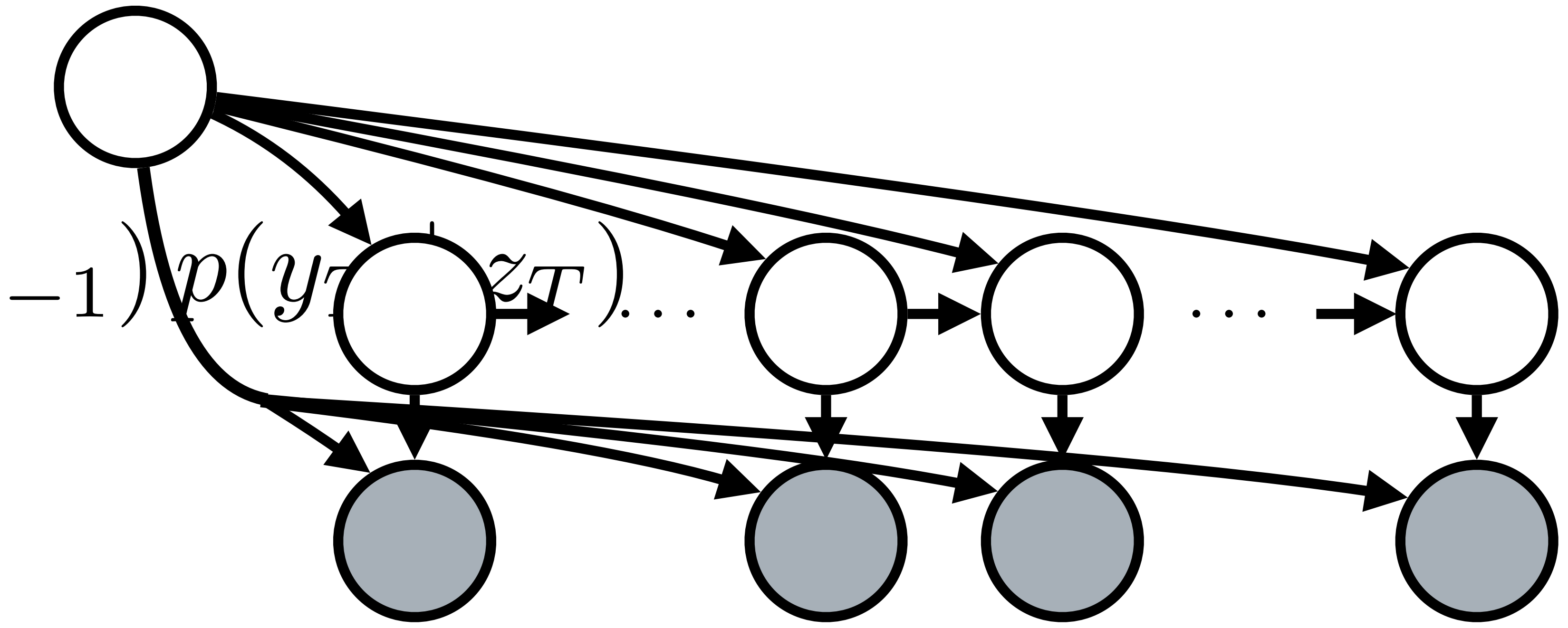
Exact Inference: The algebraic way

$$p(y) = \sum_{z_T} \cdots \sum_{z_2} \sum_{z_1} p(z_1, \ldots, z_T, y_1, \ldots, y_T)$$

$$= \sum_{z_T} \cdots \sum_{z_2} \sum_{z_1} p(z_1)\, p(y_1 \mid z_1)\, p(z_2 \mid z_1)\, p(y_2 \mid z_2)\, p(z_3 \mid z_2) \ldots p(z_T \mid z_{T-1})\, p(y_T \mid z_T)$$

$$= \sum_{z_T} \cdots \sum_{z_2} \underbrace{\sum_{z_1} p(z_1)\, p(y_1 \mid z_1)\, p(z_2 \mid z_1)}_{\alpha(z_2; y_1)}\, p(y_2 \mid z_2)\, p(z_3 \mid z_2) \ldots p(z_T \mid z_{T-1})\, p(y_T \mid z_T)$$

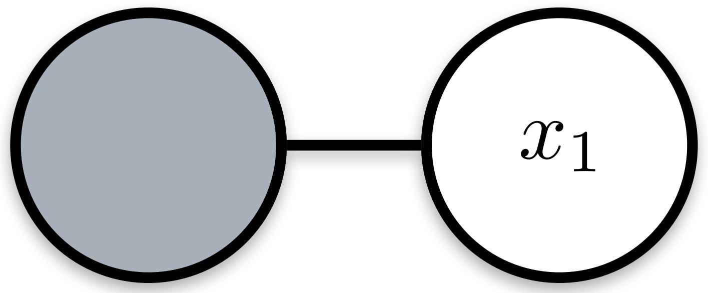
$$= \sum_{z_T} \cdots \underbrace{\sum_{z_2} \alpha(z_2; y_1)\, p(y_2 \mid z_2)\, p(z_3 \mid z_2)}_{\alpha(z_3; y_1, y_2)} \ldots p(z_T \mid z_{T-1})\, p(y_T \mid z_T)$$



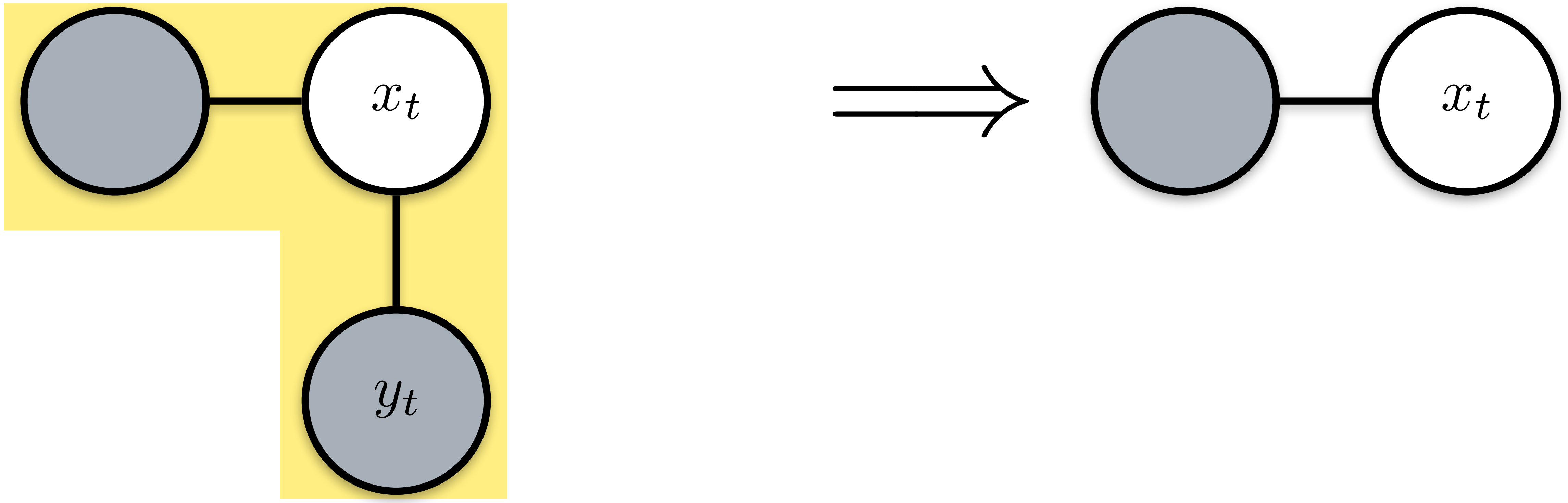
$$= \sum_{z_T} \alpha(z_T; y_1, \ldots, y_{T-1})\, p(z_T \mid z_{T-1})\, p(y_T \mid z_T)$$

*Once we have the marginal likelihood, we can derive similar algorithms to compute expectations of interest.

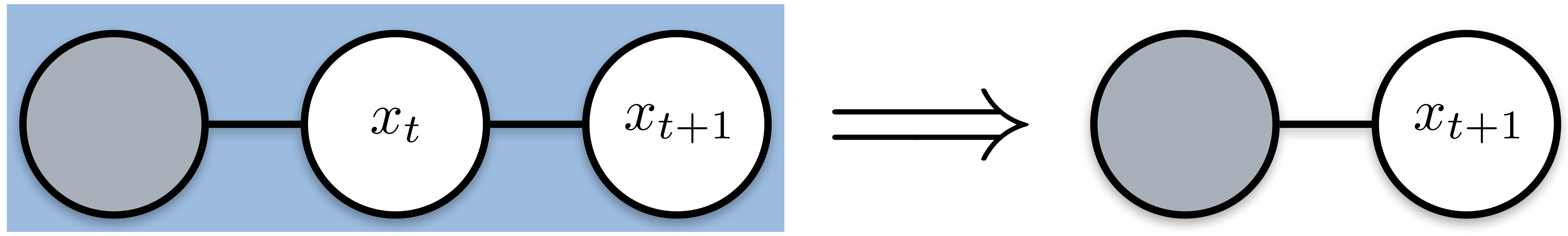
Exact Inference: The graphical way



Incoming message

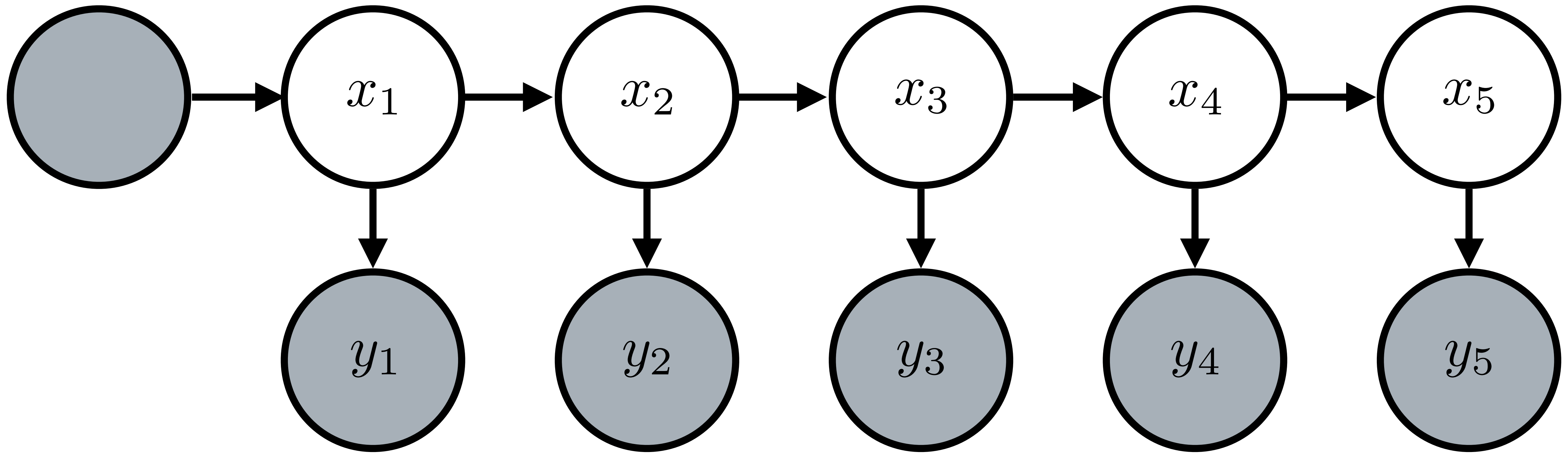


Condition on observations



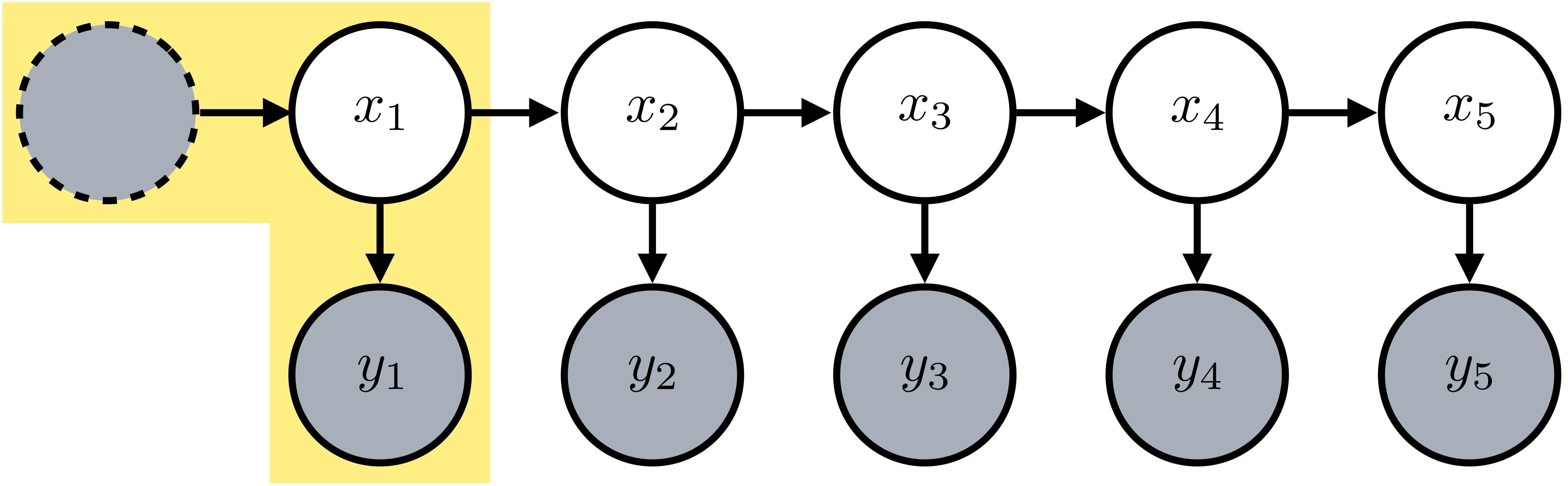
Marginalize out previous state

Exact Inference: The graphical way

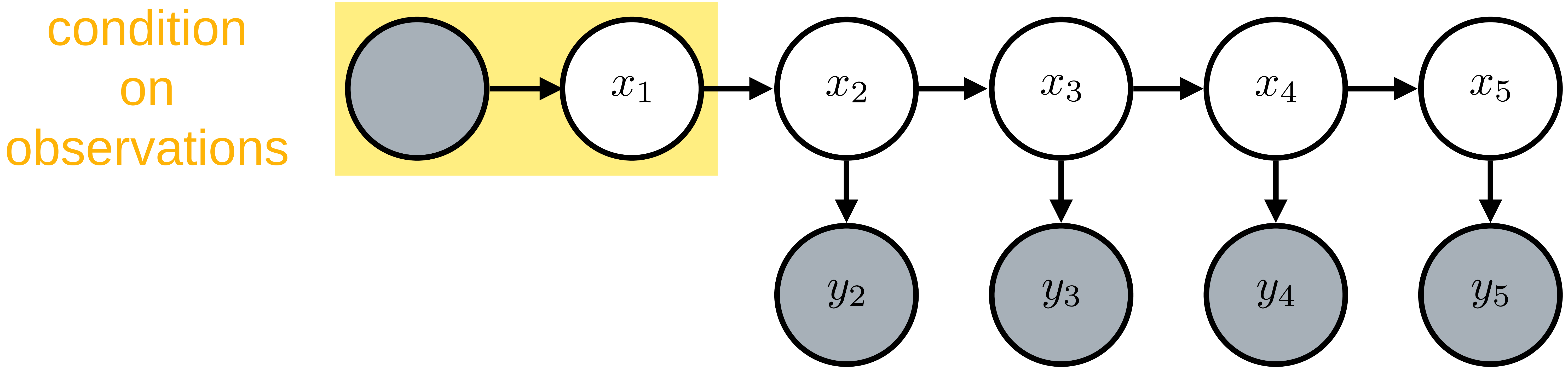


Exact Inference: The graphical way

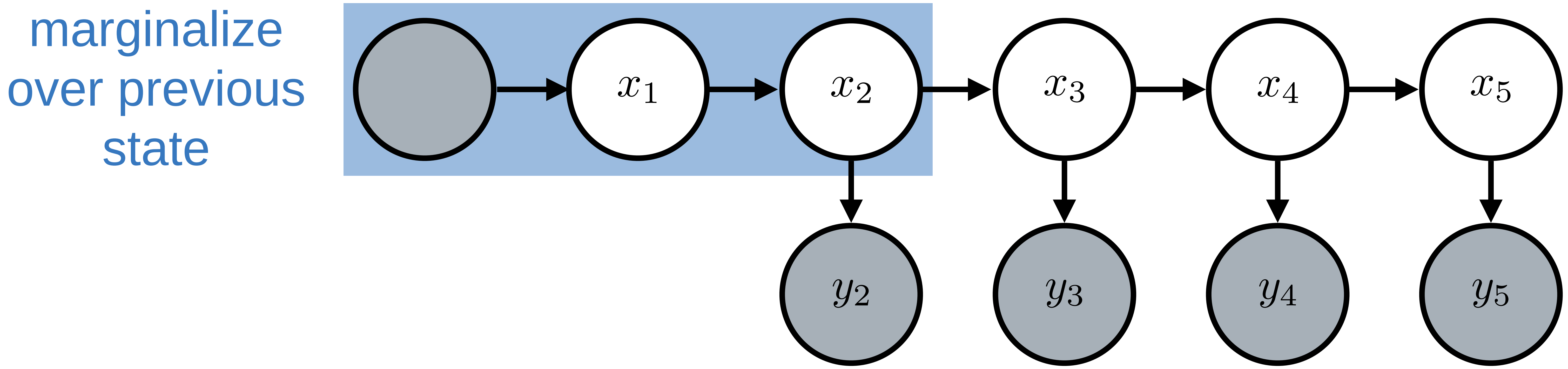
condition
on
observations



Exact Inference: The graphical way

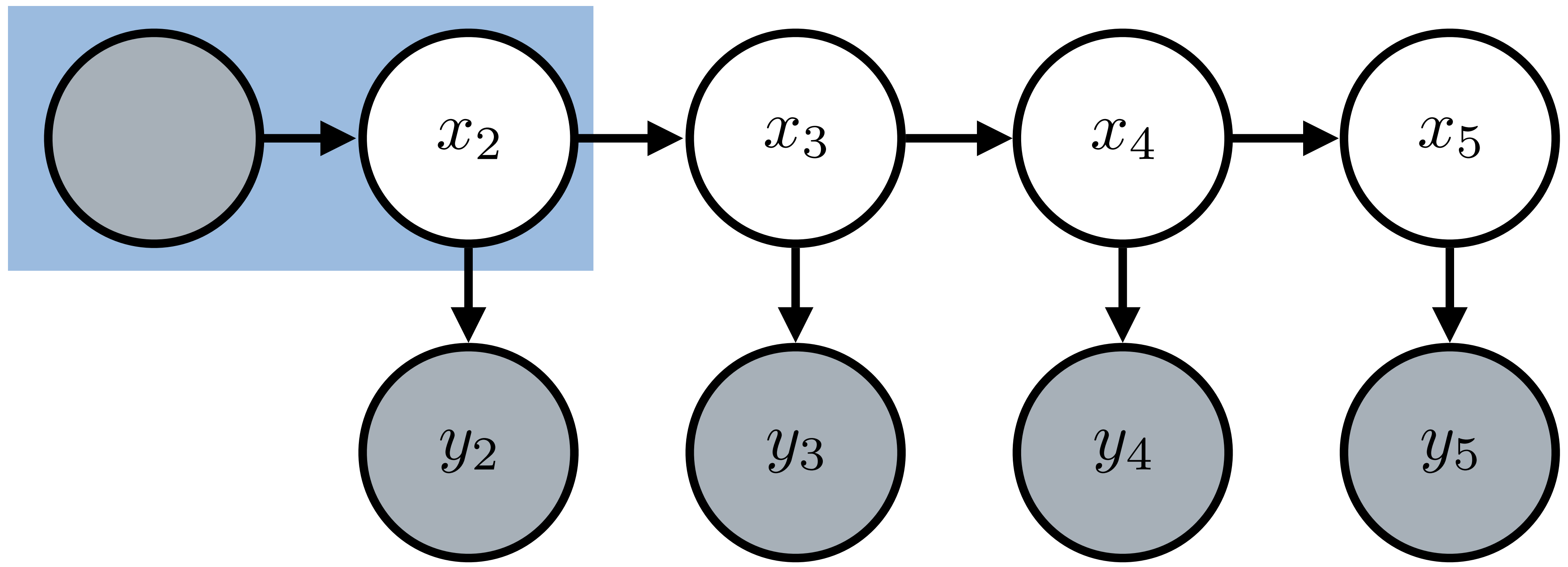


Exact Inference: The graphical way



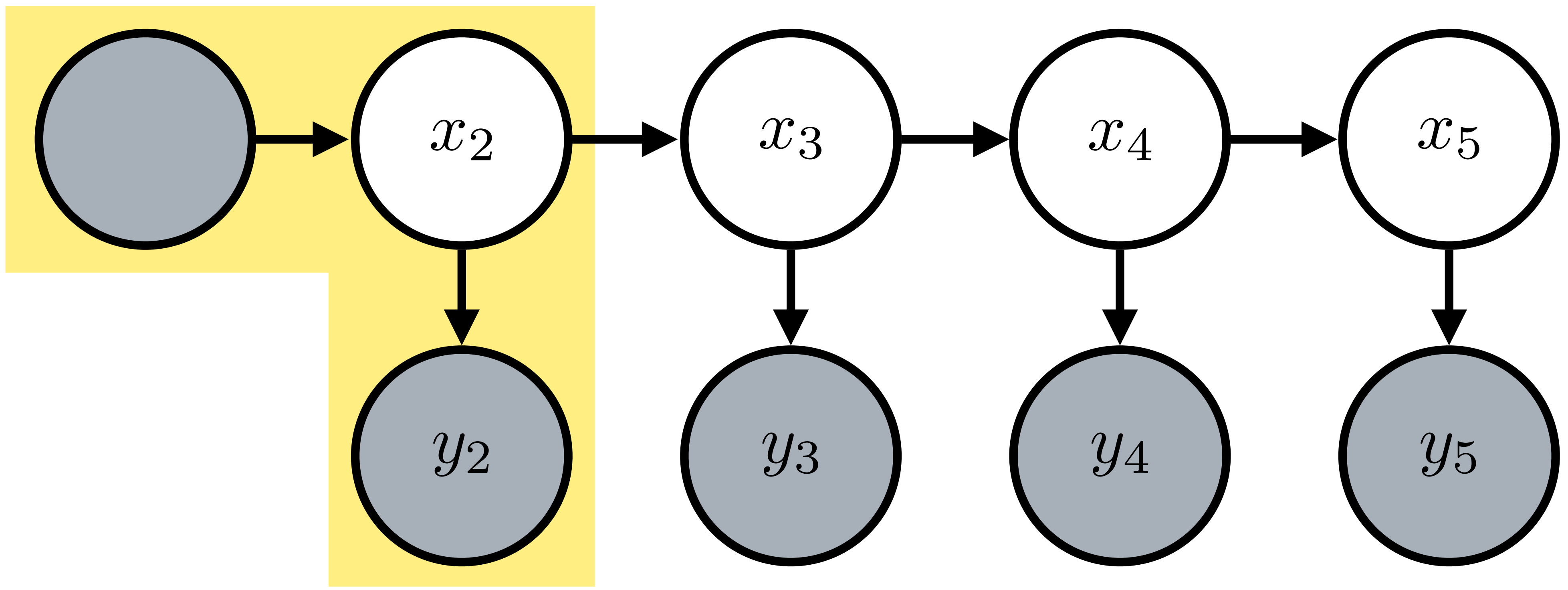
Exact Inference: The graphical way

marginalize
over previous
state



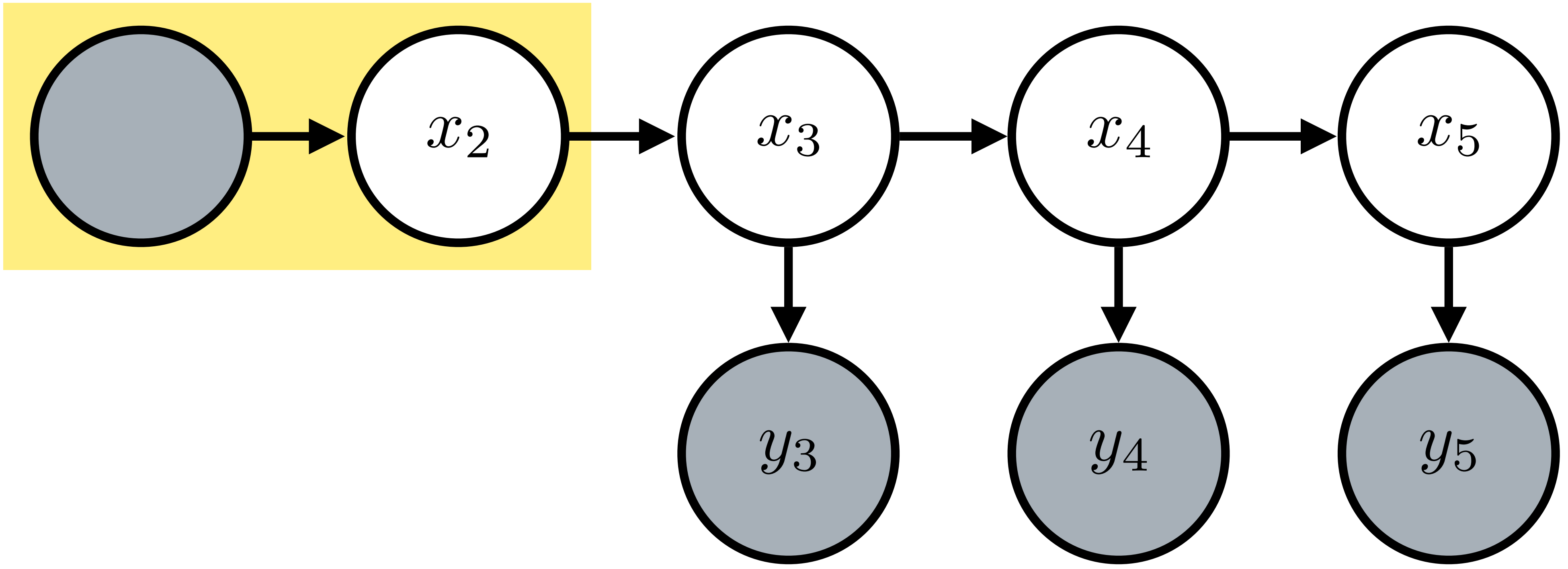
Exact Inference: The graphical way

condition
on
observations



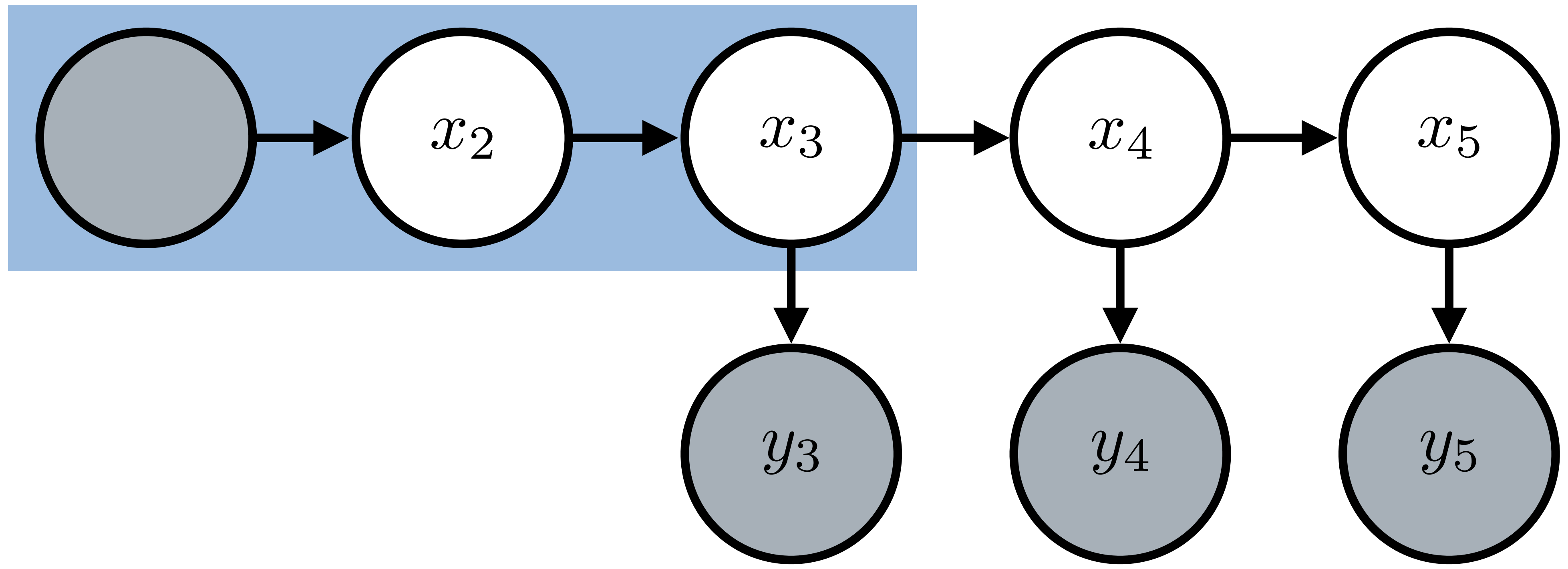
Exact Inference: The graphical way

condition
on
observations



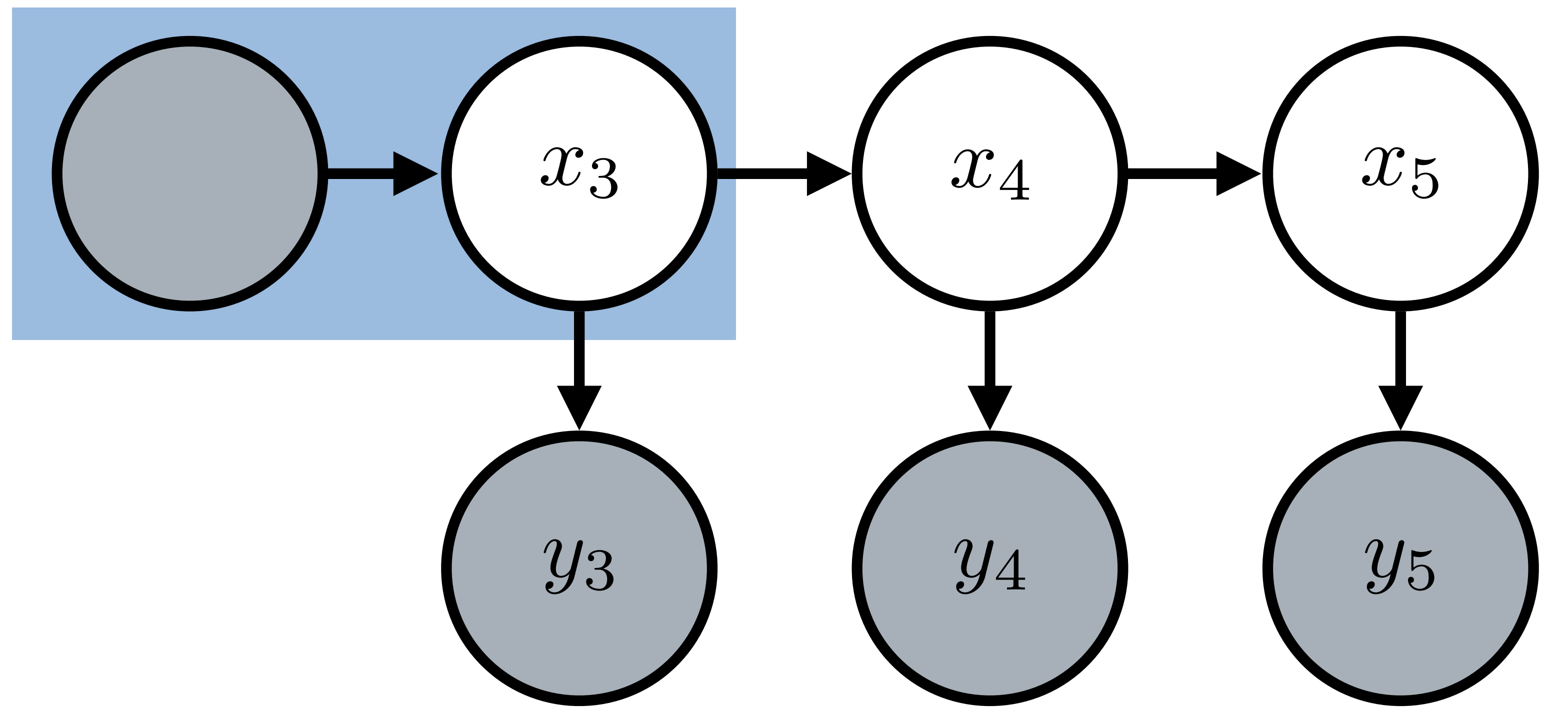
Exact Inference: The graphical way

marginalize
over previous
state



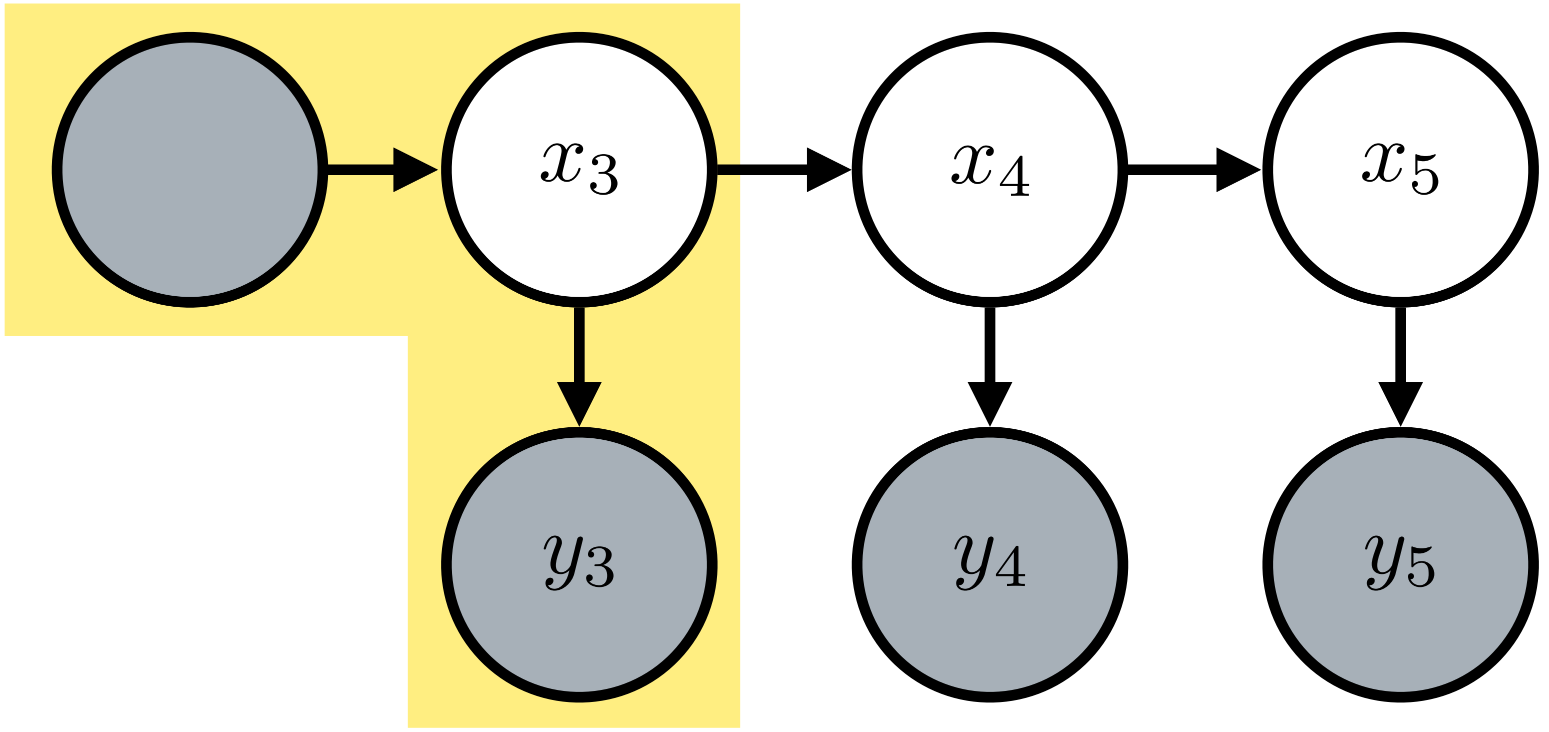
Exact Inference: The graphical way

marginalize
over previous
state



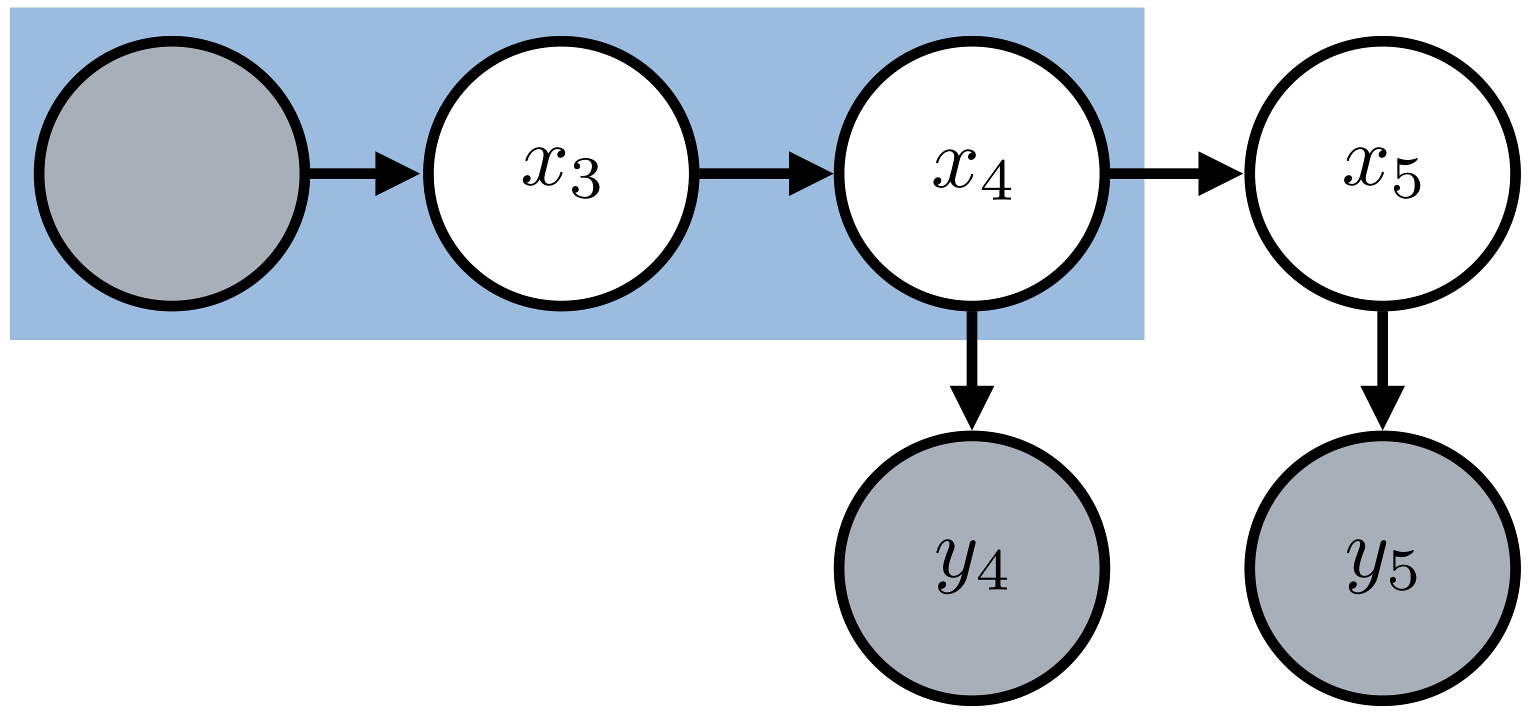
Exact Inference: The graphical way

condition
on
observations



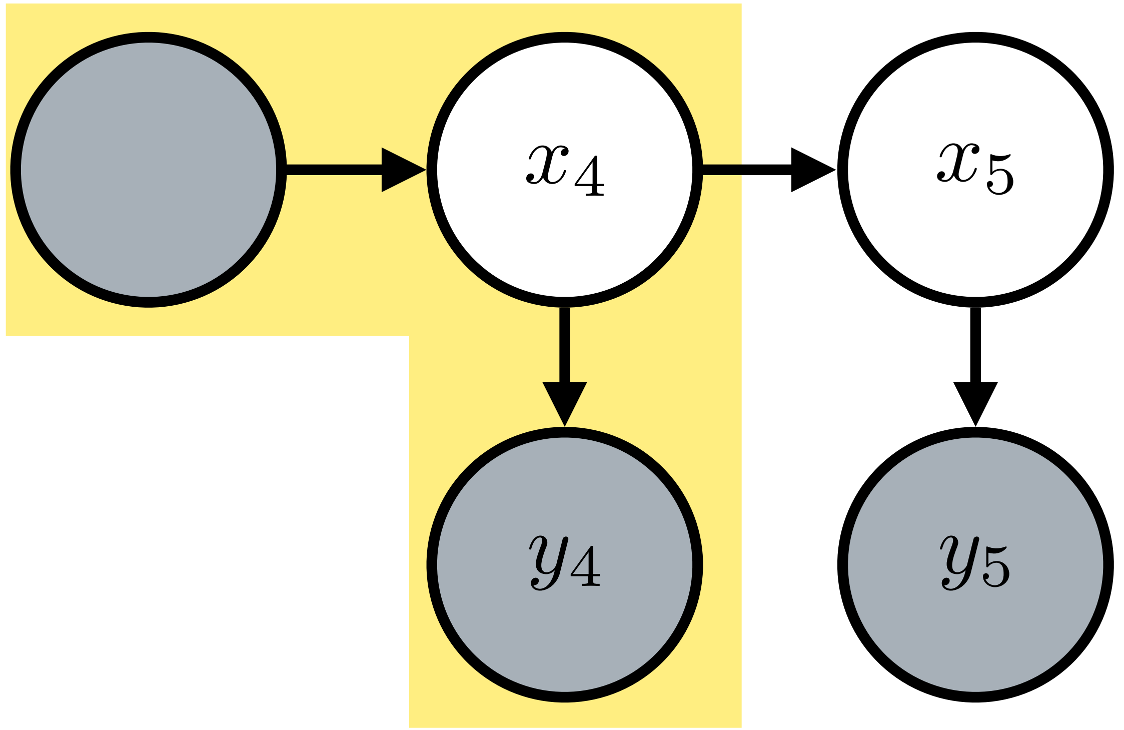
Exact Inference: The graphical way

marginalize over previous state



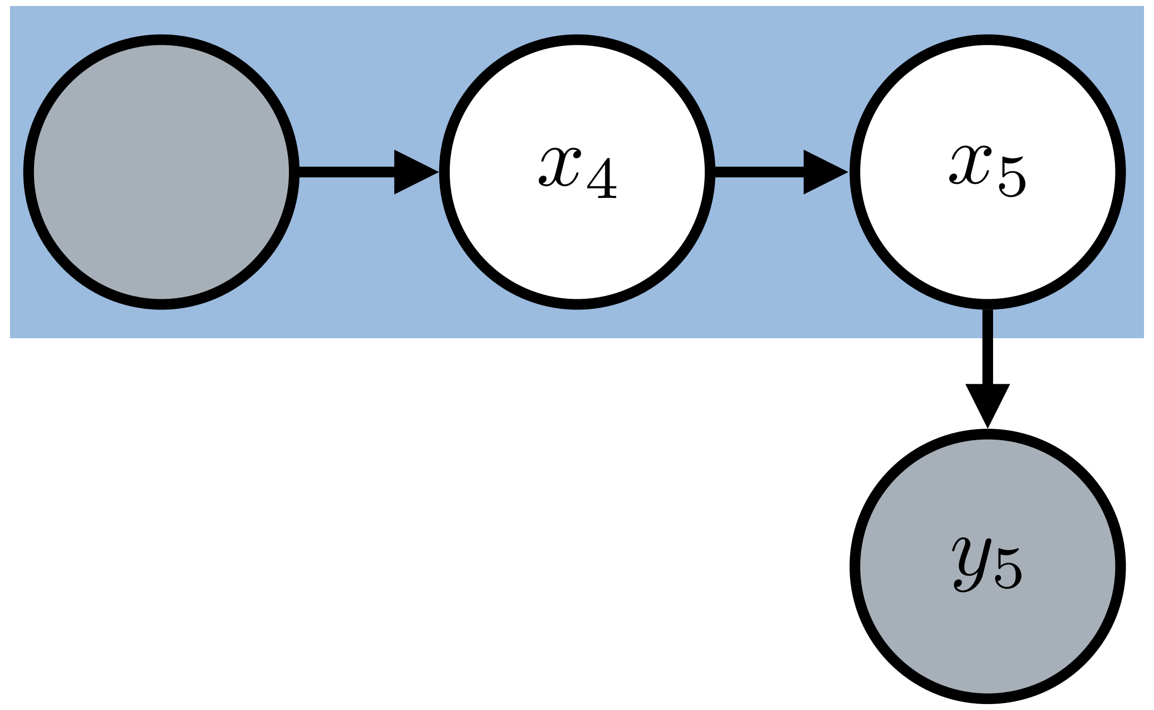
Exact Inference: The graphical way

condition
on
observations



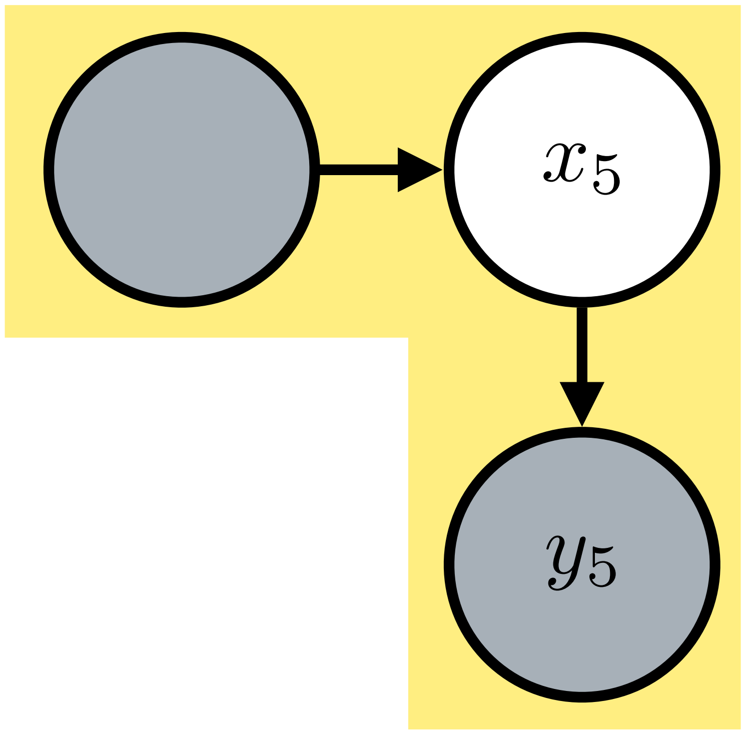
Exact Inference: The graphical way

marginalize
over previous
state

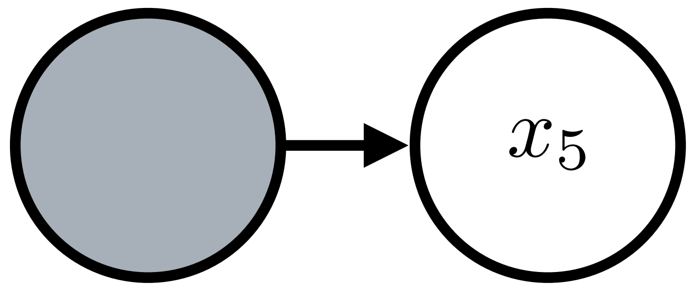


Exact Inference: The graphical way

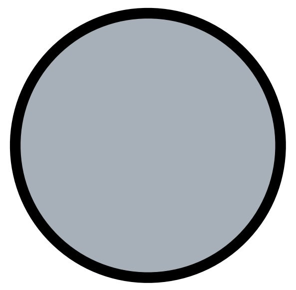
condition
on
observations



Exact Inference: The graphical way



Exact Inference: The graphical way



Message passing in chain-structured graphs

In “chain graphs,” the message passing recursion is:

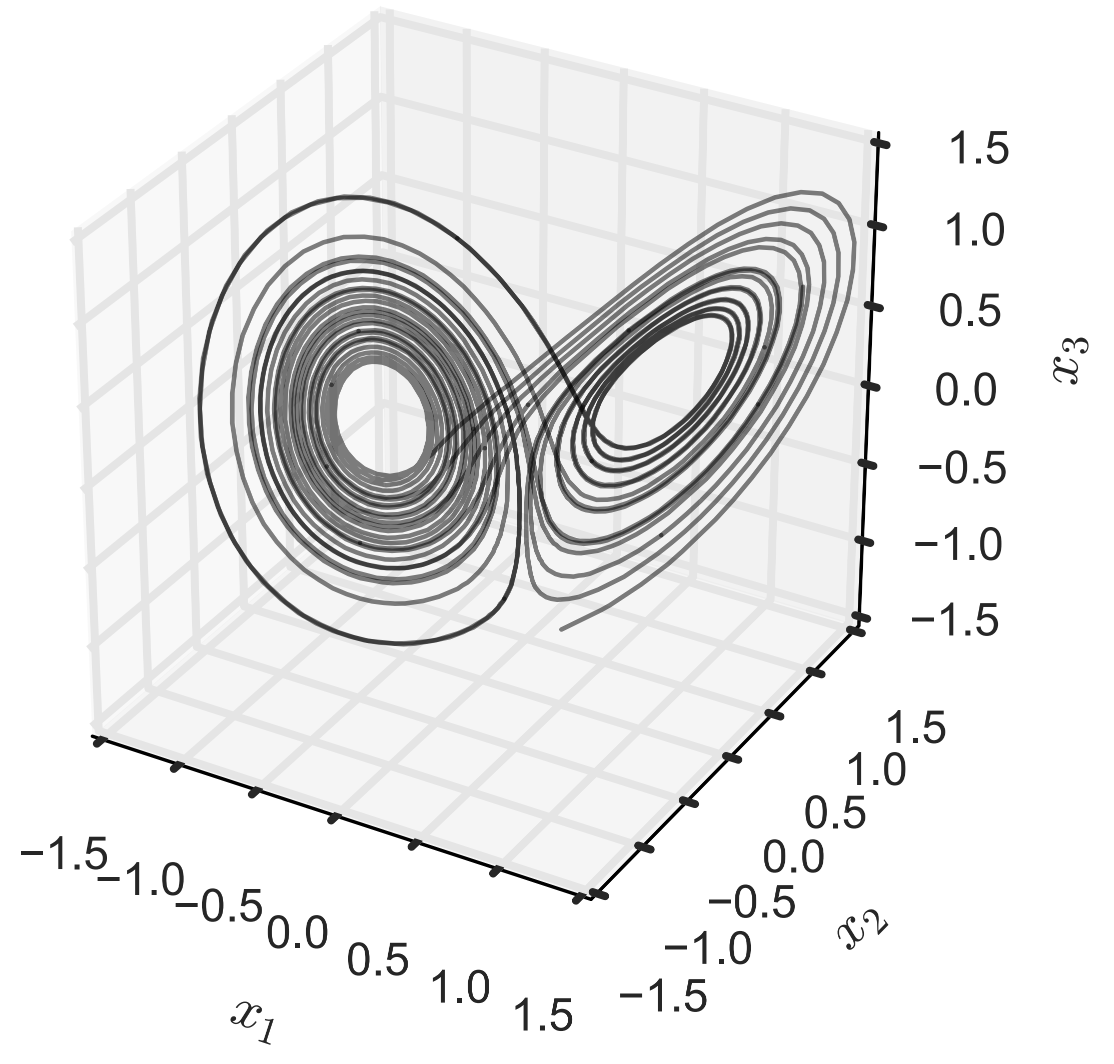
$$\alpha(x_{t+1}; y_{1:t}) = \int \alpha(x_t; y_{1:t-1})\, p(y_t \mid x_t)\, p(x_{t+1} \mid x_t)\, \mathrm{d}x_t$$

Few models admit closed form solutions.

The notable exception: **linear Gaussian** dynamics and observations.

I.e. the **Kalman filter.**

Approximate inference in nonlinear dynamical systems with Gaussian noise



$$\mathbf{x}_t \sim \mathcal{N}(f(\mathbf{x}_{t-1}), \mathbf{Q})$$
$$\mathbf{y}_t \sim \mathcal{N}(g(\mathbf{x}_t), \mathbf{R})$$

Many approximate inference methods:

- Extended Kalman Filter: linearize around the current posterior mean
- Unscented Kalman filter: approximate moments using sigma points
- Generalized Gaussian Filter: approximate moments using Gauss-Hermite quadrature
- Sequential Monte Carlo / particle filtering
- Markov chain Monte Carlo (MCMC)
- Variational Inference

Sequential Monte Carlo (SMC)

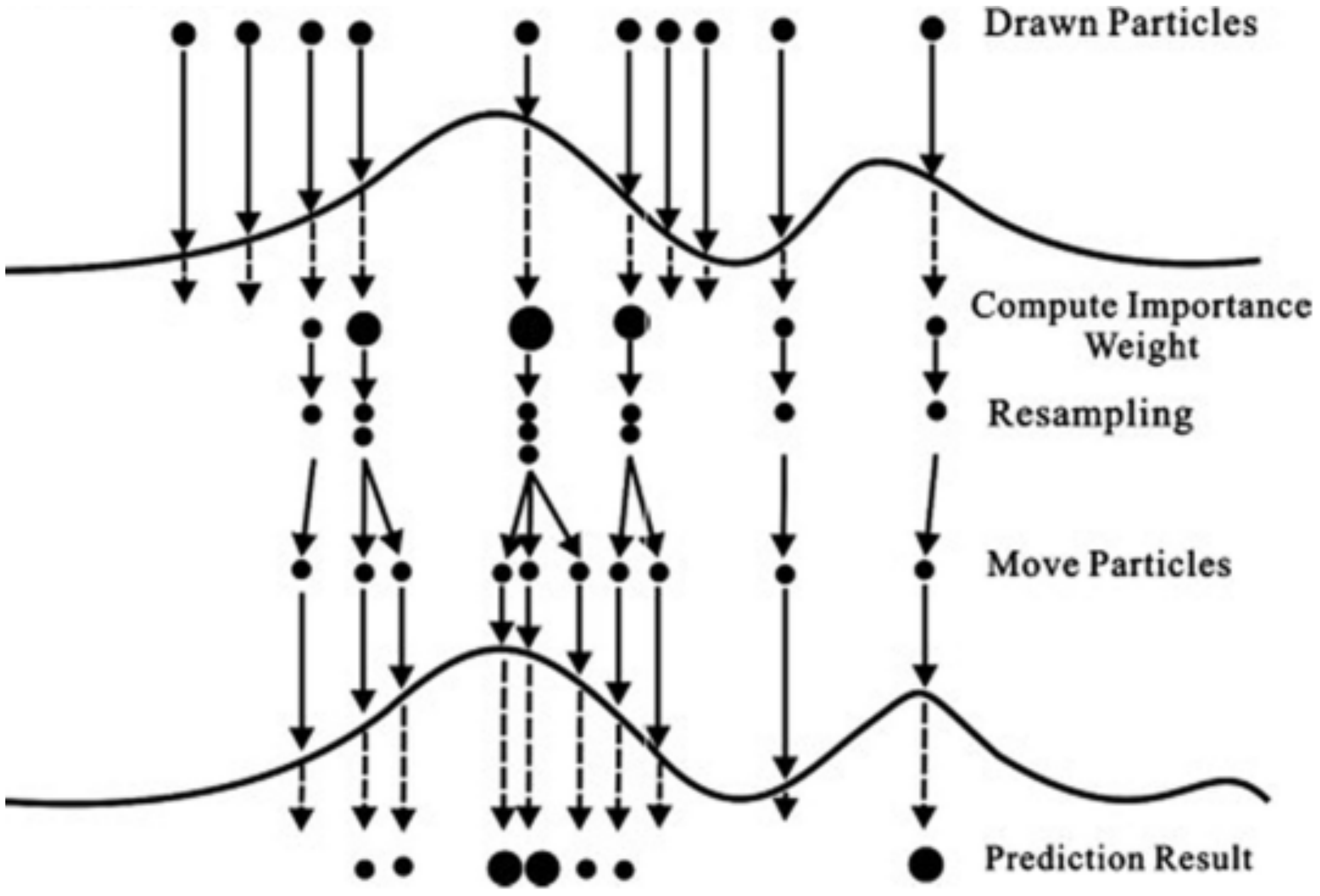
Idea: approximate the messages with **collection of weighted particles**

$$\alpha(x_{t+1}; y_{1:t}) = \int \alpha(x_t; y_{1:t-1})\, p(y_t \mid x_t)\, p(x_{t+1} \mid x_t)\, \mathrm{d}x_t$$
$$\approx \sum_{i=1}^{N} w_i \delta_{x_{t+1}^{(i)}}(x_{t+1})$$

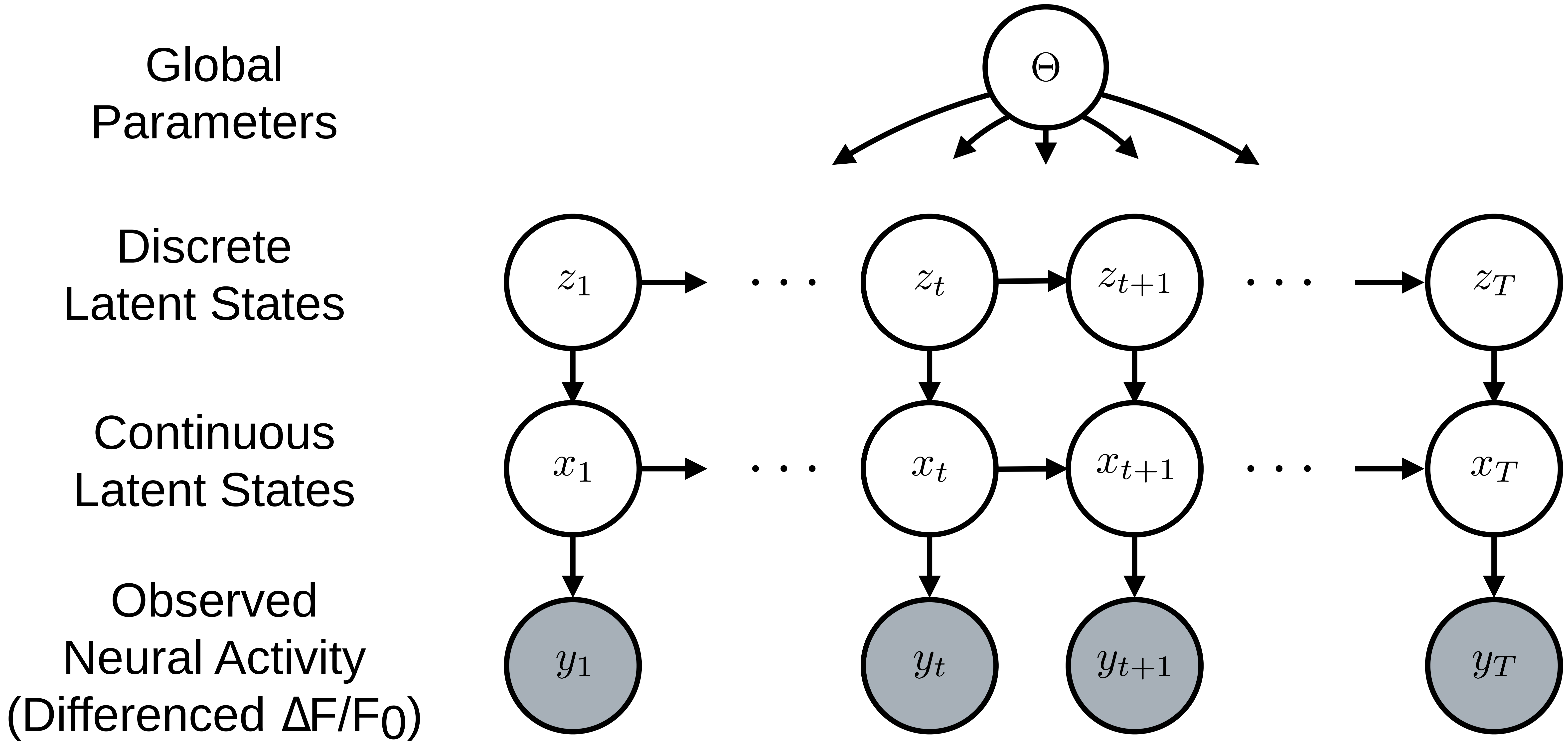
where the **importance weights** w_i are set based on the likelihood, transition, and proposal probabilities.

(We'll talk a lot more about SMC tomorrow!)

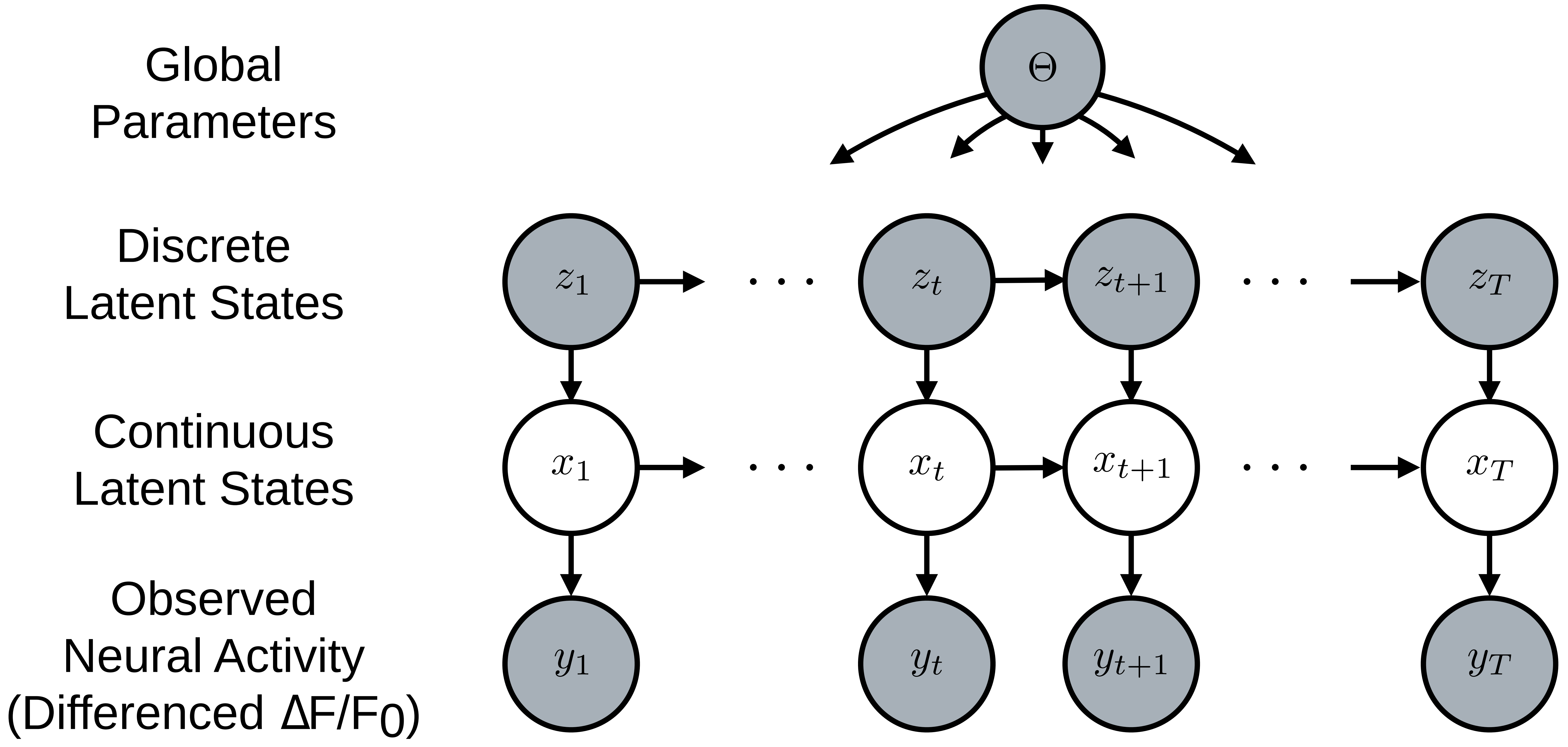
Sequential Monte Carlo



MCMC with Block Gibbs Sampling

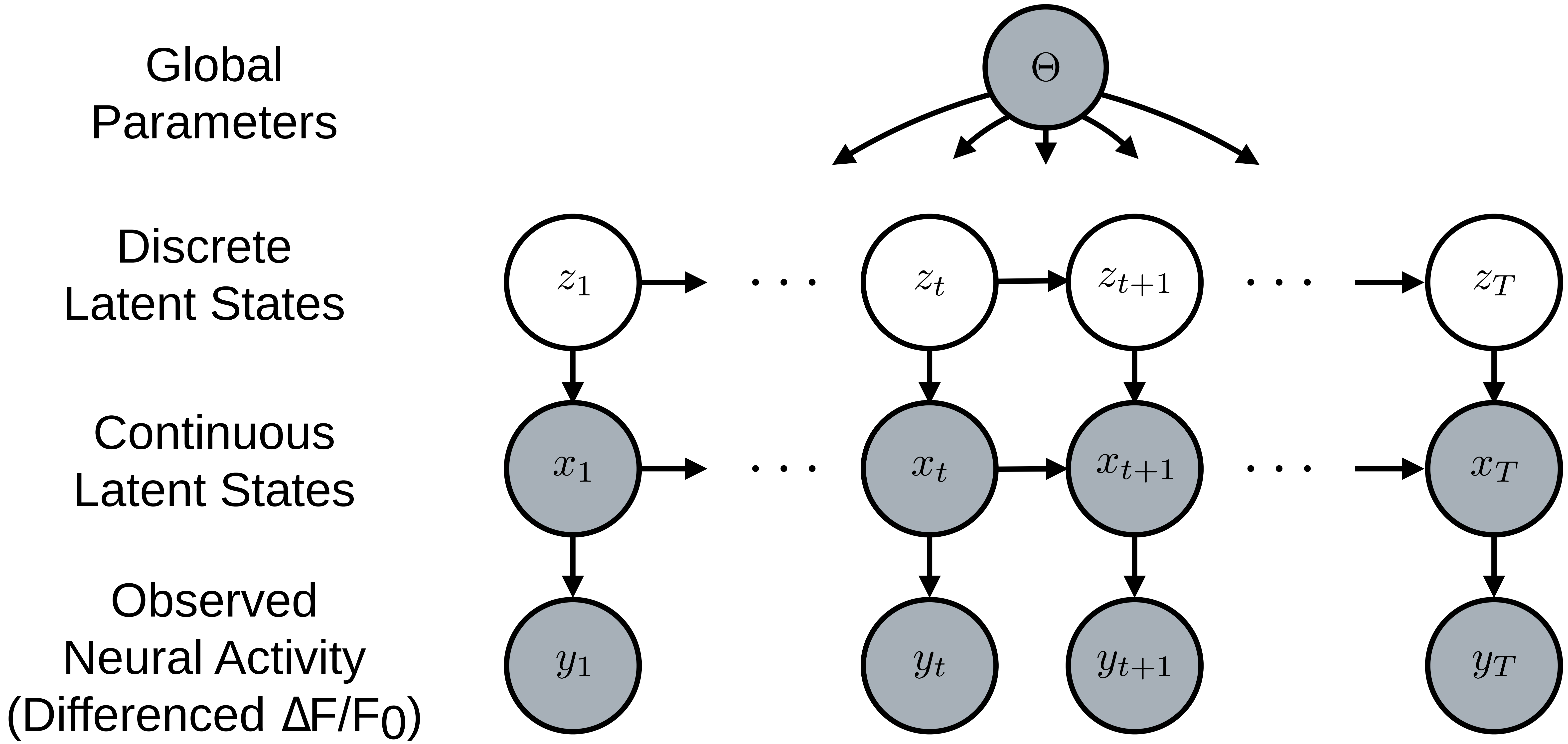


MCMC with Block Gibbs Sampling



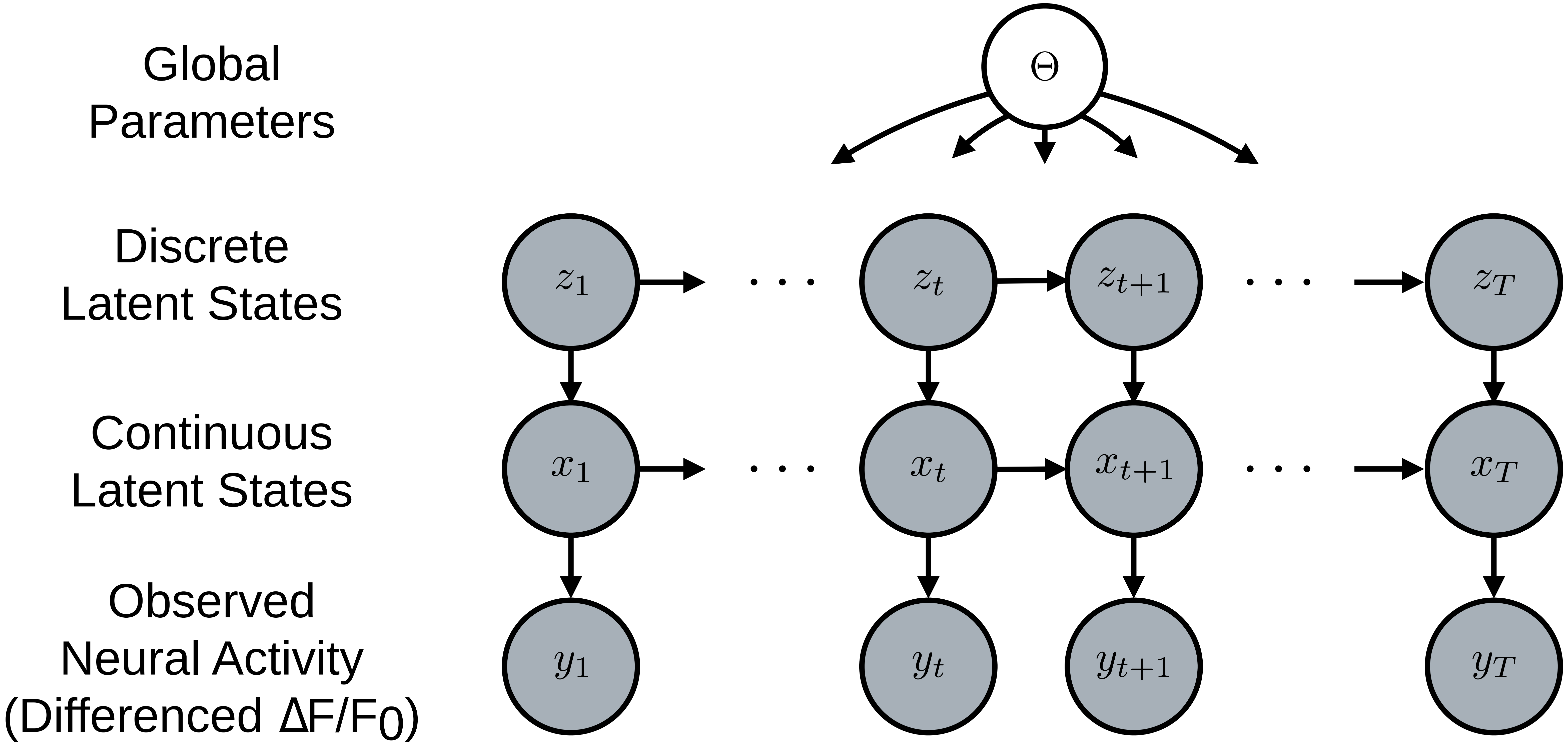
Given discrete states and parameters, the continuous states are easy to sample.

MCMC with Block Gibbs Sampling



Given continuous states and parameters, the discrete states are easy to sample.

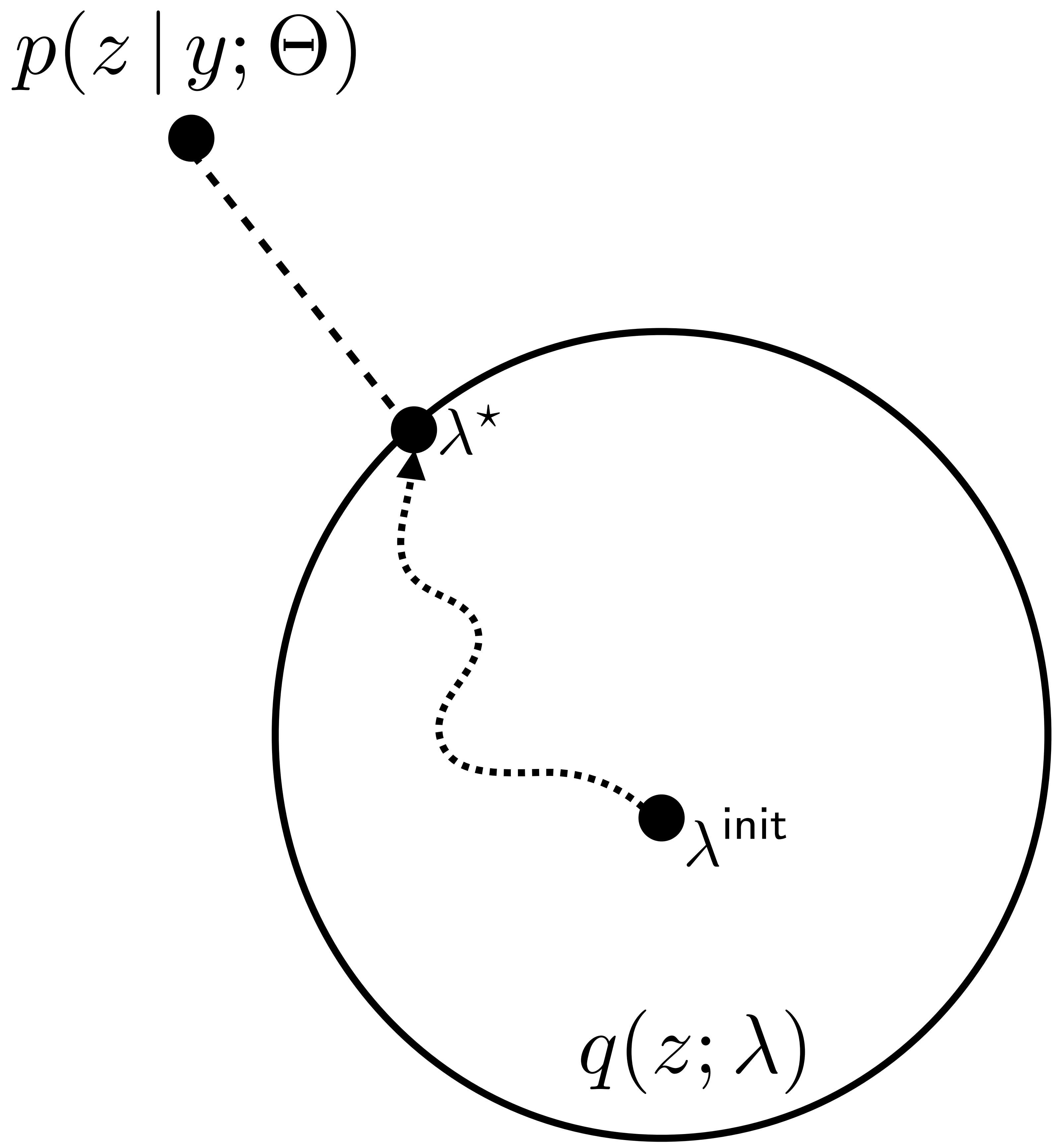
MCMC with Block Gibbs Sampling



Given continuous and discrete states, the parameters are easy to sample.

Variational Inference

Find an approximate posterior that minimizes the KL divergence to the true posterior.



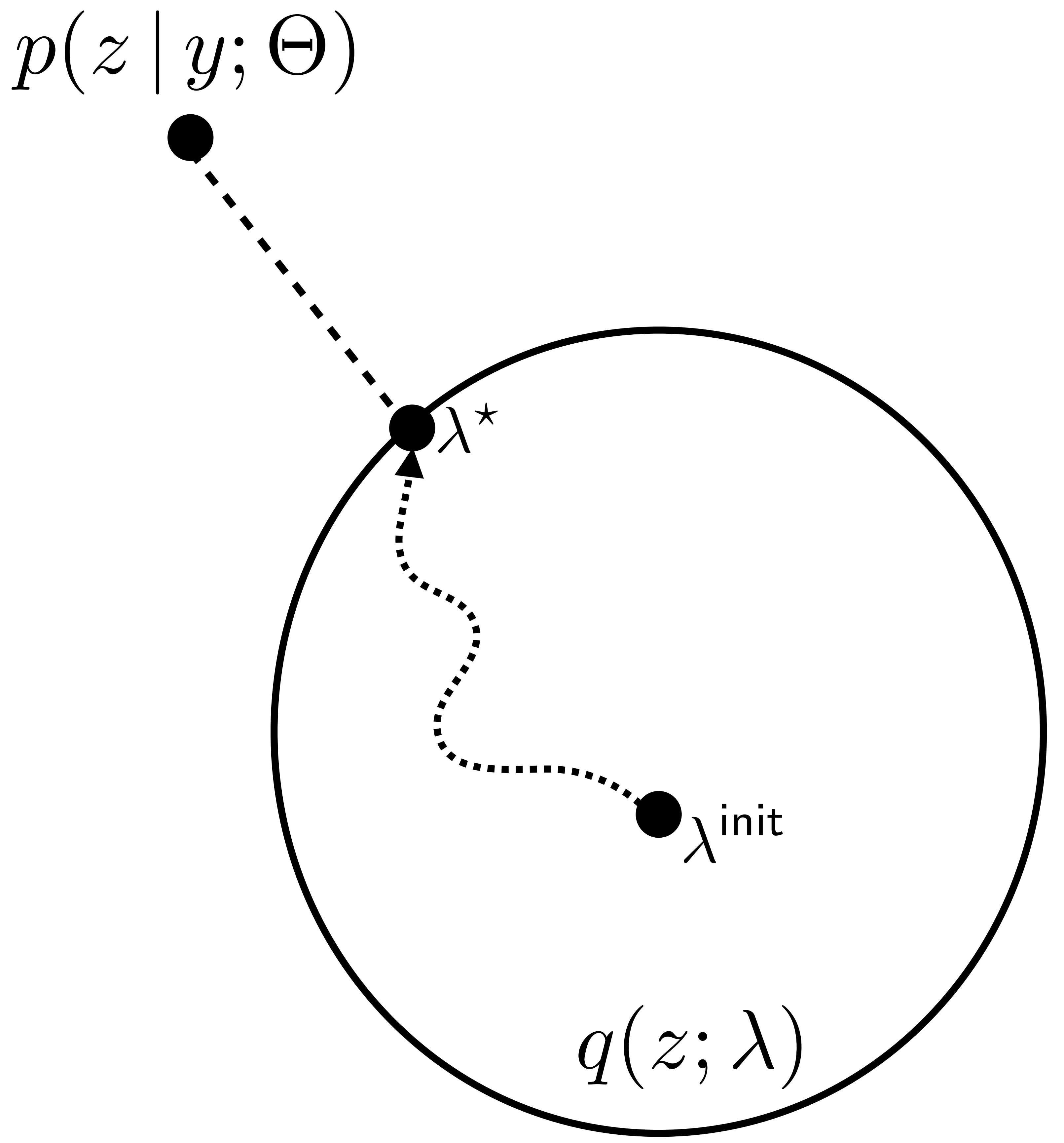
Blei, Kucukelbir, McAuliffe, JASA 2017.

Variational Inference

Find an approximate posterior that minimizes the KL divergence to the true posterior.

Minimizing KL is equivalent to maximizing the **ELBO**:

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(z;\lambda)} \left[\log p(z, y; \Theta) - \log q(z; \lambda)\right] \leq \log p(y; \Theta)$$



Blei, Kucukelbir, McAuliffe, JASA 2017.

Variational Inference

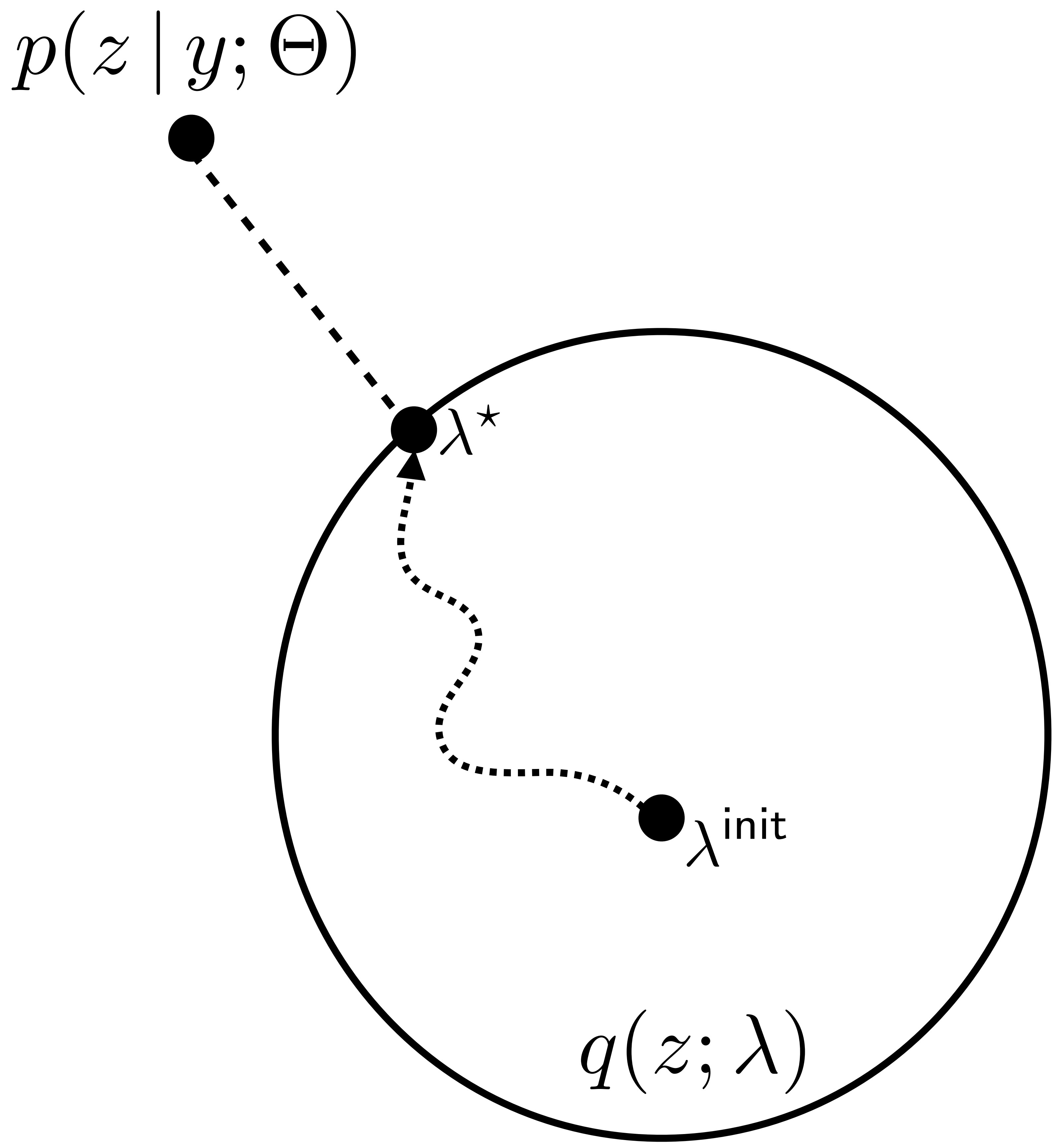
Find an approximate posterior that minimizes the KL divergence to the true posterior.

Minimizing KL is equivalent to maximizing the **ELBO**:

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(z;\lambda)} \left[\log p(z, y; \Theta) - \log q(z; \lambda)\right] \leq \log p(y; \Theta)$$

We can maximize the ELBO with (stochastic) gradient ascent, natural (preconditioned) gradient ascent, coordinate ascent, and combinations thereof.

More on this in Part 2!



Blei, Kucukelbir, McAuliffe, JASA 2017.

Learning with Expectation-Maximization

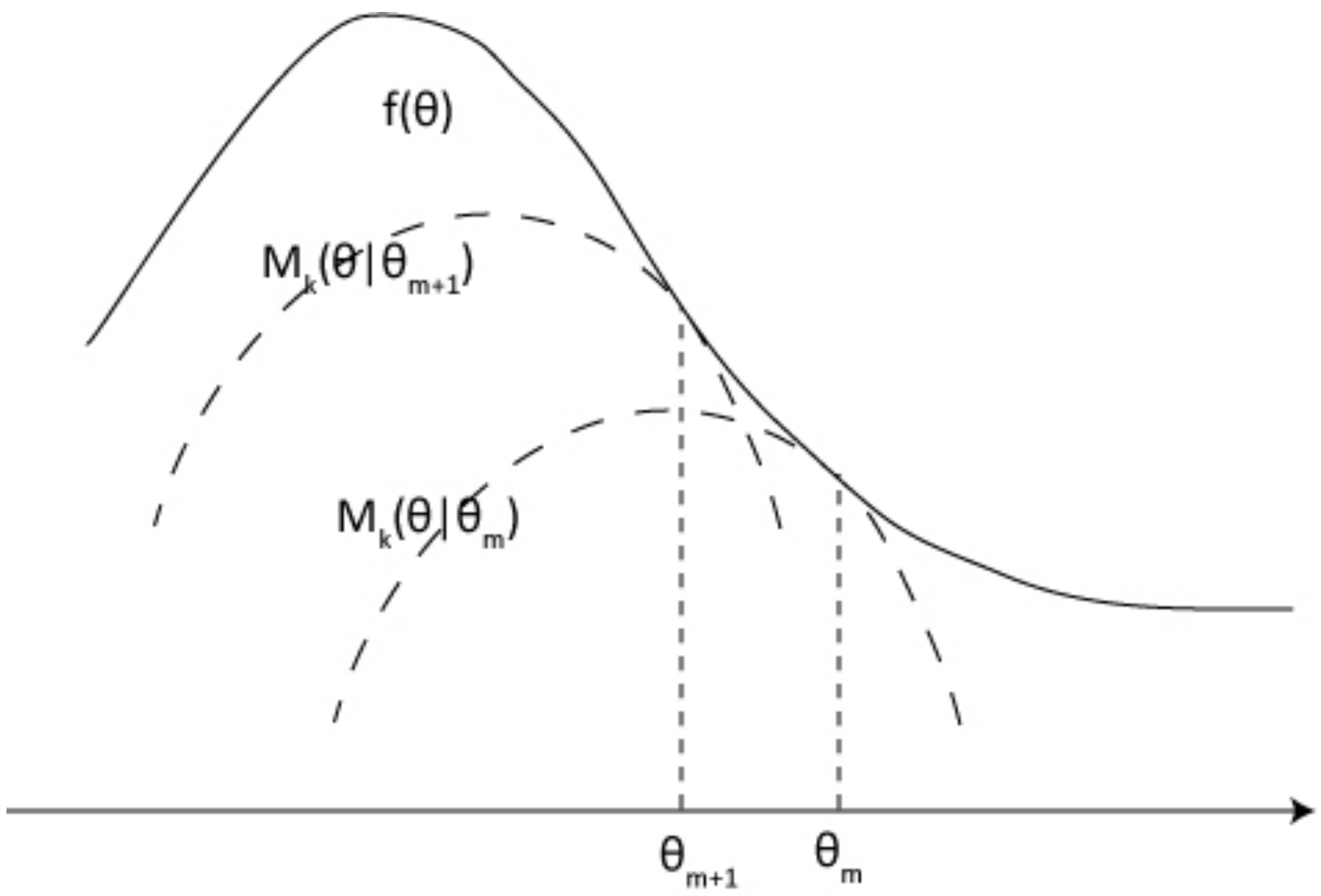
- **Idea:** iteratively maximize the marginal likelihood via a minorize-maximization (MM) algorithm.
- **E step:** Minorize the marginal log likelihood with **Jensen's inequality**:

$$\begin{aligned}\log p(y;\Theta) &\geq \mathbb{E}_{p(z|y;\Theta_m)}\left[\log p(z,y;\Theta) - \log p(z \mid y;\Theta_m)\right] \\ &\triangleq \mathcal{L}(\Theta;\Theta_m).\end{aligned}$$

- **M step:** Update parameters by **maximizing** the **bound**:

$$\Theta_{m+1} \leftarrow \arg\max_{\Theta} \mathcal{L}(\Theta;\Theta_m).$$

- Equivalently, this is **coordinate ascent** on parameters and the space of posterior distributions.
- We often substitute **approximate posteriors** in the minorization step, though we sacrifice some guarantees in doing so.



Outline

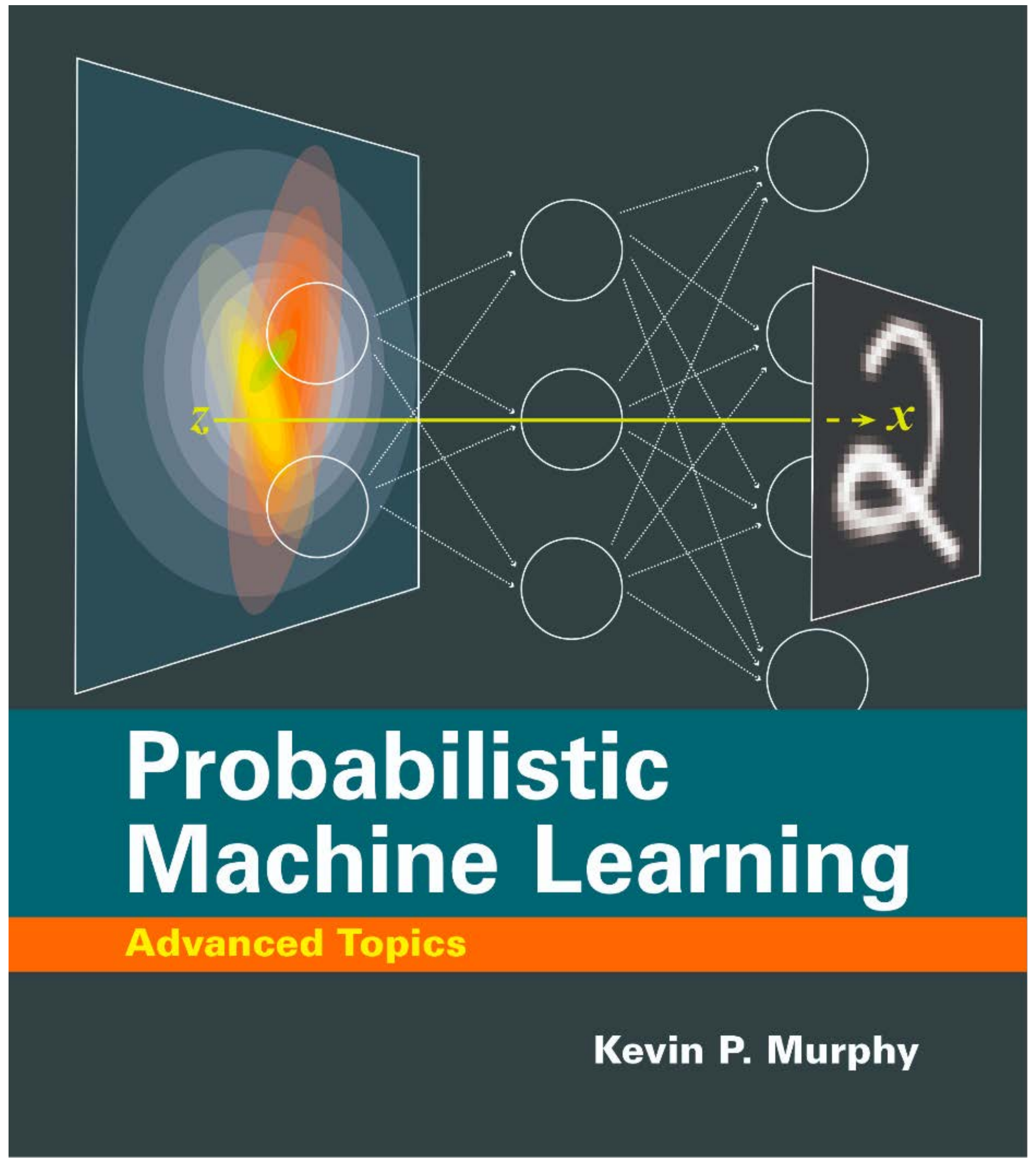
Part I: Foundations

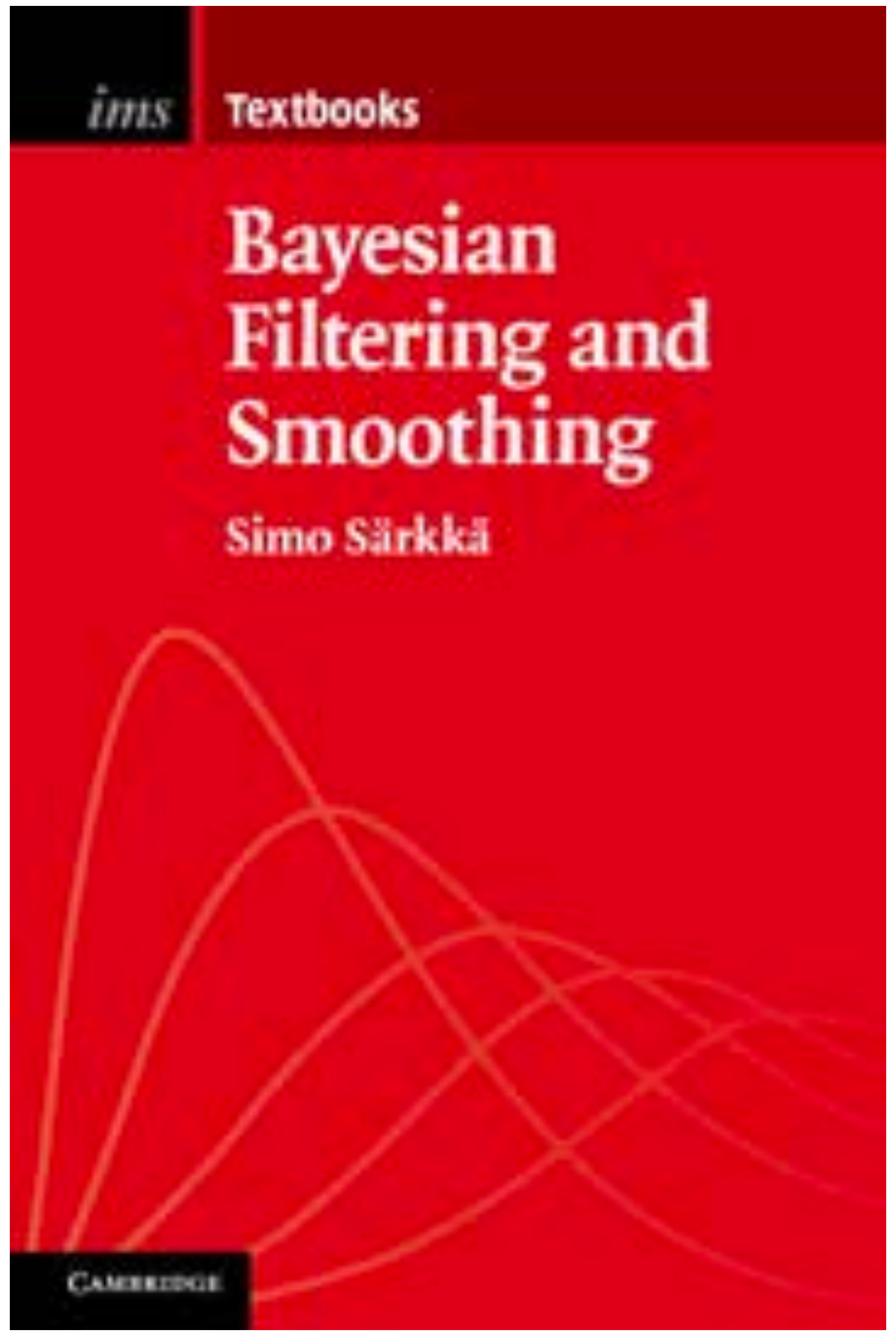
- Motivating Examples
- State Space Models (SSMs)
  - Hidden Markov Models
  - Linear Dynamical Systems
  - Nonlinear & Switching Linear Dynamical Systems
- Learning and Inference Algorithms
  - Expectation-Maximization
  - Message Passing
  - Approximate Inference (E/UKF, SMC, VI)
- **Code Pointers**

DYNAMAX

https://probml.github.io/dynamax/index.html

Further Reading





https://probml.github.io/book2

https://users.aalto.fi/~ssarkka/pub/cup_book_online_20131111.pdf

Outline

Part I: Foundations

- Motivating Examples
- State Space Models (SSMs)
  - Hidden Markov Models
  - Linear Dynamical Systems
  - Nonlinear & Switching Linear Dynamical Systems
- Learning and Inference Algorithms
  - Expectation-Maximization
  - Message Passing
  - Approximate Inference (E/UKF, SMC, VI)
- Code Pointers

Part II: Trends

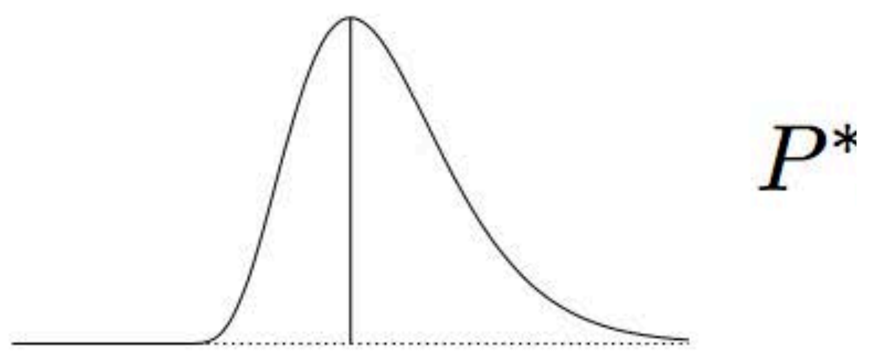
- Better Models
  - Time-Warped and Keypoint-MoSeq
  - Simple State Space Layers (S5)
- Better Algorithms
  - Variational Laplace-EM
  - Smoothing Inference with Twisted Objectives (SIXO)
  - Structured Variational Autoencoders (SVAE)

Acknowledgements

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

Laplace Approximation

1. View the joint as an unnormalized density on latent variables.

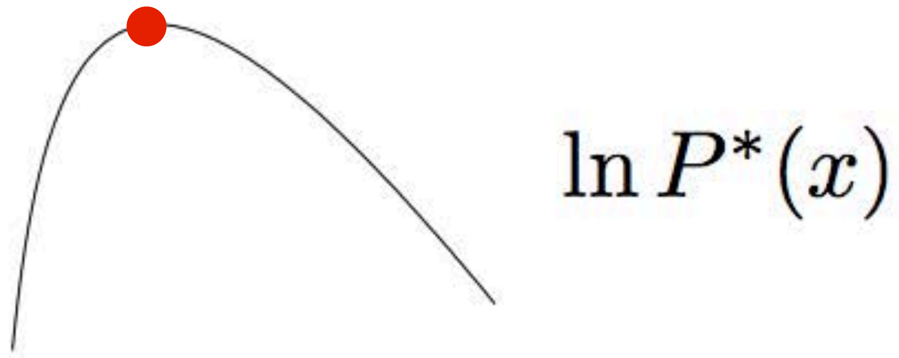


$$P^*(x) = p(x, y)$$

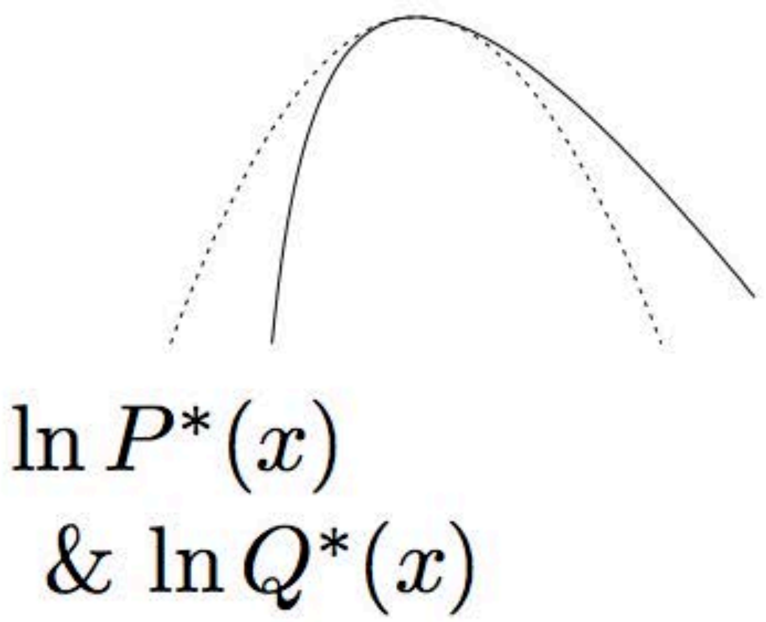
$$Z_P = \int P^*(x)\,\mathrm{d}x = p(y)$$

2. Find the mode.

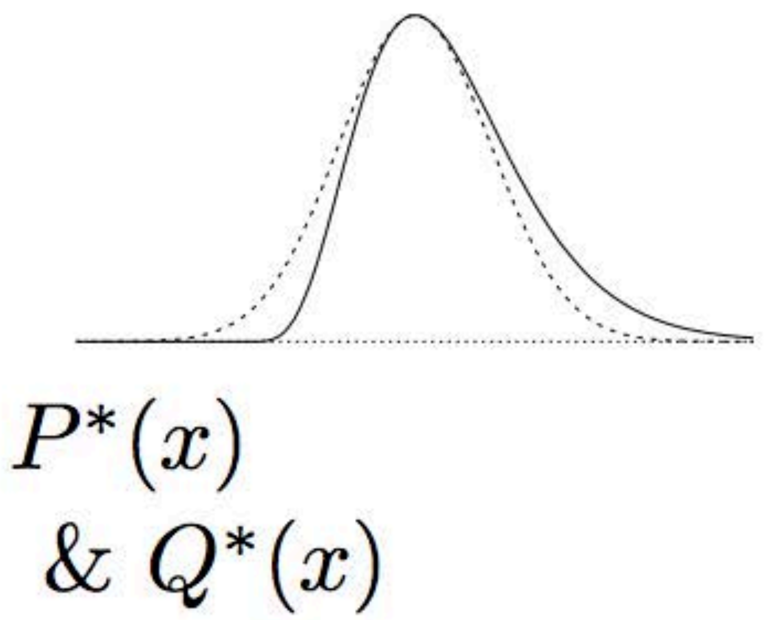
$$x^* = \arg\max P^*(x)$$



3. Form a 2nd order Taylor approximation around the mode.



4. Exponentiate to get an unnormalized Gaussian. Compute its normalization constant.



$$\ln Q^*(x) = \ln P^*(x^*) - \frac{1}{2}(x - x^*)^\top A (x - x^*)$$

$$A = -\nabla^2 \ln P^*(x)$$

$$Z_P \approx Z_Q = P^*(x^*)(2\pi)^{\frac{D}{2}} |A|^{-\frac{1}{2}}$$

Graphics adapted from MacKay (2003, Ch 27)