Machine Learning Methods for Neural Data Analysis

Lecture 13: Switching linear dynamical systems

Scott Linderman

STATS 220/320 (NBIO220, CS339N). Winter 2021.

Announcements

- **Lab 7 Errata**:
  - Don’t change log_likes in-place in Problem 1a. The test function reuses it for computing the reference answers.
  - Typo in the commented functions for downloading crowd movies.
  - How to fill in the feature vector for first couple time steps?
- **Updated proposals** due this Friday.
  - Looking for progress on downloading and extracting the data.

Agenda

- Switching linear dynamical systems (SLDS)
- Hardness of exact EM for SLDS
- Approximate message passing

Recap: Gaussian HMM

Generative Model:

$$
\begin{aligned}
z_1 &\sim \mathrm{Cat}(\pi), \\
z_t \mid z_{t-1} &\sim \mathrm{Cat}(P_{z_{t-1}}), && \text{for } t = 2, \ldots, T. \\
x_t \mid z_t &\sim \mathcal{N}(b_{z_t}, Q_{z_t}) && \text{for } t = 1, \ldots, T
\end{aligned}
$$

Parameters:

$$\Theta = \pi, P, \{b_k, Q_k\}_{k=1}^{K}$$

Joint probability:

$$p(x, z \mid \Theta) = p(z_1) \prod_{t=1}^{T-1} p(z_{t+1} \mid z_t) \prod_{t=1}^{T} p(x_t \mid z_t)$$

Recap: The Gaussian HMM

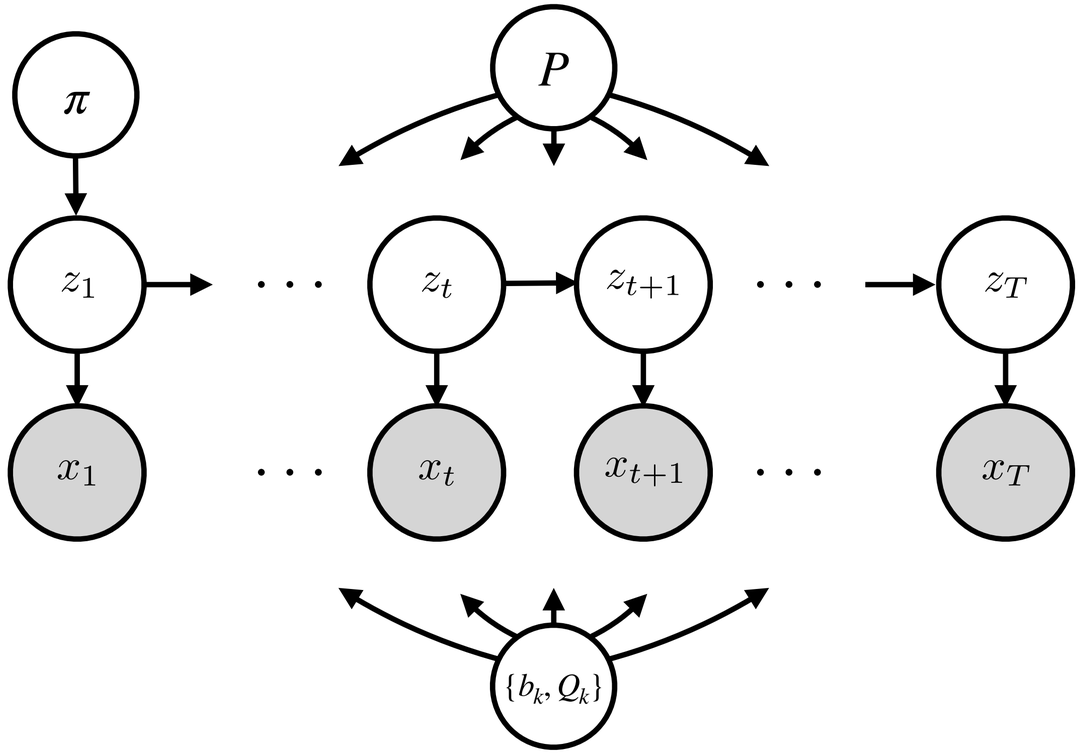
Graphical Model

Transition Probabilities

Discrete Latent States

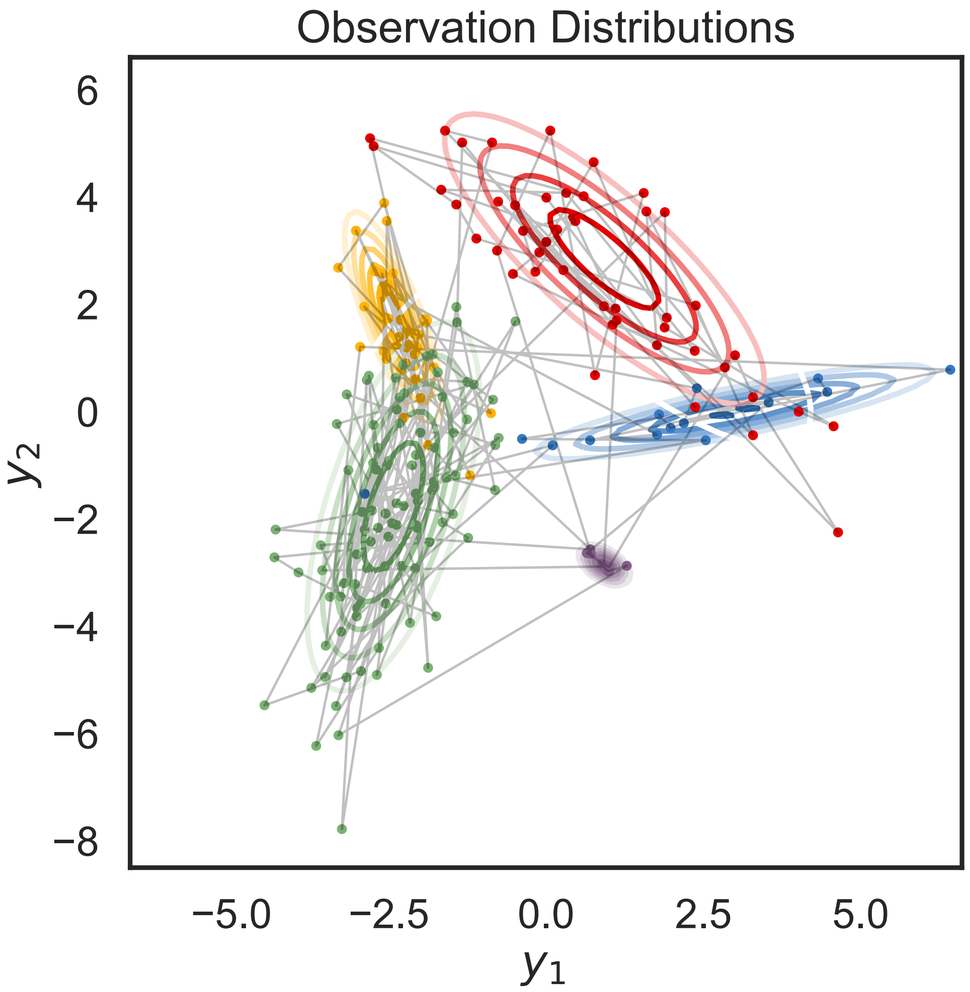
Observations (e.g. PCA loadings of each frame)

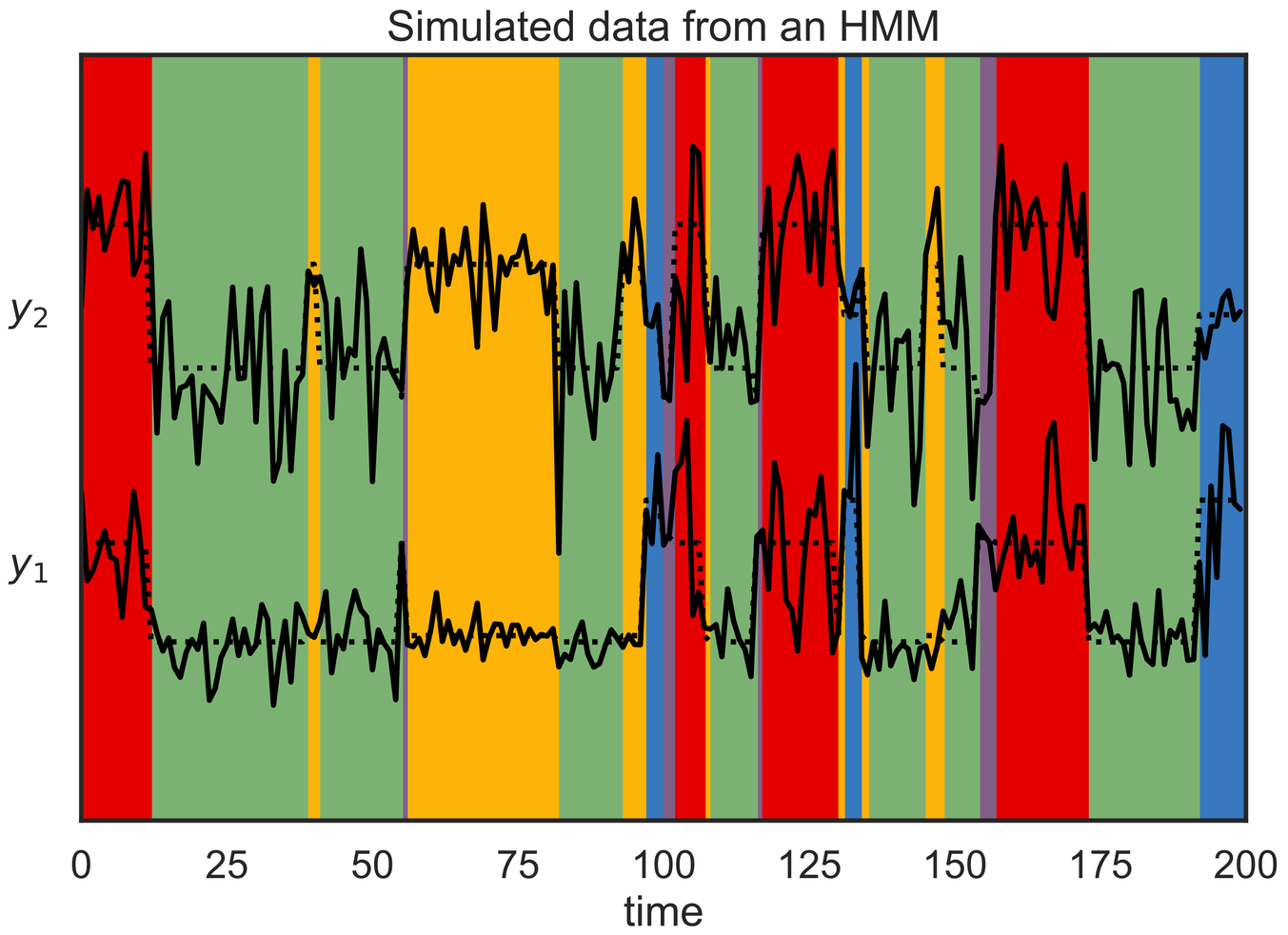
State Means and Covariances



= latent = observed = dependency

Simulated data from a Gaussian HMM





Recap: Autoregressive (AR) HMM

Generative Model:

$$
\begin{aligned}
z_1 &\sim \mathrm{Cat}(\pi), \\
z_t \mid z_{t-1} &\sim \mathrm{Cat}(P_{z_{t-1}}), && \text{for } t = 2, \ldots, T \\
x_1 \mid z_1 &\sim \mathcal{N}(b_{z_1}, Q_{z_1}) && . \\
x_t \mid x_{t-1}, z_t &\sim \mathcal{N}(A_{z_t} x_{t-1} + b_{z_t}, Q_{z_t}) && \text{for } t = 1, \ldots, T
\end{aligned}
$$

Parameters:

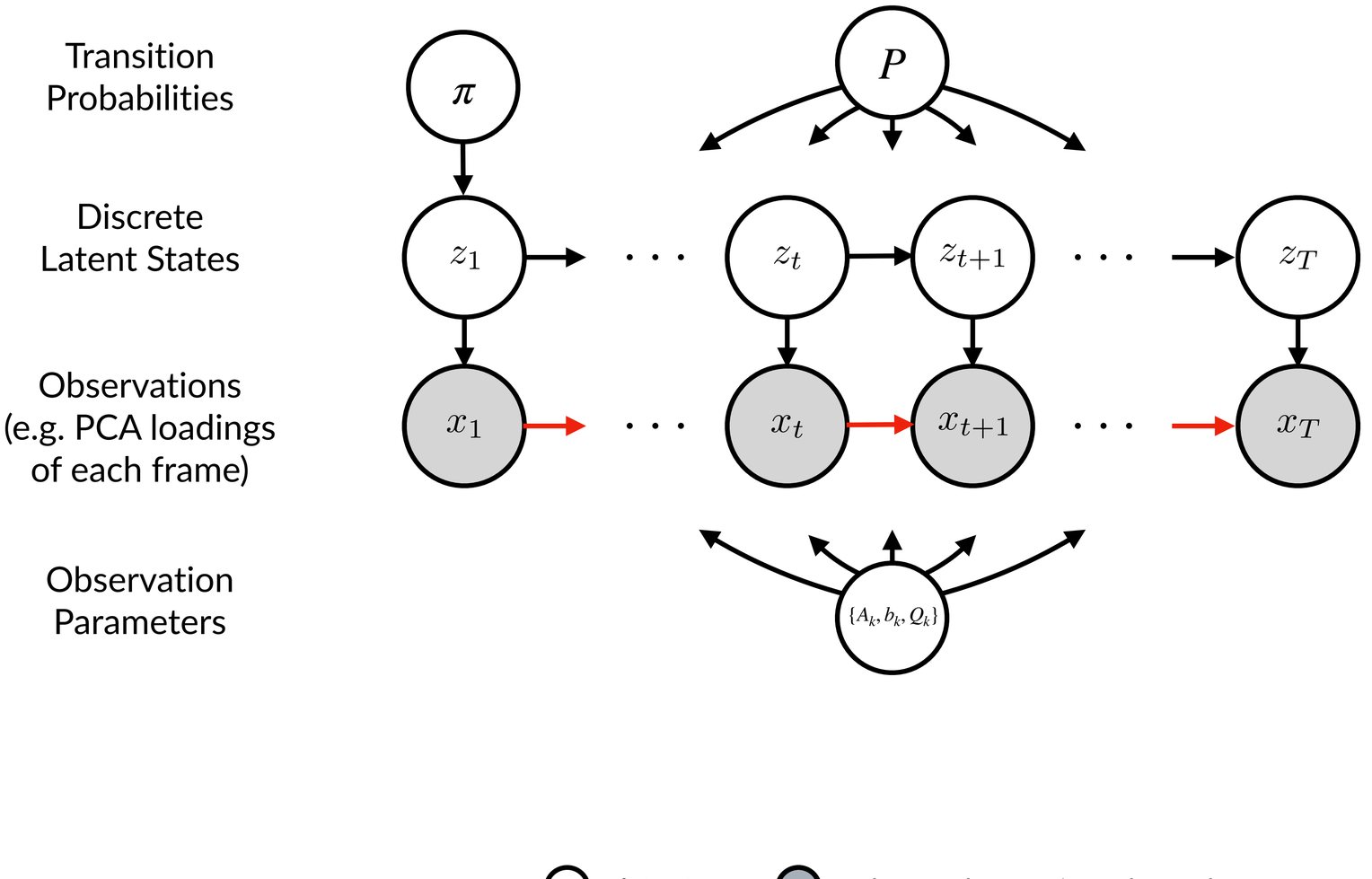
$$\Theta = \pi, P, \{A_k, b_k, Q_k\}_{k=1}^K$$

Joint probability:

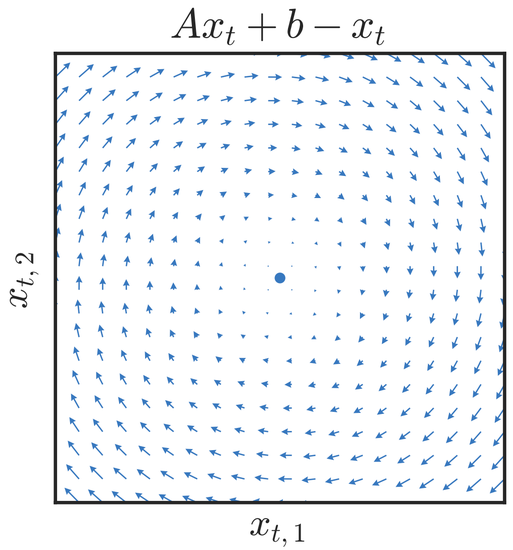
$$p(x, z \mid \Theta) = p(z_1)\, p(x_1 \mid z_1) \prod_{t=1}^{T-1} p(z_{t+1} \mid z_t) \prod_{t=1}^{T} p(x_t \mid x_{t-1}, z_t)$$

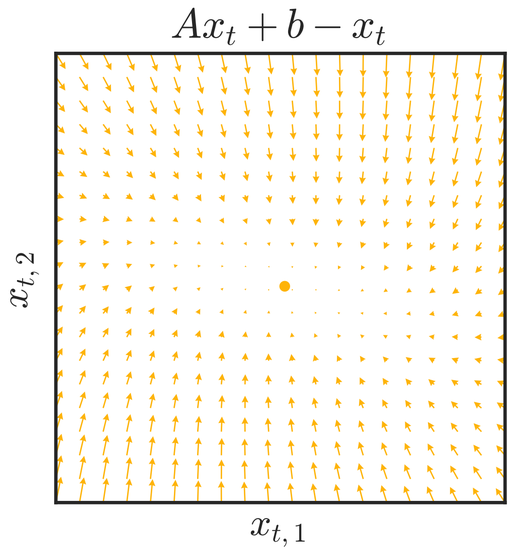
Recap: Autoregressive (AR) HMM

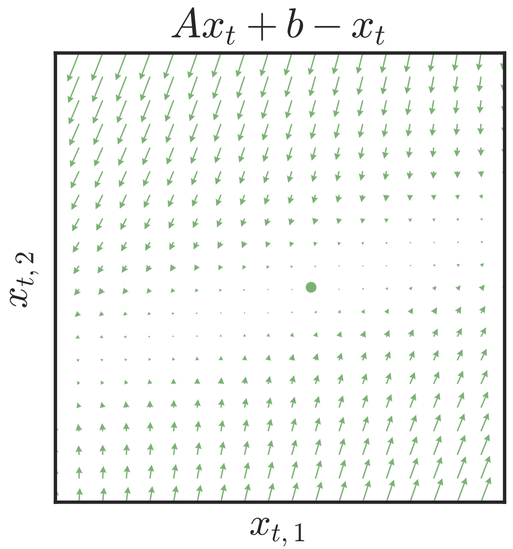
Graphical Model

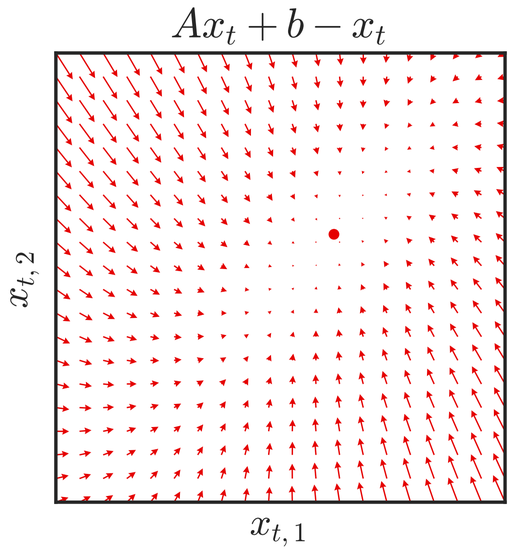


Visualizing Linear Dynamics

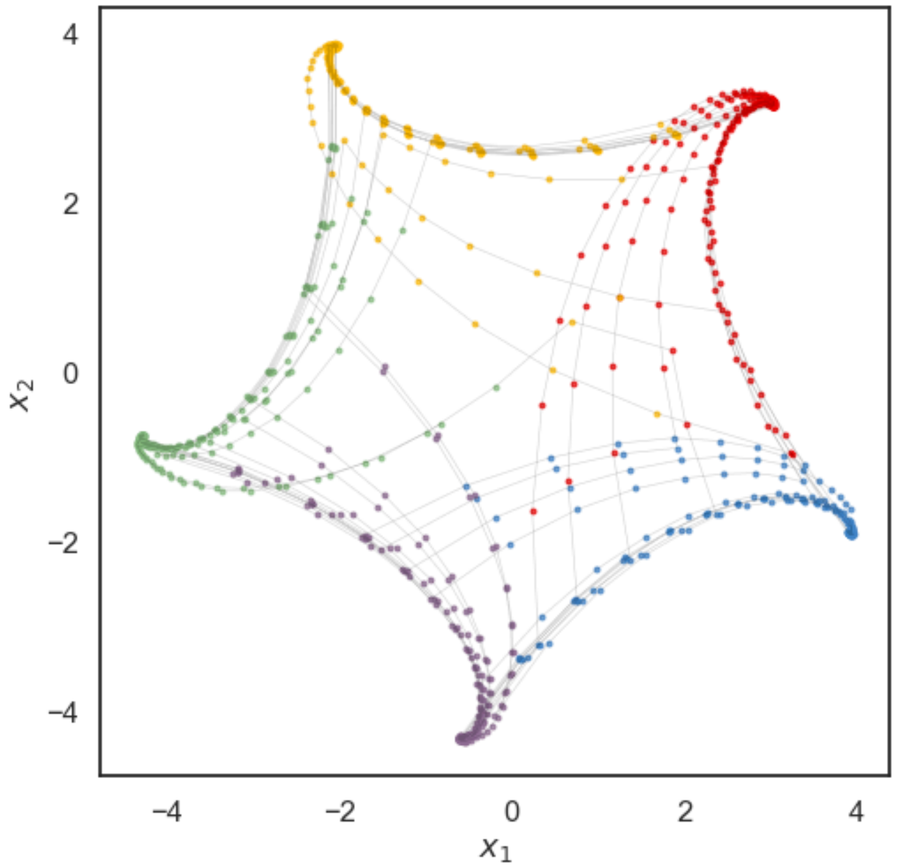


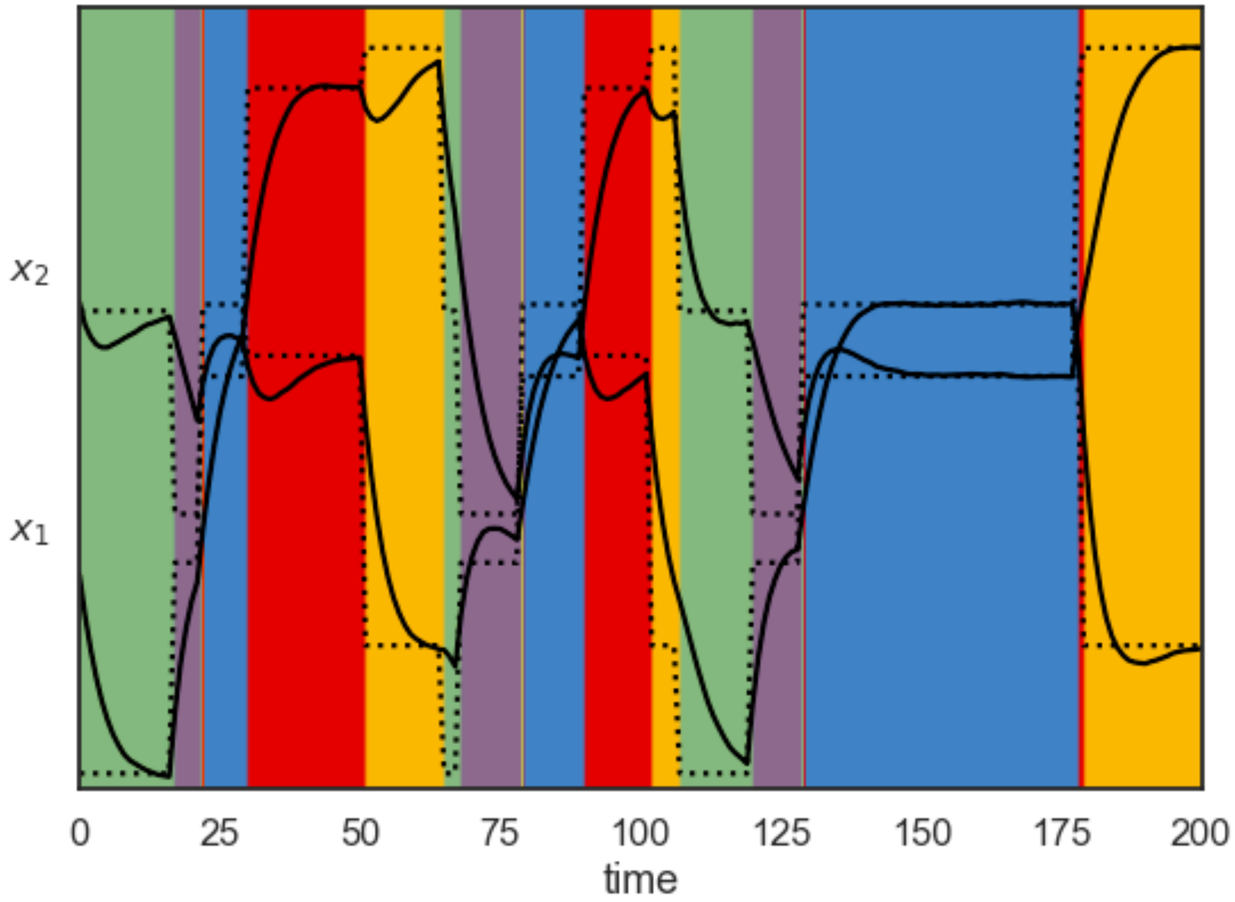






Simulated data from an ARHMM





Recap: Gaussian Linear Dynamical Systems

Generative Model:

$$
\begin{aligned}
x_1 &\sim \mathcal{N}(b, Q), \\
x_t \mid x_{t-1} &\sim \mathcal{N}(Ax_{t-1} + b, Q), && \text{for } t = 2, \ldots, T. \\
y_t \mid x_t &\sim \mathcal{N}(Cx_t + d, R) && \text{for } t = 1, \ldots, T
\end{aligned}
$$

Parameters:

$$\Theta = A, b, Q, C, d, R$$

Joint probability:

$$p(y, x \mid \Theta) = p(x_1) \prod_{t=1}^{T-1} p(x_{t+1} \mid x_t) \prod_{t=1}^{T} p(y_t \mid x_t)$$

Recap: Gaussian Linear Dynamical Systems

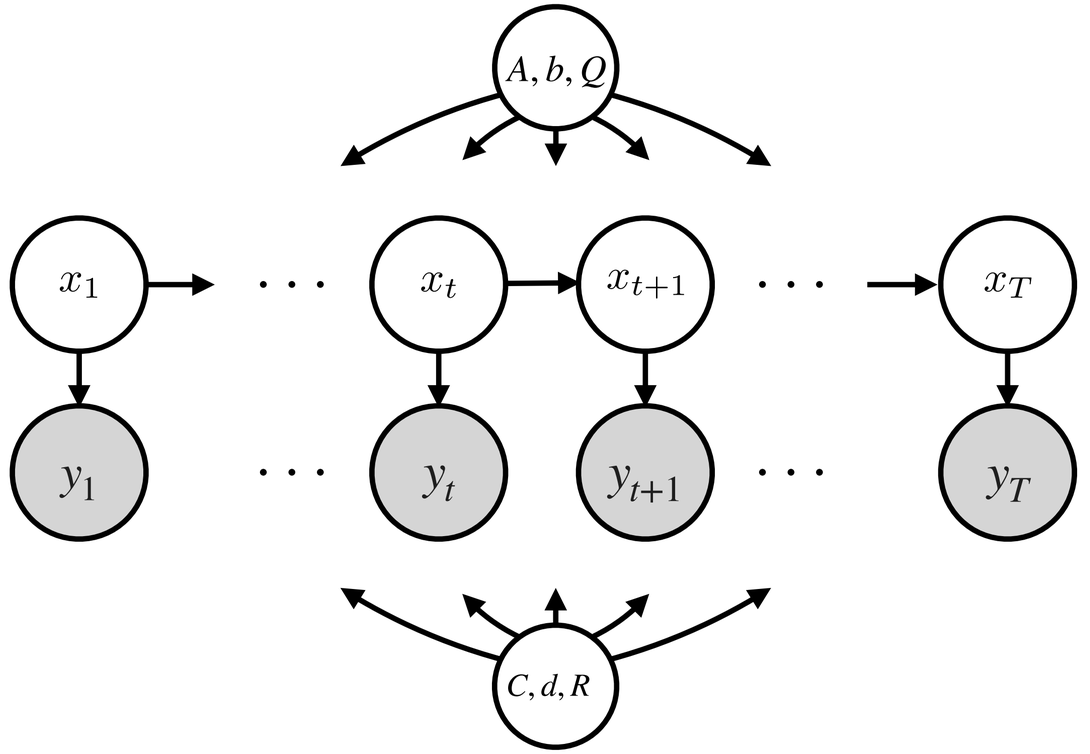
Graphical Model

Transition Probabilities

Discrete Latent States

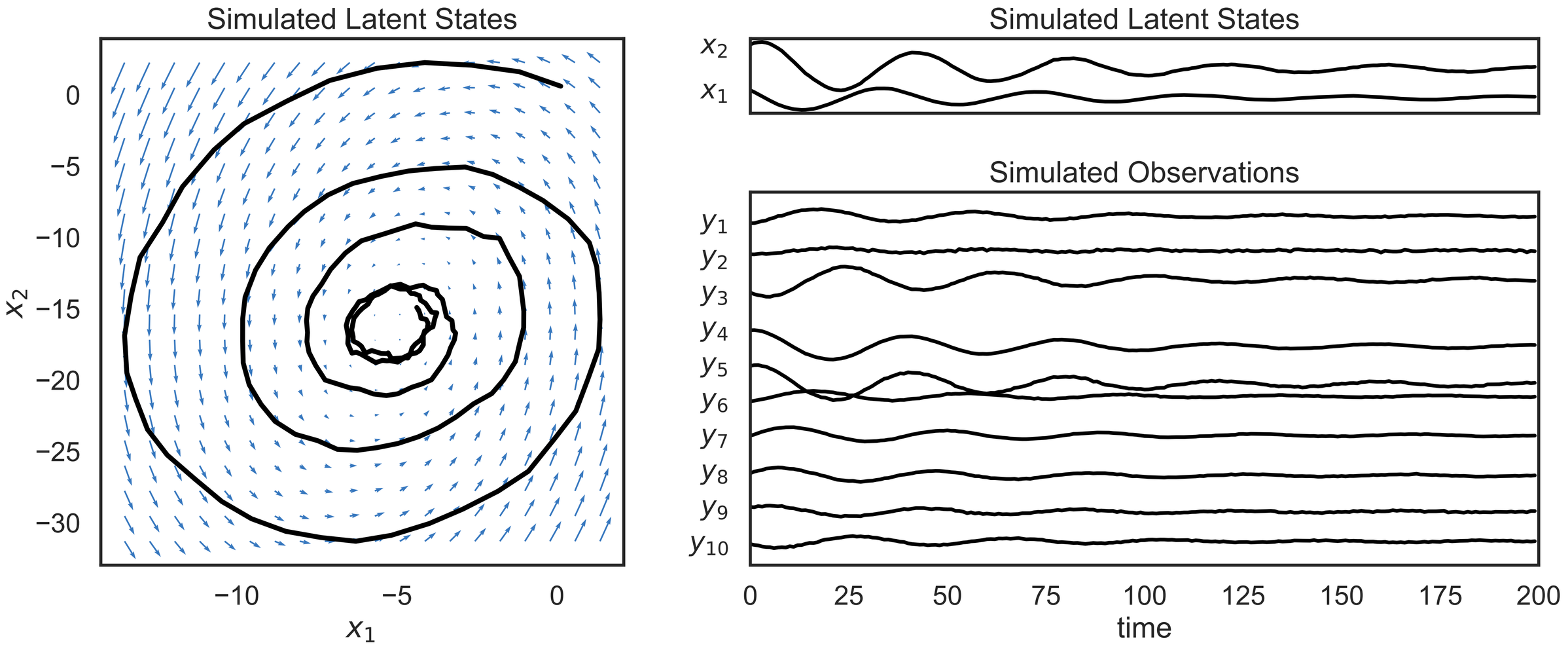
Observations (e.g. PCA loadings of each frame)

State Means and Covariances



= latent = observed = dependency

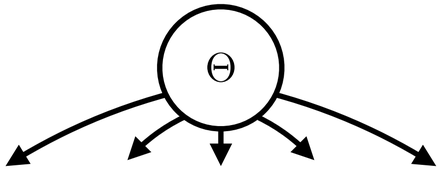
Simulated data from an LDS



Switching LDS: Best of Both Worlds

Switching LDS: Best of Both Worlds

Global
Parameters

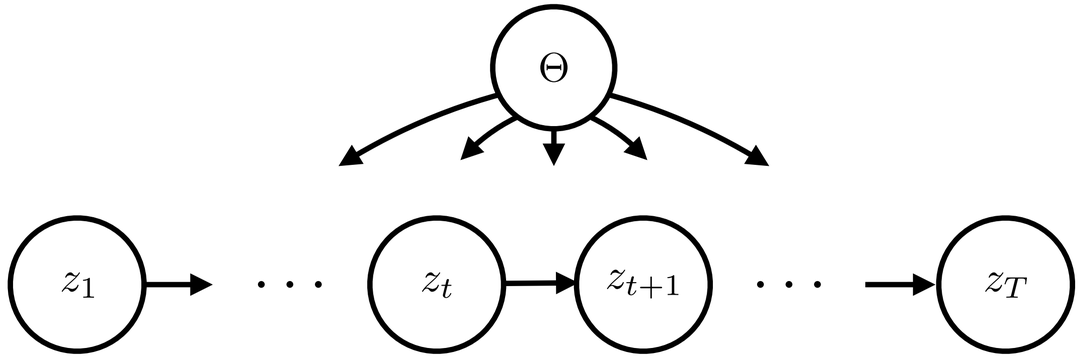


= latent = observed = dependency

Switching LDS: Best of Both Worlds

Global Parameters

Discrete Latent States



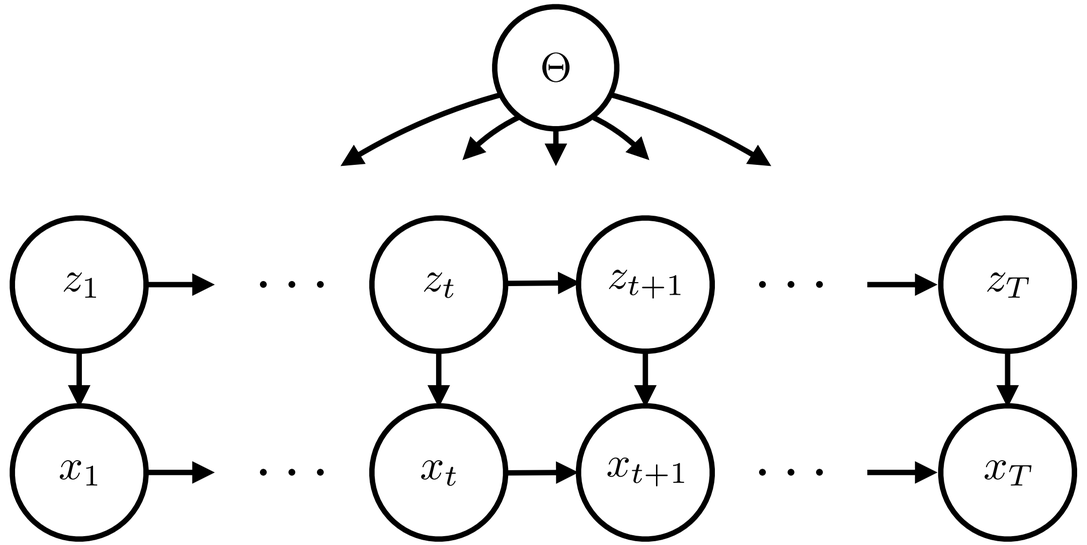
= latent = observed = dependency

Switching LDS: Best of Both Worlds

Global Parameters

Discrete Latent States

Continuous Latent States



= latent = observed = dependency

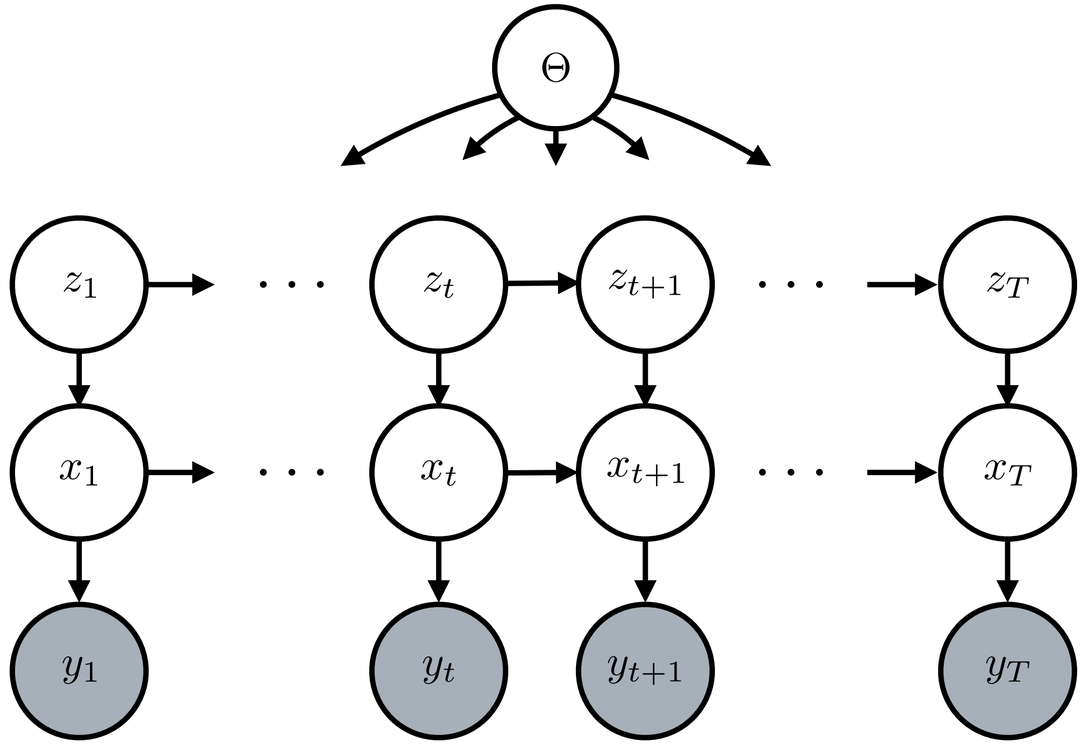
Switching LDS: Best of Both Worlds

Global Parameters

Discrete Latent States

Continuous Latent States

Observed Neural Activity ($\Delta F/F_0$)

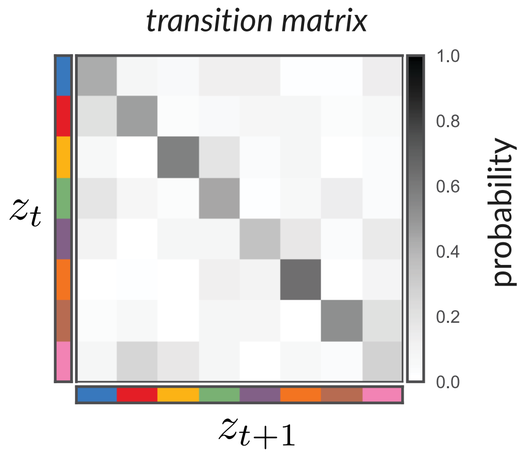


= latent = observed = dependency

Specifying the form of the dependencies

Specifying the form of the dependencies

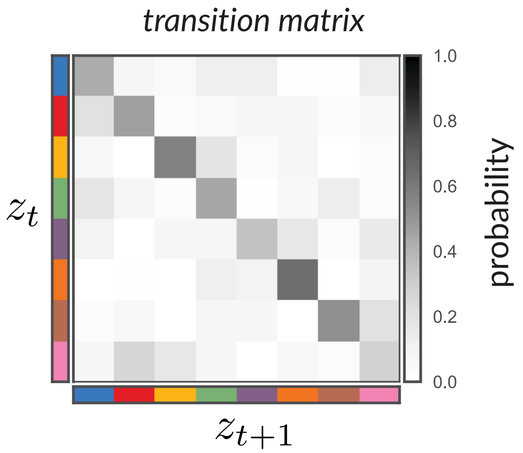
State-dependent
switching probabilities



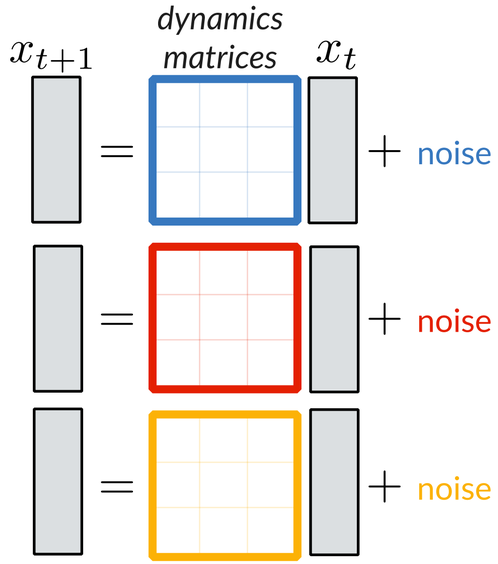
$$\Pr(z_{t+1} = j \mid z_t = i) = P_{ij}$$

Specifying the form of the dependencies

State-dependent
switching probabilities



Different **linear dynamics**
in each discrete state

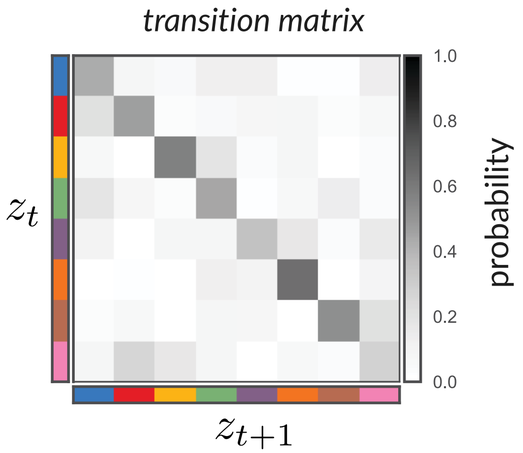


$$\Pr(z_{t+1} = j \mid z_t = i) = P_{ij}$$

$$x_{t+1} = A_{z_{t+1}} x_t + b_{z_{t+1}} + \epsilon_{t+1}$$

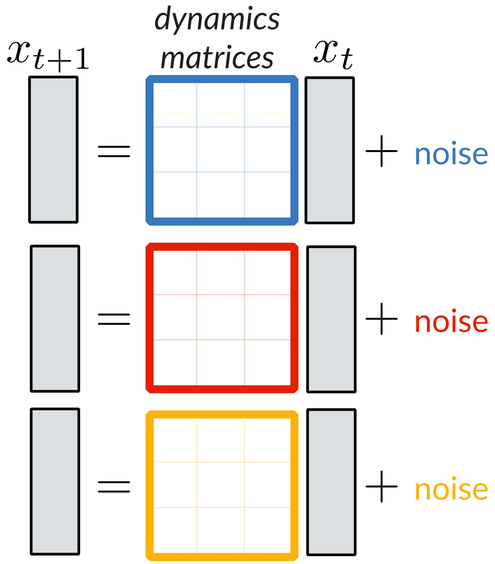
Specifying the form of the dependencies

State-dependent
switching probabilities



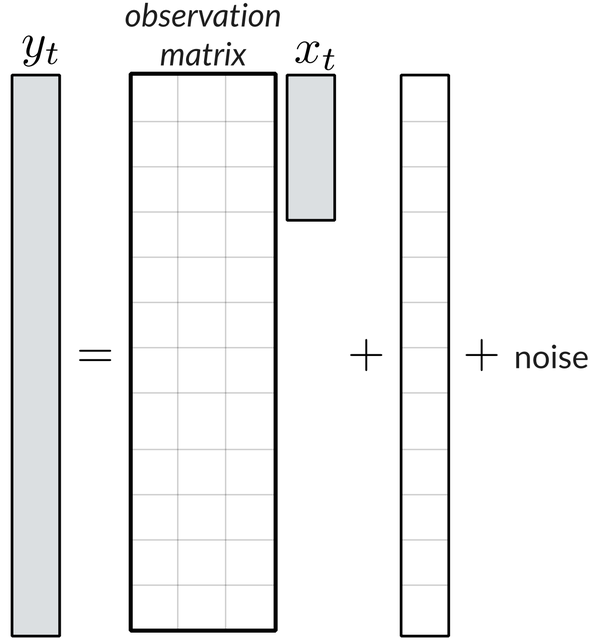
$$\Pr(z_{t+1} = j \mid z_t = i) = P_{ij}$$

Different **linear dynamics**
in each discrete state



$$x_{t+1} = A_{z_{t+1}} x_t + b_{z_{t+1}} + \epsilon_{t+1}$$

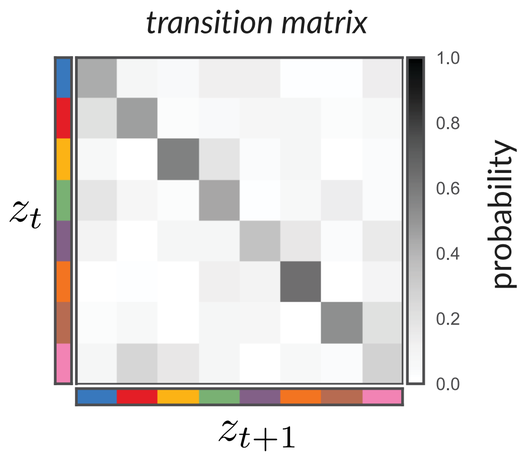
Linear mapping from continuous latent
states to observed neural activity



$$y_t = C x_t + d + \delta_t$$

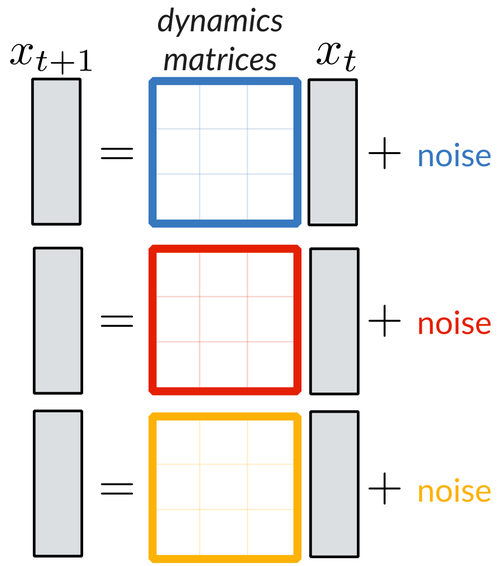
Specifying the form of the dependencies

State-dependent
switching probabilities



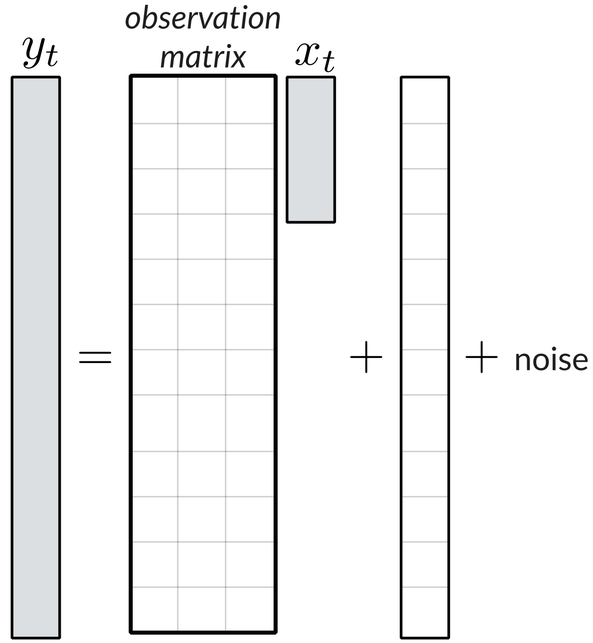
$$\Pr(z_{t+1} = j \mid z_t = i) = P_{ij}$$

Different **linear dynamics**
in each discrete state



$$x_{t+1} = A_{z_{t+1}} x_t + b_{z_{t+1}} + \epsilon_{t+1}$$

Linear mapping from continuous latent
states to observed neural activity

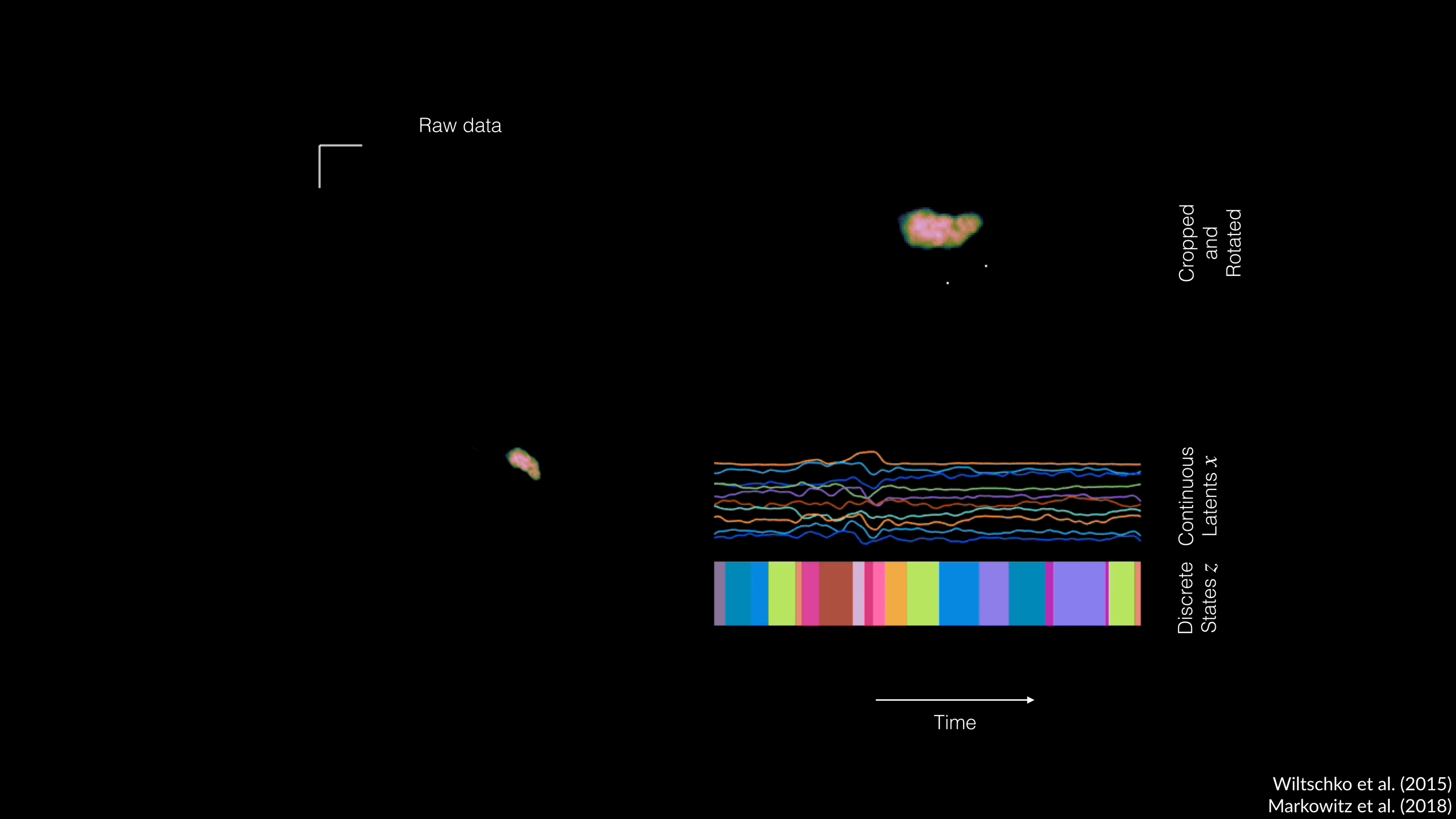


$$y_t = Cx_t + d + \delta_t$$

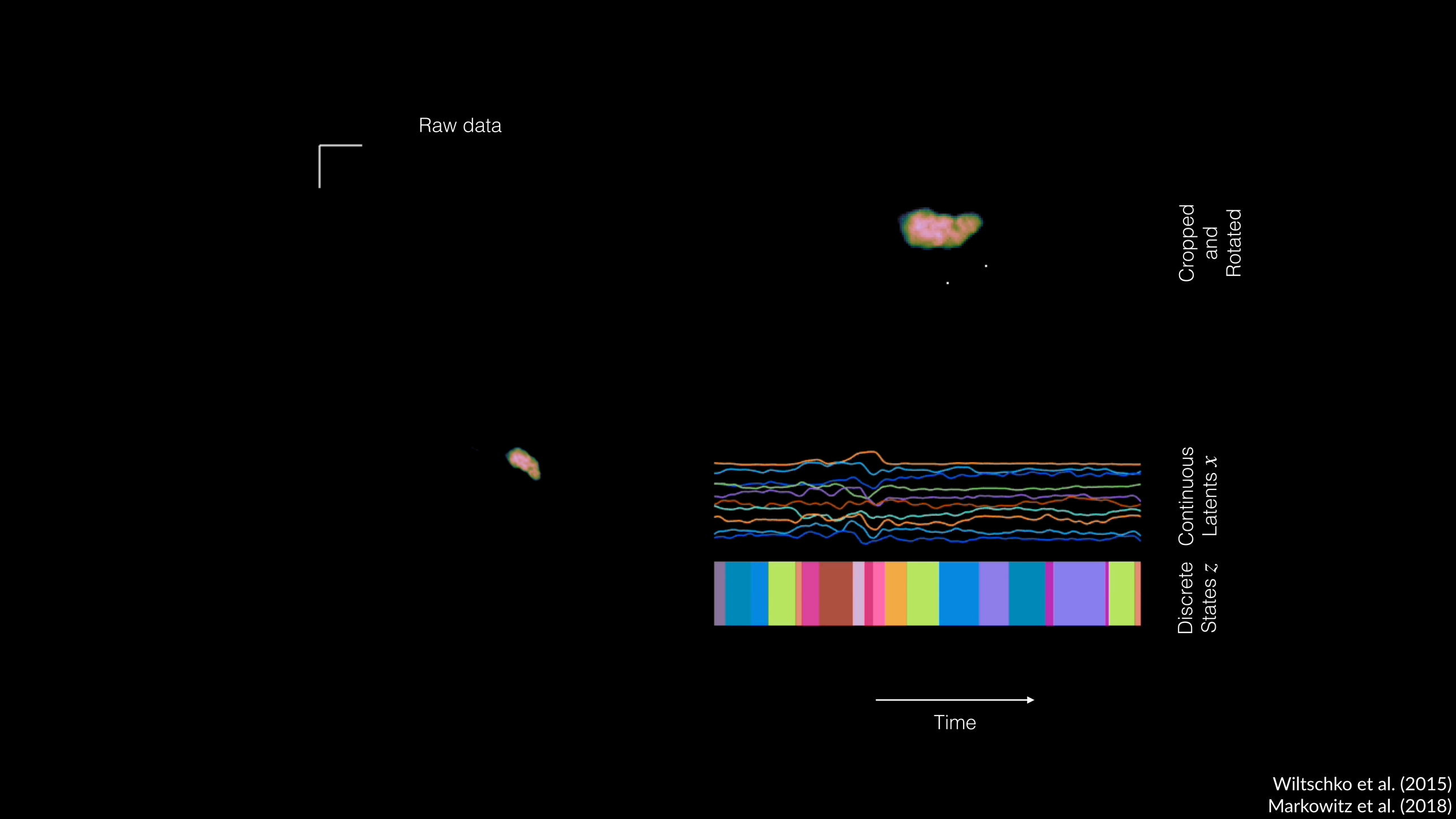
Switching Linear Dynamical System (SLDS)

Combines LDS (Kalman, 1960) and HMMs (Rabiner, 1989).

Ghahramani and Hinton (1996); Murphy (1998)



Wiltschko et al. (2015)
Markowitz et al. (2018)

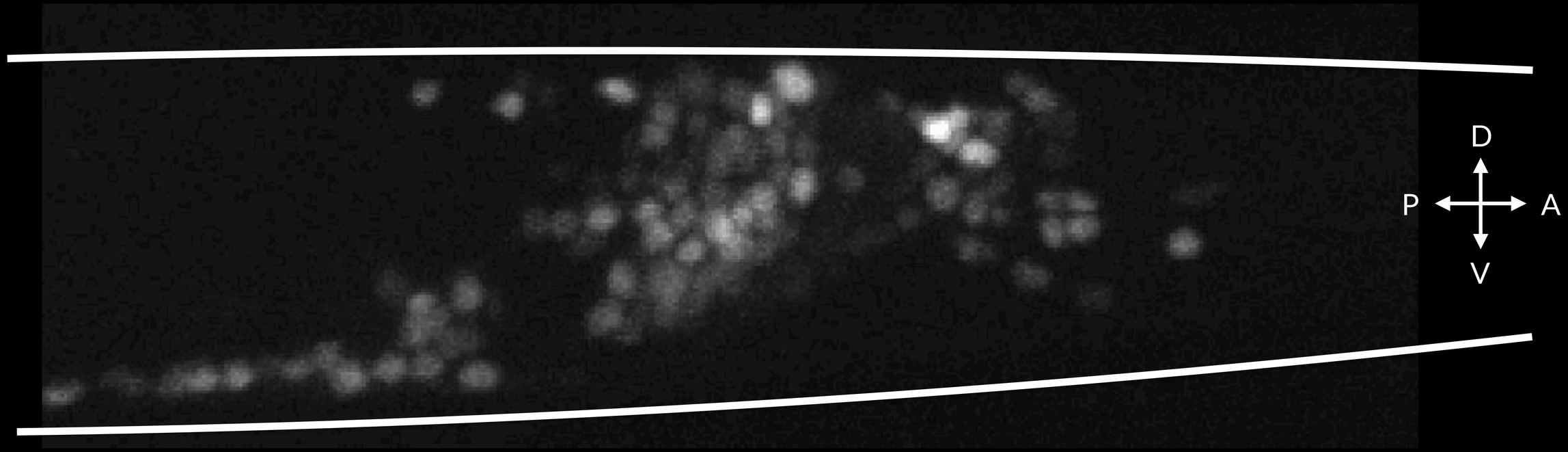


Wiltschko et al. (2015)
Markowitz et al. (2018)

A. E. X. Brown, Imperial College

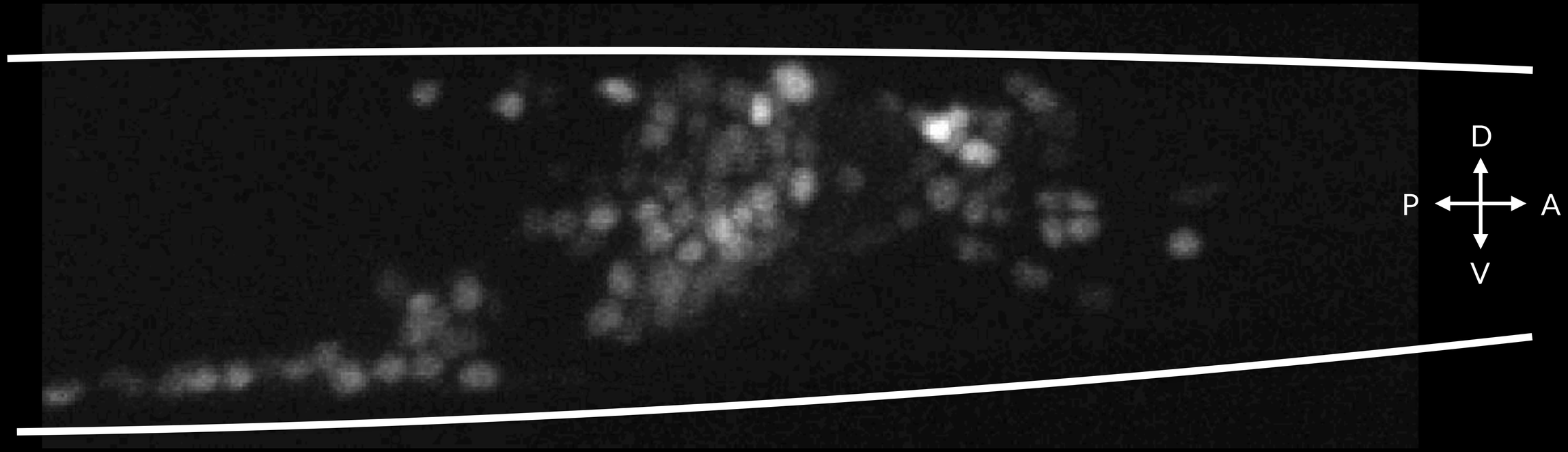
A. E. X. Brown, Imperial College

Calcium imaging of ~100 head ganglia neurons in immobilized C. elegans

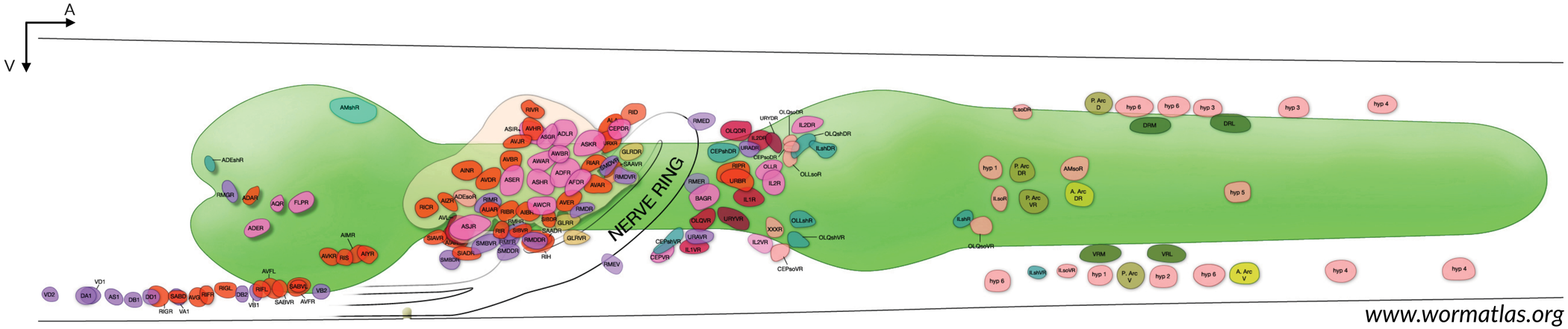


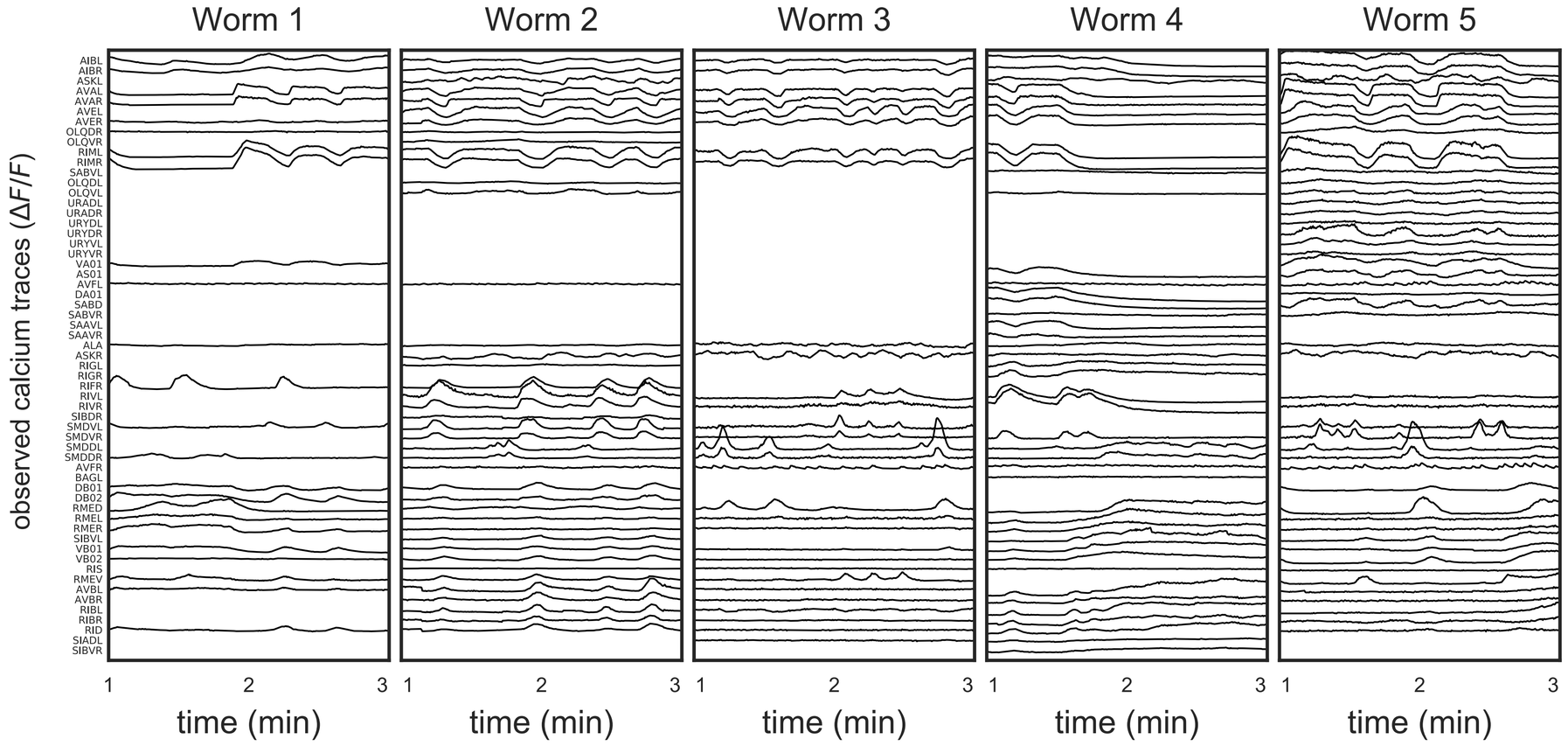
with **Nichols**, *Blei*, **Zimmer**, *Paninski*

Calcium imaging of ~100 head ganglia neurons in immobilized C. elegans

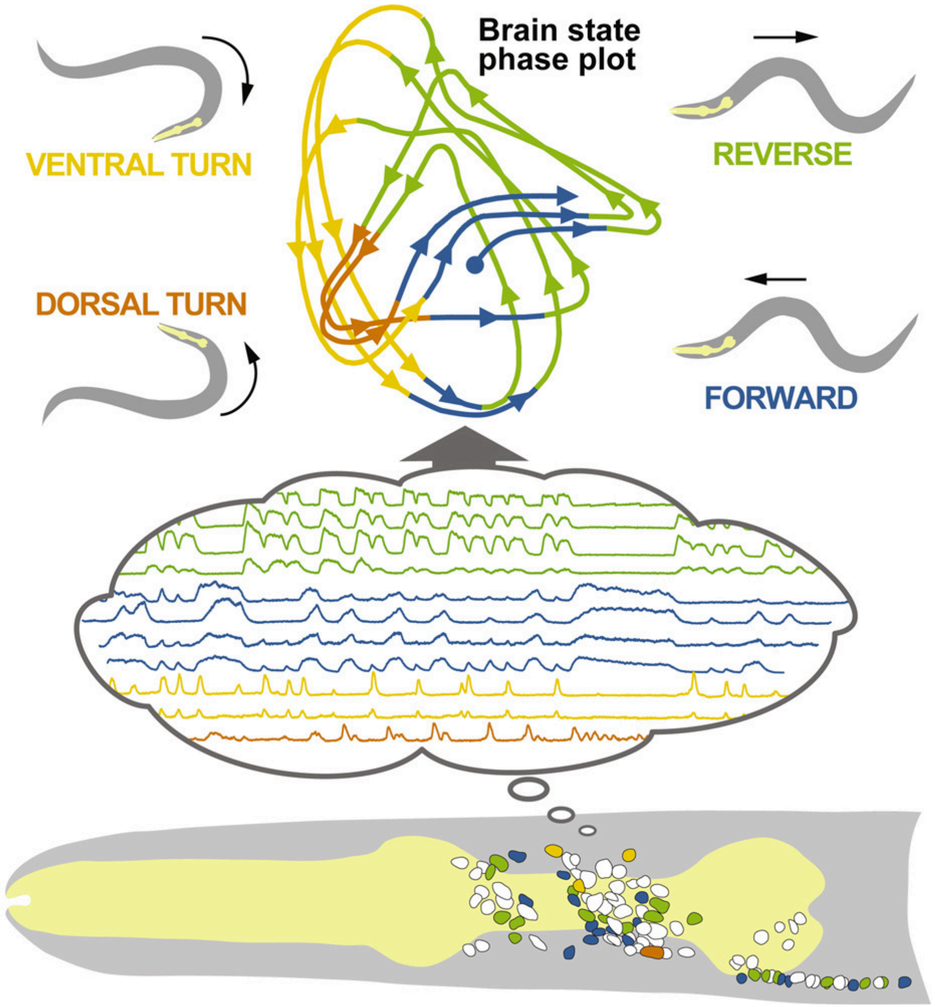


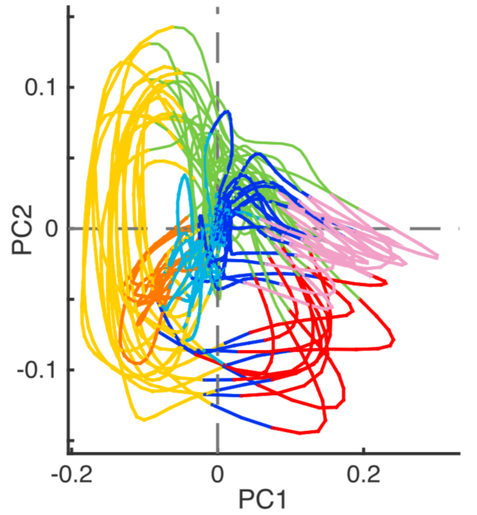
with **Nichols**, *Blei*, **Zimmer**, *Paninski*

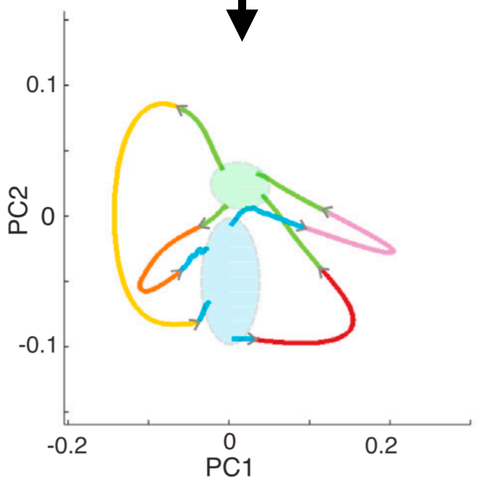

A
V
NERVE RING
www.wormatlas.org


Worm 1
Worm 2
Worm 3
Worm 4
Worm 5
observed calcium traces (ΔF/F)
AIBL
AIBR
ASKL
AVAL
AVAR
AVEL
AVER
OLQDR
OLQVR
RIML
RIMR
SABVL
OLQDL
OLQVL
URADL
URADR
URYDL
URYDR
URYVL
URYVR
VA01
AS01
AVFL
DA01
SABD
SABVR
SAAVL
SAAVR
ALA
ASKR
RIGL
RIGR
RIFR
RIVL
RIVR
SIBDR
SMDVL
SMDVR
SMDDL
SMDDR
AVFR
BAGL
DB01
DB02
RMED
RMEL
RMER
SIBVL
VB01
VB02
RIS
RMEV
AVBL
AVBR
RIBL
RIBR
RID
SIADL
SIBVR
1
2
3
time (min)

Previous work suggests that this neural activity lies on a low dimensional manifold partitioned by behavior

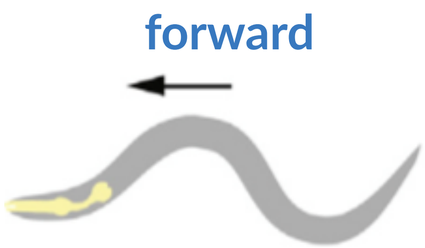


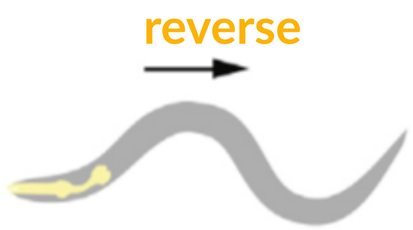


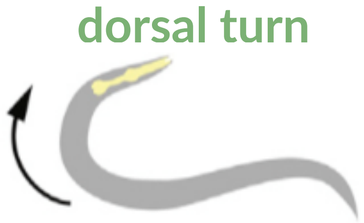


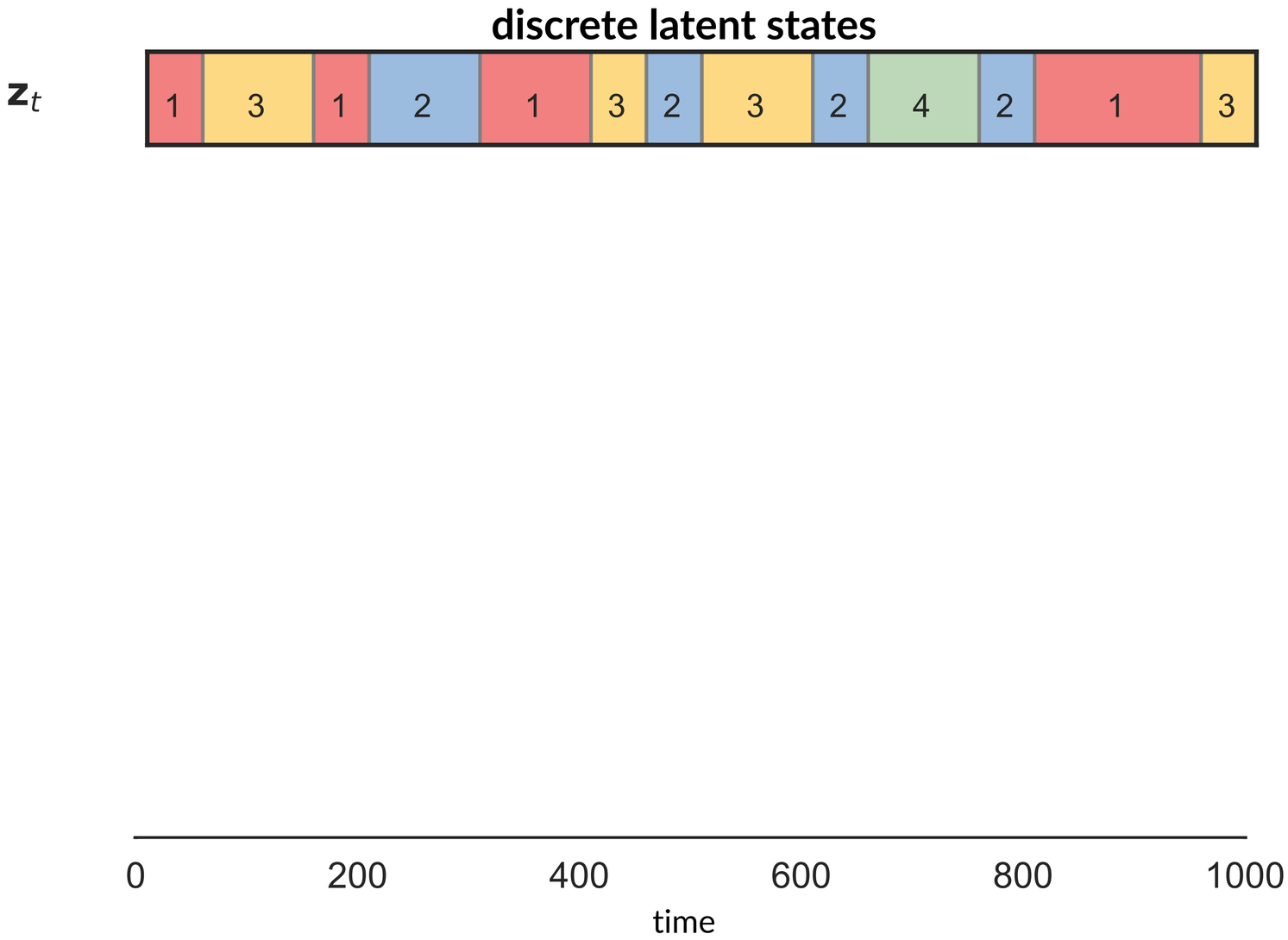
Kato et al (2015)

Building a probabilistic model of neural data









Building a probabilistic model of neural data

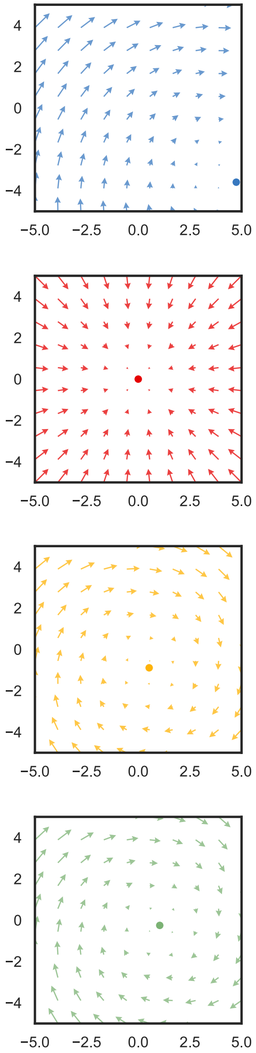
forward

ventral turn

reverse

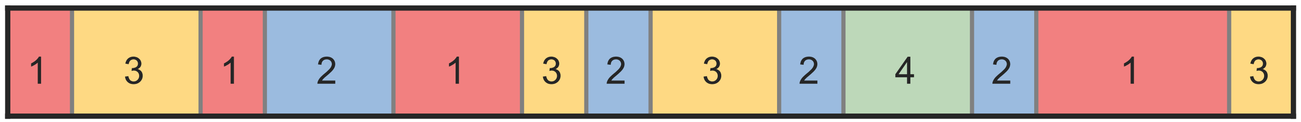
dorsal turn

dynamics

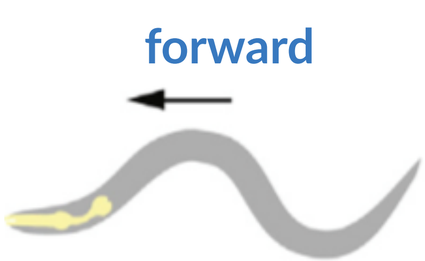


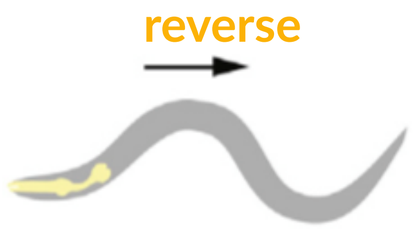
z_t

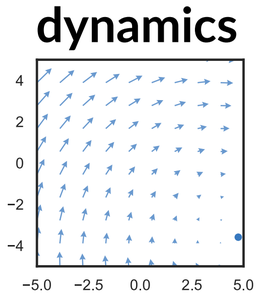
discrete latent states

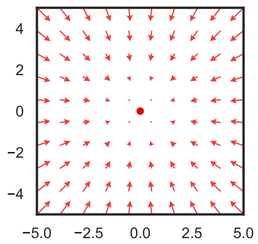


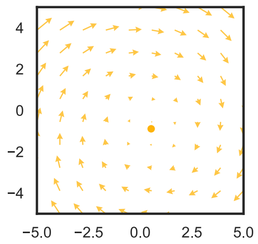
Building a probabilistic model of neural data

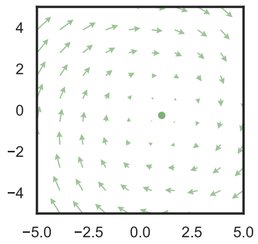






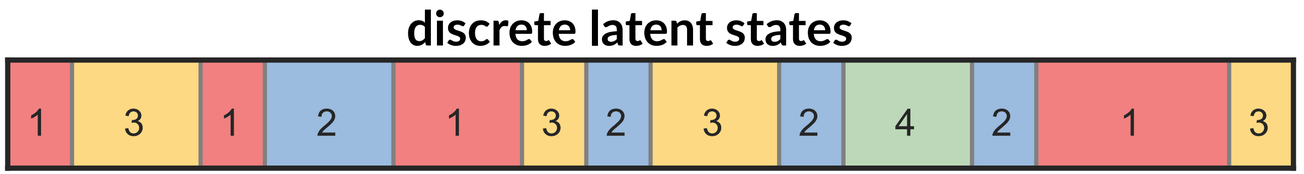


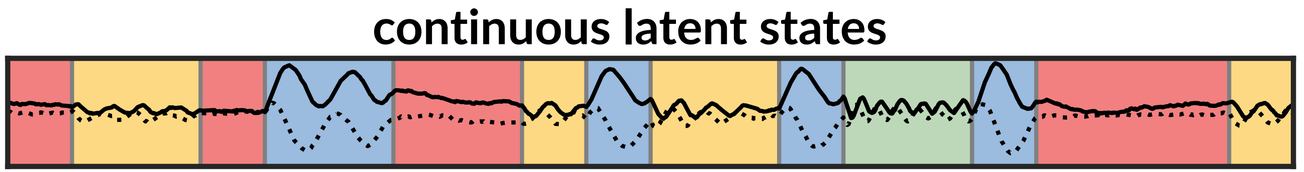




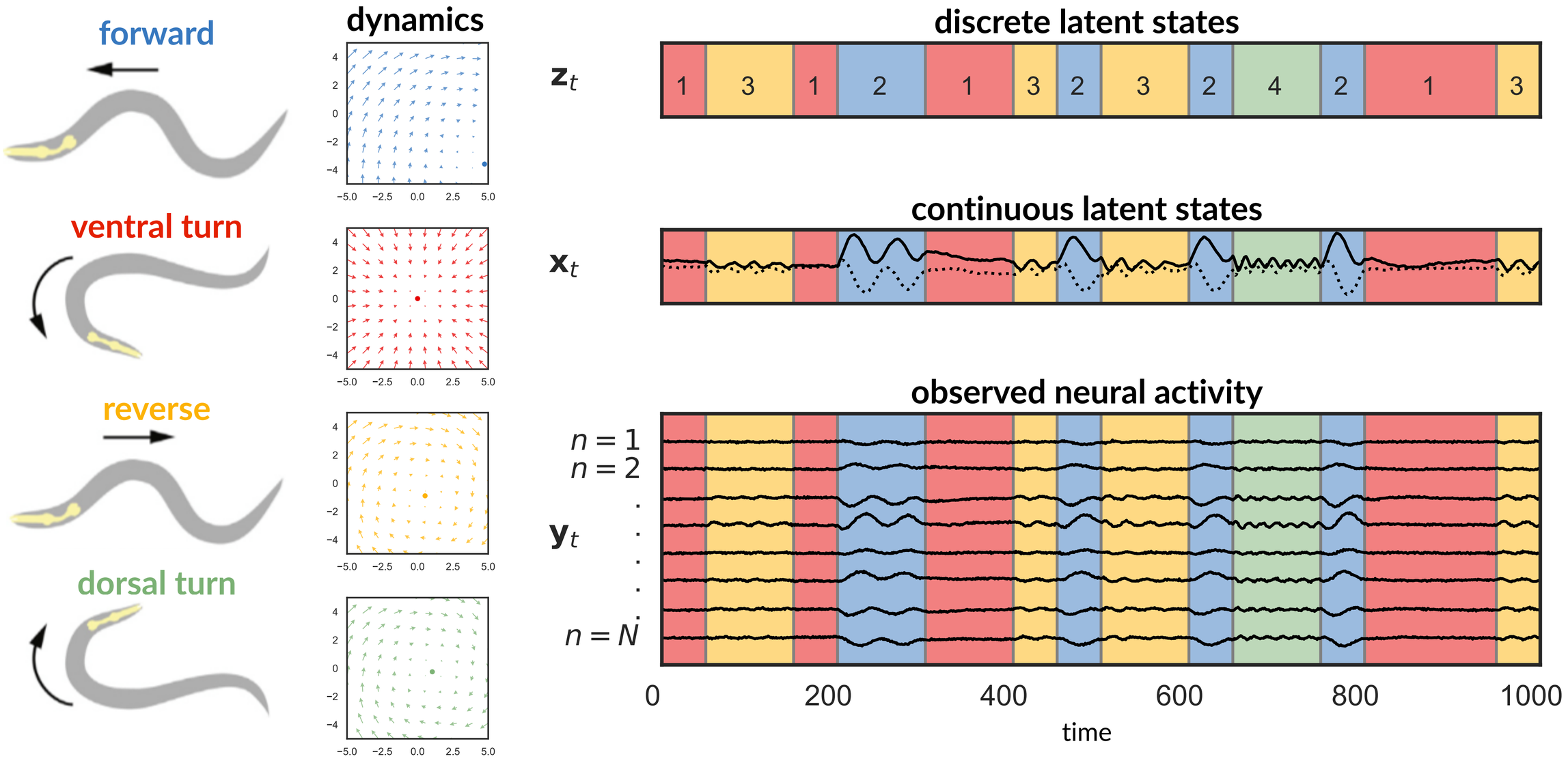
$\mathbf{z}_t$

$\mathbf{x}_t$





Building a probabilistic model of neural data



Hardness of exact EM for SLDS

Exact EM for the SLDS

- **E-step**: Update the posterior over latent variables,

$$q(z, x) \leftarrow p(z, x \mid y, \Theta) = \frac{p(z, x, y \mid \Theta)}{p(y \mid \Theta)}$$

- As before, we only need certain expectations under q,

$$\mathbb{E}_{q(z,x)}\left[\mathbb{I}[z_t = k]\right], \quad \mathbb{E}_{q(z,x)}\left[\mathbb{I}[z_t = k]x_t\right], \quad \mathbb{E}_{q(z,x)}\left[\mathbb{I}[z_t = k]x_t x_t^\top\right], \quad \mathbb{E}_{q(z,x)}\left[\mathbb{I}[z_t = k]x_t x_{t+1}^\top\right],$$

- **M-step**: Update the parameters,

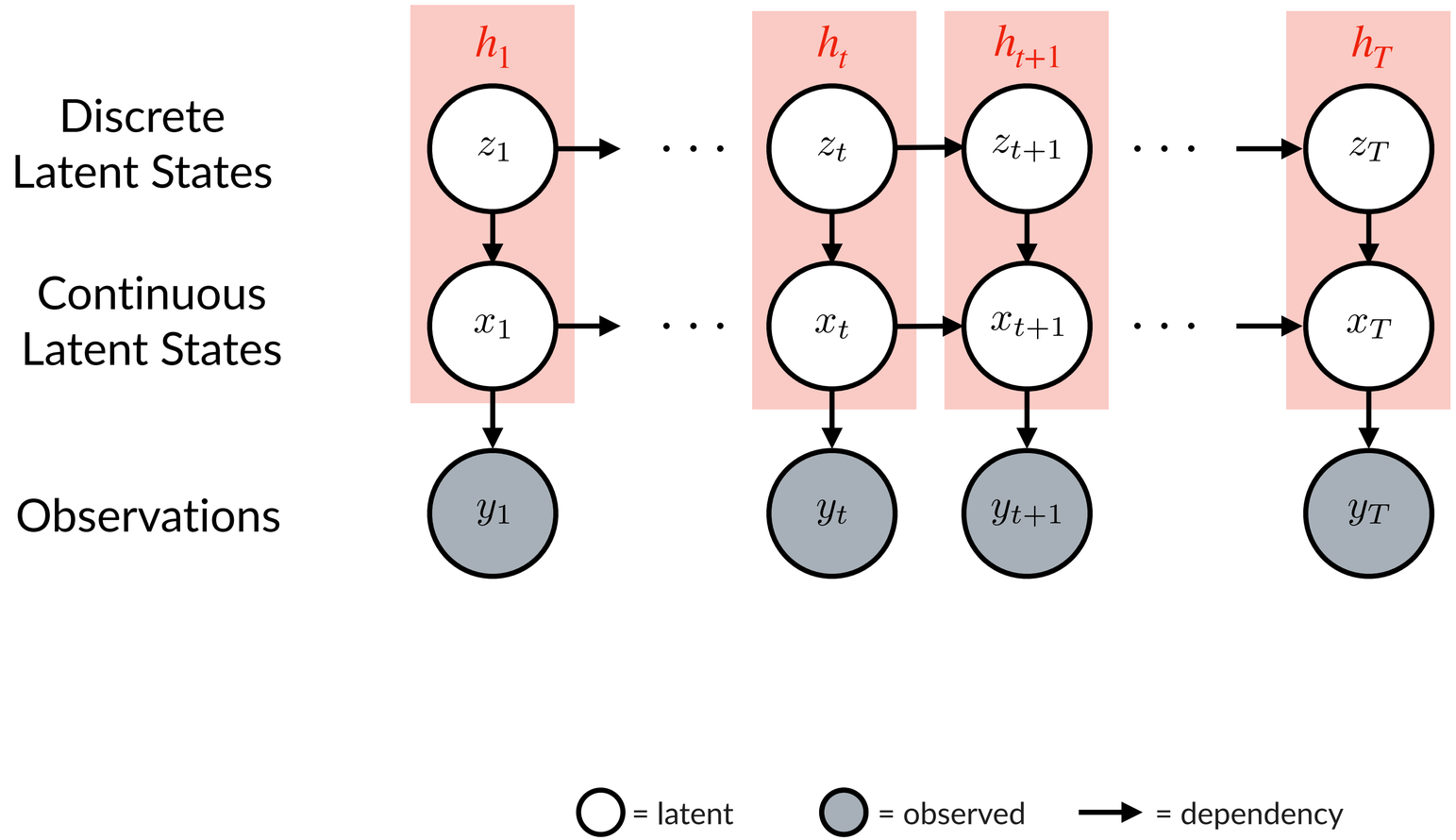
$$\Theta \leftarrow \arg\max \mathbb{E}_{q(z,x)}\left[\log p(z, x, y \mid \Theta)\right]$$

- Unfortunately, computing the necessary expectations is a lot harder now!

Combining the latent states

SLDS as a “hybrid” state space model

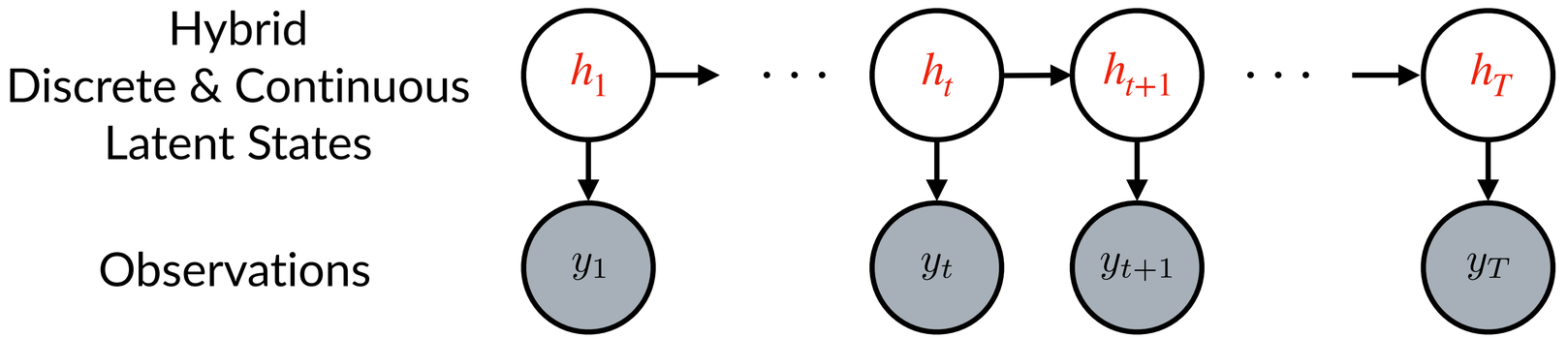
- Let $h_t = (z_t, x_t)$ denote the hybrid discrete & continuous latent state



Combining the latent states

SLDS as a “hybrid” state space model

- Let $h_t = (z_t, x_t)$ denote the hybrid discrete & continuous latent state



= latent = observed = dependency

Exact EM for SLDS

Computing the marginal distributions

- Consider the marginal probability of the latent states at time t:

$$q(h_t) = \int \mathrm{d}h_1 \cdots \int \mathrm{d}h_{t-1} \int \mathrm{d}h_{t+1} \cdots \int \mathrm{d}h_T \, q(h_1, \ldots, h_{t-1}, h_t, h_{t+1}, \ldots, h_T)$$

Exact EM for SLDS

Computing the marginal distributions

- Consider the marginal probability of the latent states at time t:

$$
\begin{aligned}
q(h_t) &= \int \mathrm{d}h_1 \cdots \int \mathrm{d}h_{t-1} \int \mathrm{d}h_{t+1} \cdots \int \mathrm{d}h_T \, q(h_1, \ldots, h_{t-1}, h_t, h_{t+1}, \ldots, h_T) \\
&\propto \left[\int \mathrm{d}h_1 \cdots \int \mathrm{d}h_{t-1} \, p(h_1) \prod_{s=1}^{t-1} p(h_s \mid h_s) \, p(h_{s+1} \mid h_s) \right] \times \left[p(y_t \mid h_t) \right] \\
&\qquad \times \left[\int \mathrm{d}h_{t+1} \cdots \int \mathrm{d}h_T \prod_{u=t+1}^{T} p(h_u \mid h_{u-1}) \, p(y_u \mid h_u) \right]
\end{aligned}
$$

Exact EM for SLDS

Computing the marginal distributions

- Consider the marginal probability of the latent states at time t:

$$
\begin{aligned}
q(h_t) &= \int \mathrm{d}h_1 \cdots \int \mathrm{d}h_{t-1} \int \mathrm{d}h_{t+1} \cdots \int \mathrm{d}h_T \, q(h_1, \ldots, h_{t-1}, h_t, h_{t+1}, \ldots, h_T) \\
&\propto \left[\int \mathrm{d}h_1 \cdots \int \mathrm{d}h_{t-1} \, p(h_1) \prod_{s=1}^{t-1} p(h_s \mid h_s) \, p(h_{s+1} \mid h_s) \right] \times \left[p(y_t \mid h_t) \right] \\
&\qquad \times \left[\int \mathrm{d}h_{t+1} \cdots \int \mathrm{d}h_T \prod_{u=t+1}^{T} p(h_u \mid h_{u-1}) \, p(y_u \mid h_u) \right] \\
&\triangleq \alpha_t(h_t) \times p(y_t \mid h_t) \times \beta_t(h_t)
\end{aligned}
$$

Exact EM for SLDS

Computing the forward messages $\alpha_t(h_t)$

- Consider the “forward messages”:

$$\alpha_t(h_t) \triangleq \int \mathrm{d}h_1 \cdots \int \mathrm{d}h_{t-1}\, p(h_1) \prod_{s=1}^{t-1} p(h_s \mid h_s)\, p(h_{s+1} \mid h_s)$$

Exact EM for SLDS

Computing the forward messages $\alpha_t(h_t)$

- Consider the "forward messages":

$$\alpha_t(h_t) \triangleq \int \mathrm{d}h_1 \cdots \int \mathrm{d}h_{t-1}\, p(h_1) \prod_{s=1}^{t-1} p(h_s \mid h_s)\, p(h_{s+1} \mid h_s)$$

$$= \int \mathrm{d}h_{t-1} \left[\left(\int \mathrm{d}h_1 \cdots \int \mathrm{d}h_{t-2}\, p(h_1) \prod_{s=1}^{t-2} p(y_s \mid h_s) p(h_{s+1} \mid h_s) \right) p(y_{t-1} \mid h_{t-1})\, p(h_t \mid h_{t-1}) \right]$$

Exact EM for SLDS

Computing the forward messages $\alpha_t(h_t)$

- Consider the "forward messages":

$$\alpha_t(h_t) \triangleq \int \mathrm{d}h_1 \cdots \int \mathrm{d}h_{t-1}\, p(h_1) \prod_{s=1}^{t-1} p(h_s \mid h_s)\, p(h_{s+1} \mid h_s)$$

$$= \int \mathrm{d}h_{t-1} \left[\left(\int \mathrm{d}h_1 \cdots \int \mathrm{d}h_{t-2}\, p(h_1) \prod_{s=1}^{t-2} p(y_s \mid h_s) p(h_{s+1} \mid h_s) \right) p(y_{t-1} \mid h_{t-1})\, p(h_t \mid h_{t-1}) \right]$$

$$= \int \mathrm{d}h_{t-1}\, \alpha_{t-1}(h_{t-1})\, p(y_{t-1} \mid h_{t-1})\, p(h_t \mid h_{t-1})$$

- We can compute these messages **recursively!**

Hardness of Exact EM for SLDS

Computing the forward messages $\alpha_t(h_t)$

- Now substitute $h_t = (z_t, x_t)$.
- Base case:

$$\begin{aligned}\alpha_1(z_1, x_1) &= p(z_1)\, p(x_1 \mid z_1) \\ &= \sum_{k=1}^{K} \left[\mathrm{Cat}(z_1 \mid \pi)\, \mathcal{N}(x_1 \mid b_k, Q_k)\right]^{\mathbb{I}[z_1=k]}\end{aligned}$$

Hardness of Exact EM for SLDS

Computing the forward messages $\alpha_t(h_t)$

- Now substitute $h_t = (z_t, x_t)$.
- Base case:

$$\begin{aligned}\alpha_1(z_1, x_1) &= p(z_1)\, p(x_1 \mid z_1) \\ &= \sum_{k=1}^{K} \left[\mathrm{Cat}(z_1 \mid \pi)\, \mathcal{N}(x_1 \mid b_k, Q_k)\right]^{\mathbb{I}[z_1=k]}\end{aligned}$$

- Second time step:

$$\begin{aligned}\alpha_2(z_2, x_2) &= \sum_{z_1=1}^{K} \int \mathrm{d}x_1\, \alpha_1(z_1, x_1)\, p(y_1 \mid x_1)\, p(z_2 \mid z_1)\, p(x_2 \mid x_1, z_2) \\ &= \sum_{z_1=1}^{K} \sum_{k=1}^{K} \left[\mathrm{Cat}\left(z_2 \mid \rho(z_1, y_1)\right) \mathcal{N}\left(x_2 \mid \mu(z_1, z_2, y_1), \Sigma(z_1, z_2, y_1)\right)\right]^{\mathbb{I}[z_2=k]}\end{aligned}$$

Hardness of Exact EM for SLDS

Computing the forward messages $\alpha_t(h_t)$

- By the t-th time step,

$$\alpha_t(z_t, x_t) = \sum_{z_1=1}^{K} \cdots \sum_{z_{t-1}=1}^{K} \sum_{k=1}^{K} \left[\mathrm{Cat}\left(z_t \mid \rho(z_{1:t-1}, y_{1:t-1})\right) \mathcal{N}\left(x_t \mid \mu(z_t, z_{1:t-1}, y_{1:t-1}), \Sigma(z_t, z_{1:t-1}, y_{1:t-1})\right) \right]^{\mathbb{I}[z_t=k]}$$

- This is still a mixture of Gaussians.
- **Question:** How many components does it have?

Approximate Message Passing

Approximating the forward messages

Assumed density filtering (ADF)

Approximating the forward messages
Assumed density filtering (ADF)

- **Idea:** Approximate the forward messages with a tractable family of distributions $\mathscr{A}$.
  - For example, assume $\alpha_t(h_t) \approx \tilde{\alpha}_t(h_t) \in \mathscr{A}$ where $\mathscr{A}$ is the set of Gaussian mixtures with at most M components.

Approximating the forward messages
Assumed density filtering (ADF)

- **Idea:** Approximate the forward messages with a tractable family of distributions $\mathscr{A}$.
  - For example, assume $\alpha_t(h_t) \approx \tilde{\alpha}_t(h_t) \in \mathscr{A}$ where $\mathscr{A}$ is the set of Gaussian mixtures with at most M components.
- Suppose we have $\tilde{\alpha}_{t-1}(h_{t-1}) \in \mathscr{A}$. Our **target for** $\alpha_t(h_t)$ is,

$$\hat{\alpha}_t(h_t) \triangleq \int \mathrm{d}h_{t-1}\, \tilde{\alpha}_{t-1}(h_{t-1})\, p(y_{t-1} \mid h_{t-1})\, p(h_t \mid h_{t-1})$$

This may not be in $\mathscr{A}$!

Approximating the forward messages
Assumed density filtering (ADF)

- **Idea:** Approximate the forward messages with a tractable family of distributions $\mathscr{A}$.
  - For example, assume $\alpha_t(h_t) \approx \tilde{\alpha}_t(h_t) \in \mathscr{A}$ where $\mathscr{A}$ is the set of Gaussian mixtures with at most M components.
- Suppose we have $\tilde{\alpha}_{t-1}(h_{t-1}) \in \mathscr{A}$. Our **target for** $\alpha_t(h_t)$ is,

$$\hat{\alpha}_t(h_t) \triangleq \int \mathrm{d}h_{t-1}\, \tilde{\alpha}_{t-1}(h_{t-1})\, p(y_{t-1} \mid h_{t-1})\, p(h_t \mid h_{t-1})$$

  This may not be in $\mathscr{A}$!

- Find the member of $\mathscr{A}$ that best approximates $\hat{\alpha}_t(h_t)$:

$$\tilde{\alpha}_t(h_t) \leftarrow \arg\min_{\mathscr{A}} D\Big(\tilde{\alpha}_t(h_t) \,\|\, \hat{\alpha}_t(h_t)\Big)$$

  where $D(\cdot \,\|\, \cdot)$ is a measure of divergence between two densities; e.g. reverse Kullback-Leibler (KL) divergence.

Approximating the forward messages
Assumed density filtering (ADF)

- **Idea:** Approximate the forward messages with a tractable family of distributions $\mathscr{A}$.
  - For example, assume $\alpha_t(h_t) \approx \tilde{\alpha}_t(h_t) \in \mathscr{A}$ where $\mathscr{A}$ is the set of Gaussian mixtures with at most M components.
- Suppose we have $\tilde{\alpha}_{t-1}(h_{t-1}) \in \mathscr{A}$. Our **target for** $\alpha_t(h_t)$ is,

$$\hat{\alpha}_t(h_t) \triangleq \int \mathrm{d}h_{t-1}\, \tilde{\alpha}_{t-1}(h_{t-1})\, p(y_{t-1} \mid h_{t-1})\, p(h_t \mid h_{t-1})$$

  This may not be in $\mathscr{A}$!

- Find the member of $\mathscr{A}$ that best approximates $\hat{\alpha}_t(h_t)$:

$$\tilde{\alpha}_t(h_t) \leftarrow \arg\min_{\mathscr{A}} D\Big(\tilde{\alpha}_t(h_t) \,\|\, \hat{\alpha}_t(h_t)\Big)$$

  where $D(\cdot \,\|\, \cdot)$ is a measure of divergence between two densities; e.g. reverse Kullback-Leibler (KL) divergence.

- This is called **assumed density filtering (ADF)**, and it is closely related to **expectation-propagation (EP)** and the **unscented/ extended Kalman filter.**

Approximating the forward messages

Assumed density filtering (ADF) with GMM messages

- For example, let $\mathscr{A}$ be the set of Gaussian mixtures with parameters ρ_t, $\{\mu_{t,k}, \Sigma_{t,k}\}_{k=1}^K$:

$$\tilde{\alpha}_t(z_t, x_t) = \tilde{\alpha}_t(z_t)\, \tilde{\alpha}_t(x_t \mid z_t) = \mathrm{Cat}(z_t \mid \rho_t)\, \mathcal{N}(x_t \mid \mu_{t,z_t}, \Sigma_{t,z_t}).$$

Approximating the forward messages

Assumed density filtering (ADF) with GMM messages

- For example, let $\mathscr{A}$ be the set of Gaussian mixtures with parameters ρ_t, $\{\mu_{t,k}, \Sigma_{t,k}\}_{k=1}^K$:

$$\tilde{\alpha}_t(z_t, x_t) = \tilde{\alpha}_t(z_t)\, \tilde{\alpha}_t(x_t \mid z_t) = \mathrm{Cat}(z_t \mid \rho_t)\, \mathcal{N}(x_t \mid \mu_{t,z_t}, \Sigma_{t,z_t}).$$

- The target is

$$\hat{\alpha}_t(z_t, x_t) \triangleq \sum_{z_{t-1}=1}^{K} \int \mathrm{d}x_{t-1}\, \tilde{\alpha}_{t-1}(z_{t-1}, x_{t-1})\, p(y_{t-1} \mid x_{t-1})\, p(z_t \mid z_{t-1})\, p(x_t \mid x_{t-1}, z_{t-1})$$

Approximating the forward messages

Assumed density filtering (ADF) with GMM messages

- For example, let $\mathscr{A}$ be the set of Gaussian mixtures with parameters ρ_t, $\{\mu_{t,k}, \Sigma_{t,k}\}_{k=1}^K$:

$$\tilde{\alpha}_t(z_t, x_t) = \tilde{\alpha}_t(z_t)\, \tilde{\alpha}_t(x_t \mid z_t) = \mathrm{Cat}(z_t \mid \rho_t)\, \mathcal{N}(x_t \mid \mu_{t,z_t}, \Sigma_{t,z_t}).$$

- The target is

$$\begin{aligned}\hat{\alpha}_t(z_t, x_t) &\triangleq \sum_{z_{t-1}=1}^{K} \int \mathrm{d}x_{t-1}\, \tilde{\alpha}_{t-1}(z_{t-1}, x_{t-1})\, p(y_{t-1} \mid x_{t-1})\, p(z_t \mid z_{t-1})\, p(x_t \mid x_{t-1}, z_{t-1}) \\ &= \sum_{z_{t-1}=1}^{K} \int \mathrm{d}x_{t-1}\, \rho_{t-1,z_{t-1}} \mathcal{N}(x_{t-1} \mid \mu_{t-1,z_{t-1}}, \Sigma_{t-1,z_{t-1}})\, \mathcal{N}(y_{t-1} \mid Cx_{t-1} + d, R)\, P_{z_{t-1},z_t}\, \mathcal{N}(x_t \mid A_{z_t} x_{t-1} + b_{z_t}, Q_{z_t})\end{aligned}$$

Approximating the forward messages

Assumed density filtering (ADF) with GMM messages

- For example, let $\mathscr{A}$ be the set of Gaussian mixtures with parameters ρ_t, $\{\mu_{t,k}, \Sigma_{t,k}\}_{k=1}^K$:

$$\tilde{\alpha}_t(z_t, x_t) = \tilde{\alpha}_t(z_t)\, \tilde{\alpha}_t(x_t \mid z_t) = \mathrm{Cat}(z_t \mid \rho_t)\, \mathcal{N}(x_t \mid \mu_{t,z_t}, \Sigma_{t,z_t}).$$

- The target is

$$\begin{aligned}
\hat{\alpha}_t(z_t, x_t) &\triangleq \sum_{z_{t-1}=1}^{K} \int \mathrm{d}x_{t-1}\, \tilde{\alpha}_{t-1}(z_{t-1}, x_{t-1})\, p(y_{t-1} \mid x_{t-1})\, p(z_t \mid z_{t-1})\, p(x_t \mid x_{t-1}, z_{t-1}) \\
&= \sum_{z_{t-1}=1}^{K} \int \mathrm{d}x_{t-1}\, \rho_{t-1,z_{t-1}} \mathcal{N}(x_{t-1} \mid \mu_{t-1,z_{t-1}}, \Sigma_{t-1,z_{t-1}})\, \mathcal{N}(y_{t-1} \mid Cx_{t-1} + d, R)\, P_{z_{t-1},z_t}\, \mathcal{N}(x_t \mid A_{z_t} x_{t-1} + b_{z_t}, Q_{z_t}) \\
&\propto \sum_{z_{t-1}=1}^{K} \rho(z_t, z_{t-1}) \mathcal{N}\left(x_t \mid \mu(z_t, z_{t-1}), \Sigma(z_t, z_{t-1})\right)
\end{aligned}$$

- This is a GMM with K components for each assignment of z_t. We want to approximate it a single Gaussian for each z_t.

Approximating the forward messages

Projecting onto the set $\mathscr{A}$

- Find the member of $\mathscr{A}$ that best approximates $\hat{\alpha}_t(z_t, h_t)$ by minimizing the reverse Kullback-Leibler divergence,

$$\mathrm{KL}\Big(\hat{\alpha}_t(z_t, x_t) \,\|\, \tilde{\alpha}_t(z_t, x_t)\Big) = \mathbb{E}_{\hat{\alpha}_t(z_t, x_t)}\left[\log \hat{\alpha}_t(z_t, x_t) - \log \tilde{\alpha}_t(z_t, x_t)\right]$$

Approximating the forward messages

Projecting onto the set $\mathscr{A}$

- Find the member of $\mathscr{A}$ that best approximates $\hat{\alpha}_t(z_t, h_t)$ by minimizing the reverse Kullback-Leibler divergence,

$$\begin{aligned}\mathrm{KL}\Big(\hat{\alpha}_t(z_t, x_t) \,\|\, \tilde{\alpha}_t(z_t, x_t)\Big) &= \mathbb{E}_{\hat{\alpha}_t(z_t, x_t)}\Big[\log \hat{\alpha}_t(z_t, x_t) - \log \tilde{\alpha}_t(z_t, x_t)\Big] \\ &= -\,\mathbb{E}_{\hat{\alpha}_t(z_t, x_t)}\Big[\log \tilde{\alpha}_t(z_t, x_t)\Big] + \mathrm{c}\end{aligned}$$

Approximating the forward messages

Projecting onto the set $\mathscr{A}$

- Find the member of $\mathscr{A}$ that best approximates $\hat{\alpha}_t(z_t, h_t)$ by minimizing the reverse Kullback-Leibler divergence,

$$\begin{aligned}
\mathrm{KL}\Big(\hat{\alpha}_t(z_t, x_t) \,\|\, \tilde{\alpha}_t(z_t, x_t)\Big) &= \mathbb{E}_{\hat{\alpha}_t(z_t, x_t)}\Big[\log \hat{\alpha}_t(z_t, x_t) - \log \tilde{\alpha}_t(z_t, x_t)\Big] \\
&= -\,\mathbb{E}_{\hat{\alpha}_t(z_t, x_t)}\Big[\log \tilde{\alpha}_t(z_t, x_t)\Big] + \mathrm{c} \\
&= -\,\mathbb{E}_{\hat{\alpha}_t(z_t, x_t)}\Big[\log \mathrm{Cat}(z_t \mid \rho_t) + \log \mathcal{N}(x_t \mid \mu_{t,z_t}, \Sigma_{t,z_t})\Big] + \mathrm{c}
\end{aligned}$$

Approximating the forward messages

Projecting onto the set $\mathscr{A}$

- Find the member of $\mathscr{A}$ that best approximates $\hat{\alpha}_t(z_t, h_t)$ by minimizing the reverse Kullback-Leibler divergence,

$$
\begin{aligned}
\mathrm{KL}\Big(\hat{\alpha}_t(z_t, x_t) \,\|\, \tilde{\alpha}_t(z_t, x_t)\Big) &= \mathbb{E}_{\hat{\alpha}_t(z_t, x_t)}\left[\log \hat{\alpha}_t(z_t, x_t) - \log \tilde{\alpha}_t(z_t, x_t)\right] \\
&= -\,\mathbb{E}_{\hat{\alpha}_t(z_t, x_t)}\left[\log \tilde{\alpha}_t(z_t, x_t)\right] + \mathrm{c} \\
&= -\,\mathbb{E}_{\hat{\alpha}_t(z_t, x_t)}\left[\log \mathrm{Cat}(z_t \mid \rho_t) + \log \mathcal{N}(x_t \mid \mu_{t,z_t}, \Sigma_{t,z_t})\right] + \mathrm{c} \\
&= -\,\mathbb{E}_{\hat{\alpha}_t(z_t, x_t)}\left[\sum_{k=1}^{K} \mathbb{I}[z_t = k]\left(\rho_{tk} - \frac{1}{2}\log|\Sigma_{tk}| - \frac{1}{2}x_t^\top \Sigma_{tk}^{-1} x_t + \mu_{tk}^\top \Sigma_{tk}^{-1} x_t - \frac{1}{2}\mu_{tk}^\top \Sigma_{tk}^{-1} \mu_{tk}\right)\right] + \mathrm{c}
\end{aligned}
$$

Approximating the forward messages

Projecting onto the set $\mathscr{A}$

- Find the member of $\mathscr{A}$ that best approximates $\hat{\alpha}_t(z_t, h_t)$ by minimizing the reverse Kullback-Leibler divergence,

$$
\begin{aligned}
\mathrm{KL}\Big(\hat{\alpha}_t(z_t, x_t) \,\|\, \tilde{\alpha}_t(z_t, x_t)\Big) &= \mathbb{E}_{\hat{\alpha}_t(z_t, x_t)}\left[\log \hat{\alpha}_t(z_t, x_t) - \log \tilde{\alpha}_t(z_t, x_t)\right] \\
&= -\,\mathbb{E}_{\hat{\alpha}_t(z_t, x_t)}\left[\log \tilde{\alpha}_t(z_t, x_t)\right] + \mathrm{c} \\
&= -\,\mathbb{E}_{\hat{\alpha}_t(z_t, x_t)}\left[\log \mathrm{Cat}(z_t \mid \rho_t) + \log \mathcal{N}(x_t \mid \mu_{t,z_t}, \Sigma_{t,z_t})\right] + \mathrm{c} \\
&= -\,\mathbb{E}_{\hat{\alpha}_t(z_t, x_t)}\left[\sum_{k=1}^{K} \mathbb{I}[z_t = k]\left(\rho_{tk} - \frac{1}{2}\log|\Sigma_{tk}| - \frac{1}{2}x_t^\top \Sigma_{tk}^{-1} x_t + \mu_{tk}^\top \Sigma_{tk}^{-1} x_t - \frac{1}{2}\mu_{tk}^\top \Sigma_{tk}^{-1}\mu_{tk}\right)\right] + \mathrm{c} \\
&= -\sum_{k=1}^{K} \langle \rho_{tk} - \frac{1}{2}\log|\Sigma_{tk}|, \bar{N}_k\rangle + \langle -\frac{1}{2}\Sigma_{tk}^{-1}, \bar{\psi}_{k,1}\rangle + \langle \Sigma_{tk}^{-1}\mu_{tk}, \bar{\psi}_{k,2}\rangle + \langle -\frac{1}{2}\mu_{tk}^\top \Sigma_{tk}^{-1}\mu_{tk}, \bar{\psi}_{k,3}\rangle + \mathrm{c}
\end{aligned}
$$

Where $\bar{N}_k = \mathbb{E}_{\hat{\alpha}_t(z_t, x_t)}[\mathbb{I}[z_t = k]],\ \bar{\psi}_{k,1} = \mathbb{E}_{\hat{\alpha}_t(z_t, x_t)}[\mathbb{I}[z_t = k]\, x_t x_t^\top],\ \bar{\psi}_{k,2} = \mathbb{E}_{\hat{\alpha}_t(z_t, x_t)}[\mathbb{I}[z_t = k]\, x_t],\ \bar{\psi}_{k,3} = \bar{N}_k.$

Conclusion

- Switching LDS combine ARHMMs and LDS to get the best of both worlds.
- They approximate nonlinear dynamical systems by switching between linear dynamical states.
- However, posterior inference is much harder! The posterior has K^T modes.
- Approximate message passing techniques leverage the Markovianity of the system with recursive updates, but approximate the messages with a simpler class.
- Next time, we'll look at alternative methods of approximate inference for SLDS.

Further Reading

- Barber, David. 2012. Bayesian Reasoning and Machine Learning. Cambridge University Press. **Chapter 25.**

- Linderman, Scott W., Matthew J. Johnson, Andrew C. Miller, Ryan P. Adams, David M. Blei, and Liam Paninski. 2017. “Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems.” In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS).