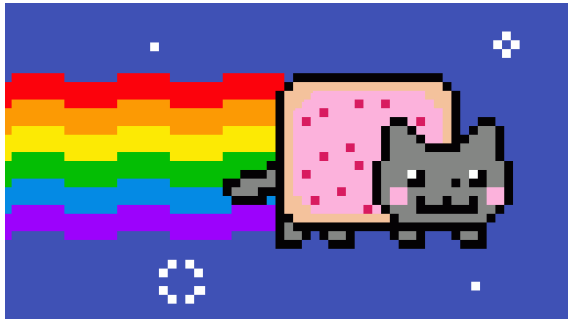
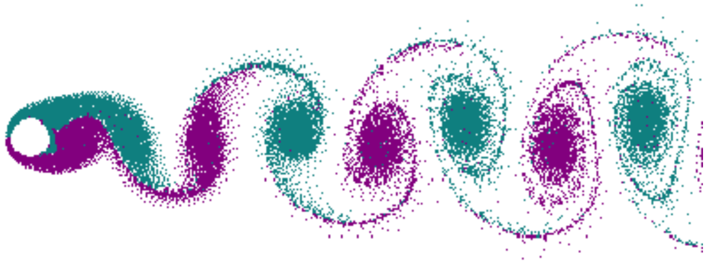
Sequential Generative Models: Some Basics & Advances

Yingzhen Li

yingzhen.li@imperial.ac.uk

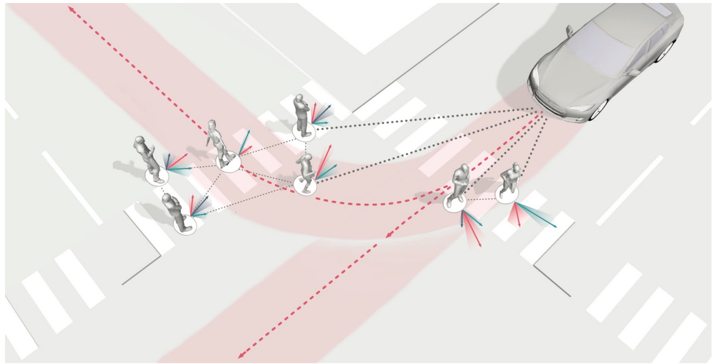
Sequence Data is Everywhere

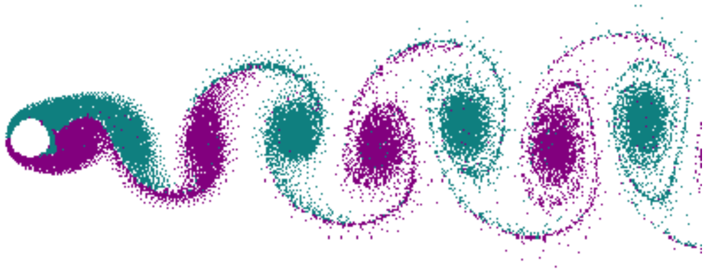


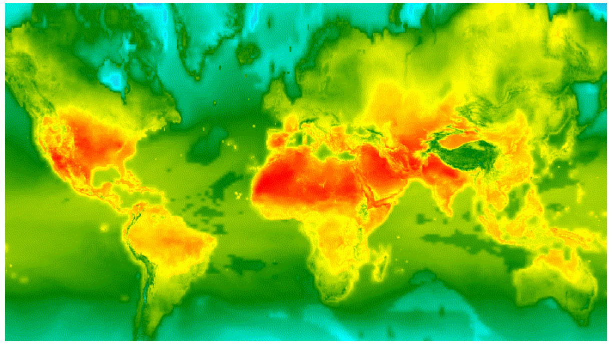


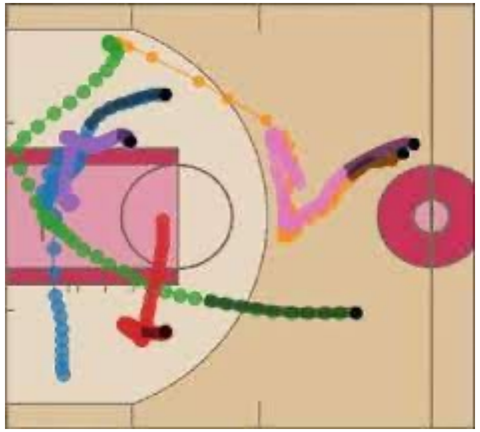
Learning Underlying Dynamics from Data with Sequential Generative Models

Representation learning: understanding the underlying dynamics









Types of Sequential Generative Models

	Discrete-Time	Continuous-Time
Deterministic Dynamics	Seq. LVM w/ deterministic states	(Latent) Neural ODE (Latent) Neural PDE Seq. LVM w/ continuous-time RNN
Stochastic Dynamics	State-Space Models	(Latent) Neural S(P)DE Gaussian Process VAEs

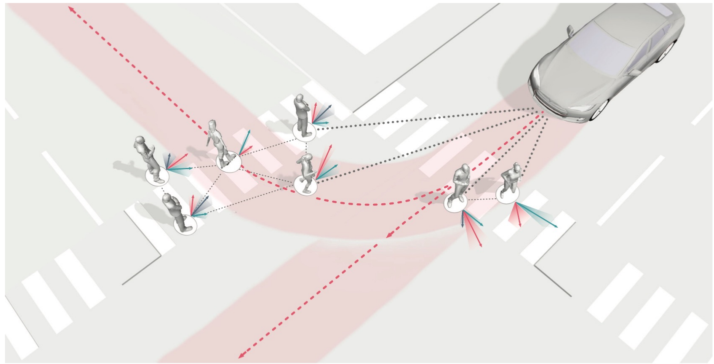
Temporal point processes

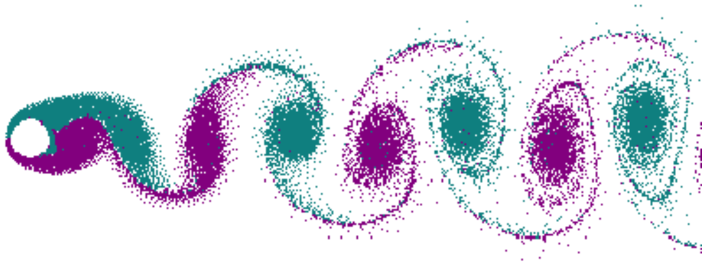
Auto-regressive models

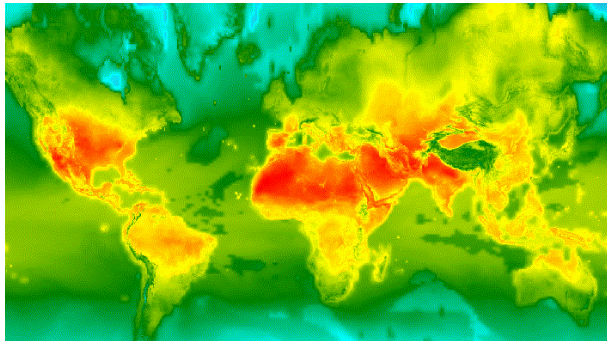
…

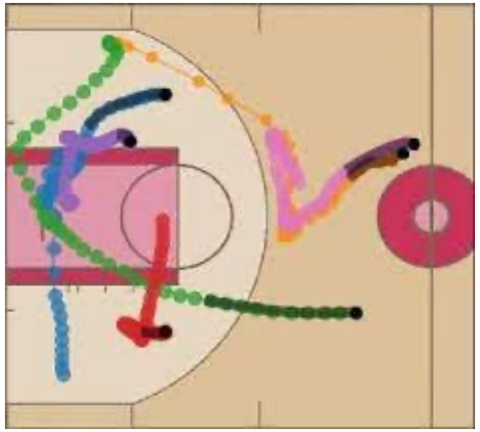
Choose Your Sword

Discrete-time or continuous-time? Deterministic or stochastic dynamics?



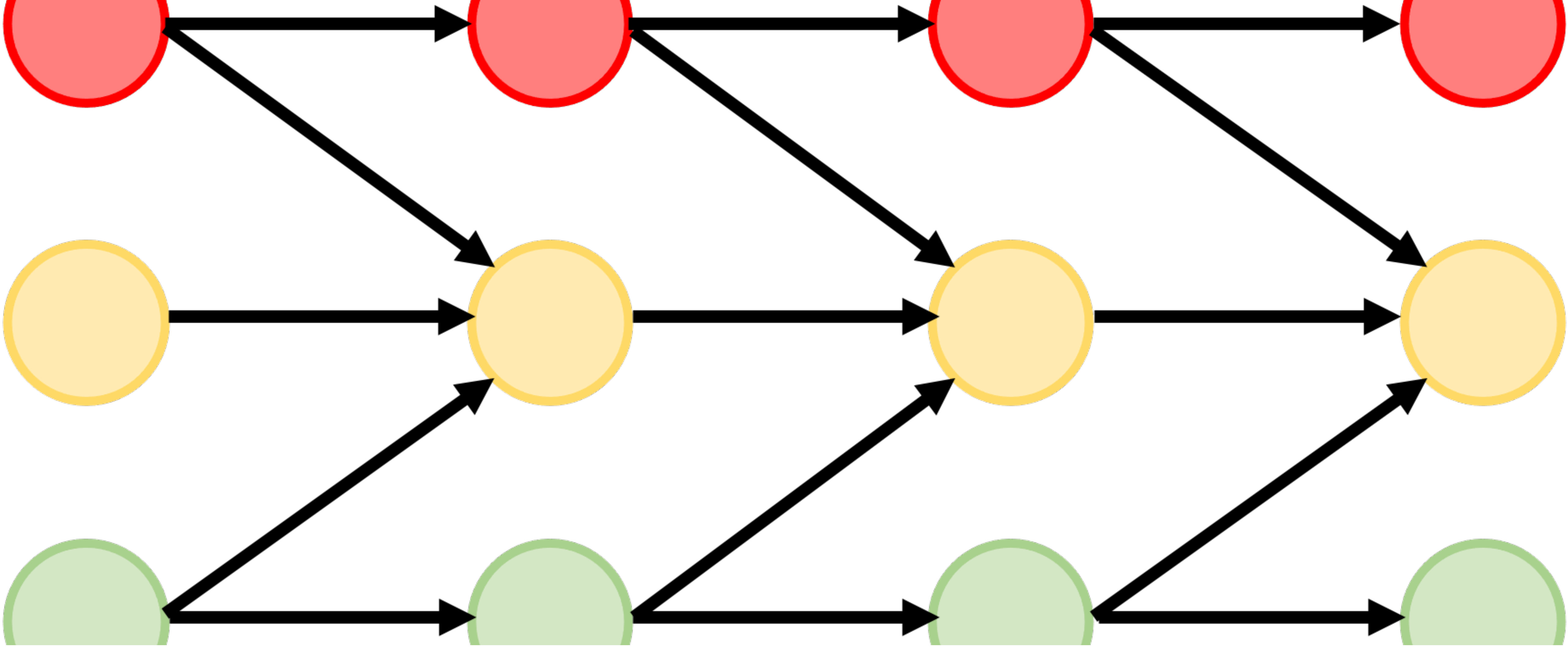






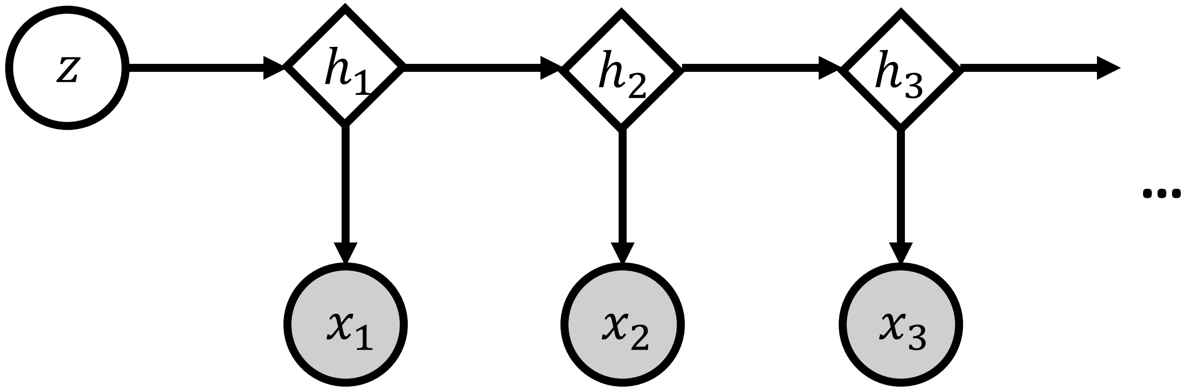
Today's Agenda

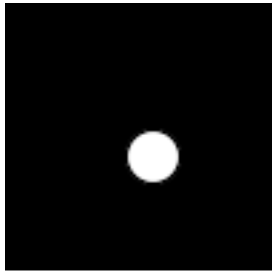
- Short tutorial on some basics
  - Discrete-time generative models
    - Deterministic dynamics
    - Stochastic dynamics
- Example: two recent works from us 😊
  - Markovian Gaussian Process VAEs
  - Identifiable Markov Switching Models



Discrete-Time Sequence Generative Models

Sequential VAE with Deterministic Dynamics





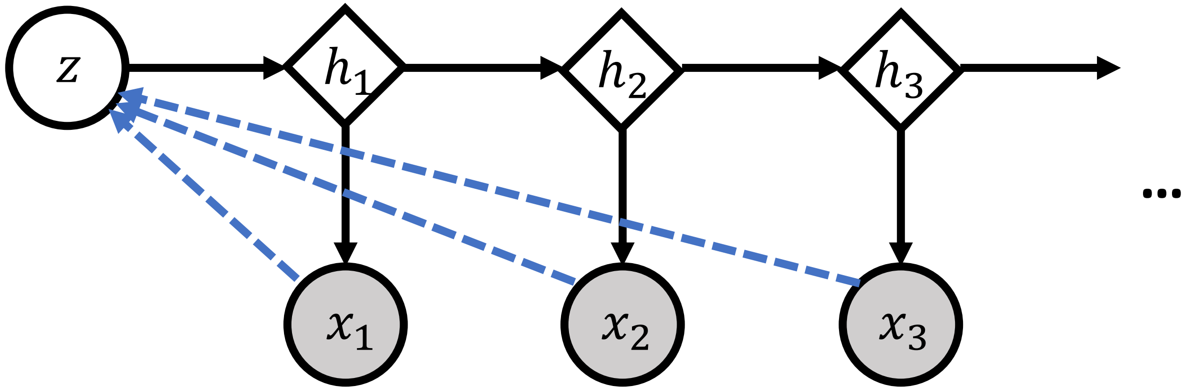
- The dynamic model is an RNN with random initial state $h_0 = z$
- Generative model (e.g., with Gaussian observations):

$$p_\theta(x_{1:T}) = \int p_\theta(x_{1:T}|z)p(z)dz, p(z) = N(z; 0, I)$$

$$p_\theta(x_{1:T}|z) = \prod_{t=1}^{T} N(x_t; G_\theta(h_t), \sigma^2 I)$$

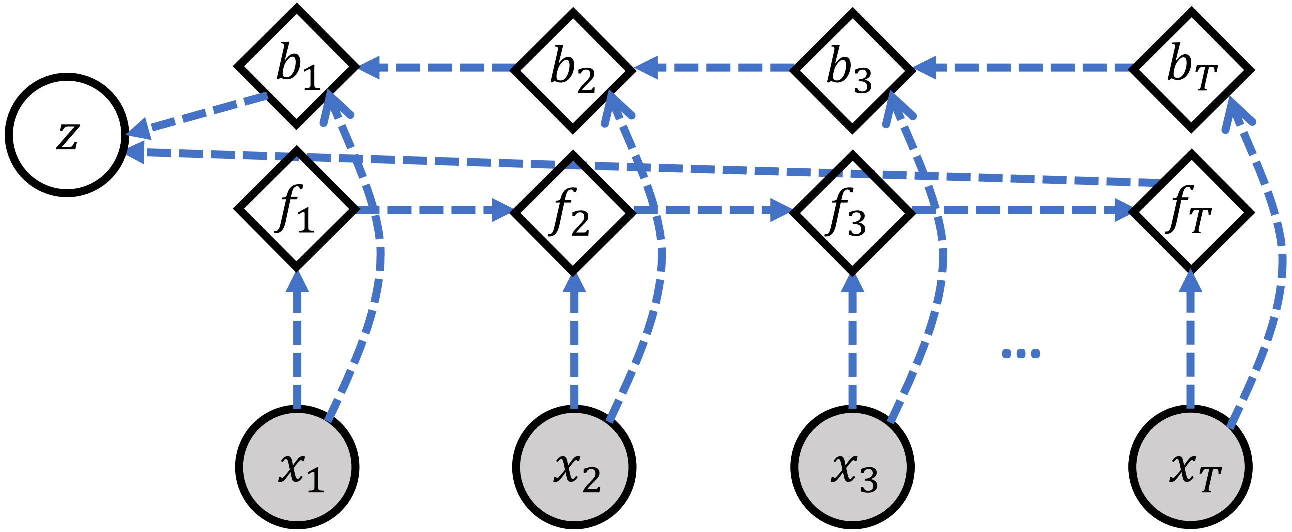
$$h_t = RNN_\theta(h_{t-1}), h_0 = z$$

Sequential VAE with Deterministic Dynamics

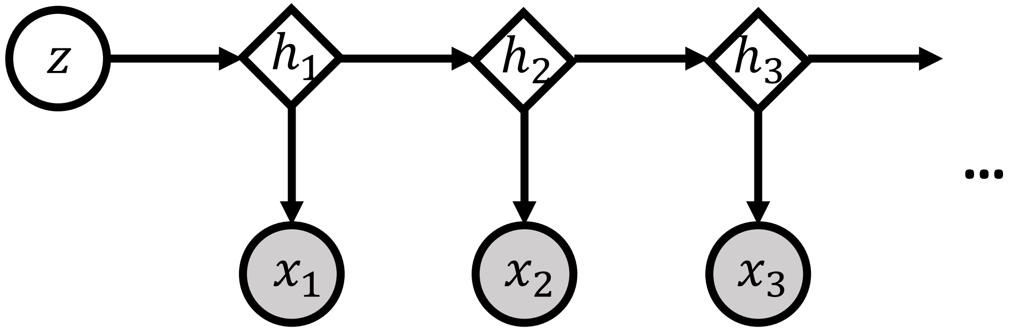


- Encoder $q(z|x_{1:T})$:
  infer z using $x_{1:T}$
- Example:

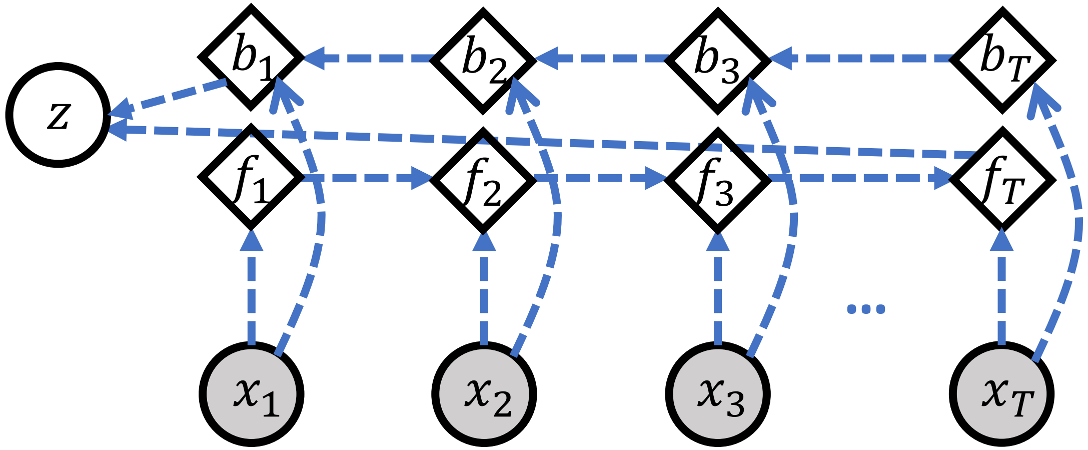
  Bi-directional RNN



Sequential VAE with Deterministic Dynamics



Generator/Decoder $p_\theta(x_{1:T}|z)$



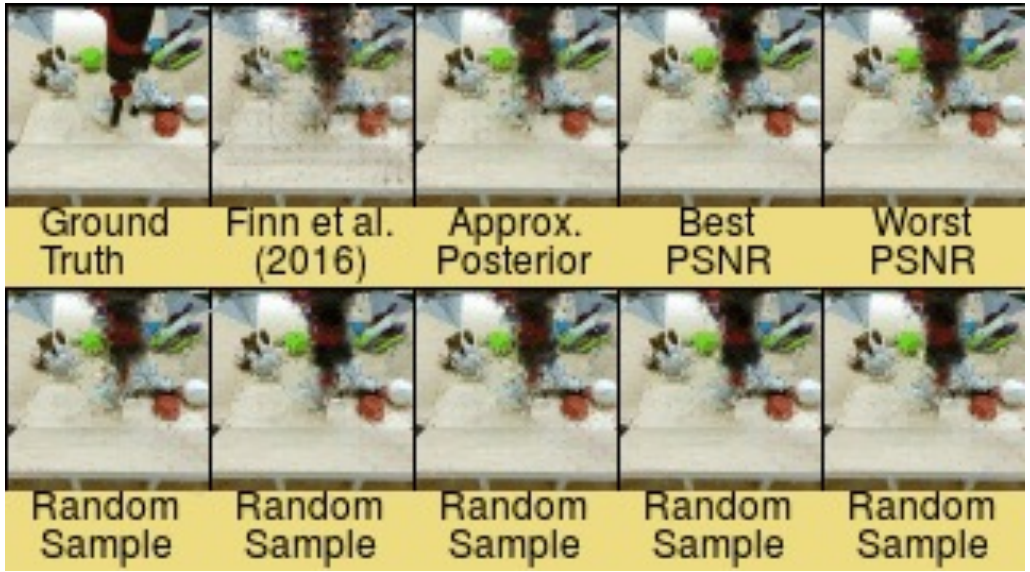
Encoder $q_\phi(z|x_{1:T})$

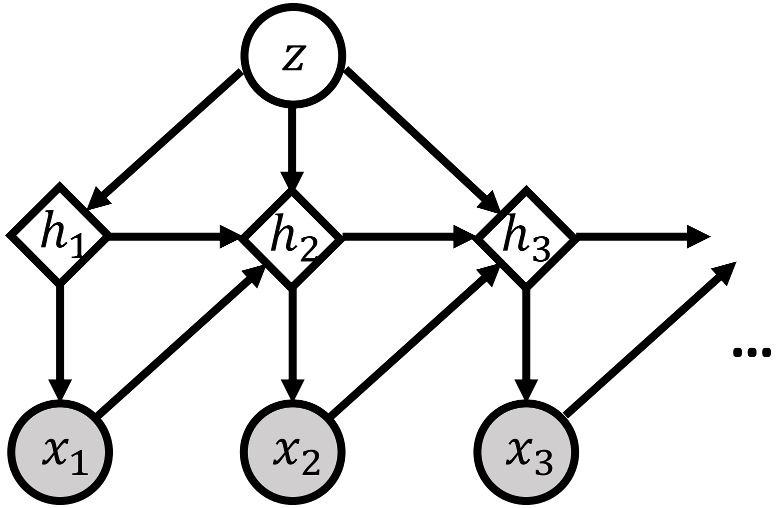
- β-ELBO:

$$ELBO = E_{q_\phi(z|x_{1:T})}[\log p_\theta(x_{1:T}|z)] - \beta KL[q_\phi(z|x_{1:T})||p(z)]$$

Sequential DGMs with Deterministic Dynamics

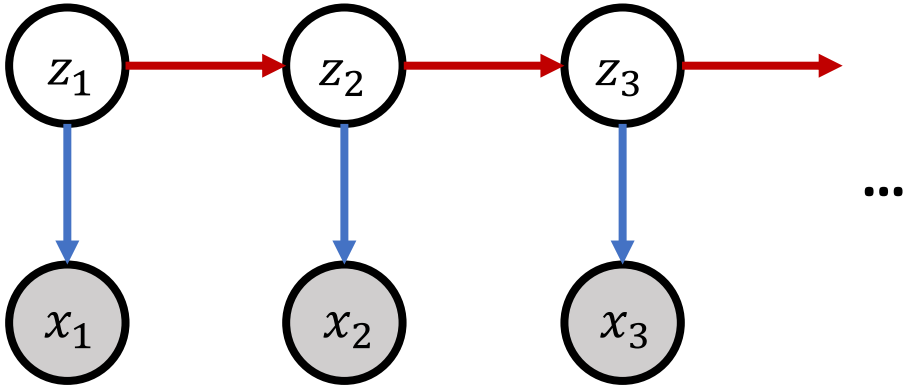
- RNN-based approaches conditioned on a latent variable z
  - Think about z as capturing "environment information"
  - With discrete z: regime-dependent sequence generative model





Babaeizadeh et al. Stochastic variational video prediction. ICLR 2017

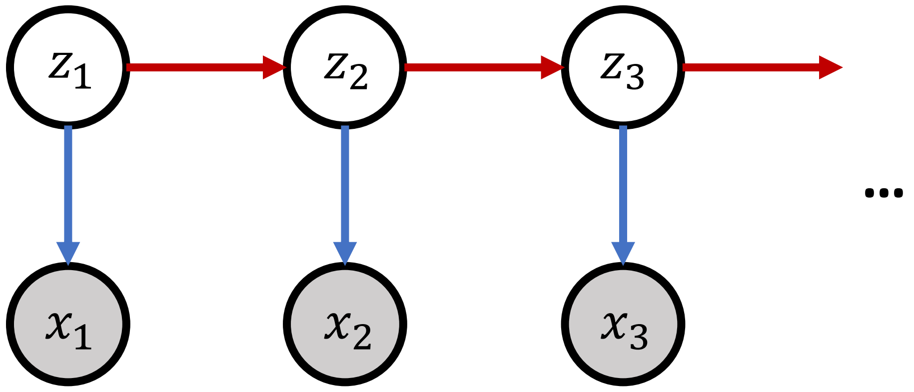
State-State Models (Stochastic Dynamics)



$$p(x_{1:T}, z_{1:T}) = \prod_{t=1}^{T} p(x_t|z_t)p(z_t|z_{t-1}),\ z_0 = \emptyset$$

- Stochastic dynamics: Given z_t, future trajectory $z_{t+1:T}$ is still stochastic

Linear-Gaussian SSM

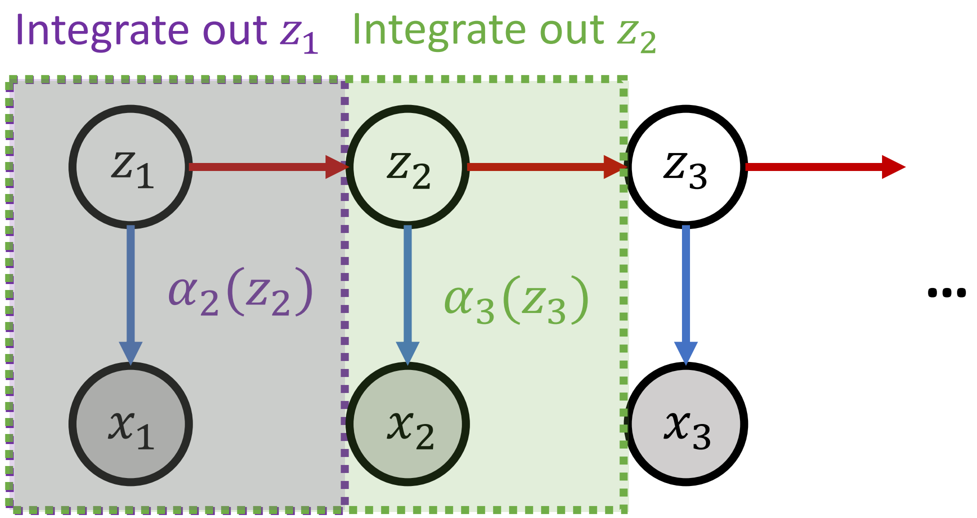


$$z_t = A_t z_{t-1} + \epsilon_t, \epsilon_t \sim N(0, R_t)$$
$$x_t = C_t z_t + \eta_t, \eta_t \sim N(0, Q_t)$$

$$p(x_{1:T}, z_{1:T}) = \prod_{t=1}^{T} p(x_t|z_t) p(z_t|z_{t-1}), \ z_0 = \emptyset$$

- Assuming stationarity: $A_t = A, R_t = R, C_t = C, Q_t = Q$
- Parameter learning by MLE: need to marginalise out $z_{1:T}$ ← Achieved by filtering
- Posterior inference: filtering (online) and smoothing

LG-SSM: Filtering



$$z_t = A_t z_{t-1} + \epsilon_t,\ \epsilon_t \sim N(0, R_t)$$

$$x_t = C_t z_t + \eta_t,\ \eta_t \sim N(0, Q_t)$$

$$p(x_{1:T}, z_{1:T}) = \prod_{t=1}^{T} p(x_t|z_t) p(z_t|z_{t-1}),\ \ z_0 = \emptyset$$

Filtering: set $\alpha_1(z_1) = p(z_1)$

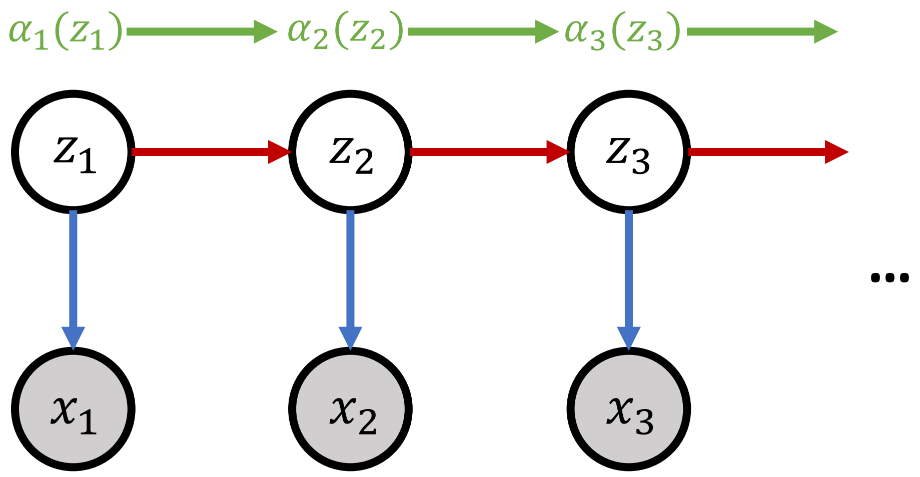
$$p(x_{1:T}) = \int p(x_{1:T}, z_{1:T}) dz_{1:T} = \int \prod_{t=1}^{T} p(x_t|z_t) p(z_t|z_{t-1})\, dz_{1:T}$$

$$= \int \underbrace{\left(\int p(x_1|z_1) p(z_2|z_1) \alpha_1(z_1) dz_1\right)}_{\alpha_2(z_2)} \prod_{t=2}^{T} p(x_t|z_t) \prod_{t=3}^{T} p(z_t|z_{t-1})\, dz_{2:T}$$

$$= \int \underbrace{\left(\int p(x_2|z_2) p(z_3|z_2) \alpha_2(z_2) dz_2\right)}_{\alpha_3(z_3)} \prod_{t=3}^{T} p(x_t|z_t) \prod_{t=4}^{T} p(z_t|z_{t-1})\, dz_{3:T}$$

$$= \cdots$$

LG-SSM: Filtering



$$z_t = A_t z_{t-1} + \epsilon_t, \epsilon_t \sim N(0, R_t)$$
$$x_t = C_t z_t + \eta_t, \eta_t \sim N(0, Q_t)$$

$$p(x_{1:T}, z_{1:T}) = \prod_{t=1}^{T} p(x_t|z_t) p(z_t|z_{t-1}), \;\; z_0 = \emptyset$$

Filtering: set $\alpha_1(z_1) = p(z_1)$

Compute for $t = 1, \ldots, T-1$:

$= p(x_{1:t}, z_{t+1})$ $\qquad$ $= p(x_{1:t-1}, z_t)$, assuming $x_{1:0} = \emptyset$

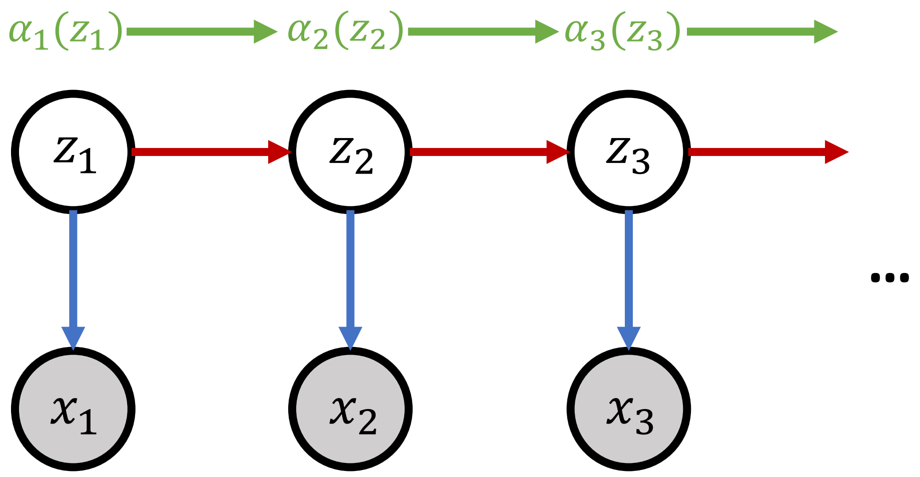
$$\alpha_{t+1}(z_{t+1}) = \int p(x_t|z_t)\; p(z_{t+1}|z_t)\; \alpha_t(z_t) dz_t$$

new forward msg $\qquad$ current obs $\qquad$ future transition $\qquad$ current forward msg

$$\Rightarrow \log \int \alpha_T(z_T) p(x_T|z_T) dz_T = \log \int p(x_{1:T-1}, z_T) p(x_T|z_T) dz_T = \log p(x_{1:T})$$

(what you need for MLE)

LG-SSM: Filtering



$$z_t = A_t z_{t-1} + \epsilon_t,\ \epsilon_t \sim N(0, R_t)$$
$$x_t = C_t z_t + \eta_t,\ \eta_t \sim N(0, Q_t)$$

$$p(x_{1:T}, z_{1:T}) = \prod_{t=1}^{T} p(x_t|z_t) p(z_t|z_{t-1}),\ \ z_0 = \emptyset$$

Posterior inference:

time step 1:t

Online inference: Given $x_{1:t}$, what is the "filtering" posterior $p(z_t|x_{1:t})$?

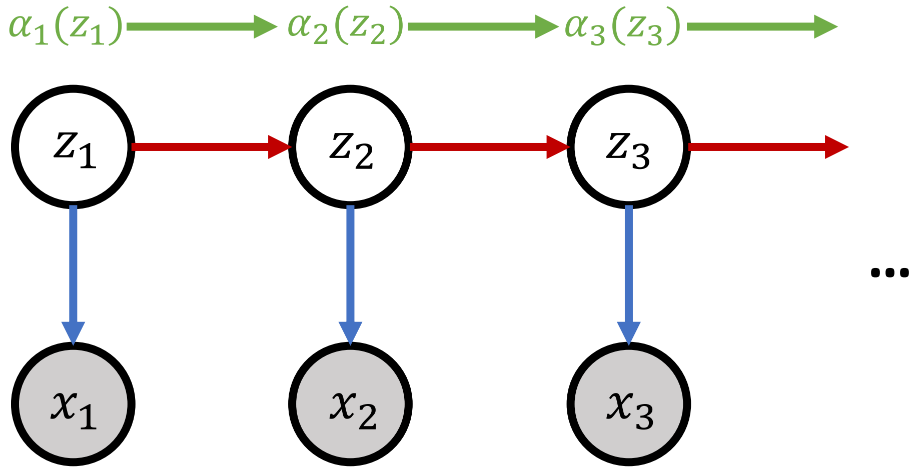
$$\alpha_t(z_t) = p(x_{1:t-1}, z_t)$$ (assuming $x_{1:0} = \emptyset$)

$$\Rightarrow p(z_t|x_{1:t}) = \frac{p(z_t, x_{1:t})}{p(x_{1:t})} = \frac{p(x_t|z_t)p(z_t, x_{1:t-1})}{\int p(x_t|z_t)p(z_t, x_{1:t-1}) z_t} = \frac{p(x_t|z_t)\alpha_t(z_t)}{\int p(x_t|z_t)\alpha_t(z_t) z_t} \propto p(x_t|z_t)\alpha_t(z_t)$$

current obs — current forward msg

filtering posterior ∝ current obs likelihood × current forward message

LG-SSM: Smoothing



$$z_t = A_t z_{t-1} + \epsilon_t, \epsilon_t \sim N(0, R_t)$$
$$x_t = C_t z_t + \eta_t, \eta_t \sim N(0, Q_t)$$

$$p(x_{1:T}, z_{1:T}) = \prod_{t=1}^{T} p(x_t|z_t) p(z_t|z_{t-1}), \ z_0 = \emptyset$$

Posterior inference:

time step 1:T

Online inference: Given $x_{1:t}$, what is the "smoothing" posterior $p(z_t|x_{1:T})$?

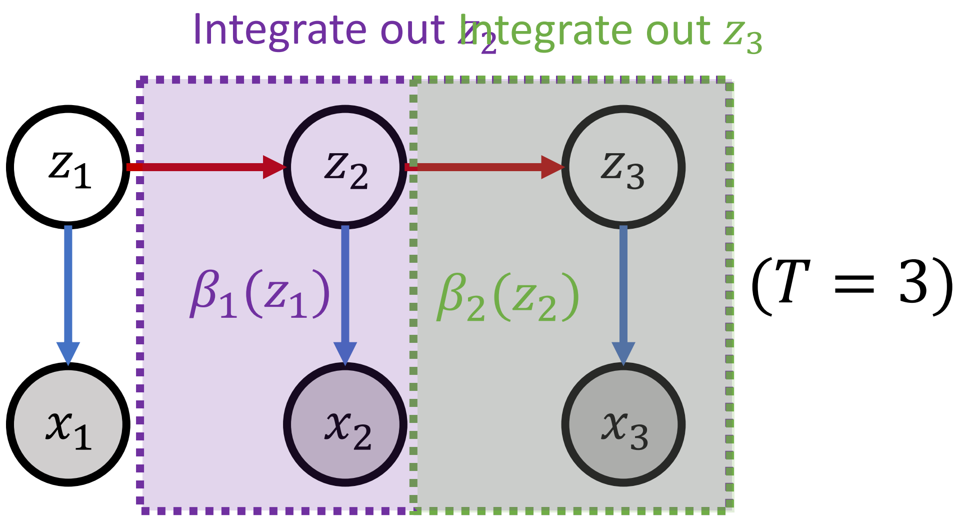
$$\alpha_t(z_t) = p(x_{1:t-1}, z_t)$$ (assuming $x_{1:0} = \emptyset$)

$$p(z_t|x_{1:T}) = \frac{p(z_t, x_{1:T})}{p(x_{1:T})} = \frac{p(x_{t:T}|z_t) p(z_t, x_{1:t-1})}{\int p(x_{t:T}|z_t) p(z_t, x_{1:t-1}) z_t} \propto p(x_{t:T}|z_t) \ \alpha_t(z_t)$$

$p(x_{t:T}|z_t)$: Current & future observations

$\alpha_t(z_t)$: current forward msg

LG-SSM: Smoothing



$z_t = A_t z_{t-1} + \epsilon_t, \epsilon_t \sim N(0, R_t)$

$x_t = C_t z_t + \eta_t, \eta_t \sim N(0, Q_t)$

$$p(x_{1:T}, z_{1:T}) = \prod_{t=1}^{T} p(x_t|z_t) p(z_t|z_{t-1}),\ \ z_0 = \emptyset$$

Smoothing: set $\beta_T(z_T) = 1$

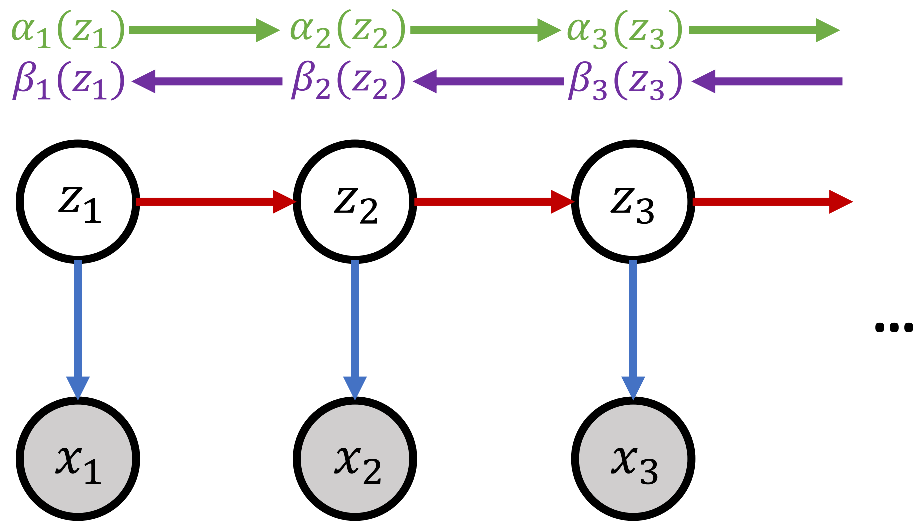
$$p(x_{t:T}|z_t) = \int p(x_{t:T}, z_{t+1:T}|z_t) dz_{t+1:T} = \int \prod_{\tau=t}^{T} p(x_\tau|z_\tau) p(z_\tau|z_{\tau-1})\, dz_{t+1:T}$$

$$= \int \underbrace{\left(\int p(x_T|z_T) p(z_T|z_{T-1}) \beta_T(z_T) dz_T\right)}_{\beta_{T-1}(z_{T-1})} \prod_{\tau=t}^{T-1} p(x_\tau|z_\tau) p(z_\tau|z_{\tau-1})\, dz_{t+1:T-1}$$

$$= \int \underbrace{\left(\int p(x_{T-1}|z_{T-1}) p(z_{T-1}|z_{T-2}) \beta_{T-1}(z_{T-1}) dz_{T-1}\right)}_{\beta_{T-2}(z_{T-2})} \prod_{\tau=t}^{T-2} p(x_\tau|z_\tau) p(z_\tau|z_{\tau-1})\, dz_{t+1:T-2}$$

$$= \cdots$$

LG-SSM: Smoothing



$$z_t = A_t z_{t-1} + \epsilon_t, \epsilon_t \sim N(0, R_t)$$

$$x_t = C_t z_t + \eta_t, \eta_t \sim N(0, Q_t)$$

$$p(x_{1:T}, z_{1:T}) = \prod_{t=1}^{T} p(x_t|z_t)p(z_t|z_{t-1}), \ z_0 = \emptyset$$

Smoothing: set $\beta_T(z_T) = 1$

Compute for $t = T-1, \dots, 1$ (backward!):

$= p(x_{t:T}|z_{t-1})$ $= p(x_{t+1:T}|z_t)$, assuming $x_{T+1:T} = \emptyset$

$$\beta_{t-1}(z_{t-1}) = \int p(x_t|z_t)\, p(z_t|z_{t-1})\, \beta_t(z_t) dz_t$$

new backward msg; current obs; current transition; current backward msg

Smoothing posterior

$$p(z_t|x_{1:T}) \propto p(x_{t:T}|z_t)\, \alpha_t(z_t) \propto p(x_t|z_t)p(x_{t+1:T}|z_t)\alpha(z_t) \propto p(x_t|z_t)\beta_t(z_t)\alpha(z_t)$$

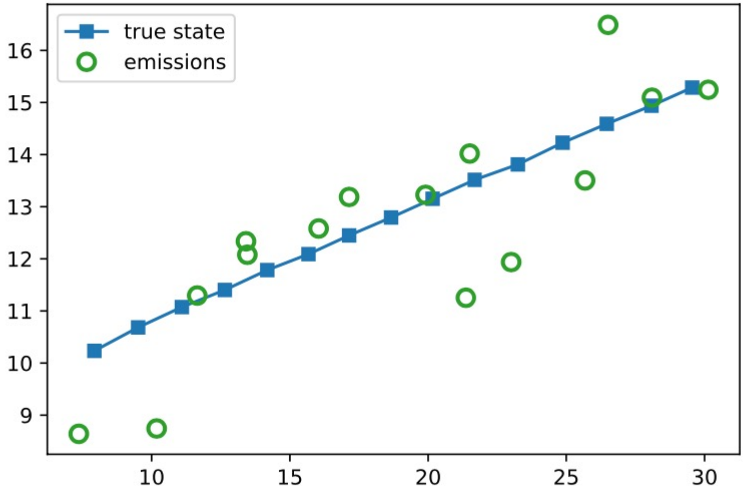
current obs; backward msg; forward msg

smoothing posterior $\propto$ obs likelihood $\times$ forward msg $\times$ backward msg

LG-SSM: Example

- Recovering the ground-truth trajectory given its noisy observations:

$$z_t = A_t z_{t-1} + \epsilon_t, \epsilon_t \sim N(0, R_t), x_t = C_t z_t + \eta_t, \eta_t \sim N(0, Q_t)$$



noisy data

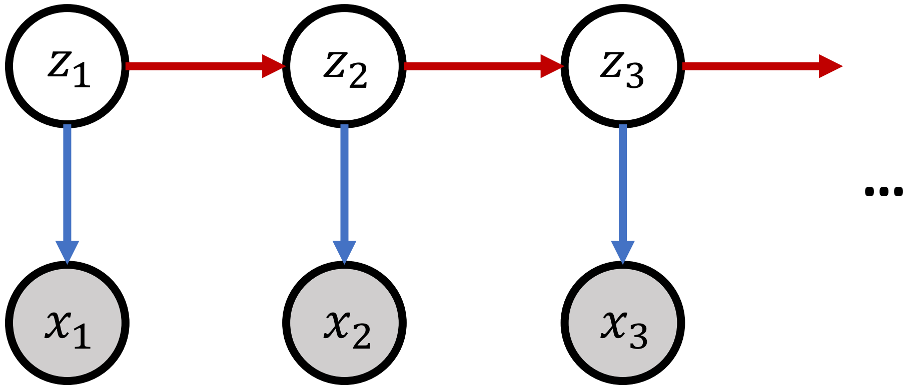
Murphy (2023). Probabilistic Machine Learning: Advanced Topics. MIT Press

SSMs with Non-Linear Dynamics

$$p(x_{1:T}, z_{1:T}) = \prod_{t=1}^{T} p(x_t|z_t)p(z_t|z_{t-1})$$

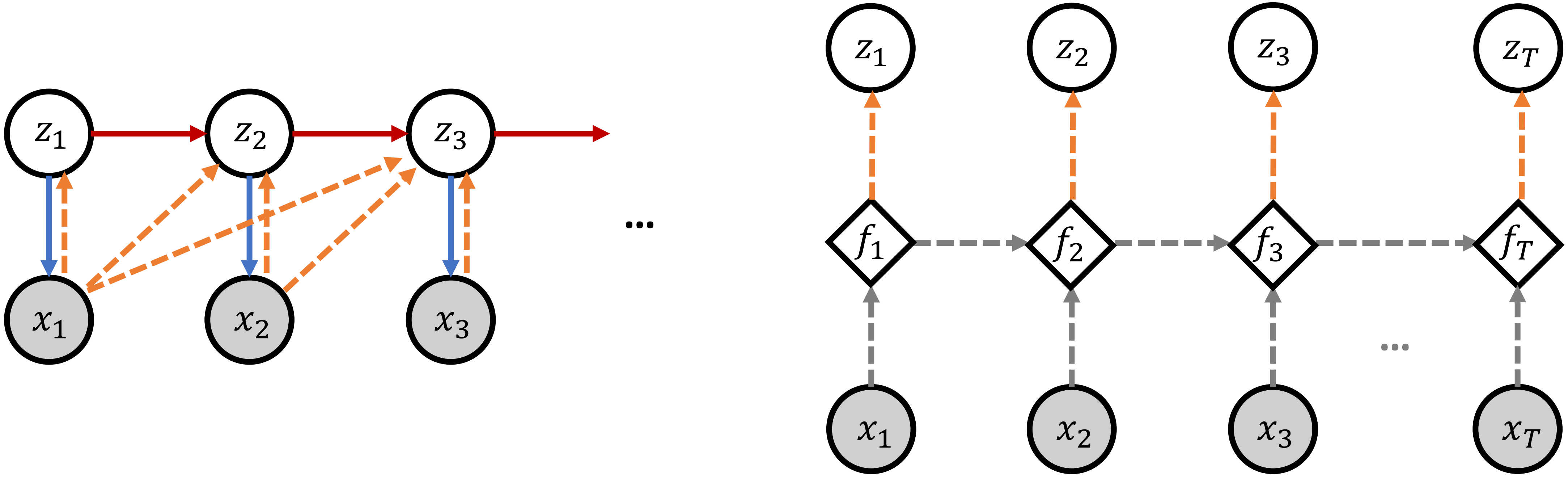
$$p(z_t|z_{t-1}) = N(z_t; f(z_{t-1}), R(z_{t-1}))$$

$$p(x_t|z_t) = N(x_t; g(z_t), Q(z_t))$$



- Non-linear dynamics: f, R, g, Q are non-linear functions (e.g., neural networks)
- Parameter learning by MLE: need to marginalise out $z_{1:T}$ ← now intractable
- Ideas for approximate inference:
  - Extended Kalman filtering/smoothing
  - Amortised variational inference (VAE + SSM)

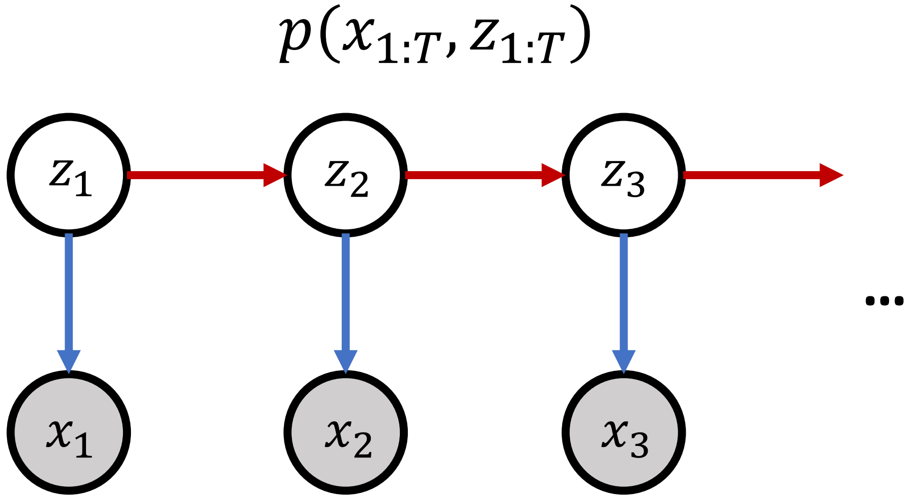
VAE + SSM

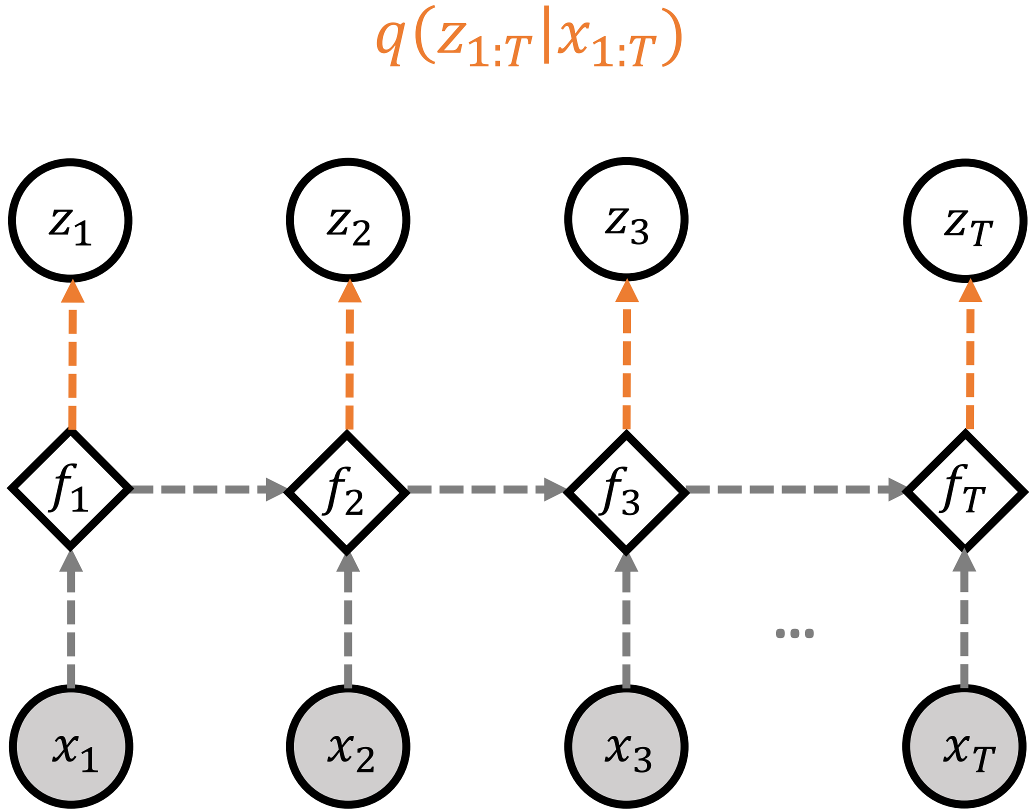


- Filtering Encoder $q(z_{1:t}|x_{1:t})$: infer $z_{1:t}$ using $x_{1:t}$
- Example: Forward RNN

time step 1:t

VAE + SSM





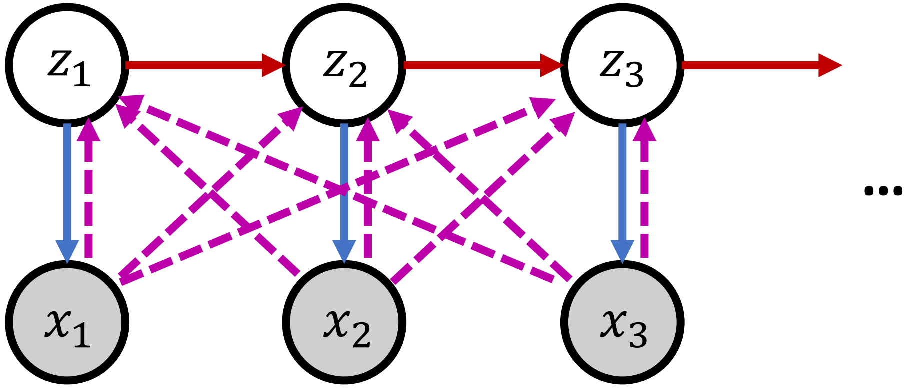
- β-ELBO:

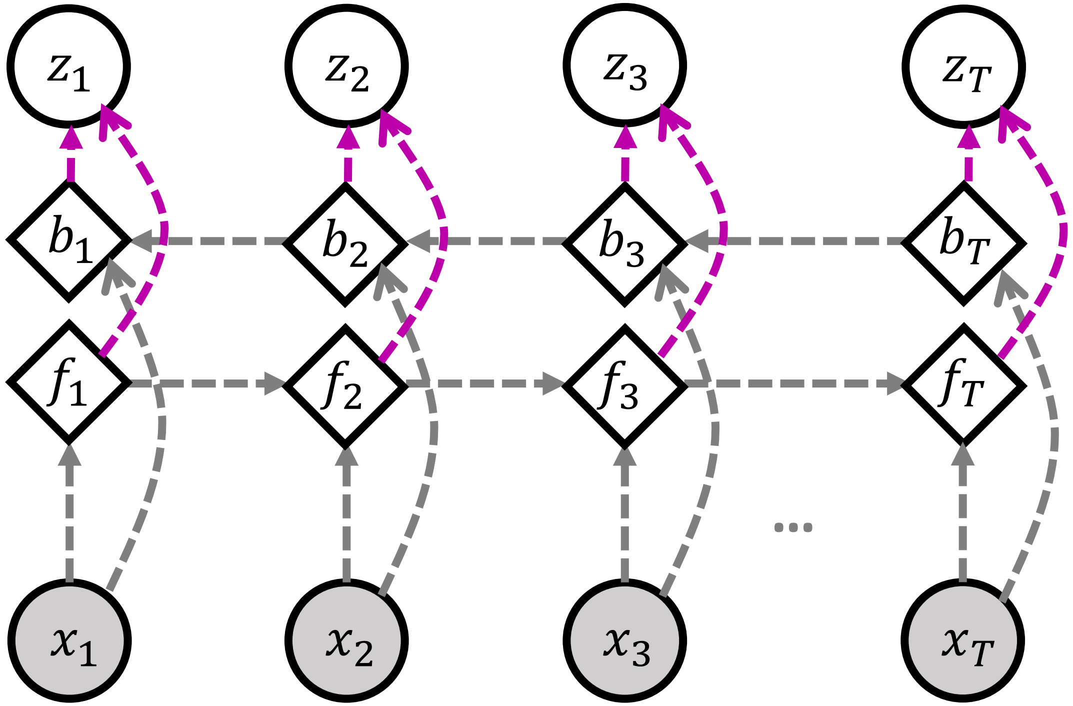
$$ELBO = E_{q(z_{1:T}|x_{1:T})}[\log p(x_{1:T}|z_{1:T})] - \beta KL[q(z_{1:T}|x_{1:T})||p(z_{1:T})]$$

$$= \sum_{t=1}^{T} E_{q(z_t|x_{1:t})}[\log p(x_t|z_t)] - \beta KL[q(z_{1:T}|x_{1:T})||p(z_{1:T})]$$

filtering posterior

VAE + SSM

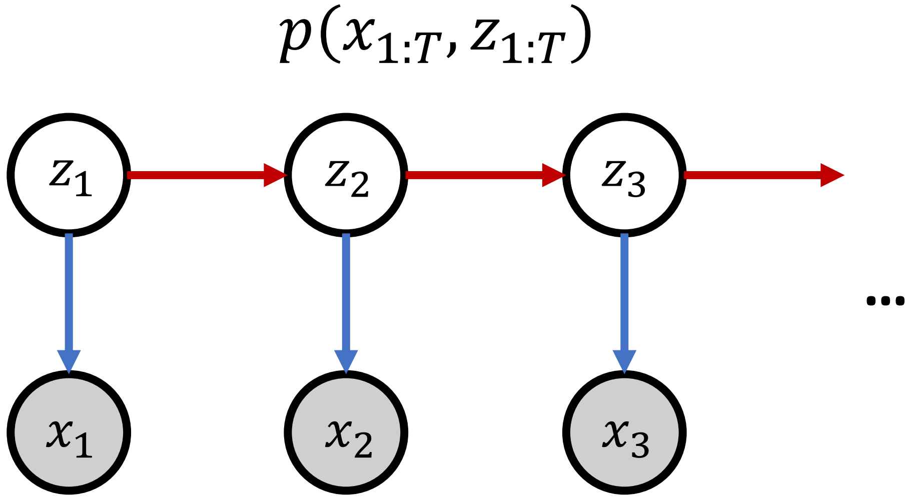


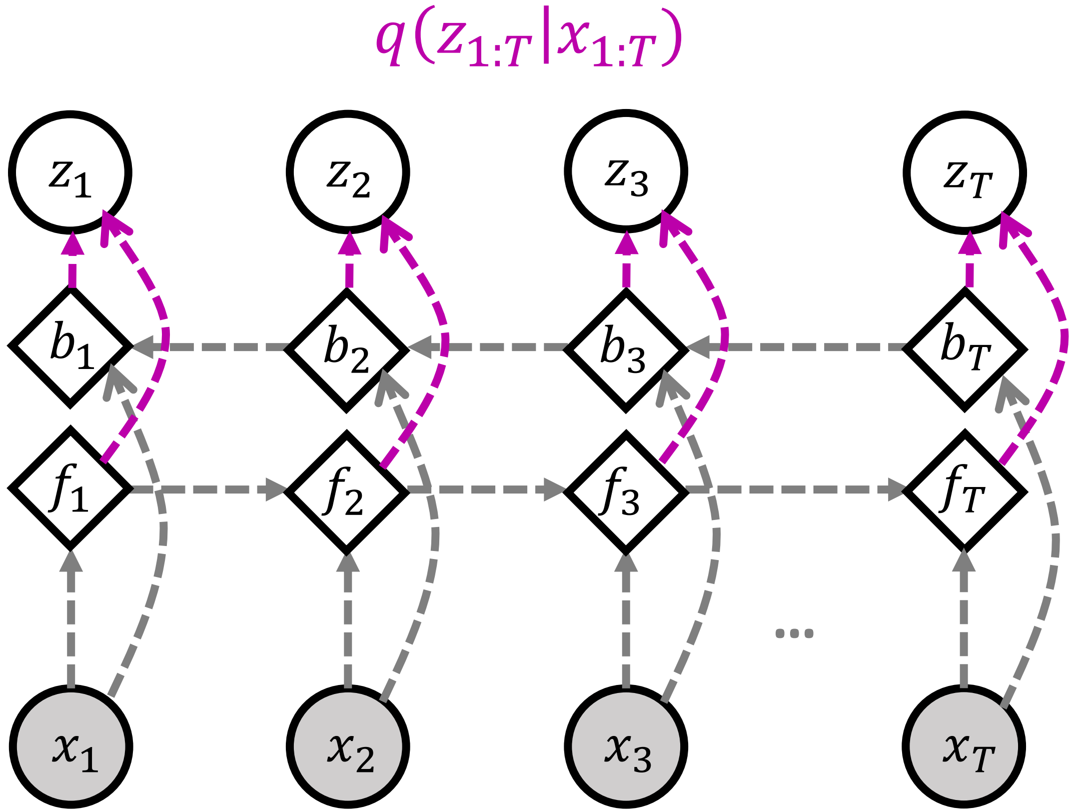


- Smoothing Encoder $q(z_{1:t}|x_{1:t})$: infer $z_{1:t}$ using $x_{1:T}$
- Example: Bi-directional RNN

time step 1:T

VAE + SSM





- β-ELBO:

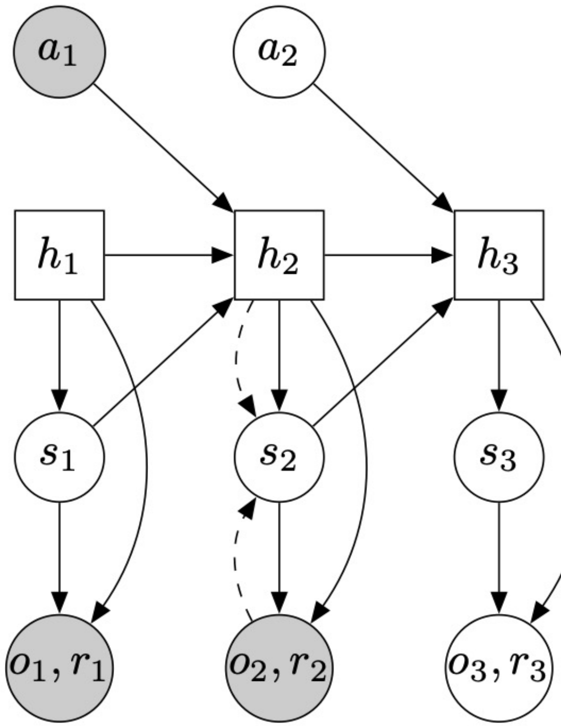
$$ELBO = E_{q(z_{1:T}|x_{1:T})}[\log p(x_{1:T}|z_{1:T})] - \beta KL[q(z_{1:T}|x_{1:T}) \| p(z_{1:T})]$$

$$= \sum_{t=1}^{T} E_{q(z_t|x_{1:T})}[\log p(x_t|z_t)] - \beta KL[q(z_{1:T}|x_{1:T}) \| p(z_{1:T})]$$

smoothing posterior

SOTA SSM: Recurrent SSM

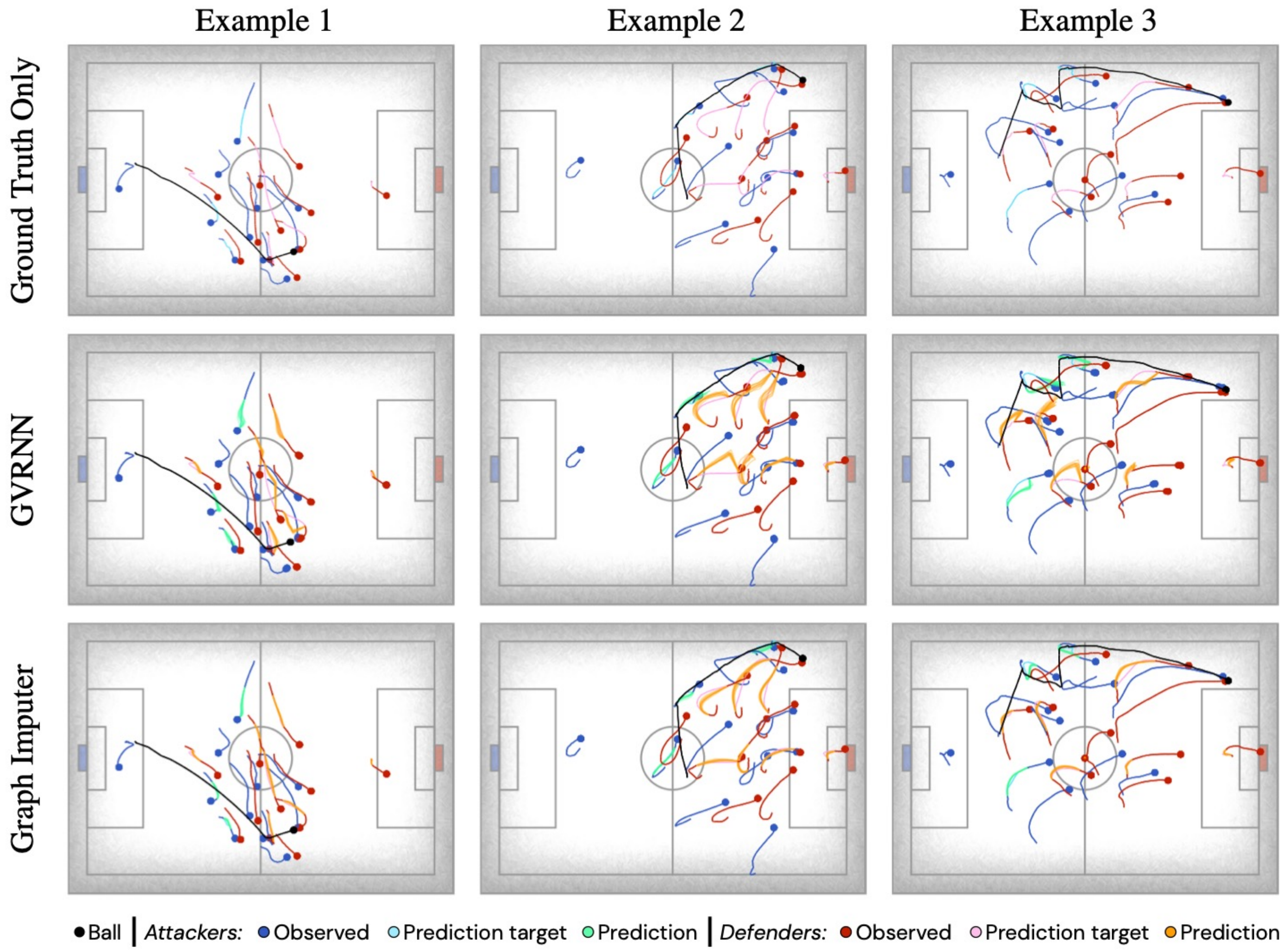
Applications in model-based RL:



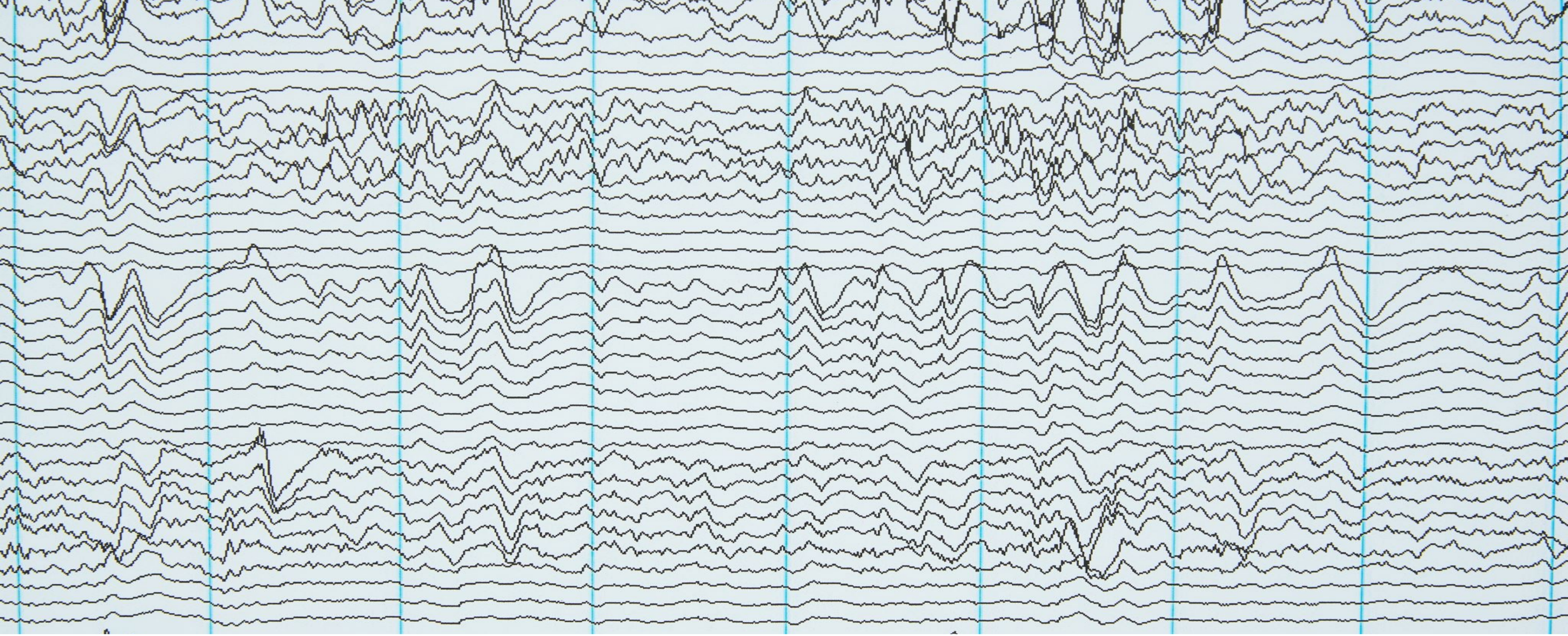
Hafner et al. Learning Latent Dynamics for Planning from Pixels. ICML 2019
Hafner et al. Mastering Diverse Domains through World Models. arXiv:2301.04104

Application to Sport Data Analysis

Graph Neural Network
+ Variational RNN
+ bidirectional model



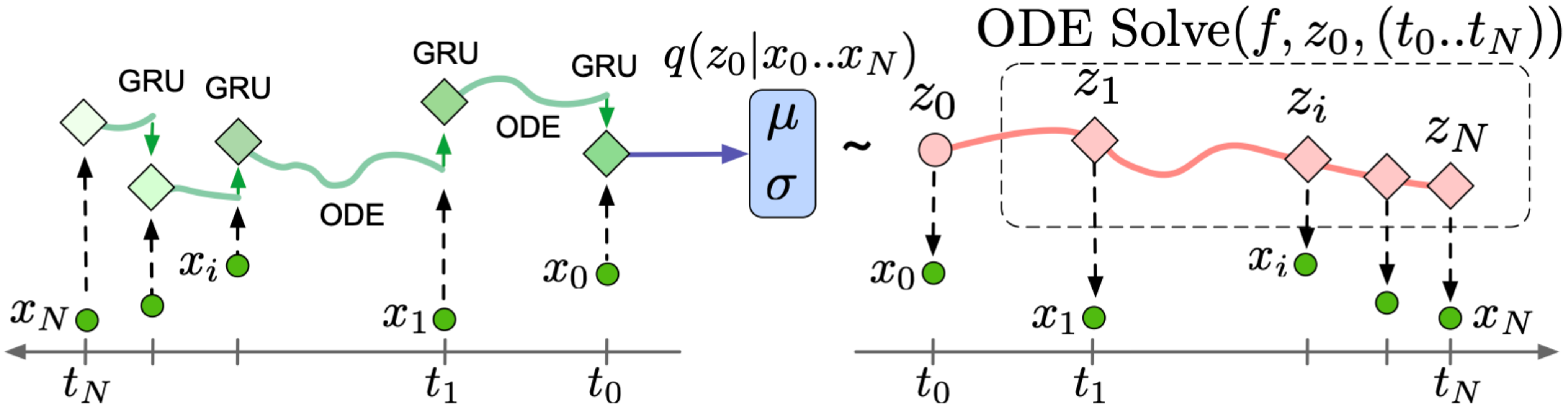
Chung et al. A Recurrent Latent Variable Model for Sequential Data. NIPS 2015
Omidshafiei et al. (2022) Multiagent off-screen behavior prediction in football. Scientific Reports.



Continuous-Time Sequence Generative Models

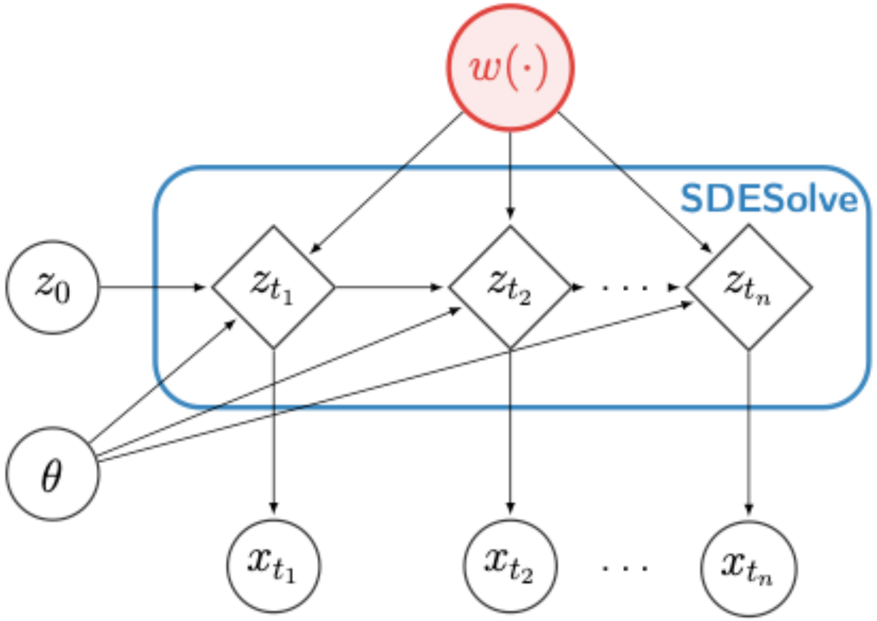
Latent Neural ODE

Handling irregularly sampled time-series with underlying deterministic dynamics:



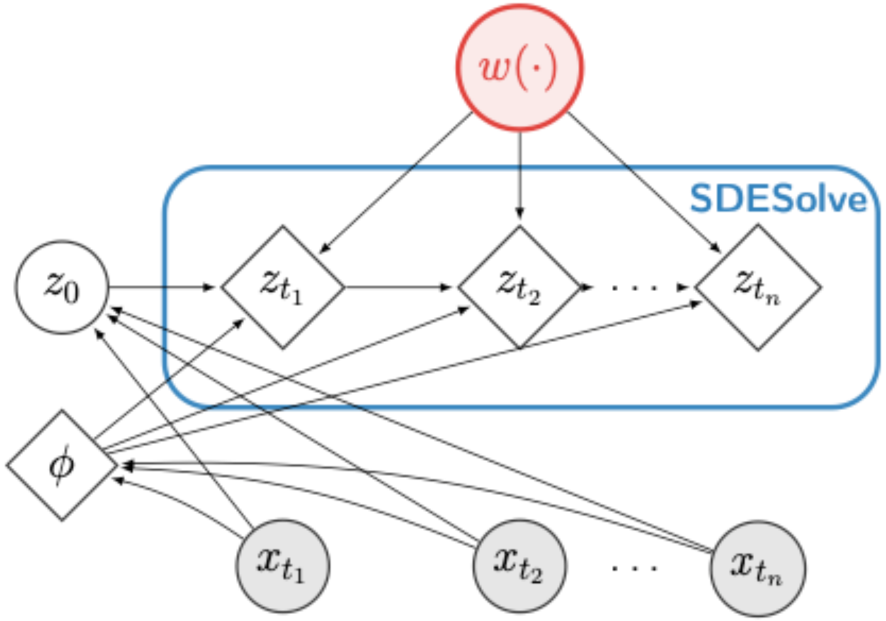
Rubanova et al. Latent ODEs for Irregularly-Sampled Time Series. NeurIPS 2019

Latent Neural SDE



$$dz_t = f_\theta(z_t, t) + \sigma_\theta(z_t, t) dW_t$$
$$z_0 \sim p(z_0)$$
$$x_{t_i} \sim p(x_{t_i} | z_{t_i})$$

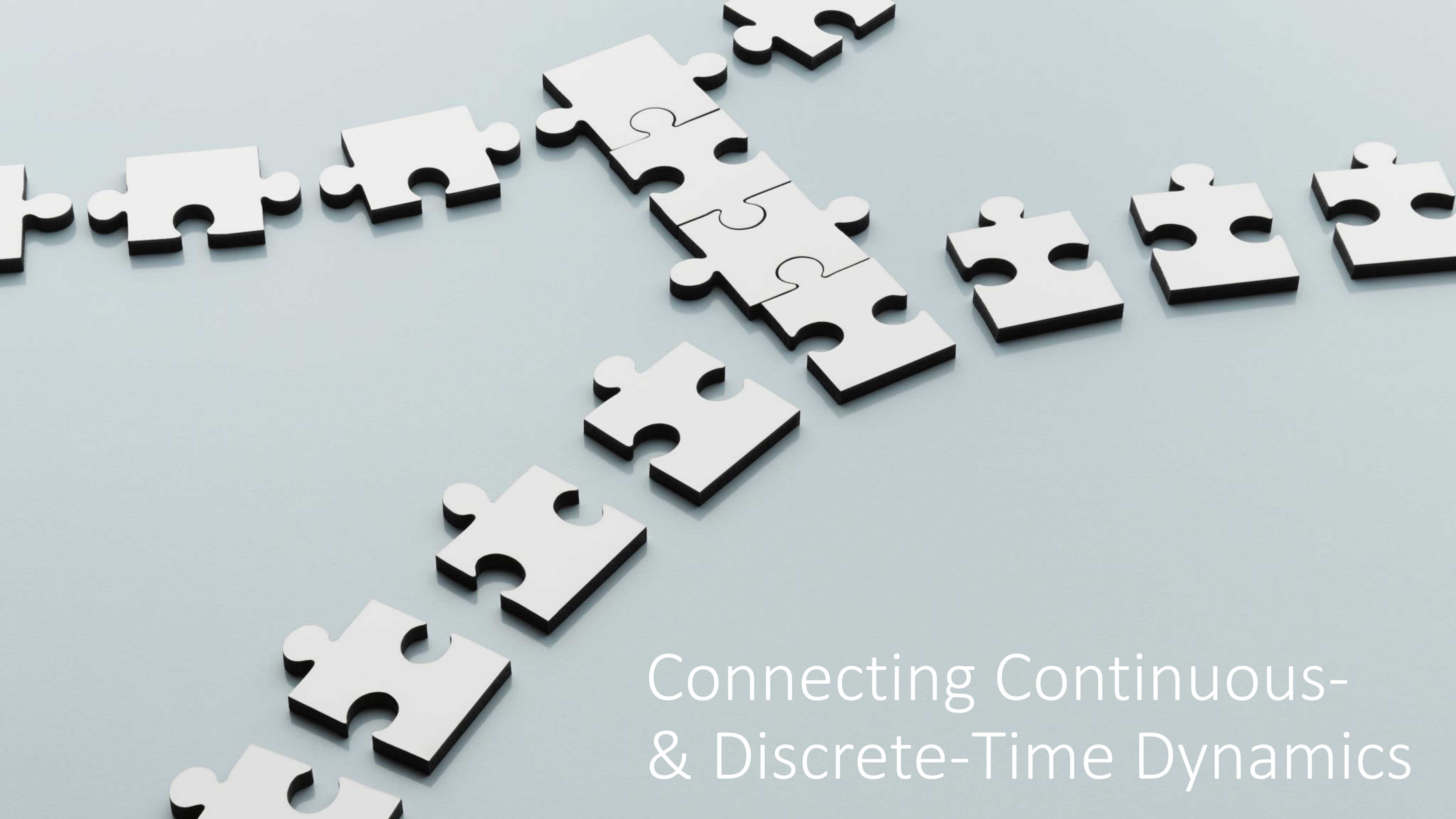
p model



$$dz_t = g_\phi(z_t, t) + \sigma_\theta(z_t, t) dW_t$$
$$z_0 \sim q(z_0 | \{x_{t_i}\})$$
$$\phi = \phi(\{x_{t_i}\})$$

q inference network

Li et al. Scalable Gradients for Stochastic Differential Equations. AISTATS 2020



Connecting Continuous-
& Discrete-Time Dynamics

Gaussian Processes for Time-Series Modelling

Gaussian Process: distribution over functions

$$f(\cdot) \sim GP\big(m(\cdot), K(\cdot,\cdot)\big), y = f(x) + \sigma\epsilon, \epsilon \sim N(0,1)$$

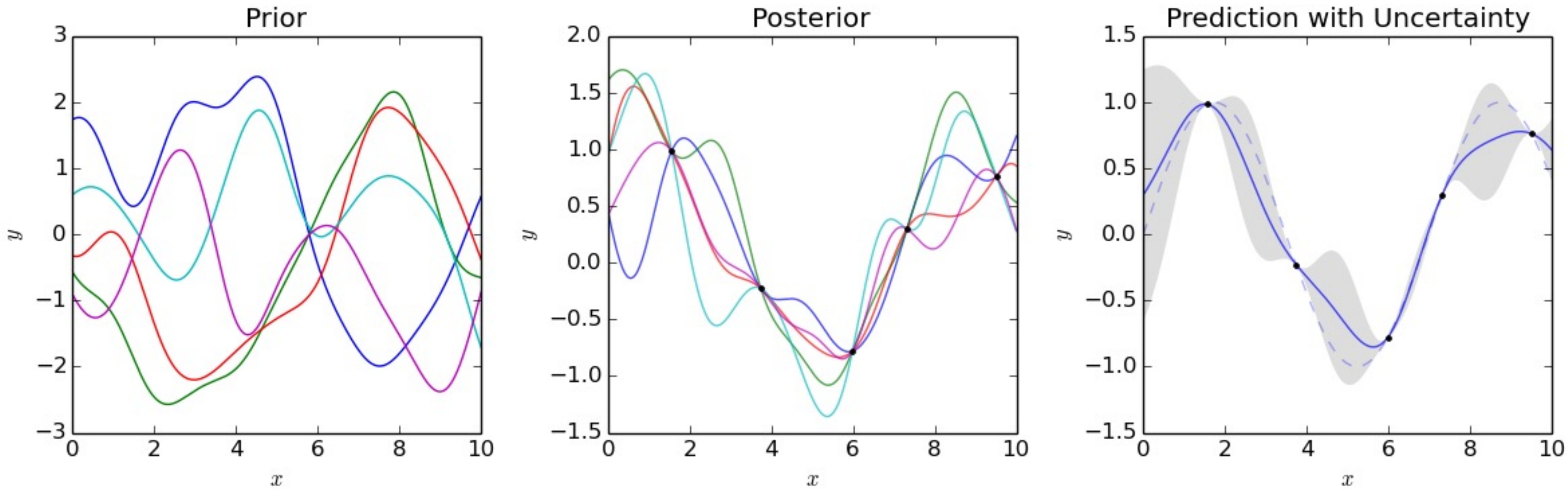


Fig credit: Richard Wilkinson's GPSS 2019 lecture

Gaussian Process VAEs

GPVAE as a sequence generative model:

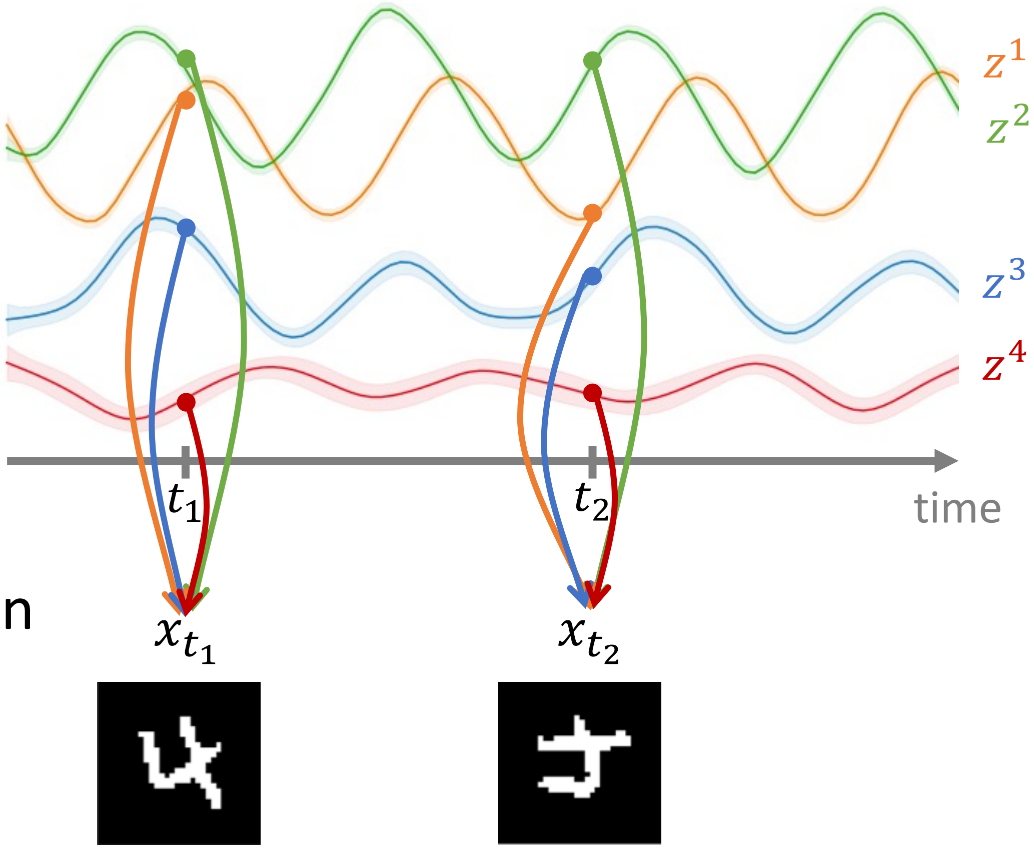
- Prior dynamics defined by specifying global behaviour:

$$z^d(\cdot) \sim GP\big(0, K(\cdot,\cdot)\big),\ d = 1, \dots, D_z$$

  - i.e., D_z number of functions with GP prior
  - c.f., Latent ODE/SDE: defining transitions

- At any time step t, use a decoder to transform the latent variables to observation

$$x_t \sim p(x_t|z_t), z_t = (z^1(t), z^2(t), \dots, z^{D_z}(t))$$

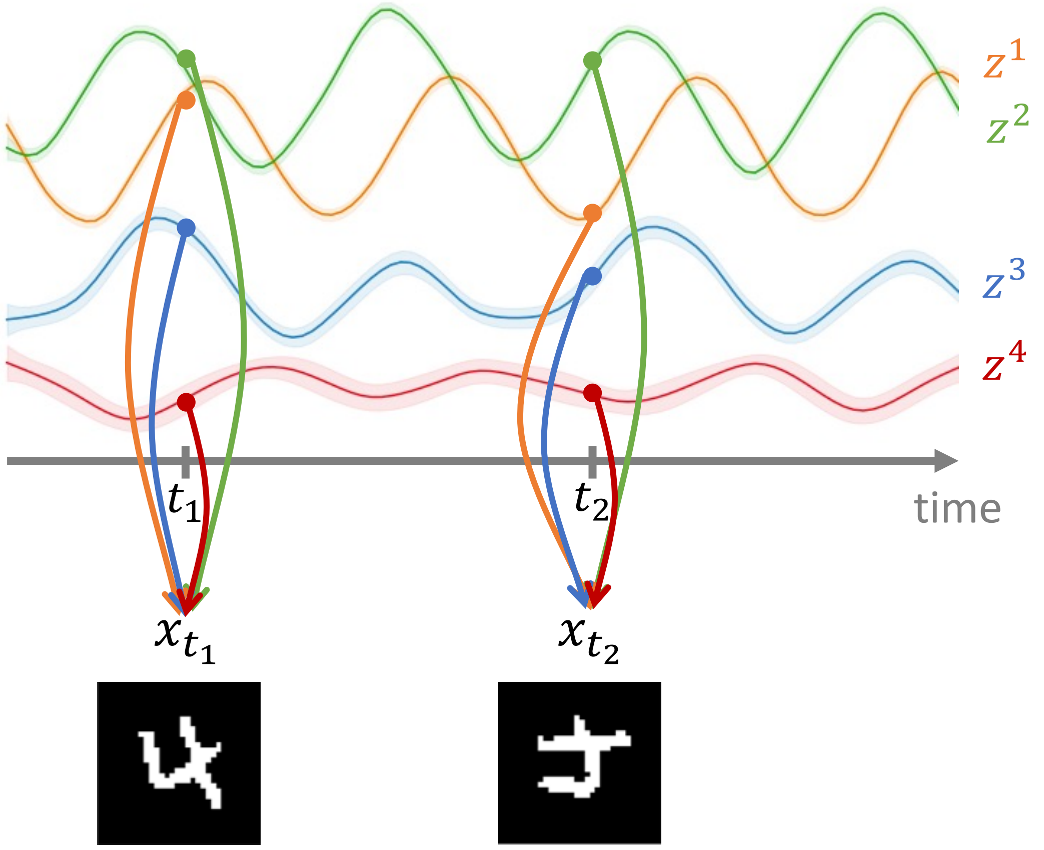


Casale et al. Gaussian process prior variational autoencoders. NeurIPS 2018
Fortuin et al. Gp-VAE: Deep probabilistic time series imputation. AISTATS 2020

Gaussian Process VAEs

GPVAE: GP dynamics prior + neural network decoder + inference network

- ☑ The kernel explicitly enforces inductive biases + global behaviour
- ☑ Continuous-time
  (Can do interpolation & handle irregular time-series
- ☒ $O(T^3)$ complexity and $O(T^2)$ storage
  - sparse inducing points with $O(Tm^2 + m^3)$ with $O(Tm + m^2)$ storage
  - Number of inducing points $m = O(\log^{\mathrm{D}} T)$



Casale et al. Gaussian process prior variational autoencoders. NeurIPS 2018
Jazbec et al. Scalable gaussian process variational autoencoders. AISTATS 2021
Burt et al. Rates of convergence for sparse variational Gaussian process regression. ICML 2019

Imperial College London

Markovian Gaussian Process Variational Autoencoders

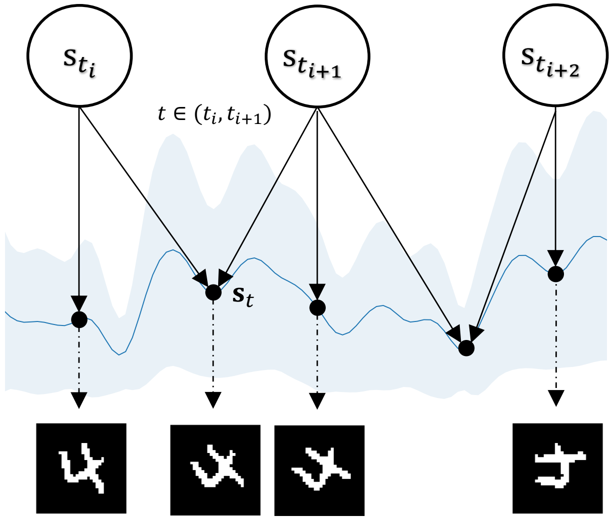
ICML 2023

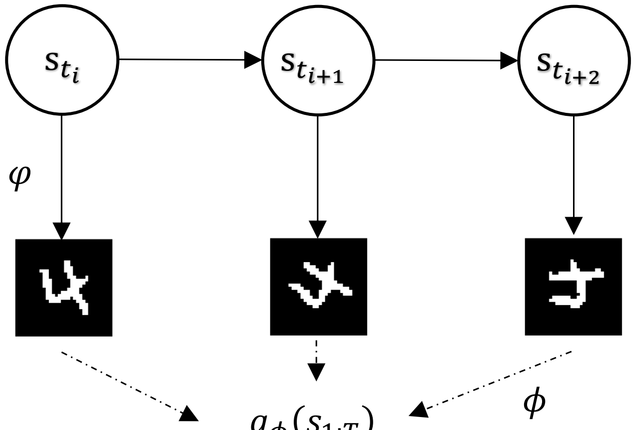
Harrison Zhu[1]* Carles Balsells Rodas[1]* Yingzhen Li[1]



[1]Imperial College London

Markovian GPVAE: Main Idea





Generative model:

- GP prior with Markovian kernel
- Equiv. to have a linear SDE prior in augmented space
- This allows discrete-time computations

Inference network (site approx.):

- Build another "generative model" $\tilde{p}$ with tractable exact posterior
- Define approximate approximation as $q(s_{1:T}) \coloneqq \tilde{p}(s_{1:T}|x_{1:T})$
- Train by optimising $ELBO(p, q)$

H Zhu, C Balsells Rodas and **Y Li.** Markovian Gaussian Process Variational Autoencoders. ICML 2023

Markovian Gaussian Processes

- GP with Markovian kernel K has an equivalent linear SDE form:

$$z(\cdot) \sim (0, K(\cdot,\cdot)) \quad \Leftrightarrow \quad \begin{gathered} \mathrm{d}\boldsymbol{s}(t) = \mathbf{Fs}(t)\mathrm{d}t + \mathbf{L}\mathrm{d}B_t, \\ z(\mathrm{t}) = \mathbf{Hs}(t) \end{gathered}$$

$\mathbf{F} \in R^{d \times d}, \mathbf{L} \in R^{d \times e}, \mathbf{H} \in R^{1 \times d}$

B_t is a e-dim Brownian motion with diffusion Q_c

Detailed derivations not important for this work, but typically:

$$\mathbf{s}(t) = \left(z(t), z'(t), \ldots, z^{(d-1)}(t)\right)^{\top}, \qquad \boldsymbol{H} = (1, 0, \ldots, 0)$$

(derivatives of the z function up to degree $d - 1$)

Särkkä and Solin (2019). Applied stochastic differential equations. Cambridge University Press

Markovian Gaussian Processes

- Discrete-time computation: Computing $\mathbf{s}_{t_{i+1}} := \boldsymbol{s}(t_{i+1})$ given $\mathbf{s}_{t_i} \coloneqq \boldsymbol{s}(t_i)$:

$$\mathrm{d}\boldsymbol{s}(t) = \mathbf{F}\boldsymbol{s}(t)\mathrm{dt} + \mathbf{L}\mathrm{d}B_t,$$
$$z(\mathrm{t}) = \mathbf{H}\boldsymbol{s}(t)$$

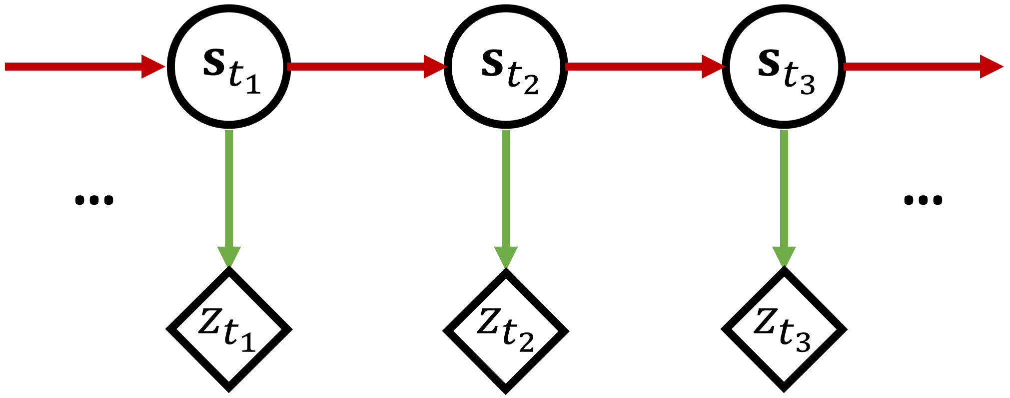
$\mathbf{F} \in R^{d\times d}, \mathbf{L} \in R^{d\times e}, \mathbf{H} \in R^{1\times d}$

B_t is a e-dim Brownian motion with diffusion Q_c

$\Rightarrow$

$$\mathbf{s}_{t_{i+1}} = \mathbf{A}_{i,i+1}\mathbf{s}_{t_i} + \mathbf{q}_i, \quad \mathbf{q}_i \sim \mathcal{N}(0, \mathbf{Q}_{i,i+1}),$$
$$\text{with } \mathbf{A}_{i,i+1} = e^{\Delta_i \mathbf{F}}, \quad \text{where } \Delta_i = t_{i+1} - t_i,$$
$$\mathbf{Q}_{i,i+1} = \int_{t_0}^{\Delta_i + t_0} e^{(\Delta_i + t_0 - \tau)\mathbf{F}} \mathbf{L}\mathbf{Q}_c\mathbf{L}^\intercal [e^{(\Delta_i + t_0 - \tau)\mathbf{F}}]^\intercal \mathrm{d}\tau.$$

- Markovian GP prior
  $\Rightarrow$ latent SSM with states $\{\mathbf{s}_t\}$:
  - Linear Gaussian transitions
  - Noiseless linear emissions

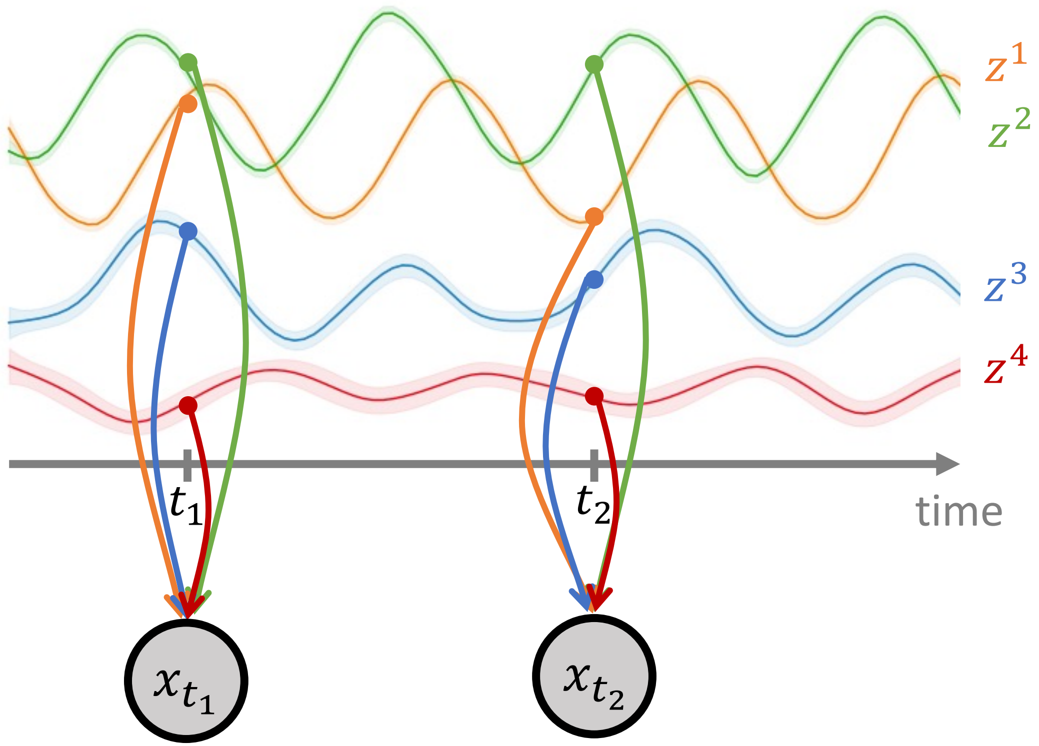


Linear-time Kalman filtering/smoothing for inference and learning!

Särkkä and Solin (2019). Applied stochastic differential equations. Cambridge University Press

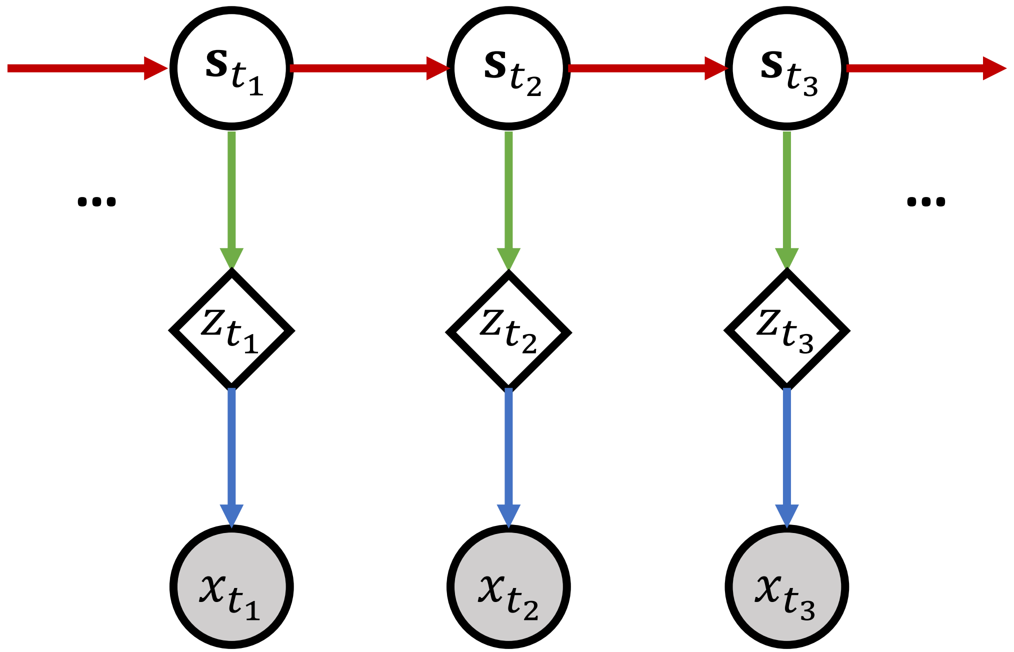
Markovian GP-VAE: Generative model

Continuous-time view



⇒

Discrete-time view & computation



Filtering/smoothing posterior no longer tractable!

(due to the use of neural network decoder for x)

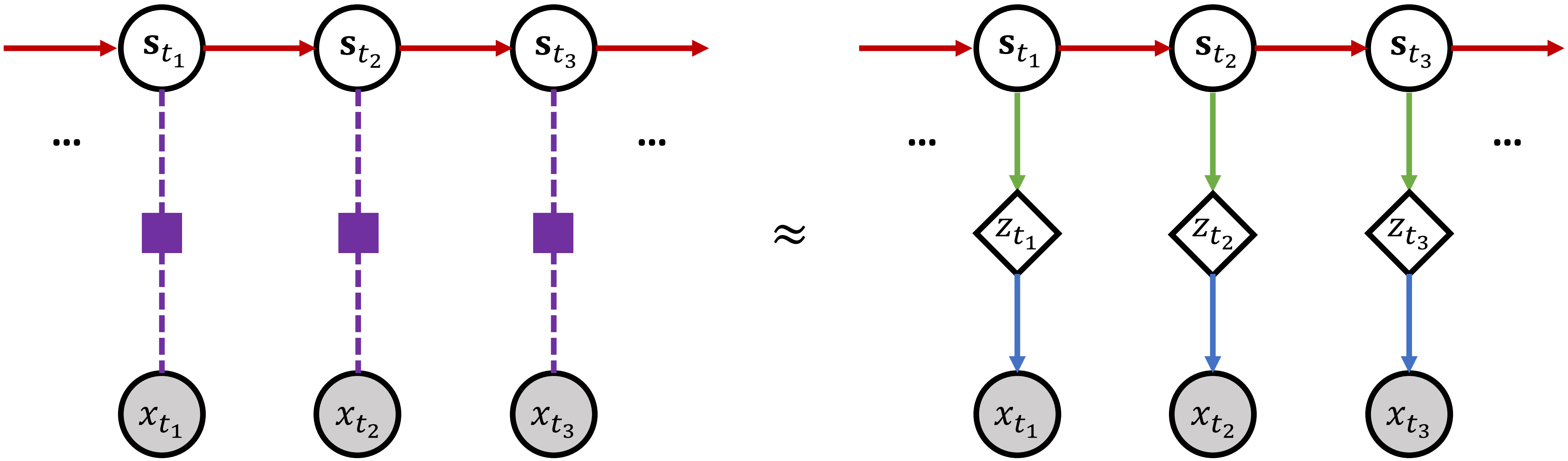
- Linear Gaussian transitions for s
- Noiseless linear emissions for z
- Neural network decoder for x

H Zhu, C Balsells Rodas and **Y Li.** Markovian Gaussian Process Variational Autoencoders. ICML 2023

Site Approximations

approximate "generative model" $\tilde{p}$, tractable posterior

generative model p, intractable posterior



- Linear Gaussian transitions for s
- Linear-Gaussian factor for connecting (s, x)

shared

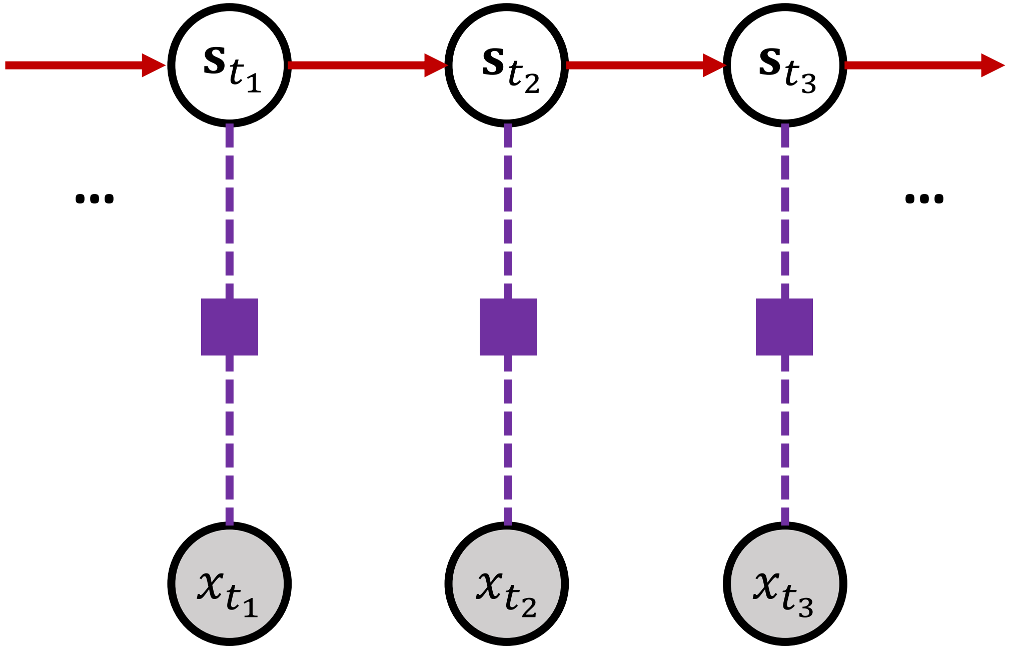
amortised this with $NN_\phi(x)$

- Linear Gaussian transitions for s
- Noiseless linear emissions for z
- Neural network decoder for x

Jazbec et al. Scalable gaussian process variational autoencoders. AISTATS 2021
H Zhu, C Balsells Rodas and **Y Li.** Markovian Gaussian Process Variational Autoencoders. ICML 2023

Site Approximations

approximate "generative model" $\tilde{p}$, tractable posterior



- Linear Gaussian transitions for s
- Linear-Gaussian factor for connecting (s, x)

Specifications for $\tilde{p}$: LG-SSM with pseudo targets

$$\tilde{p}(\{\tilde{x}_t\},\{s_t\}) = \prod_i p\left(s_{t_i} \middle| s_{t_{i-1}}\right) N(\tilde{x}_{t_i}; Hs_{t_i}, \tilde{V}_{t_i})$$

Amortised site approximation:

$N(\tilde{x}_t; Hs_t, \tilde{V}_t)$ with $(\tilde{x}_t, \tilde{V}_t) = NN_\phi(x_t)$

- Pseudo target $\tilde{x}_t$ construction:

Given x_t, find the best approximation

$$N\left(\tilde{x}_t; Hs_t, \tilde{V}_t\right) \approx p(x_t|s_t)$$

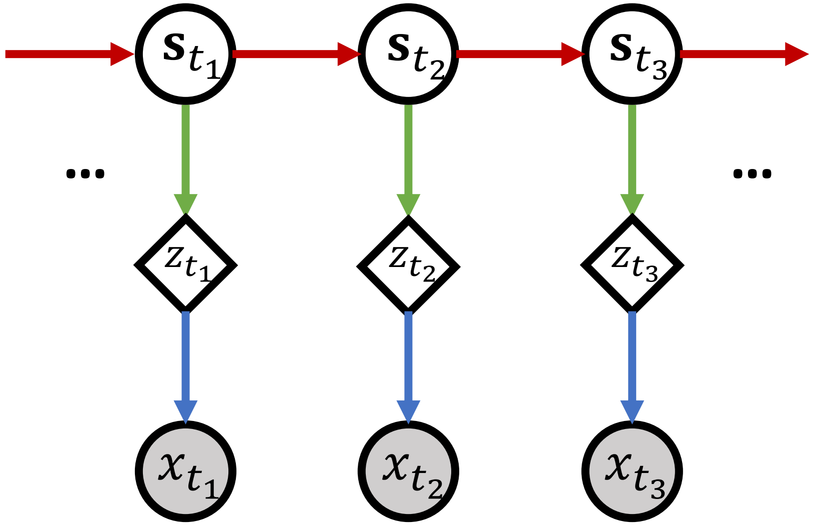
- Define the approximate posterior

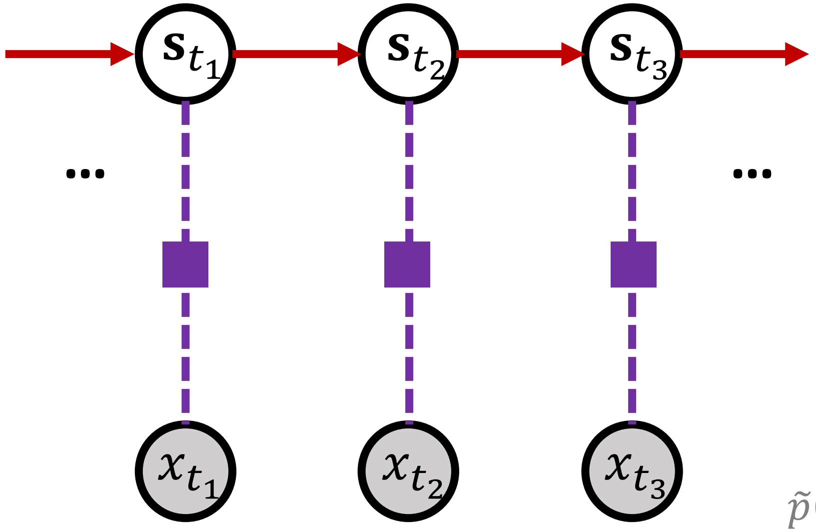
$$q(\{s_t\}|\{x_t\}) \propto \tilde{p}(\{\tilde{x}_t\},\{s_t\})$$

depends on $\{x_t\}$

Jazbec et al. Scalable gaussian process variational autoencoders. AISTATS 2021
H Zhu, C Balsells Rodas and **Y Li.** Markovian Gaussian Process Variational Autoencoders. ICML 2023

Site Approximations





$$p(\{x_t\},\{s_t\}) = \prod_i p\left(s_{t_i} \middle| s_{t_{i-1}}\right) p(x_{t_i}|s_{t_i})$$

$\tilde{p}(\{\tilde{x}_t\},\{s_t\})$

$$q(\{s_t\}|\{x_t\}) \propto \boxed{\prod_i p\left(s_{t_i} \middle| s_{t_{i-1}}\right) N(\tilde{x}_{t_i}; H s_{t_i}, \tilde{V}_{t_i})}$$

$$ELBO = E_{q(\{s_t\}|\{x_t\})}\left[\log \frac{p(\{x_t\},\{s_t\})}{q\left(\{s_t\}\middle|\{x_t\}\right)}\right] = E_{q(\{s_t\}|\{x_t\})}\left[\log \frac{\prod_i p\left(s_{t_i}\middle|s_{t_{i-1}}\right) p(x_{t_i}|s_{t_i})}{\prod_i p\left(s_{t_i}\middle|s_{t_{i-1}}\right) N(\tilde{x}_{t_i}; H s_{t_i}, \tilde{V}_{t_i})} + \log \tilde{p}(\{\tilde{x}_t\})\right]$$

$$= E_{q(\{s_t\}|\{x_t\})}\left[\sum_i \log \frac{p(x_{t_i}|s_{t_i})}{N(\tilde{x}_{t_i}; H s_{t_i}, \tilde{V}_{t_i})}\right] + \log \tilde{p}(\{\tilde{x}_t\}) = \sum_i E_{q(s_{t_i}|\{x_t\})}\left[\log \frac{p(x_{t_i}|s_{t_i})}{N(\tilde{x}_{t_i}; H s_{t_i}, \tilde{V}_{t_i})}\right] + \log \tilde{p}(\{\tilde{x}_t\})$$

LG-SSM smoothing

LG-SSM filtering

Jazbec et al. Scalable gaussian process variational autoencoders. AISTATS 2021

H Zhu, C Balsells Rodas and **Y Li.** Markovian Gaussian Process Variational Autoencoders. ICML 2023

Experimental Results – Rotating MNIST

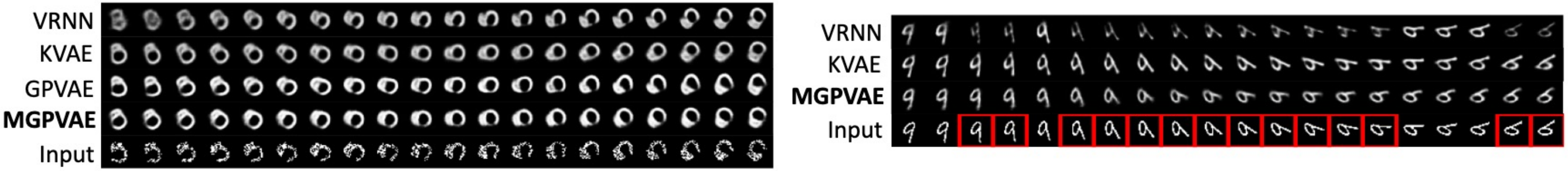
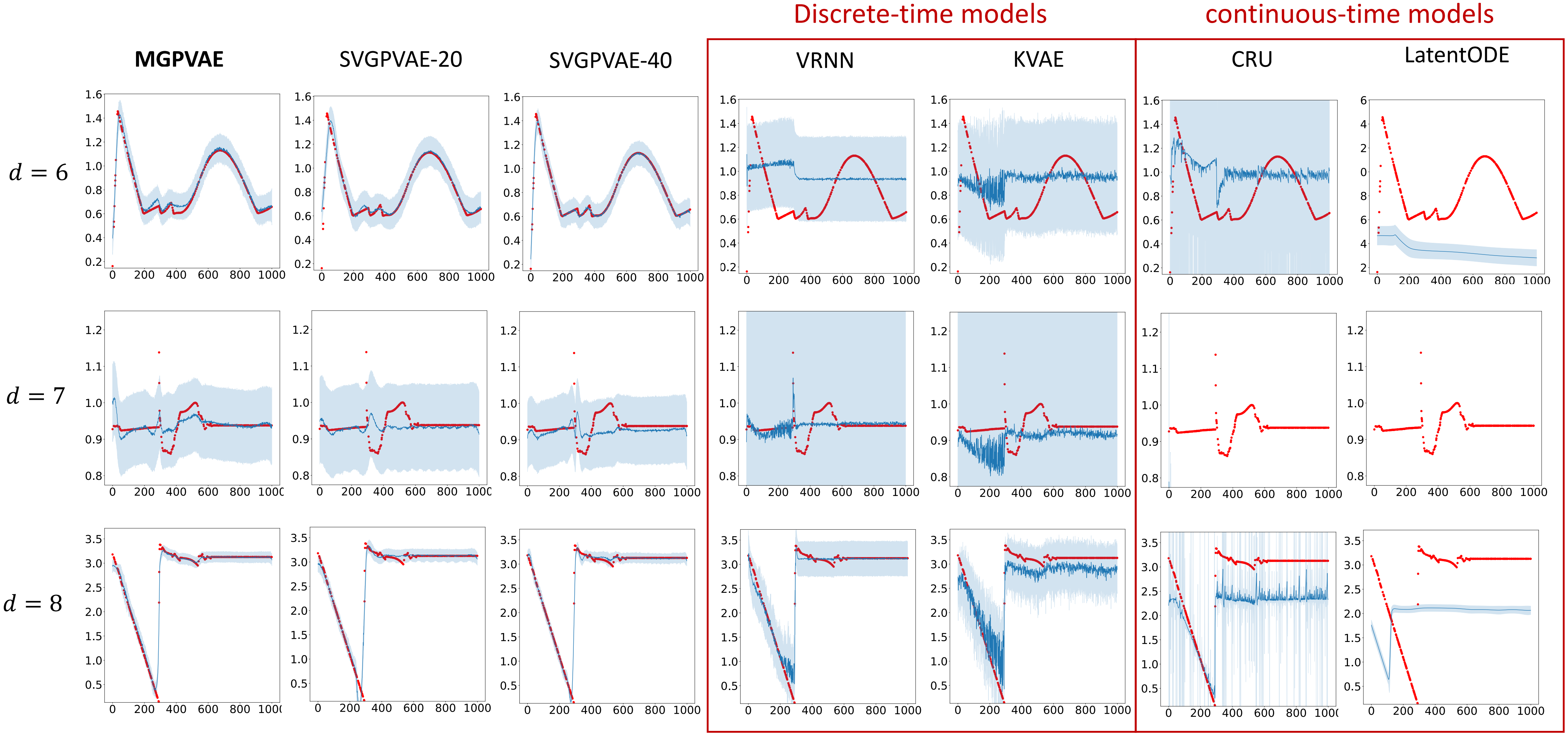


Figure 3: (Left) Corrupt frames imputation results for an unseen sequence of 5's. (Right) Missing frames imputation results for an unseen sequence of 9's. Missing frames are red frames.

Table 1: Test NLL and RMSE for both the corrupt (Cor) and missing frames (Mis) imputation tasks.

Model	NLL-Cor (↓)	RMSE-Cor (↓)	Time-Cor (s/epoch ↓)	NLL-Mis (↓)	RMSE-Mis (↓)	Time-Mis (s/epoch ↓)
VRNN	9898 ± 162.0	0.1768 ± 0.001563	63.51	16240 ± 2090	0.1796 ± 0.008002	103.6
KVAE	12500 ± 83.13	0.2025 ± 0.0006077	139.2	10730 ± 1232	0.1582 ± 0.008688	149.0
GPVAE	9026 ± 48.70	**0.1340 ± 0.0004529**	**48.93**	NA	NA	NA
MGPVAE	**8556 ± 69.66**	**0.1468 ± 0.0006738**	**50.45**	**8925 ± 53.40**	**0.1508 ± 0.0005190**	**59.43**

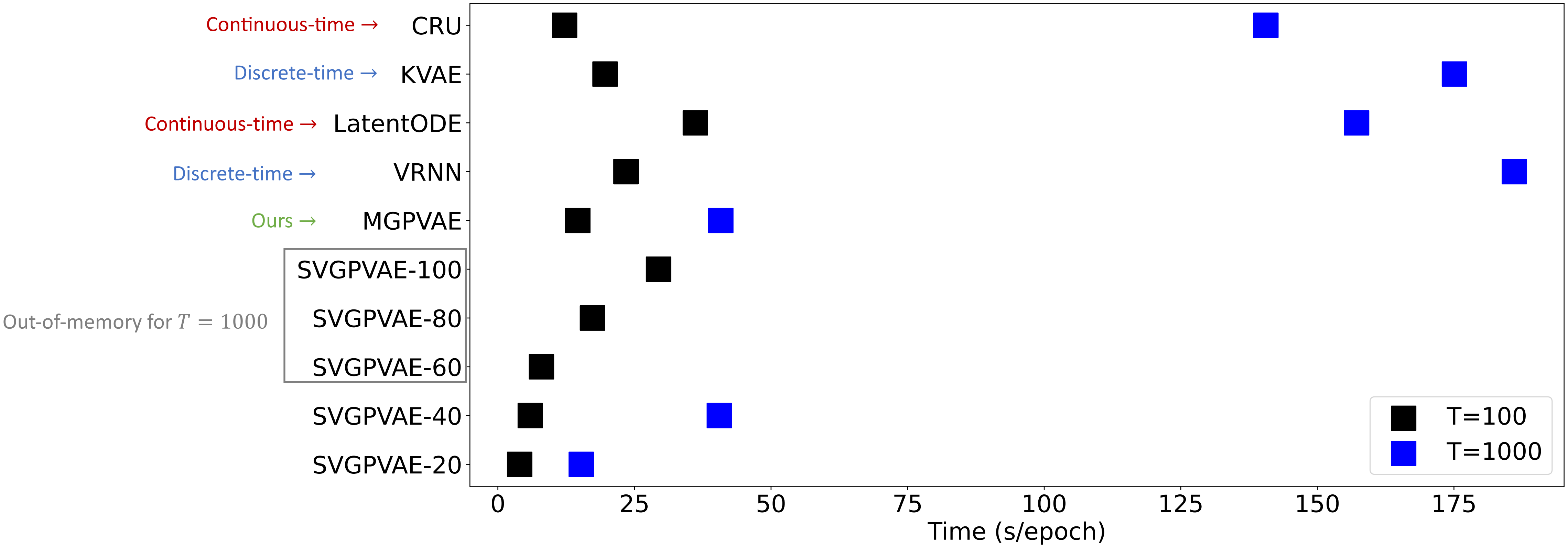
Experimental Results- Mujoco



H Zhu, C Balsells Rodas and **Y Li.** Markovian Gaussian Process Variational Autoencoders. ICML 2023

Experimental Results - Mujoco

Training run-time comparisons:

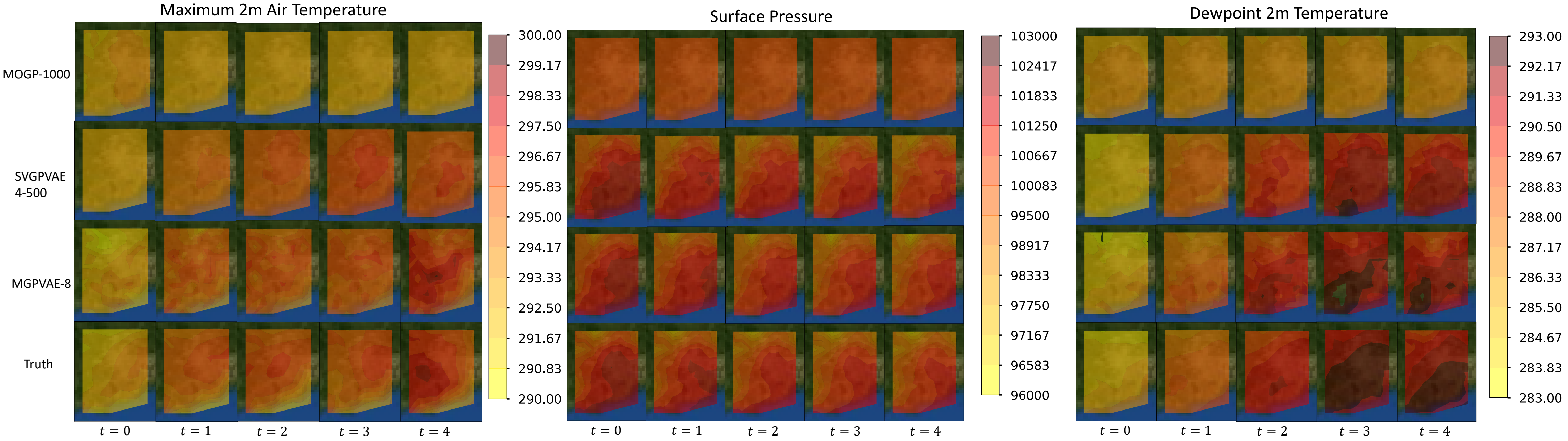


H Zhu, C Balsells Rodas and **Y Li.** Markovian Gaussian Process Variational Autoencoders. ICML 2023

Experimental Results – Climate Data

Spatial-temporal data: using product kernel for the GP prior

$$K\big((r,t),(r',t')\big) = K_{spatial}(r,r')K_{temporal}(t,t')$$

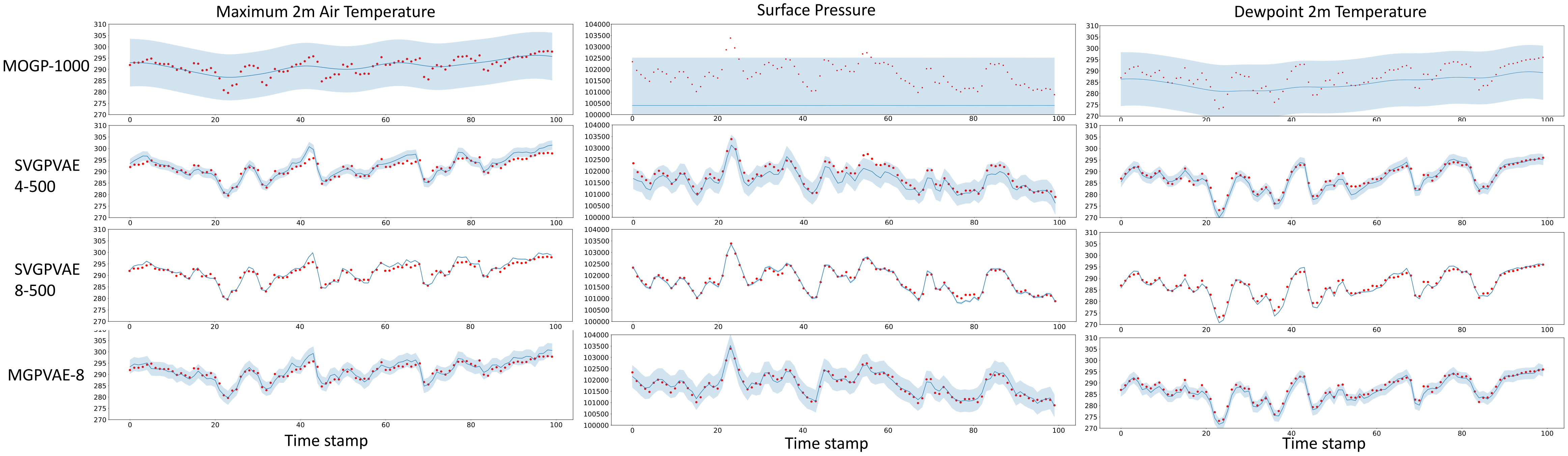


H Zhu, C Balsells Rodas and **Y Li.** Markovian Gaussian Process Variational Autoencoders. ICML 2023

Experimental Results – Climate Data

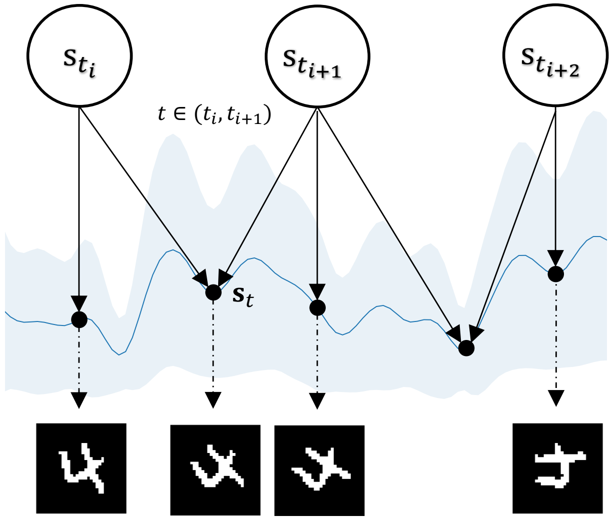
Spatial-temporal data: using product kernel for the GP prior

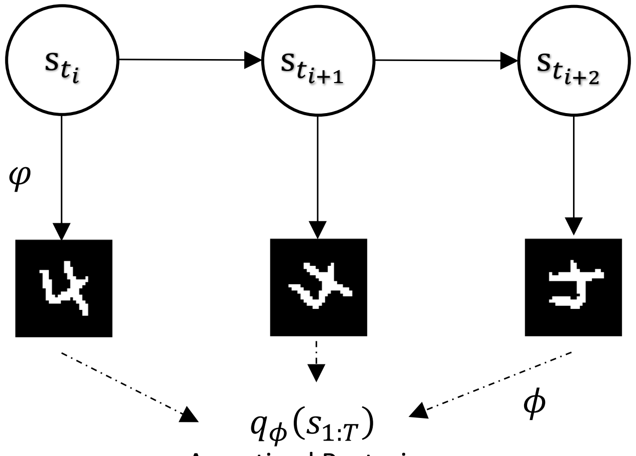
$$K\big((r,t),(r',t')\big) = K_{spatial}(r,r')K_{temporal}(t,t')$$



H Zhu, C Balsells Rodas and **Y Li.** Markovian Gaussian Process Variational Autoencoders. ICML 2023

Markovian GPVAE: Main Idea





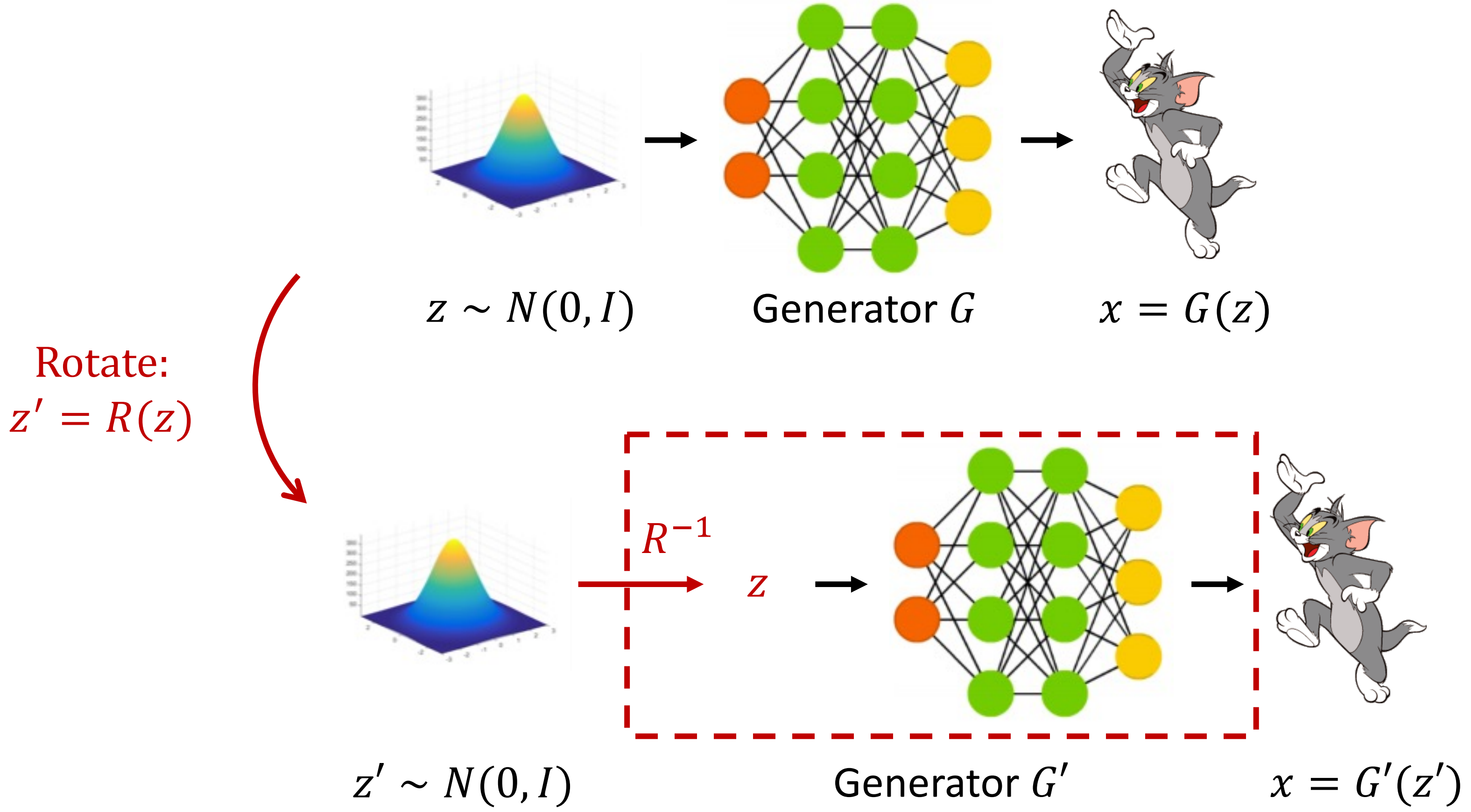
Summary:

- GPVAE: continuous-time sequence generative model with priors specified for global behaviour
- With Markovian kernel + site approximation, enabling linear-time deterministic computations
- Versatile: applications to video, physical simulation, and climate data

H Zhu, C Balsells Rodas and **Y Li.** Markovian Gaussian Process Variational Autoencoders. ICML 2023

Switching Dynamics & Identifiability

Motivation: Representation Learning



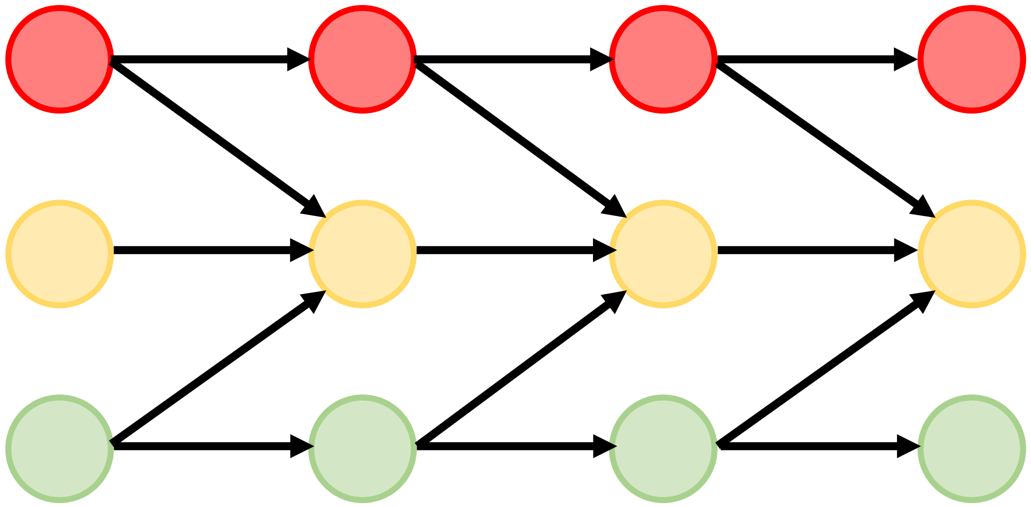
Same results!
(non-identifiable)

Locatello et al. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. ICML 2019.

Motivation: Causal Discovery in Time-Series

Use the information of time: “the cause happens prior to its effect”

- Granger causality, TiMINo, etc.:
  - Assume all the variables are observed
  - In most cases assume stationarity



Peters et al. Causal inference on time series using restricted structural equation models. NIPS 2013
Tank et al. Neural Granger Causality. IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 44, no. 08, pp. 4267-4279, 2022.

State-Dependent Causal Inference (SDCI)

Causal discovery & sequence modelling for non-stationary time series:

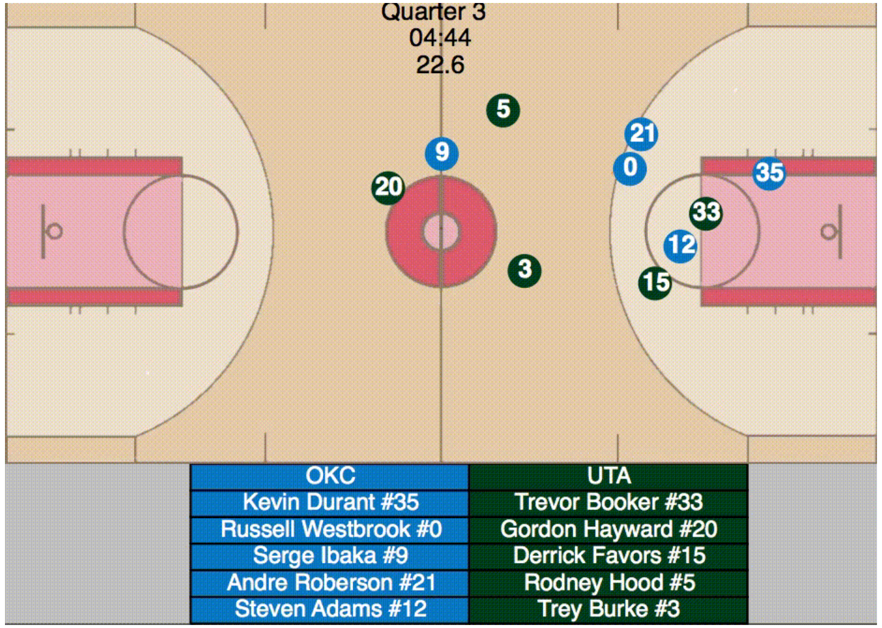
- Imagine having N agents interacting:
  - Each agent i at time step t has both its observation x_i^t and its internal discrete state s_i^t
  - Depending on the state s_i^t, x_i^t will have different functional relationship with x_j^{t+1}
- Conditional summary graph:
  - Compact summary of the causal relationship
  - When the states are all fixed to the same: reduced back to summary graph

C Balsells Rodas, R Tu, **Y Li and** H Kjellstrom. Causal Discovery from Conditionally Stationary Time Series. UAI 2022 Causal Representation Learning Workshop

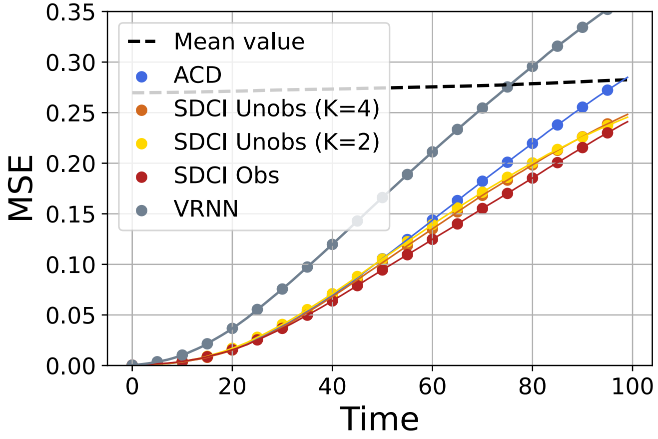
State-Dependent Causal Inference (SDCI)

Causal discovery & sequence modelling for non-stationary time series:

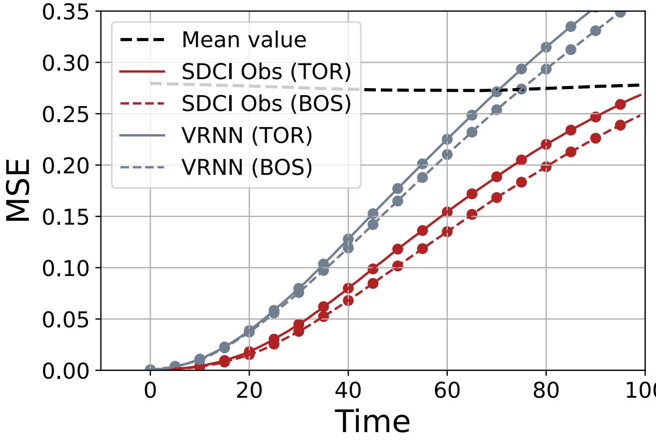
Dataset: NBA player trajectories
- multi-agent
- non-stationary



Forecasting error:



Train on full data



Train on Boston Celtics only

Learned hidden state visualisation:

C Balsells Rodas, R Tu, **Y Li and** H Kjellstrom. Causal Discovery from Conditionally Stationary Time Series. UAI 2022 Causal Representation Learning Workshop

State-Dependent Causal Inference (SDCI)

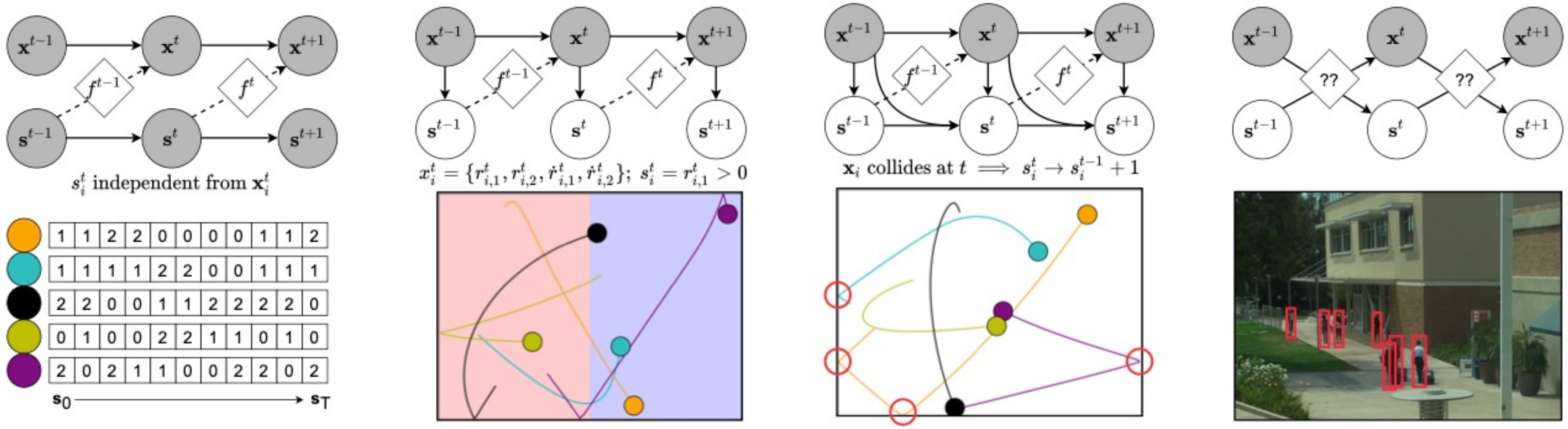
Identifiability result for SDCI (informal):

The conditional summary graph is identifiable if the states are observed.

(not realistic)

Can we do better?

Yes, but need assumptions on how the observations and states interact



C Balsells Rodas, R Tu, **Y Li and** H Kjellstrom. Causal Discovery from Conditionally Stationary Time Series. UAI 2022 Causal Representation Learning Workshop

On the Identifiability of Markov Switching Models

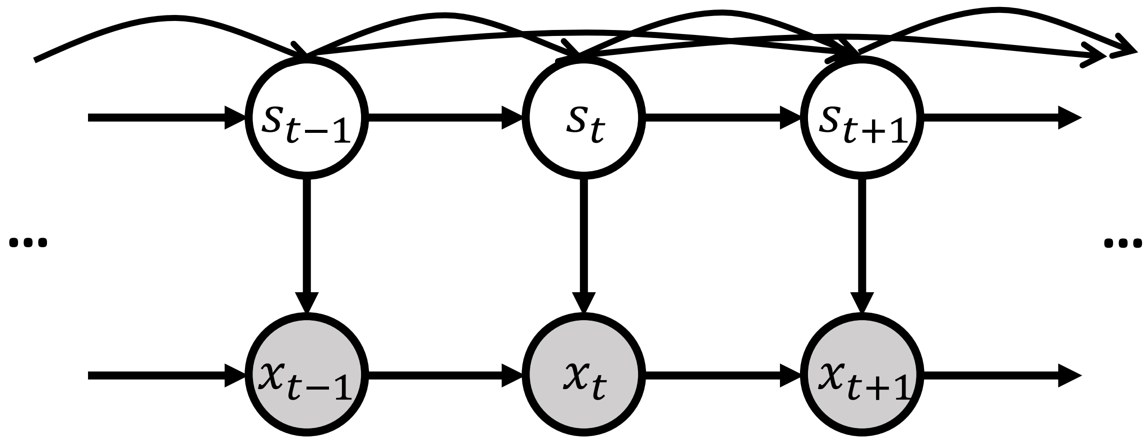
ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modelling

Carles Balsells Rodas[1] Yixin Wang[2] Yingzhen Li[1]

[1]Imperial College London [2]University of Michigan

Identifiability in Switching Dynamic Models

Markov Switching Models (first-order):

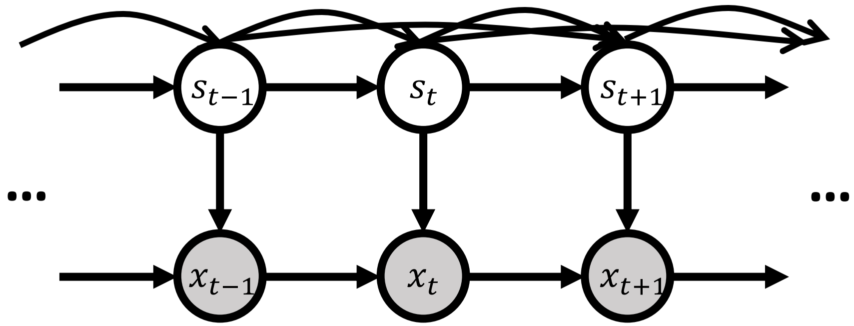


- Discrete and finite state-space: $s_t \in \{1, \dots, K\}$
- Conditional first-order Markov model: $p(x_t | x_{<t}, s_t) = p(x_t | x_{t-1}, s_t)$ (assuming $x_0 = \emptyset$)

When does this model identifiable with observations of $x_{1:T}$ only?

C Balsells Rodas, Y Wang and **Y Li.** On the identifiability of Markov Switching Models. ICML 2023 workshop

Identifiability in Switching Dynamic Models



Identifiability result (informal):

The first-order Markov Switching Model is identifiable up to state permutation when:

- *Unique indexing for the states (i.e., no repeating states):*

$$i \neq j \;\Leftrightarrow\; p(x_t|x_{t-1}, s_t = i) \neq p(x_t|x_{t-1}, s_t = j)$$

- *In Gaussian case, the mean and covariance functions are analytic in x_{t-1}:*

$$p(x_t|x_{t-1}, s_t) = N(x_t; \underline{m(x_{t-1}, s_t), S(x_{t-1}, s_t)})$$

Can use neural networks with smooth activation functions!
(here identifiability means identifying the functions)

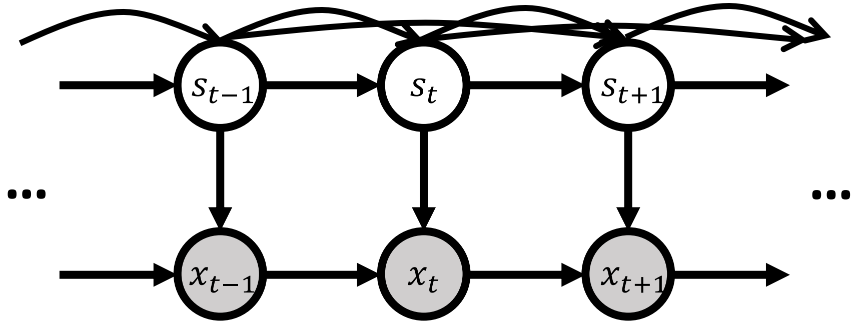
C Balsells Rodas, Y Wang and **Y Li.** On the identifiability of Markov Switching Models. ICML 2023 workshop

Identifiability in Switching Dynamic Models

Proof sketch (informal):

Think about it as a finite mixture model over paths:

$$p(x_{1:T}) = \sum_{s_{1:T} \in \{1,\ldots,K\}^T} p(x_{1:T}|s_{1:T})p(s_{1:T})$$



(1) Identifiability for finite mixture model requires linear independence of family $\{p(x_{1:T}|s_{1:T})\}$

(2) Notice the first-order Markov structure: $p(x_{1:T}|s_{1:T}) = \prod_{t=1}^{T} p(x_t|x_{t-1}, s_t)$

⇒ Show linear independence of $p(x_{1:2}|s_{1:2})$, then prove for $T \geq 3$ case by induction

(3) Work out conditions on $p(x_t|x_{t-1}, s_t)$ to make $\{p(x_t|x_{t-1}, s_t)\, p(x_{t+1}|x_t, s_{t+1})\}$ linearly independent

⇒ Obtain certain linear independence & continuity conditions in non-parametric case

(4) In Gaussian case: work out the conditions on the mean & covariance to satisfy conditions in (3)

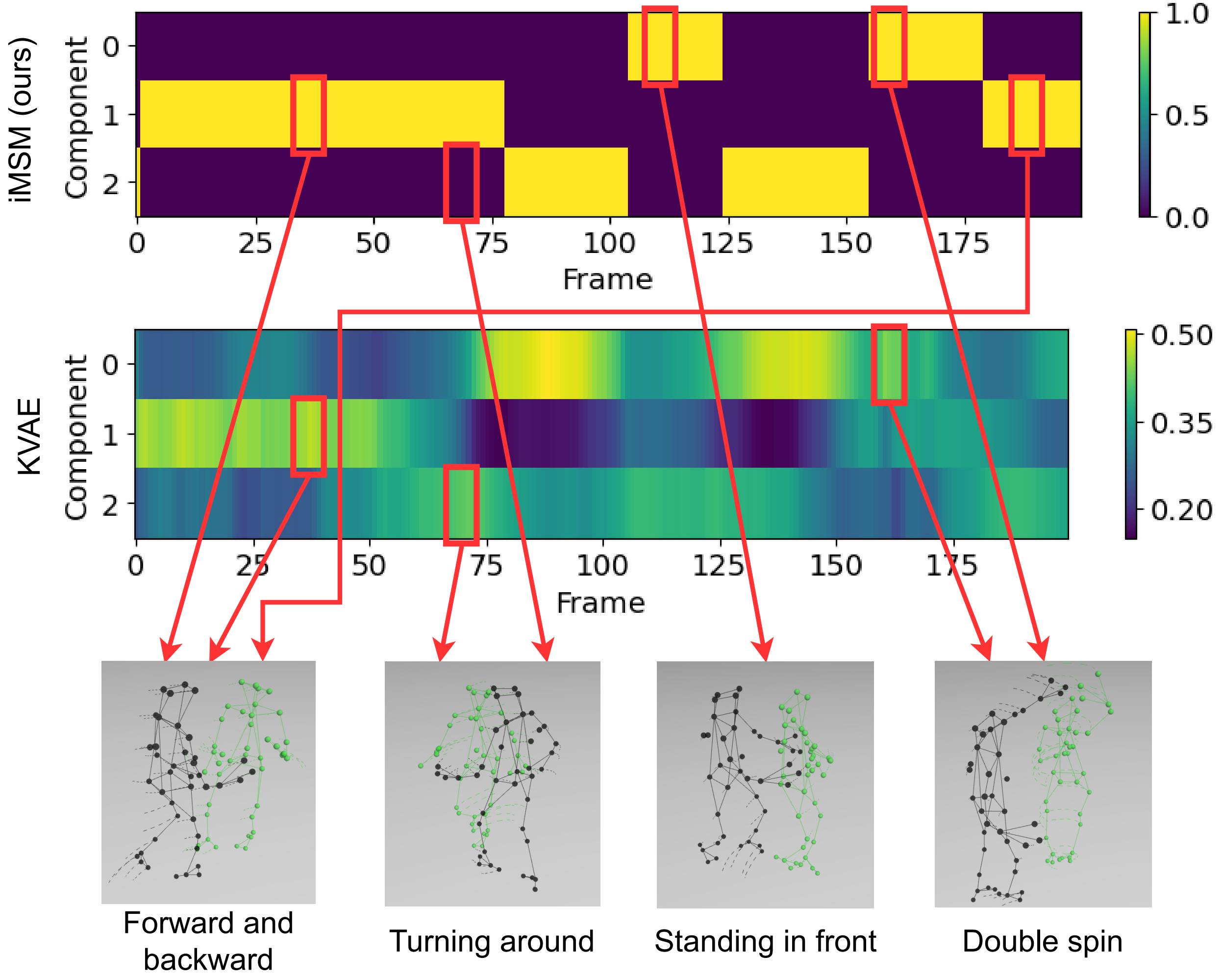
$$p(x_t|x_{t-1}, s_t) = N(x_t; \underline{m(x_{t-1}, s_t), S(x_{t-1}, s_t)})$$

⇒ Analytic in x_{t-1}

C Balsells Rodas, Y Wang and **Y Li.** On the identifiability of Markov Switching Models. ICML 2023 workshop

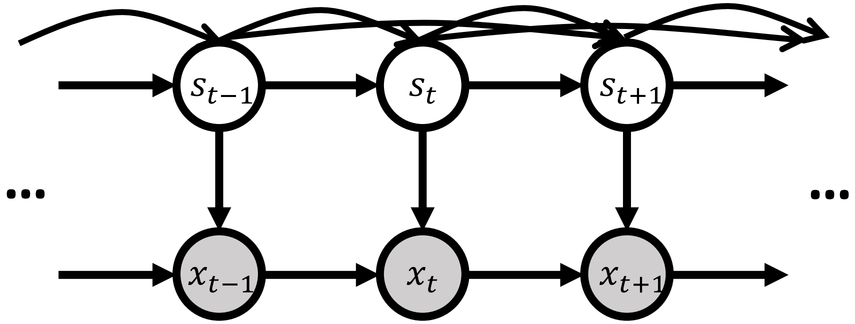
Identifiability in Switching Dynamic Models

- Experiment: discovering dancing patterns
  - Data: CMU mocap
  - DL Baseline: KalmanVAE



Fraccaro et al. A Disentangled Recognition and Nonlinear Dynamics Model for Unsupervised Learning. NeurIPS 2017
C Balsells Rodas, Y Wang and **Y Li.** On the identifiability of Markov Switching Models. ICML 2023 workshop

Some Discussions



On the proof strategy and indications:

- Cannot use the proof strategy of HMM identifiability results
  - Simply because the dynamic is not fully controlled by latent state transitions
- The proof makes NO assumption on $p(s_{1:T})$ and can identify the joint $p(s_{1:T})$
  - Works for ANY dynamic model for the states $s_{1:T}$
  - The marginal $p(x_{1:T})$ can thus be non-stationary and higher-order Markov
  - Direct extension to global regime settings by making $s_1 = s_2 = \cdots = s_T$
- Easily extendable to include observed "control signals" $u_{1:T}$:

$$p(x_{1:T}, s_{1:T} | u_{1:T}) = p(x_{1:T} | s_{1:T}) p(s_{1:T} | u_{1:T})$$

Gassiat et al. Inference in finite state space non parametric hidden Markov models and applications. Stat Comput 26, 61–71, 2016
Allman et al. Identifiability of parameters in latent structure models with many observed variables. Ann. Stat. 37, 3099–3132, 2009

Some Discussions

Future extensions:

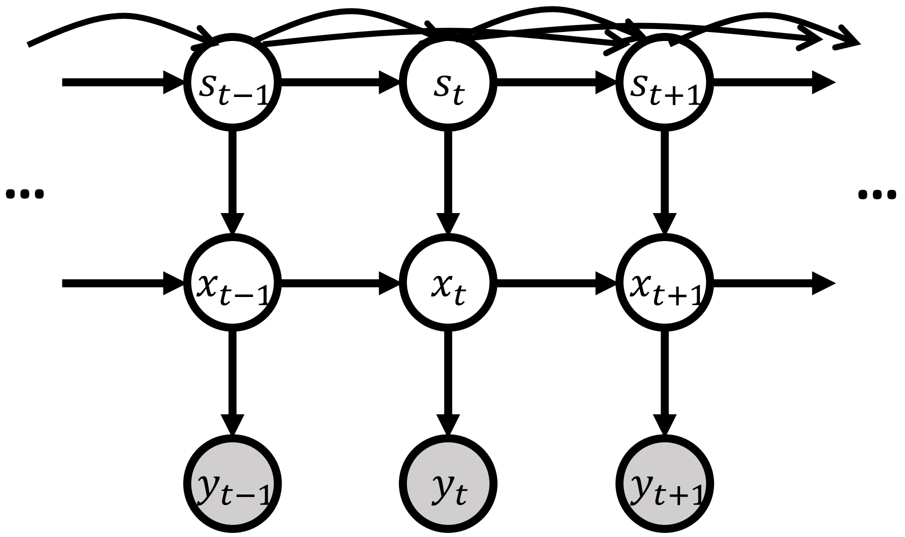
- Go for higher-order Markov conditional transitions (with time lag $M > 1$):

$$p(x_t | x_{<t}, s_t) = p(x_t | x_{t-M:t-1}, s_t)$$

  - Better assumptions for e.g., neuron activity data, energy & climate time-series

- Lift the continuous states $x_{1:T}$ to latent space:
  - More realistic for video & other high-dimensional data
  - Potential application in model-based RL

- Beyond time series?



Fraccaro et al. A Disentangled Recognition and Nonlinear Dynamics Model for Unsupervised Learning. NeurIPS 2017
Hafner et al. Mastering Atari with Discrete World Models. ICLR 2021

Take Home Messages Today

- Sequence data generation: far away from being solved!
- Methods that explicitly model the underlying dynamics:
  - Deterministic vs stochastic dynamics
  - Discrete-time vs continuous-time dynamics
- Recent trend: continuous-time dynamic model that has efficient discrete-time computation/approximation
  - Markovian GPVAE
  - S4 & CRU
- Causal representation learning for time-series
  - Sequential generative model very promising here

Gu et al. Efficiently Modeling Long Sequences with Structured State Spaces. ICLR 2022
Schirmer et al. Modeling Irregular Time Series with Continuous Recurrent Units. ICML 2022

THANK YOU!

Questions? Ask now, or email:
yingzhen.li@imperial.ac.uk

Thanks to my awesome collaborators:


Harrison Zhu


Carles Balsells Rodas


Yixin Wang


Ruibo Tu


Hedvig Kjellström


Stephan Mandt