Segmenting Hybrid Trajectories Using Latent ODEs

Ruian (Ian) Shi

University of Toronto

Vector Institute

Quaid Morris

Memorial Sloan Kettering Cancer Center

Vector Institute

University of Toronto

Computer Science UNIVERSITY OF TORONTO

VECTOR INSTITUTE

Memorial Sloan Kettering Cancer Center

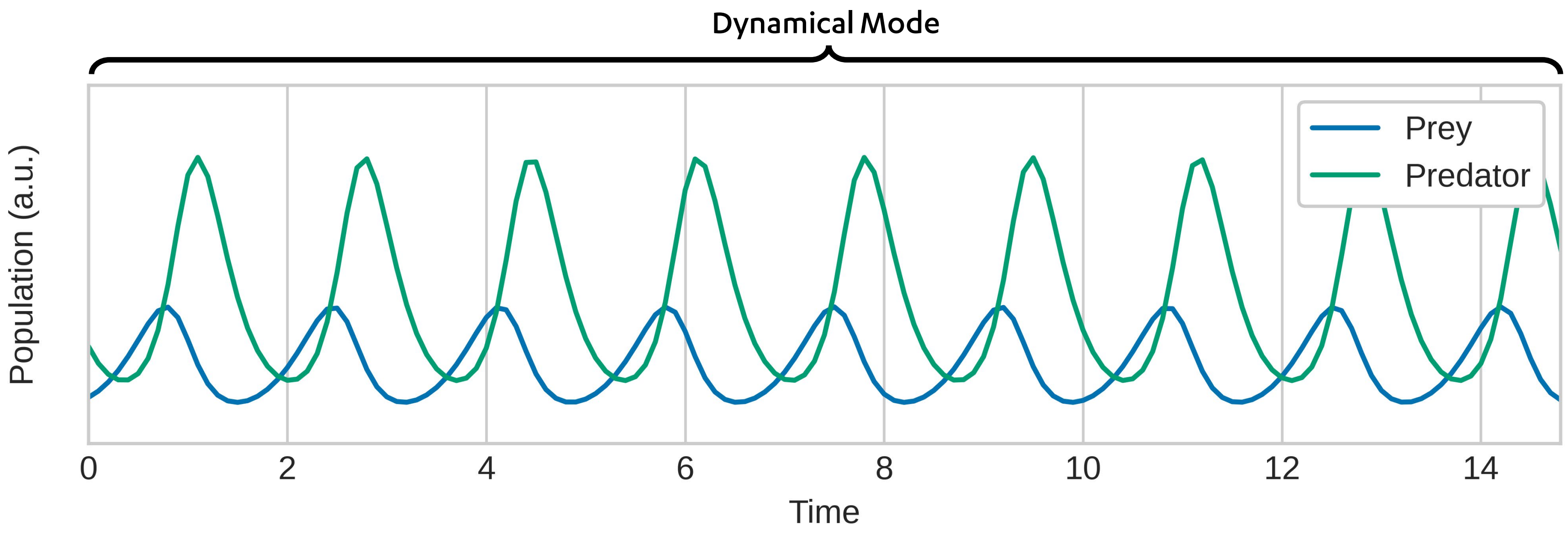
LatSegODE: Reconstructing and Segmenting Hybrid Trajectories

- The Latent Segmented ODE (**LatSegODE**) performs **reconstruction** and **segmentation** in **hybrid trajectories**

LatSegODE: Reconstructing and Segmenting Hybrid Trajectories

- The Latent Segmented ODE (**LatSegODE**) performs **reconstruction** and **segmentation** in **hybrid trajectories**
- Hybrid trajectories feature distinct dynamical modes separated by discontinuous jumps
  - Example: Lotka-Volterra hybrid trajectory

$$\frac{dx}{dt} = \boldsymbol{\alpha} x - \boldsymbol{\beta} xy \qquad \frac{dy}{dt} = \boldsymbol{\delta} xy - \boldsymbol{\gamma} y$$



LatSegODE: Reconstructing and Segmenting Hybrid Trajectories

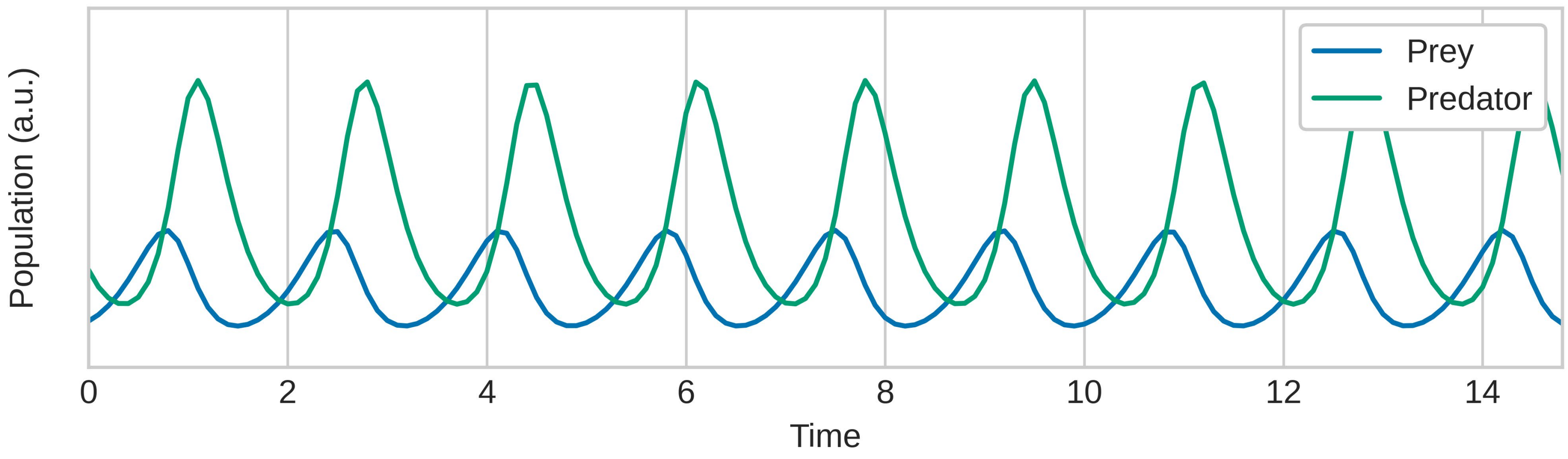
- The Latent Segmented ODE (**LatSegODE**) performs **reconstruction** and **segmentation** in **hybrid trajectories**

- Hybrid trajectories feature distinct dynamical modes separated by discontinuous jumps
  - Example: Lotka-Volterra hybrid trajectory

$$\frac{dx}{dt} = \boldsymbol{\alpha} x - \boldsymbol{\beta} xy \qquad \frac{dy}{dt} = \boldsymbol{\delta} xy - \boldsymbol{\gamma} y$$

Dynamical Coefficients

Dynamical Mode

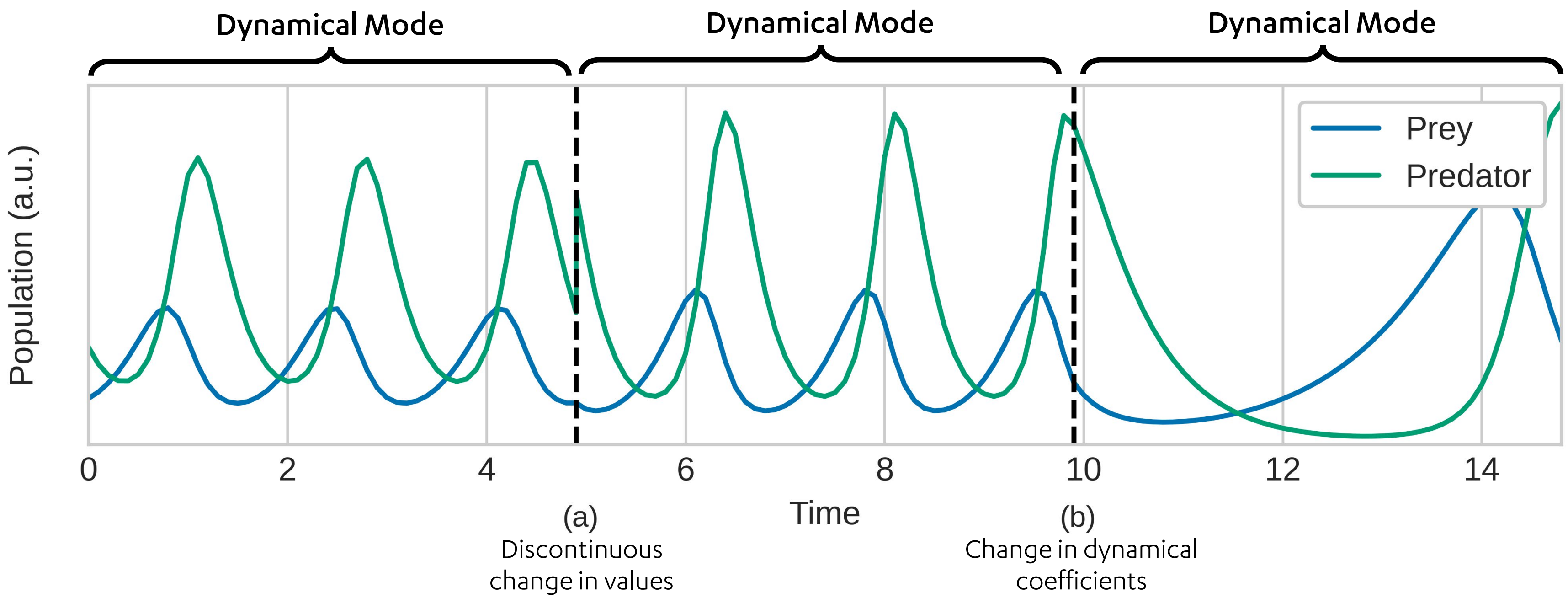


LatSegODE: Reconstructing and Segmenting Hybrid Trajectories

- The Latent Segmented ODE (**LatSegODE**) performs **reconstruction** and **segmentation** in **hybrid trajectories**
- Hybrid trajectories feature distinct dynamical modes separated by discontinuous jumps
  - Example: Lotka-Volterra hybrid trajectory

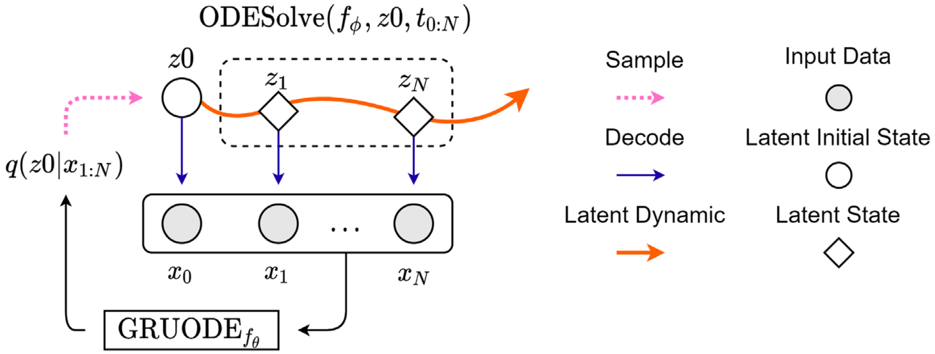
$$\frac{dx}{dt} = \boldsymbol{\alpha} x - \boldsymbol{\beta} xy \qquad \frac{dy}{dt} = \boldsymbol{\delta} xy - \boldsymbol{\gamma} y$$

Dynamical Coefficients



Baseline Approaches

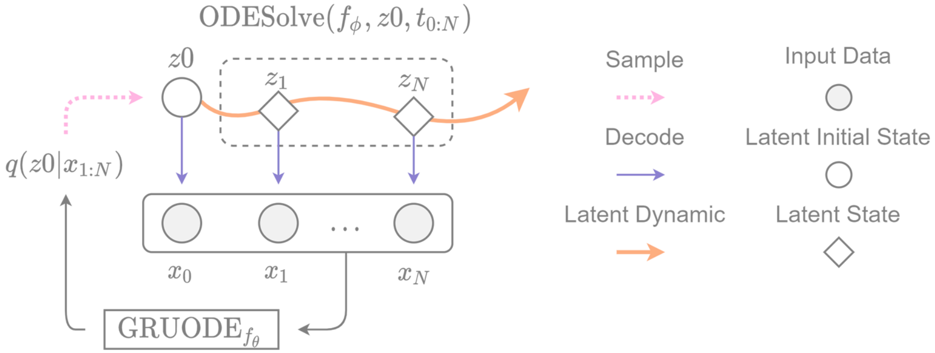
- The Latent ODE [1] is recent powerful model which combines Neural ODE in VAE setup.

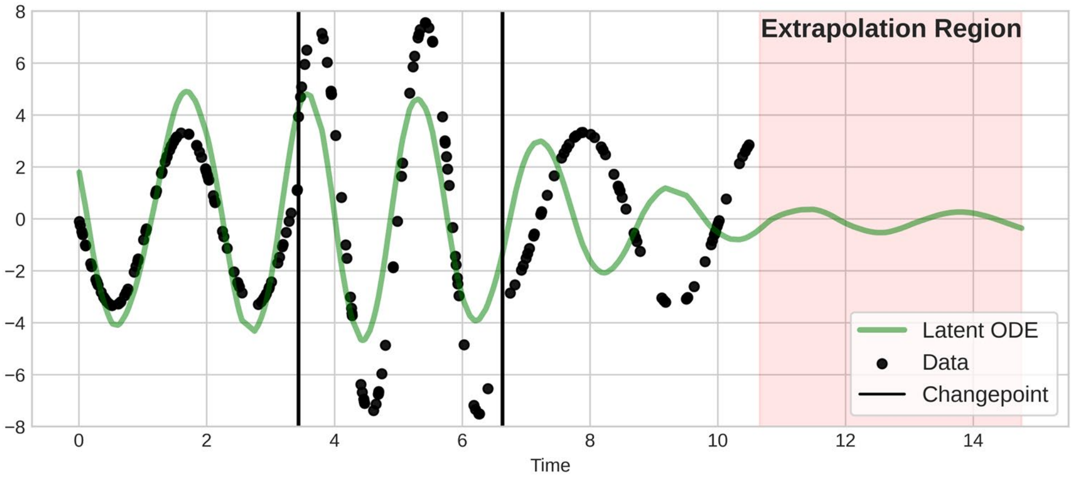


1. https://arxiv.org/abs/1907.03907

Baseline Approaches

- The Latent ODE [1] is recent powerful model which combines Neural ODE in VAE setup.



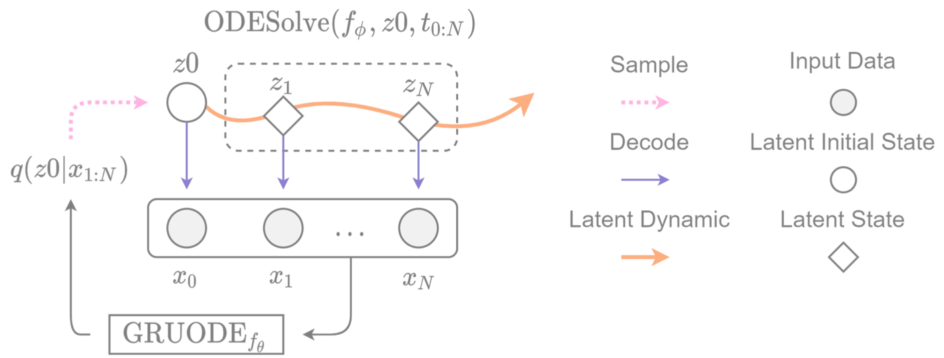


- Latent ODE performs poorly on complex families of hybrid trajectories

1. https://arxiv.org/abs/1907.03907

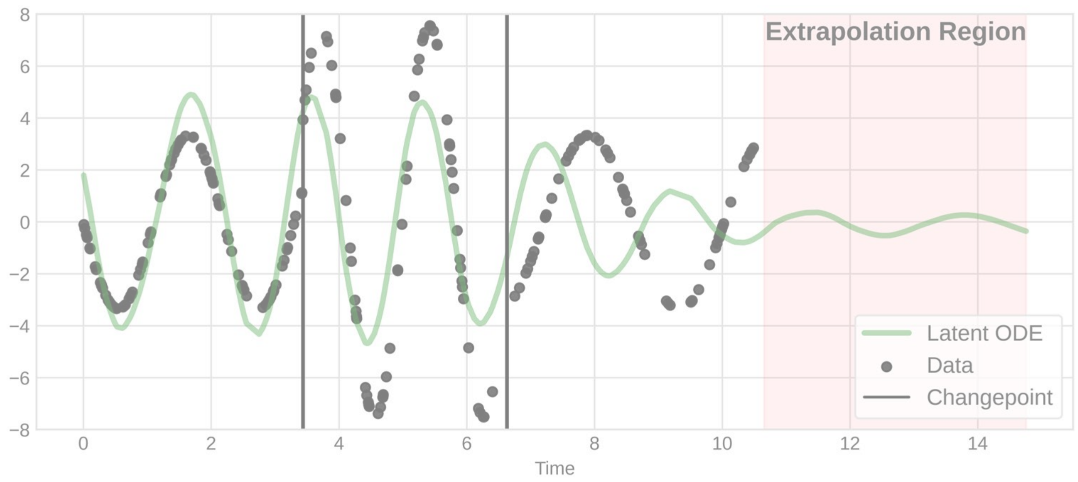
Baseline Approaches

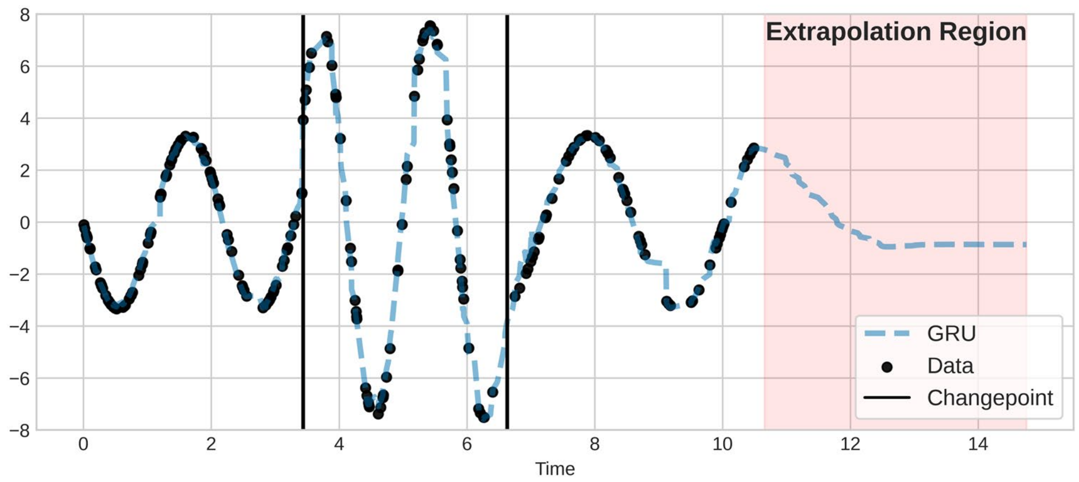
- The Latent ODE [1] is recent powerful model which combines Neural ODE in VAE setup.



- Latent ODE performs poorly on complex families of hybrid trajectories

- Classical deep time series methods (GRU [2]) perform poorly when extrapolating

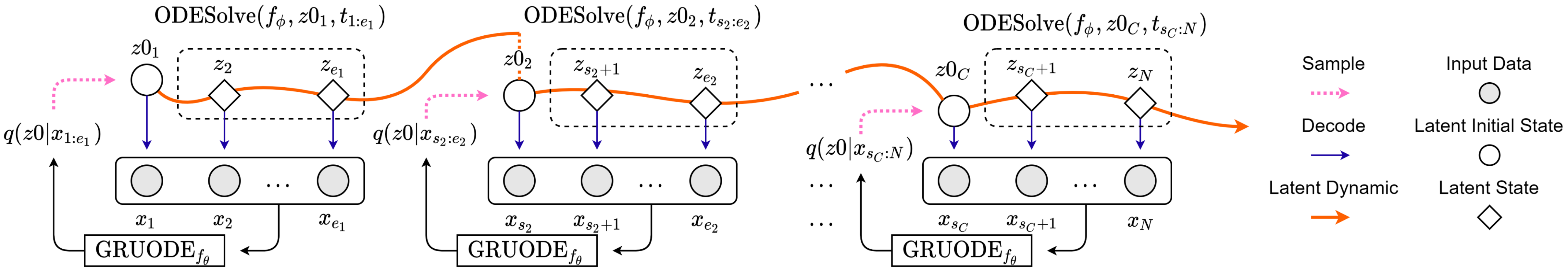




1. https://arxiv.org/abs/1907.03907 2. https://arxiv.org/abs/1406.1078

LatSegODE

- Represent complex hybrid trajectory as piece-wise sequence of simple Latent ODE dynamics

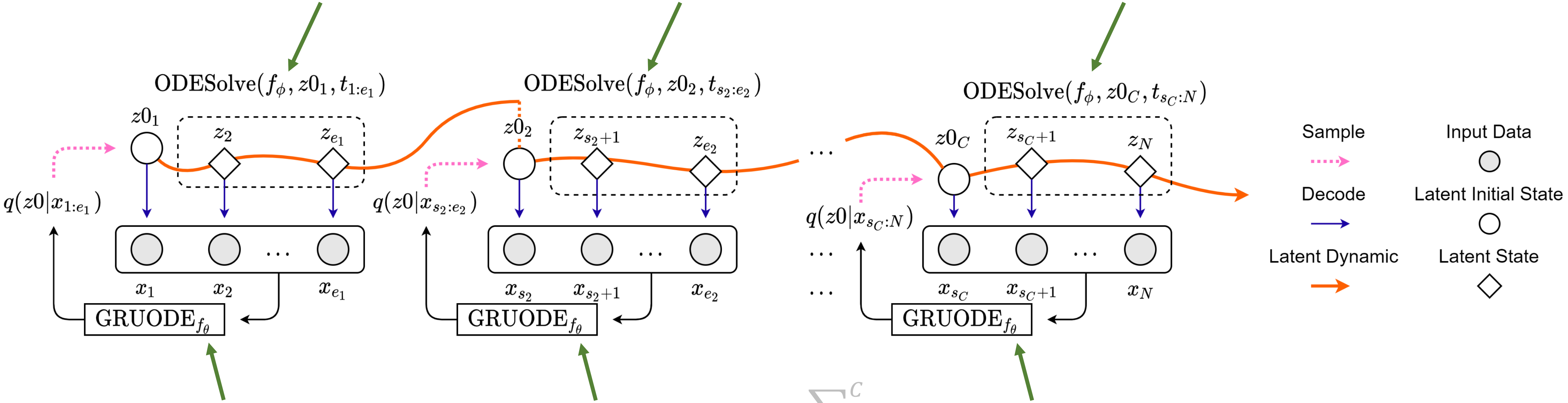


$$\log p(X, Z \mid s_{1:C}, e_{1:C}) = \sum_{i=0}^{C} \log p(x_{s_i:e_i}, z0_i)$$

LatSegODE

- Represent complex hybrid trajectory as piece-wise sequence of simple Latent ODE dynamics

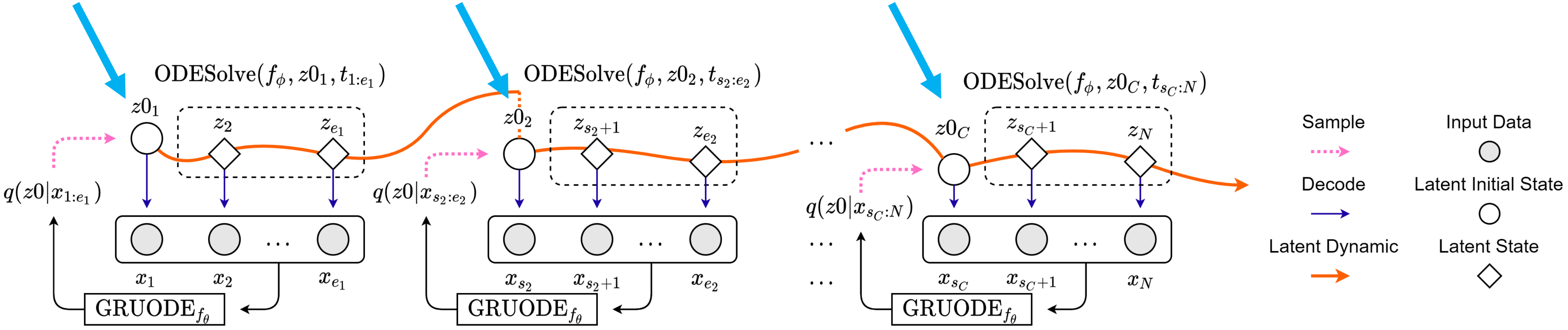
- Share Latent ODE parameters across all segments



$$\log p(X, Z \mid s_{1:C}, e_{1:C}) = \sum_{i=0}^{C} \log p(x_{s_i:e_i}, z0_i)$$

LatSegODE

- Represent complex hybrid trajectory as piece-wise sequence of simple Latent ODE dynamics
- Share Latent ODE parameters across all segments
- **Start at new latent initial position ($z0$) per segment**



$$\log p(X, Z \mid s_{1:C}, e_{1:C}) = \sum_{i=0}^{C} \log p(x_{s_i:e_i}, z0_i)$$

Segmentation Penalization with Bayesian Occam's Razor

- Use PELT [1] algorithm to search through exponential space of all possible changepoints

$$\mathcal{O}(2^N) \rightarrow O(N) \text{ to } O(N^2)$$

1. https://arxiv.org/abs/1101.1438

Segmentation Penalization with Bayesian Occam's Razor

- Use PELT[1] algorithm to search through exponential space of all possible changepoints

$$\mathcal{O}(2^N) \rightarrow O(N) \text{ to } O(N^2)$$

1. https://arxiv.org/abs/1101.1438

- Classical changepoint detection uses penalty such as Bayesian Information Criterion

$$\text{BIC} = k \log(n) - 2 \log(p(x \mid \theta, M))$$

Segmentation Penalization with Bayesian Occam's Razor

- Use PELT[1] algorithm to search through exponential space of all possible changepoints

$$\mathcal{O}(2^N) \rightarrow O(N) \text{ to } O(N^2)$$

1. https://arxiv.org/abs/1101.1438

- Classical changepoint detection uses penalty such as Bayesian Information Criterion

$$\text{BIC} = k \log(n) - 2 \log(p(x \mid \theta, M))$$

  - Difficult to estimate model complexity (k)

Segmentation Penalization with Bayesian Occam's Razor

- Use PELT[1] algorithm to search through exponential space of all possible changepoints

$$\mathcal{O}(2^N) \rightarrow O(N) \text{ to } O(N^2)$$

1. https://arxiv.org/abs/1101.1438

- Classical changepoint detection uses penalty such as Bayesian Information Criterion

$$\text{BIC} = k \log(n) - 2 \log(p(x \mid \theta, M))$$

  - Difficult to estimate model complexity (k)

- LatSegODE uses marginal likelihood over latent initial states as measure of fit

$$\log p(x_{s:e}) = \log \int p(x_{s:e} | z0)\, p(z0)\, \mathrm{d}z0$$

$$= \mathbb{E}_{z0 \sim q(z0|x_{s:e})} \left[p(x_{s:e}|z0) \frac{p(z0)}{q(z0 \mid x_{s:e})} \right]$$

Segmentation Penalization with Bayesian Occam's Razor

- Use PELT [1] algorithm to search through exponential space of all possible changepoints

$$\mathcal{O}(2^N) \rightarrow O(N) \text{ to } O(N^2)$$

1. https://arxiv.org/abs/1101.1438

- Classical changepoint detection uses penalty such as Bayesian Information Criterion

$$\mathrm{BIC} = k\log(n) - 2\log(p(x\,|\theta, M))$$

  - Difficult to estimate model complexity (k)

- LatSegODE uses marginal likelihood over latent initial states as measure of fit

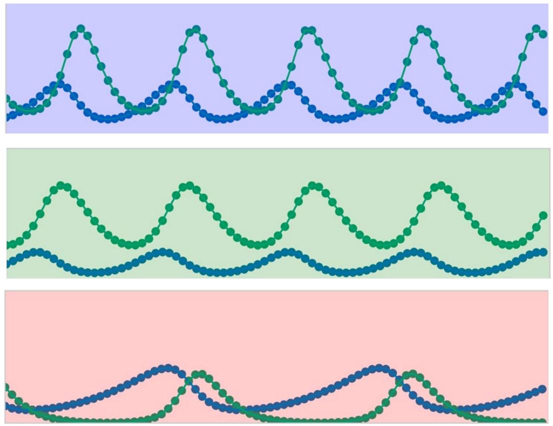
$$\log p(x_{s:e}) = \log \int p(x_{s:e}|z0)\, p(z0)\, \mathrm{d}z0$$

$$= \mathbb{E}_{z0\sim q(z0|x_{s:e})}\left[p(x_{s:e}|z0)\frac{p(z0)}{q(z0\,|x_{s:e})}\right]$$

- Marginal likelihood automatically regularized through Bayesian Occam's Razor

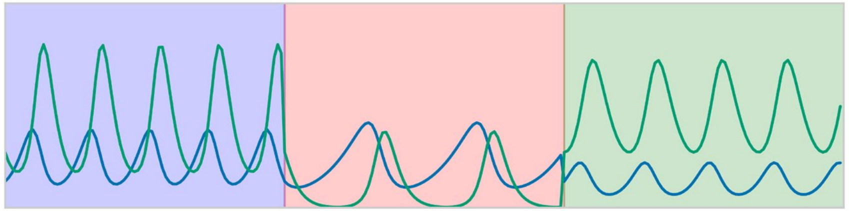
Training the LatSegODE Base Model

Simple Dynamical Primitives

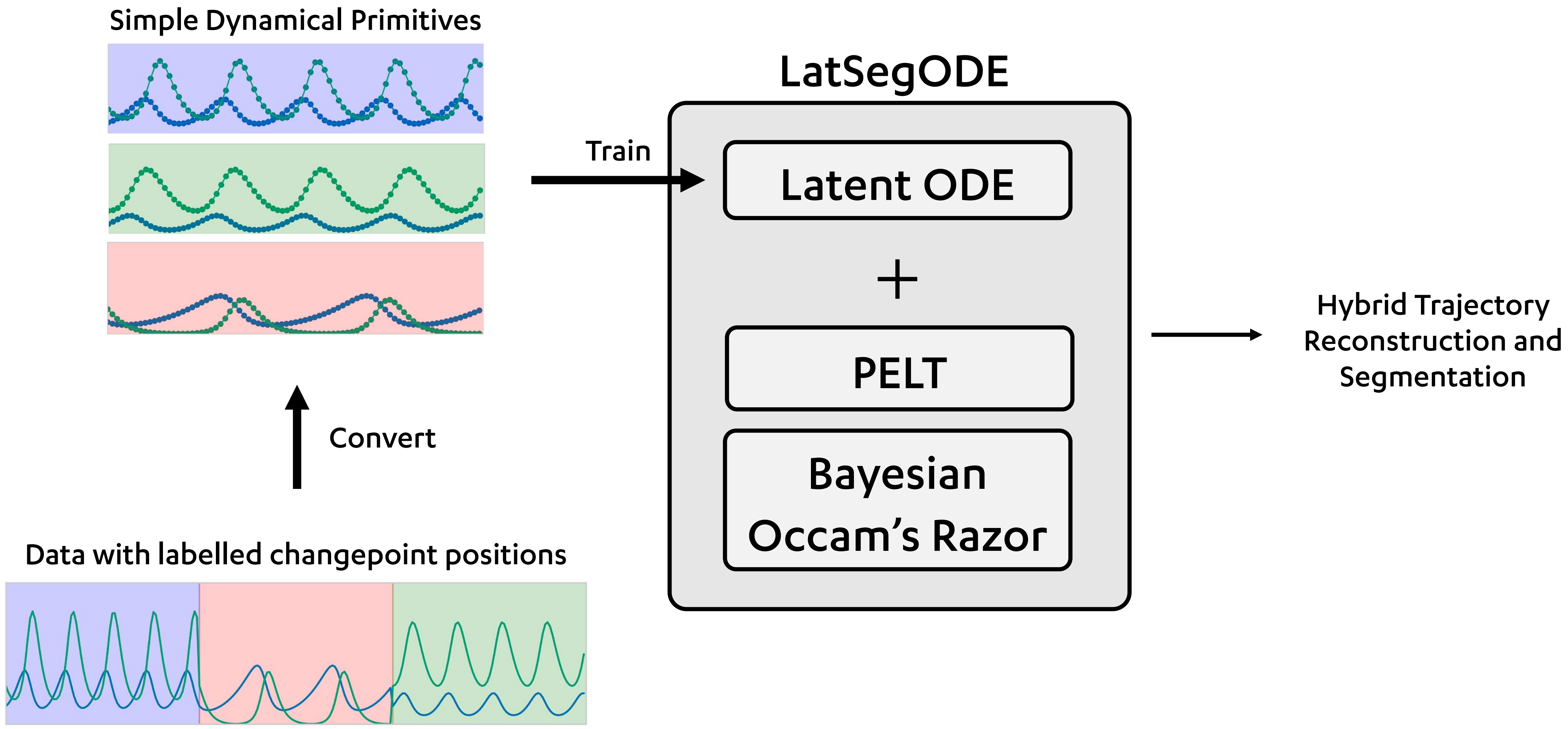


Convert

Data with labelled changepoint positions

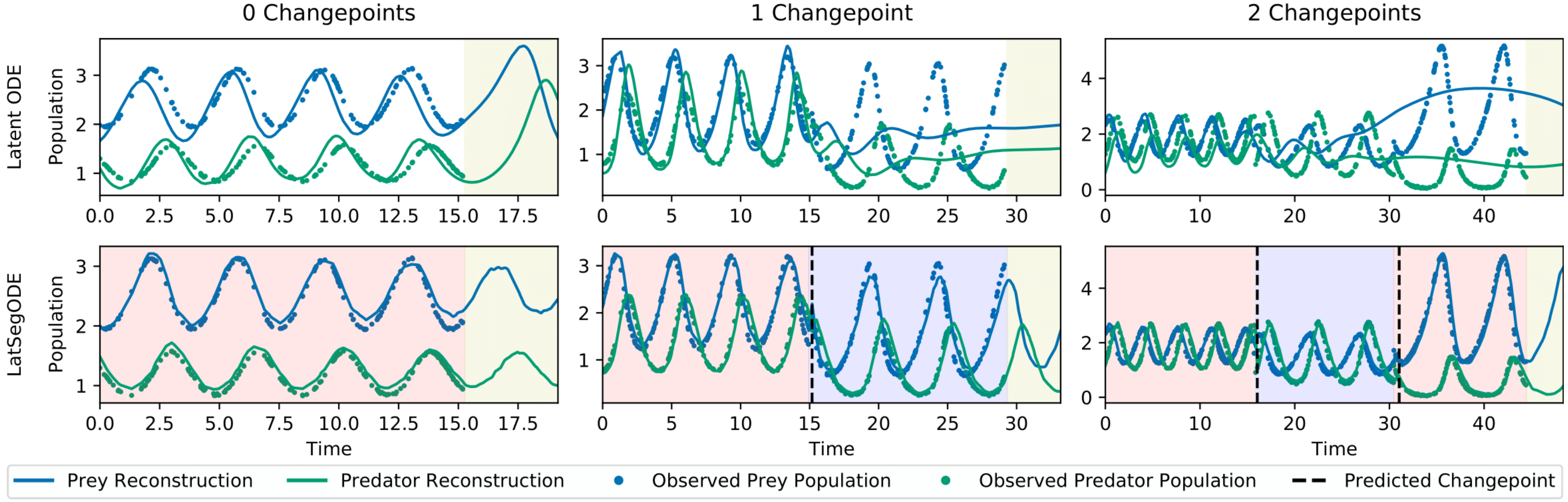


Training the LatSegODE Base Model



Results

- Experiment: Lotka-Volterra hybrid trajectories with up to 3 segments, randomly sampled coefficients.



Method	Test MSE	Rand Index	Hausdorff Metric
LatSegODE	**0.068**	**0.9464**	**47.67**
GRUΔt	0.1718	-	-
GRU-ODE	0.2747	-	-
Latent ODE	0.6155	-	-
RPT-RBF	-	0.7956	84.7
RPT-AR	-	0.6994	164.65
RPT-NORM	-	0.7693	105.92

- Latent ODE time series segmentation of hybrid trajectories
- High accuracy reconstruction and changepoint detection

Thanks!

Future Directions

- Semi-supervised or unsupervised training procedures
- Integration of alternative base models

Come visit our poster!