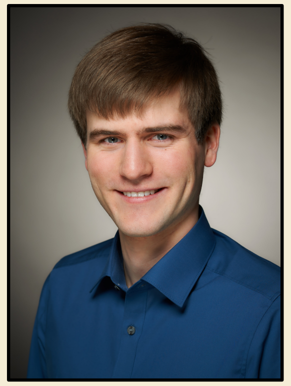
Neural Information Processing Systems 2024

Neural Persistence Dynamics


Sebastian Zeng[†/‡], Florian Graf[†], Martin Uray[†/‡], Stefan Huber[‡], Roland Kwitt[†]

[†]University of Salzburg, Austria
[‡]Josef Ressel Centre for Intelligent and Secure Industrial Automation
University of Applied Sciences, Salzburg, Austria

Neural Persistence Dynamics   @rkwitt1982   Sebastian Zeng

Motivation

- Observation of a coherently moving flock of birds, understood as an evolving 3D point cloud $\mathcal{P} = \{\mathbf{x}^k\}_{k=1}^{K}$:

*proprietary data (2024).

Motivation

- Observation of a coherently moving flock of birds, understood as an evolving 3D point cloud $\mathcal{P} = \{\mathbf{x}^k\}_{k=1}^K$:

- The `D'Orsogna` model [D'Orsogna et al. '06] describes the dynamics of individual entities $\mathbf{x}^k$

$$m\ddot{\mathbf{x}}^k = \left(\alpha - \beta\|\dot{\mathbf{x}}^k\|^2\right)\dot{\mathbf{x}}^k - \frac{1}{K}\nabla_{\mathbf{x}^k}\sum_{l\neq k}\underbrace{U\left(\|\mathbf{x}^k - \mathbf{x}^l\|, C_r, l_r\right)}_{\text{Attraction \& Repulsion}}.$$

*proprietary data (2024).

Motivation

- Observation of a coherently moving flock of birds, understood as an evolving 3D point cloud $\mathcal{P} = \{\mathbf{x}^k\}_{k=1}^{K}$:

*

- The D'Orsogna model [D’Orsogna et al. ’06] describes the dynamics of individual entities $\mathbf{x}^k$

$$m\ddot{\mathbf{x}}^k = \left(\alpha - \beta \|\dot{\mathbf{x}}^k\|^2\right) \dot{\mathbf{x}}^k - \frac{1}{K} \nabla_{\mathbf{x}^k} \sum_{l \neq k} \underbrace{U\left(\|\mathbf{x}^k - \mathbf{x}^l\|, C_r, l_r\right)}_{\text{Attraction \& Repulsion}} .$$

- Solving the **inverse problem**, i.e., predicting $\boldsymbol{\beta} = (m, \alpha, C_r, l_r)$, is inherently difficult due to:
    - the large number of observed entities, and
    - the difficulty of identifying individual motion trajectories $\mathbf{x}^k(t)$.

*proprietary data (2024).

Motivation

- Observation of a coherently moving flock of birds, understood as an evolving 3D point cloud $\mathcal{P} = \{\mathbf{x}^k\}_{k=1}^K$:

- The `D'Orsogna` model [D'Orsogna et al. '06] describes the dynamics of individual entities $\mathbf{x}^k$

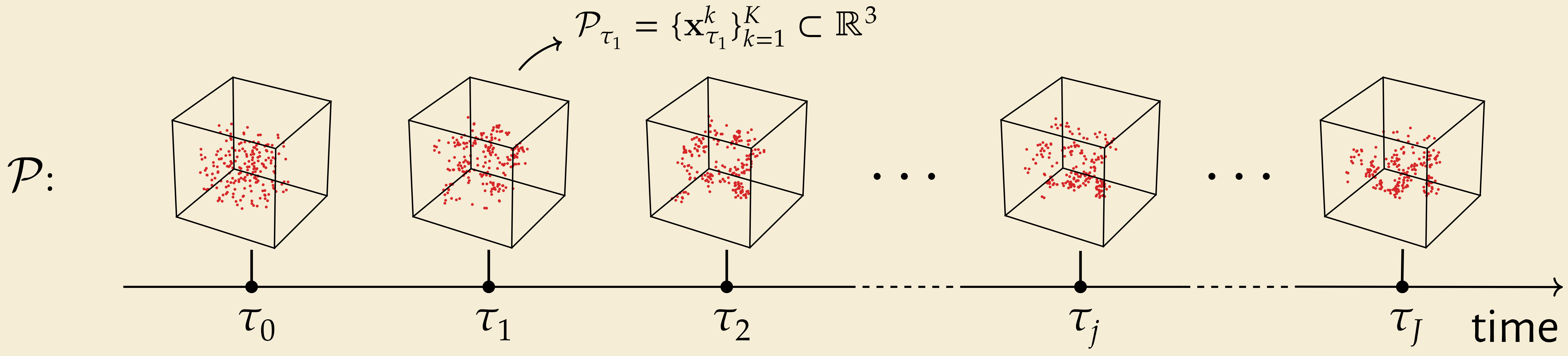
- To predict the **models parameters** β, understanding the evolving **behavioral patterns** of a collective is key.
- Hence, we learn the **dynamics in the topology** of time evolving **point clouds**.

- Solving th
  - the large number of observed entities, and
  - the difficulty of identifying individual motion trajectories $\mathbf{x}^k(t)$.

*proprietary data (2024).

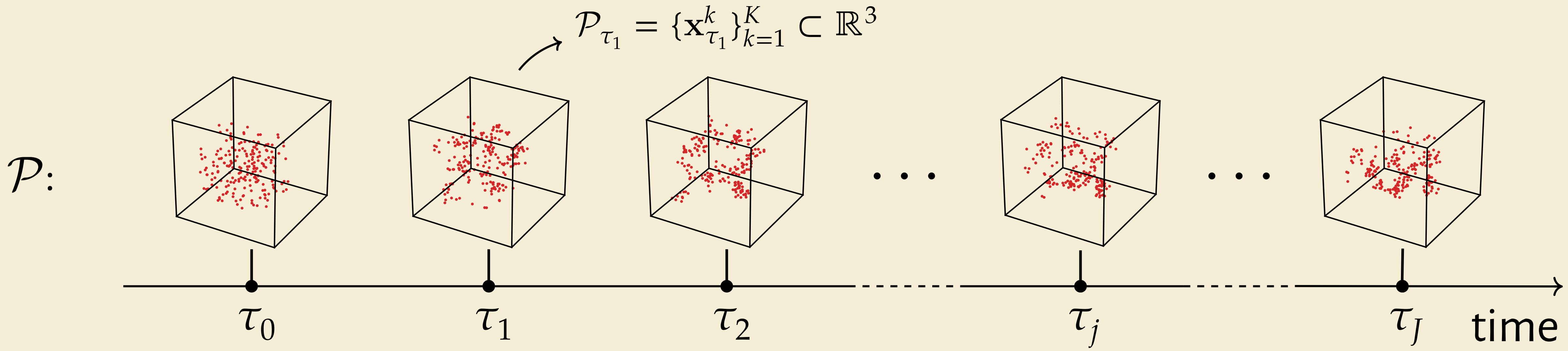
Problem setting

- For learning, we consider datasets of N time-evolving 3D point clouds $\mathcal{P}^1, \ldots, \mathcal{P}^N$; e.g.,



Problem setting

- For learning, we consider datasets of N time-evolving 3D point clouds $\mathcal{P}^1, \ldots, \mathcal{P}^N$; e.g.,



- We assume...
  (1) individual trajectories of points $\mathbf{x}^k$ are governed by a coupled equation of motion

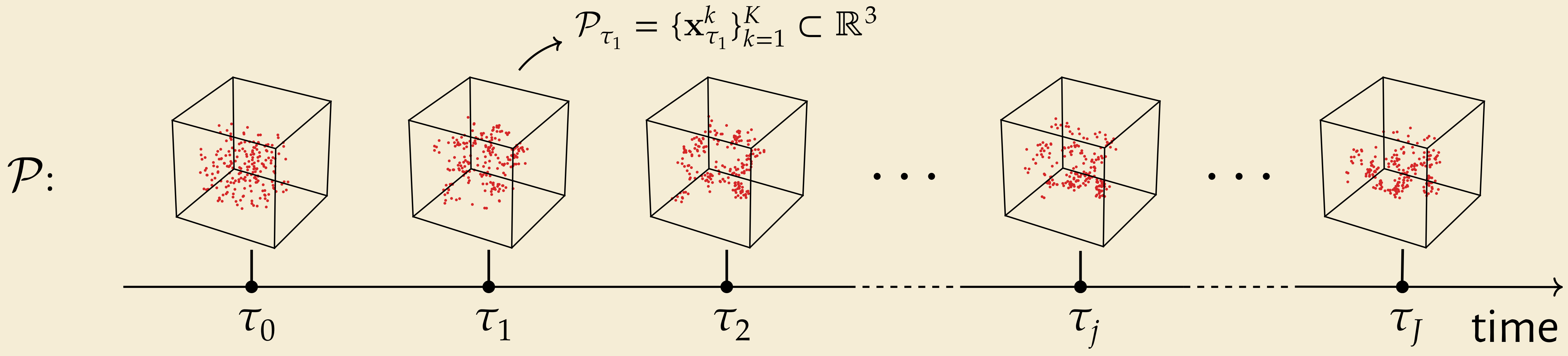
$$\ddot{\mathbf{x}}^k = f_{\boldsymbol{\beta}}\left(\{\mathbf{x}^l\}_{l=1}^K, \dot{\mathbf{x}}^k\right) ,$$

  (2) $\boldsymbol{\beta}$ control such motions and specify (local) interactions among neighboring points, <u>and</u>

  (3) the dynamics in the topology of the point clouds are determined by a simpler latent process $\mathbf{Z}$.

Problem setting

- For learning, we consider datasets of N time-evolving 3D point clouds $\mathcal{P}^1, \ldots, \mathcal{P}^N$; e.g.,



- We assume…
  (1) individual trajectories of points $\mathbf{x}^k$ are governed by a coupled equation of motion

$$\ddot{\mathbf{x}}^k = f_{\boldsymbol{\beta}}\left(\{\mathbf{x}^l\}_{l=1}^K, \dot{\mathbf{x}}^k\right) ,$$

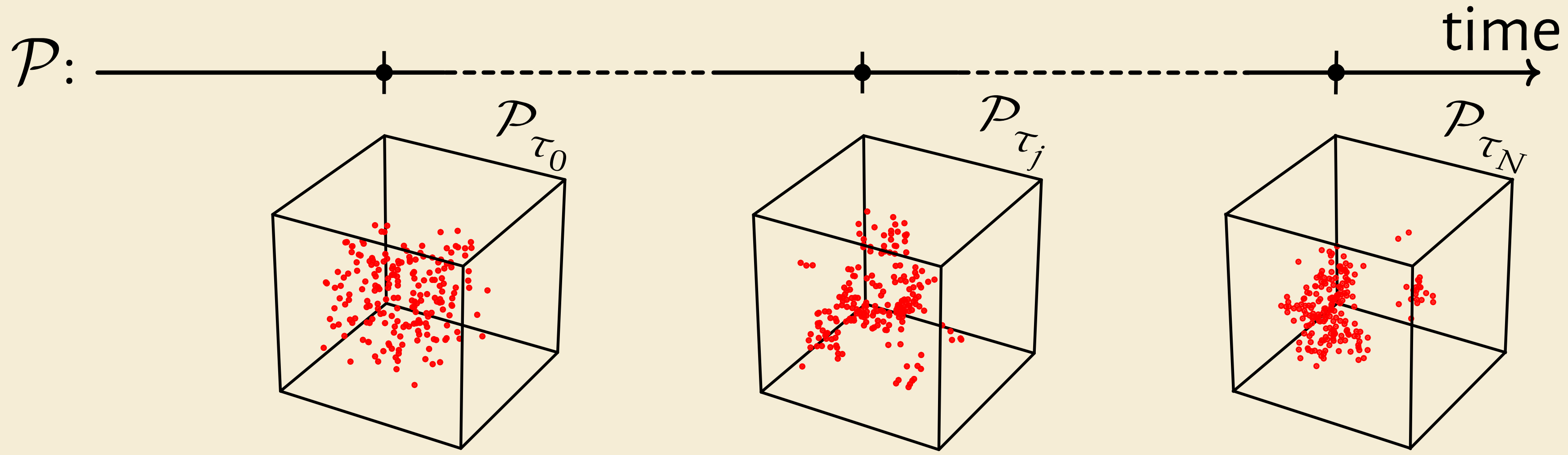
  (2) $\boldsymbol{\beta}$ control such motions and specify (local) interactions among neighboring points, and

  (3) the dynamics in the topology of the point clouds are determined by a simpler latent process $\mathbf{Z}$.

We seek to learn $\mathbf{Z}$ and thus predict $\boldsymbol{\beta}$!

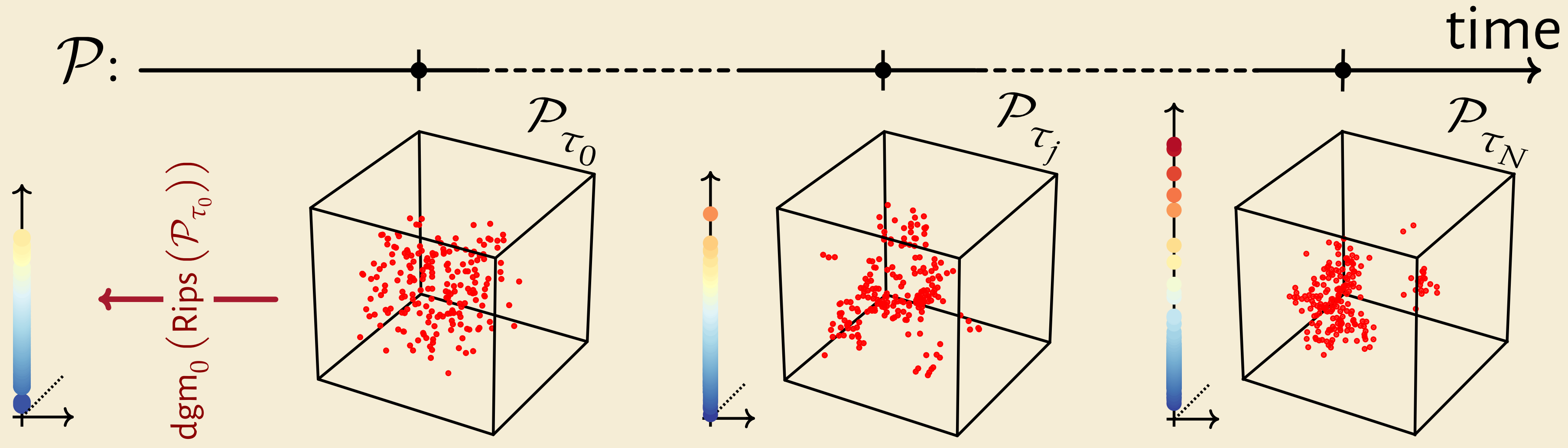
Neural persistence dynamics

- For each sequence $\mathcal{P}$, we pre-compute topological features **per time point**, by...



Neural persistence dynamics

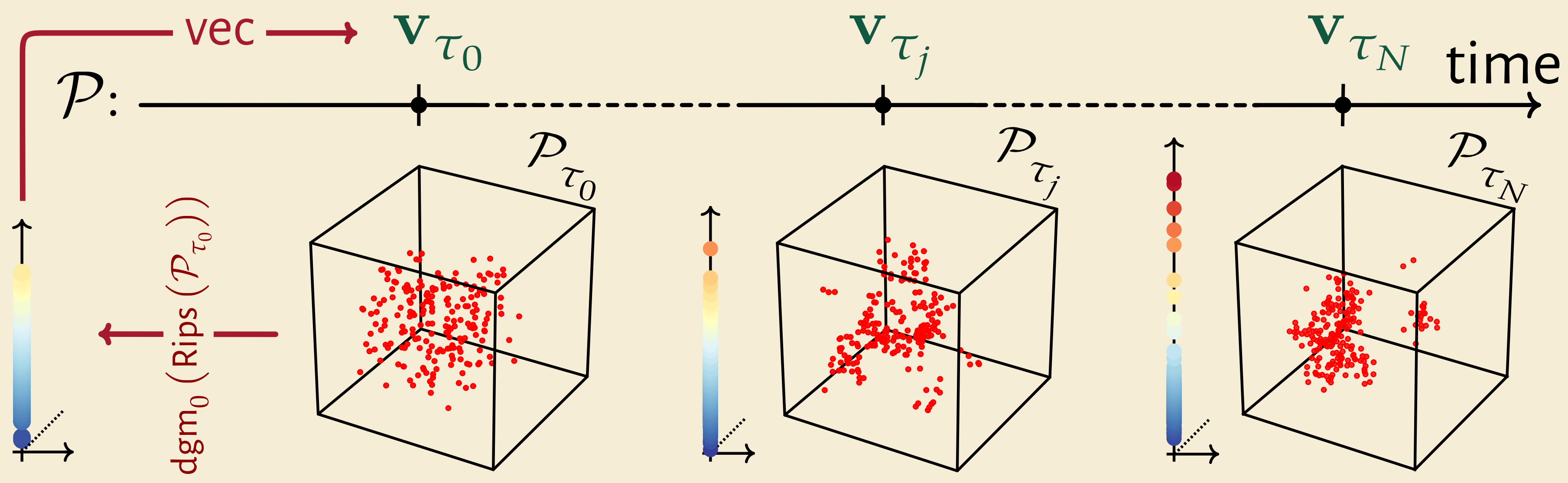
- For each sequence $\mathcal{P}$, we pre-compute topological features **per time point**, by...



(1) applying Vietoris-Rips persistent homology computation, `Rips`, and

Neural persistence dynamics

- For each sequence $\mathcal{P}$, we pre-compute topological features **per time point**, by...

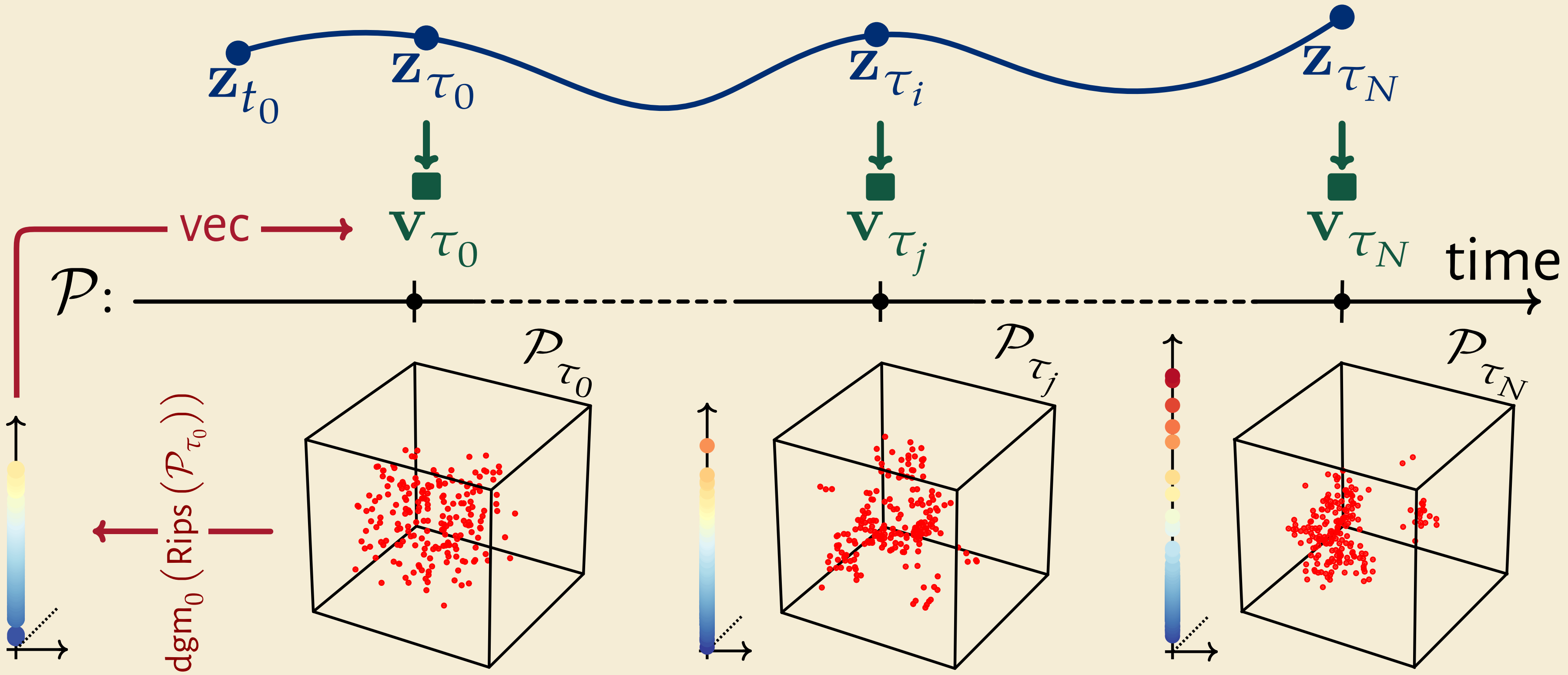


(1) applying Vietoris-Rips persistent homology computation, `Rips`, and

(2) vectorizing the persistence diagrams, `dgm(Rips)`, using Hofer et al. '19.

Neural persistence dynamics

- For each sequence $\mathcal{P}$, we pre-compute topological features **per time point**, by...



(1) applying Vietoris-Rips persistent homology computation, `Rips`, and

(2) vectorizing the persistence diagrams, `dgm(Rips)`, using Hofer et al. '19.

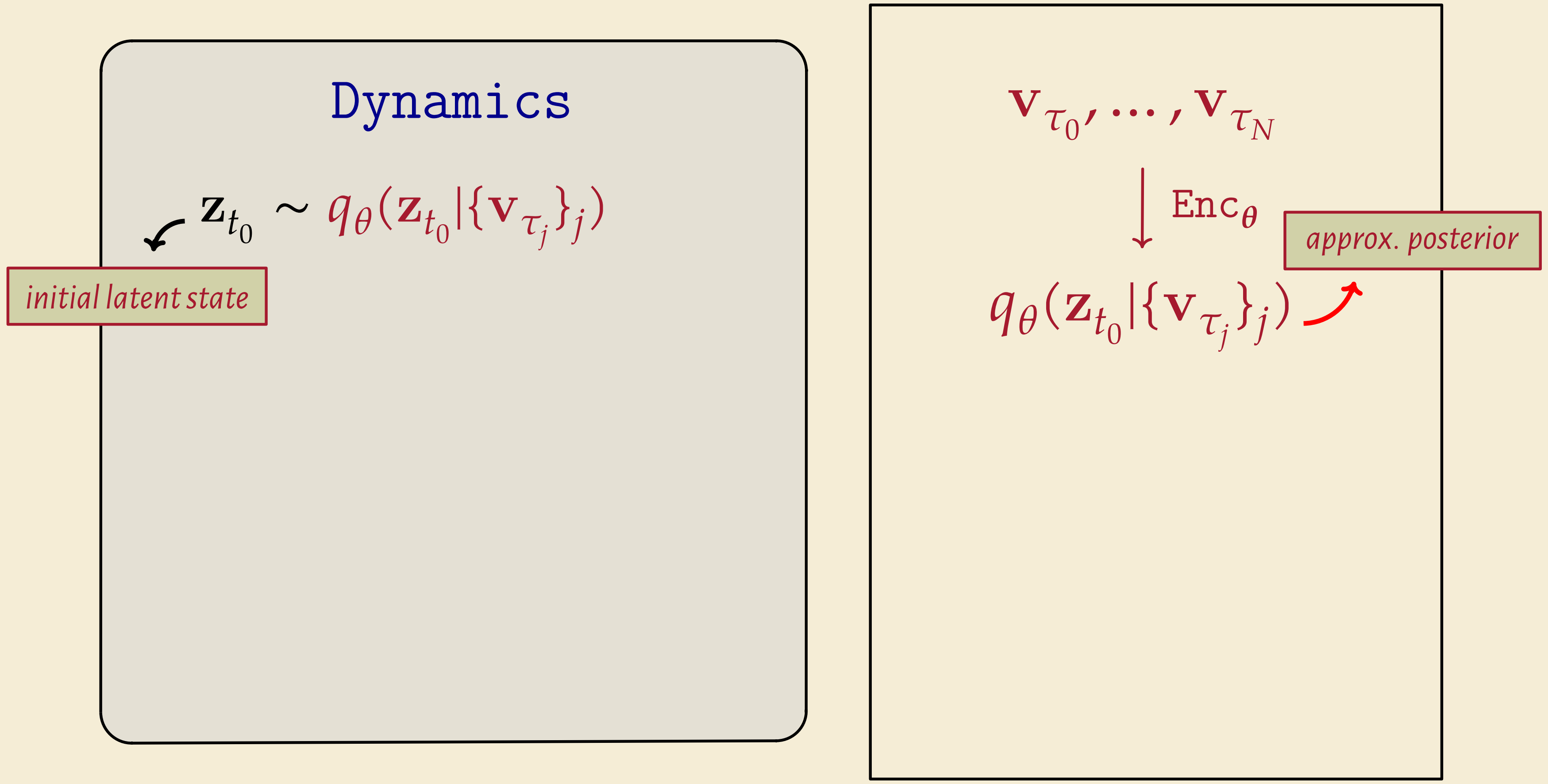
- **Prior works** predominantly extracted **one** topological summary over time.
- **We** learn a latent process $\mathbf{Z}$ whose paths $\{\mathbf{z}_{\tau_j}\}_j$ can **(i)** reproduce the vectorizations, and **(ii)** serve as input for predicting $\boldsymbol{\beta}$.

A model incarnation

- $\boldsymbol{Z}$ is modeled via a neural ODE by Rubanova et al. ’19 and learned in a variational Bayes regime.

A model incarnation

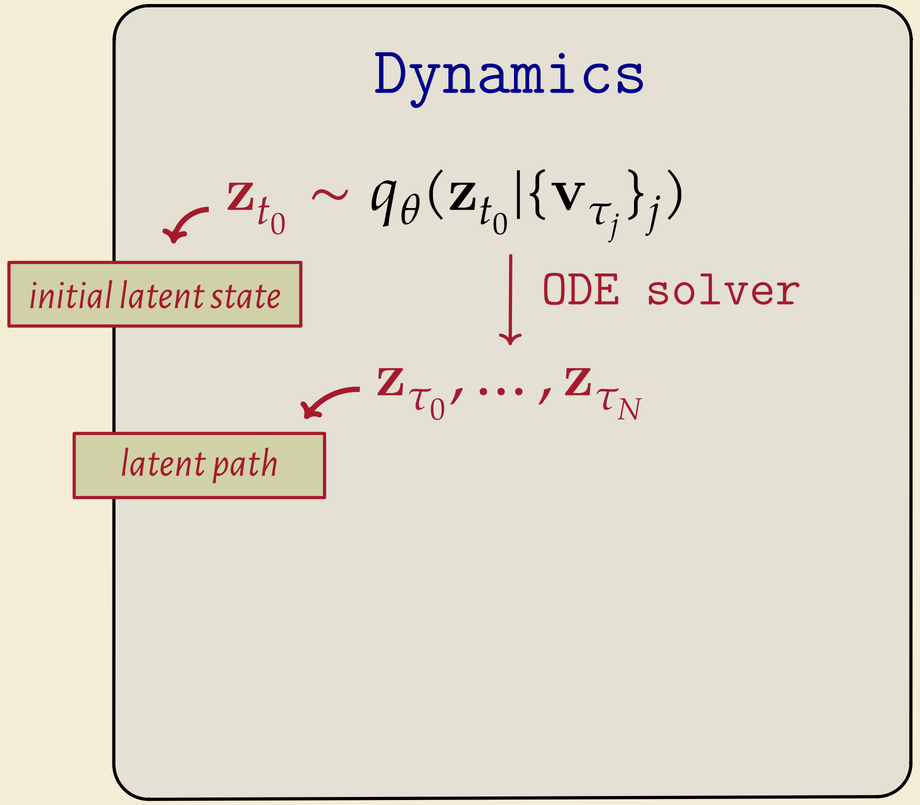
- $\mathbf{Z}$ is modeled via a neural ODE by Rubanova et al. ’19 and learned in a variational Bayes regime.
- In this setting one chooses…

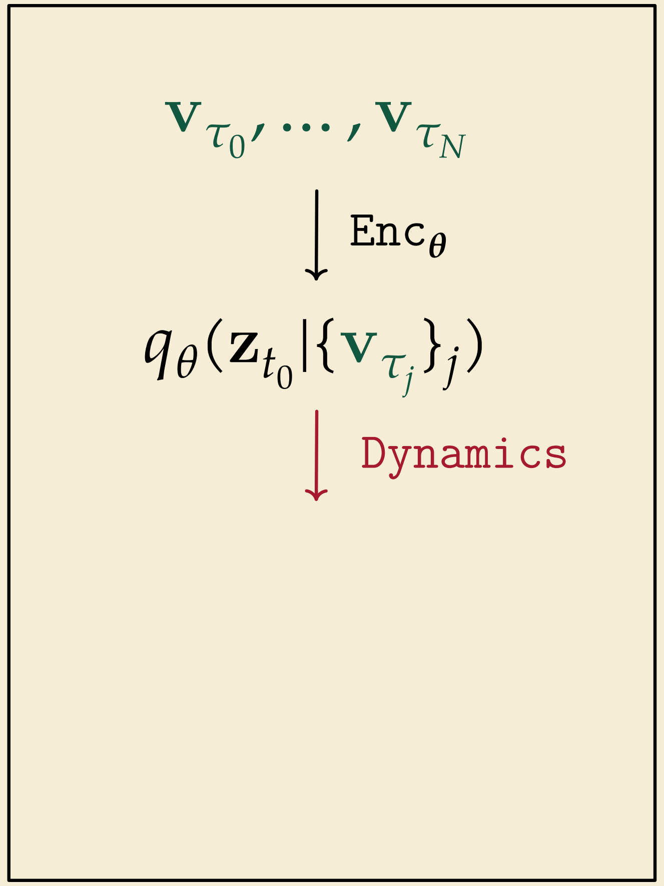


(1) an encoder network ($\mathtt{Enc}_{\boldsymbol{\theta}}$)

A model incarnation

- $\mathbf{Z}$ is modeled via a neural ODE by Rubanova et al. '19 and learned in a variational Bayes regime.
- In this setting one chooses...



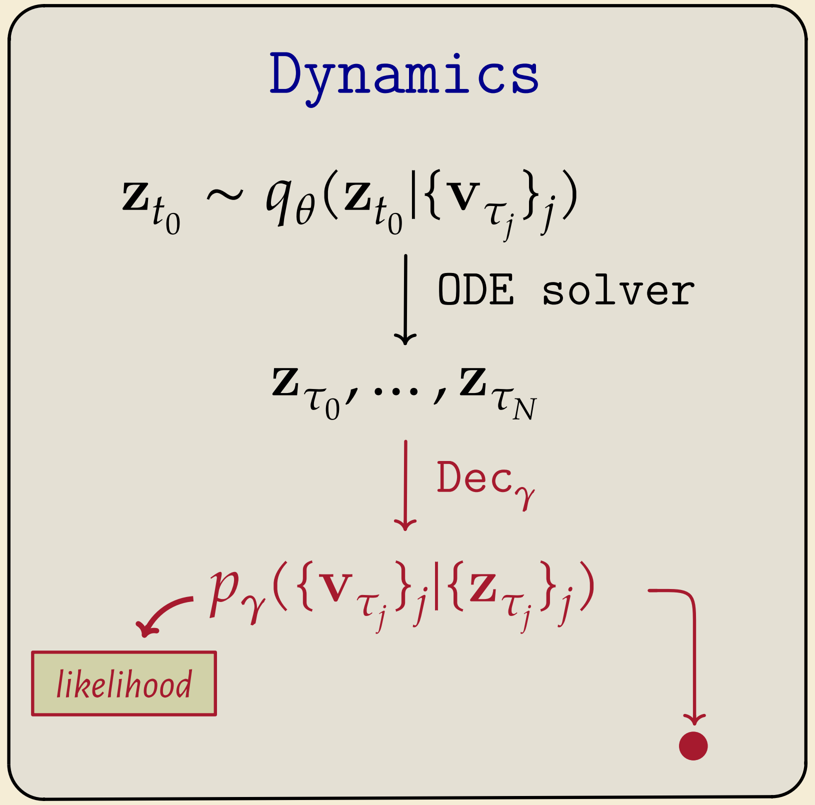


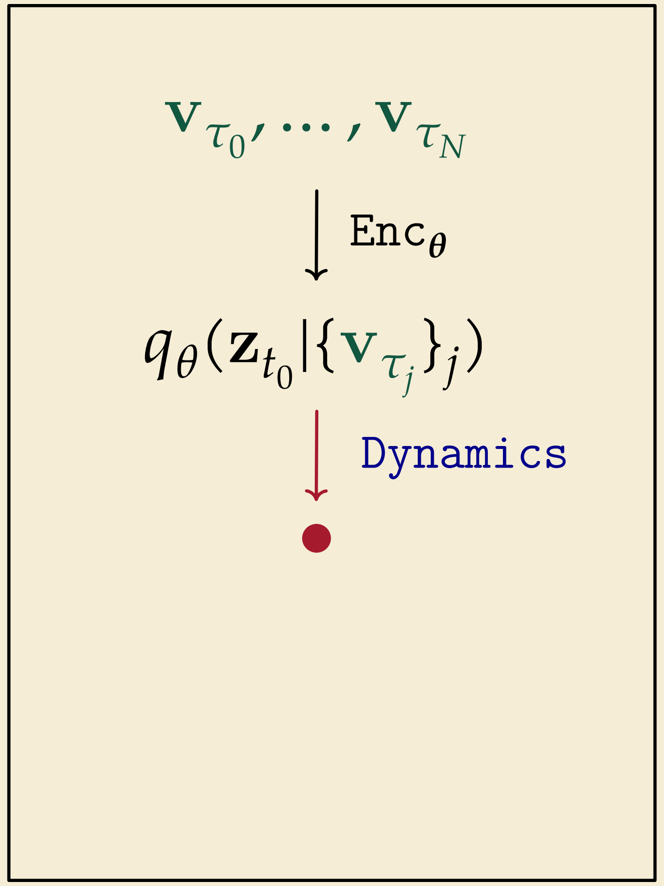
(1) an encoder network ($\mathtt{Enc}_{\boldsymbol{\theta}}$),

(2) an ODE solver

A model incarnation

- $\mathbf{Z}$ is modeled via a neural ODE by Rubanova et al. '19 and learned in a variational Bayes regime.
- In this setting one chooses...





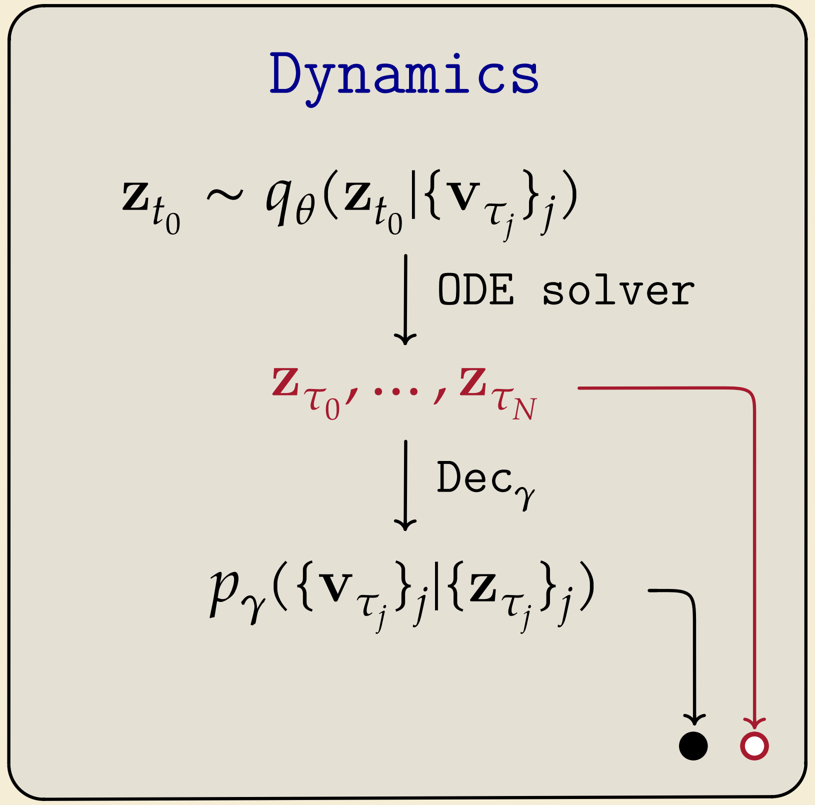
(1) an encoder network ($\mathtt{Enc}_{\boldsymbol{\theta}}$),

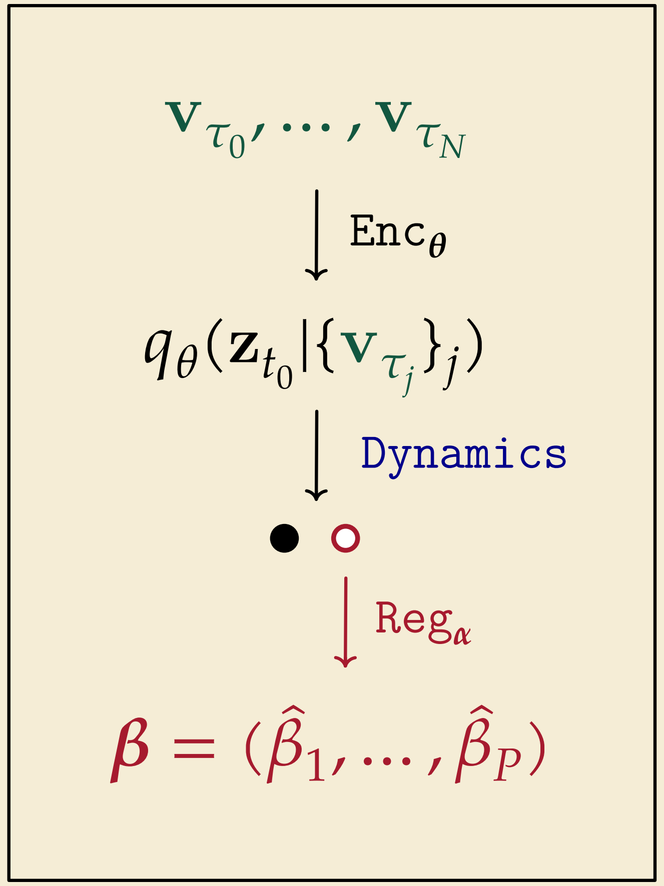
(2) an ODE solver,

(3) a suitable decoder network ($\mathtt{Dec}_\gamma$)

A model incarnation

- $\mathbf{Z}$ is modeled via a neural ODE by Rubanova et al. '19 and learned in a variational Bayes regime.
- In this setting one chooses...





(1) an encoder network ($\mathtt{Enc}_{\boldsymbol{\theta}}$),

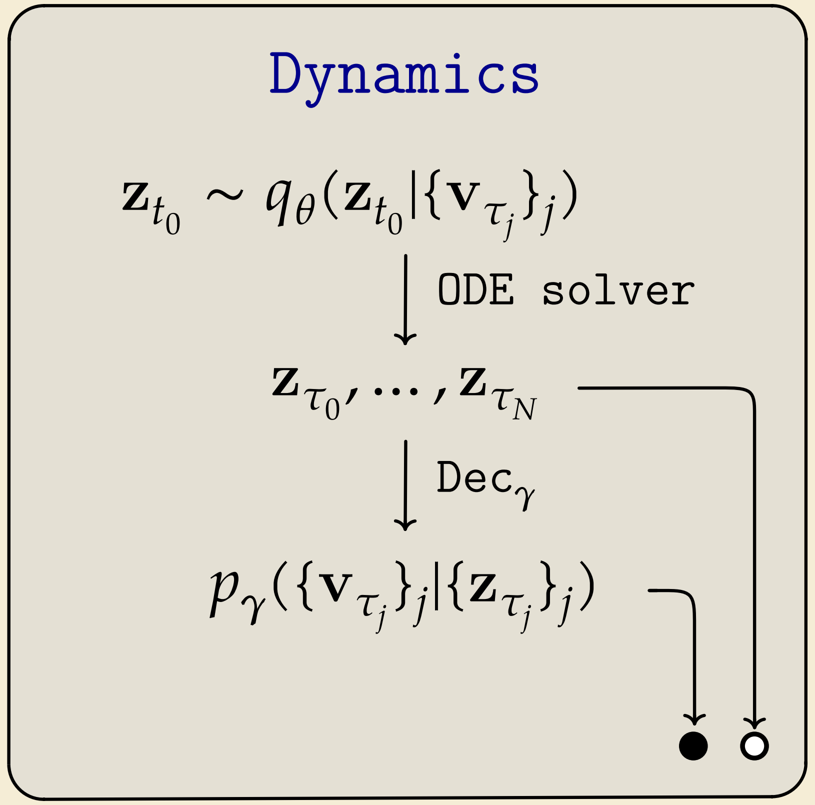
(2) an ODE solver,

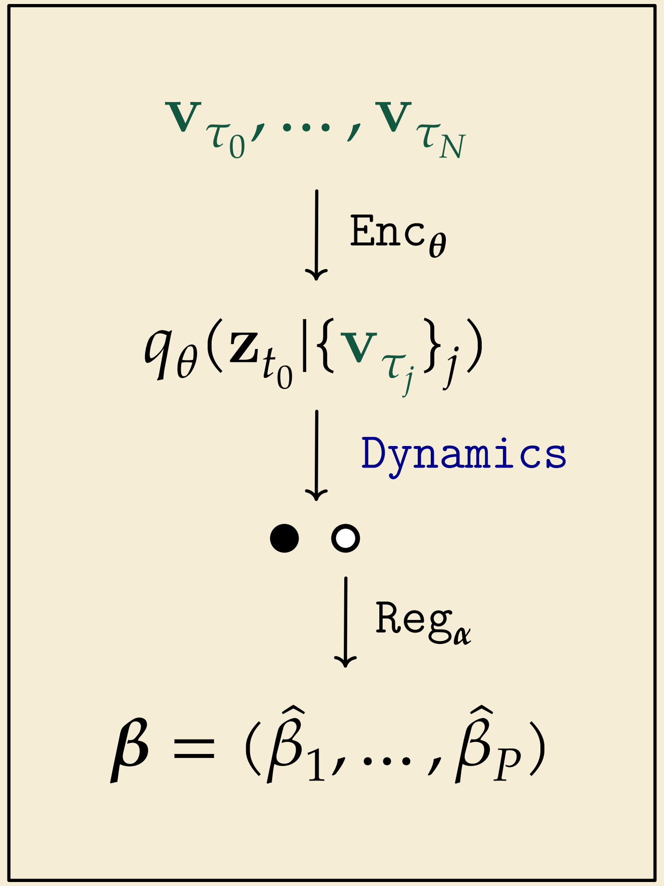
(3) a suitable decoder network ($\mathtt{Dec}_\gamma$), and

(4) a suitable regression network ($\mathtt{Reg}_{\boldsymbol{\alpha}}$) .

A model incarnation

- $\mathbf{Z}$ is modeled via a neural ODE by Rubanova et al. '19 and learned in a variational Bayes regime.
- In this setting one chooses...





(1) an encoder network ($\mathtt{Enc}_{\boldsymbol{\theta}}$),

(2) an ODE solver,

(3) a suitable decoder network ($\mathtt{Dec}_\gamma$), and

(4) a suitable regression network ($\mathtt{Reg}_{\boldsymbol{\alpha}}$) .

- The model is trained upon choosing a prior $p(\mathbf{z}_{t_0})$ and maximizing (ELBO − loss_{aux}), i.e.,

$$\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\alpha} = \underset{\boldsymbol{\theta},\boldsymbol{\gamma},\boldsymbol{\alpha}}{\arg\max} \underbrace{\mathbb{E}_{\mathbf{z}_{t_0} \sim q_{\boldsymbol{\theta}}} \left[\sum_j \log p_\gamma \left(\mathbf{v}_{\tau_j} | \mathbf{z}_{\tau_j} \right) \right] - \mathcal{D}_{\text{KL}} \left(q_{\boldsymbol{\theta}} \left(\mathbf{z}_{t_0} | \{\mathbf{v}_{\tau_j}\}_j \right) \| p \left(\mathbf{z}_{t_0} \right) \right)}_{\text{ELBO}} - \underbrace{\text{loss}_{\text{aux}} \left(\mathtt{Reg}_{\boldsymbol{\alpha}}(\{\mathbf{z}_{\tau_j}\}_j), \boldsymbol{\beta} \right)}_{\text{auxiliary loss}} .$$

Some results

		⌀ VE ↑	⌀ SMAPE ↓
`dorsogna-10k`	**Ours**	**0.851**±0.008	**0.097**±0.005
	PSK	0.828±0.016	0.096±0.006
	Crocker Stacks	0.746±0.023	0.150±0.005
`vicsek-10k`	**Ours**	**0.579**±0.034	**0.146**±0.006
	PSK	0.466±0.009	0.173±0.003
	Crocker Stacks	0.345±0.005	0.190±0.001

- Listed are parameter regression results from two models [D'Orsogna et al. '06 & Vicsek et al. '95] for collective behavior with $|\boldsymbol{\beta}| = 4$, resp.

Some results

		⌀ VE ↑	⌀ SMAPE ↓
`dorsogna-10k`	**Ours**	**0.851**$\pm$0.008	**0.097**$\pm$0.005
	PSK	0.828$\pm$0.016	0.096$\pm$0.006
	Crocker Stacks	0.746$\pm$0.023	0.150$\pm$0.005
`vicsek-10k`	**Ours**	**0.579**$\pm$0.034	**0.146**$\pm$0.006
	PSK	0.466$\pm$0.009	0.173$\pm$0.003
	Crocker Stacks	0.345$\pm$0.005	0.190$\pm$0.001

- Listed are parameter regression results from two models [D'Orsogna et al. '06 & Vicsek et al. '95] for collective behavior with $|\beta| = 4$, resp.
- Each dataset contains 10,000 point cloud sequences simulated using `SiSyPHE` [Diez '21].

Some results

		⌀ VE ↑	⌀ SMAPE ↓
`dorsogna-10k`	**Ours**	**0.851**±0.008	**0.097**±0.005
	PSK	0.828±0.016	0.096±0.006
	Crocker Stacks	0.746±0.023	0.150±0.005
`vicsek-10k`	**Ours**	**0.579**±0.034	**0.146**±0.006
	PSK	0.466±0.009	0.173±0.003
	Crocker Stacks	0.345±0.005	0.190±0.001

- Listed are parameter regression results from two models [D'Orsogna et al. '06 & Vicsek et al. '95] for collective behavior with $|\beta| = 4$, resp.
- Each dataset contains 10,000 point cloud sequences simulated using `SiSyPHE` [Diez '21].
- We compare against ***path signature kernel (PSK)*** [Giusti & Lee '23] & ***crocker stacks*** [Xian et al. '22] and report *Variance Explained (VE)* and *Symmetric Mean Absolute Percentage Error (SMAPE)*.

Some results

		⌀ VE ↑	⌀ SMAPE ↓
	Ours	**0.851**±0.008	**0.097**±0.005
`dorsogna-10k`	PSK	0.828±0.016	0.096±0.006
	Crocker Stacks	0.746±0.023	0.150±0.005
	Ours	**0.579**±0.034	**0.146**±0.006
`vicsek-10k`	PSK	0.466±0.009	0.173±0.003
	Crocker Stacks	0.345±0.005	0.190±0.001

- Listed are parameter regression results from two models [D'Orsogna et al. '06 & Vicsek et al. '95] for collective behavior with $|\beta| = 4$, resp.
- Each dataset contains 10,000 point cloud sequences simulated using `SiSyPHE` [Diez '21].
- We compare against ***path signature kernel (PSK)*** [Giusti & Lee '23] & ***crocker stacks*** [Xian et al. '22] and report *Variance Explained (VE)* and *Symmetric Mean Absolute Percentage Error (SMAPE)*.
- Overall, ***Neural Persistence Dynamics (Ours)*** largely outperforms the state-of-the-art in all tasks.

In summary, ***Neural Persistence Dynamics***...

(1) scales to a **large number** of observation sequences,

(2) is trained with **fixed hyperparameters** across all datasets, and

(3) **outperforms** the state-of-the-art across numerous regression tasks.

In summary, ***Neural Persistence Dynamics***...

(1) scales to a **large number** of observation sequences,

(2) is trained with **fixed hyperparameters** across all datasets, and

(3) **outperforms** the state-of-the-art across numerous regression tasks.

Thanks for your attention!

Come see us at our **poster**.

Fr. 13 Dec 11 a.m. PST – 2 p.m. PST @ Poster Session 5

Full source code is available!

References

[Y. Rubanova, R. T. Q. Chen & D. Duvenaud]
"Latent Ordinary Differential Equations for Irregularly-Sampled Time Series".
In: NeurIPS. 2019.

[C. Hofer, R. Kwitt & M. Niethammer]
"Learning representations of persistence barcodes".
In: JMLR, 20, 126, 1–45, 2019.

[M. R. D'Orsogna, Y. L. Chuang, A. L. Bertozzi & L. S. Chayes]
"Self-Propelled Particles with Soft-Core Interactions: Patterns, Stability, and Collapse".
In: Phys. Rev. Lett., 96, 104302, 2006.

[T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen & O. Shochet]
"Novel type of phase transition in a system of self-driven particles".
In: Phys. Rev. Lett., 75, 6. 1226-1229, 1995.

[A. Diez]
"`SiSyPHE`: A Python package for the Simulation of Systems of interacting mean-field Particles with High Efficiency".
In: J. Open Source Softw., 65, 3653, 6, 2021

[C. Giusti & D. Lee]
"Signatures, Lipschitz-Free Spaces, and Paths of Persistence Diagrams".
In: SIAM J. Appl. Algebra Geom., 4, 7, 828-866, 2023.

[L. Xian, H. Adams, C. M. Topaz & L. Ziegelmeier]
"Capturing dynamics of time-varying data via topology".
In: Found. Data Sci., 1, 4, 1-36, 2022.