Neural Persistence Dynamics

Sebastian Zeng$^{\dagger/\ddagger}$, Florian Graf†, Martin Uray$^{\dagger/\ddagger}$, Stefan Huber‡, Roland Kwitt†
†University of Salzburg, Austria
‡Josef Ressel Centre for Intelligent and Secure Industrial Automation,
University of Applied Sciences, Salzburg, Austria
`{sebastian.zeng, florian.graf, roland.kwitt}@plus.ac.at`,
`{martin.uray, stefan.huber}@fh-salzburg.ac.at`

Abstract

We consider the problem of learning the dynamics in the topology of time-evolving point clouds, the prevalent spatiotemporal model for systems exhibiting collective behavior, such as swarms of insects and birds or particles in physics. In such systems, patterns emerge from (local) interactions among self-propelled entities. While several well-understood governing equations for motion and interaction exist, they are notoriously difficult to fit to data, as most prior work requires knowledge about individual motion trajectories, i.e., a requirement that is challenging to satisfy with an increasing number of entities. To evade such confounding factors, we investigate collective behavior from a *topological perspective*, but instead of summarizing entire observation sequences (as done previously), we propose learning a latent dynamical model from topological features *per time point*. The latter is then used to formulate a downstream regression task to predict the parametrization of some a priori specified governing equation. We implement this idea based on a latent ODE learned from vectorized (static) persistence diagrams and show that a combination of recent stability results for persistent homology justifies this modeling choice. Various (ablation) experiments not only demonstrate the relevance of each model component but provide compelling empirical evidence that our proposed model – *Neural Persistence Dynamics* – substantially outperforms the state-of-the-art across a diverse set of parameter regression tasks.

1 Introduction

Understanding emerging behavioral patterns of a collective through the interaction of individual entities is key to elucidate many phenomena in nature on a macroscopic and microscopic scale. Prominent examples are coherently moving flocks of birds, the swarming behavior of insects and fish or the development of cancerous cells, all understood as 2D/3D point clouds that *evolve over time*. Importantly, several widely-accepted governing equations for collective behavior exist [18, 40, 56] which, *when appropriately parameterized*, can reproduce different incarnations of typically observed patterns. Even more importantly, these equations are tied to physically interpretable parameters and can provide detailed insights into the intrinsic mechanisms that control various behavioral regimes. However, while it is fairly straightforward to simulate collective behavior from governing equations (see [21]), the *inverse* problem, i.e., *identifying* the model parameters from the data, turns out to be inherently difficult. Confounding factors include the often large number of observed entities and the difficulty of reliably identifying individual trajectories across point clouds at possibly non-equidistant observation times.

However, as several works [4, 23, 52] have recently demonstrated, it may not be necessary to rely on individual trajectories for parameter identification. In fact, collective behavior is characterized by global patterns that emerge from local interactions, and we observe the emergence of these patterns through changes to the “shape” of point clouds over time. For instance, birds may form a flock, split into groups, and then merge again. This perspective has prompted the idea of summarizing such topological events over time and then phrasing model identification as a downstream *prediction/regression task*. The key challenge here lies in the transition from topological summaries of point clouds at specific observation times, typically obtained via *persistent homology (PH)* [3, 9, 22], to the *dynamic* regime where the temporal dimension plays a crucial role.

Yet, despite the often remarkable performance of topological approaches in terms of parameter identification for models of collective behavior, it remains unclear in which situations they are preferable over more traditional (learning) methods that can handle point cloud data, as, e.g., used in computer vision problems [24, 46]. In the spirit of [53], we highlight this point by previewing a snapshot of one ablation experiment from Sec. 4. In particular, Tbl. 1 compares the parameter regression performance of our proposed approach, using PH, to a variant where we instead use PointNet++ [45] representations. Referring to Fig. 1, this means that the $\mathbf{v}_{\tau_i}$ are computed by a PointNet++ model. As can be seen in Tbl. 1, relying on representations computed via PH works well on *both datasets*, while PointNet++ representations fail on one dataset (`vicsek-10k`), but indeed perform reasonably well on the other (`dorsogna-1k`).

Table 1: PointNet++ [45] vs. persistent homology (PH) representations; ↑ means higher and ↓ means lower is better.

	⊘ **VE** ↑	⊘ **SMAPE** ↓
	`vicsek-10k`	
Ours (PointNet++, `v2`)	0.274±0.085	0.199±0.014
Ours (PH, `v1`)	**0.579**±0.034	**0.146**±0.006
	`dorsogna-1k`	
Ours (PointNet++, `v2`)	0.816±0.031	0.132±0.018
Ours (PH, `v1`)	**0.851**±0.008	**0.096**±0.004

In light of this observation, it is worth emphasizing that there is a clear distinction in terms of the *source* of point cloud dynamics when comparing collective behavior to problems in computer vision where moving objects are of primary interest and PointNet variants (see [24]) have shown stellar performance: in particular, point clouds in vision primarily evolve due to a change in pose or camera motion, whereas collective behavior is driven by (local) intrinsic point interactions. The latter induces changes to the “shape” of the point clouds, i.e., a phenomenon that is typically not observed in vision.

Contribution(s). Our key idea is to learn a generative model that can reproduce topological summaries *at each point in time*. This contrasts prior work, which primarily aims to extract *one* summary representation of the entire timeline. In detail, we advocate for modeling the dynamics of vectorized persistence diagrams via a continuous latent variable model (e.g., a latent ODE), see Fig. 1, and to use the resulting latent paths as input to a downstream parameter regression task. Recent stability results for vectorizations of persistence diagrams – relating distances among the latter to the Wasserstein distance between point clouds – justify this modeling choice. Aside from state-of-the-art performance on various parameter identification tasks for established models of collective behavior, our approach scales favorably with the number of observed sequences, accounts for non-equidistant observation times, and is easily combinable with other sources of information.

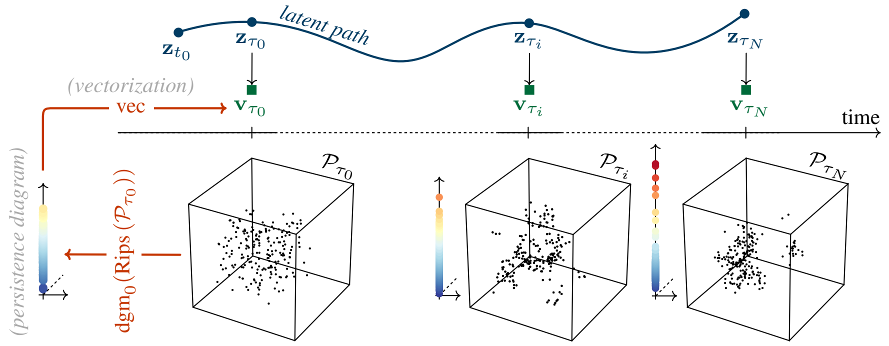

Figure 1: Conceptual overview of *Neural Persistence Dynamics*. Given is a sequence of observed point clouds $\mathcal{P}_{\tau_0}, \ldots, \mathcal{P}_{\tau_N}$. *First*, we summarize each $\mathcal{P}_{\tau_i}$ via (Vietoris-Rips) persistent homology into persistence diagrams (zero-, one- and two-dimensional; only the zero-dimensional diagrams are shown) which are then vectorized into $\mathbf{v}_{\tau_i}$ via existing techniques. *Second*, we model the dynamics in the sequence $\mathbf{v}_{\tau_0}, \ldots, \mathbf{v}_{\tau_N}$ via a continuous latent variable model (in our case, a latent ODE) and then use a summary of the latent path to predict the *parameters* of specific governing equation(s) of collective behavior. Precomputed steps are highlighted in red.

2 Related work

Our work is partially related to the literature on learning with dynamic point clouds in vision problems, but *primarily* connects to work on summarizing topological features over time and inferring interaction laws of collective behavior from data.

Summarizing topological features over time. A common denominator in the literature on encoding topological changes in dynamic point clouds is the use of *persistent homology*, extended to accommodate the temporal dimension in various ways. One of the early works along this direction are *persistence vineyards* [16], introduced as a means to study folding trajectories of proteins. Based on the idea of tracking points in persistence diagrams over time, vineyards come with stability properties similar to persistent homology [41], but are expensive to compute and compare on a large scale. In cases where one would know the "right" scale at which to compute homology, one may also use *zigzag persistence* [8], as done in [17, 54], to track homological changes over time. Nevertheless, for problems with a large number of observation sequences, scale selection per sequence is nontrivial and highly impractical.

Alternatively, instead of thinking about the evolution of individual points in persistence diagrams over time, one may discard the matching between points and instead focus on *sequences of summary representations* of said diagrams. In [25], for instance, the authors work directly with persistence diagrams (per time point) to identify changes in the topology of time-varying graphs. In terms of temporal summary representations, [52] introduce *crocker plots* to encode the evolution of topological features by stacking discretized Betti curves over time. *Crocker stacks* [57], an extension of this concept, adds a smoothing step that gradually reduces the impact of points of low persistence and, upon discretization, yields a third dimension to crocker plots. In our context, both crocker plots & stacks have been successfully used as input to regression methods to estimate the parametrization of models of collective behavior [4]. By drawing on prior work on kernels for sequentially ordered data [33], [23] follow a conceptually similar strategy as [52, 57], introducing a *path signature kernel (PSK)* for sequences of summary representations of persistence diagrams.

Along a different line of research, [28, 29] propose *formigrams* as summaries of dynamic metric data, encoding the evolution of *connected components*. In subsequent work, [30] construct multidimensional (i.e., spatiotemporal) persistent homology modules from dynamic metric data and compare invariants of these modules. While these works provide important theoretical stability results for zero-dimensional homological features, i.e., connected components, it is unclear how their construction extends to homological features of higher dimension in a tractable manner. However, it is worth pointing out that recent progress along the lines of vectorizations of multiparameter persistent homology [36] in combination with [30] (who essentially construct multiparameter persistence modules) might, in the future, constitute a tractable approach to study dynamically changing point clouds with learning methods.

Notably, computational challenges also arise in the context of crocker plots/stacks and PSKs. Despite their remarkable performance in distinguishing different configurations of models for collective behavior, both approaches suffer scalability issues: either (i) in terms of unfavorable scalability with respect to the dimensionality of vectorized persistence diagrams (as with the PSK approach of [23]), or (ii) in terms of unfavorable scalability with the number of observation sequences (as is the case for crocker plots/stacks, due to the need for extensive cross-validation of the discretization parameters). As we show in Sec. 4, our method not only outperforms these techniques by a large margin, but also scales to large amounts of training sequences and requires little hyperparameter tuning.

In addition to the closely related works discussed above, we highlight that there is a large body of literature on the topological analysis of time-varying signals, such as studying fMRI data via cubical persistence [47], or the persistent homology of delay embeddings [43, 44] of time series. In our context, however, these works are only partially related/relevant, as they do assume precise knowledge about individual trajectories over time (e.g., voxel IDs in fMRI data), which is unrealistic when seeking to infer parametrizations of models for collective behavior from data.

Inferring interaction laws for models of collective behavior. In the context of studying characteristics of collective behavior, there is a second line of closely related work on inferring interaction laws (see, e.g., [2, 5, 6, 27, 39, 61]), ranging from metric-distance-based models and topological interaction models to non-parametric estimators of interaction kernels. While a thorough survey of this literature is beyond the scope of this paper, we highlight that one *common denominator* in these works is their reliance on correspondences between points across time, e.g., to infer individual point velocities or trajectories. To give an example, in recent work [39, 61], the authors derive non-parametric estimators for interaction kernels (certain functions of pairwise distances) between observed points which crucially hinges on the traceability of each particle over time. While such approaches are conceptually appealing and even allow for reconstructing or extrapolating trajectories, they operate in the *observation space*. Even under a moderate number of points m, computing these estimators becomes prohibitively expensive (as, e.g., in 3D, the state space is $\mathbb{R}^{3m}$). In contrast, our approach only requires positional information and can handle larger point clouds. Furthermore, although we only present results on predicting parameters for a class of a priori specified governing equations, by formulating an auxiliary regression task, our underlying model may also be used to predict other quantities of interest.

Finally, taking a slightly broader perspective on prior art, we want to highlight recent progress on learning-based approaches that study inverse problems in the context of classic partial differential equations (PDEs). It might be possible to apply such approaches [37, 55, 60] to specific models of collective behavior such as volume exclusion [40], for which the asymptotics of infinitely many particles are solutions to parametric PDEs [42]. However, for other models, the number of particles strongly influences the dynamics. For instance, in the D'Orsogna model [18], particle distances can collapse to zero as the number of particles tends to infinity.

3 Method

Below, we present our method with a focus on its application to modeling collective behavior in point clouds. Importantly, the latter represents only one of many conceivable use cases. In fact, the core ideas can be applied in exactly the same way whenever one can extract topological features from time-dependent data (e.g., graphs or images).

Notation. In the following, $\mathcal{P} = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\} \subset \mathbb{R}^3$ denotes a point cloud with M points and $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$ denotes the Euclidean metric. Point clouds may be indexed by t_i (or τ_i) to highlight the dependence on time t_i. If necessary, we clearly distinguish between τ_i as a time point with an available observation, and t_i as a general placeholder for time.

Problem statement. Given a sequence of point clouds $\mathcal{P}_{\tau_0}, \ldots, \mathcal{P}_{\tau_N}$, observed at possibly non-equidistant time points τ_i, we (1) seek to model their topological evolution over time and then (2) use this model to predict the parametrization of an a priori defined governing equation of collective behavior. The latter is typically specified by a small number of parameters $\beta_1, \ldots, \beta_P$ that control the motions $\mathrm{d}\mathbf{x}_i/\mathrm{d}t$ of individual points $\mathbf{x}_i$ and specify (local) interactions among neighboring points.

As preparation for our model description, we first briefly establish how one may extract topological features from a point cloud $\mathcal{P}$, using *persistent homology* [3, 9, 22] – the arguably most prominent and computationally most feasible approach.

Persistent homology of point clouds. Persistent homology seeks to uncover and concisely summarize topological features of $\mathcal{P}$. To this end, one constructs a topological space from $\mathcal{P}$ in the form of a simplicial complex, and studies its homology across multiple scales. The most relevant construction for our purposes is the *Vietoris-Rips* complex $\mathrm{Rips}(\mathcal{P})_\delta$, with vertex set $\mathcal{P}$. This complex includes an m-simplex $[\mathbf{x}_0, \ldots, \mathbf{x}_m]$ iff $d(\mathbf{x}_i, \mathbf{x}_j) \leq \delta$ for all $0 \leq i, j \leq m$ at a given threshold δ. The "shape" of this complex can then be studied using homology, a tool from algebraic topology, with zero-dimensional homology (H_0) encoding information about connected components, one-dimensional homology (H_1) encoding information about loops and two-dimensional homology (H_2) encoding information about voids; we refer to this information as *homological/topological features*. Importantly, if $\delta_b \leq \delta_d$, then $\mathrm{Rips}(\mathcal{P})_{\delta_b} \subseteq \mathrm{Rips}(\mathcal{P})_{\delta_d}$, inducing a sequence of inclusions when varying δ, called a *filtration*. The latter in turn induces a sequence of vector spaces $\mathrm{H}_k(\mathrm{Rips}(\mathcal{P})_{\delta_b}) \to \mathrm{H}_k(\mathrm{Rips}(\mathcal{P})_{\delta_d})$ at the homology level. Throughout this sequence, homological features (corresponding to basis vectors of the different vector spaces) may appear and disappear; we say they are *born* at some δ_b and *die* at $\delta_d > \delta_b$. For instance, a one-dimensional hole might appear at a particular value of δ_b and disappear at a later δ_d; in other words, the hole *persists* from δ_b to δ_d, hence the name *persistent homology*. As different features may be born and die at the same time, the collection of (birth, death) tuples is a multiset of points, often represented in the form of a persistence diagram, which we denote as $\mathrm{dgm}_k(\mathrm{Rips}(\mathcal{P}))$. Here, the notation $\mathrm{Rips}(\mathcal{P})$ refers to the full filtration and dgm_k indicates that we have tracked k-dimensional homological features throughout this filtration.

Clearly this construction does not account for any temporal changes to a point cloud, but reveals features that are present at a *specific* time point. Furthermore, due to the inconvenient multiset structure of persistence diagrams for learning problems, one typically resorts to appropriate *vectorizations* (see, e.g., [1, 7, 10, 26]), all accounting for the fact that points close to the diagonal $\Delta = \{(x, x) : x \in \mathbb{R}\}$ contribute less (0 at Δ) to the vectorization. The latter is important to preserve stability (see paragraph below) with respect to perturbations of points in the diagram. In the following, we refer to a vectorized persistence diagram of a point cloud $\mathcal{P}_{t_i}$ as

$$\mathbf{v}_{t_i,k} := \mathrm{vec}(\mathrm{dgm}_k(\mathrm{Rips}(\mathcal{P}_{t_i}))) \ , \tag{1}$$

where $\mathbf{v}_{t_i,k} \in \mathbb{R}^d$ and d is controlled by hyperparameter(s) of the chosen vectorization technique. For brevity, we omit the subscript k when referring to vectorizations unless necessary.

Remark 1. *As our goal is to model the dynamics of vectorized persistence diagrams using a* continuous *latent variable model, it is important to discuss the dependence of the vectorizations on the input data. In particular, for our modeling choice to be sound, vectorizations should vary (Lipschitz) continuously with changes in the point clouds over time. We discuss this aspect next.*

Stability/Continuity aspects. First, we point out that persistence diagrams can be equipped with different notions of *distance*, most prominently the bottleneck distance and Wasserstein distances, both based upon the cost of an optimal matching between two diagrams, allowing for matches to the diagonal Δ. We refer the reader to [12, 51] for a detailed review. *Stability* in the context of persistent homology is understood as persistence diagrams varying (Lipschitz) continuously with the input data. While seminal stability results exist for the Wasserstein distances [15] and the bottleneck distance [12, 14], most results from the literature focus on the latter.

In the case of *vectorization techniques* for persistence diagrams (e.g., [1, 26]), it turns out that, most of the time, vectorizations are only Lipschitz continuous with respect to Wasserstein distances, and existing results are typically of the form

$$d(\mathrm{vec}(F), \mathrm{vec}(G)) \le K\, W_1(F, G) \ , \tag{2}$$

where W_1 denotes the $(1, 2)$-Wasserstein distance as defined in [51, Def. 2.7], F, G are two persistence diagrams and $K > 0$ is a Lipschitz constant. In the context of Rem. 1, one may be tempted to combine Eq. (2) with the seminal *Wasserstein stability theorem* from [15] to infer stability of vectorizations with respect to the input data. Yet, the conditions imposed in [15] actually eliminate a direct application of this result in most practical cases, as discussed in [51]. However, the latter work also provides an alternative: in our particular case of Vietoris-Rips persistent homology, one can upper bound W_1 in terms of the standard *point set* Wasserstein distance $\mathcal{W}_1$. Specifically, upon relying on a stable vectorization technique, we obtain the inequality chain

$$\begin{aligned} d(\mathrm{vec}(F_k), \mathrm{vec}(G_k)) &\overset{[26,\text{ Thm. 12}]}{\le} K\, W_1(F_k, G_k) \\ &\overset{[51,\text{ Thm. 5.9}]}{\le} 2K \binom{M-1}{k} \mathcal{W}_1(\mathcal{P}, \mathcal{Q}) \ , \end{aligned} \tag{3}$$

where $F_k = \mathrm{dgm}_k(\mathrm{Rips}(\mathcal{P}))$, $G_k = \mathrm{dgm}_k(\mathrm{Rips}(\mathcal{Q}))$ denote the k-dimensional persistence diagrams of point clouds $\mathcal{P}$ and $\mathcal{Q}$ of equal cardinality M. In particular, we realize vec via the approach outlined in [26, Def. 10] using *exponential structure elements* [26, Def. 19].

Overall, the continuity property in Eq. (3) guarantees that small changes in dynamic point clouds over time only induce small changes in their vectorized persistence diagrams, and therefore provides a solid justification for the model discussed next.

Latent variable model for persistence diagram vectorizations. As we presume that the dynamic point clouds under consideration are produced (or can be described sufficiently well) by equations of motions with only a few parameters, cf. Fig. 3, it is reasonable to assume that the dynamics of the (vectorized) persistence diagrams are equally governed by a somewhat simpler unobserved/latent dynamic process in $\mathbb{R}^z$ with $z \ll d$. We model this latent dynamic process via a neural ODE [13, 49], learned in a variational Bayes regime[1] [32]. In this setting, one chooses a recognition/encoder network ($\mathsf{Enc}_{\boldsymbol{\theta}}$) to parametrize an approximate variational posterior $q_{\boldsymbol{\theta}}(\mathbf{z}_{t_0} | \{\mathbf{v}_{\tau_i}\}_i)$, an ODE solver

[1]Other choices, e.g., a latent SDE as in [35, 58] are possible, but we did not explore this here.

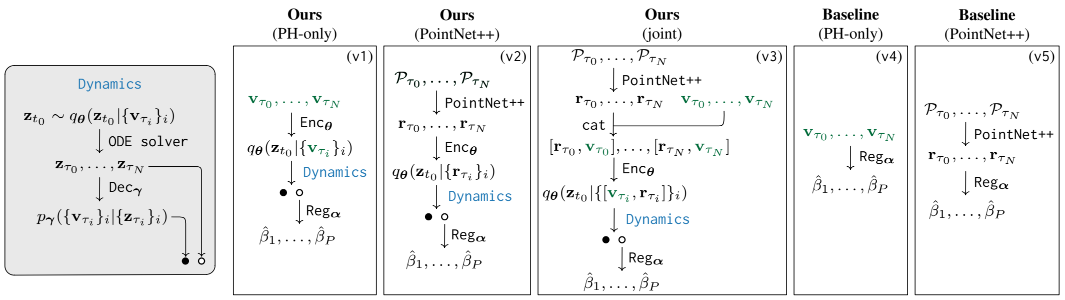

Figure 2: Schematic illustration of different model variants. The first three variants (*left* to *right*) explicitly model latent dynamics (later denoted as *w/ dynamics*), the baseline variants do not (later denoted as *w/o dynamics*), but still incorporate the attention mechanism of the encoder from [50], which we use throughout.

to yield latent states $\{\mathbf{z}_{\tau_i}\}_i$ at observed time points $\tau_i \in [0, T]$ and a suitable generative/decoder network ($\mathtt{Dec}_{\boldsymbol{\gamma}}$) to implement the likelihood $p_{\boldsymbol{\gamma}}(\mathbf{v}_{\tau_i}|\mathbf{z}_{\tau_i})$. Upon choosing a suitable prior $p(\mathbf{z}_{t_0})$, one can then train the model via ELBO maximization, i.e.,

$$\boldsymbol{\theta}, \boldsymbol{\gamma} = \arg\max_{\boldsymbol{\theta},\boldsymbol{\gamma}} \mathbb{E}_{\mathbf{z}_{t_0}\sim q_{\boldsymbol{\theta}}} \left[\sum\nolimits_i \log p_{\boldsymbol{\gamma}}(\mathbf{v}_{\tau_i}|\mathbf{z}_{\tau_i}) \right] - D_{\mathrm{KL}}(q_{\boldsymbol{\theta}}(\mathbf{z}_{t_0}|\{\mathbf{v}_{\tau_i}\}_i) \| \, p(\mathbf{z}_{t_0})) \ . \tag{4}$$

Different to [49], we do not implement the recognition/encoder network via another neural ODE, but rather choose an attention-based approach (mTAN) [50] which can even be used in a standalone manner as a strong baseline (see Sec. 4). In our implementation, the recognition network yields the parametrization $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ of a multivariate Gaussian in $\mathbb{R}^z$ with diagonal covariance, and the prior is a standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I}_z)$. Furthermore, the ODE solver (e.g., Euler) can yield $\mathbf{z}_{t_i}$ at any desired t_i, however, we can only evaluate the ELBO at observed time points τ_i.

Regression objective. To realize our downstream regression task, i.e., predicting parameters $\beta_1, \ldots, \beta_P$ of an underlying governing equation for collective behavior (see Fig. 3) from a given observation sequence, we have multiple choices. By our assumption of a latent dynamic process that carries information about the dynamic nature of the topological changes, it is reasonable to tie the simulation parameter estimates $\hat{\beta}_1, \ldots, \hat{\beta}_P$ to the *latent path* $\{\mathbf{z}_{t_i}\}_i$ via a regression network $\mathtt{Reg}_{\boldsymbol{\alpha}}$ that accepts $\{\mathbf{z}_{t_i}\}_i$ as input. In particular, we re-use the attention-based encoder architecture $\mathtt{Enc}_{\boldsymbol{\theta}}$ to allow attending to different parts of this path. However, different to $\mathtt{Enc}_{\boldsymbol{\theta}}$, which parametrizes the approximate posterior from *observations* at time points τ_i, the regression network $\mathtt{Reg}_{\boldsymbol{\alpha}}$ (with its own set of parameters $\boldsymbol{\alpha}$) receives *latent states* $\mathbf{z}_{t_i}$ at equidistant $t_i \in [0, T]$, then summarizes this sequence into a vector and linearly maps the latter to $\hat{\beta}_1, \ldots, \hat{\beta}_P$, cf. Fig. 2. As in [49], for training, we extend the ELBO objective of Eq. (4) by an additive auxiliary regression loss, such as the mean-squared error (MSE), between the predictions $\hat{\beta}_p$ and the ground truth β_p, implicitly making a Gaussian noise assumption.

A schematic overview of our Neural Persistence Dynamics approach is shown in Fig. 1. Additional details, including different model variants, are illustrated in Fig. 2.

Remark 2. *Clearly, our presented framework allows for many architectural choices. While some components affect downstream (regression) performance only marginally, others have a more profound impact, and we have already identified some recommended choices above (based on our experiments in Sec. 4). This includes (1) our choice of a stable persistent homology vectorization* vec, *where we rely on [26] due to consistently reliable performance without much hyperparameter tuning, and (2) our choice of recognition network, where we choose an attention-based approach (mTAN) [50] which has proven to be very effective in practice [58]. In Sec. 4.2, we will provide additional configuration details for our experimental study.*

4 Experiments

4.1 Datasets

Similar to previous work [4, 23, 52, 57], we evaluate and compare our approach on simulation data that is generated from parametric models of collective behavior. Specifically, we consider

D'Orsogna (`dorsogna`) [18]

$$\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = \mathbf{v}_i,\; m\frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = \left(\alpha - \beta|\mathbf{v}_i|^2\right)\mathbf{v}_i - \frac{1}{M}\nabla_{\mathbf{x}_i}\sum_{j=1,j\neq i}^{M} U(\|\mathbf{x}_i-\mathbf{x}_j\|),\; U(r) = C_r e^{-\frac{r}{l_r}} - C_a e^{-\frac{r}{l_a}} \tag{5}$$

Vicsek (`vicsek`) [20, 56]

$$\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = c\,\mathbf{v}_i,\; \mathrm{d}\mathbf{v}_i = \left(\mathbf{I}-\mathbf{v}_i\mathbf{v}_i^\top\right)\left(\nu\,\bar{\mathbf{v}}_i\,\mathrm{d}t + \sqrt{2D}\,\mathrm{d}\mathbf{B}_t\right),\; \bar{\mathbf{v}}_i = \frac{\mathbf{a}_i}{\|\mathbf{a}_i\|},\; \mathbf{a}_i = \sum_{j,\|\mathbf{x}_i-\mathbf{x}_j\|\leq R}\mathbf{v}_j \tag{6}$$

Volume exclusion [40] (`volex`) variant from [21]

$$\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = -\frac{\alpha}{R}\sum_{j=1,i\neq j}^{M}\phi\left(\frac{\|\mathbf{x}_j-\mathbf{x}_i\|^2}{4R^2}\right)(\mathbf{x}_i-\mathbf{x}_j),\quad \text{with } \phi(r) = \begin{cases}\frac{1}{r}-1, & 0<r\leq 1\\ 0, & \text{else}\end{cases} \tag{7}$$

$+$ *cell division* & *cell death* at constant rate(s) $\lambda_b, \lambda_d \in [0,1]$

Figure 3: Models of collective behavior. Parameters that are varied to obtain different behavior are highlighted in red; the range of each parameter is listed in Appendix A. In the **Vicsek** model, $\mathbf{B}_t$ denotes Brownian motion.

three different models in $\mathbb{R}^3$: D'Orsogna [18], Vicsek [20] and volume exclusion [40], using the publicly-available implementations in the `SiSyPHE` library [21][2]. The corresponding equations of motion (and interaction laws) are summarized in Fig. 3. The parameters that are varied to generate the datasets, i.e., the response variables of the regression tasks, are highlighted in red. We sample these parameters, as specified in Appendix A, to cover a wide range of macroscopic behavioral regimes. For each sampled parameter configuration, we simulate one sequence of point clouds, to a total of 10,000 sequences per model. All simulations are run for 1,000 time steps (with step size 0.01, starting at $t = 0$) on point clouds of size $M = 200$. We take every 10th time step as an observation, yielding observation sequences of length 100. At $t = 0$, points are drawn independently and uniformly in $[-0.5, 0.5]^3$, and initial velocities are uniformly distributed on the unit sphere. For direct comparison to [23], we also simulated their parameter configuration of the D'Orsogna model. This setup yields four datasets: `dorsogna-1k` (from [23]), `dorsogna-10k`, `vicsek-10k` and `volex-10k`.

4.2 Implementation, training & evaluation metrics

Implementation. The model variants from Fig. 2 can be realized in many ways. Below, we specify the configuration that was used to run the experiments. For the encoder $\mathtt{Enc}_{\boldsymbol{\theta}}$, as well as the regression network $\mathtt{Reg}_{\boldsymbol{\alpha}}$, we use the attention-based mTAN architecture [50]. As decoder network $\mathtt{Dec}_{\boldsymbol{\gamma}}$, we choose a two-layer MLP with ReLU activations, and as `ODE solver`, we select the Euler method. Each model is trained for 150 epochs using ADAM [31] (with a weight decay of 0.001), starting at a learning rate of 0.001 (decaying according to a cosine annealing schedule) and MSE as a reconstruction (i.e., to evaluate the first term in Eq. (4)) and regression loss. In case a model uses topological features, we use Ripser++ [59] to compute (prior to training) persistent homology of dimension up to two, i.e., H_0, H_1, and H_2. Vectorizations of *each* persistence diagram are then obtained using *exponential structure elements* from [26, Def. 19]. In particular, we use 20 structure elements per diagram, which yields a $d = 3\cdot 20$ dimensional representation per point cloud and time point. The location of each structure element is set to one of 20 k-means++ cluster centers, obtained by running the latter on a random subset of 50,000 points (per dimension) selected from all persistence diagrams available during training; the scale parameter of each structure element is set according to [48, Eq. (2)]. To model the dynamics, we fix the latent space dimensionality to $z = 20$ and scale the ODE integration time to $[0, 1]$. While other settings are undoubtedly possible, we did not observe any noticeable benefits from increasing the dimensionality of the vectorizations or the latent space. Our publicly available reference implementation can be found at `https://github.com/plus-rkwitt/neural_persistence_dynamics`.

Evaluation metrics. We randomly partition each dataset into five training/testing splits of size 80/20. To obtain a robust performance estimate for different regimes of missing and unevenly spaced observations, we train three models (per split) using only a fraction (i.e., 20%, 50%, and 80%) of

[2]`https://github.com/antoinediez/Sisyphe`

randomly chosen time points per sequence. Similarly, all testing splits undergo the same sampling procedure. All scores (see below) are reported as an average ($\pm$ one standard deviation) over the five splits and the three time point sampling percentages. Specifically, we report the *variance explained (VE)* [34] (in $[0, 1]$; higher is better $\uparrow$) and the *symmetric mean absolute percentage error (SMAPE)* (in $[0, 1]$; lower is better $\downarrow$). For each testing sequence in each split, the scores are computed from the *true* simulation parameters (see variables marked red in Fig. 3) and the corresponding *predictions* (denoted by $\hat{\beta}_i$ in Fig. 2). Finally, when reporting results on a dataset, we mark the best score in **bold**, as well as all other scores that *do not* show a statistically significant difference in mean (assessed via a Mann-Whitney test at 5% significance and correcting for multiple comparisons).

4.3 Ablation

In our ablation study, we assess (1) the relevance of different point cloud representations (PH vs. PointNet++), (2) any potential benefits of modeling latent dynamics and (3) the impact of varying the observation timeframe. Additional ablation experiments can be found in Appendix B.3.

Are representations complementary? We first investigate the impact of different point cloud representations (from PH and PointNet++, resp.), by comparing variants `v1`,`v2` and `v3` from Fig. 2.

Tbl. 2 shows an extension of Tbl. 1, listing results for the `vicsek-10k` and `dorsogna-1k` data. While using PointNet++ representations on `vicsek-10k` yields rather poor performance, combining them with representations computed from PH is at least *not* detrimental. On the other hand, on `dorsogna-1k`, where PointNet++ yields decent SMAPE and VE scores, the combination of both sources substantially outperforms each single source in isolation. Although, the complementary nature of a topological perspective on data has been pointed out many times in the literature, it is rarely as pronounced as in this particular experiment. *Hence, we will stick to the combination of PH+PointNet++ representations (i.e.,* `v3` *in Fig. 2) in any subsequent experiments.*

Table 2: Ablation study on the relevance of different point cloud representations.

Source	⊘ **VE** $\uparrow$	⊘ **SMAPE** $\downarrow$
	`vicsek-10k`	
PointNet++ (`v2`)	0.274±0.085	0.199±0.014
PH (`v1`)	**0.579**±0.034	**0.146**±0.006
PH+PointNet++ (`v3`)	**0.576**±0.030	**0.144**±0.006
	`dorsogna-1k`	
PointNet++ (`v2`)	0.816±0.031	0.132±0.018
PH (`v1`)	0.851±0.008	0.097±0.005
PH+PointNet++ (`v3`)	**0.931**±0.004	**0.067**±0.002

Are explicit latent dynamics beneficial? To assess the impact of explicitly modeling the dynamics of persistence diagram vectorizations, we ablate the latent ODE part of our model. In detail, we compare `v3` from Fig. 2 against using $\texttt{Reg}_\alpha$ operating *directly* on concatenated point cloud representations from PH and PointNet++ (i.e., a combination of variants `v4` & `v5` from Fig. 2). The latter approach constitutes an already *strong baseline* (cf. [58]) as $\texttt{Reg}_\alpha$ incorporates an attention mechanism that allows attending to relevant parts of each sequence. For a fair comparison, we increase the size of $\texttt{Reg}_\alpha$ to approximately match the number of parameters to our latent dynamic model.

As in Tbl. 2, we list results for the `vicsek-10k` and `dorsogna-1k` data in Tbl. 3. First, we see that explicitly modeling the latent dynamics is beneficial, with considerable improvements in the reported scores across both datasets. While differences are quite prominent on the `dorsogna-1k` data, they are less pronounced in the SMAPE score on `vicsek-10k`, but still statistically significant. Second, it is important to point out that even without any explicit latent dynamics, the regression performance (see Tbl. 4) is already above the current state-of-the-art (i.e., compared to PSK [23]), highlighting the necessity for strong baselines.

Table 3: Results on the relevance of latent dynamics.

	⊘ **VE** $\uparrow$	⊘ **SMAPE** $\downarrow$
	`vicsek-10k`	
w/ dynamics	**0.576**±0.030	**0.144**±0.006
w/o dynamics	0.512±0.020	0.155±0.004
	`dorsogna-1k`	
w/ dynamics	**0.931**±0.004	**0.067**±0.002
w/o dynamics	0.850±0.006	0.098±0.004

Impact of the observation timeframe. So far, we have presented results (as in prior work) where observation sequences are extracted from the beginning of a simulation (i.e., starting from $t_0 = 0$), which corresponds to observing the emergence of patterns out of a random yet qualitatively similar, initial configuration of points. In the following experiment, we investigate the impact of extracting observation sequences with *starting points randomly spread across a much longer simulation time*.

This is relevant as it is unlikely to observe uniform initial positions in any data obtained from real-world experiments. To this end, we create variations of the `dorsogna-10k` data by progressively increasing the simulation time T from 1,000 to 20,000 and then randomly extracting sub-sequences of length 1,000 (again, as before, taking each 10th step as an observation). The extension of the simulation timeframe has the effect that the difficulty of the learning task considerably increases with T. We argue that this is due to the increased variation across the observed sequences while the amount of training data remains the same. For comparison, we also list results for the PSK approach of [23], however, only for the *optimal* case of observing *all* time points within a sequence (due to the massive PSK computation time; $>$ 3 days). As the PSK relies on persistent homology only, we limit our model to the `v1` variant from Fig. 2. As seen from Fig. 4, we observe an increase/drop in SMAPE and VE, resp., for both approaches, yet our approach degrades much slower with T.

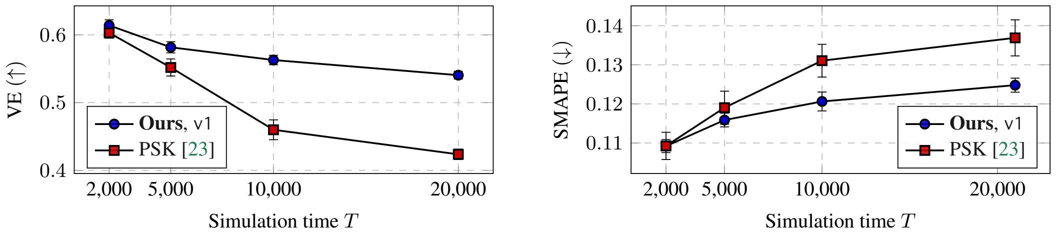

Figure 4: Impact of the maximal simulation time T for extracting training/testing sequences starting at $\tau_0 \in [0, T - 1000]$, assessed on the `dorsogna-10k` dataset.

4.4 Comparison to the state-of-the-art

Finally, we present parameter regression results on the full battery of datasets and compare against two of the most common approaches from the literature: the *path signature kernel (PSK)* of [23] and the *crocker stacks* approach from [57]. For both competitors, we report results when *all* time points are observed to establish a baseline of *what could be achieved* in the best case. Note that we evaluate the PSK approach [23] on exactly the same vectorizations as our approach, and we replicate their experimental setting of choosing the best hyperparameters via cross-validation. Similarly, for crocker stacks, we cross-validate the discretization parameters, see Appendices B.1 and B.2. Tbl. 4 lists the corresponding results. Aside from the observation that our approach largely outperforms the state-of-the-art across all parameter regression tasks, we also highlight that our model is trained with *exactly* the same hyperparameters on all datasets.

Remark 3. *A closer look at the* `volex-10k` *dataset, in particular its governing equations in Fig. 3, shows that the cardinality of the point clouds may change over time due to cell division or death. While this is no practical limitation for our approach, our stability arguments from Eq.* (3) *no longer apply. We conjecture that one may be able to extend the result of [51] to account for such cases, but this might require a case-by-case analysis and possibly include additional assumptions.*

5 Discussion

We introduced a framework to capture the dynamics observed in topological summary representations over time and presented one incarnation using a latent ODE on top of vectorized Vietoris-Rips persistence diagrams, addressing the problem of *predicting parametrizations for models of collective behavior*. Conceptually, *Neural Persistence Dynamics* embodies the idea of learning the dynamics in a temporal sequence of *vectorized* topological summaries instead of, e.g., trying to track individual homology classes over time (as with persistence vineyards, see Sec. 2). Our approach successfully scales to a large number of observation sequences, requires little to no parameter tuning, and vastly outperforms the current state-of-the-art, as demonstrated by several experiments on dynamic point cloud datasets with varying characteristics. Finally, we emphasize that the fundamental ideas of *Neural Persistence Dynamics* may also be applied to other types of time-varying data (e.g., graphs or images) and downstream objectives simply by adjusting the filtration choice. We hope that our work will stimulate further research in this direction.

Table 4: Comparison to the state-of-the-art across four diverse datasets of collective behavior. Ours (joint) refers to the variant `v3` of Fig. 2, Ours (PH-only) refers to variant `v1`. The latter setting is directly comparable to the PSK and crocker stacks approach. Best results are marked **bold**. Multiple bold entries indicate that there is no significant difference in mean. Note that `dorsogna-1k` is simulated exactly as in [23] varying only *two parameters*, whereas for `dorsogna-10k` we vary *four* parameters (as with `vicsek-10k` and `volex-10k`).

		$\oslash$ **VE** $\uparrow$	$\oslash$ **SMAPE** $\downarrow$
`dorsogna-1k`	Ours (joint, `v3`)	**0.931**±0.004	**0.067**±0.002
	Ours (PH-only, `v1`)	0.851±0.008	0.097±0.005
	PSK [23]	0.828±0.016	0.096±0.006
	Crocker Stacks [57]	0.746±0.023	0.150±0.005
`dorsogna-10k`	Ours (joint, `v3`)	**0.689**±0.021	**0.088**±0.004
	Ours (PH-only, `v1`)	**0.680**±0.025	**0.090**±0.005
	PSK [23]	0.647±0.005	0.100±0.003
	Crocker Stacks [57]	0.343±0.016	0.145±0.001
`vicsek-10k`	Ours (joint, `v3`)	**0.576**±0.030	**0.144**±0.006
	Ours (PH-only, `v1`)	**0.579**±0.034	**0.146**±0.006
	PSK [23]	0.466±0.009	0.173±0.003
	Crocker Stacks [57]	0.345±0.005	0.190±0.001
`volex-10k`	Ours (joint, `v3`)	**0.871**±0.019	**0.081**±0.006
	Ours (PH-only, `v1`)	**0.869**±0.018	**0.082**±0.007
	PSK [23]	0.509±0.003	0.190±0.003
	Crocker Stacks [57]	0.076±0.019	0.292±0.004

Limitation(s). One obvious limitation in the presented implementation is the reliance on Vietoris-Rips persistent homology of point clouds. In fact, the underlying simplicial complex will become prohibitively large (especially for H_2) once we scale up the number of points by, e.g., an order of magnitude. Using an Alpha [22] or a Witness complex [19] might be a viable alternative to mitigate this issue. Similarly, one may explore subsampling strategies for persistent homology, as in [11], and learn a latent ODE from a combination of estimates.

Societal impact. The present work mainly deals with an approach to capture the dynamics observed in data representing collective behavior. We assume that this has no direct societal impact, but point out that any application of such a model (aside from synthetic data), e.g., in the context of behavioral studies on living beings or health sciences (e.g., to study the development of cancer cells), should be carefully reviewed. This is especially advisable when drawing conclusions from missing measurements, as these are based on imputations and may be biased.

Acknowledgments

This work was supported by the Land Salzburg within the EXDIGIT project 20204-WISS/263/6-6022 and projects 0102-F1901166- KZP, 20204-WISS/225/197-2019. M. Uray and S. Huber are supported by the Christian Doppler Research Association (JRC ISIA). *We also thank all reviewers for the valuable feedback during the review process.*

References

[1] H. Adams, T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta, and L. Ziegelmeier. “Persistence Images: A stable vector representation of persistent homology”. In: *JMLR* 18.8 (2017), pp. 1–35.

[2] M. Ballerini, N. Cabibbo, R. Candelier, V. D. F. D., E. Marinari, O. Pellizzola, G. Parisi, A. Viale, and V. Zdravkovic. “Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study”. In: *PNAS* 105.4 (2008), pp. 1232–1237.

[3] S. Barannikov. “Framed Morse complex and its invariants”. In: *Adv. Soviet Math.* 21 (1994), pp. 93–115.

[4] D. Bhaskar, A. Manhart, J. Milzman, J. T. Nardini, K. M. Storey, C. M. Topaz, and L. Ziegelmeier. “Analyzing collective motion with machine learning and topology”. In: *Chaos* 29.12 (2019), p. 123125.

[5] W. Bialek, A. Cavanga, I. Giardina, and A. Walczak. “Statistical mechanics for natural flocks of birds”. In: *PNAS* 109.13 (2012), pp. 4786–4791.

[6] S. L. Brunton, J. L. Proctor, and J. N. Kutz. “Discovering govering equations from data by sparse identification of nonlinear dynamical systems”. In: *PNAS* 113.15 (2016), pp. 3932–3937.

[7] P. Bubenik. “Statistical Topological Data Analysis using Persistence Landscapes”. In: *JMLR* 16.1 (2015), pp. 77–102.

[8] G. Carlsson and V. de Silva. “Zigzag Persistence”. In: *Found. Comput. Math.* 10.4 (2010), pp. 367–405.

[9] G. Carlsson and M. Vejdemo-Johansson. *Topological Data Analysis with Applications*. Cambridge University Press, 2021.

[10] M. Carrière, F. Chazal, Y. Ike, T. Lacombe, M. Royer, and Y. Umeda. “PersLay: A Neural Network Layer for Persistence Diagrams and New Graph Topological Signatures”. In: *AISTATS*. 2020.

[11] F. Chazal, B. Fasy, F. Lecci, B. Michel, A. Rinaldo, and L. Wasserman. “Subsampling Methods for Persistent Homology”. In: *ICML*. 2015.

[12] F. Chazal, V. de Silva, and S. Y. Oudot. “Persistence Stability for Geometric Complexes”. In: *Geometriae Dedicata* 173.1 (2014), pp. 193–214.

[13] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. “Neural Ordinary Differential Equations”. In: *NeurIPS*. 2018.

[14] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. “Stability of persistence diagrams”. In: *Discrete Comput. Geom.* 37.1 (2007), pp. 103–120.

[15] D. Cohen-Steiner, H. Edelsbrunner, J. Harer, and Y. Mileyko. “Lipschitz Functions Have L_p-Stable Persistence”. In: *Found. Comput. Math.* 10.2 (2010), pp. 127–139.

[16] D. Cohen-Steiner, H. Edelsbrunner, and D. Morozov. “Vines and vineyards by updating persistence in linear time”. In: *SCG*. 2006.

[17] P. Corcoran and C. B. Jones. “Modelling Topological Features of Swarm Behaviour in Space and Time With Persistence Landscapes”. In: *IEEE Access* 5 (2017), pp. 18534–18544.

[18] M. R. DOrsogna, Y. L. Chuang, A. L. Bertozzi, and L. S. Chayes. “Self-Propelled Particles with Soft-Core Interactions: Patterns, Stability, and Collapse”. In: *Phys. Rev. Lett.* 96.10 (2006), p. 104302.

[19] V. De Silva and G. Carlsson. “Topological estimation using witness complexes”. In: *Sympos. Point-Based Graphics*. 2004.

[20] P. Degond and S. Motsch. “Continuum limit of self-driven particles with orientation interaction”. In: *Math. Models Methods Appl. Sci.* 18.supp01 (2008), pp. 1193–1215.

[21] A. Diez. “`SiSyPHE`: A Python package for the Simulation of Systems of interacting mean-field Particles with High Efficiency”. In: *J. Open Source Softw.* 6.65 (2021), p. 3653.

[22] H. Edelsbrunner and J. Harer. *Computational Topology. An Introduction.* AMS, 2010.

[23] C. Giusti and D. Lee. “Signatures, Lipschitz-Free Spaces, and Paths of Persistence Diagrams”. In: *SIAM J. Appl. Algebra Geom.* 7.4 (2023), pp. 828–866.

[24] Y. Guo, H. Wang, Q. Hu, H. Liu, and M. Bennamoun. “Deep Learning for 3D point clouds: A survey”. In: *TPAMI* 43.12 (2021), pp. 4338–4364.

[25] M. Hajij, B. Wang, C. Scheidegger, and P. Rosen. “Visual Detection of Structural Changes in Time-Varying Graphs Using Persistent Homology”. In: *PacificVis*. 2018.

[26] C. Hofer, R. Kwitt, and M. Niethammer. “Learning representations of persistence barcodes”. In: *JMLR* 20.126 (2019), pp. 1–45.

[27] Y. Katz, K. Tunstrøm, C. C. Ioannou, C. Huepe, and I. D. Couzin. “Inferring individual rules from collective behavior”. In: *PNAS* 108.46 (2010), pp. 18720–18725.

[28] W. Kim and F. Memoli. “Formigrams: Clustering Summaries of Dynamic Data”. In: *CCCG*. 2018.

[29] W. Kim and F. Mémoli. “Extracting Persistent Clusters in Dynamic Data via Möbius inversion”. In: *Discrete Comput. Geom.* 71 (2024), pp. 1276–1342.

[30] W. Kim and F. Mémoli. “Spatiotemporal Persistent Homology for Dynamic Metric Spaces”. In: *DCG* 66.4 (2021), pp. 831–875.

[31] D. P. Kingma and J. Ba. “Adam: A Method for Stochastic Optimization”. In: *ICLR*. 2015.

[32] D. P. Kingma and M. Welling. “Auto-Encoding Variational Bayes”. In: *ICLR*. 2014.

[33] F. J. Kiraly and H. Oberhauser. “Kernels for sequentially ordered data”. In: *JMLR* 20.31 (2019), pp. 1–45.

[34] J. Li. “Assessing the accuracy of predictive models for numerical data: Not r nor r^2, why not? Then what?” In: *PLOS ONE* 12.8 (2017), e0183250.

[35] X. Li, T.-K. L. Wong, R. T. Q. Chen, and D. Duvenaud. “Scalable Gradients for Stochastic Differential Equations”. In: *AISTATS*. 2020.

[36] D. Loiseaux, L. Scoccola, M. Carrière, M. B. Botnan, and S. Oudot. “Stable Vectorization of Multiparameter Persistent Homology using Signed Barcodes as Measures”. In: *NeurIPS*. 2023.

[37] D. Long and S. Zhe. *Invertible Fourier Neural Operators for Tackling Both Forward and Inverse Problems*. 2024. arXiv: 2402.11722 [cs.LG].

[38] M. Löning, A. Bagnall, S. Ganesh, V. Kazakov, J. Lines, and F. J. Király. “sktime: A Unified Interface for Machine Learning with Time Series”. In: *Workshop on Systems for ML at NeurIPS*. 2019.

[39] F. Lu, M. Zhong, S. Tang, and M. Maggioni. “Nonparametric inference of interaction laws in systems of agents from trajectory data”. In: *PNAS* 116.29 (2019), pp. 14424–14433.

[40] S. Motsch and D. Peurichard. “From short-range repulsion to Hele-Shaw problem in a model of tumor growth”. In: *J. Math. Biol.* 76 (2018), pp. 205–234.

[41] E. Munch. “Applications of Persistent Homology to Time Varying Systems”. PhD thesis. Duke University, 2013.

[42] K. Oelschläger. “Large systems of interacting particles and the porous medium equation”. In: *J. Differ. Equ.* 88.2 (1990), pp. 294–346.

[43] J. A. Perea, A. Deckard, S. B. Haase, and J. Harer. “SW1PerS: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data”. In: *BMC Bioinform* 16.1 (2015), p. 257.

[44] J. A. Perea and J. Harer. “Sliding windows and persistence: An application of topological methods to signal analysis”. In: *Found. Comput. Math.* 15.3 (2015), pp. 799–838.

[45] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space”. In: *NeurIPS*. 2017.

[46] D. Rempe, T. Birdal, Y. Zhao, Z. Gojcic, S. Sridhar, and L. J. Guibas. “CaSPR: Learning Canonical Spatiotemporal Point Cloud Representations”. In: *NeurIPS*. 2020.

[47] B. Rieck, T. Yates, C. Bock, K. Borgwardt, G. Wolf, N. Turk-Browne, and S. Krishnaswamy. “Uncovering the topology of time-varying fMRI data using cubical persistence”. In: *NeurIPS*. 2020.

[48] M. Royer, F. Chazal, C. Levrard, Y. Umeda, and Y. Ike. “ATOL: Measure Vectorization for Automatic Topologically-Oriented Learning”. In: *AISTATS*. 2021.

[49] Y. Rubanova, R. T. Q. Chen, and D. K. Duvenaud. “Latent Ordinary Differential Equations for Irregularly-Sampled Time Series”. In: *NeurIPS*. 2019.

[50] S. N. Shukla and B. M. Marlin. “Multi-Time Attention Networks for Irregularly Sampled Time Series”. In: *ICLR*. 2021.

[51] P. Skraba and K. Turner. *Wasserstein Stability for Persistence Diagrams*. 2023. arXiv: 2006.16824 [math.AT].

[52] C. M. Topaz, L. Ziegelmeier, and T. Halverson. "Topological Data Analysis of Biological Aggregation Models". In: *PLOS ONE* 10.5 (2015), pp. 1–26.

[53] R. Turkes, G. Montufar, and N. Otter. "On the Effectiveness of Persistent Homology". In: *NeurIPS*. 2022.

[54] S. Tymochko, E. Munch, and F. A. Khasawneh. "Using Zigzag Persistent Homology to Detect Hopf Bifurcations in Dynamical Systems". In: *Algorithms* 13.11 (2020), p. 278.

[55] A. Vadeboncoeur, Ö. D. Akyildiz, I. Kazlauskaite, M. Girolami, and F. Cirak. "Fully probabilistic deep models for forward and inverse problems in parametric PDEs". In: *J. Comput. Phys.* 491.C (2023).

[56] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet. "Novel type of phase transition in a system of self-driven particles". In: *Phys. Rev. Lett.* 75.6 (1995), pp. 1226–1229.

[57] L. Xian, H. Adams, C. M. Topaz, and L. Ziegelmeier. "Capturing dynamics of time-varying data via topology". In: *Found. Data Sci.* 4.1 (2022), pp. 1–36.

[58] S. Zeng, F. Graf, and R. Kwitt. "Latent SDEs on Homogeneous Spaces". In: *NeurIPS*. 2023.

[59] S. Zhang, M. Xiao, and H. Wang. "GPU-Accelerated Computation of Vietoris-Rips Persistence Barcodes". In: *SCG*. 2020.

[60] Q. Zhao, D. B. Lindbell, and G. Wetzstein. "Learning to Solve PDE-constrained Inverse Problems with Graph Networks". In: *ICML*. 2022.

[61] M. Zhong, J. Miller, and M. Maggioni. "Data-driven discovery of emergent behavior in collective dynamics". In: *Physica D* 411 (2020), p. 132542.

Supplementary material

A Simulation settings

As mentioned in Sec. 4.1, we create datasets of dynamic point clouds from simulations. All simulations are run with 200 points for 1,000 steps with a step size of 0.01, and every 10th step is taken as an observation. Only for the final ablation experiment of Sec. 4.3 (i.e., *Impact of the observation timeframe*), we simulate 20,000 time steps (of the D'Orsogna model) and extract training/testing sequences from this extended timeframe. The model parameters for these simulations are randomly sampled as specified below, and for each sampled parameter tuple, we create one simulation.

To create the `dorsogna-10k` dataset, we vary the following model parameters (see Fig. 3): *overall, we have* ***four*** *parameters that need to be predicted.* As macroscopic regimes mainly depend on the ratios C_r/C_a and l_r/l_a, cf. [18, Fig. 1], we fix $C_a = l_a = 1$ and sample $C_r = 2^{t_C}$, $l_r = 2^{t_l}$ with uniformly distributed $t_C \sim \mathcal{U}_{[-1,1]}$ and $t_l \sim \mathcal{U}_{[-1.5,0.5]}$. Similarly, we sample $\alpha = 2^{t_\alpha}$ with $t_\alpha \sim \mathcal{U}_{[-2,2]}$ and $m = 2^{t_m}$ with $t_m \sim \mathcal{U}_{[-2,2]}$. The model from the `SiSyPHE` library used to implement this simulation is `AttractionRepulsion`. Note, that the `SiSyPHE` library implements the mass m in terms of the interaction radius parameter R, i.e., $m = R^3$ (as we simulate point clouds in 3D).

Remark 4. *For comparability (and interpretability of the parameters) to the original D'Orsogna model from [18], we adjusted the* `AttractionRepulsion` *implementation of* `SiSyPHE` *to directly match [18, Eqs. (2) & (3)].*

The `dorsogna-1k` dataset is created by re-running the simulation provided as part of the public (Julia) implementation[3] of [23]. This dataset has ***two*** *parameters to predict.* Note that in [23], the authors simulated 500 sequences. For our work, we simulated 1,000 to have a larger dataset, but still one magnitude smaller than `dorsogna-10k`, `vicsek-10k` and `volex-10k`. For this dataset, particle masses are $m = 1$, propulsion is $\alpha = 1$, $C_a = l_a = 1$ and C_r, l_r vary uniformly in $[0.1, 2]$, and are selected if the generated point clouds satisfy a certain scale condition, leading to the parameter pairs illustrated in [23, Fig. 6].

For the `vicsek-10k` dataset, we sample R, c, ν uniformly from $\mathcal{U}_{[0.5,5]}$ and D from $\mathcal{U}_{[0,2]}$. *Overall, this gives* ***four*** *parameters that need to be predicted.*

Finally, for the `volex-10k` dataset, we sample ***four*** *parameters* as follows: $\alpha \sim \mathcal{U}_{[0,2]}$, interaction radii $R \sim \mathcal{U}_{[0,2]}$ and birth/death rates $\lambda_b, \lambda_d \sim \mathcal{U}_{[0,1]}$. However, for the latter two parameters, we discard settings where $\lambda_d > \lambda_b$ as, in this case, death rates are almost impossible to distinguish; also, we discard settings where $\lambda_b \gg \lambda_d$ as the resulting point cloud cardinalities exceed 2000 points during the simulation. The resulting (λ_b, λ_d) combinations are illustrated in Fig. 5.

[3] `https://github.com/ldarrick/paths-of-persistence-diagrams`

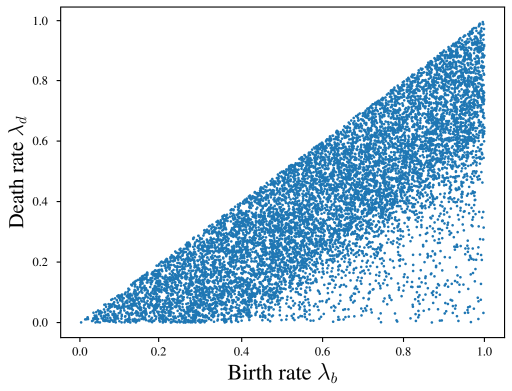

Figure 5: Birth rates λ_b and death rates λ_d used for generating the `volex-10k` dataset.

For reproducibility, we will release the simulation data publicly.

B Comparison(s) to prior work & Additional ablation experiments

B.1 Path Signature Kernel (PSK) [23]

For the PSK, we rely on the publicly available implementation in `sktime`[4] [38]. We compute the PSK (truncated at signature depth 3) using our vectorized persistence diagrams per time point as input. For concatenated zero-, one- and two-dimensional persistence diagrams, we use 20-dimensional vectorizations (as specified in Sec. 4.2), yielding 60-dimensional input vectors per time point. The computed kernel is then input to a kernel support vector regressor (kernel-SVR).

Following the experimental protocol in [23, Sec. 7.3], we cross-validate the sliding window embedding (across lags $[1, 2, 3]$) and kernel-SVR hyperparameters on a 20% validation portion of the training data, with hyperparameters selected based on the average MSE (per governing equation parameter β_i to be predicted) across 5-folds. Due to the excessive runtime (approx. 8.3 hours per lag on the system specified in Appendix C), we only report results when *all* time points per observation sequence are considered. Hence, *no subsampling* of time points (as is done when evaluating our approach) is performed, and the reported performance can be considered an *optimistic* estimate of what can be achieved with the PSK.

B.2 Crocker stacks [57]

As mentioned in Sec. 4.4, we compare against *crocker stacks*, introduced in [57], as one of the state-of-the-art approaches. Crocker stacks are an extension to *crocker plots* [52] and constitute a topological summary for time-varying persistence diagrams. For the crocker stacks, we adapted the publicly available implementation of the crocker plots in the `teaspoon`[5] library.

The crocker plot is computed as follows: for each time step, the persistence diagrams are computed up to a scale parameter ε. In discretized steps of the scale parameter, the Betti numbers are computed from the persistence diagrams, which results in a 2D representation (i.e., ϵ vs. Betti number) for each homology dimension.

The extension to crocker stacks is achieved by adding a third dimension k, induced by the smoothing factor α. For given steps of α, a smoothing operation is applied, i.e., values within a specified distance (smoothing factor) from the diagonal of the persistence diagram are ignored. Hence, for each homology dimension, a crocker stack is a tensor in $\mathbb{R}^3$, with axes corresponding to discretizations of the (1) scale parameter $\varepsilon \in [0, \infty)$, the (2) time $t \in [0, T]$, and the (3) smoothing factor $\alpha \in [0, \infty)$.

We note that in [57], the authors set the scale parameter to $\delta = 0.35$, as they use data normalized to the range of 0 to 1. We *did not* scale the data, however, we adjusted the scale parameter correspondingly:

[4] `https://github.com/sktime/sktime`
[5] `https://teaspoontda.github.io/teaspoon`

for all experiments, we set the scale parameter to $\delta = 1/3 \cdot \text{maxPers}(\text{dgm}_k(\text{Rips}(\mathcal{P})))$, where $\mathcal{P}$ denotes the point cloud, and $\text{maxPers}(\text{dgm}_k(\text{Rips}(\mathcal{P})))$ is the maximum persistence obtained from all point clouds in an observed sequence.

Hyperparameter choices. We made the following hyperparameter choices: (1) the Vietoris-Rips filtration is considered up to δ (see above), where the computations are discretized with 25 equally spaced values in $[0, \varepsilon]$; (2) the smoothing values are discretized with 18 steps equally spaced in $[0, 0.5 \cdot \text{maxPers}]$.

Eventually, the crocker stacks per homology dimension are vectorized (i.e., the tensor is flattened) and concatenated into a single vector per observation sequence. These vectorizations are then input to a linear support vector regressor (SVR). For each of the (varied) parameters in the governing equation of Fig. 3, a *separate* SVR is trained and evaluated. Hyperparameters of the SVR and the usage of preprocessing steps (feature scaling) are tuned using Bayesian optimization. Each parameter configuration is evaluated using a 5-fold cross-validation, and the best configuration is then used to train the final model on the full dataset.

B.3 Additional ablation experiments

To assess whether homology dimensions > 0 are beneficial to the downstream regression task, we experimented on `dorsogna-1k`. We find that when using our approach with H_0-features only, the regression quality drops. When additionally including H_2-features (i.e., H_0, H_1 and H_2), the situation is less clear, as the results are not noticeably different. Quantitative results can be found in Tbl. 5 below.

Table 5: Quantitative assessment of the impact of including higher-dimensional homology features on downstream regression performance, evaluated on `dorsogna-1k`.

	⊘ **VE** ↑	⊘ **SMAPE** ↓
Ours (PH-only, v1; H_0)	0.819 ± 0.015	0.101 ± 0.003
Ours (PH-only, v1; H_0, H_1)	0.844 ± 0.021	0.098 ± 0.007
Ours (PH-only, v1; $\text{H}_0, \text{H}_1, \text{H}_2$)	0.846 ± 0.011	0.097 ± 0.005

C Computational resources

All experiments were run on an Ubuntu Linux system (22.04), running kernel 5.15.0-100-generic, with 34 Intel® Core™ i9-10980XE CPU @ 3.00GHz cores, 128 GB of main memory, and two NVIDIA GeForce RTX 3090 GPUs.

D Runtime analysis

In the following, we present a runtime breakdown of the pre-processing steps (i.e., Vietoris-Rips persistent homology (PH) computation, and the vectorization of persistence diagrams), as well as a runtime comparison to prior work (*PSK* and *crocker stacks*) and our baseline approach. Runtime is measured on the system listed above, using our publicly-available reference implementation.

At this point, it is also worth highlighting that our pre-processing step, i.e., PH computation & vectorization, is trivially parallelizable and can easily be distributed across multiple CPUs/GPUs if needed.

Vietoris-Rips PH computation. In Tbl. 6, we list wall clock time measurements (using Ripser++ [59] on one GPU) per point cloud. In particular, the table lists runtime for computing PH of dimension zero and one (i.e., H_0 and H_1), as well as PH of dimension zero, one and two (i.e., H_0, H_1 and H_2). All point clouds in this experiment are of size 200 in $\mathbb{R}^3$.

Persistence diagram vectorization. Persistence diagram (PD) vectorization can essentially be broken down into two steps: (1) parameter fitting for the structure elements of [26] and (2) mapping PDs to vector representations using those structure elements. Tbl. 7 lists the runtime for both steps on the `dorsogna-1k` data when vectorizing zero-, and one-dimensional persistence diagrams. Throughout

Table 6: Runtime comparison for *PH computation* for different homology dimensions. Reported is the average runtime per point cloud (in seconds), and the overall runtime (in seconds) estimated from this average by multiplying by the number of processed point clouds on `dorsogna-1k`.

	⊘ **Time per point cloud**	**Overall**
PH computation $(\mathrm{H}_0, \mathrm{H}_1)$	0.018 s	1800 s
PH computation $(\mathrm{H}_0, \mathrm{H}_1, \mathrm{H}_2)$	0.330 s	33000 s

all of our experiments, parameter fitting for the structure elements is done by first collecting all persistence diagrams for each homology dimension and then running k-means++ clustering on 50,000 points uniformly sampled points from those diagrams.

Table 7: Runtime breakdown of *PD vectorization* (per diagram and overall for H_0 and H_1 on `dorsogna-1k`), split into (**Step 1**) time spent for fitting the parameters of the *exponential structure elements* from [26], and (**Step 2**) actually mapping PDs to vector representations.

	⊘ **Time per diagram**	**Overall**
Step 1: Parameter fitting	n/a	17 s
Step 2: Mapping PDs to vectors	9.4e-05	18 s

Comparison to prior work. Next, we compare the runtime of our approach (here, variant `v1` from Fig. 2) to prior work that uses persistence diagrams as input. In particular, we compare against the PSK method from [23] and crocker stacks [57].

Table 8: Training time comparison to prior work (on `dorsogna-1k`).

	Training time
Crocker Stacks	24600 s
PSK	646 s
Ours (PH-only, `v1`)	190 s

In Tbl. 8, we list the overall *training time* (on `dorsonga-1k`), where runtime for pre-processing (see above) is excluded from these measurements. Also, PSK and crocker stacks timings do include hyperparameter optimization, as suggested in the corresponding references. Importantly, this is not optional but required to obtain decent performance with respect to EV and SMAPE. Notably, for the PSK approach, kernel computation scales quadratically with the number of sequences, and kernel-SVR training takes time somewhere between quadratic and cubic. Hence, scaling up the number of training sequences quickly becomes computationally prohibitive, especially in light of the required hyperparameter tuning. Finding suitable hyperparameters is also the main bottleneck for crocker stacks (which rely on a linear SVR).

Comparison to baseline model. Finally, in Tbl. 9, we present a training time comparison to our *baseline model* which does not explicitly model any dynamics via a latent ODE (denoted as w/o dynamics). In this experiment, the training protocol remains unchanged, and we vary the type of input data, i.e., from using vectorized persistence diagrams only (H_0 and H_1) to using a combination of PointNet++ features *and* vectorized persistence diagrams. Importantly, as remarked in the main part of the manuscript, the baseline models (PH-only, `v1`) and (PH+PointNet++, `v3`) already yields strong performance across all parameter prediction problems.

Table 9: Training time comparison (on `dorsogna-1k`) of our approach vs. the baseline model (w/o dynamics), using the same input data.

	Training time
Ours (PH-only, `v1`)	190 s
Ours (PointNet++, `v2`)	3780 s
Ours (PH+PointNet++, `v3`)	4100 s
Baseline (w/o dynamics; PH-only, `v1`)	50 s
Baseline (w/o dynamics; PointNet++, `v2`)	525 s
Baseline (w/o dynamics; PH+PointNet++, `v3`)	600 s

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: *Claims made in the abstract are backed up by ablation experiments and comparisons to prior work & baselines.*

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: *Sec. 5 of the manuscript specifically addresses limitations of the proposed approach.*

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: *Our justification for the modeling choice relies on two previous stability results for persistent homology. When discussing the stability claim in Sec. 3, we list exact references to theorems in the literature that lead to this result.*

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: *We provide all experimental details in the main part of the paper and the appendix. Furthermore, source code (and data) is publicly-available at* `https://github.com/plus-rkwitt/neural_persistence_dynamics`.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: *We provide source code with this submission and full details about the (simulated) experimental data we use throughout. The code is released publicly and access to the already simulated data is provided to the community (via the code repository, see checklist item 4). Moreover, the data can easily be reproduced, as it is generated from publicly available simulation libraries, i.e.,* `https://github.com/antoinediez/Sisyphe` *and* `https://github.com/ldarrick/paths-of-persistence-diagrams`.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so No is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: *Sec. 4 lists all relevant hyperparameters for training. Exact architecture specifications are provided as part of the submitted source code.*

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: *All reported results include error bars ($\pm$ 1 standard deviation); Sec. 4.2 clearly states how these were computed. Furthermore, we assessed the statistical significance of any results in the listed Tables against the strongest obtained result in the respective Table using a Mann-Whitney test (correcting for multiple comparisons).*

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: *Computational resources and memory/compute requirements are listed in the appendix.*

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We follow the guidelines as specified.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: *Sec. 5 of the manuscript specifically addresses possible negative societal impact of the proposed approach. However, in view of the demonstrated improvements over the current state of the art it seems reasonable to expect a positive (societal) impact in related research areas.*

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: *Datasets that are part of our experimental studies and that we release after publication for reasons of reproducibility come exclusively from publicly accessible repositories as described in Section 4. Pretrained models will not be part of the submission.*

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: *The use of existing assets (i.e., code packages) was exclusively carried out with indication of the source in the manuscript.*

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: *All newly introduced assets have been carefully documented, i.e. details of the model have been described in Sec. 3 and limitations have been described in Sec. 5 of the manuscript. Furthermore, source code is attached to the submission and will be released publicly in case of acceptance (including datasets used in this work as well as a documentation).*

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: *The paper does not involve crowdsourcing nor research with human subjects.*

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: *The paper does not involve crowdsourcing nor research with human subjects.*

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.