TOPOLOGICALLY REGULARIZED DATA EMBEDDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised feature learning often finds low-dimensional embeddings that capture the structure of complex data. For tasks for which expert prior topological knowledge is available, incorporating this into the learned representation may lead to higher quality embeddings. For example, this may help one to embed the data into a given number of clusters, or to accommodate for noise that prevents one from deriving the distribution of the data over the model directly, which can then be learned more effectively. However, a general tool for integrating different prior topological knowledge into embeddings is lacking. Although differentiable topology layers have been recently developed that can (re)shape embeddings into prespecified topological models, they have two important limitations for representation learning, which we address in this paper. First, the currently suggested topological losses fail to represent simple models such as clusters and flares in a natural manner. Second, these losses neglect all original structural (such as neighborhood) information in the data that is useful for learning. We overcome these limitations by introducing a new set of topological losses, and proposing their usage as a way for *topologically regularizing* data embeddings to naturally represent a prespecified model. We include thorough experiments on synthetic and real data that highlight the usefulness and versatility of this approach, with applications ranging from modeling high-dimensional single-cell data, to graph embedding.

1 INTRODUCTION

Motivation Modern data often arrives in complex forms that complicate their analysis. For example, high-dimensional data cannot be visualized directly, whereas relational data such as graphs lack the natural vectorized structure required by various machine learning models (Bhagat et al., 2011; Kazemi & Poole, 2018; Goyal & Ferrara, 2018). Representation learning aims to derive mathematically and computationally convenient representations to process and learn from such data. However, obtaining an effective representation is often challenging, for example, due to the accumulation of noise in high-dimensional biological expression data (Vandaele et al., 2021b). In other examples such as community detection in social networks, graph embeddings struggle to clearly separate communities with interconnections between them. In such cases, expert prior knowledge of the topological model may improve learning from, visualizing, and interpreting the data. Unfortunately, a general tool for incorporating prior topological knowledge in representation learning is lacking.

In this paper, we introduce such tool under the name of *topological regularization*. Here, we build on the recently developed differentiation frameworks for optimizing data to capture topological properties of interest (Brüel-Gabrielsson et al., 2020; Solomon et al., 2021; Carriere et al., 2021). Unfortunately, such *topological optimization* has been poorly studied within the context of representation learning. For example, the used *topological losses* are indifferent to any structure other than topological, such as neighborhoods, which may be useful for learning. Therefore, topological optimization often destructs natural and informative properties of the data in favor of the topological loss.

Our proposed method of *topological regularization effectively resolves this by learning an embedding representation that incorporates the topological prior*. As we will see in this paper, these priors can be directly postulated through topological loss functions. For example, if the prior is that the data lies on a circular model, we design a loss function that is lower whenever a more prominent cycle is present in the embedding. By extending the previously suggested topological losses to fit a wider set of models, we show that topological regularization effectively embeds data according to a variety of topological priors, ranging from clusters, cycles, and flares, to any combination of these.

Related Work Certain methods that incorporate topological information into representation learning have already been developed. For example, Deep Embedded Clustering (Xie et al., 2016) simultaneously learns feature representations and cluster assignments using deep neural networks. Constrained embeddings of Euclidean data on spheres have also been studied by Bai et al. (2015). However, such methods often require extensive development for one particular kind of input data and topological model. Contrary to this, incorporating topological optimization into representation learning provides a simple yet versatile approach towards combining any data embedding method with topological priors, that generalizes well to any input data as long as the output is a point cloud.

Topological autoencoders (Moor et al., 2020) already combine topological optimization with a data embedding procedure. The main difference here is that the topological information used for optimization is obtained from the original high-dimensional data, and not passed as a prior. While this may sound as a major advantage—and certainly can be as shown by Moor et al. (2020)—obtaining such topological information heavily relies on distances between observations, which are often meaningless and unstable in high dimensions (Aggarwal et al., 2001). Furthermore, certain constructions such as the α-*filtration* obtained from the Delaunay triangulation—which we will use extensively and is further discussed in Appendix A—are expensive to obtain from high-dimensional data (Cignoni et al., 1998), and are therefore best computed from the low-dimensional embedding.

Our work builds on a series of recent papers (Brüel-Gabrielsson et al., 2020; Solomon et al., 2021; Carriere et al., 2021), which showed that topological optimization is possible in various settings and developed the mathematical foundation thereof. However, studying the use of topological optimization for data embedding and visualization applications, as well as the new losses we develop therefore and the insights we derive from them in this paper, are to best of our knowledge novel.

Contributions We include a sufficient background on *persistent homology*—the method behind topological optimization—in Appendix A (note that all of its concepts important for this paper are summarized in Figure 1). We summarize the previous idea behind topological optimization of point clouds (Section 2.1). We introduce a new set of losses to model a wider variety of shapes in a natural manner (Section 2.2). We show how these can be used to topologically regularize embedding methods for which the output is a point cloud (Equation (1)). We include experiments on synthetic and real data that show the usefulness and versatility of topological regularization, and provide additional insights into the performance of data embedding methods (Section 3 & Appendix B). We discuss open problems in topological representation learning and conclude on our work (Section 4).

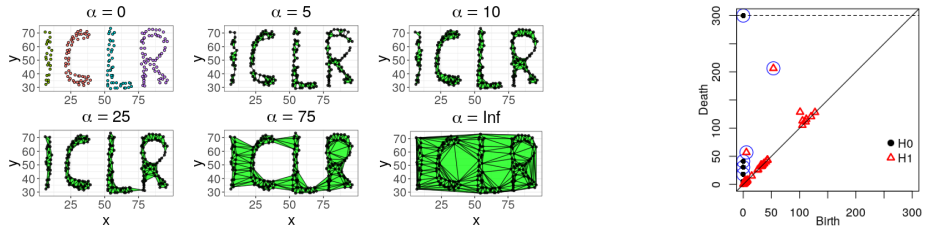
2 METHODS

The main purpose of this paper is to present a method to incorporate *prior topological knowledge* in a point cloud embedding $\boldsymbol{E}$ (dimensionality reduction, graph embedding, ...) of a data set $\mathbb{X}$. As will become clear below, these topological priors can be directly postulated through *topological loss functions* $\mathcal{L}_{\mathrm{top}}$. Then, the goal is to find an embedding that that minimizes a total loss

$$\mathcal{L}_{\mathrm{tot}}(\boldsymbol{E}, \mathbb{X}) \coloneqq \mathcal{L}_{\mathrm{emb}}(\boldsymbol{E}, \mathbb{X}) + \lambda_{\mathrm{top}} \mathcal{L}_{\mathrm{top}}(\boldsymbol{E}), \tag{1}$$

where $\mathcal{L}_{\mathrm{emb}}$ is a loss that aims to preserve structural attributes of the original data, and $\lambda_{\mathrm{top}} > 0$ controls the strength of *topological regularization*. Note that, $\mathbb{X}$ itself is not required to be a point cloud, or reside in the same space as $\boldsymbol{E}$, which is especially useful for representation learning.

In this section, we mainly focus on topological optimization of point clouds, that is, the loss $\mathcal{L}_{\mathrm{top}}$. The basic idea behind this recently introduced method—as presented by Brüel-Gabrielsson et al. (2020)—is illustrated in Section 2.1. We show that direct topological optimization may neglect important structural information such as neighborhoods, which can effectively be resolved through (1). Hence, as we will also see in Section 3, while representation learning may benefit from topological losses for incorporating prior topological knowledge, topological optimization itself may also benefit from other structural losses to represent the topological prior in a more truthful manner. Nevertheless, some topological models remain difficult to naturally represent through topological optimization. Therefore, we introduce a new set of topological losses, and provide an overview of how different models can be postulated through them in Section 2.2. Experiments with and comparisons to topological regularization of embeddings through (1) will be presented in Section 3.



(a) (b)

Figure 1: The two basic concepts from persistent homology important for our method. (a) Persistent homology quantifies topological changes in a *filtration*, i.e., a changing sequence of *simplicial complexes* ordered by inclusion, parameterized by a time parameter $\alpha \in \mathbb{R}$. Intuitively, the parameter α determines the scale at which points become connected (see Definition A.0.1 in Appendix A for a formal definition of the α-*complex*). Various topological holes are either born or die when α increases. The filtration starts with one connected component per point (*0-dimensional holes*), which can only merge together (resulting in their death) when including additional edges. For larger α, we observe the birth of cycles (*1-dimensional holes*, here 'white spaces surrounded by green'), which get filled in (and thus die) when α increases further. Eventually, one connected component persists indefinitely. (b) Persistent homology is commonly visualized through *persistence diagrams*. Here, two are plotted on top of each other (H0 and H1). A tuple (b, d) marks a hole—here a connected component (H0) or cycle (H1)—born at time b that dies at (possibly infinite) time d in the filtration. Prominently elevated points (encircled in blue) capture important topological properties, such as the four clusters 'I', 'C', 'L', and 'R' (H0), and the hole in the 'R', and between the 'C' and the 'L' (H1).

2.1 Background on Topological Optimization of Point Clouds

Topological optimization is performed through a *topological loss function* evaluated on one or more *persistence diagrams* (Barannikov, 1994; Carlsson, 2009). These diagrams—obtained through *persistent homology* as formally discussed in Appendix A—summarize all from the finest to coarsest topological holes (connected components, cycles, voids, ...) in the data, as illustrated in Figure 1.

While methods that learn from persistent homology are now both well developed and diverse (Pun et al., 2018), optimizing the data representation for the persistent homology thereof only gained recent attention (Brüel-Gabrielsson et al., 2020; Solomon et al., 2021; Carriere et al., 2021). Persistent homology has a rather abstract mathematical foundation within algebraic topology (Hatcher, 2002), and its computation is inherently combinatorial (Barannikov, 1994; Zomorodian & Carlsson, 2005). This complicates working with usual derivatives for optimization. To accommodate for this, topological optimization makes use of Clarke subderivatives (Clarke, 1990), whose applicability to persistence builds on arguments from o-minimal geometry (van den Dries, 1998; Carriere et al., 2021). Fortunately, thanks to the recent work of Brüel-Gabrielsson et al. (2020) and Carriere et al. (2021), powerful tools for topological optimization have been developed for software libraries such as PyTorch and TensorFlow, allowing their application without deeper knowledge of these subjects.

Topological optimization optimizes the data representation with respect to topological information summarized by its persistence diagram(s) $\mathcal{D}$. We will use the approach by Brüel-Gabrielsson et al. (2020), where (birth, death) tuples $(b_1, d_1), (b_2, d_2), \ldots, (b_{|\mathcal{D}|}, d_{|\mathcal{D}|})$ in $\mathcal{D}$ are first ordered by decreasing persistence $d_k - b_k$. The points (b, ∞), usually plotted on top of the diagram such as in Figure 1b, form the *essential part* of $\mathcal{D}$. The points with finite coordinates form the *regular part* of $\mathcal{D}$. In case of point clouds, one and only one topological hole, i.e., a connected component born at time $\alpha - 0$, will always persist indefinitely. Other gaps and holes will eventually be filled (Figure 1). Thus, we only optimize for the regular part in this paper. This is done through a *topological loss function*, which for a choice of $i \leq j$ (which, along with the dimension of topological hole, will specify our topological prior as we will see below) and a function $g : \mathbb{R}^2 \to \mathbb{R}$, is defined as

$$\mathcal{L}_{\text{top}}(\mathcal{D}) \coloneqq \sum_{k=i, d_k<\infty}^{j} g(b_k, d_k), \qquad \text{where } d_1 - b_1 \geq d_2 - b_2 \geq \ldots. \tag{2}$$

It turns out that for many useful definitions of g, $\mathcal{L}_{\text{top}}(\mathcal{D})$ has a well-defined Clarke subdifferential with respect to the parameters defining the filtration from which the persistence diagram $\mathcal{D}$ is ob-

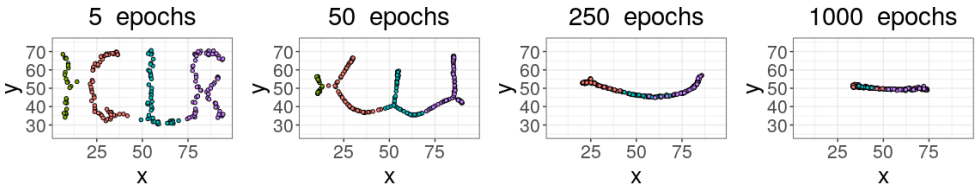


Figure 2: The data set in Figure 1a, optimized to have a low total 0-dimensional persistence. Points are colored according to their initial grouping along one of the four letters in the 'ICLR' acronym.

tained. In this paper, we will consistently use the α-*filtration* as shown in Figure 1a (see Appendix A for its formal definition), and these parameters are entire point clouds (in this paper embeddings) $\boldsymbol{E} \in (\mathbb{R}^d)^n$ of size n in the d-dimensional Euclidean space. $\mathcal{L}_{\mathrm{top}}(\mathcal{D})$ can then be easily optimized with respect to these parameters through stochastic subgradient algorithms (Carriere et al., 2021).

As it directly measures the prominence of topological holes, we let $g : \mathbb{R}^2 \to \mathbb{R} : (b, d) \mapsto \mu(d - b)$ be proportional to the *persistence function*. By ordering the points by persistence, $\mathcal{L}_{\mathrm{top}}$ is a *function of persistence*, i.e., it is invariant to permutations of the points in $\mathcal{D}$ (Carriere et al., 2021). The factor of proportionality $\mu \in \{1, -1\}$ indicates whether we want to *minimize* ($\mu = 1$) or *maximize* ($\mu = -1$) persistence, i.e, the prominence of topological holes. Thus, μ determines whether more clusters, cycles, …, should be present ($\mu = -1$) or missing ($\mu = 1$). The loss (2) then reduces to

$$\mathcal{L}_{\mathrm{top}}(\boldsymbol{E}) \coloneqq \mathcal{L}_{\mathrm{top}}(\mathcal{D}) = \mu \sum_{k=i, d_k < \infty}^{j} (d_k - b_k)\,, \qquad \text{where } d_1 - b_1 \geq d_2 - b_2 \geq \dots. \tag{3}$$

Here, the data matrix $\boldsymbol{E}$ (in this paper the embedding) defines the diagram $\mathcal{D}$ through persistent homology of the α-filtration of $\boldsymbol{E}$, and a persistence (topological hole) dimension to optimize for.

For example, consider (3) with $i = 2$, $j = \infty$, $\mu = 1$, restricted to 0-dimensional persistence (measuring the prominence of connected components) of the α-filtration. Figure 2 shows the data from Figure 1 optimized for this loss function for various epochs. The optimized point cloud quickly resembles a single connected component for smaller numbers of epochs. This is the single goal of the current loss, which neglects all other structural structural properties of the data such as its underlying cycles (e.g., the circular hole in the 'R') or local neighborhoods. Larger numbers of epochs mainly affect the scale of the data. While this scale has an absolute effect on the total persistence, thus, the loss, the point cloud visually represents a single connected topological component equally well. We also observe that while local neighborhoods are preserved well during the first epochs simply by nature of topological optimization, they are increasingly distorted for a larger number of epochs.

2.2 Newly Proposed Topological Loss Functions

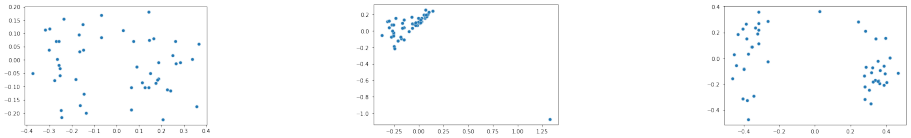
In this paper, the prior topological knowledge incorporated into the point cloud data matrix embedding $\boldsymbol{E}$ is directly postulated through a topological loss function. For example, letting $\mathcal{D}$ be the 0-dimensional persistence diagram of $\boldsymbol{E}$ (H0 in Figure 1b), and $i = 2$, $j = \infty$, and $\mu = 1$ in (3), corresponds to the criterion that $\boldsymbol{E}$ should represents one closely connected component, as illustrated in Figure 2. Therefore, we often regard a *topological loss* as a *topological prior*, and vice versa.

Although persistent homology effectively measures the prominence of topological holes, topological optimization is often ineffective for representing such holes in a natural manner, for example, through sufficiently many data points. An extreme example of this are clusters, despite the fact that they are captured through the simplest form of persistence, i.e., 0-dimensional. This is shown in Figure 3. Optimizing the point cloud in Figure 3a for (at least) two clusters can be done by defining $\mathcal{L}_{\mathrm{top}}(\boldsymbol{E})$ as in (3), letting $\mathcal{D}$ be the 0-dimensional persistence diagram of $\boldsymbol{E}$, $i = j = 2$, and $\mu = -1$. However, we observe that topological optimization simply displaces one point away from all other points (Figure 3b). Purely topologically, this is indeed a correct representation of two clusters.

To represent topological holes through more points, we propose topological optimization for the loss

$$\widetilde{\mathcal{L}}_{\mathrm{top}}(\boldsymbol{E}) \coloneqq \mathbb{E}\left(\mathcal{L}_{\mathrm{top}}\left(\{\boldsymbol{x} \in \mathbf{S} : \mathbf{S} \text{ is a random sample of } \boldsymbol{E} \text{ with sampling fraction } f_{\mathcal{S}}\}\right)\right), \tag{4}$$

where $\mathcal{L}_{\mathrm{top}}$ is defined as in (3). In practice, during each optimization iteration, $\widetilde{\mathcal{L}}_{\mathrm{top}}$ is approximated by the mean of $\mathcal{L}_{\mathrm{top}}$ evaluated over $n_{\mathcal{S}}$ random samples of $\boldsymbol{E}$. Figure 3 shows the result



(a) Initial point cloud. (b) Ordinary top. optimization. (c) Optimization with (4).

Figure 3: A point cloud topologically optimized for (at least) two clusters, without and with sampling. Optimization with sampling encourages topological holes to be represented by more points.

for a sampling fraction $f_{\mathcal{S}} = 0.1$ and repeats $n_{\mathcal{S}} = 1$. The new data representation visualizes the clusters already well and far more naturally. An added benefit of the new loss (4) is that topological optimization can be conducted significantly faster, as the α-filtration and persistent homology are evaluated on smaller samples (a computational analysis can be found at the end of Appendix A).

In summary, various topological priors can now be formulated through topological losses as follows.

***k*-dimensional holes** Optimizing for k-dimensional holes ($k = 0$ for clusters), can generally be done through (3) or (4), by letting $\mathcal{D}$ be the corresponding k-dimensional persistence diagram. The terms i and j in the summation are used to express how many holes one exactly, at least, or at most wants. Finally, μ can be chosen to either decrease ($\mu = 1$) or increase ($\mu = -1$) persistence.

Flares Persistent homology is invariant to certain topological changes. For example, both a linear 'I'-structured model and a bifurcating 'Y'-structured model consist of one connected component, and no higher-dimensional holes. These models are indistinguishable based on the (persistent) homology thereof, even though they are topologically different in terms of their singular points.

Capturing such additional topological phenomena is possible through a refinement of persistent homology under the name of *functional persistence*, also well discussed and illustrated by Carlsson (2014). The idea is that instead of evaluating persistent homology on a data matrix $\boldsymbol{E}$, we evaluate it on a subset $\{\boldsymbol{x} \in \boldsymbol{E} : f(\boldsymbol{x}) \leq \tau\}$ for a well chosen function $f : \boldsymbol{E} \to \mathbb{R}$ and hyperparameter τ.

Inspired by this approach, for a diagram $\mathcal{D}$ of a point cloud $\boldsymbol{E}$, we propose the topological loss

$$\widetilde{\mathcal{L}}_{\text{top}}(\boldsymbol{E}) \coloneqq \mathcal{L}_{\text{top}}\left(\{\boldsymbol{x} \in \boldsymbol{E} : f_{\boldsymbol{E}}(\boldsymbol{x}) \leq \tau\}\right), \text{ informally denoted } \left[\mathcal{L}_{\text{top}}(\mathcal{D})\right]_{f_{\boldsymbol{E}}^{-1}]-\infty,\tau]}, \tag{5}$$

where f is a real-valued function on $\boldsymbol{E}$, possibly dependent on $\boldsymbol{E}$—which changes during optimization—itself, τ a hyperparameter, and $\mathcal{L}_{\text{top}}$ is an ordinary topological loss as defined by (3). In particular, we will focus on the case where f equals a scaled centrality measure on $\boldsymbol{E}$:

$$f_{\boldsymbol{E}} \equiv \mathcal{E}_{\boldsymbol{E}} \coloneqq 1 - \frac{g_{\boldsymbol{E}}}{\max g_{\boldsymbol{E}}}, \text{ where } g_{\boldsymbol{E}}(\boldsymbol{x}) \coloneqq \left\| \boldsymbol{x} - \frac{1}{|\boldsymbol{E}|} \sum_{\boldsymbol{y} \in \boldsymbol{E}} \boldsymbol{y} \right\|. \tag{6}$$

For $\tau \geq 1$, $\widetilde{\mathcal{L}}_{\text{top}}(\boldsymbol{E}) = \mathcal{L}_{\text{top}}(\boldsymbol{E})$. For sufficiently small $\tau > 0$, $\widetilde{\mathcal{L}}_{\text{top}}$ evaluates $\mathcal{L}_{\text{top}}$ on the points 'far away' from the center of $\boldsymbol{E}$. As we will see in the experiments, this is especially useful in conjunction with 0-dimensional persistence to optimize for flares, i.e., star-structured shapes.

Combinations Naturally, through linear combination of loss functions, different topological priors can be combined, e.g., if we want the represented model to both be connected and include a cycle.

3 Experiments

In this section, we show how our proposed topological regularization of data embeddings (1) leads to a powerful and versatile approach for representation learning. In particular, we show that

- embeddings benefit from prior topological knowledge through topological regularization;
- conversely, topological optimization may also benefit from incorporating structural information as captured through embedding losses, leading to more qualitative representations;
- subsequent learning tasks may benefit from expert prior topological knowledge.

In Section 3.1, we show how topological regularization improves standard PCA dimensionality reduction and allows better understanding of its performance when noise is accumulated over many dimensions. In Section 3.2, we present applications to high-dimensional single-cell trajectory data and graph embedding. Quantitative results are discussed in Section 3.3.

Topological optimization was performed in Pytorch, using code adapted from Brüel-Gabrielsson et al. (2020). Data sizes, hyperparameters, losses, and optimization times are summarized in Tables 1 & 2. Code for this project is provided with our submission and will be made available on github.

Appendix B presents supplementary experiments that show topological regularization is consistent with earlier topological data analysis of the Harry Potter network (Section B.1), a domain application on real biological cells in the 'cell cycle' differentiation model (Section B.2), and how topological regularization reacts to different topological losses and hyperparameters (Section B.3).

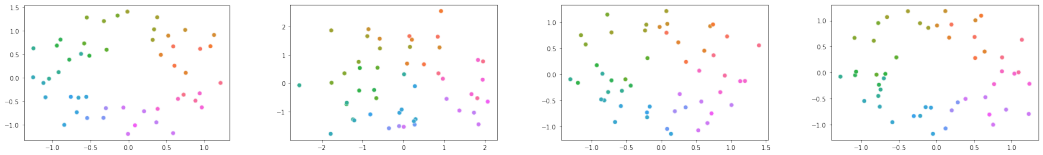
3.1 Synthetic Data

We sampled 50 points uniformly from the unit circle in $\mathbb{R}^2$. We then added 500-dimensional noise to the resulting data matrix $\boldsymbol{X}$, where the noise in each dimension is sampled uniformly from $[-0.45, 0.45]$. Since the additional noisy features are irrelevant to the topological (circular) model, an ideal projection embedding $\boldsymbol{X}$ is its restriction to its first two data coordinates (Figure 4a).

However, it is probabilistically unlikely that that the irrelevant features will have a zero contribution to a PCA embedding of the data (Figure 4b). Measuring the feature importance of each feature as the sum of its two absolute contributions (the loadings) to the projection, we observe that most of the 498 irrelevant features have a small nonzero effect on the PCA embedding (Figure 5). Intuitively, each added feature slightly shifts the projection plane away from the plane spanned by the first two coordinates. As a result, the circular hole is less prominent in the PCA embedding of the data. Note that *we observed this to be a notable problem for 'small n large p' data sets*, as similar to other machine learning models and as also recently studied by Vandaele et al. (2021a), more data can significantly accommodate for the effect of noise and result in a better embedding model on its own.

We can regularize this embedding using a topological loss function $\mathcal{L}_{\mathrm{top}}$ measuring the persistence of the most prominent 1-dimensional hole in the embedding ($i = j = 1$ in (3)). For a simple Pytorch compatible implementation, we used $\mathcal{L}_{\mathrm{emb}}(\boldsymbol{W}, \boldsymbol{X}) := \mathrm{MSE}\left(\boldsymbol{X}\boldsymbol{W}\boldsymbol{W}^T, \boldsymbol{X}\right)$, as to minimize the reconstruction error between $\boldsymbol{X}$ and its linear projection obtained through $\boldsymbol{W}$. To this, we added the loss $10^4\mathcal{L}_{\perp}(\boldsymbol{W})$, where $\mathcal{L}_{\perp}(\boldsymbol{W}) := \|\boldsymbol{W}^T\boldsymbol{W} - \boldsymbol{I}_2\|_2$ is used to encourage orthonormality of the matrix $\boldsymbol{W}$ to be optimized, initialized with the PCA-loadings. The resulting embedding is shown in Figure 4d (here $\mathcal{L}_{\perp}(\boldsymbol{W}) \sim 0.03$), which better captures the circular hole. Furthermore, we see that irrelevant features now more often contribute less to the embedding according to $\boldsymbol{W}$ (Figure 5).

For comparison, Figure 4c shows the optimization of $\boldsymbol{W}$ without accounting for the reconstruction loss ($\mathcal{L}_{\perp}(\boldsymbol{W})$ still included, here $\mathcal{L}_{\perp}(\boldsymbol{W}) \sim 0.2$). From this and also from Figure 5, we observe that $\boldsymbol{W}$ struggles more to converge to the correct projection, resulting in a less prominent hole.



(a) First two data coord. (b) Ordinary PCA emb. (c) Top. optimized proj. (d) Top. regularized emb.

Figure 4: Various representations of the synthetic data, colored by ground truth circular coordinates.

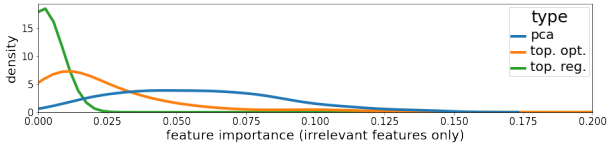
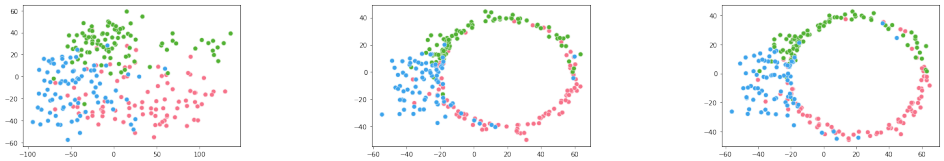
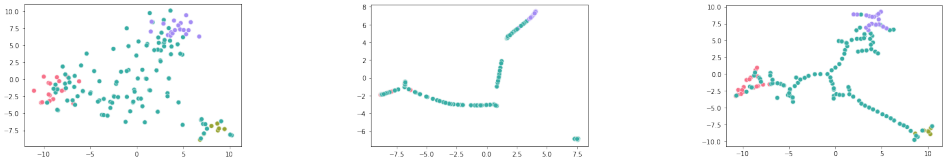


Figure 5: Feature importance densities of the 498 irrelevant features in the PCA embedding (blue), the top. optimized PCA embedding (orange), and the top. regularized PCA embedding (green).



(a) Ordinary PCA embedding. (b) Top. optimized projection. (c) Top. regularized embedding.

Figure 6: Various representations of the cyclic cell data. Colors represent the cell grouping.



(a) Ordinary UMAP embedding. (b) Top. optimized embedding. (c) Top. regularized embedding.

Figure 7: Various representations of the bifurcating cell data. Colors represent the cell grouping.

3.2 Real Data

Circular Cell Trajectory Data We considered a single-cell trajectory data set of 264 cells in a 6812-dimensional gene expression space (Cannoodt et al., 2018; Saelens et al., 2019). This data can be considered a snapshot of the cells at a fixed time. The ground truth is a circular model connecting three cell groups through cell differentiation (see also Section B.2). It has been shown by Vandaele (2020) that real single-cell data with such model are difficult to embed in a circular manner.

To explore this, we repeated the experiment with the same losses as in Section 3.1 on this data, where the (expected) topological loss is now modified through (4) with $f_{\mathcal{S}} = 0.25$, and $n_{\mathcal{S}} = 1$. From Figure 6a, we see that while the ordinary PCA embedding does somehow respect the positioning of the cell groups (marked by their color), it indeed struggles to embed the data in a manner that visualizes the present cycle. However, as shown in Figure 6c, by topologically regularizing the embedding (here $\mathcal{L}_{\perp}(\boldsymbol{W}) \sim 4e^{-3}$) we are able to embed the data much better in a circular manner.

Figure 6b shows the optimization of $\boldsymbol{W}$ without the reconstruction loss ($\mathcal{L}_{\perp}(\boldsymbol{W}) \sim 4e^{-3}$). The embedding is similar to the one in Figure 6c, with the pink colored cell group slightly more dispersed.

Bifurcating Cell Trajectory Data We considered a second cell trajectory data set of 154 cells in a 1770-dimensional expression space (Cannoodt et al., 2018). The ground truth here is a bifurcating model connecting four different cell groups through cell differentiation. However, this time we used the UMAP loss for the embeddings. We used a topological loss $\mathcal{L}_{\text{top}} \equiv \mathcal{L}_{\text{conn}} - \mathcal{L}_{\text{flare}}$, where $\mathcal{L}_{\text{conn}}$ measures the total (sum of) finite 0-dimensional persistence in the embedding to encourage connectedness of the representation, and $\mathcal{L}_{\text{flare}}$ is as in (5), measuring the persistence of the third most prominent 0-dimensional hole in $\{\boldsymbol{y} \in \boldsymbol{E} : \mathcal{E}_{\boldsymbol{E}}(\boldsymbol{y}) \leq 0.75\}$, where $\mathcal{E}_{\boldsymbol{E}}$ is as in (6). Thus, $\mathcal{L}_{\text{flare}}$ is used to optimize for a 'flare' with (at least) three clusters away from the embedding mean. We observe that while the ordinary UMAP embedding is more 'blobby' (Figure 7a), the topologically regularized embedding is more constrained towards a connected bifurcating shape (Figure 7c).

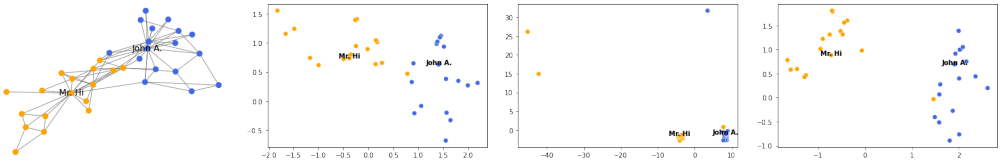
For comparison, we conducted topological optimization for the loss $\mathcal{L}_{\text{flare}}$ of the initialized UMAP embedding without the UMAP embedding loss. The resulting embedding is now more fragmented (Figure 7b). We thus see that topological optimization may also benefit from the embedding loss.

Graph Embedding The topological loss in (1) can be evaluated on any embedding, and does not require a point cloud as original input. We can thus use topological regularization for embedding a graph G, to learn a representation of the nodes of G in $\mathbb{R}^d$ that well respects properties of G.

To explore this, we considered the Karate network (Zachary, 1977), a well known and studied network within graph mining that consists of two different communities. The communities are represented by two key figures (John A. and Mr. Hi), as shown in Figure 8a. To embed the graph, we used a DeepWalk variant adapted from Dagar et al. (2020). While the ordinary DeepWalk em-

bedding (Figure 8b) well respects the ordering of points according to their communities, the two communities remained close to each other. We thus regularized this embedding for (at least) two clusters using the topological loss as defined by (4), where $\mathcal{L}_{\text{top}}$ measures the persistence of the second most prominent 0-dimensional hole, and $f_{\mathcal{S}} = 0.25$, $n_{\mathcal{S}} = 10$. The resulting embedding (Figure 8d) now nearly perfectly separates the two ground truth communities present in the graph.

Topological optimization of the initialized DeepWalk embedding with the same topological loss but without the DeepWalk loss creates some natural community structure, but also results in a few outliers (Figure 8c). Thus, although our introduced loss (4) enables more natural topological modeling to some extent, we again observe that using this in conjunction with embedding losses, i.e., our proposed method of topological regularization, leads to the best qualitative results.



(a) The Karate network. (b) Ordinary DeepWalk embedding. (c) Top. optimized embedding. (d) Top. regularized embedding.

Figure 8: The Karate network and various of its embeddings.

3.3 Quantitative Evaluation

Table 3 summarizes the embedding and topological losses we obtained for the ordinary embeddings, the topologically optimized embeddings (initialized with the ordinary embeddings, but not using the embedding loss), as well as for the topologically regularized embeddings. As one would expect, topological regularization balances the embedding losses between the embedding losses of the ordinary and topologically optimized embeddings. More interestingly, topological regularization may actually result in a more optimal, i.e., lower topological loss than topological optimization only, here in particular for the synthetic cycle data and Harry Potter graph. This suggest that combining topological information with other structural information may facilitate convergence to the correct embedding model, as we also qualitatively confirmed for these data sets (see also Section B.1). We also observe that there are more significant differences in the obtained topological losses than in the embedding losses with and without regularization. This suggests that the optimum region for the embedding loss may be somewhat flat with respect to the corresponding region for the topological loss. Thus, slight shifts in the local embedding optimum, e.g., as caused by noise, may result in much worse topological embedding models, which can be resolved through topological regularization.

Table 1: Summarization of the data, hyperparameters and optimization times.

data	size	method	lr	epochs	λ_{top}	t w/o top	t with top
Synthetic Cycle	50×500	PCA	1e-1	500	1e1	<1s	5s
Cell Cycle	264×6812	PCA	5e-4	1000	1e2	<1s	35s
Cell Bifurcating	154×1770	UMAP	1e-1	100	1e1	<1s	8s
Karate	34×78	DeepWalk	1e-2	50	5e1	29s	29s
Harry Potter	58×217	InnerProd	1e-1	100	1e-1	36s	34s

Table 2: Summary of the topological losses computed from persistence diagrams $\mathcal{D}$ with points (b_k, d_k) ordered by persistence $d_k - b_k$. Note that for 0-th dimensional homology diagrams $d_1 = \infty$.

data	top. loss function	dimension of hole	$f_{\mathcal{S}}$	$n_{\mathcal{S}}$
Synthetic Cycle	$-(d_1 - b_1)$	1	N/A	N/A
Cell Cycle	$-(d_1 - b_1)$	1	0.25	1
Cell Bifurcating	$\sum_{k=2}^{\infty}(d_k - b_k) - [d_3 - b_3]_{\mathcal{E}_{\boldsymbol{E}}^{-1}]-\infty,0.75]}$	0 - 0	N/A	N/A
Karate	$-(d_2 - b_2)$	0	0.25	10
Harry Potter	$-(d_1 - b_1)$	1	N/A	N/A

Table 3: Embedding/reconstruction and topological losses of the final embeddings. Lowest in bold.

data	embedding loss			topological loss		
	ord. emb.	top. opt.	top. reg.	ord. emb.	top. opt.	top. reg.
Synthetic Cycle	$\mathbf{6.3e^{-2}}$	$6.7e^{-2}$	$6.6e^{-2}$	-0.15	-0.35	$\mathbf{-0.75}$
Cell Cycle	$\mathbf{6.70}$	7.00	6.99	-13.4	$\mathbf{-50.9}$	-49.7
Cell Bifurcating	$\mathbf{8576}$	9933	8871	117	$\mathbf{23}$	63
Karate	$\mathbf{2006}$	N/A	2112	-1.3	$\mathbf{-28.5}$	-2.4
Harry Potter	$\mathbf{0.20}$	1.11	0.23	-0.82	-2.37	$\mathbf{-3.05}$

Table 4: Embedding performance evaluations for label prediction. Highest in bold.

data	metric	ord. emb.	top. opt.	top. reg.
Synthetic Cycle	r^2	0.56 ± 0.47	0.77 ± 0.24	$\mathbf{0.85 \pm 0.14}$
Cell Cycle	accuracy	0.79 ± 0.07	0.79 ± 0.07	$\mathbf{0.81 \pm 0.07}$
Cell Bifurcating	accuracy	0.79 ± 0.08	0.81 ± 0.07	$\mathbf{0.82 \pm 0.08}$
Karate	accuracy	$\mathbf{0.97 \pm 0.08}$	0.91 ± 0.14	$\mathbf{0.97 \pm 0.08}$

We evaluated the quality of the embedding visualizations presented in this section, by assessing how informative they are for predicting ground data truth labels. For the Synthetic Cycle data, these labels are the 2D coordinates of the noise-free data on the unit circle in $\mathbb{R}^2$, and we used a multi-ouput support vector regressor model. For the cell trajectory data and Karate network, we used the ground truth cell groupings and community assignments, respectively, and a support vector machine model. All points in the 2D embeddings were then split into 90% points for training and 10% for testing. Consecutively, we used 5-fold CV on the training data to tune the regularization hyperparameter $C \in \{1\mathrm{e}-2, 1\mathrm{e}-1, 1, 1\mathrm{e}1, 1\mathrm{e}2\}$. Other settings were the default from SCIKIT-LEARN. The performance of the final tuned and trained model was then evaluated on the test data, through the r^2 coefficient of determination for the regression problem, and the accuracy for all classification problems. The averaged test performance metrics and their standard deviations, obtained over 100 random train-test splits, are summarized in Table 4. Although topological regularization consistently leads to more informative visualization embeddings, quantitative differences may be minor, as we can see from the figures in this section that most local similarities are preserved with regularization.

4 Discussion and Conclusion

We proposed a new approach for representation learning under the name of *topological regularization*, which builds on the recently developed differentiation frameworks for topological optimization. This led to a versatile and effective way for embedding data according to expert prior topological knowledge, directly postulated through (some newly introduced) topological loss functions.

A clear limitation of topological regularization is that prior topological knowledge is not always available. How to select the best from a list of priors is thus open to further research (see also Section B.3). Furthermore, designing topological loss functions currently requires some understanding of persistent homology. It may be useful to study how to facilitate that design process for lay users. From a foundational perspective, our work provides new research opportunities into extending the developed theory for topological optimization (Carriere et al., 2021) to our newly introduced losses and their integration into data embeddings. Finally, topological optimization based on combinatorial structures other than the α-complex may be of theoretical and practical interest. For example, point cloud optimization based on graph-approximations such as the minimum spanning tree (Vandaele et al., 2021b), or varying the functional threshold τ in the loss (5) alongside the filtration time (Chazal et al., 2009), may lead to new topological loss functions with fewer hyperparameters.

Nevertheless, through our approach, we already provided new and important insights into the performance of embedding methods, such as their potential inability to converge to the correct topological model due to the flatness of the embedding loss near its (local) optimum, with respect to the topological loss. Furthermore, we showed that including prior topological knowledge provides a promising way to improve consecutive—even non-topological—learning tasks (see also Section B.2). In conclusion, topological regularization enables both improving and better understanding representation learning methods, for which we provided and thoroughly illustrated the first directions in this paper.

REFERENCES

Aaron Adcock, Erik Carlsson, and Gunnar Carlsson. The ring of algebraic functions on persistence bar codes. *arXiv preprint arXiv:1304.0530*, 2013.

Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pp. 420–434. Springer, 2001.

Nina Amenta, Dominique Attali, and Olivier Devillers. Complexity of delaunay triangulation for points on lower-dimensional polyhedra. In *Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms*, pp. 1106–1113, 2007.

Shuanghua Bai, Huo-Duo Qi, and Naihua Xiu. Constrained best euclidean distance embedding on a sphere: a matrix optimization approach. *SIAM Journal on Optimization*, 25(1):439–467, 2015.

Serguei Barannikov. The framed morse complex and its invariants. *American Mathematical Society Translations, Series 2*, 1994.

Smriti Bhagat, Graham Cormode, and S Muthukrishnan. Node classification in social networks. In *Social network data analytics*, pp. 115–148. Springer, 2011.

Jean-Daniel Boissonnat, Frédéric Chazal, and Mariette Yvinec. *Geometric and topological inference*, volume 57. Cambridge University Press, 2018.

Rickard Brüel-Gabrielsson, Bradley J Nelson, Anjan Dwaraknath, Primoz Skraba, Leonidas J Guibas, and Gunnar Carlsson. A topology layer for machine learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 1553–1563. PMLR, 2020.

Florian Buettner, Kedar N Natarajan, F Paolo Casale, Valentina Proserpio, Antonio Scialdone, Fabian J Theis, Sarah A Teichmann, John C Marioni, and Oliver Stegle. Computational analysis of cell-to-cell heterogeneity in single-cell rna-sequencing data reveals hidden subpopulations of cells. *Nature biotechnology*, 33(2):155–160, 2015.

Robrecht Cannoodt, Wouter Saelens, and Yvan Saeys. Computational methods for trajectory inference from single-cell transcriptomics. *European Journal of Immunology*, 46(11):2496–2506, nov 2016. ISSN 00142980.

Robrecht Cannoodt, Wouter Saelens, Helena Todorov, and Yvan Saeys. Single-cell -omics datasets containing a trajectory, October 2018. URL https://doi.org/10.5281/zenodo.1443566.

Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, jan 2009. ISSN 0273-0979.

Gunnar Carlsson. Topological pattern recognition for point cloud data. *Acta Numerica*, 23:289–368, 2014.

Mathieu Carriere, Frédéric Chazal, Marc Glisse, Yuichi Ike, Hariprasad Kannan, and Yuhei Umeda. Optimizing persistent homology based functions. In *International Conference on Machine Learning*, pp. 1294–1303. PMLR, 2021.

Frédéric Chazal, David Cohen-Steiner, Leonidas J Guibas, Facundo Mémoli, and Steve Y Oudot. Gromov-hausdorff stable signatures for shapes using persistence. In *Computer Graphics Forum*, volume 28, pp. 1393–1403. Wiley Online Library, 2009.

Paolo Cignoni, Claudio Montani, and Roberto Scopigno. Dewall: A fast divide and conquer delaunay triangulation algorithm in ed. *Computer-Aided Design*, 30(5):333–341, 1998.

Frank H Clarke. *Optimization and nonsmooth analysis*. SIAM, 1990.

A. Dagar, A. Pant, S. Gupta, and S. Chandel. graph_nets, 2020. URL https://github.com/dsgiitr/graph_nets.

Boris Delaunay et al. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800):1–2, 1934.

Olivier Devillers and Charles Duménil. A poisson sample of a smooth surface is a good sample. In *EuroCG 2019-35th European Workshop on Computational Geometry*, 2019.

Rex A Dwyer. Higher-dimensional voronoi diagrams in linear expected time. *Discrete & Computational Geometry*, 6(3):343–367, 1991.

Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

Allen Hatcher. *Algebraic topology*. Cambridge University Press, 2002. ISBN 0521795400.

Seyed Mehran Kazemi and David Poole. Simple embedding for link prediction in knowledge graphs. In *Advances in neural information processing systems*, pp. 4284–4295, 2018.

Giampaolo Luiz Libralon, André Carlos Ponce de Leon Ferreira, Ana Carolina Lorena, et al. Preprocessing for noise detection in gene expression classification data. *Journal of the Brazilian Computer Society*, 15(1):3–11, 2009.

Zehua Liu, Huazhe Lou, Kaikun Xie, Hao Wang, Ning Chen, Oscar M Aparicio, Michael Q Zhang, Rui Jiang, and Ting Chen. Reconstructing cell cycle pseudo time-series via single-cell transcriptome data. *Nature communications*, 8(1):1–9, 2017.

Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. Topological autoencoders. In *International conference on machine learning*, pp. 7045–7054. PMLR, 2020.

Nina Otter, Mason A. Porter, Ulrike Tillmann, Peter Grindrod, and Heather A. Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6(1):17, Aug 2017. ISSN 2193-1127.

Steve Y. Oudot. *Persistence Theory: From Quiver Representations to Data Analysis*. Number 209 in Mathematical Surveys and Monographs. American Mathematical Society, 2015.

Chi Seng Pun, Kelin Xia, and Si Xian Lee. Persistent-homology-based machine learning and its applications–a survey. *arXiv preprint arXiv:1811.00252*, 2018.

Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. Neural collaborative filtering vs. matrix factorization revisited. In *Fourteenth ACM Conference on Recommender Systems*, pp. 240–248, 2020.

Wouter Saelens, Robrecht Cannoodt, Helena Todorov, and Yvan Saeys. A comparison of single-cell trajectory inference methods. *Nature Biotechnology*, 37:1, 04 2019.

Yitzchak Solomon, Alexander Wagner, and Paul Bendich. A fast and robust method for global topological functional optimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 109–117. PMLR, 2021.

Eashwar V Somasundaram, Shael E Brown, Adam Litzler, Jacob G Scott, and Raoul R Wadhwa. Benchmarking r packages for calculation of persistent homology. *The R Journal*, 2021.

The GUDHI Project. *GUDHI User and Reference Manual*. GUDHI Editorial Board, 3.4.1 edition, 2021. URL `https://gudhi.inria.fr/doc/3.4.1/`.

Csaba D Toth, Joseph O'Rourke, and Jacob E Goodman. *Handbook of discrete and computational geometry*. CRC press, 2017.

Lou van den Dries. *Tame topology and o-minimal structures*, volume 248. Cambridge university press, 1998.

Robin Vandaele. Topological data analysis of metric graphs for evaluating cell trajectory data representations. Master's thesis, Ghent University, 2020.

Robin Vandaele, Yvan Saeys, and Tijl De Bie. Mining topological structure in graphs through forest representations. *Journal of Machine Learning Research*, 21(215):1–68, 2020.

Robin Vandaele, Bo Kang, Tijl De Bie, and Yvan Saeys. The curse revisited: a newly quantified concept of meaningful distances for learning from high-dimensional noisy data. *arXiv preprint arXiv:2109.10569*, 2021a.

Robin Vandaele, Bastian Rieck, Yvan Saeys, and Tijl De Bie. Stable topological signatures for metric trees through graph approximations. *Pattern Recognition Letters*, 147:85–92, 2021b.

Robin Vandaele, Yvan Saeys, and Tijl De Bie. Graph approximations to geodesics on metric graphs. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 7328–7334. IEEE, 2021c.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pp. 478–487. PMLR, 2016.

W.W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

Ruoyu Zhang, Gurinder S Atwal, and Wei Keat Lim. Noise regularization removes correlation artifacts in single-cell rna-seq data preprocessing. *Patterns*, 2(3):100211, 2021.

Afra Zomorodian and Gunnar Carlsson. Computing Persistent Homology. *Discrete & Computational Geometry*, 33(2):249–274, feb 2005. ISSN 0179-5376.

A INTRODUCTION TO PERSISTENT HOMOLOGY

In this part of the appendix, we provide a visual introduction to *persistent homology* and *persistence diagrams* that targets a broad machine learning audience. Although the fundamental results leading to persistent homology are rather abstract (Hatcher, 2002), its purpose and what it computes are rather easy to visualize.

A custom case in data science for illustrating how persistent homology works, is that we have a point cloud data set $\boldsymbol{X}$ embedded in a Euclidean space $\mathbb{R}^d$ (in the main paper, $\boldsymbol{X}$ is our embedding $\boldsymbol{E}$). In our case, we will take a look at the point cloud $\boldsymbol{X}$ that is visualized in the top left plot in Figure 9, which is the data set resembling the 'ICLR' acronym from the main paper. Note that Figure 9 is identical to Figure 1a from the main paper. The task is now to infer topological properties of the model underlying $\boldsymbol{X}$, by means of persistent homology.

Simplicial Complexes Looking at $\boldsymbol{X}$ (Figure 9, Top Left), the only topological information that can be deduced from it, is that it is a set of points, since no two point clouds of the same size can be topologically distinguished (at least according to any metric defined on them, which we assume to be the Euclidean distance metric here). Indeed, the displacement of isolated points in the Euclidean space, and more generally continuous stretchings and bendings, correspond to *homeomorphisms*, which are functions between topological spaces that preserve all topological information.

A partial solution to this can be obtained by constructing a *simplicial complex* from $\boldsymbol{X}$. In general, a simplicial complex can be seen as a generalization of a graph, where apart from nodes (0-simplices) and edges (1-simplices), it may also include triangles (2-simplices), tetrahedra (3-simplices), and so on. More specifically, the two defining properties of a simplicial complex Δ are that

- Δ is a set of finite subsets of some given set $\mathcal{S}$;
- If $\sigma' \subseteq \sigma \in \Delta$, then $\sigma' \in \Delta$

Each element σ in such a simplicial complex is called a *face* or *simplex*. If $|\sigma| = k + 1$, it is also called a k-simplex, and k is called the *dimension* of σ.

One should note that this is formally the definition of an *abstract* simplicial complex, and the term simplicial complex commonly refers to a *geometrical realization* of such a complex. Such realization can be seen as a continuous drawing of the complex in some Euclidean space $\mathbb{R}^d$, such that the intersection of two simplices σ and σ' corresponds to a common face of σ and σ'. This is similar to how a drawing of a graph (without intersecting edges) or a metric graph is a geometric realization of a graph (Vandaele et al., 2021c), but we also 'fill in' the 2-simplices, 3-simplices, ..., which is usually how we visualize (abstract) simplicial complexes, such as in Figure 9. Sometimes the term 'simplex' is only used to refer to geometric realizations of their discrete counterparts, and the dis-

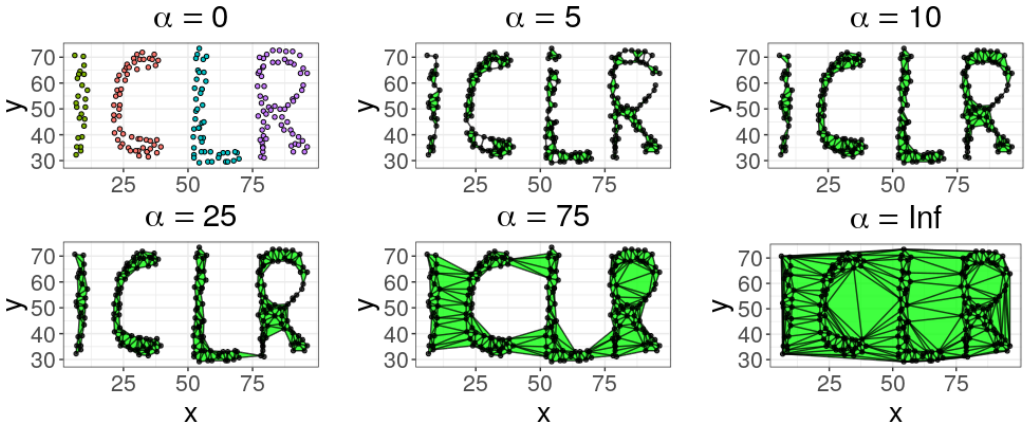


Figure 9: An example of six simplicial complexes $\Delta_\alpha(\boldsymbol{X})$ in the α-filtration filtration of a point cloud data set $\boldsymbol{X}$ resembling the 'ICLR' conference acronym in the Euclidean plane. $\alpha = \infty$ is informally used to denote the complex that contains all simplices, which here equals the Delaunay triangulation, and will necessarily be identical to $\Delta_\alpha(\boldsymbol{X})$ for some finite $\alpha \in \mathbb{R}_{>0}$.

crete counterparts are only referred to as 'faces'. However, for the purpose of this paper, it is not an issue for one to mix up this terminology, i.e., to identify simplicial complexes with their discrete counterparts, and we will often proceed to do this.

The most important thing to be aware of, is that (persistent) homology is concerned with topological properties of the 'continuous versions' (geometric realizations) of simplicial complexes, which are computed through their 'discrete' (abstract) counterparts. Compare this to how the connectedness of a graph (as a graph) determines the connectedness of any of its geometric realizations (as a topological space).

The α-complex When given a point cloud $\boldsymbol{X}$ (or in this paper an embedding $\boldsymbol{E}$), we consider the *α-complex* $\Delta_\alpha(\boldsymbol{X})$ at 'time' $\alpha \in \mathbb{R}_{\geq 0}$. This complex summarize topological information from the continuous model underlying the discrete point cloud $\boldsymbol{X}$ at the 'scale' determined by α. It is a *subcomplex* of the *Delaunay triangulation* (Delaunay et al., 1934) of $\boldsymbol{X}$, as shown in Figure 9 (the Delaunay triangulation equals the simplicial complex at the informally used time $\alpha = \infty$). Note that its definition below as a subset of the Delaunay triangulation (Boissonnat et al., 2018) can be more difficult to grasp than other simplicial complexes, such as the *Vietoris-Rips complex* (Otter et al., 2017), that are not used in this paper. However, this definition is not required to understand the basic ideas presented in the main paper, and is mainly included for completeness. What is most important is that the α-complex aims to quantify topological properties underlying $\boldsymbol{X}$, as is illustrated in Figure 9 and will be further discussed below. For a good overview and visualization of how the α-filtration is constructed, we refer the interested reader to The GUDHI Project (2021), from which we use the following definition.

Definition A.0.1 *(α-complex). Let $\boldsymbol{X}$ be a point cloud in the Euclidean space $\mathbb{R}^d$, and Δ the Delaunay triangulation of $\boldsymbol{X}$. For $\alpha = 0$, the α-complex $\Delta_{\alpha=0}(\boldsymbol{X})$ of $\boldsymbol{X}$ simply equals the set of points of $\boldsymbol{X}$ (thus a simplicial complex with only 0-simplices). More generally, to every simplex σ in Δ, we assign a time value α, which equals the square of the circumradius of σ if its circumsphere contains no other vertices than those in σ, in which case σ is said to be* Gabriel*, and as the minimum time values of the $(|\sigma| + 1)$-simplices containing σ that make it not Gabriel otherwise. For $\alpha > 0$, the α-complex $\Delta_\alpha(\boldsymbol{X})$ of $\boldsymbol{X}$ is now defined as the set of simplices in Δ with time value at most α.*

Topological Holes and Betti Numbers Once we have constructed a α-complex $\Delta_\alpha(\boldsymbol{X})$ from $\boldsymbol{X}$, we can now infer topological properties that are more interesting than $\boldsymbol{X}$ being a set of isolated points. For example, in Figure 9, we see that $\Delta_{10}(\boldsymbol{X})$ captures many important topological properties of the model underlying $\boldsymbol{X}$, which consists of four disconnected components (one for each of the letters 'I', 'C', 'L', and 'R' in the conference acronym), and also captures the circular hole that is present in the letter 'R'. *Homology* exactly quantifies such information by associating *Betti numbers* β_k to the simplicial complex. β_k corresponds to how many *k-dimensional holes* there are in the complex. In this sense, a 0-dimensional hole correspond to a gap between two components, and β_0 equals the number of connected components. A 1-dimensional hole corresponds to a loop (which can be seen as a circle, ring, or a handle of a coffee mug), and a 2-dimensional hole corresponds to a void (which can be seen as the inside of a balloon). In general, an n-dimensional hole corresponds to the interior of a n-sphere in $\mathbb{R}^{n+1}$. Note that no n-dimensional holes can occur in the Euclidean space $\mathbb{R}^d$ whenever $d \leq n$, and therefore (as well as for computational purposes) one often restricts the dimension k of the simplicial complexes that are constructed from the data.

Filtrations and α-Filtrations The difficulty lies in pinpointing an exact value for α for which $\Delta_\alpha(\boldsymbol{X})$ truthfully captures the topological properties of the model underlying $\boldsymbol{X}$. For example, $\Delta_{25}(\boldsymbol{X})$ still captures the hole in the 'R', but connects the points representing the letters 'L' and 'R' into one connected component. $\Delta_{75}(\boldsymbol{X})$ captures only one connected component, but also quantifies a prominent hole that is composed between the letters 'C' and 'R' which can also be seen as a particular topological characteristic of the underlying model. This is where 'persistent' homology comes into play. Rather than inferring these topological properties (holes) for one particular simplicial complex, the task of persistent homology is to track the change of these topological properties across a varying sequence of simplicial complexes

$$\Delta_0 \subseteq \Delta_1 \subseteq \ldots \subseteq \Delta_n,$$

which is termed a *filtration*. The definition of an α-simplex directly gives rise to such a filtration, termed the α-*filtration*:

$$(\Delta_\alpha(\boldsymbol{X}))_{\alpha \in \mathbb{R}_{\geq 0}}$$

Indeed, for $\alpha < \alpha'$, it follows directly from the definition of α-complexes that $\Delta_\alpha(\boldsymbol{X}) \subseteq \Delta_{\alpha'}(\boldsymbol{X})$. Note that in practice, for finite data, the simplicial complex across this filtration changes for only finitely many time values α, and we may indeed regard this filtration to be of the (finite) form $\Delta_0 \subseteq \Delta_1 \subseteq \ldots \subseteq \Delta_n$. The time parameter α is considered to parameterize the filtration, and is thus also termed the *filtration time*.

Persistent Homology When given a filtration parameterized by a time parameter t (in our case $t = \alpha$), one may observe changes in the topological holes when increasing the time parameter t. For example, as illustrated in Figure 9, different connected components may become connected through edges, or cycles may either appear or disappear. When a topological hole comes to existence in the filtration, we say that it is *born*. Vice versa, we say that a topological hole *dies* when it disappears. The filtration time at which these events occur are called the *birth time* and *death time*, respectively. Simply put, *persistent homology* tracks the birth and death times of topological holes across a filtration. 'Homology' refers to the topic within algebraic topology that considers the study and computation of these holes in topological models, and rests on (often abstract) concepts from mathematical fields such as group theory and linear algebra. As it may distract one from the main purpose of persistent homology we aim to introduce in our paper, we will not give such introduction in this paper, but we refer the interested reader to Hatcher (2002). 'Persistent' refers to the fact that we track homology, or thus topological holes, across an increasing filtration time parameter, and that persistent holes—those that remain present for many consecutive time values t—are commonly regarded to be informative for the (underlying) topological model in the data. This is the interpretation of persistent homology that we maintain for topological regularization, as well as for designing topological loss functions, in the main paper. However, more recent work shows that also finer topological holes that persist for short time intervals can be effective for machine learning applications, for example, for protein structure classification (Pun et al., 2018).

Persistence Diagrams *Persistence diagrams* are simply a tool to visualize the information quantified through persistent homology of a filtration. Formally, a persistence diagram is a set

$$\mathcal{D}_k = \underbrace{\{(t_{a_i}, \infty) : 1 \leq i \leq N\}}_{=:\mathcal{D}_k^{\text{ess}}} \cup \underbrace{\left\{(t_{b_j}, t_{d_j}) : 1 \leq j \leq M \wedge t_{b_j} < t_{d_j}\right\}}_{=:\mathcal{D}_k^{\text{reg}}} \cup \{(x, x) : x \in \mathbb{R}\},$$

where $t_{a_i}, t_{b_j}, t_{d_j}$, $1 \leq i \leq N$, $1 \leq j \leq M$, correspond to the birth and death times of k-dimensional holes across the filtration. Points (t_{a_i}, ∞), are usually displayed on top of the diagram. They correspond to holes that never die across the filtration, and form the *essential part* of the persistence diagram. In our case when using α-filtrations, we always have that $D_0^{\text{ess}} = \{(0, \infty)\}$, and $D_k^{\text{ess}} = \emptyset$ for $k \geq 1$, since eventually, the simplicial complex in the filtration will consist of one connected component that never dies, and any higher dimensional hole will be 'filled in', as illustrated by Figure 9. Hence, during topological regularization of data embeddings as discussed in our main paper, we cannot topologically optimize for the essential part of any persistence diagram of our embedding. The points (t_{b_j}, t_{d_j}) in $\mathcal{D}_k$ with finite coordinates $t_{b_j} < t_{d_j}$ form the *regular part* D_k^{reg} of the persistence diagram, which are the points for which we optimize in the main paper. Finally, the diagonal is included in a persistence diagram, as to allow for a well-defined distance metric between persistence diagram, called the *bottleneck distance*.

Definition A.0.2 *(Bottleneck Distance). Let $\mathcal{D}$ and $\mathcal{D}'$ be two persistence diagrams. The* bottleneck distance *between them is defined as*

$$d_{\text{b}}(\mathcal{D}, \mathcal{D}') \coloneqq \inf_{\varphi} \sup_{x} \|x - \varphi(x)\|_\infty \in \mathbb{R} \cup \{\infty\},$$

where φ ranges over all bijections from $\mathcal{D}$ to $\mathcal{D}'$, and x ranges over all points in $\mathcal{D}$. By convention, we let $\infty - \infty = 0$ when calculating the distance between two diagram points. Since persistence diagrams include the diagonal by definition, $|\mathcal{D}| = |\mathcal{D}'| = |\mathbb{R}|$. Thus, $d_{\text{b}}(\mathcal{D}, \mathcal{D}')$ is well-defined, as unmatched points in the essential or regular part of a diagram can always be matched to the diagonal.

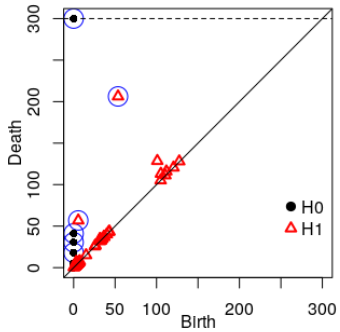


Figure 10: The diagrams $\mathcal{D}_0$ (black points, H0) and $\mathcal{D}_1$ (red points, H1) for the α-filtration filtration of the point cloud data set in Figure 9, plotted on top of each other. The six highly elevated points (encircled in blue) identify the presence of four connected components (H0), corresponding to the 'I', 'C', 'L', and 'R' letters in the conference acronym, and two cycles (H1), corresponding to the hole in the letter 'R' (born earlier) and composed between the letters 'R' and 'C' (born later).

The bottleneck distance commonly justifies the use of persistent homology as a stable method, i.e., robust to small data permutations, for quantifying topological information in data (Oudot, 2015). Note that although the bottleneck distance is not used in our main paper, we can also conduct topological optimization with respect to this metric (Carriere et al., 2021).

Figure 10 shows the persistence diagrams $\mathcal{D}_0$ and $\mathcal{D}_1$ of the α-filtration for our considered point cloud data $\boldsymbol{X}$ in Figure 9 (plotted on top of each other, which is common practice in topological data analysis). Note that all points in $\mathcal{D}_0$ have 0 as a first coordinate. This corresponds to the fact that all connected components are born at time $\alpha = 0$ in the α-filtration, at which time all points are added but no edges between them. $\mathcal{D}_0$ also has exactly one point equal to $(0, \infty)$ plotted on top of the diagram, which quantifies that the simplicial complex will eventually constitute one connected component that can never die, as illustrated by Figure 9. Birth times for points in $\mathcal{D}_1$ are nonzero, since edges are necessary for cycles to be present. They all have finite death-times, since they must be filled in at least at some point in time, as illustrated by Figure 9.

We see that in the combined diagrams, there are six prominently elevated points (b, d) for which the *persistence* $d - b$, measuring the vertical distance from the point to the diagonal, is significantly higher than all other points (encircled in blue in Figure 10). These quantify the prominent topological features in the model underlying the data $\boldsymbol{X}$ which we discussed earlier. For $\mathcal{D}_0$, these are the four connected components, one for each one of the letters 'I', 'C', 'L', and 'R'. Note that the single point $(0, \infty)$ on top may give a deceptive view that there is only one prominently elevated point in $\mathcal{D}_0$. However, there are as many points in $\mathcal{D}_0$ as data points in $\boldsymbol{X}$ (which may be difficult to deduce from Figure 10 due to overlapping points) and the four encircled points in $\mathcal{D}_0$ are indeed significantly more elevated. For $\mathcal{D}_1$, the two prominently elevated points quantify the hole in the 'R' and the hole between the letters 'C' and 'R'. Note that the birth time of the hole between the letters 'C' and 'L' is much later, as it occurs on a larger scale than the hole in the letter 'R', which occurs at the scale where the four clusters, one for each letter, occur as well (Figures 9 & 10).

Computational Cost Persistent Homology A remaining disadvantage of persistent homology is its computational cost. Although it is an unparalleled tool for quantifying all from the finest to coarsest topological holes in data, it relies on algebraic computations which can be costly for larger sized simplicial complexes. In worst case, these are cubic in the number of simplices (Otter et al., 2017).

For a data set $\boldsymbol{X} \subseteq \mathbb{R}^d$ of n points, the α-complex has size and computation complexity $\mathcal{O}\left(n^{\lceil \frac{d}{2} \rceil}\right)$ (Otter et al., 2017; Toth et al., 2017; Somasundaram et al., 2021), although this can be significantly lower in practice, for example, if the points are distributed nearly uniformly on a polyhedron (Amenta et al., 2007). An other popular type of filtration is the Vietoris-Rips filtration—for which topological optimization is also implemented by Brüel-Gabrielsson et al. (2020)—which has has size and computation complexity $\mathcal{O}(\min(2^n, n^{k+1}))$, where $d \geq k$ is the homology dimension of interest (Otter et al., 2017; Somasundaram et al., 2021). In practice, the choice of k will also signifi-

cantly reduce the size of the α-complex prior to persistent homology computation. However, the full complex, i.e., the Delaunay triangulation, still needs to be constructed first (Boissonnat et al., 2018), which is often the main bottleneck when working with α-complexes in high-dimensional data.

In practice, α-complexes are favorable for computing persistent homology from low dimensional point clouds, whereas fewer points in higher dimensions will favor Vietoris-Rips complexes (Somasundaram et al., 2021). Within the context of topological regularization and data embedding, we aim to achieve a low dimensionality d of the point cloud embedding. This justifies our choice for α-filtrations in the main paper.

Computational Cost Topological Loss Function in Main Paper When using a sampling fraction $0 < f_{\mathcal{S}} < 1$ with $n_{\mathcal{S}}$ repeats, the computational complexity of the topological loss function reduces to $\mathcal{O}\left(n_{\mathcal{S}}\left((f_{\mathcal{S}}n)^{3\lceil\frac{d}{2}\rceil}\right)\right)$ when using α-filtrations, where d is the embedding dimensionality. The added benefit of sampling is that some topological models can even be more naturally model, as shown in the main paper.

Nevertheless, as discussed above, this complexity may often be significantly lower in practice, due to constraining the dimension of homology for which we optimize, and that common distributions across natural shapes admit reduced size and time complexities of the Delaunay triangulations (Dwyer, 1991; Amenta et al., 2007; Devillers & Duménil, 2019).

Note that many computational improvements as well as approximation algorithms for persistent homology are already available (Otter et al., 2017). However, their integration into topological optimization is open to further research.

B Supplementary Experiments

In this part of the appendix, we include additional experiments that illustrate the usefulness and effectiveness of topologically regularization. Section B.1 includes an additional qualitative evaluation on the Harry Potter network, which shows that its topologically regularized graph embedding, optimized for a circular prior, is consistent with earlier and independent topological data analysis of this network (Vandaele et al., 2020). In Section B.2, we show how including expert prior topological information in the embedding procedure may facilitate automated pseudotime inference in our real cyclic single-cell expression data presented in the main paper. Finally, in Section B.3, we show how topological regularization reacts to different topological loss functions, either designed to model the same prior topological information, or different and potentially wrong prior information.

B.1 Additional Qualitative Evaluation on the Harry Potter Network

We considered an additional experiment on the Harry Potter graph obtained from `https://github.com/hzjken/character-network`. This graph is composed of characters from the Harry Potter novel (these constitute the nodes in the graph), and edges marking friendly relationships between them (Figure 11). Only the largest connected component is used. This graph has previously been analyzed by Vandaele et al. (2020), who identified a circular model therein that transitions between the 'good' and 'evil' characters from the novel.

We furthermore regularized this embedding using a topological loss function $\mathcal{L}_{\text{top}}$ that measures the persistence of the most prominent 1-dimensional hole in the embedding (see also Table 2 in the main paper), the result of which is shown in Figure 12c. Interestingly, the topologically regularized embedding now better captures the circularity of the model identified by Vandaele et al. (2020), and focuses more on distributing the characters along it. Note that although this model is included in the visualizations, it is not used to derive the embeddings, nor is it derived from them.

For comparison, Figure 12b shows the result of optimizing the initialized ordinary graph embedding for the same topological loss, but without the graph embedding loss. We observe that this results in a sparse enlarged cycle. Most characters are positioned poorly along the circular model, and concentrate near a small region. Interestingly, even though we only optimized for the topological loss here, it is actually less optimal, i.e., higher, than when we applied topological regularization (see also Table 3 in the main paper). This is also a result from the sparsity of the circle, which constitutes to a larger birth time, and thus a lower persistence, of the corresponding hole.

To embed the Harry Potter graph, we used a simple graph embedding model where the sigmoid of the inner product between embedded nodes captures the (Bernoulli) probability of an edge occurrence (Rendle et al., 2020). Thus, this probability will be high for nodes close to each other in the embedding, and low for more distant nodes. These probabilities are then optimized to match the binary edge indicator vector. Figure 12a shows the result of this embedding, along with the circular model presented by Vandaele et al. (2020). For clarity, character labels are only annotated for a subset of the nodes (the same as by Vandaele et al. (2020)).

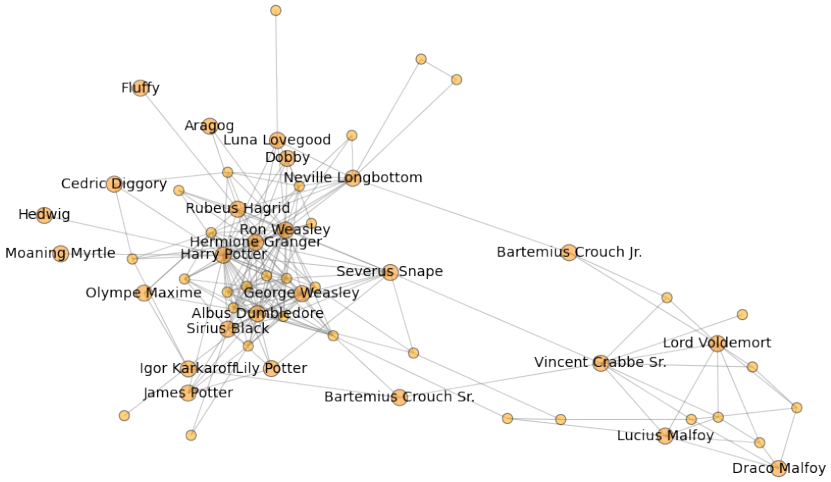
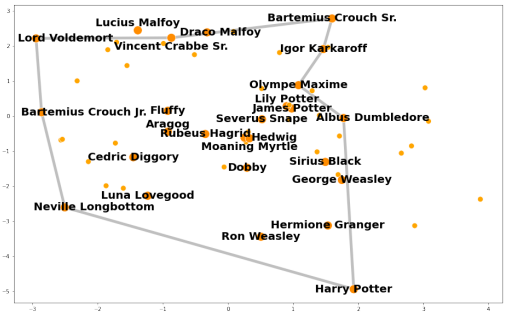
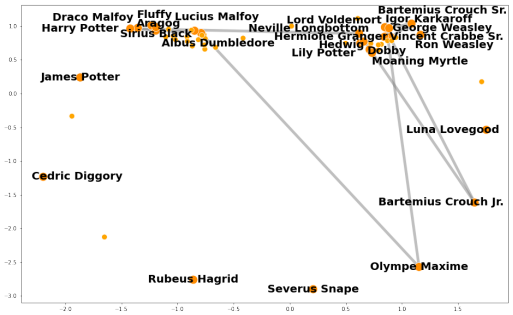


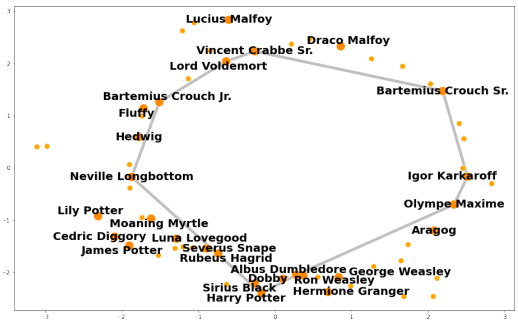
Figure 11: The major connected component in the Harry Potter graph. Edges mark friendly relationships between characters.



(a) Ordinary graph embedding.



(b) Topologically optimized embedding (initialized with the ordinary graph embedding).



(c) Topologically regularized embedding.

Figure 12: Various embeddings of the Harry Potter graph and the circular model therein.

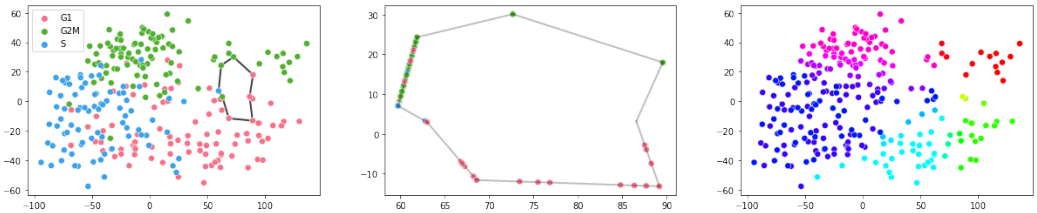
B.2 APPLICATION: PSEUDOTIME INFERENCE IN CELL TRAJECTORY DATA

Single-cell omics data include various types of data collected on cellular level, such as transcriptomics, proteomics and epigenomics data. Studying the topological model underlying the data may lead to better understanding of the dynamic processes of biological cells and the regulatory interactions involved therein. Such dynamic processes can be modeled through trajectory inference (TI) methods, also called *pseudotime analysis*, which order cells along a trajectory based on the similarities in their expression patterns (Saelens et al., 2019).

For example, the cell cycle is a well known biological differentiation model that takes place in a cell as it grows and divide. The cell cycle consists of different stages, namely growth (G1), DNA synthesis (S), growth and preparation for mitosis (G2), and mitosis (M), where the latter two are often grouped together in a G2M stage. Hence, by studying expression data of cells that follow such differentiation model, one may identify the genes involved in and between particular stages of the cell cycle (Liu et al., 2017). Pseudotime analysis allows such study by assigning to each cell a time during the differentiation process in which it occurs, and thus, the relative positioning of all cells within the cell cycle model.

Hence, *the analysis of single cell cycle data constitutes a problem where prior topological information is available*. As the signal-to-noise ratio is commonly low in high-dimensional expression data (Libralon et al., 2009; Zhang et al., 2021), this data is usually preprocessed through a dimensionality reduction method prior to automated pseudotime inference (Cannoodt et al., 2016; Saelens et al., 2019). *Topological regularization provides a tool to enhance the desired topological signal during the embedding procedure, and as such, facilitate automated inference that depends on this signal*.

To explore this, we used persistent homology for an automated (cell) cycle and pseudotime inference method, with and without topological regularization during the PCA embedding of the data. The data we use for this experiment is our real cell cycle data presented in the main paper, also previously analyzed by Buettner et al. (2015). The topological regularization procedure we followed here is the same as in the main paper, i.e., we regularized for a circular prior (see also Tables 1 & 2 in the main paper). Our automated pseudotime inference method now consists of the following steps.

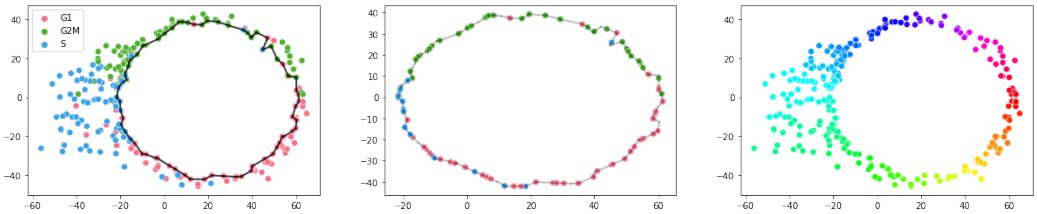


(a) The representation of the most prominent cycle obtained through persistent homology.

(b) Orthogonal projection of the embedded data onto the cycle representation.

(c) The pseudotimes inferred from the orthogonal projection, quantified on a continuous color scale.

Figure 13: Automated pseudotime inference of real cell cycle data through persistent homology, from the PCA embedding of the data.



(a) The representation of the most prominent cycle obtained through persistent homology.

(b) Orthogonal projection of the embedded data onto the cycle representation.

(c) The pseudotimes inferred from the orthogonal projection, quantified on a continuous color scale.

Figure 14: Automated pseudotime inference of real cell cycle data through persistent homology, from the topologically regularized PCA embedding of the data.

1. First, a representation of the most prominent cycle in the embedding is obtained through persistent homology from the α-filtration, using the Python software library Dionysus (`https://pypi.org/project/dionysus/`). It can be seen as a circular representation—discretized in edges between data points—of the point in the 1st-dimensional persistence diagram that corresponds to the most persisting cycle (Figures 13a & 14a).
2. An orthogonal projection of the embedded data onto the representative cycle is obtained. This is an intermediate step to derive continuous pseudotimes from a discretized topological representation, as earlier described by Saelens et al. (2019) (Figures 13b & 14b).
3. The lengths between points on the orthogonal projection are used to obtain circular pseudotimes between 0 and 2π (Figures 13c & 14c).

Note that within the scope of the current paper, we do not advocate this to be a new and generally applicable automated single cell cycle and pseudotime inference method. However, we chose this particular method because it illustrates well what persistent homology identifies in the data, and what we aim to optimize through topological regularization.

For example, we observe that the (representation of) the most prominent cycle in the ordinary PCA embedding is rather spurious, and mostly linearly separates G1 and G2M cells. Projecting cells onto the edges of this cycle, places the majority of the cells on a single edge (Figure 13b). The resulting pseudotimes are mostly continuous only for cells projected onto this edge, whereas they are very discretized for all other cells (Figure 13c). However, by incorporating our prior topological knowledge into the PCA embedding, the (representation of) the most prominent cycle in the embedding now much better characterizes the transmission between the G1, G2M, and S stages in the cell cycle model (Figure 14a). The automated procedure for pseudotime inference now also reflects a more continuous transmission between the cell stages (Figures 14b & 14c).

B.3 Topological Regularization for Different Loss Functions

In its current stage, postulating topological prior information is directly done by designing a topological loss function. This requires some understanding of persistent homology, and how to facilitate that design process for lay users is open to further research. Therefore, it may be useful to explore how topological regularization reacts to different (and potentially even wrong) prior topological information and loss functions, which is the topic of this section.

Different Loss Functions for the same Topological Prior (varying $f_{\mathcal{S}}, n_{\mathcal{S}}$, or τ) As we have introduced new topological loss functions (4) and (5) in the main paper, it may be useful to investigate how topological regularization reacts to different choices of the hyperparameters that are important for these losses. These are the sampling fraction $f_{\mathcal{S}}$ and number of repeats $n_{\mathcal{S}}$ in (4), and the functional threshold τ in (5). Note that although changing these hyperparameters may result in changing the *geometrical* characteristics that are specified, the *topological* characteristics may be considered to roughly remain the same with respect to the chosen topological loss function.

Varying $f_{\mathcal{S}}$ and $n_{\mathcal{S}}$ Figure 15 shows the topologically regularized PCA embedding of our real single-cell data following the cell cycle differentiation model from the main paper and also discussed in Section 13c, for varying choices of the values $f_{\mathcal{S}}$ and $n_{\mathcal{S}}$. The topological loss function and all other hyperparameters are identical to those in the main paper for this data (see also Tables 1 & 2 in the main paper).

We observe that both by increasing $f_{\mathcal{S}}$ and $n_{\mathcal{S}}$, the cycle tends to be more well defined, i.e., we can better visually deduce a subset of the data that captures a nearly perfect circle in the plane. Nevertheless, we also conclude that using more data during topological regularization, i.e., higher values of $f_{\mathcal{S}}$, does not necessarily result in better embeddings. Indeed, this gives a higher chance that the topological loss function will focus more on spurious holes during optimization, for example as illustrated by Figure 13a. From Figure 15—although more difficult to deduce due to overlapping points—we see that higher values of $f_{\mathcal{S}}$ generally result in slightly worse clustering, most notably distributing the pink colored cells (or as in Section B.2 discussed, G1 cells) more across the entire circular model.

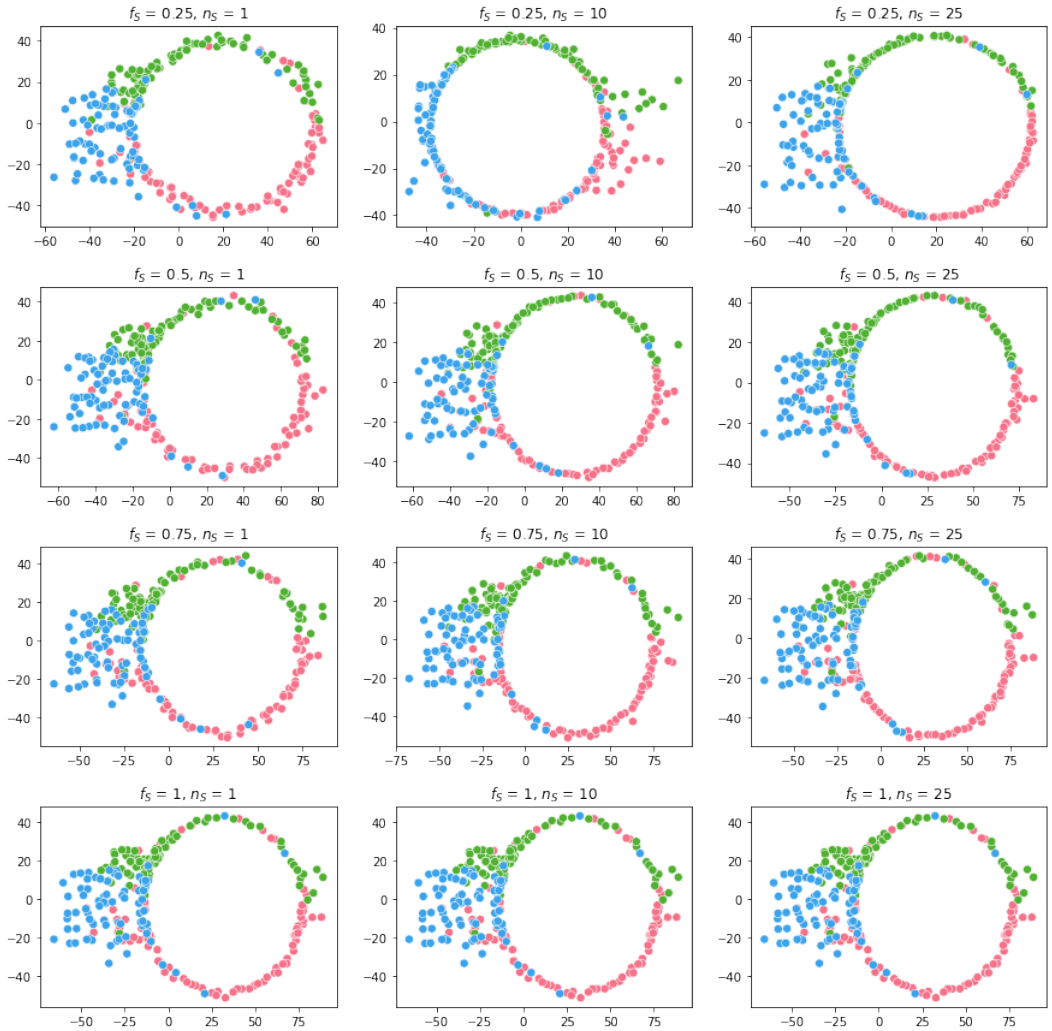


Figure 15: The topologically regularized PCA embedding of the cell cycle data set from the main paper and Section B.2, for varying sampling hyperparameters $f_{\mathcal{S}}$ (the sampling fraction) and $n_{\mathcal{S}}$ (the number of repeats). Note that the embeddings in the bottom row are all identical to $n_{\mathcal{S}} = 1$.

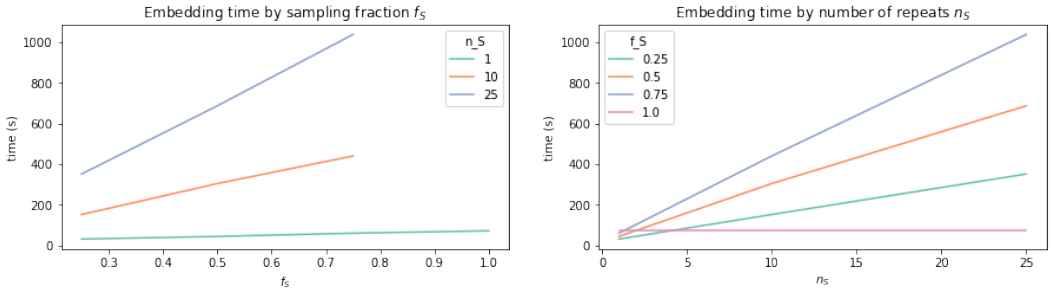


Figure 16: Total computation time of the topological regularized embedding in seconds (ran for 1000 epochs as in the main paper), by sampling fraction $f_{\mathcal{S}}$ and number of repeats $n_{\mathcal{S}}$. Note that for $f_{\mathcal{S}} = 1$ it is unnecessary to consider $n_{\mathcal{S}} > 1$, and the computation time is constant over $n_{\mathcal{S}}$. Therefore, computation times for $f_{\mathcal{S}} = 1$ and $n_{\mathcal{S}} > 1$ are also not included in the left plot, as they may give a deceptive view. During each step of the topologically regularized embedding optimization, the topological loss function is computed over $264 f_{\mathcal{S}}$ points, repeatedly when specified so by $n_{\mathcal{S}}$.

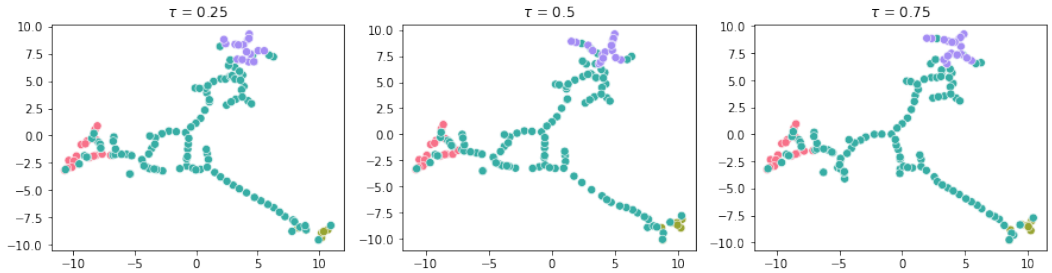


Figure 17: The topologically regularized UMAP embeddings of our bifurcating real single cell data presented in the main paper according to (8), for various functional thresholds τ.

An added advantage of our introduced topological loss function (4), which is approximated through a sampling strategy, is that we can use lower sampling fractions $f_{\mathcal{S}}$ not only to avoid optimizing for spurious holes, but also conduct this optimization significantly faster, as also discussed in our computational analysis in Section A. This can also be seen from Figure 16, which compares the computation times in seconds of the topologically regularized embeddings for different sampling fractions $f_{\mathcal{S}}$ and number of repeats $n_{\mathcal{S}}$.

Varying τ Figure 17 shows the topologically regularized UMAP embedding of our real single-cell data following a bifurcating cell differentiation model from the main paper, for varying choices of the functional threshold τ. Recall that $0 \leq \tau \leq 1$ was a threshold on the normalized distance of the embedded points to their mean as defined in (6) in the main paper. Intuitively, τ specified how close we can look to our embedding mean, with higher values of τ allowing more points (thus closer to the embedding mean) to be included when optimizing for three clusters away from the center.

This is consistent with what we observe from Figure 17. While little differences can be observed between $\tau = 0.25$ and $\tau = 0.5$, we observe that for $\tau = 0.75$ (which was the value chosen in our main paper), points near the center of bifurcation are more stretched towards the leaves, i.e., endpoints, in the embedded model. This agrees with the fact that for higher values of τ, more points that are close to the center are included when optimizing for three clusters away from the center.

Different Loss Functions for the same Topological Prior (varying g) For one particular type of prior topological information, we may also design different topological loss functions by varying the real-valued function g evaluated on the points in the persistence diagram in the loss (2) from the main paper. In particular, our topological loss function (3) in the main paper, where we let $g \coloneqq b_k - d_k$, is inspired by the topological loss function

$$\mathcal{L}_{\mathrm{top}}(\boldsymbol{E}) \coloneqq \mathcal{L}_{\mathrm{top}}(\mathcal{D}) = \mu \sum_{\substack{i=k \\ (b_k,d_k)\in\mathcal{D}^{\mathrm{reg}}}}^{|\mathcal{D}^{\mathrm{reg}}|} (d_k - b_k)^p \left(\frac{d_k + b_k}{2}\right)^q, \tag{7}$$

introduced by Brüel-Gabrielsson et al. (2020), and more formally investigated within an algebraic setting by Adcock et al. (2013). Here, p and q are hyperparameters that control the strength of penalization, whereas i and the choice of persistence diagram (or homology dimension) is used to postulate the topological information of interest. The choice of $\mu \in \{1, -1\}$ determines whether we want to increase ($\mu = -1$) or decrease ($\mu = 1$) the topological information for which we designed the topological loss function in the data. The term $(d_k - b_k)^p$ can be used to control the prominence, i.e., the persistence, of the most prominent topological holes, whereas the term $\left(\frac{d_k+b_k}{2}\right)^q$ can be used to control whether prominent topological features persist either early or later in the filtration.

Our topological loss function (3) in the main paper is identical to (7) with $p = 1$ and $q = 0$, and the additional change that we sum to the hyperparameter j instead of $|\mathcal{D}^{\mathrm{reg}}|$, which allows for even more flexible prior topological information. Note that this one of the most simple and intuitive, yet flexible, manners to postulate topological loss functions. Indeed, $(d_k - b_k)^{p=1}$ directly measures the the persistence of the k-th most prominent topological hole for a chosen dimension of interest. The role of $q \neq 0$ is to be further investigated within the context of topological regularization and formulating prior topological information.

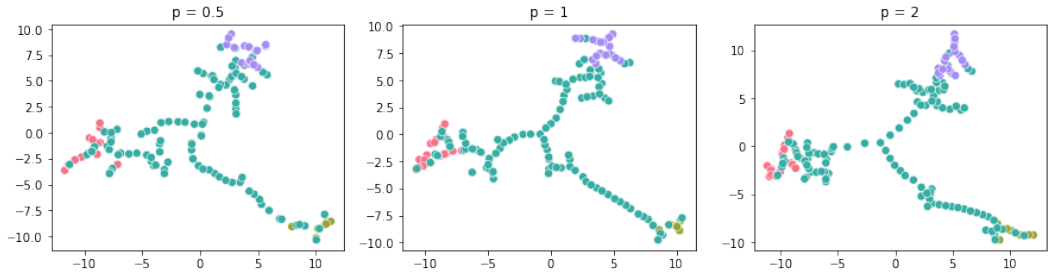


Figure 18: The topologically regularized UMAP embeddings of our bifurcating real single cell data presented in the main paper according to (8), for various values of p.

It may already be clear that the same topological information could be postulated through different nonzero choices for $p > 0$. To explore this, we considered two other topologically regularized UMAP embeddings of our real bifurcating single cell data presented in the main paper, for the same topological loss function where only the parameter p is varied, i.e.,

$$\mathcal{L}_{\text{top}}(\boldsymbol{E}) = \sum_{k=2}^{\infty}(d_k - b_k)^p - [(d_3 - b_3)^p]_{\mathcal{E}_{\boldsymbol{E}}^{-1}]-\infty,0.75]} \,. \tag{8}$$

Figure 18 shows the resulting embeddings for $p = \frac{1}{2}$ and $p = 2$, as well as the original embedding for $p = 1$ from the main paper for easy comparison.

In the first term of our topological loss (8), higher values of p will more strongly penalize larger gaps between neighboring points. This can be seen in Figure 18 *for the majority of the embedded points*, which are more pulled towards the leaves (endpoints) of the represented topological model. However, this results in a few more displaced points near the center of bifurcation, which try to remain connected as we simultaneously fit the UMAP loss function. Indeed, in the main paper we have seen that without the UMAP loss function, the model represented in the topologically optimized embedding will be more fragmented. Hence, in the current experiment, larger values of p in the term $\sum_{k=2}^{\infty}(d_k - b_k)^p$ seem to better motivate many smaller gaps (at the cost of a few larger gaps), whereas lower values appear to motivate more evenly spaced points. The fact that more points are pulled from the center towards the leaves for larger values of p and not from the leaves towards the center, is consistent with the fact that in the original high-dimensional space, more points are far away from each other and the center than they are close, as previously shown by Vandaele (2020) for this particular data set.

Nevertheless, although different values of p appear to affext the overall spacing between points, we see the embedded topological shape remains recognizable and consistent (Figure 18).

Different Loss Functions for different Topological Priors (example for PCA) As a user may not always have strong expert prior topological knowledge available, it may be useful to study how topological regularization reacts to different topological loss functions that are designed to model weaker or even wrong topological information. Indeed, if the impact of topological regularization on embeddings may be too strong, the user might wrongfully conclude that a particular topological model is present in the data when it can be easily fitted according to *any* specified shape.

To explore this, we considered various additional topological functions to topologically regulariz the PCA embedding of our synthetic data set of points lying on a circle with added noise in high dimension, presented in the main paper. Note that here we know that there is exactly one topological model, i.e., a circle, truthfully present in the data. We performed the topologically regularized PCA embedding of this data for the exact same hyperparameters as summarized in Table 1 in the main paper, but now for the three other topological loss functions as summarized in Table 5. The intuition of these topological loss functions, in order of appearance in Table 5, is as follows.

- The first topological loss function can be considered to specify a weaker form of prior topological information than in our original embedding, where we restricted to the term $-(d_1 - b_1)$. Now, the topological loss function states that the sum of persistence of the two

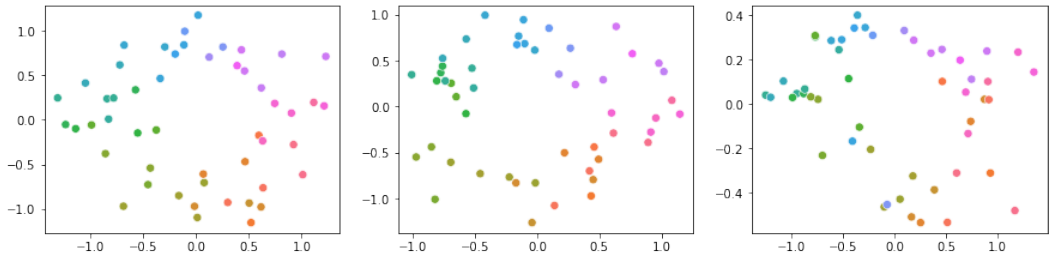


Figure 19: Various topologically regularized embeddings of the 500-dimensional synthetic data $\boldsymbol{X}$ for which the ground truth model is a circle, as presented in the main paper. (Left to Right) The topological loss function used for topological regularization of the PCA embedding equals the loss function formulated in Table 5 (Top to Bottom).

Table 5: Summary of the additional topological losses computed from persistence diagrams $\mathcal{D}$ with points (b_k, d_k) ordered by persistence $d_k - b_k$, used to study the topologically regularized PCA embedding of our Synthetic Cycle from the main paper for weaker or wrong topological information.

top. loss function	**dimension of hole**	$f_{\mathcal{S}}$	$n_{\mathcal{S}}$
$-(d_1 - b_1) - (d_2 - b_2)$	1	N/A	N/A
$-(d_2 - b_2)$	1	N/A	N/A
$\sum_{k=2}^{\infty}(d_k - b_k) - [d_3 - b_3]_{\mathcal{E}_{\boldsymbol{E}}^{-1}]-\infty,0.75]}$	0 - 0	N/A	N/A

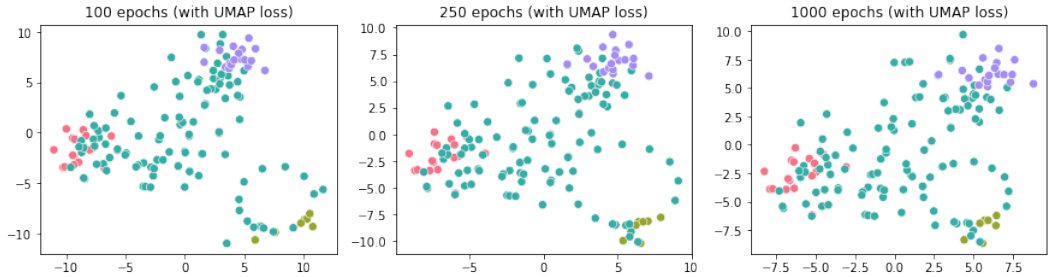
most prominent cycles in the embedding should be high as well. However, this does note impose that the persistence of the second most prominent cycle must be high.

- The second topological loss function can be considered to specify a partially wrong form of prior topological information. Here, we optimize for a high persistence of the second most prominent cycle in the embedding. Of course, this imposes that the persistence of the most prominent cycle in the embedding must be high as well. Note that according to the ground truth, only one (and thus the first most) prominent cycle should be present.
- Our third topological loss function can be considered to specify completely wrong prior topological information. Here, the topological loss function is exactly the one we used for our bifurcating real single cell data presented in the main paper, or thus, used for in Figure 18 ($p = 1$) in the Appendix. Hence, this topological loss function is used to optimize for a connected flare with at least three clusters away from the embedding center.

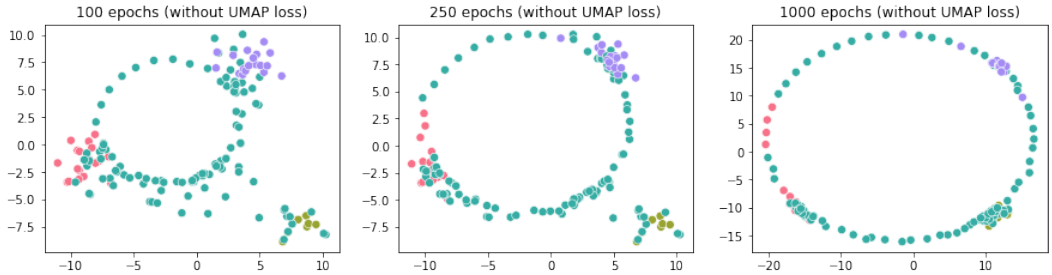
Figure 19 shows the topologically regularized PCA embeddings for these topological loss functions in order. While we observe small changes between the embeddings, which somewhat do agree with their respective designed topological loss function, in all cases, the embeddings do not strongly deviate from the ground truth topological model, which is one (and only one) prominent circle.

Different Loss Functions for different Topological Priors (example for UMAP) Note that in case of an orthogonal projection, which we encouraged in Figure 19 as we did in the main paper, it may be difficult to provide an embedding that well captures nontrivial topological models such as in Figure 18, when they are not truthfully present. However, non-linear dimensionality reduction methods such as UMAP may be more prone to model wrong prior topological information, as they directly optimize for the data point coordinates in the embedding space. To explore this, we considered an experiment where we embed our real bifurcating single cell data for a circular prior through the topological loss function $-(d_1 - b_1)$ evaluated on the 1st-dimensional persistence diagram. Thus, we topologically regularized the UMAP embedding of this data set for the exact same topological loss function as we originally used for our synthetic data lying on a circle (see also Figure 4 in the main paper). The result of topological regularization, using the exact same hyperparameters that we originally used for this data set as summarized in Table 1 in the main paper, is shown in Figure 20a.

We observe that when the embedding optimization is ran for 100 epochs (as was the embedding in the main paper), a cycle indeed starts to appear in the embedding. To further explore this, we increased the number of epochs to 1000, which showed to not increase the prominence of the cycle



(a) The topologically regularized UMAP embeddings.



(b) Topologically optimized embedding initialized with the ordinary UMAP embedding from the main paper. The UMAP loss function is not used during topological optimization here.

Figure 20: Topologically regularized and optimized UMAP embeddings of our bifurcating cell data, using potentially wrong prior topological (circular) information.

much further (Figure 20a). However, by omitting the UMAP loss during the same topological optimization procedure of the embedding, we observe that a cycle becomes much more prominently present in the embedding (Figure 20b). Hence, inclusion of the structural UMAP embedding loss function through topological regularization prevents the cycle to be prominently and falsely represented in the entire data, and the cycle remains spurious in the embedding.

Naturally, higher topological regularization strengths λ_{top} will more significantly impact the topological representation in the data embedding, so caution will need to be maintained when such information is absent. Figure 21 shows the topologically regularized UMAP embeddings for the circular prior for different topological regularization strengths λ_{top}. Here, we again ran for 1000 epochs to explore long-term effects of topological regularization during the optimization.

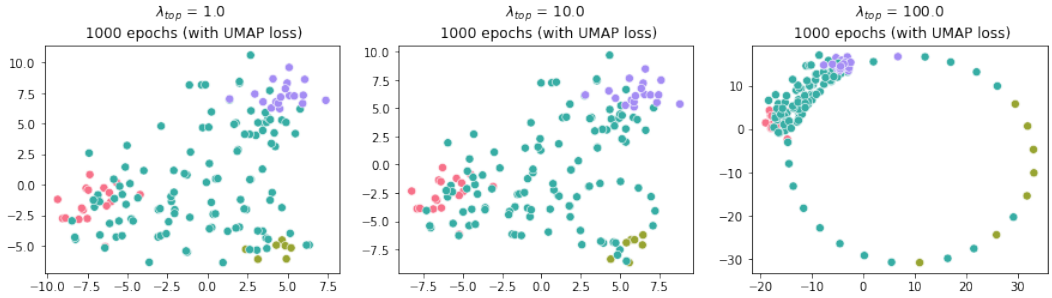


Figure 21: Topologically regularized and optimized UMAP embeddings of our bifurcating cell data, using potentially wrong prior topological (circular) information and for various topological regularization strengths λ_{top}. The choice of $\lambda_{\text{top}} = 10$ corresponds to the regularization strength in the main paper, which is included for easy comparison. For $\lambda_{\text{top}} = 10$, the plot is thus identical to the right plot in Figure 20a.

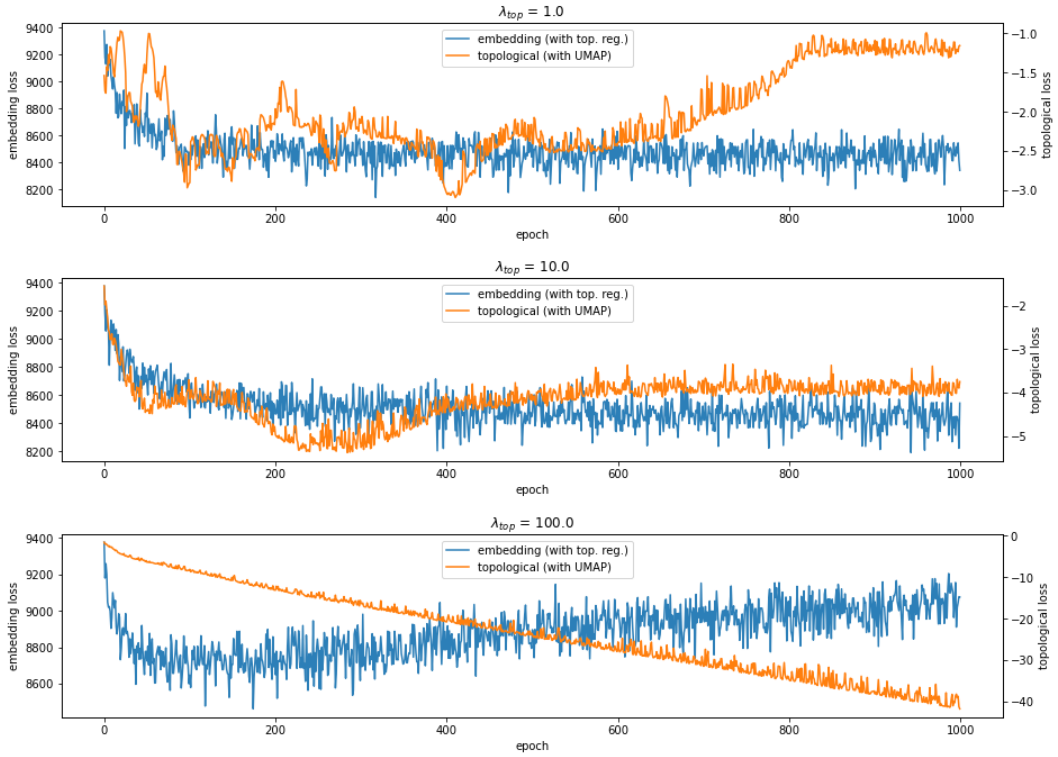


Figure 22: Evolution of the UMAP embedding and topological loss functions during optimization of the topologically regularized embeddings, according to the number of epochs and for various topological regularization strengths λ_{top}. The regularization strengths are not factored in the plotted topological loss functions for easy comparison. Thus, the topological loss functions directly equal the persistence of the most prominent cycle in the embeddings.

When multiplying our original regularization strength $\lambda_{\text{top}} = 10$ with a factor of 10, we see that the circular prior becomes much more prominently present in the topologically regularized UMAP embedding ($\lambda_{\text{top}} = 100$). For lower topological regularization strengths, the topological prior struggles to become represented in the embedded (for $\lambda_{\text{top}} = 10$ as in Figure 20a), or even appears to have no effect on the embedding at all (for $\lambda_{\text{top}} = 1$). This can also be observed from Figure 22, where the embedding and topological loss functions are plotted according to the number of epochs and for the different topological regularization strengths λ_{top}. We observe that for $\lambda_{\text{top}} \in \{1, 10\}$, the topological loss functions eventually increase again and stagnate. For $\lambda_{\text{top}} = 100$, the topological loss function appears to decrease indefinitely, at the cost of the embedding loss function which eventually increases again.

For all three cases, we also see that the incorrectly specified topological prior is not represented naturally in the data (Figure 21). For example, for $\lambda_{\text{top}} = 100$, the cycle is prominently present, but the majority of points remain clustered together and neglect the circular shape that we aim to model through our topological loss functions. This artefact of topological optimization as well as the evolution of the loss functions (which are more generally applicable for higher dimensional embeddings) may provide some visual feedback to the user that the bias to the topological prior may be too weak or too strong. How these observations can be quantified, as to validate the passed topological prior and tune hyperparameters such as topological the regularization strength in an automated manner, is open to further research.