Towards a Robust Persistence Diagram via Data-dependent Kernel

HANG ZHANG, National Key Laboratory for Novel Software Technology, School of Artificial Intelligence, Nanjing University, China
KAIFENG ZHANG, National Key Laboratory for Novel Software Technology, School of Artificial Intelligence, Nanjing University, China
KAI MING TING, National Key Laboratory for Novel Software Technology, School of Artificial Intelligence, Nanjing University, China
YE ZHU, School of IT, Deakin University, Australia

Topological Data Analysis (TDA) is used to extract topological features such as rings from point clouds. Recent works have identified that existing methods, which construct persistence diagrams in TDA, are not robust to noise and varied densities in a point cloud. This causes these methods to obtain incorrect topological features. We analyze the necessary properties of an approach that can address these two issues, and propose a new filter function for TDA based on a new data-dependent kernel that possesses these properties. Our empirical evaluation reveals that (i) the proposed kernel provides a better mean for UMAP dimensionality reduction (ii) the proposed filter function can significantly improve the performance of Topological Point Cloud Clustering (iii) the proposed filter function is a more effective way of constructing Persistence Diagram for t-SNE visualization and SVM classification than three existing methods of TDA, In addition, we explore the proposed filter's performance on a more complex deformation named Riemannian stretching. Our proposed filter equipped with Sample Fermat distance outperforms all the other filters when noise and Riemannian stretching coexist. Code is available at https://github.com/IsolationKernel/Codes/tree/main/Lambda-kernel.

JAIR Associate Editor: Sriraam Natarajan

JAIR Reference Format:
Hang Zhang, Kaifeng Zhang, Kai Ming Ting, and Ye Zhu. 2025. Towards a Robust Persistence Diagram via Data-dependent Kernel. *Journal of Artificial Intelligence Research* 84, Article 24 (December 2025), 41 pages. DOI: 10.1613/jair.1.17116

1 Introduction

Topological Data Analysis (TDA) is an important approach to studying the shape of a point cloud using techniques from topology (Wasserman 2018). An important and popular technique is persistent homology (PH) (Edelsbrunner et al. 2000), which typically produces a persistence diagram (PD) to describe the topological characteristics of a point cloud.

PH has been applied to many fields. In biology, it has been applied to (a) systematically study DNA structures, and has successfully discriminated three types of DNA (Meng et al. 2020); and (b) protein-ligand binding affinity

*Kaifeng Zhang and Hang Zhang contributed equally to this work.

Authors' Contact Information: Hang Zhang, ORCID: 0000-0003-0401-4066, zhanghang@lamda.nju.edu.cn, National Key Laboratory for Novel Software Technology, School of Artificial Intelligence, Nanjing University, Nanjing, China; Kaifeng Zhang, ORCID: 0009-0002-4296-6543, zhangkf2022@lamda.nju.edu.cn, National Key Laboratory for Novel Software Technology, School of Artificial Intelligence, Nanjing University, Nanjing, China; Kai Ming Ting, ORCID: 0000-0001-7892-6194, tingkm@nju.edu.cn, National Key Laboratory for Novel Software Technology, School of Artificial Intelligence, Nanjing University, Nanjing, China; Ye Zhu, ORCID: 0000-0003-4776-4932, ye.zhu@ieee.org, School of IT, Deakin University, Geelong, Australia.

DOI: 10.1613/jair.1.17116

prediction (Liu et al. 2022). In chemistry, it was employed to accelerate a first-principle screening for the discovery of the next generation of functionalized molecules and materials, such as screening a large database to discover the CO_2-philic functional groups (Townsend et al. 2020).

PD is beneficial for machine learning methods because it captures the topological structure information of a point cloud (Ali et al. 2023; Hensel et al. 2021) which cannot be obtained by other means. Recent efforts have been devoted to transforming a PD into some features, to be used as the input representation for machine learning algorithms (Chan et al. 2022; Hofer et al. 2019; Kusano et al. 2017). Most of the works have focused on different vectorization methods (Adams et al. 2017; Bubenik 2020; Carrière et al. 2017; Chevyrev et al. 2018; Kusano et al. 2017; Le and Yamada 2018; Polanco and Perea 2019; Reininghaus et al. 2015) and the similarity measure of PH (Reani and Bobrowski 2022) in order to improve the results of machine learning algorithms, with a strong assumption that a correct and good PD has been derived.

This work begins by asking a question: Do existing methods produce a correct and good PD? Unless we are certain that they do, any vectorization method and similarity measure used only propagate the errors introduced by PD, leading to poor final task-specific performance.

Existing works use a distance measure in their core computations, for example, (Zomorodian and Carlsson 2004). Some have identified shortcomings in using a distance measure (Berry and Sauer 2019; Blumberg and Lesnick 2022; Chazal, Fasy, et al. 2017; Vipond 2020; Vishwanath et al. 2020). There are two fundamental issues, i.e., PD is not robust to outliers (Chazal, Fasy, et al. 2017; Vishwanath et al. 2020) and varied densities (Berry and Sauer 2019) in a point cloud. Current research has provided some solutions to each of these issues independently.

The multi-parameter approach (Blumberg and Lesnick 2022) attempts to address both issues. But it still suffers from the same flaw of the one-parameter approach. No attempts have been made to address both issues in the one-parameter approach. To the best of our knowledge, we are the first to address both issues, i.e., when both noise and varied densities coexist in a point cloud.

The main contributions of this work are:

(1) Investigating the shortcomings of existing methods to produce correct and good PDs. And highlighting the importance of the problem of the lack of robustness of PD when noise and varied densities coexist in a point cloud.
(2) Proposing a new data-dependent kernel and a new filter to produce a PD in order to address the above problem.
(3) Showing that using a good PD improves the visualization outcome of t-Distributed Stochastic Neighbor Embedding (Van der Maaten and Hinton 2008) and the classification accuracy of Support Vector Machines (Hearst et al. 1998).
(4) Showing that our proposed data-dependent kernel can be integrated into UMAP (McInnes et al. 2018) and the performance of Topological Point Cloud Clsutering (TPCC) (Grande and Schaub 2023) can be significantly improved by replacing Rips with our proposed filter.

Comparison with our conference version. This paper is an extended version of our conference paper (Zhang et al. 2023), which only analyzes the robustness of Λ-filter to noise and varied densities caused by isotropic stretching and shrinking. The conference paper demonstrated the superiority of the proposed filter by combining PD and machine learning algorithms (t-SNE and SVM) to perform tasks at point cloud level, i.e., for a dataset containing many point clouds, we utilize PD as the feature of each point cloud. We extend it to demonstrate the superiority of the proposed filter by combining some topology-based algorithms (UMAP and TPCC) at point level, i.e., for a dataset consisting of individual points, we perform visualization or clustering. This paper expands the conference version with the following new contents:

(1) We establish the theoretical guarantee on Λ-filter's robustness against noise, as shown in Theorem 15.

Table 1. Key symbols and notations used.

X	A data set of $x \in \mathbb{R}^d$, where $\|X\| = n$
$\mathcal{X}$	Sample space of X
$\mathcal{D}$	A subset of X, where $\|\mathcal{D}\| = \psi$
Λ	A Voronoi Diagram
θ	A Voronoi cell in Λ
v_i	Center of Voronoi cell θ_i
κ	Kernel function
$\mathfrak{D}$	Persistence diagram
$\ell(\cdot,\cdot)$	Distance measure
$\ell_k(x)$	Distance to x's k-th nearest neighbor

(2) We extend the cause of varied densities from isotropic stretching to Riemannian stretching. Our proposed filter equipped with Fermat distance is the only filter that can produce a robust Persistence Diagram when noise and Riemannian stretching coexist, as shown in Section 9.
(3) We demonstrate Λ-kernel's application in dimensionality reduction (UMAP) and Λ-filter's application in clustering (TPCC) at point level, as shown in Section 5.2 and 8.2.

2 Background

Here we present the pertinent details on PH (see (Chazal and Michel 2021) for a detailed introduction). The key symbols and notations are shown in Table 1.

Let $f : \mathcal{X} \to \mathbb{R}^+$ be a function on metric space $(\mathcal{X}, \ell)$ and $X \subset \mathcal{X}$ be a parameter of f. At level $\epsilon > 0$, the sublevel set $\mathcal{X}_\epsilon^f = \{x \in \mathcal{X} \mid f(x) \le \epsilon\}$ encodes the topological information in $\mathcal{X}$. For $s \le \epsilon$, the sublevel sets are nested, i.e., $\mathcal{X}_s^f \subseteq \mathcal{X}_\epsilon^f$. Thus $\{\mathcal{X}_\epsilon^f\}_{0\le\epsilon<\infty}$ is a nested sequence of topological spaces, called a **filtration**, denoted by $Sub(f)$, and f is called the **filter function**.

The evolution of the topology (new cycles appear or existing cycles disappear through merging) is captured in the filtration as ϵ ranges from 0 to ∞, where the 0-dimensional cycle is connected component and the 1-dimensional cycle is loop/ring whose numbers are denoted as Betti numbers (Milnor 1964) b_0 and b_1, respectively. A cycle is said to have been born at $c \in \mathbb{R}$ when it starts to appear in $\mathcal{X}_c^f$; and it dies at $g \in \mathbb{R}$ when it no longer exists in $\mathcal{X}_p^f$ for any $p > g$. PH is an algebraic module that tracks the persistence pair (c, g) in $Sub(f)$. The persistence pairs (c, g) are shown in the form of Persistence Diagram (PD) $\mathfrak{D}(f)$, where c is the x-coordinate and $g > c$ is the y-coordinate. PD can be equivalently represented by Persistence Barcode (Carlsson, Zomorodian, et al. 2004), where (c, g) in PD corresponds to a bar ranging from c to g.

There is an equivalent way to perform filtration from a distance matrix, such as Rips filtration (Hausmann et al. 1995). In Rips filtration, Rips complex at ϵ is $VR_\epsilon(X) = \{\sigma \subset X \mid dia(\sigma) \le \epsilon\}$, where $X \subset \mathcal{X}$ is a finite point cloud and $dia(\sigma)$ represents the diameter of σ. This process is equivalent to connect $x, y \in X$ if $\ell(x, y) \le \epsilon$.

PH tracks the evolution of topology in $\{VR_\epsilon(X)\}_{0\le\epsilon<\infty}$. The same result can be obtained through a filter function $f(\cdot) = 2 \min_{x\in X} \ell(\cdot, x)$. An example is given in Figure 1.

3 Related Work

PH is a method that captures different topological features of a point cloud at different spatial resolutions. Features that persist over a wide range of spatial scales are regarded to be the true features of the underlying space, rather than the artifacts of sampling, noise, or specific selection of parameters. Note that not all short-persistence

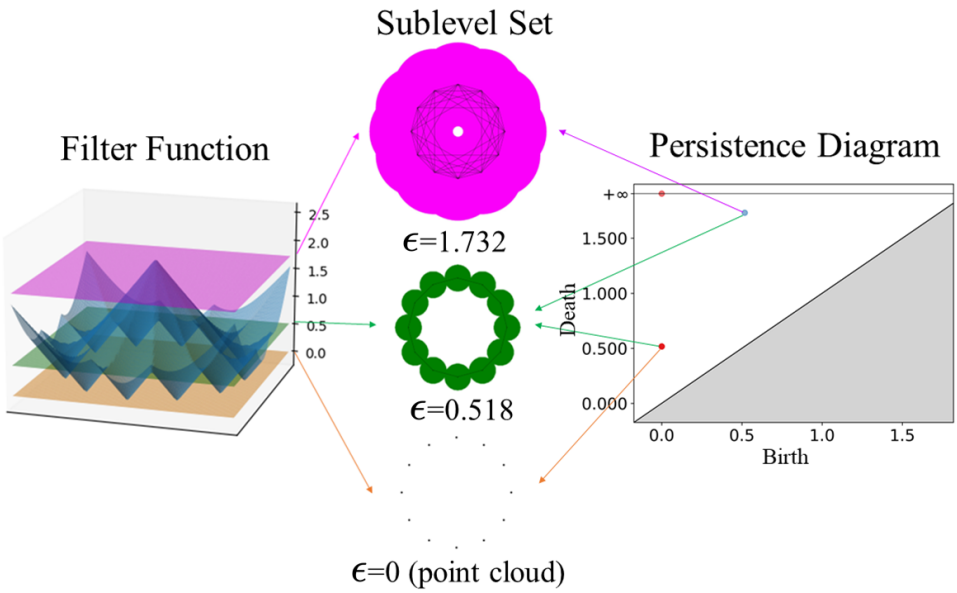


Fig. 1. An example filter function, three sublevel sets and a PD.

features are noise. There are short-persistence features which are important for some intended task, for example in protein structures (Xia and Wei 2014). However, these short-persistent features can be easily mistaken as noise if the measure is sensitive to noise. we aim to suppress noise and preserve the persistence of real features.

(Vietoris–)Rips (Hausmann et al. 1995) and Čech (Ghrist 2014) complexes are most commonly used in PH to produce a PD. However, the PD based on either (Vietoris–)Rips complex or Čech complex is sensitive to noise (Chazal, Fasy, et al. 2017; Lesnick and Wright 2015). Furthermore, both of them have trouble with varied densities (Berry and Sauer 2019; Blumberg and Lesnick 2022). As a result, they are not robust to the stretching and shrinking of a point cloud.

In order to be robust to noise, Distance-to-a-measure (DTM) is a commonly used approach to generate a robust PD from a point cloud with noise. PD is obtained via a distance-to-a-measure function which uses the average squared distances of k-nearest neighbors (Chazal, Fasy, et al. 2017). However, we found that DTM has difficulty in a point cloud with varied densities.

Several strategies have been proposed to address this issue, and can be divided into two frameworks: 1-parameter and multi-parameter frameworks. In the 1-parameter framework, fixing a density threshold and fixing the scale parameter are two commonly used approaches (Carlsson, Ishkhanov, et al. 2008; Chazal, Guibas, et al. 2013; Chazal, Guibas, et al. 2011). The multi-parameter filtration constructs a bi-filtration by using additional parameters, e.g., using density to capture the topological features of varied densities (Blumberg and Lesnick 2022; Carlsson and Zomorodian 2007; Frosini and Mulazzani 1999). Multi-parameter persistent homology can consider points of different densities separately, such as by codensity, thereby eliminating the influence of noise. However, even if points of different densities are considered separately, if the filter MPH uses, such as Rips filter, is data-independent, MPH will obtain different persistence when the point cloud is stretched or shrunk. In contrast, our data-dependent Λ-filter can obtain the same persistence. Detailed discussion and experimental demonstration are provided in Appendix B.6.

However, none of the approaches mentioned above can deal with the problem of varied densities satisfactorily: the detected feature in the dense region is still less persistent than that in the sparse region. Continuous k-nearest neighbors (CkNN) has been claimed to be the only approach to identify the 'correct topology' by a filtration based on k-nearest neighbors distance (Berry and Sauer 2019) as far as we know; and it belongs to the 1-parameter framework.

A common disadvantage of the above-mentioned approaches is that they cannot capture the correct topological features when noise and varied densities coexist in a point cloud. Because they are all proposed to address only one of the two issues.

We analyze the reasons why they cannot obtain a correct PD when noise and varied densities coexist in the following section.

4 How to Construct a Correct PD

In this section, we provide two requirements that an approach must satisfy in order to construct a correct PD. Based on these requirements, we analyze the reason why the existing approaches mentioned above cannot obtain a correct PD. Furthermore, we propose the necessary properties of a filter function in order to satisfy these requirements.

Proposition 1. *A correct PD must be robust to noise and varied densities.*

A correct PD must be robust to noise in order to determine the correct topological features because the topological feature can be severely affected by noise. Rips fails because it uses the nearest Euclidean distance, which is sensitive to noise. Improving upon Rips, DTM (Chazal, Fasy, et al. 2017) uses a more robust measure, which is realized as the average squared distances of k-nearest neighbors (k-nn). However, DTM is not robust to varied densities because the k-nn distance is easily affected by varied densities.

A correct PD must be robust to varied densities. With a point cloud with vary densities that caused by either stretching, shrinking, the PD produced should be the same before and after the data modification. This is because the topological features remain unchanged before and after stretching, shrinking. Rips and DTM are not robust to varied densities because they get large DTM values in sparse regions and small DTM values in dense regions. CkNN attempts to remove this effect by normalization: the distance between two points x, y is normalized by the square root of the product of x's and y's k-th nearest neighbor distances. However, CkNN is not robust to noise because the distance to the k-th nearest neighbor is sensitive to noise.

No existing filters can produce PD that is robust to both noise and varied densities, as far as we know.

To satisfy the requirements in Proposition 1, a filter function, which is used to produce PD, must have the following properties.

Property 2. ***Tolerant to noise.***
We say that a filter f is tolerant to noise if f produces approximately the same value from a finite point cloud X, with and without noise, i.e., $f(x|X) \approx f(x|X \cup X_n)$, where X_n is a set of noise points (defined in Definition 3), and $f(x|X)$ means the filter value of x based on the given X. The definition of noise is given as following:

Definition 3. *Let $\rho(\cdot)$ and $\rho_n(\cdot)$ be the probability density function (pdf) of $\mathcal{X}$ and $\mathcal{X}_n$, $X \subset \mathcal{X}_n$, where $\mathcal{X}$ is the sample space of normal points, and $\mathcal{X}_n$ is the sample space of noise points. $X \in \mathcal{X}$ and $X_n \in \mathcal{X}_n$ are sampled according to $\rho(x)$ and $\rho_n(x)$ respectively. $\gamma = \frac{|X_n|}{|X|}$ controls the noise level.*

$\rho_n(X)$ is set to be a uniform distribution in our experiments.

Let a finite point cloud S sampled from $\mathcal{S}$ be obtained from another finite point cloud T sampled from $\mathcal{T}$ through stretching(shrinking) $m : \mathcal{T} \to \mathcal{S}$, s.t. $\forall x, y \in \mathcal{T}, \ell(m(x), m(y)) = r \cdot \ell(x, y)$, where stretch factor $r > 1$ (stretching) or $r < 1$ (shrinking). This definition of stretching/shrinking is applicable globally or locally in a

point cloud. $m(\cdot)$ is referred as isotropic stretching if $\ell(\cdot,\cdot)$ is Euclidean distance. A more complex Riemannian stretching is defined in Section 9.2.

Property 4. ***Tolerant to varied densities.***
If sample space $\mathcal{S}$ is a stretched/shrunken version of $\mathcal{T}$, i.e., $\mathcal{S}$ is sparser/denser than $\mathcal{T}$, we say that a filter f is tolerant to varied densities if $\forall x \in \mathcal{S}, y = m(x) \in \mathcal{T}, f(x|S) = f(y|T)$, where S, T are sampled from $\mathcal{S}, \mathcal{T}$ respectively.

According to CkNN (Berry and Sauer 2019), a filter f being tolerant to varied densities implies that f is capable of capturing multi-scale geometry.

5 The Data-dependent Λ-kernel

Motivated by Properties 2 and 4, we propose a new data-dependent kernel and its associated new filter function which produces PDs that conform to Proposition 1.

We propose a new kernel called Λ-kernel as a similarity measure, which is based on Voronoi Diagram. Voronoi Diagram has been widely used in geometric and topological analyses because it has many favorable topological properties. For example, it forms small partitions in high-density regions and large partitions in low-density regions (Aurenhammer 1991; Aurenhammer and Klein 2000; Okabe et al. 2000).

5.1 Definition and Properties

Let X be a finite point cloud sampled from $\mathcal{X} \subseteq \mathbb{R}^d$, and $\mathcal{D} = \{v_1, v_2, ..., v_\psi\}$ be sampled uniformly from X. A Voronoi Diagram Λ constructed from $\mathcal{D}$ has Voronoi cell θ_i centred at $v_i \in \mathcal{D}$ defined as follows:

$$\theta_i = \left\{ x \in \mathcal{X} \mid v_i = \arg\min_{v \in \mathcal{D}} \ell(x, v) \right\}$$

An example of Voronoi Diagram is shown in Figure 2a. Λ-mapping $\Phi : \mathbb{R}^d \to \mathbb{R}^\psi$ is defined as:

$$\Phi(x \mid \Lambda) = \left[\frac{e^{-\eta\ell(x,v_1)}}{\Upsilon_\ell}, ..., \frac{e^{-\eta\ell(x,v_\psi)}}{\Upsilon_\ell} \right]^\top,$$

where $\Upsilon_\ell = \left(\sum_{j=1}^{\psi} e^{-2\eta\ell(x,v_j)} \right)^{\frac{1}{2}}$ is the normalization term which ensures $\|\Phi(x|\Lambda)\|_2 = 1$, η is the hyperparameter controlling the relative importance of the entries of $\Phi(x|\Lambda)$ and $\eta > 0$. Λ-mapping is agnostic towards the choice of $\ell(\cdot,\cdot)$. Unless otherwise specified, $\ell(\cdot,\cdot)$ represents the Euclidean distance.

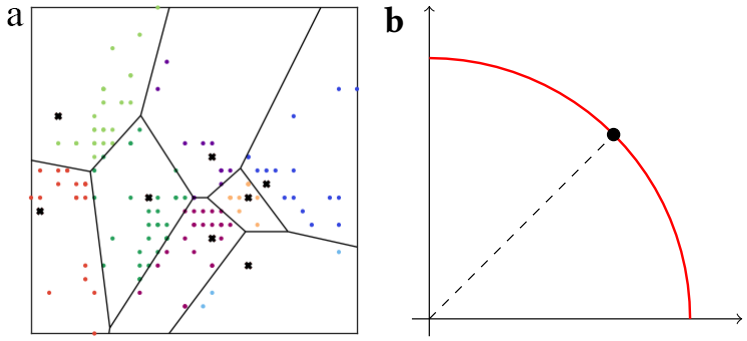


Fig. 2. (a) Example Voronoi Diagram where the ψ (=8) black points in $\mathbb{R}^d$ are sampled from the point cloud. (b) Data distribution after Λ-mapping $\Phi(\cdot \mid \Lambda)$ when $\psi = 2$.

$\Phi(\cdot|\Lambda)$-mapped points in $\mathbb{R}^\psi$ are distributed on the surface of a $(\psi-1)$-sphere $S^{\psi-1} = \{z \in \mathbb{R}^\psi : \|z\|_2 = 1\}$. Figure 2b shows an example of S^1 ($\psi = 2$), all mapped points are located on the red arc. The enlarged black point on the arc in the figure indicates the mapped location of the boundary in Voronoi Diagram Λ for $\psi = 2$ points. The points from the same cell in $\mathbb{R}^d$ are mapped to the same side of the mapped boundary in $\mathbb{R}^\psi$.

Let $V(X)$ be the set containing all the possible Voronoi Diagrams. For a Voronoi diagram $\Lambda \in V(X)$, ψ points randomly sampled from X are used as the centers of Voronoi cells. Sampling different ψ points will construct different Voronoi diagrams. There are $\binom{n}{k}$ different combinations of points that can be sampled from X, so there are a total of $\binom{n}{k}$ different Voronoi diagrams that can be constructed by X. $V(X)$ is the set containing all these possible Voronoi Diagrams.

Given that $\mathbf{e}_x \coloneqq \lim_{\eta\to\infty} \Phi(x|\Lambda)$, map $\Phi(x|\Lambda)$ has the following properties:

Property 5. *$\forall \epsilon > 0, \exists \hat{\eta} > 0, \forall \eta \geq \hat{\eta}$, $x \in X$, $\Lambda \in V(X)$, $\|\Phi(x \mid \Lambda) - \mathbf{e}_x\| \leq \epsilon$, and $\mathbf{e}_x$ is in one-hot form, under the assumption that x isn't exactly located on the boundary of Voronoi Diagram Λ.*

Property 5 shows that $\Phi(x|\Lambda)$ is increasingly influenced by x's nearest neighbors (in $\mathcal{D}$) as η increases. When η tends to infinity, $\Phi(x|\Lambda)$ is determined by x's nearest point only in $\mathcal{D}$. Proofs not given in this paper are provided in the supplementary materials.

The similarity between two points is defined as the inner product after they have been mapped using $\Phi(\cdot|\Lambda)$.

Definition 6. *For any $x, y \in \mathbb{R}^d$, the similarity of x and y based on Voronoi Diagram Λ is defined as:*

$$s\left(x, y \mid \Lambda\right) = \left\langle \Phi\left(x \mid \Lambda\right), \Phi\left(y \mid \Lambda\right) \right\rangle .$$

This similarity has the following property:

Property 7. *For a point cloud X, $\exists \tilde{\eta}, \forall \eta \geq \tilde{\eta}, \forall x, y, x', y' \in X$, if x, y belong to the same Voronoi cell; and x', y' belong to different Voronoi cells, then $s(x, y|\Lambda) \geq s(x', y'|\Lambda)$.*

Given a point cloud X, we denoted the set of all possible Voronoi Diagrams built from X using ψ seeds as $V(X)$. From Definition 6, Λ-kernel of x and y with respect to X, a similarity based on all possible Voronoi Diagrams derived from X, is defined as follows:

Definition 8. *$\forall x, y \in \mathbb{R}^d$, Λ-kernel derived from X is defined as:*

$$\kappa\left(x, y|X\right) = \mathbb{E}_\Lambda\left[s\left(x, y|\Lambda\right)\right] = \frac{1}{|V(X)|} \sum_{\Lambda \in V(X)} s\left(x, y|\Lambda\right) .$$

In practice, $\kappa(x, y|X)$ is estimated from a finite number of Voronoi Diagrams $\Lambda_i \in V(X)$, $i = 1, \ldots, t$:

$$\hat{\kappa}\left(x, y \mid X\right) = \frac{1}{t} \sum_{i=1}^{t} s\left(x, y \mid \Lambda_i\right) .$$

Lemma 9. *$\hat{\kappa}(x, y \mid X)$ is a valid kernel.*

The empirical feature map $\hat{\Phi}(x)$ of kernel $\hat{\kappa}$ is expressed as

$$\hat{\Phi}(x) = \frac{1}{\sqrt{t}} \left[\Phi_1(x)^\top, \Phi_2(x)^\top, ..., \Phi_t(x)^\top\right]^\top ,$$

where $\Phi_i(x) \coloneqq \Phi(x|\Lambda_i)$. Map $\hat{\Phi}(\cdot)$ has Property 10, which is a direct result of Property 5.

Property 10. *$\forall \epsilon > 0, \exists \hat{\eta} > 0, \forall \eta > \hat{\eta}, x \in X$, $\|\hat{\Phi}(x) - \hat{\mathbf{e}}_x\| \leq \epsilon$, where $\hat{\mathbf{e}}_x \coloneqq \lim_{\eta\to\infty} \hat{\Phi}(x)$.*

Here we propose distance d_Λ which satisfies the following Lemma and has data-dependent Property 12.

Lemma 11. *For $x, y \in X$, $d_\Lambda(x, y|X) \coloneqq 1 - \hat{\kappa}(x, y|X)$ is a distance metric when $\eta \to \infty$.*

Property 12. *For any two points x, y in dense region $\mathcal{S}$ and their corresponding x', y' in sparse region $\mathcal{T}$, it holds that $d_\Lambda(x, y|S) = d_\Lambda(x', y'|T)$ or equivalently $\hat{\kappa}(x, y|S) = \hat{\kappa}(x', y'|T))$, where $S \subset \mathcal{S}$, $T \subset \mathcal{T}$, and $\mathcal{S} = \{m(\mathcal{T}) + r(x - m(\mathcal{T}))|x \in \mathcal{T}\}$ for $r > 1$.*

Property 12 shows that the Λ-kernel can obtain the same similarity for point pairs before and after stretching and shrinking. This property works as the theoretical underpinning of Theorem 17, which shows that the PDs obtained by Λ-filter (see Section 6) are the same before and after stretching and shrinking.

5.2 Λ-kernel for UMAP Dimensionality Reduction

We show that the current similarity used in UMAP (or Uniform Manifold Approximation and Projection for Dimension Reduction (McInnes et al. 2018)) can be replaced with Λ-kernel. The new UMAP algorithm using Λ-kernel is named Λ-MAP and it produces a better dimensionality reduction performance than the original UMAP.

In brief, UMAP is a recent technique for reducing the dimensionality of high-dimensional data, which is based on the assumption that there exists a manifold on which the data would be uniformly distributed. UMAP finds a low-dimensional representation of the data that preserves the *topological structure* of this manifold by minimizing the cross-entropy between the two topological representations (simplicial set) of high-dimensional and low-dimensional data. (McInnes et al. 2018) achieves this by the following 5 steps:

* Graph Construction
  1. Construct a k-NN graph (using Euclidean distance).
  2. Assign weight on the edges of the k-NN graph, based on the graph connectivity
  3. Deal with the inherent asymmetry of the k-NN graph: treat the weight in step 2 as the probability that a directed edge exists and get pairwise similarity as the probability that at least one directed edge exists.
* Optimized Representation
  4. Define an objective function that preserves the desired characteristics of this k-NN graph.
  5. Use Spectral Embedding as initialization and find a low dimensional representation that optimizes this objective function.

A key issue in UMAP: UMAP assigns weights on the edges of the k-NN graph in step 2 to ensure that the points are uniformly distributed w.r.t. the similarity obtained through step 3. This weighting scheme is designed for points within each cluster and heavily depends on the k-NN graph connectivity. However, when points between two clusters are connected in the k-NN graph, the scheme reduces the distance between the two clusters inadvertently; hence leading to low inter-cluster distance.

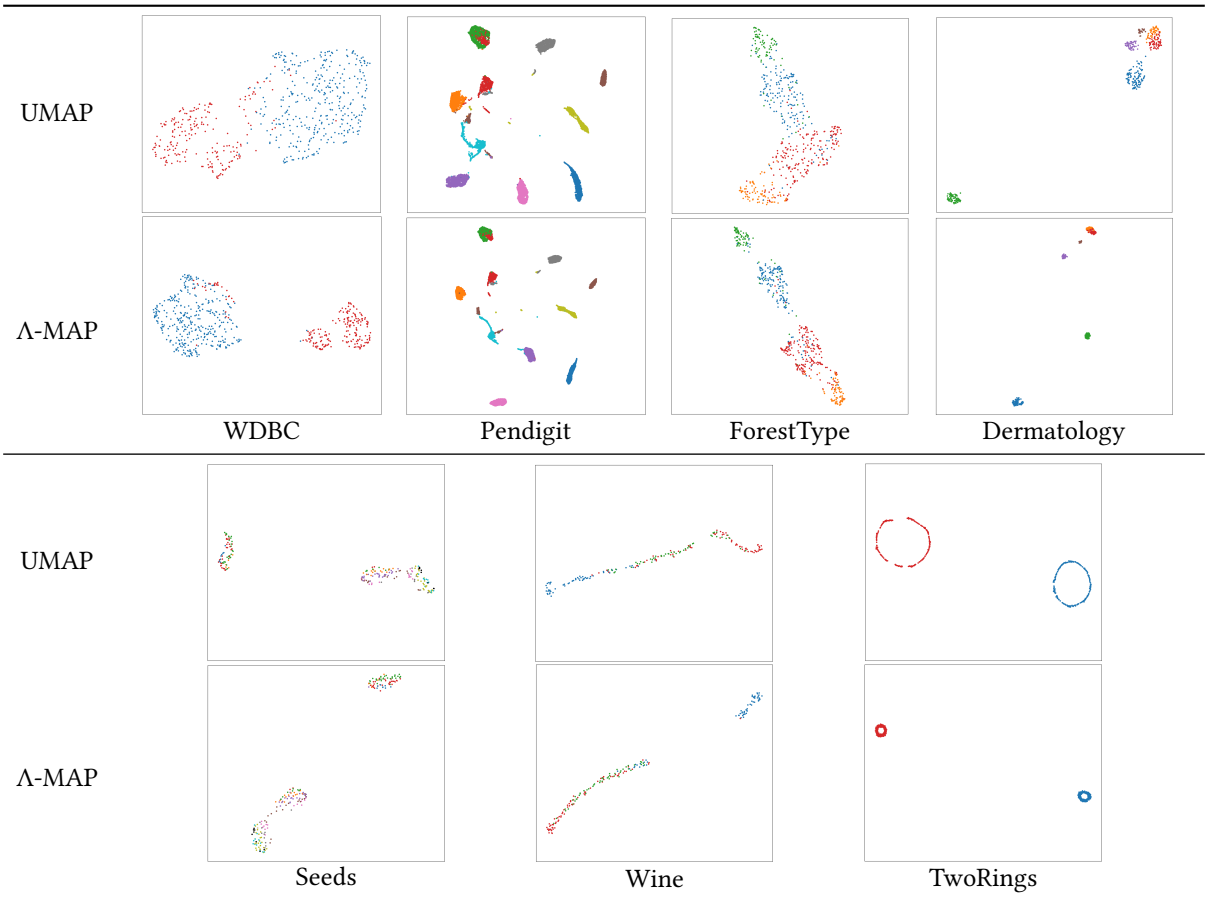
The above issue of UMAP can be addressed by using Λ-kernel[1] as pairwise similarity, without using k-NN and its weights. Λ-kernel is computed purely on the attributes of data. And it can enlarge inter-cluster distance because of its graph-independent nature. **We replace steps 2 and 3 in UMAP with a single step that computes Λ-kernel as pairwise similarity, and name the resultant algorithm Λ-MAP.** And we compare the performance of UMAP and Λ-MAP on six real datasets and one artificial dataset.

The dimensionality reduction results on seven datasets are shown in Table 3. Since all the datasets we used have ground truths, we select three commonly used indices of cluster validation and dimensionality reduction validation, i.e. Calinski-Harabasz index (CH) (Caliński and Harabasz 1974), Davies-Bouldin index (DB) (Davies and Bouldin 1979) and (AR) (Lee, Peluffo-Ordóñez, et al. 2015; Lee, Renard, et al. 2013; Lee and Verleysen 2009; Zhu and

[1]The task is to map a point in high-dimensional space to a point in low-dimensional space, which does not need to compute the PD. Thus only Λ-kernel is required, not Λ-filter.

Ting 2021), whose definition is provided in Supplementary Material, to evaluate whether all clusters are correctly identified after dimensionality reduction by UMAP and Λ-UMAP. Both CH and DB consider the similarity of points within a cluster and the similarity between different clusters; while AR measures the preservation of k-ary neighbourhoods after dimensionality reduction.

Table 2. Dimension reduction results of UMAP and Λ-MAP on seven datasets.



Λ-MAP outperforms UMAP on six datasets in terms of CH, and five datasets in terms of DB. UMAP performs slightly better than Λ-MAP in terms of AR on five datasets. This is not surprising because UMAP preserves k-ary neighbourhoods—the exact condition AR is measuring. In other words, AR is a measure that biases in favor of UMAP.

In addition, the results in low-dimensional space $\mathbf{R}^2$ are shown in Table 2. Visually, Λ-MAP produces better results than UMAP in two ways:

- On the TwoRings dataset, the red ring is better preserved by Λ-MAP than UMAP, where the one by UMAP is broken into three segments. This is despite the fact that both produce clusters of similar density, where the two rings originally had varied densities.

- On the WDBC, ForestType and Dermatology datasets, an obvious advantage of Λ-MAP over UMAP is that the former separates different clusters much better in low-dimensional space than the latter. UMAP often has a much smaller inter-cluster distance, the inter-cluster distance is significantly reduced when the k-NN graph connects points between different clusters. Λ-kernel avoids this issue without using a k-NN graph.

Table 3. Dimensionality reduction results. ↑ indicates that a higher score means a better dimensionality reduction result, and ↓ indicates the opposite.

Dataset	UMAP			Λ-MAP		
	CH (↑)	DB (↓)	AR (↑)	CH (↑)	DB (↓)	AR (↑)
WDBC	1046	0.62	**0.35**	**1521**	**0.50**	0.31
Pendig	7147	1.56	**0.23**	**9423**	**1.25**	0.16
ForestType	519	0.86	**0.38**	**838**	**0.72**	0.34
Dermatology	22621	0.37	**0.38**	**61622**	**0.30**	0.36
Seeds	**78**	2.94	**0.56**	75	**2.61**	0.54
Wine	196	**1.50**	0.61	**257**	1.52	**0.65**
TwoRings	6754	0.27	**0.53**	**239819**	**0.03**	0.49

The results of dimensionality reduction show that Λ-kernel has good properties in retaining the topological structure of the data. In the next section, we will introduce Λ-filter based on Λ-kernel.

6 Filter Function Based on Λ-kernel

Here we give a new filter function $f_\Lambda(\cdot)$ called Λ-filter based on Λ-kernel, whose corresponding filtration is referred to as Λ-filtration. We also prove that PD built from Λ-filtration is not only stable with respect to f_Λ, but also stable with respect to the perturbation of input point cloud X.

The new Λ-filter, is given as follows:

$$f_\Lambda(x) = -\frac{1}{|V(X)|}\max_{y\in X}\sum_{\Lambda\in V(X)}\rho(\Lambda)s(x,y\mid\Lambda)$$

A linearly-transformed version of $f_\Lambda(\cdot)$: $4(1+f_\Lambda(\cdot))$ is used for further analysis, whose empirical estimation is

$$\hat{f}_\Lambda(x) = 4\min_{y\in X}\left(1-\hat{\kappa}(x,y|X)\right). \tag{1}$$

Let $B(x,\epsilon) := \{y\in\mathbb{R}^d | \ell(x,y)\le\epsilon\}$, and $\mathcal{X}_\epsilon^{\hat{f}_\Lambda} := \{x\in\mathcal{X}\mid \hat{f}_\Lambda(x)\le\epsilon\}$. Then we have $\mathcal{X}_\epsilon^{\hat{f}_\Lambda} = \hat{\Phi}^{-1}(U)$, where $U = \bigcup_{x\in\hat{\Phi}(X)} B(x,\sqrt{\epsilon/2})$ and U is homotopy equivalent to the geometric realization of the Čech complex $C_{\sqrt{\epsilon/2}}(\hat{\Phi}(X))$, which is usually replaced with Rips complex $VR_{\sqrt{2\epsilon}}(\hat{\Phi}(X))$ to improve computational efficiency, since only pairwise distance is needed in Rips.

So $\mathcal{X}_\epsilon^{\hat{f}_\Lambda}$ can be fully represented by a set of edges :

$$\left\{(x,y)\in X^2 \mid \left\|\hat{\Phi}(x)-\hat{\Phi}(y)\right\|_2 \le \sqrt{2\epsilon}\right\},$$

which is equivalent to $\{(x,y)\in X^2 \mid d_\Lambda(x,y|X)\le\epsilon\}$, where $d_\Lambda(\cdot,\cdot|X)$ is defined in Lemma 11 and $d_\Lambda(x,y|X) = 1-\left\langle\hat{\Phi}(x),\hat{\Phi}(y)\right\rangle = \left\|\hat{\Phi}(x)-\hat{\Phi}(y)\right\|_2^2/2$.

This illustrates that using f_Λ as a filter function for PH is equivalent to using $d_\Lambda(x,y)$ to replace Euclidean distance in the filter function of Rips.

A fundamental property of PD is stability, i.e., a small perturbation in the input point cloud only leads to a small perturbation of its persistence diagram with respect to bottleneck distance (Chazal and Michel 2021). It is a

Wasserstein distance defined as:

$$W_\infty(P,Q) \coloneqq \inf_{\gamma\in\Gamma(P,Q)} \left(\int \|x-\gamma(x)\|_\infty dP(x) \right),$$

where P and Q are two distributions and $\Gamma(P,Q)$ is the set containing all the possible maps from the support of P to that of Q that transports the mass from P to Q.

If P and Q are empirical distributions based on sets X and Y, i.e. $P = \frac{1}{|X|}\sum_{x\in X}\delta_x$ and $Q = \frac{1}{|Y|}\sum_{y\in Y}\delta_y$, where δ_x is the Dirac point mass at x, then:

$$W_\infty(X,Y) \coloneqq \inf_{\varphi:X\to Y} \sup_{x\in X} \|x-\varphi(x)\|_\infty,$$

where the infimum is over all bijections between X and Y. For PDs $\mathfrak{D}_1$ and $\mathfrak{D}_2$, the bottleneck distance is the maximum distance between the points of the two PDs, after minimizing over all possible pairings of the points, including the points on the diagonal $\partial\Omega$, i.e. the boundary of half plane $\Omega = \{(t_1,t_2)\in\mathbb{R}^2 | t_2 > t_1\}$.

Furthermore, we show that Λ-filter is also stable to the input point cloud X, i.e., a small perturbation in X implies a small change in PD. The theorem is given as follows:

Theorem 13. *The bottleneck distance between two persistence diagrams $\mathfrak{D}(f)$ and $\mathfrak{D}(g)$, derived from Λ-filters $f(X)$ and $g(X')$, respectively, is bounded as follows:*

$$W_\infty\left(\mathfrak{D}(f),\mathfrak{D}(g)\right) \le \|f-g\|_\infty \le d_H\left(X,X'\right),$$

where X' is the perturbed version of X; and $d_H(X,X')$ is the Hausdorff Distance between X and X'.

7 Robust to Noise

7.1 Definition and Experiments on Artificial Dataset

We start by providing the definition of robustness to noise.

Definition 14. *Let $\delta > 0$ be a sufficiently small real number, For two given point clouds X and X_n, sampled from $\mathcal{X}$ and $\mathcal{X}_n$, respectively, we say that filter function f is a robust to noise if it satisfies the following conditions:*

(a) $W_\infty\left(\mathfrak{D}(X|f),\mathfrak{D}(X\cup X_n|f)\right) < \delta$, and
(b) the numbers of valid persistent features are the same in $\mathfrak{D}(X|f)$ and $\mathfrak{D}(X\cup X_n|f)$,

Theorem 15. *When using Λ-filter f, if $\psi^{2t} > (1-e^{-2\eta R})n^{2t}$, then for any order $k>0$, $\mathfrak{D}_k^\delta(X\cup X_n|f) = \mathfrak{D}_k^\delta(X|f)$ holds with probability more than $(1-\frac{m}{n+m})^{\psi t}$, where $\mathfrak{D}_k^\delta$ stands for the persistence diagram of the k-th order when the max filtration value is δ , R stands for the diameter of $X\cup X_n$, $\delta = 1-e^{-2\eta R}$, and $n=|X|, m=|X_n|$.*

DTM's filter function is a robust approximation of that of Vietoris-Rips (Chazal, Fasy, et al. 2017), estimated as:

$$\hat{f}_{DTM}(x) = \sqrt{\frac{1}{k}\sum_{i=1}^{k}\ell_i^2(x)}$$

where $\ell_i(x)$ is the i-th nearest neighbor distance of x in $X\cup X_n$ and k is the user-specified number of neighbors.

Filter $\hat{f}_{DTM}(x)$ using $\ell_i(x)$ to reduce the effect of noise points. When $|X_n|$ is small, for any $x\in\mathbb{R}^d$, its k nearest neighbors in X have a small change only after adding X_n. Hence $\hat{f}_{DTM}(x)$ almost does not change. Therefore DTM has Property 2.

Our proposed Λ-filter[2] is robust to noise because it is based on the Voronoi Diagram Λ constructed from the sampled subset $\mathcal{D}$. $\mathcal{D}$ sampled from X is close to $\mathcal{D}'$ sampled from $X\cup X_n$, i.e. $\mathcal{D}\approx\mathcal{D}'$ when there is only a small amount of noise. Therefore, $f_\Lambda(x|X)\approx f_\Lambda(x|X\cup X_n)$, i.e., Λ-filter also has Property 2.

Furthermore, compared with DTM, Λ-filter considers more topological information via ψ points sampled from a point cloud: (i) Λ-filter contains the global distributional information of the point cloud, while DTM utilizes local information pertaining to k-nn only. (ii) In addition to the distance to the ψ sampled points, Λ-filter utilizes the order of these distances. While DTM only uses the square mean of these distances.

We verify the behaviors of Λ-filter and DTM with respect to Definition 14 on the Cassini dataset (Chazal, Fasy, et al. 2017), where we add noise with ratio $\gamma = |X_n|/|X|$ ($|X|$=1000 in our experiments) from 0 to 0.5 in steps of 0.025, based on Definition 3.

Figure 3 shows that Λ-filter outperforms DTM and GK-filter[2] from the following aspects:

(i) The bottleneck distance due to Λ-filter is smaller than those of DTM and GK-filter at most values of γ.

(ii) The bottleneck distances of Λ-filter and DTM are 0.03 and 0.08 respectively, while GK-filter is 0.17 when $\gamma = 0.025$. This shows that GK-filter is very sensitive to noise. When $\gamma > 0.2$, the bottleneck distance due to DTM gradually increases as γ increases; while that due to Λ-filter remains below 0.2.

(iii) The Cassini dataset has one ring only, so the number of valid features is 1. GK-filter produces the correct number only when $\gamma < 0.05$. DTM always yields two rings incorrectly; while Λ-filter produces the correct number except at γ=0.2.

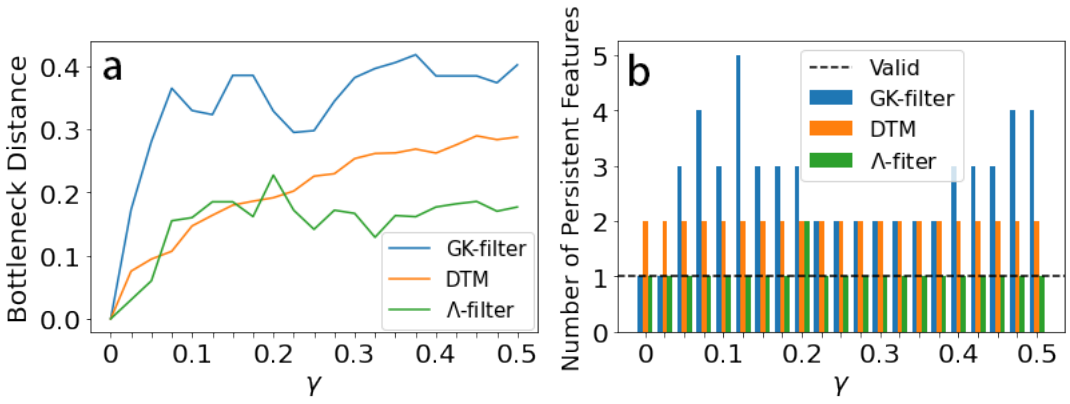


Fig. 3. (a) Bottleneck distance between $\mathfrak{D}(X)$ and $\mathfrak{D}(X \cup X_n)$ at different values of γ. (b) Number of valid 1-dim features in $\mathfrak{D}(X \cup X_n)$ at different values of γ. Here we treat a feature (c, g) as a persistent feature if its lifespan $g - c$ is greater than half of the maximum lifespan.

Then we demonstrate the effectiveness of robust PD obtained by Λ-filter in two tasks: visualization of immune cells from the tumor area and classification of bone scripts in the next two subsections. Both the immune cell dataset and the bone script dataset have numerous noisy data points.

7.2 Visualization of Immune Cells by Capturing Spatial Patterns via Persistent Homology

Here we treat each image independently, where its set of pixels is treated as a point cloud. The point cloud is transformed into a PD using each of Rips, DTM, CkNN, and Λ-filter. The distance between two images can then be computed as Wasserstein distance between two PDs. Following (Vishwanath et al. 2020), Wasserstein-1 distance matrices $\Delta_h[i, j] = W_1(\mathfrak{D}(C_i), \mathfrak{D}(C_j))$ for h-dimensional PDs is computed, where $h = 0, 1$, and C is a point cloud.

[2]We use a filter function that employs data-independent Gaussian Kernel (GK) in Equation 1 to represent noise-sensitive methods like Rips. We denote it as GK-filter.

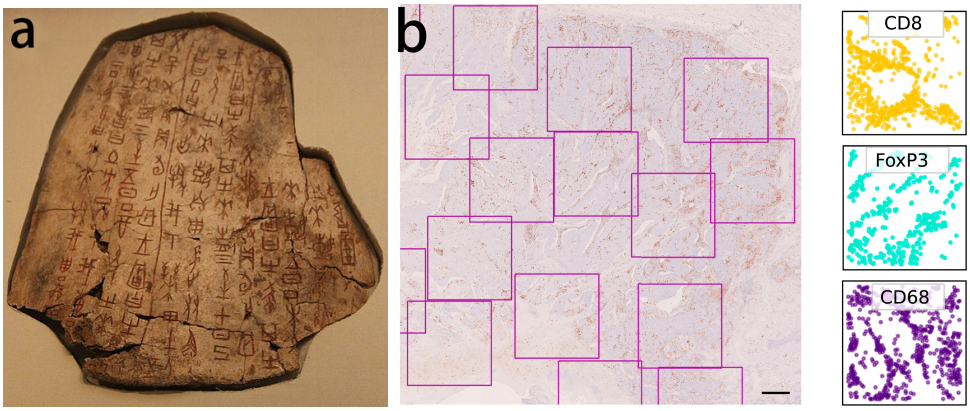


Fig. 4. (a) An example of bone scripts[3]. (b) A region of a head and neck tumor IHC slide stained to show CD8 cells. The regions of interest are highlighted.[4]. A point cloud of immune cell locations is extracted for each region of interest. Three example point clouds are shown here.

Finally, we feed $\Delta_{max} = max\{\Delta_0, \Delta_1\}$ to t-SNE (Van der Maaten and Hinton 2008) and get the visualization result, which is shown in Figure 5.

The dataset we used consists of 150 images (or point clouds $C_1, ..., C_{150}$) from 3 types of cells in tumor regions (Vipond et al. 2021), each type contains 50 point clouds. CD8, FoxP3 and CD68 are 3 types of immune cells, and their cell locations are extracted from IHC slides as (x, y)-coordinates, shown in Figure 4b, as conducted by (Vipond et al. 2021).

The results in Figure 5 show the superiority of Λ-filter over Rips, DTM, and CkNN. CD68 is well separated from the other two classes in the case of Λ-filter, except two points. CD8 has three groups at the fringes of FoxP3. DTM is the second best, followed by CkNN and Rips. DTM has a slightly poorer outcome than Λ-filtration, with seven points from CD8 and one point from CD68 mixed with FoxP3, and the fringe between CD8 and CD68 has some minor mixing up. Using the data after dimensionality reduction by t-SNE for kNN classifier over 10 trials of 5-fold Cross-Validation, the average accuracies of DTM and Λ-filter are 0.8 and 0.89, respectively.

7.3 Classification of Bone Scripts

Bone scripts are the earliest known written characters in China, so named because they are engraved on bones. In this experiment, we examine the use of PDs in a classification task on a bone-scripts dataset[5], as shown in Figure 4a.

1-dim PDs from Rips, DTM, CkNN, and Λ-filter are constructed and then vectorized as Persistence Image (PI) (Adams et al. 2017) which is a finite-dimensional vector representation for PD. We restrict the vector length to 400, which corresponds to 20×20 PI resolution. A SVM classifier using rbf kernel is then trained on the PIs.

We vary the PI bandwidth from 0.1 to 0.4, and report the mean classification accuracy and the corresponding standard deviation of 10 random train/test splits for each PI bandwidth. All the methods are relatively stable with respect to the bandwidth. But in terms of classification accuracy, Λ-filter outperforms the other three methods for every bandwidth, as shown in Figure 6a.

[3]Sourced from https://en.wikipedia.org/wiki/Oracle_bone_script.
[4]Sourced from https://www.pnas.org/doi/full/10.1073/pnas.2102166118. The image is used with permission from Christopher W. Pugh, one of the authors of (Vipond et al. 2021).
[5]The dataset is available at http://jgw.aynu.edu.cn/.

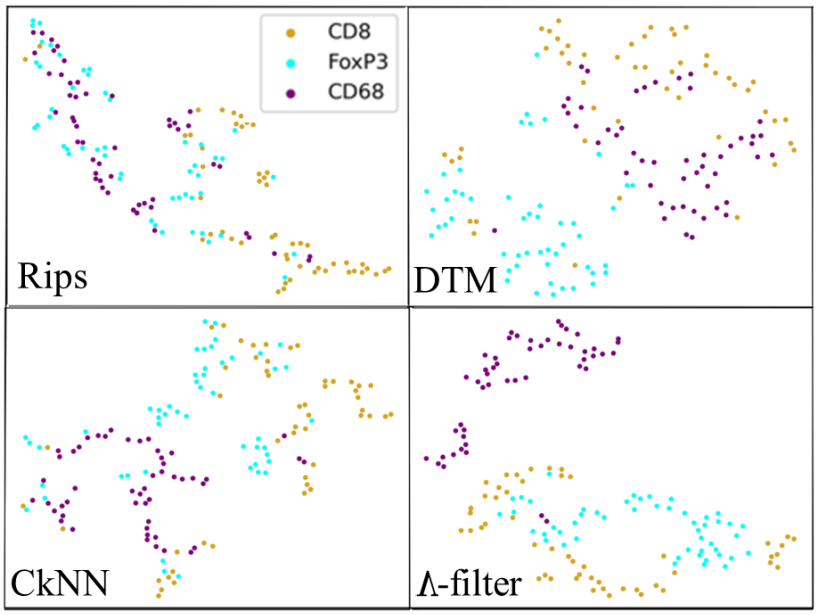


Fig. 5. t-SNE visualization results.

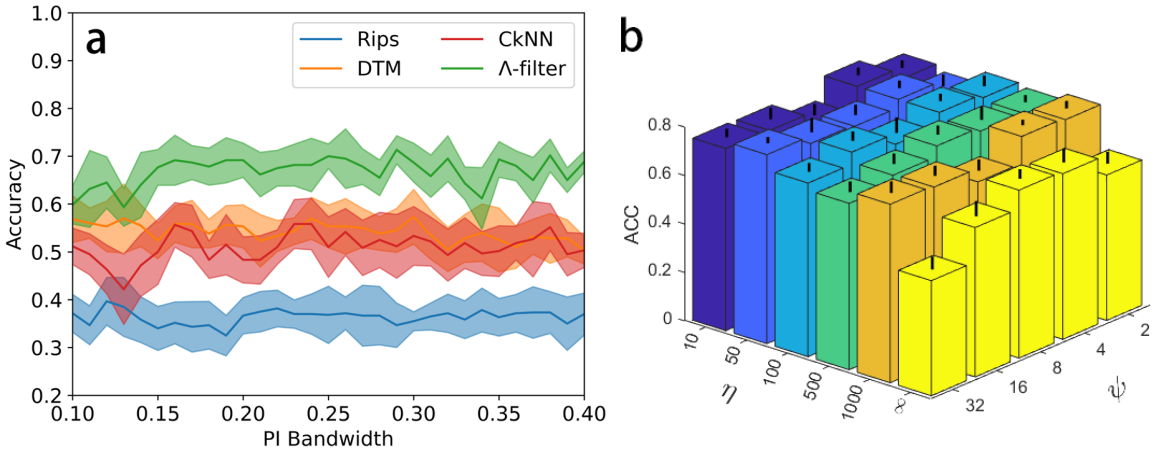


Fig. 6. (a) Accuracy of SVM classification of bone scripts. (b) Accuracy at different values of η as ψ increases.

In addition, we show the accuracy of SVM classification employing Λ-filter at different values of η and ψ in Figure 6b. When $\eta \neq \infty$, the accuracy is significantly higher and much more stable with respect to ψ than that when $\eta = \infty$. When $\eta = \infty$, the feature map of Λ-filter is located on grid (the coordinate is either 1 or 0). When $\eta \neq \infty$, the feature map are located more smoothly, which will better preserve the topological feature from original space and produce higher classification accuracy.

8 Robust to Isotropic Stretching and Shrinking

In this and next sections, we show the robustness of Λ-filter to varied densities, In this section, we explore the robustness on varied densities due to the simple isotropic stretching and shrinking.

In addition to robustness to noise, as we stated earlier, robustness to varied densities is also important, whose definition is given as follows:

Definition 16. *A filter function f is robust to varied densities (due to stretching/shrinking) if the PDs produced from two finite point clouds S and T (a stretched/shrunken version of S) are approximately the same, i.e., $W_\infty(\mathfrak{D}(f(S)), \mathfrak{D}(f(T))) = 0$.*

CkNN (Berry and Sauer 2019) is the only approach that can handle the issue of varied densities as far as we know. CkNN connects points x, y if $\frac{\ell(x,y)}{\sqrt{\ell_k(x)\ell_k(y)}} < \epsilon$, where the parameter ϵ is allowed to vary continuously to perform filtration. With the normalization term $\sqrt{\ell_k(x)\ell_k(y)}$, the lifespan of the connection of the two points is almost the same as that before stretching. Hence CkNN's filtration is robust to varied densities due to streching/shrinking.

Theorem 17. *Λ-filter is robust to varied densities.*

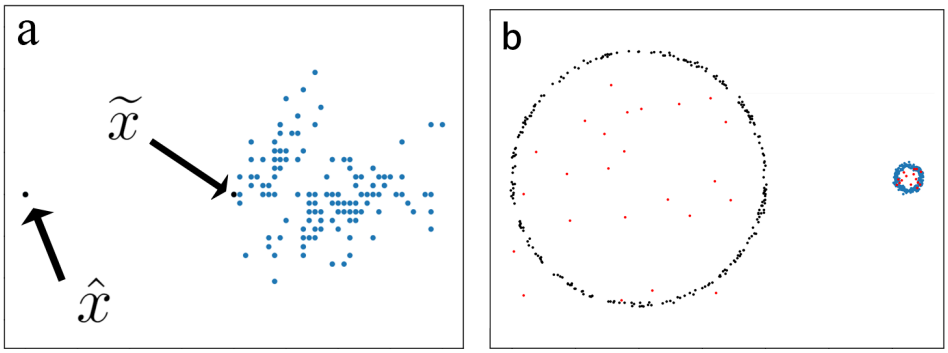


Fig. 7. (a) An example of $X_r \cup \{\hat{x}\}$ when $r = 1$. (b) Point cloud with noise and topological features of different densities. (The radius of the left ring is 10 times larger than that of the right one.)

The intuition behind the proof is that each point x's membership in a Voronoi cell remains almost the same after stretching/shrinking. Then the distance between x and other points, as measured by d_Λ, remains unchanged. Hence the value of Λ-filter remains unchanged.

Figure 7a and Figure 8 show an example of the influence of stretching and shrinking on PD. X_r is stretched from X with stretch factor r. Meanwhile, we keep the Euclidean distance between the noise point $\hat{x}$ and its nearest neighbor $\widetilde{x}$ in X_r to be constant. An example of $X_1 \cup \{\hat{x}\}$ is shown in Figure 7a.

The bottleneck distance between the PD of X and the PD of X_r is shown in Figure 8, where we have the following observations:

(i) The PD produced by Λ-filter is robust to stretching and shrinking. It has almost zero bottleneck distance, outperforming the other three measures over all values of r.

(ii) DTM and Rips are very sensitive to stretching and shrinking: their bottleneck distance grows as r increases.

(iii) CkNN's robustness to stretching and shrinking is severely affected by noise. Having only one noise point $\hat{x}$, its bottleneck distance grows dramatically as r decreases when $r < 1$, as shown in Figure 8b.

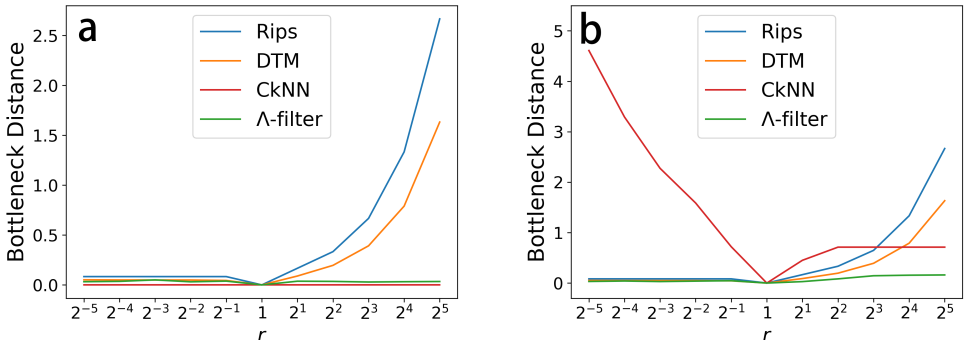


Fig. 8. Comparisons in varied densities. (a) Without noise: The bottleneck distance between $\mathfrak{D}(X)$ and $\mathfrak{D}(X_r)$ at different values of r. (b) With one noise point $\hat{x}$: The bottleneck distance between $\mathfrak{D}(X \cup \{\hat{x}\})$ and $\mathfrak{D}(X_r \cup \{\hat{x}\})$ at different values of r.

8.1 Noise and Varied Densities Coexist

The comparison of the four approaches (Rips, DTM, CkNN and Λ-filter) is summarized in Table 4.

Table 4. Summary of filter functions and their robustness.

Approach	Filter Function $f(x)$	Robust to	
		noise	varied densities
Rips	$2\min_{y\in X} \ell(x,y)$	×	×
DTM	$\sqrt{\frac{1}{k}\sum_{i=1}^{k} \ell_i^2(x)}$	✓	×
CkNN[6]	$\min_{y\in X} \frac{2\ell(x,y)}{\ell_k(y)}$	×	✓
Λ-filter	$4\min_{y\in X}(1-\kappa(x,y))$	✓	✓

A simple example is used to show the difference between these four approaches. The point cloud contains two rings with radius 5 (Figure 9a), whose filter function (Rips) is shown in Figure 9b. Then we stretch the left ring to radius 10, shrink the right ring to radius 1 and add 40 random noises (as shown in Figure 7b).

Table 5 shows the results of the comparison. Both Rips and DTM can detect one prominent (left) ring only because the less persistent (right) ring is indistinguishable from the topological features introduced by the noise around the more persistent (left) ring. We refer to this phenomenon as masking: the less persistent feature is masked out by the noise.

Although DTM is more robust to noise than Rips, DTM is unable to provide a completely noise-free PD. The masking effect has an adverse impact on DTM.

CkNN and Λ-filter have no masking effect: the two rings are identified as almost equally prominent, having their persistent levels significantly greater than that of the noise. However, there are some seemingly persistent features due to noise in the 1-dim barcode (and PD) of CkNN; but none in Λ-filter.

[6]The original paper (Berry and Sauer 2019) of CkNN gives the distance definition as $\frac{l(x,y)}{\sqrt{l_k(x)l_k(y)}}$ only. Here we provide an approximated filter function of CkNN.

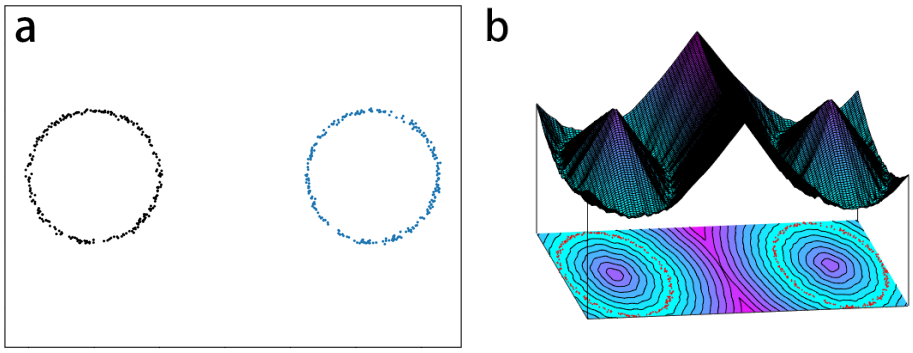
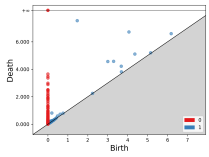
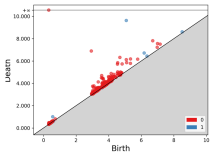
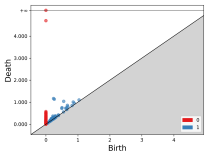
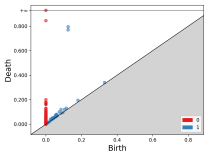
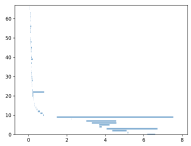
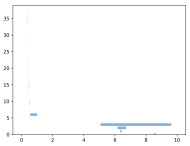
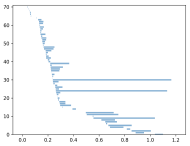
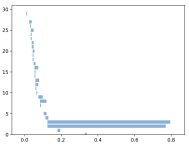
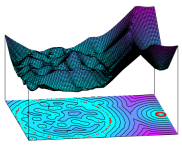
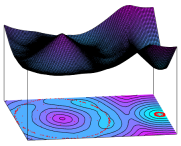
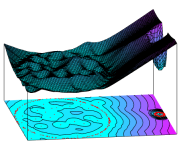
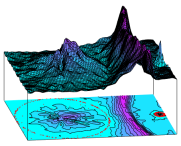


Fig. 9. (a) Data with no noise or varied densities. (b) The corresponding filter function (Rips) values and its contour map.

Table 5. A comparison result based on the point cloud shown in Figure 7b. The Persistence Diagram contains both 0-dim (red) and 1-dim (blue) feature. The Persistence Barcode row only shows 1-dim feature.

	Rips	DTM	CkNN	Λ-filter
Persistence Diagram				
Persistence Barcode				
Filter function				

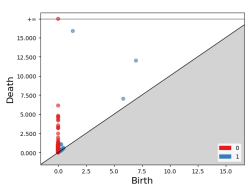
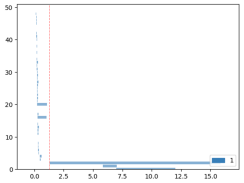
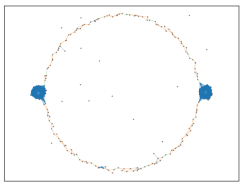
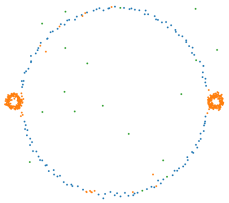
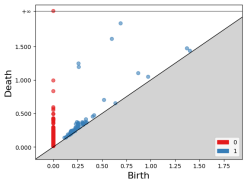
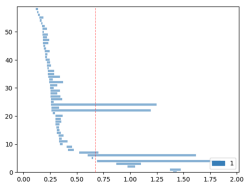
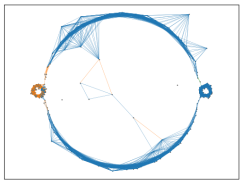
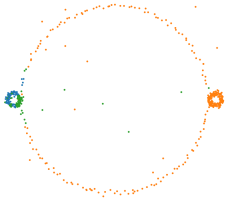
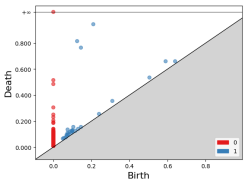
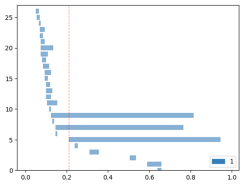
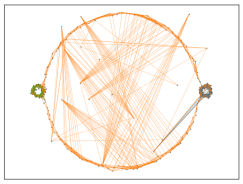
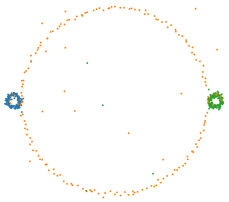
In a nutshell, only Λ-filter can handle the problem of noise and varied densities at the same time. DTM fails when the point cloud has varied densities; CkNN cannot handle the influence of noise; and Rips fails in both cases.

8.2 Λ-filter in Topological Point Cloud Clustering

We show the limitations of employing Rips in Topological Point Cloud Clustering (Grande and Schaub 2023) (TPCC) and propose to use the Λ-filter to address this issue and produce better clustering result.

TPCC clusters the points according to what topological features of the point cloud they contribute to. It works by the following 4 steps:

Table 6. Persistence Diagram , 1-dim Barcode, clustering result on edges and final clustering result. The red vertical dashed line in Barcode indicates the ϵ TPCC selects in the first step. The point cloud has 500 points and consists of three adjacent rings. Each ring has 160 points. The middle ring's radius is 15 times of that of the left and right ring. In addition, 20 noise points from a uniform distribution are added to the point cloud.

	Persistence Diagram	Persistence Barcode	Clustering Edge	Final Cluster
Rips				
CkNN				
Λ-filter				

1. Build simplicial complex under hyperparameter ϵ from point cloud
2. Extract Simplex Features by subspace clustering simplices
3. Build point feature (topological signature) from simplex clustering result
4. Cluster topological signatures

In the first step, TPCC selects ϵ in a a heuristic way which uses Rips to compute the persistence diagram of the point cloud, and then picks the ϵ as the birth time of the most persistent feature. However, when the point cloud has varied densities and Rips is employed, it may occur that the most persistent feature's birth time is larger than the death time of the less persistent ones, as shown in the Rips row in Table 6. This will cause that only the middle ring exists in the simplicial complex. The edges from the left and right rings contribute equally to the middle ring. So the points from the left and right rings are clustered together incorrectly.

For CkNN, under the selected ϵ, there exists four persistent rings instead of three. The middle rings is split because of the presence of noise, which affects the clustering result. Only Λ-filter is capable of selecting an ϵ value that generates a simplicial complex that reflects three rings, the actual topology of the point cloud. Evaluation of the clustering result is given in Table 7.

DTM is not considered here because TPCC aims to cluster all the points and the simplicial complex of DTM may not contain all the points, especially when ϵ is small.

Table 7. Adjusted Rand Index (ARI) and Normalized Mutual Infomation (NMI) of the clustering result in Table 6.

	Rips	CkNN	Λ-filter
ARI	0.38	0.41	0.71
NMI	0.52	0.54	0.68

9 Robust to Riemannian Stretching

We previously demonstrated in Section 8 that CkNN and Λ-filter can generate a PD that is robust to varied densities, caused by isotropic stretching. We now broaden the scope of the cause of varied densities from isotropic stretching to a more general form, Riemannian stretching, and investigate its solution, experimentally.

In Section 9.1, we emphasize the importance of Riemannian distance by pointing out the current filters equipped with $\ell(\cdot,\cdot)$ as Euclidean distance, including Λ-filter, have trouble in obtaining the correct topological features of a manifold with complex shape. And this problem can be addressed, to a certain extent, when Riemannian distance[7] is employed as $\ell(\cdot,\cdot)$ instead of Euclidean distance. In Section 9.2, we extend the cause of varied densities from isotropic stretching w.r.t. Euclidean distance to stretching w.r.t. Riemannian distance, named Riemannian stretching, as defined in Definition 18. When no noise in the point cloud, we show that Λ-filter and CkNN equipped with $\ell(\cdot,\cdot)$ as Geodesic distance can produce a PD robust to varied densities caused by Riemannian stretching. In Section 9.3, we improve the robustness of Λ-filter and CkNN against noise by using Sample Fermat distance (Fernández et al. 2023), a density-weighted Geodesic distance.

9.1 Importance of Riemannian Distance

Existing methods have difficulty in obtaining correct topological features for a point cloud sampled from a manifold with complex shape. For example, on the eyeglasses dataset with no noise or perturbation, as shown in Figure 10a, we plot the selected simplicial complex, together with corresponding sublevel set of each filter in Table 8. In every filter, there exists a simplicial complex in the filtration, where two rings exist at the same ϵ. This lead to a PD containing two (incorrect) persistent rings instead one persistent ring.

As we have described in Section 2, the Rips filtration works by having a ball, centered at each point of a point cloud, growing with radius ϵ. When two balls intersect with each other, an edge is added to connect the ball centers, resulting a simplicial complex. The balls centered at the middle narrow region, e.g. C, D in Figure 10a, intersect each other with a smaller ϵ than the balls centered at A, B. As a result, the process splits one large ring into two small rings. DTM, CkNN, Λ-filter fail for the same reason.

In a nutshell, the four filters suffer from the same issue: the distance of two points is based on positions of points in the ambient space.

Recall the manifold assumption (Bernstein et al. 2000): finite data points in ambient space $\mathbb{R}^{d_a}$ are assumed to lie on a smooth submanifold $\mathcal{M}^{d_i}$ of low dimension $d_i \ll d_a$, where d_i, d_a are the numbers of intrinsic and ambient dimensions, respectively. For example, in Figure 11, we show a manifold in the shape of a ring with $d_a = 3$ and $d_i = 1$.

We can address the forementioned issue (the distance of two points is based on their positions in ambient dimension) by computing the distance in the intrinsic dimension via Riemannian distance. For a manifold $\mathcal{M}$,

[7]In practice, we use its estimator named graph-based Geodesic distance (Bernstein et al. 2000), referred as Geodesic distance for short.

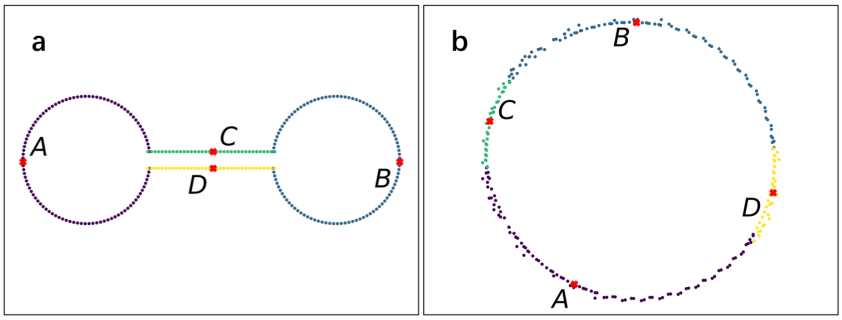
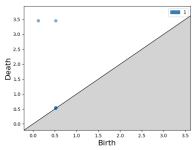
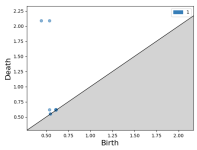
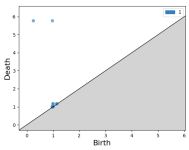
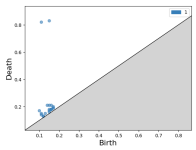
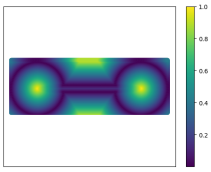
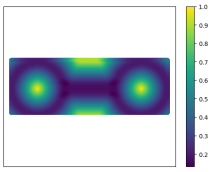
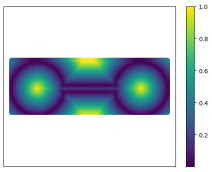
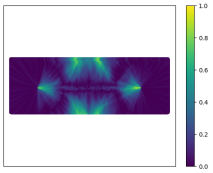
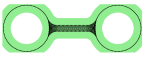
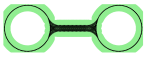


Fig. 10. (a) Eyeglasses dataset (b) When the eyeglasses dataset is measured by using Geodesic Distance (k=5) as input to MDS.

Table 8. Persistence Diagram of different filters on the eyeglasses dataset, filter function value, selected simplicial complex and corresponding sublevel set (Rips: $\epsilon = 0.6$, DTM: $\epsilon = 0.65$, CkNN: $\epsilon = 1.5$, Λ-filter: $\epsilon = 0.25$), when using Euclidean distance.

	Rips	DTM	CkNN	Λ-filter
Persistence Diagram				
Filter Function				
Simplicial Complex				

the (unweighted) Riemannian distance (Bernstein et al. 2000) is defined as

$$d_{\mathcal{M}}(x, y) = \inf_{\gamma} \{\text{length}(\zeta)\},$$

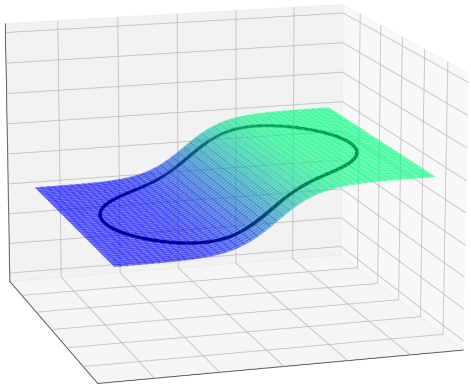


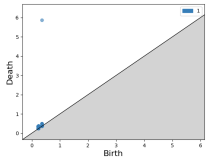
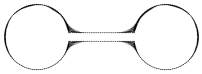
Fig. 11. A manifold (in black) in the shape of a ring with the number of ambient dimensions = 3 and the number of intrinsic dimensions = 1.

where $length(\zeta) = \int_I \sqrt{\langle \dot{\zeta}(t), \dot{\zeta}(t) \rangle} dt, I = [0, 1], \zeta$ is a piecewise smooth curve $\zeta : I \to \mathcal{M}$ with $\zeta(0) = x, \zeta(1) = y$ and $\dot{\zeta}$ is the derivative of ζ.

The length of the shortest path of two points in the k-NN graph (i.e., Geodesic distance) is an estimator of Riemannian distance (Bernstein et al. 2000). Two points having short Euclidean distance do not necessarily possess short Geodesic distance. Recall our example in Fig 10a, A, B have longer Euclidean distance than that of C, D, but the Geodesic distance between A, B is almost the same as that of C, D.

The issue with the four existing filters is avoided by directly utilizing Geodesic distance, instead of Euclidean distance, on the eyeglasses dataset to obtain PD, as shown in Table 9 (Eyeglasses row). We observe that only one persistent ring is obtained in the PD. And the simplicial complex forms one ring when ϵ ranges from 0.5 to 1.5. In fact, from the MDS result shown in Figure 10b, we deduce that this one ring will never spilt into two smaller rings, which is consistent with the PD.

Table 9. Persistence Diagram (PD) and selected simplicial complex[8] (SC) by using Geodesic distance in Rips filtration on the eyeglasses dataset shown Figure 10a.

	PD	SC ($\epsilon = 0.5$)	SC ($\epsilon = 1$)	SC ($\epsilon = 1.5$)
Eyeglasses				

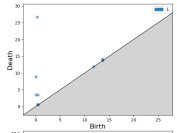
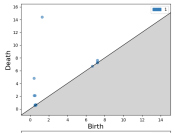
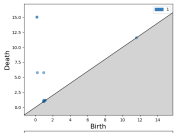
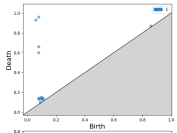
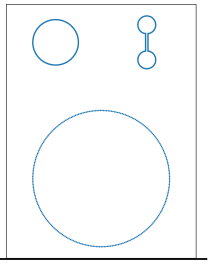
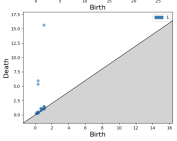
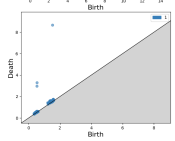
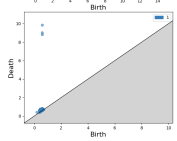
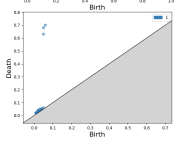
[8]Geodesic distance only works inside point cloud X, where the k-NN graph is built. For all the other points in the whole space, no Geodesic distance can be computed. As a consequence, sublevel set is unavailable here.

9.2 Riemannian Stretching

Here we study the case of stretching w.r.t. Riemannian distance, which creates the same problem of varied densities, studied under isotropic stretching in Section 8. The isotropic stretching is equivalent to a deformation that multiplies the pairwise Euclidean distance with $r \geq 1$ (or $r \leq 1$ for shrinking). Isotropic stretching is a special case of Riemannian stretching which is defined as follows:

Definition 18. *A manifold $\mathcal{M}'$ is Riemannian stretched from another manifold $\mathcal{M}$ if for any $x, y \in \mathcal{M}$ and their corresponding $x', y' \in \mathcal{M}'$, $d_{\mathcal{M}'}(x, y) = r \cdot d_{\mathcal{M}}(x', y')$, with $r \geq 1$, and $r \leq 1$ if $\mathcal{M}'$ is shrunk from $\mathcal{M}$.*

Table 10. Persistence Diagram (PD) of different filters on the point cloud shown in the rightmost column. The ideal PD should contain three equally persistent rings. The point cloud contains both isotropic stretching (from top left ring to bottom sparse ring, $r = 3$) and Riemannian stretching (from top right eyeglasses to bottom sparse ring, $r = 3$).

	Rips	DTM	CkNN	Λ-filter	point cloud
Euclidean					
Geodesic					

We show here that a PD that is robust to varied densities caused by Riemannian stretching can be obtained by simply replacing $\ell(\cdot, \cdot)$ in CkNN and Λ-filter from Euclidean distance to Geodesic distance[9].

Table 10 shows that: 1) When Geodesic distance is used instead of Euclidean distance, all filters obtain the correct three persistent rings instead of four because Geodesic distance can prevent the split of the eyeglasses into two small rings. 2) Only CkNN and Λ-filter equipped with Geodesic distance can correctly detect three rings with almost the same persistence. Two rings' persistence is significantly smaller than the other one's persistence in the PDs produced by Rips and DTM. This happens because the Geodesic distance of the bottom sparse ring is much longer than those of the top dense rings.

9.3 Improve Robustness against Noise with Fermat Distance

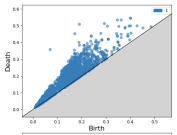
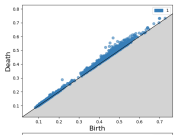
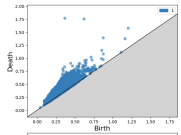
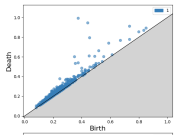
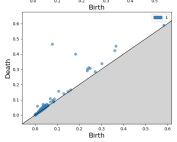
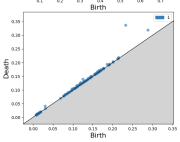
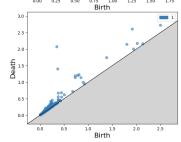
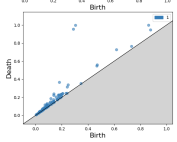
Although Λ-filter and CkNN equipped with Geodesic distance can deal with Riemannian stretching, their performance can suffer from noise. As shown in the Geodesic row of Table 11, CkNN and Λ-filter produce too many noise features in the PDs and the persistences of these noise features are quite large.

We can improve their robustness against noise by using Sample Fermat distance instead of Geodesic distance. Sample Fermat distance (Groisman et al. 2018; Mckenzie and Damelin 2019), a density-weighted Geodesic distance on a complete graph, is defined as:

$$d_F(x, y) = \inf_{\zeta} \sum_{i=0}^{k} \|x_{i+1} - x_i\|_2^p, \tag{2}$$

[9]In practice, when the Geodesic distance between two points is ∞, we set it to be twice of the maximum finite Geodesic distance.

Table 11. Persistence Diagram (PD) of different filters on the point cloud shown in the rightmost column. The ideal PD should contain two equally persistent rings. The persistence of noise should be as close to 0 as possible.

	Rips	DTM	CkNN	Λ-filter	Point Cloud
Geodesic					
Fermat					

where the infimum is taken over all paths $\zeta = (x_0, x_1, ..., x_{k+1})$ of finite length with $x_0 = x$, $x_{k+1} = y$ and $p > 1$. It is robust to noise and can be used to obtain PD (Fernández et al. 2023). It finds the shortest path by bypassing the low density (noise) region. Hereafter we will refer Sample Fermat distance as Fermat distance[10] for short.

Our experiment is conducted on real dataset MPEG7[11] (Latecki et al. 2000; Vishwanath et al. 2020), which contains 70 shape categories with 20 different images for each category. This dataset is part of the MPEG-7 standard, which has been used for characterizing multimedia content. We choose two instances from the deer and beetle classes and uniformly sample points on the image boundary, together with some injected noise, as shown in the point cloud column of Table 11. We enlarge beetle to create Riemannian stretching from deer to beetle.

As shown in Table 11, when using Fermat distance, the number of noise features is significantly reduced for all the filters. And only Λ-filter produces a PD which contains two persistent rings with almost equal persistence and noise features with near-zero persistence.

In summary, only Λ-filter equipped with Fermat distance can handle the problem of having both noise and varied densities caused by Riemannian stretching.

10 Discussion and Future Works

As defined in (Prasolov 1998), topology studies the properties of geometrical objects that remain unchanged under transformations called homeomorphisms and deformations, such as stretching, twisting, crumpling, and bending.

However, the persistence of a topological feature in existing methods to produce PD is sensitive to the scale of a point cloud. In the case of Rips and DTM, a stretched point cloud leads to a more persistent feature than that of the original point cloud. This contradicts with the stated aim of topology mentioned above.

Hence, a data-dependent measure is a way to ensure that the persistence of a topological feature remains the same after continuous deformations while keeping the number of valid features unchanged. Λ-filter and CkNN are examples of this attempt.

The intuition behind Λ-filter is based on the point correspondence before/after stretching the manifold. However, if we have two point clouds X, Y sampled respectively from manifold $\mathcal{X}$ and its stretched version $\mathcal{Y}$ and $|X| \neq |Y|$, the point correspondence no longer holds, hence Λ-filter fails in this case. When comparing two point clouds of different sizes, for the point cloud with the bigger size, we can downsample to ensure the sizes

[10]When not used in CkNN or Λ-filter, Fermat distance alone cannot handle the isotropic stretching, not to mention Riemannian stretching. According to Equation 2, isotropic stretching ($r > 1$) will directly multiply Fermat distance with r^p. Consequently, the persistence of topological feature in PD is significantly enlarged.

[11]The dataset is available at https://github.com/sidv23/robust-PDs.

of two point clouds to be the same as a potential method. For the point cloud with smaller size, one potential way is to use data generation methods to generate some pseudo points to make the sizes of two point clouds to be the same. In addition to changing point cloud size, when constructing the random Voronoi Diagrams, we could use sampling methods that is invariant to the point cloud size, such as Farthest Point Sampling. It would be interesting to consider a filter function that can produce stretching-invariant PD without the point correspondence requirement in the future.

Note that Λ-filter only works for isotropic stretching. And we experimentally show that Λ-filter equipped with Geodesic Distance works for Riemannian stretching. Finding a filter function for all continuous deformation with a theoretical guarantee remains an open problem.

11 Conclusions

We are the first to address the problem of the lack of robustness of existing PDs when noise and varied densities coexist in a point cloud in the field of PH. Existing methods are partial, solving either the noise or varied densities issue only.

The key to addressing the noise issue is a noise-tolerance kernel; and the key to addressing the varied densities (stretching/shrinking) issue is a data-dependent kernel that adapts to local distribution with Property 12; The proposed Λ-kernel possesses both properties. Existing methods do not have both properties. Then, Λ-filter is derive from Λ-kernel to build PD.

For Λ-kernel, we replace the similarity in UMAP with Λ–kernel, and the resultant Λ–MAP outperforms UMAP in dimensionality reduction task.

For Λ-filter, we have verified the superiority of computing PD throught the proposed Λ-filter over three existing methods in three tasks: t-SNE visualization, SVM classification and Topological Point Cloud Clustering.

Acknowledgments

This project is supported by National Natural Science Foundation of China (Grant No. 92470116 and W2531050).

References

H. Adams et al.. 2017. "Persistence images: A stable vector representation of persistent homology." *Journal of Machine Learning Research*, 18.

D. Ali, A. Asaad, M.-J. Jimenez, V. Nanda, E. Paluzo-Hidalgo, and M. Soriano-Trigueros. 2023. "A survey of vectorization methods in topological data analysis." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45, 12, 14069–14080.

F. Aurenhammer. 1991. "Voronoi diagrams—a survey of a fundamental geometric data structure." *ACM Computing Surveys*, 23, 3, 345–405.

F. Aurenhammer and R. Klein. 2000. "Voronoi Diagrams." *Handbook of Computational Geometry*, 5, 10, 201–290.

M. Bernstein, V. De Silva, J. C. Langford, and J. B. Tenenbaum. 2000. *Graph approximations to geodesics on embedded manifolds*. Tech. rep. Citeseer.

T. Berry and T. Sauer. 2019. "Consistent manifold representation for topological data analysis." *Foundations of Data Science*, 1, 1, 1–38.

A. J. Blumberg and M. Lesnick. 2022. *Stability of 2-parameter persistent homology*. (2022).

P. Bubenik. Jan. 2015. "Statistical topological data analysis using persistence landscapes." *Journal of Machine Learning Research*, 16, (Jan. 2015), 77–102.

P. Bubenik. 2020. *The Persistence Landscape and Some of Its Properties*. (2020).

T. Caliński and J. Harabasz. 1974. "A dendrite method for cluster analysis." *Communications in Statistics-theory and Methods*, 3, 1, 1–27.

G. Carlsson, T. Ishkhanov, V. De Silva, and A. Zomorodian. 2008. "On the local behavior of spaces of natural images." *International Journal of Computer Vision*, 76, 1, 1–12.

G. Carlsson and A. Zomorodian. 2007. "The Theory of Multidimensional Persistence." In: *Proceedings of Annual Symposium on Computational Geometry*, 184–193.

G. Carlsson, A. Zomorodian, A. Collins, and L. Guibas. 2004. "Persistence Barcodes for Shapes." In: *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, 124–135.

M. Carrière, M. Cuturi, and S. Oudot. 2017. "Sliced Wasserstein Kernel for Persistence Diagrams." In: *Proceedings of International Conference on Machine Learning*. Vol. 70, 664–673.

K. C. Chan, U. Islambekov, A. Luchinsky, and R. Sanders. 2022. "A computationally efficient framework for vector representation of persistence diagrams." *Journal of Machine Learning Research*, 23, 268, 1–33.

F. Chazal, L. J. Guibas, S. Y. Oudot, and P. Skraba. 2013. "Persistence-based clustering in Riemannian manifolds." *Journal of the ACM*, 60, 6, 1–38.

F. Chazal, B. Fasy, F. Lecci, B. Michel, A. Rinaldo, A. Rinaldo, and L. Wasserman. 2017. "Robust topological inference: Distance to a measure and kernel distance." *Journal of Machine Learning Research*, 18, 1, 5845–5884.

F. Chazal, L. J. Guibas, S. Y. Oudot, and P. Skraba. 2011. "Scalar field analysis over point cloud data." *Discrete & Computational Geometry*, 46, 4, 743–775.

F. Chazal and B. Michel. 2021. "An Introduction to Topological Data Analysis: Fundamental and Practical Aspects for Data Scientists." *Frontiers in Artificial Intelligence*, 4, 667963.

I. Chevyrev, V. Nanda, and H. Oberhauser. 2018. "Persistence paths and signature features in topological data analysis." *IEEE transactions on pattern analysis and machine intelligence*, 42, 1, 192–202.

D. L. Davies and D. W. Bouldin. 1979. "A cluster separation measure." *IEEE transactions on pattern analysis and machine intelligence*, 1, 2, 224–227.

H. Edelsbrunner, D. Letscher, and A. Zomorodian. 2000. "Topological persistence and simplification." In: *Proceedings of Annual Symposium on Foundations of Computer Science*, 454–463.

X. Fernández, E. Borghini, G. Mindlin, and P. Groisman. 2023. "Intrinsic persistent homology via density-based metric learning." *Journal of Machine Learning Research*, 24, 75, 1–42.

P. Frosini and M. Mulazzani. 1999. "Size homotopy groups for computation of natural size distances." *Bulletin of the Belgian Mathematical Society-Simon Stevin*, 6, 3, 455–464.

R. Ghrist. 2014. *Elementary Applied Topology*. CreateSpace Independent Publishing Platform.

V. P. Grande and M. T. Schaub. July 2023. "Topological Point Cloud Clustering." In: *Proceedings of the 40th International Conference on Machine Learning* (Proceedings of Machine Learning Research). Ed. by A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett. Vol. 202. PMLR, (July 2023), 11683–11697.

P. Groisman, M. Jonckheere, and F. Sapienza. 2018. *Weighted Geodesic Distance Following Fermat's Principle*. (2018).

J.-C. Hausmann et al.. 1995. "On the Vietoris-Rips complexes and a cohomology theory for metric spaces." *Annals of Mathematics Studies*, 138, 175–188.

M. Hearst, S. Dumais, E. Osuna, J. Platt, and B. Scholkopf. 1998. "Support vector machines." *IEEE Intelligent Systems and their Applications*, 13, 4, 18–28.

F. Hensel, M. Moor, and B. Rieck. 2021. "A survey of topological machine learning methods." *Frontiers in Artificial Intelligence*, 4, 681108.

C. D. Hofer, R. Kwitt, and M. Niethammer. 2019. "Learning Representations of Persistence Barcodes." *Journal of Machine Learning Research*, 20, 126, 1–45.

G. Kusano, K. Fukumizu, and Y. Hiraoka. 2017. "Kernel method for persistence diagrams via kernel embedding and weight factor." *Journal of Machine Learning Research*, 18, 1, 6947–6987.

L. J. Latecki, R. Lakamper, and T. Eckhardt. 2000. "Shape descriptors for non-rigid shapes with a single closed contour." In: *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*. Vol. 1, 424–429.

T. Le and M. Yamada. 2018. "Persistence fisher kernel: A riemannian manifold kernel for persistence diagrams." *Advances in Neural Information Processing Systems*, 31.

J. A. Lee, D. H. Peluffo-Ordóñez, and M. Verleysen. 2015. "Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure." *Neurocomputing*, 169, 246–261.

J. A. Lee, E. Renard, G. Bernard, P. Dupont, and M. Verleysen. 2013. "Type 1 and 2 Mixtures of Kullback-Leibler Divergences as Cost Functions in Dimensionality Reduction Based on Similarity Preservation." *Neurocomput.*, 112, 92–108.

J. A. Lee and M. Verleysen. 2009. "Quality assessment of dimensionality reduction: Rank-based criteria." *Neurocomputing*, 72, 7, 1431–1443.

M. Lesnick and M. Wright. 2015. *Interactive visualization of 2-d persistence modules*. (2015).

X. Liu, H. Feng, J. Wu, and K. Xia. 2022. "Dowker complex based machine learning (DCML) models for protein-ligand binding affinity prediction." *PLoS Computational Biology*, 18, 4, 1–17.

L. McInnes, J. Healy, N. Saul, and L. Großberger. 2018. "UMAP: Uniform Manifold Approximation and Projection." *Journal of Open Source Software*, 3, 29, 861.

D. Mckenzie and S. Damelin. 2019. "Power weighted shortest paths for clustering Euclidean data." *Foundations of Data Science*, 1, 3, 307–327.

A. W. Mead. 1992. "Review of the Development of Multidimensional Scaling Methods." *The Statistician*, 41, 27–39.

Z. Meng, D. V. Anand, Y. Lu, J. Wu, and K. Xia. 2020. "Weighted persistent homology for biomolecular data analysis." *Scientific Reports*, 10, 1, 1–15.

J. Milnor. 1964. "On the Betti numbers of real varieties." *Proceedings of American Mathematical Society*, 15, 2, 275–280.

K. Muandet, K. Fukumizu, B. Sriperumbudur, B. Schölkopf, et al.. 2017. "Kernel mean embedding of distributions: A review and beyond." *Foundations and Trends® in Machine Learning*, 10, 1-2, 1–141.

A. Okabe, B. Boots, K. Sugihara, and S. Chiu. 2000. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley Series in Probability and Statistics. Wiley.
L. Polanco and J. A. Perea. 2019. "Adaptive Template Systems: Data-Driven Feature Selection for Learning with Persistence Diagrams." In: *IEEE International Conference On Machine Learning And Applications*, 1115–1121.
V. Prasolov. 1998. *Intuitive Topology*. Mathematical world. Universities Press (India) Pvt. Limited.
Y. Reani and O. Bobrowski. 2022. "Cycle registration in persistent homology with applications in topological bootstrap." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45, 5, 5579–5593.
D. Reem. 2011. "The geometric stability of Voronoi diagrams with respect to small changes of the sites." In: *Proceedings of the twenty-seventh annual symposium on Computational geometry*, 254–263.
J. Reininghaus, S. Huber, U. Bauer, and R. Kwitt. 2015. *A stable multi-scale kernel for topological machine learning*. (2015).
M. Short. 2013. "Improved inequalities for the poisson and binomial distribution and upper tail quantile functions." *International Scholarly Research Notices*, 2013.
J. Townsend, C. P. Micucci, J. H. Hymel, V. Maroulas, and K. D. Vogiatzis. 2020. "Representation of molecular structures with persistent homology for machine learning applications in chemistry." *Nature Communications*, 11, 1, 1–9.
L. Van der Maaten and G. Hinton. 2008. "Visualizing data using t-SNE." *Journal of Machine Learning Research*, 9, 11.
O. Vipond. 2020. "Multiparameter Persistence Landscapes." *Journal of Machine Learning Research*, 21, 61, 1–38.
O. Vipond, J. A. Bull, P. S. Macklin, U. Tillmann, C. W. Pugh, H. M. Byrne, and H. A. Harrington. 2021. "Multiparameter persistent homology landscapes identify immune cell spatial patterns in tumors." *Proceedings of National Academy of Sciences*, 118, 41, e2102166118.
S. Vishwanath, K. Fukumizu, S. Kuriki, and B. K. Sriperumbudur. 2020. "Robust persistence diagrams using reproducing kernels." *Advances in Neural Information Processing Systems*, 33, 21900–21911.
L. Wasserman. 2018. "Topological data analysis." *Annual Review of Statistics and Its Application*, 5, 501–532.
K. Xia and G.-W. Wei. 2014. "Persistent homology analysis of protein structure, flexibility, and folding." *International Journal for Numerical Methods in Biomedical Engineering*, 30, 8, 814–844.
H. Zhang, K. Zhang, K. M. Ting, and Y. Zhu. 2023. "Towards a Persistence Diagram that is Robust to Noise and Varied Densities." In: *Proceedings of the 40th International Conference on Machine Learning*. Vol. 202, 41952–41972.
Y. Zhu and K. M. Ting. 2021. "Improving the Effectiveness and Efficiency of Stochastic Neighbour Embedding with Isolation Kernel." *Journal of Artificial Intelligence Research*, 71, 667–695.
A. Zomorodian and G. Carlsson. 2004. "Computing Persistent Homology." In: *Proceedings of Annual Symposium on Computational Geometry*, 347–356.

A Proofs

A.1 Proof of Property 5

Property 5 $\forall\, \epsilon \in (0,1), \exists\, \hat{\eta} > 0, \forall\, \eta \geq \hat{\eta},\ x \in X,\ \Lambda \in V(X), \|\Phi(x \mid \Lambda) - \mathbf{e}_x\| \leq \epsilon$, where $\mathbf{e}_x \coloneqq \lim_{\eta\to\infty} \Phi(x|\Lambda)$ and it is in one-hot form, under the assumption that x isn't exactly located on the boundary of Voronoi Diagram Λ.

Proof. First we start by proving that $\mathbf{e}_x$ is in one-hot form when $\eta \to \infty$. Denote the i-th component of $\Phi(x|\Lambda)$ as $\Phi(x|\Lambda)[i]$ as follows :

$$\Phi(x|\Lambda)[i] = \frac{e^{-\eta \ell(x,v_i)}}{\left(\sum_{j=1}^{\psi} e^{-2\eta \ell(x,v_j)}\right)^{\frac{1}{2}}},$$

where $i = 1, ..., \psi$. Then we know that

$$\lim_{\eta\to\infty} \Phi(x|V)[i] = \begin{cases} 1, & i = \hat{i}, \\ 0, & i \neq \hat{i} \end{cases},$$

where $\hat{i} \coloneqq \arg\min_i \ell(x, v_i)$.

This means that the points are limited in a very small angle near the $\hat{i}$-axis when η is very large.
$\forall x \in X, \Lambda \in V(X)$, set $\hat{\eta}_{x,\Lambda} = \frac{1}{2q_{x,\Lambda}} \ln \frac{(\psi-1)(2-\epsilon^2)}{\epsilon^2}$, where $q_{x,\Lambda} = \arg\min_{i \neq \hat{i}} (\ell(x, v_i) - \ell(x, v_{\hat{i}}))$, then

$$
\begin{aligned}
&\|\Phi(x|\Lambda) - \mathbf{e}_x\|^2 \\
&= (\Phi(x|\Lambda)[\hat{i}] - 1)^2 + \sum_{j=1, j\neq\hat{i}}^{\psi} \Phi(x|\Lambda)[j]^2 \\
&= 2 - \frac{2e^{-\hat{\eta}_{x,\Lambda}\ell(x,v_{\hat{i}})}}{\left(\sum_{j=1}^{\psi} e^{-2\hat{\eta}_{x,\Lambda}\ell(x,v_j)}\right)^{\frac{1}{2}}} \\
&= 2 - \frac{2}{\left(1 + \sum_{j=1, j\neq\hat{i}}^{\psi} e^{-2\hat{\eta}_{x,\Lambda}(\ell(x,v_j)-\ell(x,v_{\hat{i}}))}\right)^{\frac{1}{2}}} \\
&\leq 2 - \frac{2}{\left(1 + (\psi-1)e^{-2\hat{\eta}_{x,\Lambda}q_{x,\Lambda}}\right)^{\frac{1}{2}}} \\
&\leq \epsilon^2.
\end{aligned}
$$

To find the η for the entire point cloud, take $\hat{\eta}$ as $\max_{x\in X, \Lambda\in V(X)} \hat{\eta}_{x,\Lambda}$, then $\forall x \in X, \Lambda \in V(X)$, we have

$$
\begin{aligned}
\|\Phi(x|\Lambda) - \mathbf{e}_x\|^2 &\leq 2 - \frac{2}{\left(1 + (\psi-1)e^{-2\hat{\eta}q_{x,\Lambda}}\right)^{\frac{1}{2}}} \\
&\leq 2 - \frac{2}{\left(1 + (\psi-1)e^{-2\hat{\eta}_{x,\Lambda}q_{x,\Lambda}}\right)^{\frac{1}{2}}} \\
&\leq \epsilon^2.
\end{aligned}
$$

□

A.2 Proof of Property 7

Property 7 $\exists \tilde{\eta}, \forall \eta \geq \tilde{\eta}, \forall x, y, x', y' \in \mathbb{R}^d$, if x, y belong to the same Voronoi cell θ_i; and x', y' belong to different Voronoi cells θ_j and θ_k, then $s(x, y|\lambda) \geq s(x', y'|\Lambda)$.

Proof. Set $\tilde{\epsilon} = \frac{\sqrt{2}}{4}$, according to Property 4, $\exists \tilde{\eta}, \forall \eta \geq \tilde{\eta}, \forall x \in X, \|\Phi(x|\Lambda) - \mathbf{e}_x\|_2 \leq \tilde{\epsilon}$. And we have $\mathbf{e}_x = \mathbf{e}_y, \mathbf{e}_{x'} \neq \mathbf{e}_{y'}$, since x, y are in the same cell and x', y' are in different cells. So we have $\|\Phi(x|\Lambda) - \Phi(y|\Lambda)\|_2 \leq (\|\Phi(x|\Lambda) - \mathbf{e}_x\|_2 + \|\Phi(y|\Lambda) - \mathbf{e}_y\|_2) \leq 2\tilde{\epsilon}$ and $\|\Phi(x'|\Lambda) - \Phi(y'|\Lambda)\|_2 \geq (\|\mathbf{e}_{x'} - \mathbf{e}_{y'}\|_2 - 2\tilde{\epsilon}) \geq \sqrt{2} - 2\tilde{\epsilon}$.

Then because $\tilde{\epsilon} = \frac{\sqrt{2}}{4}$, we have

$$\|\Phi(x|\Lambda) - \Phi(y|\Lambda)\|_2 \leq \|\Phi(x'|\Lambda) - \Phi(y'|\Lambda)\|_2,$$

which indicates that $s(x, y|\Lambda) \geq s(x', y'|\Lambda)$, since $\forall x \in \mathcal{X}, \|\Phi(x|\Lambda)\|_2 = 1$. □

A.3 Proof of Lemma 9

Lemma 9 $\hat{\kappa}(x, y \mid X)$ is a valid kernel.

Proof. We only need to show that the matrix produced by $\hat{\kappa}$ is a positive definite matrix (Muandet et al. 2017). The symmetry of the matrix comes from $\hat{\kappa}(x, y|X) = \hat{\kappa}(y, x|X)$, since $\forall \Lambda \in V(X)$, $s(x, y|\Lambda) = s(y, x|\Lambda)$.

Kernel $\hat{\kappa}(x, y \mid X)$ can be re-expressed in a quadratic form as follows:

$$\hat{\kappa}(x, y \mid X) = \frac{1}{t}\sum_{i=1}^{t} \Phi(x \mid \Lambda_i)^\top \Phi(y \mid \Lambda_i).$$

So the matrix produced by $\hat{\kappa}$ is positive definite. □

A.4 Proof of Lemma 11

Lemma 11 For $x, y \in X$, $d_\Lambda(x, y|X) \coloneqq 1 - \hat{\kappa}(x, y|X)$ is a distance metric when $\eta \to \infty$.

Proof. We prove d_Λ is a distance metric when $\eta \to \infty$ from four aspects:

(1) The distance from a point to itself is zero: $d_\Lambda(x, x|X) = 0$, since $\hat{\kappa}(x, x|X) = 1$.

(2) The distance between two distinct points x, y is always positive when $\eta \to \infty$:
$d_\Lambda(x, y|X)$ can be re-expressed as

$$d_\Lambda(x, y|X) = \frac{1}{t}\sum_{i=1}^{t}(1 - s(x, y|\Lambda_i)).$$

And we have $\lim_{\eta\to\infty} s(x, y|\Lambda_i) = 1$, if x, y are in the same Voronoi cell of Λ_i. Otherwise, $\lim_{\eta\to\infty} s(x, y|\Lambda_i) = 0$. If t is large enough, then there will be at least one $\hat{i} \in [t]$, such that $\lim_{\eta\to\infty} s(x, y|\Lambda_{\hat{i}}) = 0$. So $\lim_{\eta\to\infty} d_\Lambda(x, y|X) > 0$.

(3) The distance from x to y is always the same as the distance from y to x: $d_\Lambda(x, y) = d_\Lambda(y, x)$, since $d_\Lambda(x, y|X) = 1 - \hat{\kappa}(x, y|X) = 1 - \hat{\kappa}(y, x|X) = d_\Lambda(y, x|X)$.

(4) The triangle inequality holds, when $\eta \to \infty$:
$d_\Lambda(x, y|X)$ can be re-expressed as

$$d_\Lambda(x, y|X) = \frac{1}{2t}\sum_{k=1}^{t} d'(x, y|\Lambda_i),$$

where $d'(x, y|\Lambda_i) = \|\Phi(x|\Lambda_i) - \Phi(y|\Lambda_i)\|_2^2$.
So it suffices to prove that triangle inequality holds for $d'(x, y|\Lambda)$ when $\eta \to \infty$. The positions of x, y, z with respect to Voronoi cells of Λ when $\eta \to \infty$ can be divided into 4 cases:

a) x, y, z are in the same Voronoi cell θ_i: $d'(x, y|\Lambda) = d'(x, z|\Lambda) = d'(y, z|\Lambda) = 0$. Hence $d'(x, z|\Lambda) \le d'(x, y|\Lambda) + d'(y, z|\Lambda)$ holds.

b) x, z are in the same Voronoi cell θ_i; and y is in a different Voronoi cell θ_j: $d'(x, z|\Lambda) = 0$, $d'(x, y|\Lambda) = d'(y, z|\Lambda) = 2$. Hence $d'(x, z|\Lambda) \le d'(x, y|\Lambda) + d'(y, z|\Lambda)$ holds.

c) x, z are in different Voronoi cells θ_i and θ_j, respectively; and y is in either θ_i or θ_j: $d'(x, z|\Lambda) = 2$, $d'(x, y|\Lambda) = 0$ (or 2) and $d'(y, z|\Lambda) = 2$ (or 0). Hence $d'(x, z|\Lambda) \le d'(x, y|\Lambda) + d'(y, z|\Lambda)$ holds.

d) x, y, z are in three different Voronoi cells θ_i, θ_j and θ_k, respectively: $d'(x, y|\Lambda) = d'(x, z|\Lambda) = d'(y, z|\Lambda) = 2$. Hence $d'(x, z|\Lambda) \le d'(x, y|\Lambda) + d'(y, z|\Lambda)$ holds.

□

A.5 Proof of Property 12

Property 12 For any two points x, y in dense region $\mathcal{S}$ and their corresponding x', y' in sparse region $\mathcal{T}$, $d_\Lambda(x, y|S) \approx d_\Lambda(x', y'|T)$, where $S \subset \mathcal{S}$, $T \subset \mathcal{T}$, and $\mathcal{S} = \{m(\mathcal{T}) + r(x - m(\mathcal{T}))|x \in \mathcal{T}\}$ for $r > 1$.

Proof. Because one of S and T can be obtained by isotropic stretching from the other, there is a mapping h such that: $\forall x \in S, x' \in T, h(x) = x'$, where x' is the stretched version of x. S and T have the same topology implication: $O_x = O_{h(x)} = O_{x'}, \forall x \in S$, where O_x is the order: $y \le_x z \iff ||x - y|| \le ||x - z||$.

Let $\mathcal{D}$ be the subset sampled uniformly from S: $\mathcal{D} = \{x_1, x_2, ..., x_\psi\}$, $\mathcal{D}' = h(\mathcal{D}) = \{h(x_1), h(x_2), ..., h(x_\psi)\} = \{x'_1, x'_2, ..., x'_\psi\}$. The j-th Voronoi cell built from S and T: θ_j and θ'_j can be expressed as:

$$\theta_j = \left\{x \in \mathcal{S} | x_j \leq_x x_i, \forall i \in [\psi], i \neq j\right\}$$

$$\theta'_j = \left\{x' \in \mathcal{T} | x'_j \leq_{x'} x'_i, \forall i \in [\psi], i \neq j\right\} = h(\theta_j)$$

Then $\forall x, y \in \theta_j \iff x', y' \in \theta'_j$, where $x' = h(x), y' = h(y)$. And the following two equations hold when $\eta \to \infty$:

$$\frac{e^{-\eta\ell(x', x'_i)}}{\Upsilon'^m_\ell} = \frac{e^{-\eta\ell(x, x_i)}}{\Upsilon^m_\ell}, i \in 1, ...\psi,$$

$$\frac{e^{-\eta\ell(y', x'_i)}}{\Upsilon'^m_\ell} = \frac{e^{-\eta\ell(y, x_i)}}{\Upsilon^m_\ell}, i \in 1, ...\psi.$$

Hence $\Phi(x|\Lambda) = \Phi(x'|\Lambda')$ and $\Phi(y|\Lambda) = \Phi(y'|\Lambda')$. This directly produces the following result:

$$\begin{aligned}
\kappa(x', y'|T) &= \mathbb{E}_{\Lambda'}[s(x', y'|\Lambda')] \\
&= \frac{1}{|V(X)|} \sum_{\Lambda' \in V(X)} s(x', y'|\Lambda') \\
&= \frac{1}{|V(X)|} \sum_{\Lambda' \in V(X)} \langle \Phi(x'|\Lambda'), \Phi(y'|\Lambda') \rangle \\
&= \frac{1}{|V(X)|} \sum_{\Lambda \in V(X)} \langle \Phi(x|\Lambda), \Phi(y|\Lambda) \rangle \\
&= \frac{1}{|V(X)|} \sum_{\Lambda \in V(X)} s(x, y|\Lambda) \\
&= \mathbb{E}_{\Lambda}[s(x, y|\Lambda)] \\
&= \kappa(x, y|S).
\end{aligned}$$

Since $\hat{\kappa}$ is the empirical estimation of κ, for $x, y \in S$, $x', y' \in T$, and $x' = h(x), y' = h(y)$, we have the following result:

$$d_\Lambda(x, y|S) = d_\Lambda(x', y'|T) \equiv \hat{\kappa}(x, y|S) = \hat{\kappa}(x', y'|T).$$

□

A.6 Proof of Theorem 13

Theorem 13 The bottleneck distance between two persistence diagrams $\mathfrak{D}(f)$ and $\mathfrak{D}(g)$, derived from Λ-filters $f(X)$ and $g(X')$, respectively, is bounded as follows:

$$W_\infty(\mathfrak{D}(f), \mathfrak{D}(g)) \leq \|f - g\|_\infty \leq d_H(X, X'),$$

where X' is the perturbed version of X; and $d_H(X, X')$ is the Hausdorff distance between X and X'.

Proof. Let X be the original point cloud and X' be the perturbed point cloud. Let $\mathcal{D}$ and $\mathcal{D}'$ be two given nonempty subsets from X and X', respectively, where $|\mathcal{D}| = |\mathcal{D}'| = \psi$. $\forall \phi \in \Psi = \{1, 2, ..., \psi\}$. Let $A_\phi = \cup_{j \neq \phi} \mathcal{D}_j$, $A'_\phi = \cup_{j \neq \phi} \mathcal{D}'_j$, $\delta = d_H(X, X')$.

Definition 19. *Given two nonempty subsets $A, B \subseteq X$, $\forall x \in X$, $dom(A, B) = \{x \in X : \ell(x, A) \leq \ell(x, B)\}$, here $\ell(x, B) = \inf\{\ell(x, b) : b \in B\}$, $\ell(A_1, A_2) = \inf\{\ell(a_1, a_2) : a_1 \in A_1, a_2 \in A_2\}$.*

Definition 20. $R_\phi = dom(\mathcal{D}_\phi, \cup_{j\neq\phi}\mathcal{D}_j) = \{x \in X : \ell(x, \mathcal{D}_\phi) \leq \ell(x, \mathcal{D}_j),\ \forall j \in \Psi \setminus \phi\}$.

In other words, the Voronoi cell R_ϕ associated with the partition point $\mathcal{D}_\phi$ is the set of all $x \in X$ whose distance to $\mathcal{D}_\phi$ is not greater than their distance to the union of the other sites $\mathcal{D}_j$.

Suppose that the following conditions hold:

(1) $\eta := \inf\{\ell(\mathcal{D}_\phi, \mathcal{D}'_j) : j, \phi \in \psi, j \neq \phi\} > 0$,

(2) $\exists\, \alpha \in (0, \infty)$ such that for all $\phi \in \Psi$ and for all $x \in X$, the open ball $B(x, \alpha)$ intersects A_ϕ.

Then the following theorem holds (Reem 2011):

Theorem 21. *For each $\phi \in \Psi$ let $R_\phi = dom(\mathcal{D}_\phi, A_\phi)$, $R'_\phi = dom(\mathcal{D}'_\phi, A'_\phi)$ be the Voronoi cells associated with the original $\mathcal{D}_\phi$ and the perturbed one $\mathcal{D}'_\phi$ respectively. Then for each $\epsilon \in (0, \eta/6)$, $\exists\Delta > 0$ such that if $d_H(\mathcal{D}_\phi, \mathcal{D}'_\phi) < \Delta$ for each $\phi \in \Psi$, then $d_H(R_\phi, R'_\phi) < \epsilon$ for each $\phi \in \Psi$.*

Consider the points close the boundary l and l' of two partitions, which may belong to different partitions after perturbation. Let $R_c = \{x \in X : d(x, l) < \delta || d(x, l') < \delta\}$, where the $\delta > \epsilon$ is the maximum perturbation distance. R_c can be divided into two parts. First, the region between l and l', the points in this region must be changed. Second, the partition of other points in the other area be changed with 50 percent probability. So the probability for a point x to change the partition is:

$$P(C(x)) = \frac{\epsilon + \frac{\delta-\epsilon+\delta}{2}}{\eta} = \frac{0.5\epsilon + \delta}{\eta} \leq \frac{3\delta}{2\eta}.$$

After doing partition for t times, the probability (Short 2013) of the partitions of point x be changed $\delta' * t$ $(0 < \delta' \leq \delta)$ times is:

$$\begin{aligned}
&P(d_{kH}(\Phi(X), \Phi(X')) \leq \delta') \\
&= \lim_{t\to\infty} \sum_{i=0}^{\lfloor t*\delta' \rfloor} \frac{t!}{(t-i)!(i)!} \left(\frac{3\delta}{2\eta}\right)^i * \left(1 - \frac{3\delta}{2\eta}\right)^{t-i} \\
&= e^{-t\left(\frac{3\delta}{2\eta}\right)} \sum_{i=0}^{\lfloor t*\delta' \rfloor} \frac{\left(\frac{3\delta}{2\eta}\right)^i}{i!} \\
&\geq e^{-t\left(\frac{3\delta}{2\eta}\right)},
\end{aligned}$$

where $\Phi(x)$ is the feature of x mapped by kernel k, and $d_{kH}(\Phi(X), \Phi(X')$ is the Hausdorff distance between X and X' based on $1- < \Phi(X), \Phi(X') >$.

As $\delta << \eta$. $d_{kH}(\Phi(X), \Phi(X')) \leq \delta$ holds with high probability more than $e^{-t\left(\frac{3\delta}{2\eta}\right)}$.

For $c > 4$, we have

$$\begin{aligned}
&P(d_{kH}(\Phi(X), \Phi(X')) \leq \delta/c) \\
&= \lim_{t\to\infty} \sum_{i=0}^{\lfloor t*\delta/c \rfloor} \frac{t!}{(t-i)!(i)!} \left(\frac{3\delta}{2\eta}\right)^i * \left(1 - \frac{3\delta}{2\eta}\right)^{t-i} \\
&= e^{-t\left(\frac{3\delta}{2\eta}\right)} \sum_{i=0}^{\lfloor t*\delta/c \rfloor} \frac{\left(\frac{3\delta}{2\eta}\right)^i}{i!} \\
&\geq e^{-t\left(\frac{3\delta}{2\eta}\right)},
\end{aligned}$$

So $d_{kH}(\Phi(X), \Phi(X')) \leq \delta/c$ holds with high probability. And we have

$$\begin{aligned}\|f-g\|_\infty &= 2\|\ell^2(\Phi(\cdot), \Phi(X)) - \ell^2(\Phi(\cdot), \Phi(X'))\|_\infty \\ &\leq 2\|(\ell(\Phi(\cdot), \Phi(X)) - \ell(\Phi(\cdot), \Phi(X')) \\ &(\ell(\Phi(\cdot), \Phi(X)) + \ell(\Phi(\cdot), \Phi(X'))\|_\infty \\ &\leq 4\|\ell(\Phi(\cdot), \Phi(X)) - \ell(\Phi(\cdot), \Phi(X')\|_\infty \\ &\leq 4 * d_{kH}(\Phi(X), \Phi(X'))\end{aligned}$$

Finally, we get

$$\begin{aligned}W_\infty(\mathfrak{D}(f), \mathfrak{D}(g)) &\leq \|f-g\|_\infty \leq 4 * d_{kH}(\Phi(X), \Phi(X')) \\ &\leq \delta \leq d_H(X, X'),\end{aligned}$$

where the d_H is the Hausdorff distance, $d_H(X, X') := \max\left\{\sup_{x\in X} d(x, X\prime), \sup_{x\prime\in X,} d(X, x\prime)\right\}$, f and g is the filter function of X and X' respectively.

□

A.7 Proof of Theorem 15

Theorem 15 When using Λ-filter, if $\psi^{2t} > (1-e^{-2\eta R})n^{2t}$, then for any order $k > 0$, $\mathfrak{D}_k^\delta(X \cup X_n) = \mathfrak{D}_k^\delta(X)$ holds with probability more than $(1-\frac{m}{n+m})^{\psi t}$, where $\mathfrak{D}_k^\delta$ stands for persistence diagram of the k-th order when the max filtration value is δ , R stands for the diameter of $X \cup X_n$, $\delta = 1-e^{-2\eta R}$, and $n = |X|, m = |X_n|$.

Proof. When building Λ-filter on $X \cup X_n$, the probability that all the ψ points in t repetitions are sampled from X is $(1-\frac{m}{m+n})^{\psi t}$. When this happens, for any $x, y \in X, x \neq y$, we have

$$\begin{aligned}&d_\Lambda(x, y|X \cup X_n) = d_\Lambda(x, y|X) \\ &= 1 - \frac{1}{t}\sum_{j=1}^{t} \frac{\sum_{i=1}^{\psi} e^{-\eta(d(x,x_i^j)+d(y,x_i^j))}}{\sqrt{\sum_{i=1}^{\psi} e^{-2\eta d(x,x_i^j)}}\sqrt{\sum_{i=1}^{\psi} e^{-2\eta d(y,x_i^j)}}} \\ &\leq 1 - \frac{1}{t}\sum_{j=1}^{t} \frac{\sum_{i=1}^{\psi} e^{-2\eta R}}{\psi} \\ &\leq 1 - e^{-2\eta R},\end{aligned} \tag{3}$$

where $x_i^j \in X$ is the i-th sampled point in j-th repetition.

Then for noise set X_n, we assume that for any two different $u, v \in X_n$, their nearest neighbors in X in terms of Euclidean distance d are different. For any $u, v \in X_n$, $d_\Lambda(u, v|X \cup X_n) > (\frac{\psi}{n})^{2t}$ when $\eta \to \infty$. So there exists an $\hat{\eta}$, such that $d_\Lambda(u, v|X \cup X_n) \geq (\frac{\psi}{n})^{2t}$.

Then $\forall x, y \in X(x \neq y), u, v \in X_n(u \neq v)$, we have

$$d_\Lambda(u, v|X \cup X_n) > \delta \geq d_\Lambda(x, y|X \cup X_n), \tag{4}$$

if $\psi^{2t} > (1-e^{-2\eta R})n^{2t}$. There is no connection between any two points in X_n before δ. All connections within X_n can only happen after X is pairwise connected, when all the topological features of X are dead. Although X_n can be connetced to X, since there's no connection in X_n, the noise set X_n has no impact on the birth and death of the X's topological features that have order higher or equal than 1. Hence, we have

$$\mathfrak{D}_k^\delta(X \cup X_n) = \mathfrak{D}_k^\delta(X), \tag{5}$$

where $k > 0$.
Note that 0-dimensional PD of X and that of $X \cup X_n$ are different, since the number of points □

A.8 Proof of Theorem 17

Theorem 17 The PD produced via Λ-filter is robust to varied densities.

Proof. Because one of S and T can be obtained by isotropic stretching from the other, there is a mapping h such that: $\forall x \in S, x' \in T, h(x) = x'$, where x' is the stretched version of x. For $x,\ y \in S$, let $x' = h(x), y' = h(y)$. From Section A.5 we have $\hat{\kappa}(x, y|S) = \hat{\kappa}(x', y'|T)$, hence

$$\begin{aligned}\hat{f}_\Lambda(x|S) &= 4 \min_{y \in S}(1 - \hat{\kappa}(x, y|S)) \\ &= 4 \min_{y' \in T}(1 - \hat{\kappa}(x', y'|T)) \\ &= \hat{f}_\Lambda(x'|T)\end{aligned}$$

□

A.9 Proof of CkNN's Filter Function

In CkNN, x_i and x_j are connected if $\ell(x_i, x_j) \leq \epsilon\sqrt{\ell_k(x_i)\ell_k(x_j)}$. And the filtration is formed by varying ϵ. We state the connection between x_i and x_j in an equivalent form in Lemma 22, which is used for the proof of Theorem 23.

Lemma 22. *$\forall \epsilon > 0$, $\frac{\ell(x_i,x_j)}{\sqrt{(\ell_k(x_i)\ell_k(x_j)}} \leq \epsilon$ iff $\exists x, s.t.\ \frac{\ell(x,x_i)}{\sqrt{\ell_k(x_i)}} w(i,j) \leq \epsilon$ and $\frac{\ell(x,x_j)}{\sqrt{\ell_k(x_j)}} w(i,j) \leq \epsilon$, where $w(i,j) = \frac{1}{\sqrt{\ell_k(x_i)}} + \frac{1}{\sqrt{\ell_k(x_j)}}$.*

Proof. (1) If $\exists x, s.t. \frac{\ell(x,x_i)}{\sqrt{\ell_k(x_i)}} \leq \frac{\epsilon}{w(i,j)}$ and $\frac{\ell(x,x_j)}{\sqrt{\ell_k(x_j)}} \leq \frac{\epsilon}{w(i,j)}$, then by triangle inequality we have

$$\frac{\ell(x_i, x_j)}{\sqrt{\ell_k(x_i)\ell_k(x_j)}} \leq \frac{\ell(x, x_i) + \ell(x, x_j)}{\sqrt{\ell_k(x_i)\ell_k(x_j)}} \leq \epsilon.$$

(2) If $\frac{\ell(x_i,x_j)}{\sqrt{\ell_k(x_i)\ell_k(x_j)}} \leq \epsilon$, choose $x = \mu x_i + (1-\mu)x_j$,where $\mu = \frac{\sqrt{\ell_k(x_j)}}{\sqrt{\ell_k(x_i)}+\sqrt{\ell_k(x_j)}}$. Then we get

$$\begin{aligned}\frac{\ell(x, x_i)}{\sqrt{\ell_k(x_i)}} = \frac{\ell(x, x_j)}{\sqrt{\ell_k(x_j)}} &= \frac{1}{w(i,j)} \frac{\ell(x_i, x_j)}{\sqrt{\ell_k(x_i)\ell_k(x_j)}} \\ &\leq \frac{\epsilon}{w(i,j)}.\end{aligned}$$

□

Define the graph $G(\epsilon)$ formed via CkNN under the scale parameter ϵ as

$$G_\epsilon(X) = \left\{(x_i, x_j) \in X^2 | \frac{\ell(x_i, x_j)}{\sqrt{\ell_k(x_i)\ell_k(x_j)}} \leq \epsilon\right\}.$$

Then we can have an approximation of CkNN's filter function, as shown in Theorem 23.

Theorem 23. *$\forall \epsilon > 0, G^f_\epsilon(X) \subset G_\epsilon(X) \subset G^g_\epsilon(X)$, where*

$$G^f_\epsilon(X) = \left\{x_i, x_j) \in (X \cap \mathcal{X}^f_\epsilon)^2 | B(x_i, r^f_i(\epsilon)) \cap B(x_j, r^f_j(\epsilon)) \neq \emptyset\right\},$$

$$G^g_\epsilon(X) = \left\{(x_i, x_j) \in (X \cap \mathcal{X}^g_\epsilon)^2 | B(x_i, r^g_i(\epsilon)) \cap B(x_j, r^g_j(\epsilon)) \neq \emptyset\right\},$$

$$f(x) = \min_{i\in[n]} \max_{j\in[n]} \left(\frac{\ell(x, x_i)}{\sqrt{\ell_k(x_i)}} w(i, j) \right),$$

$$g(x) = \min_{i\in[n]} \min_{j\in[n]} \left(\frac{\ell(x, x_i)}{\sqrt{\ell_k(x_i)}} w(i, j) \right),$$

$$\mathcal{X}_\epsilon^f = \{x \in \mathcal{X} | f(x) \le \epsilon\} = \cup_{i=1}^n B\left(x_i, r_i^f(\epsilon)\right),$$

$$\mathcal{X}_\epsilon^g = \{x \in \mathcal{X} | g(x) \le \epsilon\} = \cup_{i=1}^n B\left(x_i, r_i^g(\epsilon)\right),$$

$r_i^f(\epsilon) = \frac{\sqrt{\ell_k(x_i)}}{max_{p\in[n]} w(i,p)}\epsilon$ and $r_i^g(\epsilon) = \frac{\sqrt{\ell_k(x_i)}}{min_{p\in[n]} w(i,p)}\epsilon$. B is defined in Section 6, and $w(\cdot,\cdot)$ is defined in lemma 22.

Proof. Unlike the case in the Rips filtration, where the radius of a ball centered at x_i is the given scale parameter ϵ, we assign individual radius r_i^f to each x_i under given $r_i^f(\epsilon) = \frac{\sqrt{\ell_k(x_i)}}{max_{p\in[n]} w(i,p)}\epsilon$.

If $(x_i, x_j) \in G_\epsilon^f(X))$, $B\left(x_i, r_i^f(\epsilon)\right)$ intersects $B\left(x_j, r_j^f(\epsilon)\right)$. We have that $\exists\, x,\ s.t.\, \ell(x, x_i) \le r_i^f(\epsilon), \ell(x, x_j) \le r_j^f(\epsilon)$, then we have

$$\frac{\ell(x, x_i)}{\sqrt{\ell_k(x_i)}} w(i, j) \le \frac{\ell(x, x_i)}{\sqrt{\ell_k(x_i)}} max_{p\in[n]} w(i, p) \le \epsilon,$$
$$\frac{\ell(x, x_j)}{\sqrt{\ell_k(x_j)}} w(i, j) \le \frac{\ell(x, x_j)}{\sqrt{\ell_k(x_j)}} max_{p\in[n]} w(j, p) \le \epsilon.$$

So according to Lemma 22, we can get $\frac{\ell(x_i,x_j)}{\sqrt{\ell_k(x_i)\ell_k(x_j)}} \le \epsilon$, that is to say, $(x_i, x_j) \in G_\epsilon(X)$. Hence $G_\epsilon^f(X) \subset G_\epsilon(X)$.

Similarly, we can have $G_\epsilon(X) \subset G_\epsilon^g(X)$ with $r_i^g(\epsilon) = \frac{\sqrt{\ell_k(x_i)}}{min_{p\in[n]} w(i,p)}\epsilon$ as following: if $(x_i, x_j) \in G_\epsilon(X)$, then $\frac{\ell(x_i,x_j)}{\sqrt{\ell_k(x_i)\ell_k(x_j)}} \le \epsilon$. Then by Lemma 22, there exists x such that

$$\ell(x, x_i) \le \frac{\sqrt{\ell_k(x_i)}}{w(i, j)}\epsilon \le \frac{\sqrt{\ell_k(x_i)}}{min_{p\in[n]} w(i, p)}\epsilon = r_i^g(\epsilon),$$
$$\ell(x, x_j) \le \frac{\sqrt{\ell_k(x_j)}}{w(i, j)}\epsilon \le \frac{\sqrt{\ell_k(x_j)}}{min_{p\in[n]} w(j, p)}\epsilon = r_j^g(\epsilon).$$

□

As for the precision of the approximation, $\frac{r_i^f}{r_i^g}$ measures how well the approximation is. We have the bound of $\frac{r_i^f}{r_i^g}$ as follows:

$$\sqrt{\frac{min_{p\in[n]}\ell_k(x_p)}{max_{p\in[n]}\ell_k(x_p)}} \le \frac{1 + \frac{\sqrt{\ell_k(x_i)}}{\sqrt{max_{p\in[n]}\ell_k(x_p)}}}{1 + \frac{\sqrt{\ell_k(x_i)}}{\sqrt{min_{p\in[n]}\ell_k(x_p)}}} \le \frac{r_i^f}{r_i^g} \le 1.$$

A higher lower bound leads to a better approximation of CKNN's filter function. But this lower bound can be very small if the densities are hugely different.

To get a better approximation, it is reasonable to assume that x_i only connects with its k nearest neighbors when ϵ is relatively small. Then we can get $\hat{f}, \hat{g}$ as better approximations of CkNN's filter function, where

$$\hat{f}(x) = \min_{i\in[n]} \max_{j\in ID_k(x_i)} \left(\frac{\ell(x, x_i)}{\sqrt{\ell_k(x_i)}} w(i, j) \right),$$

$$\hat{g}(x) = \min_{i\in[n]} \min_{j\in ID_k(x_i)} \left(\frac{\ell(x, x_i)}{\sqrt{\ell_k(x_i)}} w(i, j) \right)$$

and $ID_k(x_i)$ is the set of indexes of x_i's k nearest neighbors in X.

Then for each x_i, we can improve the lower bound from $\sqrt{\frac{min_{p\in[n]}\ell_k(x_p)}{max_{p\in[n]}\ell_k(x_p)}}$ to $\sqrt{\frac{min_{p\in ID_k(x_i)}\ell_k(x_p)}{max_{p\in ID_k(x_i)}\ell_k(x_p)}}$, while $G_\epsilon(X)$ is bounded by $G_\epsilon^{\hat{f}}(X)$ and $G_\epsilon^{\hat{g}}(X)$.

However, the goal is to use one function to approximate CkNN's filter function instead of two bounds $\hat{f}$ and $\hat{g}$. To achieve this goal, the neighborhood of x_i is assumed to be uniformly distributed, i.e.

$$\min_{p\in ID_k(x_i)} \ell_k(x_p) = \max_{p\in ID_k(x_i)} \ell_k(x_p) = \ell_k(x_i).$$

And finally we can have a good approximation $h(x)$ of CkNN's filter function in the early stage of the topological feature's formation process (small ϵ), where

$$h(x) = \min_{i\in[n]} \frac{2\ell(x, x_i)}{\ell_k(x_i)}.$$

B Additional Experiments

B.1 Robustness to Noise

This section provides the details of Figure 3 described in Section 7. It presents the advantages of Λ-filter over other methods in terms of robustness to noise.

Table 12 shows the Cassini dataset and corresponding 1-dim PDs, built from GK-filter, DTM and Λ-filter when $\gamma = 0, 0.025$ and 0.5. The noise points are sampled from a uniformly distribution ρ_n.

When $\gamma = 0$ (no noise), only Λ-filter detects one single ring without any noise feature. GK-filter detects the valid ring, but with two additional rings due to noise. DTM detects two equally persistent rings (two overlapping points in PD).

When adding some noise ($\gamma = 0.025$) in the Cassini dataset, shown in the second row, the PD of GK-filter is dramatically different from that obtained previously ($\gamma = 0$). There are many misleading rings due to the effect of noise, resulting in the structure of the data cannot be correctly judged from the PD. DTM and Λ-filter are robust to noise. But DTM obtains two rings incorrectly, and the two rings have different sizes.

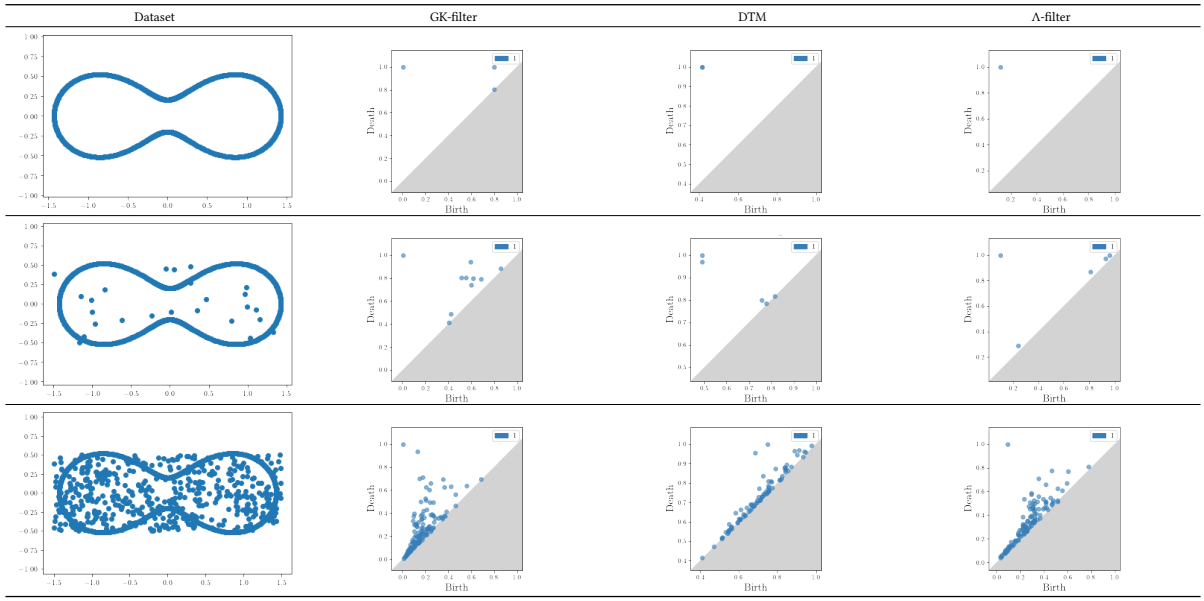
When γ reaches 0.5, only Λ-filter produces a PD which has one distinct persistent ring.

B.2 Property 12 of d_Λ

This section provides an example to show that d_Λ has the ability of preserve the distance after stretching or shrinking (Property 12 in Section 5.1).

The root cause of a wrong PD is that when using Euclidean distance, points are closer to each other in the dense region than in the sparse region. A direct way to address this issue is to design a data-dependent distance metric that enables the distance between two points to remain almost the same after stretching or shrinking. We will show that d_Λ is such a distance metric in Figure 12, where the dense cluster T is obtained by a tenfold shrinking of sparse cluster S.

Table 12. The PDs (1-dimensional homology are reported) of the data with noise of three different ratios ($\gamma = 0, 0.025, 0.5$).



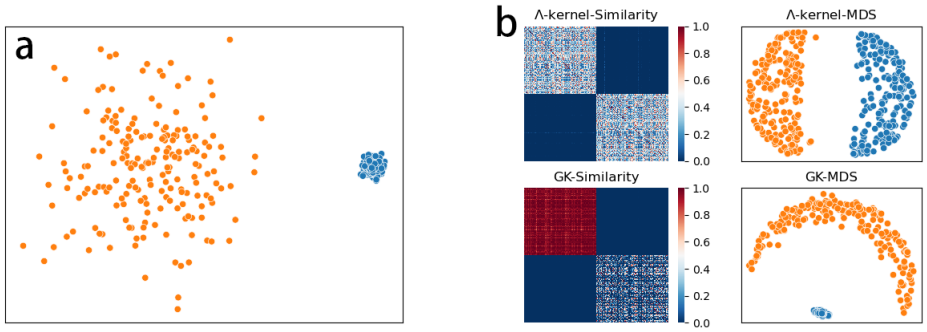


Fig. 12. (a) Original data with sparse and dense clusters. (b) Similarity matrices and MDS visualizations of Λ-kernel and GK.

In Figure 12, Λ-kernel is compared with Gaussian Kernel (GK), which completely depends on Euclidean distance. In the case of GK, the points in the dense cluster are much more similar to each other than those in the sparse cluster. While in the case of Λ-kernel, the similarity matrices of the sparse cluster and dense cluster are quite similar to each other.

The MDS (Multidimensional Scaling (Mead 1992)) visualization, which preserves the distance in the mapped space, maps the given dataset via distance matrix, is shown in Figure 12b.

Clusters S and T have the same density in the MDS-mapped space when Λ-kernel is employed; but they still have different densities when GK is employed.

This illustrates that d_Λ can preserve the distance after stretching or shrinking. That is to say, d_Λ has Property 12.

In addition, Section 6 states that using f_Λ as a filter function for PH is equivalent to using $d_\Lambda(x, y)$ to replace Euclidean distance in the Rips filter function. This subsection demonstrates that Λ-filter has Property 4 experimentally.

In summary, the PD built from Λ-filter is robust to varied densities.

B.3 Details of Table 5: Growth of Sublevel Sets

This section gives the corresponding growth of sublevel sets of the Persistence Diagrams in Table 5 from Section 8. For each filter function, 3 snippets of the growth of sublevel sets are provided in Table 13.

Table 13 shows some details of the PH by three sublevel sets of different filter functions. First, when DTM and Rips are employed, the right ring died before the left ring was born. Second, when CkNN is employed, in addition to the persistent rings, many other rings are formed due to the influence of noise. Finally, when Λ-filter is employed, two prominent rings are born at about the same time and then die at about the same time, and the persistence of other noise rings is extremely small.

B.4 Parameter Setting

This section provides the parameter setting of the experiments in Section 7.

Parameter setting used in the experiments: For Λ-kernel, $t = 200$, $\eta = \infty$, ψ is searched over $\{2, 4, 8, 16, 32\}$. For DTM and CkNN, the k is searched in $\{m * n | m = 0.02, 0.04, 0.06, 0.08, 0.1\}$, where n is the dataset size. The experiments are performed on a machine with 1500MHz CPUs and 2TB RAM.

B.5 Details of Experiments

B.5.1 Classification of Bone Scripts. This section describes the experiment setting of the bone scripts classification from Section 7.

We use a dataset which contains ten classes of bone scripts, which are referred to as ten 'heavenly stems' in Chinese culture. The dataset has 20 images of the bone scripts in each class. We compress each image from the original 400*400 pixels to 120*120 pixels for efficient PD computation. The pixels of each script image are extracted as points in a 2-dimensional point cloud, and 20 noise points sampled from a uniform distribution are added.

In Figure 6a, we run SVM for 10 random train/test splits of the dataset and report the mean accuracy and standard deviation. In each split, we take 70% of the whole dataset for training and 30% for testing. 3-fold cross-validation on the training set is used to select the best hyperparameters for each approach: smoothing factor of rbf kernel, regularization weight in SVM, filtration factor (k for DTM and CkNN, ψ for Λ-filter).

The result of one such split of the dataset (with random seed set to 2022) is shown in Figure 13 in the form of confusion matrices. These results demonstrate the superiority of PDs produced by Λ-filter, where Λ-filter has the highest accuracy.

In Figure 6b, for each (η, ψ) pair, we select the best PI bandwidth to explore the effect of η, ψ in Λ-filter.

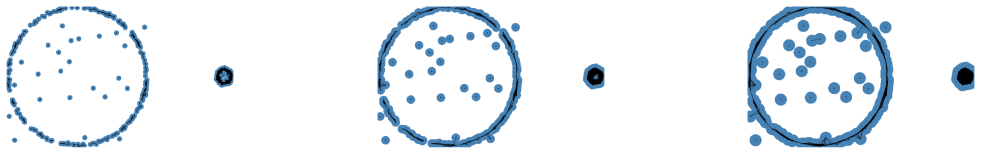
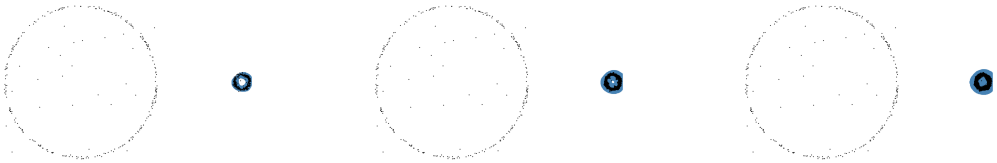
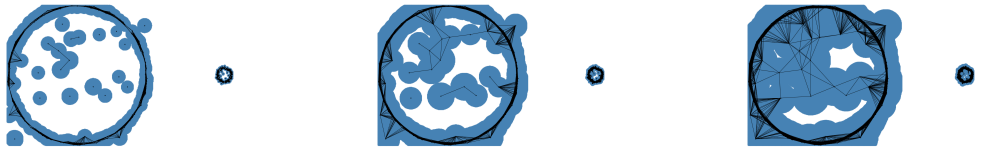
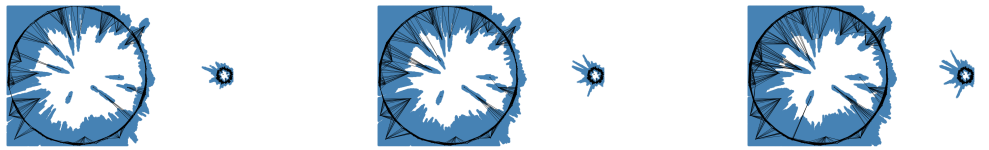
B.5.2 Λ-kernel for UMAP Dimensionality Reduction. For UMAP, the number of neighbors is search in [5, 10, 20, 50, 100, 200]. For Λ-MAP, ψ is searched in [2,4,8,16,32,64], $\eta = \infty$ and $t = 500$.

B.6 Multi-parameter Persistent Homology

This section shows the flaw of Multiparameter Persistent Homology, mentioned in Section 1.

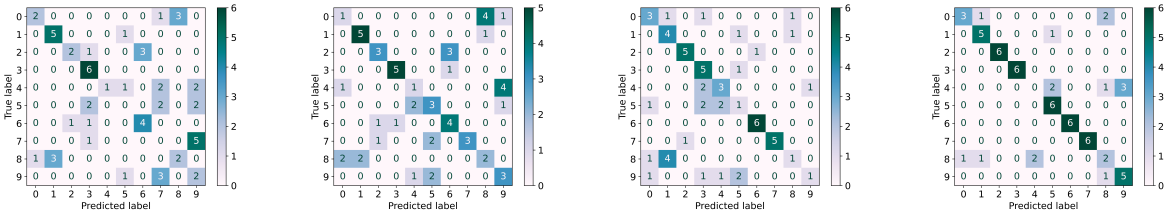
Multi-parameter Persistent Homology (MPH (Blumberg and Lesnick 2022; Vipond 2020)) claims that it is robust to varied densities. When a data cloud has varied densities, MPH is claimed to be able to detect the features in

Table 13. Growth of sublevel sets for each of the four filter functions.

			
Rips	$\epsilon = 0.5$	$\epsilon = 1.0$	$\epsilon = 1.5$
			
DTM	$\epsilon = 0.6$	$\epsilon = 0.7$	$\epsilon = 0.8$
			
CkNN	$\epsilon = 0.3$	$\epsilon = 0.5$	$\epsilon = 0.7$
			
Λ-filter	$\epsilon = 0.125$	$\epsilon = 0.150$	$\epsilon = 0.175$

sparse and dense regions. We examine this claim in the following experiment with Multiparameter Persistence Landscape (MPL (Vipond et al. 2021)), a stable representation of PD that generalizes from Persistence Landscape (PL (Bubenik 2015)). MPL is presented in the form of a matrix, where each column represents a PL.

We employ a commonly-used MPL (Rips and 1NN Codensity) on the point cloud shown in right subfigure in Figure 7, where there is one dense ring, one sparse ring, and some noise. The result is shown in Figure 14. MPH



(a) Rips (b) DTM (c) CkNN (d) Λ-filter

Fig. 13. Confusion matrices

can detect the two rings in the first landscape, and the dense ring can be detected earlier with respect to 1NN Codensity. The second landscape contains the noise features, which are well separated from the true topological features in the first landscape.

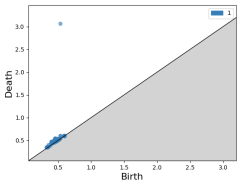
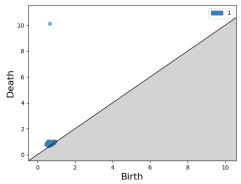
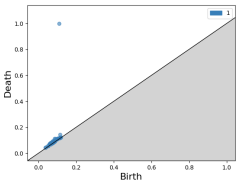
Note that although the two rings can be detected in the first MPL, the persistence of the dense ring is still significantly smaller than that of the sparse ring, which is the same issue we met in the Rips filtration.

In summary, MPH can tackle the problem of noise and it does not deal with the problem of varied densities satisfactorily.

B.7 Using Geodesic Distance in Different Filters on Eyeglasses Dataset

The result of using Geodesic distance in different filters on the eyeglasses dataset is shown in Table 14.

Table 14. Persistence Diagram obtained by using Geodesic distance in DTM,CkNN,Λ-filter on the eyeglasses dataset from Figure 10 a.

	DTM	CkNN	Λ-filter
Persistence Diagram			

C Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP)

C.1 UMAP Introduction

Here we briefly introduce UMAP [9]: UMAP is a novel technique for reducing the dimensionality of high-dimensional data, which is based on the assumption that there exists a manifold on which the data would be uniformly distributed. UMAP finds a low-dimensional representation of the data that preserves the *topological structure* of this manifold by minimizing the cross-entropy between the two topological representations (simplicial set) of high-dimension and low-dimensional data. (McInnes et al. 2018) achieve this by the following 5 steps:

* Graph Construction

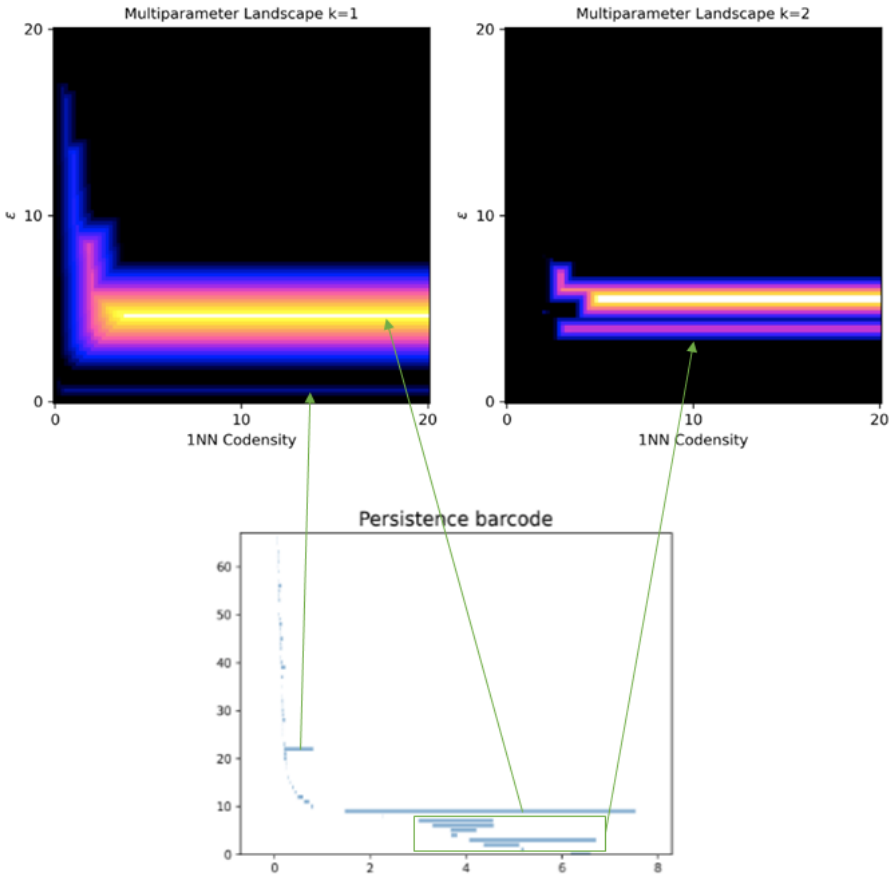


Fig. 14. First MPL and second MPL, and PD from the Rips filtration.

1. Construct a k-NN graph (using Euclidean distance).
2. Apply some transform on the edges of the k-NN graph to ambient local distance (see Equation 6 below).
3. Deal with the inherent asymmetry of the k-NN graph (see Equation 7 below).

* Optimized Representation

4. Define an objective function that preserves the desired characteristics of this k-NN graph.
5. Use Spectral Embedding as initialization and find a low dimensional representation which optimizes this objective function.

[9]The code of UMAP is available at https://github.com/lmcinnes/umap/tree/master.

UMAP defines a weighted k-NN graph $\mathcal{G} = (V, E, w)$. The vertices V are X, $E = \{(x_i, x_{i_j}) | 1 \leq j \leq k, 1 \leq i \leq |X|\}$ (x_{i_j} is x_i's j-th nearest neighbor), w is defined as follows:

$$w(x_i, x_{i_j}) = exp\left(\frac{-max(0, \ell(x_i, x_{i_j}) - \rho_i)}{\sigma_i}\right), \tag{6}$$

where

$$\rho_i = min\{\ell(x_i, x_{i_j}) | 1 \leq j \leq k, \ell(x_i, x_{i_j}) > 0\}$$

and σ_i is set to be the value such that

$$\sum_{j=1}^{k} exp\left(\frac{-max(0, \ell(x_i, x_{i_j}) - \rho_i)}{\sigma_i}\right) = log_2(k).$$

Let A be the weighted adjacency matrix of $\mathcal{G}$ via Equation 6, which may be asymmetric. UMAP gets its symmetric version

$$B = A + A^\top - A \circ A^\top, \tag{7}$$

where $\circ$ is the Hadamard (or pointwise) product.

C.2 Limitation of UMAP

In this subsection, we give an illustrative example of UMAP's limitation. Suppose we have a dataset consisting of two clusters, whose boundaries are two spheres, as shown in Figure 15, and the minimal pairwise distance inside each cluster equal to 1 and the minimal distance between two clusters equals to 9.

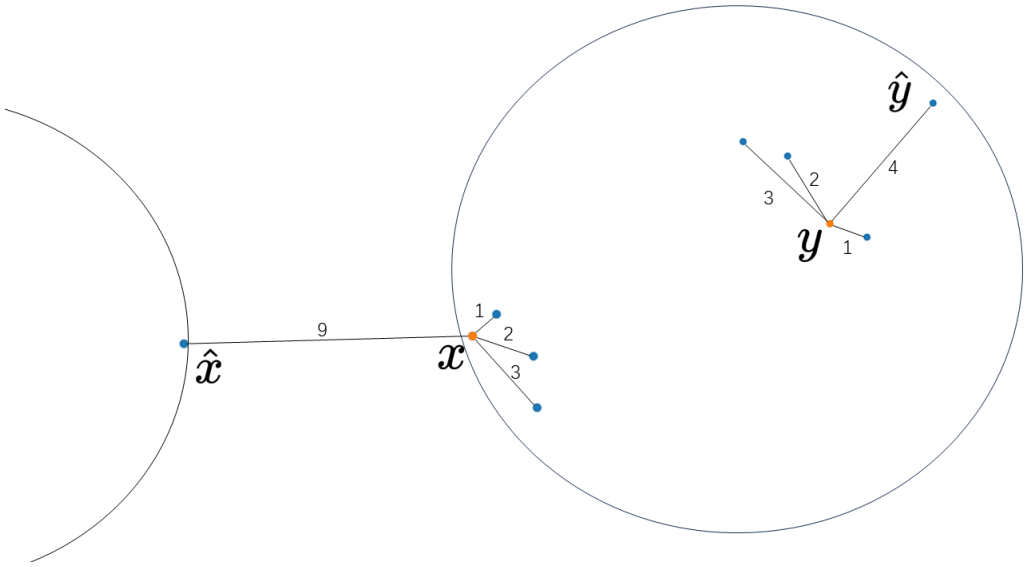


Fig. 15. Example of UMAP's limitation.

In the right cluster, we visualize two points x, y, together with k nearest neighbors (k=4). Note that there are other points in the cluster, we just didn't plot them, we only plot x, y for clarity. The k nearest neighbors of y are all within the right cluster, with distance to y equals to $1, 2, 3, 4$. 3 nearest neighbors of x are within the right

cluster (with distance to x equals to 1, 2, 3), and one nearest neighbors $\hat{x}$, whose distance to x is 9, is from the left cluster.

We obtain UMAP weight through Equation 6: $w(x, \hat{x}) = 0.02, w(y, \hat{y}) = 0.54$. By treating $1 - w$ as distance, and normalizing the distance (maximum distance equals to 1), we can see that the minimal distance between two clusters changes from 9 in the original space to 2.13 in the metric space UMAP produces.

In summary, the limitation of UMAP is that the inter-cluster distance can be significantly reduced when the k-NN graph connects points between different clusters.

C.3 Intuition of Λ-MAP

We replace B with Λ-kernel, that is, replacing steps 2 and 3 in the original UMAP with one step that computes Λ-kernel and the new UMAP algorithm using Λ-kernel is named Λ-MAP.

Since the aim of UMAP is to preserve the topological structure of the data, and as we mentioned earlier, when there are varied densities between different clusters, the topology of sparse clusters interferes with the topology of dense clusters. Hence, a density-invariant similarity is desired. Across-cluster Uniformity Similarity, as defined in 24, is such one. Λ-distance , as defined in Lemma 11 ($d_\Lambda(x, y|X) := 1 - \hat{\kappa}(x, y|X)$), is an Across-cluster Uniformity Similarity, which is a direct result of Property 12. UMAP's weight $1 - B$ is also an Across-cluster Similarity, since global uniformity ensures across-cluster uniformity.

In order to ensure the spectral embedding (the initialization of Optimization in step 5) of two clusters of different densities to be the same in terms of density in the embedded space, UMAP uses the global uniform metric B. However, it's only necessary to use an Across-cluster Uniformity Similarity like Λ-distance. Compared with the metric used in UMAP, Λ-distance can better preserve the intra-cluster density and enlarge inter-cluster distance. When the minimal Euclidean distance between two clusters A, B is smaller than the diameter of each cluster, $\hat{\kappa}(a, b|A \cup B) = 0$ for any pair $(a, b) \in A \times B$ as $\eta \to \infty$.

Definition 24. ***Across-cluster Uniformity Similarity***
*Given a dataset $X = \{C_1, ..., C_k\} \subset R^d$ that satisfies **Assumption 1**, $\mathfrak{M}$ is an Across-cluster Similarity Metric if for any $i \in [k], x \in C_i, \tau > 0, \forall j \neq i, B(h_{i,j}(x), \tau)$ under metric $\mathfrak{M}$ covers the same number of points in X, where $B(x, \tau) := \{y \in \mathbb{R}^d | \mathfrak{M}(x, y) \geq \tau\}$.*

Λ-distance is an Across-cluster Uniformity Metric, which is a direct result of Property 12.

Assumption 1 : a dataset X with k clusters is sampled from a manifold of finite intrinsic dimension. Each cluster has the same number of points and the underlying distribution for each cluster is from the same distribution class $\{\hat{f}(x) = \frac{1}{\alpha} f(\frac{s(x)}{\alpha}) | \alpha > 0\}$, where f is any probability density function and s is an isometric transformation (reflection, translation, rotation or their combination).

This assumption indicates that the submanifold, from which cluster i is sampled, can be stretched or shrunk via a submanifold from which another cluster j is sampled. That is, there exists a natural bijection $h_{i,j}$ between any two different clusters i and j.

Received 28 August 2024; accepted 2 October 2025