Towards a Persistence Diagram that is Robust to Noise and Varied Densities

Hang Zhang [*1] **Kaifeng Zhang** [*1] **Kai Ming Ting** [1] **Ye Zhu** [2]

Abstract

Recent works have identified that existing methods, which construct persistence diagrams in Topological Data Analysis (TDA), are not robust to noise and varied densities in a point cloud. 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 which possesses these properties. Our empirical evaluation reveals that the proposed filter function provides a better means for t-SNE visualization and SVM classification than three existing methods of TDA.

1. Introduction

Topological Data Analysis (TDA) is an important approach to study the shape of a point cloud using techniques from topology (Wasserman, 2016). 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 successfully discriminates three types of DNA (Meng et al., 2020); and (b) protein-ligand binding affinity 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 (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; Kusano et al., 2017; Hofer et al., 2019). Most of the work (Reininghaus et al., 2015; Adams et al., 2017; Carrière et al., 2017; Kusano et al., 2017; Le & Yamada, 2018; Polanco & Perea, 2019; Bubenik, 2020) have focused on different vectorization methods 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 used only propagates the errors introduced by PD, leading to poor final task-specific performance.

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

The multi-parameter approach (Blumberg & Lesnick, 2022) attempts to address both issues. But it still suffers from the same flaw of the one-parameter approach (see Appendix B.6 for details). No attempts have been afforded 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.

2. Highlighting the importance of the problem of the lack of robustness of PD when noise and varied densities coexist in a point cloud.

3. Proposing a new data-dependent kernel and a new method to produce a PD in order to address the above problem.

[*]Equal contribution [1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China [2]Centre for Cyber Resilience and Trust, Deakin University, Burwood, VIC, Australia. Correspondence to: Kai Ming Ting <tingkm@nju.edu.cn>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

4. Showing that using a good PD significantly improves the visualization outcome of t-Distributed Stochastic Neighbor Embedding (Van der Maaten & Hinton, 2008) and the classification accuracy of Support Vector Machines (Hearst et al., 1998).

2. Background

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

Table 1. Key symbols and notations used

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

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 $\mathcal{X}_\epsilon^f = \{x \in \mathcal{X} \mid f(x) \leq \epsilon\}$ encodes the topological information in $\mathcal{X}$. For $s \leq \epsilon$, the sublevel sets are nested, i.e., $\mathcal{X}_s^f \subseteq \mathcal{X}_\epsilon^f$. Thus $\{\mathcal{X}_\epsilon^f\}_{0 \leq \epsilon < \infty}$ 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 $g \in \mathcal{R}$ 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 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 $VR_\epsilon(X) = \{\sigma \subset X \mid dia(\sigma) \leq \epsilon\}$, 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 $\ell(x, y) \leq \epsilon$.

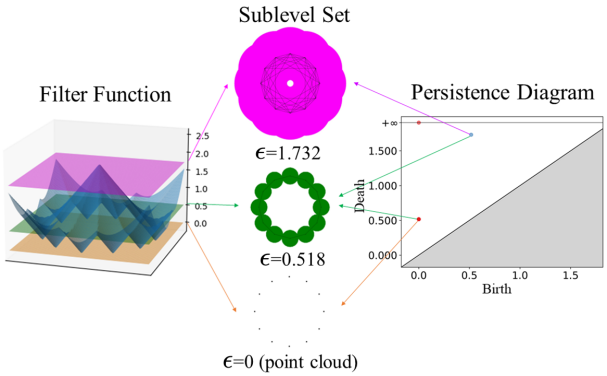


Figure 1. An example filter function, three sublevel sets and a PD.

PH tracks the evolution of topology in $\{VR_\epsilon(X)\}_{0 \leq \epsilon < \infty}$. 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.

(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 (Lesnick & Wright, 2015; Chazal et al., 2017). Furthermore, both of them have trouble with varied densities (Berry & Sauer, 2019; Blumberg & 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 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 et al., 2008; Chazal et al., 2011; 2013). The multi-parameter filtration constructs a bi-filtration by using additional parameters, e.g., using density to capture the topological features of varied densities (Frosini & Mulazzani, 1999; Carlsson & Zomorodian, 2007; Blumberg & Lesnick, 2022).

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 & 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 4.1. *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 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 that has been either stretched or shrunk, 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 or 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 approaches are robust to both noise and varied densities, as far as we know.

To satisfy the requirements in Proposition 4.1, a filter function must have the following properties.

Property 4.2. ***Tolerant to noise.***
The filter function must produce 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 7.1).

Let a finite point cloud S sampled from $\mathcal{S}$ be obtained from another finite point cloud T sampled from $\mathcal{T}$, i.e., $\mathcal{S} = \{m(\mathcal{T}) + r(x - m(\mathcal{T}))|x \in \mathcal{T}\}$, where $r > 1$ (stretching) or $r < 1$ (shrinking), and $m(\cdot)$ denotes the center. This definition of shrinking/shrinking is applicable globally or locally in a point cloud.

Property 4.3. ***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 $\mathcal{T}$, $\forall x \in \mathcal{S}, y \in \mathcal{T}, f(x|S) \approx f(y|T)$.

5. Λ-kernel: Definition and Properties

Motivated by Properties 4.2 and 4.3, we propose a new data-dependent kernel and its associated new filter function which produces PDs that conform to Proposition 4.1.

Voronoi Diagram has been widely used in the 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 & Klein, 2000; Okabe et al., 2000).

Here we propose a new kernel called Λ-kernel as a similarity measure, which is based on Voronoi Diagram.

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 randomly 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:

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

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

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

where $\Upsilon_\ell = (\sum_{j=1}^{\psi} e^{-2\eta \ell(x, v_j)})^{\frac{1}{2}}$ is the normalization term which ensures $\|\Phi(x|\Lambda)\|_2 = 1$, and η is the hyperparameter controlling the relative importance of the entries of $\Phi(x|\Lambda)$.

$\Phi(\cdot|\Lambda)$-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 2(b) shows an example of S^1 ($\psi = 2$), all mapped

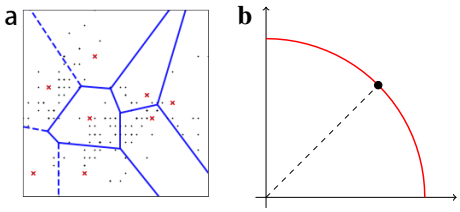


Figure 2. (a) Example Voronoi Diagram where the ψ red points in $\mathbb{R}^d$ are sampled from a point cloud. (b) Data distribution after Λ-mapping $\Phi(\cdot \mid \Lambda)$ when $\psi = 2$.

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 derived from X and $\mathbf{e}_x \triangleq \lim_{\eta\to\infty} \Phi(x|\Lambda)$.

$\Phi(x|\Lambda)$ has the following property:

Property 5.1. *$\forall \epsilon > 0$, $\exists \hat{\eta} > 0, \forall \eta \geq \hat{\eta}$, $x \in X$, $\Lambda \in V(X)$, $\|\Phi(x \mid \Lambda) - \mathbf{e}_x\| \leq \epsilon$, and $\mathbf{e}_x$ is in one-hot form.*

Property 5.1 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 $\mathcal{D}$. See the proof in Appendix A.1.

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

Definition 5.2. For any $x, y \in \mathbb{R}^d$, the similarity of x and y based on Voronoi Diagram Λ is defined as:

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

This similarity has the following property:

Property 5.3. *$\exists \tilde{\eta}, \forall \eta \geq \tilde{\eta}, \forall x, y, x', y' \in \mathbb{R}^d$, if x, y belong to the same Voronoi cell; and x', y' belong to different Voronoi cells, then $s(x, y|\Lambda) \geq s(x', y'|\Lambda)$.*

See the proof in Appendix A.2.

Given a point cloud X, we denote the probability space of Voronoi Diagram as $(\Omega, V(X), \rho)$, where Ω is the sample space of $V(X)$ and $\rho(\cdot)$ is the probability density function. From Definition 5.2, Λ-kernel of x and y wrt X, a similarity based on all possible Voronoi Diagrams derived from X, is defined as follows:

Definition 5.4. $\forall x, y \in \mathbb{R}^d$, Λ-kernel derived from X is defined as:

$$\kappa(x, y|X) = \mathbb{E}_\Lambda[s(x, y|\Lambda)] = \int_{V(X)} \rho(\Lambda) s(x, y|\Lambda) d\Lambda.$$

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

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

Lemma 5.5. *$\hat{\kappa}(x, y \mid X)$ is a valid kernel.*

Proof. See the proof in Appendix A.3 □

The empirical feature map $\hat{\Phi}(x)$ of kernel $\hat{\kappa}$ is expressed as $\hat{\Phi}(x) = \frac{1}{\sqrt{t}}[\Phi_1(x)^\top, \Phi_2(x)^\top, ..., \Phi_t(x)^\top]^\top$, where $\Phi_i(x) \triangleq \Phi(x|\Lambda_i)$. $\hat{\Phi}(\cdot)$ has Property 5.6, which is a direct result from Property 5.1.

Property 5.6. *$\forall \epsilon > 0, \exists \hat{\eta} > 0, \forall \eta > \hat{\eta}, x \in X$, $\|\hat{\Phi}(x) - \hat{\mathbf{e}}_x\| \leq \epsilon$, where $\hat{\mathbf{e}}_x \triangleq \lim_{\eta\to\infty} \hat{\Phi}(x)$.*

Here we propose distance d_Λ which satisfies the following Lemma and has data-dependent Property 5.8, whose proof is included in Appendix A.7. The experiment that demonstrates this property is given in Appendix B.2.

Lemma 5.7. *$d_\Lambda(x, y|X) \triangleq 1 - \hat{\kappa}(x, y|X)$ is a distance metric when $\eta \to \infty$.*

Proof. See the proof in Appendix A.4 □

Property 5.8. *For any two points x, y in dense region $\mathcal{S}$ and their corresponding x', y' in sparse region $\mathcal{T}$, $d_\Lambda(x, y|S) \approx d_\Lambda(x', y'|T)$ or equivalently $\hat{\kappa}(x, y|S) \approx \hat{\kappa}(x', y'|T)$), 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 $r > 1$.*

Proof. See the proof in Appendix A.5, and an example is shown in Appendix B.2. □

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 wrt f_Λ, but also stable wrt the perturbation of input point cloud X.

The new Λ-filter, is given as follows:

$$f_\Lambda(x) = -\max_{y \in X} \int_{V(X)} \rho(\Lambda) s(x, y \mid \Lambda) d\Lambda$$

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

$$\hat{f}_\Lambda(x) = 4 \min_{y \in X} (1 - \hat{\kappa}(x, y|X)). \tag{2}$$

Let $B(x,\epsilon) \triangleq \{y \in \mathbb{R}^d | \ell(x,y) \leq \epsilon\}$, and $\mathcal{X}_\epsilon^{\hat{f}_\Lambda} \triangleq \{x \in \mathcal{X} \,|\, \hat{f}_\Lambda(x) \leq \epsilon\}$. 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 :

$$\{(x,y) \in X^2 \mid \left\|\hat{\Phi}(x) - \hat{\Phi}(y)\right\|_2 \leq \sqrt{2\epsilon}\}, \tag{3}$$

which is equivalent to $\{(x,y) \in X^2 \mid d_\Lambda(x,y|X) \leq \epsilon\}$, where $d_\Lambda(\cdot,\cdot|X)$ is defined in Lemma 5.7 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 wrt bottleneck distance (Chazal & Michel, 2021). It is a Wasserstein distance, defined as:

$$W_\infty(X,Y) := \inf_{\varphi: X \to Y} \sup_{x \in X} \|x - \varphi(x)\|_\infty, \tag{4}$$

where the infimum is over all bijections between X and Y.

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 6.1. *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:*

$$W_\infty(\mathfrak{D}(f), \mathfrak{D}(g)) \leq \|f - g\|_\infty \leq d_H(X, X'), \tag{5}$$

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

Proof. See the proof in Appendix A.6 □

7. PDs Built From Λ-filter Are Robust to Noise and Varied Densities

7.1. Robust to Noise

We start by providing the definition of noise and the definition of robustness to noise.

[1]The code of Λ-filter is available at https://github.com/IsolationKernel/Codes/tree/main/Lambda-kernel

Definition 7.1. Let $\rho(\cdot)$ and $\rho_n(\cdot)$ be the probability density function of $\mathcal{X}$ and $\mathcal{X}_n$, where $\mathcal{X}$ is the sample space of normal points, and $\mathcal{X}_n$ is the sample space of noise points. For any $x \in \mathbb{R}^d$, $\rho(x) \cdot \rho_n(x) \leq \mu$, where μ is a sufficiently small positive real number.

Intuitively, we cannot distinguish between normal points and points in high-density regions, and only points in low-density regions are considered as noise, where the density is denoted by $\rho(x)$.

Definition 7.2. For two given point clouds X and X_n, sampled from $\mathcal{X}$ and $\mathcal{X}_n$, respectively, (δ, X, X_n)-$\mathfrak{D}$ is a PD which is robust to noise-corrupted point cloud $X \cup X_n$, and it satisfies the following conditions:

(a) $W_\infty(\mathfrak{D}(X), \mathfrak{D}(X \cup X_n)) < \delta$, and

(b) the numbers of valid persistent features remain the same in $\mathfrak{D}(X)$ and $\mathfrak{D}(X \cup X_n)$,

DTM's filter function is a robust approximation of that of Vietoris-Rips (Chazal 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.

$\hat{f}_{DTM}(x)$ 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 4.2.

Our proposed Λ-filter[1] 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 4.2.

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 7.2 on the Cassini dataset (Chazal 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 7.1.

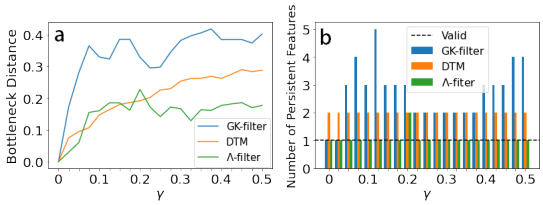


Figure 3. (a) Bottleneck distance between $\mathfrak{D}(X)$ and $\mathfrak{D}(X \cup X_n)$ at different values of γ. (b) Number of valid 1-dim features in $\mathfrak{D}(X \cup X_n)$ at different values of γ. Here we treat a feature (c, g) as a persistent feature if its lifespan $g - c$ is greater than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

Figure 3 shows that Λ-filter outperforms DTM and GK-filter[2] from the following aspects (the details of the point cloud and the PDs produced are provided in Table 4 in Appendix B.1):

(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 (see Table 4), 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.

7.2. Robust to 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 7.3. A PD produced by a filter function 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., $W_\infty(\mathfrak{D}(f(S)), \mathfrak{D}(f(T))) \approx 0$.

CkNN (Berry & 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.

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

Theorem 7.4. *The PD produced via Λ-filter is robust to varied densities.*

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-filer remains unchanged. The proof is provided in Appendix A.7.

Figure 4(a) and Figure 5 show an example of the influence of stretching and shrinking on PD. X_r is stretched from X with ratio r: $X_r = \{m(X) + r(x - m(X)) | x \in X\}$, where $m(X)$ denotes the center of X. 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 4(a).

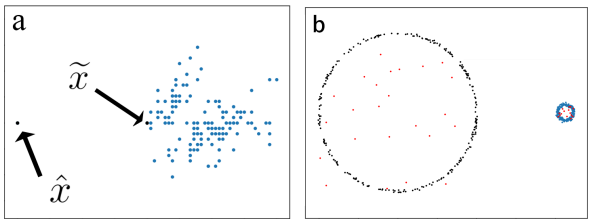


Figure 4. (a) An example of $X_r \cup \{\hat{x}\}$ when $r = 1$. (b) Point cloud with noise and topological features of different density. (The radius of the left ring is 10 times larger than that of the right one.)

The bottleneck distance between the PD of X and the PD of X_r is shown in Figure 5, 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 5(b).

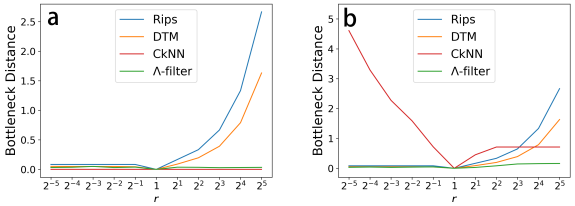


Figure 5. Comparisons in varied densities. (a) Without noise: The bottleneck distance between $\mathfrak{D}(X)$ and $\mathfrak{D}(X_r)$ at different values of r. (b) With one noise point $\hat{x}$: The bottleneck distance between $\mathfrak{D}(X \cup \{\hat{x}\})$ and $\mathfrak{D}(X_r \cup \{\hat{x}\})$ at different values of r.

7.3. Noise and Varied Densities Coexist

The comparison of the four approaches (Rips, DTM, CkNN and Λ-filter) is summarized in Table 2.

Table 2. 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[3]	$\min_{y\in X}\frac{\ell(x,y)}{2\ell_k(y)}$	×	✓
Λ-filter	$4\min_{y\in X}(1-\kappa(x,y))$	✓	✓

A simple example is used to show the difference between these four approaches.[4] The point cloud contains two rings with radius 5 (Figure 6(a)), whose filter function (Rips) was shown in Figure 6(b). 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 4(b)).

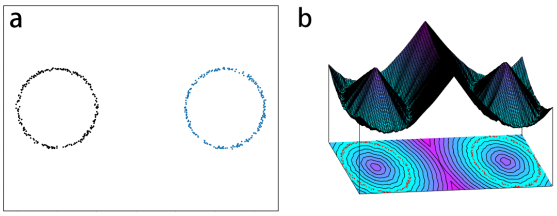


Figure 6. (a) Data with no noise or varied densities. (b) The corresponding filter function (Rips) values and its contour map.

Table 3 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 by the noise.

The details of the growth of sublevel sets are given in Appendix B.3.

Although DTM is more robust to noise than Rips, DTM is unable to provide a completely noise-free PD. The masking effect still lingers for DTM.

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

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. Experiments with Real-world Datasets

We illustrate the performance of robust PDs in machine learning applications through two tasks: 1. dimensionality reduction and visualization of biological cells. 2. classification of bone scripts. In addition to these two tasks, we also explore the effect of parameter η on the classification task. The parameter settings are provided in Appendix B.4.

8.1. 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. Finally, we feed $\Delta_{max} = max\{\Delta_0, \Delta_1\}$ to t-SNE (Van der Maaten & Hinton, 2008) and get the visualization result, which is shown in Figure 8.

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 7(b), as conducted by (Vipond et al., 2021).

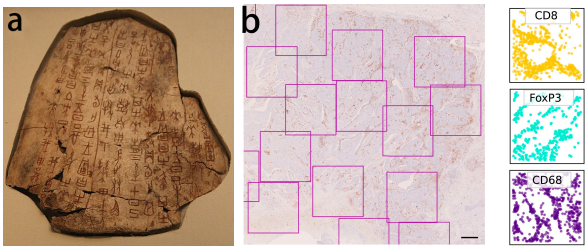


Figure 7. (a) An example of bone scripts[5]. (b) A region of a head and neck tumor IHC slide stained to show CD8 cells. The regions of interest are highlighted.[6]

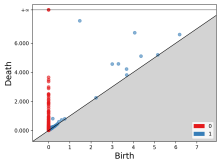
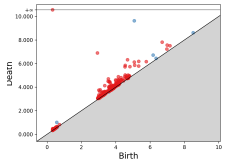
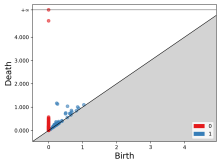
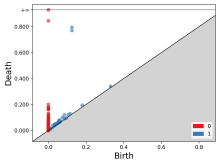
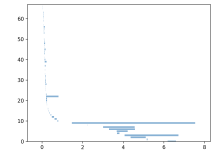
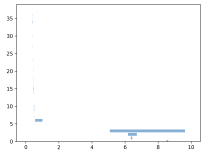
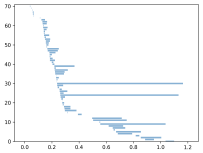
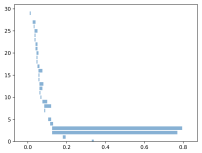
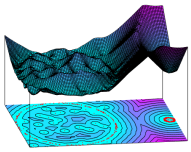
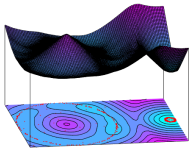
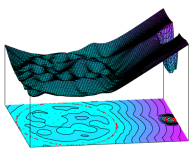
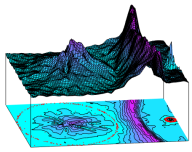
[3]The original paper (Berry & Sauer, 2019) of CkNN gives the distance definition as $\frac{l(x,y)}{\sqrt{l_k(x)l_k(y)}}$ only. Here we provide an approximated filter function of CkNN (see Appendix A.8).

[4]The comparison with the multi-parameter approach is shown in Appendix B.6.

[5]Sourced from https://en.wikipedia.org/wiki/Oracle_bone_script.

[6]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).

Table 3. A comparison result based on the point cloud shown in Figure 4(b).

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

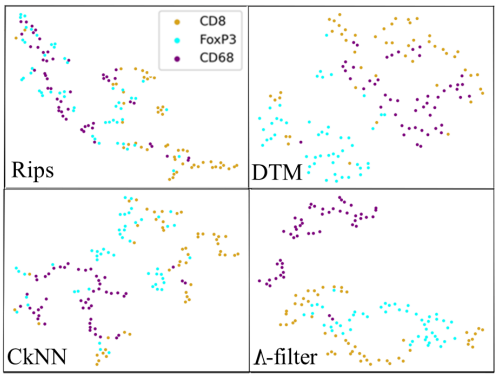


Figure 8. t-SNE visualization results.

The results 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.

8.2. 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[7], as shown in Figure 7(a). Details of this experiment are provided in Appendix B.5.

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.

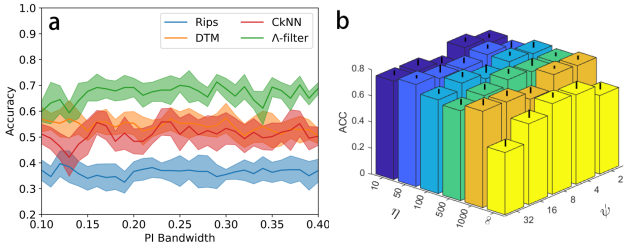


Figure 9. (a) Accuracy of SVM classification of bone scripts. (b) Accuracy at different values of η as ψ increases.

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 9(a).

In addition, we show the accuracy of SVM classification employing Λ-filter at different values of η and ψ in Figure 9(b). When $\eta \neq \infty$, the accuracy is significantly higher and

[7]The dataset is available at `http://jgw.aynu.edu.cn/`.

much more stable respective to ψ than that when $\eta = \infty$.

9. Discussion

9.1. Valid Short-persistence Features

Note that we do not claim that all short-persistence features are noise. There are short-persistence features which are important for some intended task, for example in protein structures (Xia & Wei, 2014). However, these short-persistent features can be easily mistaken as noise if the measure is sensitive to noise. The Λ-filter we proposed aims to suppress the persistence of noise and preserve the persistence of real features.

9.2. The need of data-dependent measure in topology

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.

Note that Λ-filter only works in a dataset with simple stretching/shrinking. Finding a filter function for more complicated continuous deformation remains an open problem.

10. 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 noise or varied densities issue only.

The key to addressing the varied densities issue is a data-dependent kernel that adapts to local distribution with Property 5.8; and the key to addressing the noise issue is a noise-tolerance kernel. The proposed Λ-kernel possesses both properties. Existing methods use a distance measure which does not have both properties.

We have verified the superiority of the proposed Λ-filter over three existing methods in t-SNE visualization and SVM classification tasks.

Acknowledgements

This project is supported by National Natural Science Foundation of China (Grant No. 62076120).

References

Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F., and Ziegelmeier, L. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18, 2017.

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

Aurenhammer, F. and Klein, R. Voronoi diagrams. *Handbook of Computational Geometry*, 5(10):201–290, 2000.

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

Blumberg, A. J. and Lesnick, M. Stability of 2-parameter persistent homology. *Foundations of Computational Mathematics*, pp. 1–43, 2022.

Bubenik, P. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16:77–102, 01 2015.

Bubenik, P. The persistence landscape and some of its properties. *Topological Data Analysis*, 2020.

Carlsson, G. and Zomorodian, A. The theory of multidimensional persistence. In *Proceedings of Annual Symposium on Computational Geometry*, pp. 184–193. Association for Computing Machinery, 2007.

Carlsson, G., Zomorodian, A., Collins, A., and Guibas, L. Persistence barcodes for shapes. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pp. 124–135. Association for Computing Machinery, 2004.

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

Carrière, M., Cuturi, M., and Oudot, S. Sliced wasserstein kernel for persistence diagrams. In *Proceedings of International Conference on Machine Learning*, volume 70, pp. 664–673, 2017.

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

Chazal, F. and Michel, B. An introduction to topological data analysis: Fundamental and practical aspects for data scientists. *Frontiers in Artificial Intelligence*, 4:667963, 2021.

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

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

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

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

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

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

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

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

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

Hofer, C. D., Kwitt, R., and Niethammer, M. Learning representations of persistence barcodes. *Journal of Machine Learning Research*, 20(126):1–45, 2019.

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

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

Lesnick, M. and Wright, M. Interactive visualization of 2-d persistence modules. *arXiv preprint arXiv:1512.00180*, 2015.

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

Mead, A. W. Review of the development of multidimensional scaling methods. *The Statistician*, 41:27–39, 1992.

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

Milnor, J. On the betti numbers of real varieties. *Proceedings of American Mathematical Society*, 15(2):275–280, 1964.

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

Okabe, A., Boots, B., Sugihara, K., and Chiu, S. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley Series in Probability and Statistics. Wiley, 2000.

Polanco, L. and Perea, J. A. Adaptive template systems: Data-driven feature selection for learning with persistence diagrams. In *IEEE International Conference On Machine Learning And Applications*, pp. 1115–1121, 2019.

Prasolov, V. *Intuitive Topology*. Mathematical world. Universities Press (India) Pvt. Limited, 1998.

Reem, D. The geometric stability of voronoi diagrams with respect to small changes of the sites. In *Proceedings of Annual Symposium on Computational Geometry*, pp. 254–263, 2011.

Reininghaus, J., Huber, S., Bauer, U., and Kwitt, R. A stable multi-scale kernel for topological machine learning. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4741–4748, 2015.

Townsend, J., Micucci, C. P., Hymel, J. H., Maroulas, V., and Vogiatzis, K. D. Representation of molecular structures with persistent homology for machine learning applications in chemistry. *Nature Communications*, 11(1): 1–9, 2020.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

Vipond, O. Multiparameter persistence landscapes. *Journal of Machine Learning Research*, 21(61):1–38, 2020.

Vipond, O., Bull, J. A., Macklin, P. S., Tillmann, U., Pugh, C. W., Byrne, H. M., and Harrington, H. A. Multiparameter persistent homology landscapes identify immune cell spatial patterns in tumors. *Proceedings of National Academy of Sciences*, 118(41):e2102166118, 2021.

Vishwanath, S., Fukumizu, K., Kuriki, S., and Sriperumbudur, B. K. Robust persistence diagrams using reproducing kernels. *Advances in Neural Information Processing Systems*, 33:21900–21911, 2020.

Wasserman, L. Topological data analysis. *arXiv preprint arXiv:1609.08227*, 2016.

Xia, K. and Wei, G.-W. Persistent homology analysis of protein structure, flexibility, and folding. *International Journal for Numerical Methods in Biomedical Engineering*, 30(8):814–844, 2014.

Zomorodian, A. and Carlsson, G. Computing persistent homology. In *Proceedings of Annual Symposium on Computational Geometry*, pp. 347–356, 2004.

A. Proofs

A.1. Proof of Property 5.1

Property 5.1 $\forall\, \epsilon \in (0,1), \exists\, \hat{\eta} > 0, \forall\, \eta \geq \hat{\eta},\ x \in X,\ \Lambda \in V(X), \|\Phi(x \mid \Lambda) - \mathbf{e}_x\| \leq \epsilon$, where $\mathbf{e}_x \triangleq \lim\limits_{\eta\to\infty} \Phi(x|\Lambda)$ and it is in one-hot form.

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 :

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

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 $\hat{i} \triangleq \arg\min\limits_{i} \ell(x, v_i)$.

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 $q_{x,\Lambda} = \arg\min\limits_{i\neq\hat{i}}(\ell(x,v_i) - \ell(x,v_{\hat{i}}))$, then

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

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

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

□

A.2. Proof of Property 5.3

Property 5.3 $\exists\, \tilde{\eta}, \forall\, \eta \geq \tilde{\eta}, \forall\, x, y, x', y' \in \mathbb{R}^d$, 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 5.1, $\exists\, \tilde{\eta}, \forall \eta \geq \tilde{\eta}, \forall x \in X, \|\Phi(x|\Lambda) - \mathbf{e}_x\|_2 \leq \tilde{\epsilon}$. 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

$$\|\Phi(x|\Lambda) - \Phi(y|\Lambda)\|_2 \leq (\|\Phi(x|\Lambda) - \mathbf{e}_x\|_2 + \|\Phi(y|\Lambda) - \mathbf{e}_y\|_2) \leq 2\tilde{\epsilon} \text{ and}$$

$$\|\Phi(x'|\Lambda) - \Phi(y'|\Lambda)\|_2 \geq (\|\mathbf{e}_{x'} - \mathbf{e}_{y'}\|_2 - 2\tilde{\epsilon}) \geq \sqrt{2} - 2\tilde{\epsilon}.$$

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 5.5

Lemma 5.5 $\hat{\kappa}(x, y \mid X)$ 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 $\forall\, \Lambda \in V(X),\ s(x, y|\Lambda) = s(y, x|\Lambda)$.

Equation 1 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 5.7

Lemma 5.7 $d_\Lambda(x, y|X) \triangleq 1 - \hat{\kappa}(x, y|X)$ 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 $\lim_{\eta\to\infty} s(x, y|\Lambda_i) = 0$, if x, y are in the same Voronoi cell of Λ_i. Otherwise, $\lim_{\eta\to\infty} s(x, y|\Lambda_i) = 1$. 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 wrt 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 $d'(x, z|\Lambda) \leq d'(x, y|\Lambda) + d'(y, z|\Lambda)$ 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 $d'(x, z|\Lambda) \leq d'(x, y|\Lambda) + d'(y, z|\Lambda)$ 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 $d'(x, z|\Lambda) \leq d'(x, y|\Lambda) + d'(y, z|\Lambda)$ 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 $d'(x, z|\Lambda) \leq d'(x, y|\Lambda) + d'(y, z|\Lambda)$ holds.

□

A.5. Proof of Property 5.8

Property 5.8 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: $y \leq_x z \iff ||x - y|| \leq ||x - z||$.

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:

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

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

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:

$$\begin{aligned}\kappa(x', y'|T) &= \mathbb{E}_{\Lambda'}[s(x', y'|\Lambda')] = \int_{V(T)} \rho(\Lambda') s(x', y'|\Lambda') d\Lambda' = \int_{V(T)} \rho(\Lambda') \langle \Phi(x'|\Lambda'), \Phi(y'|\Lambda') \rangle d\Lambda' \\ &= \int_{V(S)} \rho(\Lambda) \langle \Phi(x|\Lambda), \Phi(y|\Lambda) \rangle d\Lambda = \int_{V(S)} \rho(\Lambda) s(x, y|\Lambda) d\Lambda = \mathbb{E}_\Lambda[s(x, y|\Lambda)] \\ &= \kappa(x, y|S).\end{aligned}$$

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:

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

□

A.6. Proof of Theorem 6.1

Theorem 6.1 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 A.1. 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 A.2. $R_\phi = dom(\mathcal{D}_\phi, \cup_{j \neq \phi} \mathcal{D}_j) = \{x \in X : \ell(x, \mathcal{D}_\phi) \leq \ell(x, \mathcal{D}_j), \forall j \in \Psi \setminus \phi\}$.

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 A.3. *For each $\phi \in \Psi$ let $R_\phi = dom(\mathcal{D}, A_\phi), R'_\phi = dom(\mathcal{D}'_\psi, A'_\phi)$ 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 of the partitions of point x be changed $\delta' * t$ $(0 < \delta' \leq \delta)$ times is:

$$P(d_{kH}(\Phi(X), \Phi(X')) \leq \delta') = \sum_{i=0}^{\lfloor t * \delta' \rfloor} \frac{t!}{(t-i)!(i)!} (\frac{3\delta}{2\eta})^i * (1 - \frac{3\delta}{2\eta})^{t-i}.$$

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.

For $c > 4$, we have

$$P(d_{kH}(\Phi(X), \Phi(X')) \leq \delta/c) = \sum_{i=0}^{\lfloor t * \delta/c \rfloor} \frac{t!}{(t-i)!(i)!} (\frac{3\delta}{2\eta})^i * (1 - \frac{3\delta}{2\eta})^{t-i}$$

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

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

where the d_H is the Hausdorff distance, f and g is the filter function of X and X' respectively.

□

A.7. Proof of Theorem 7.4

Theorem 7.4 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 $\hat{\kappa}(x, y|S) \approx \hat{\kappa}(x', y'|T)$, hence

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

□

A.8. 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 A.4, which is used for the proof of Theorem A.5.

Lemma A.4. *$\forall \epsilon > 0$, $\frac{\ell(x_i,x_j)}{\sqrt{(\ell_k(x_i)\ell_k(x_j)}} \leq \epsilon$ 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 $w(i,j) = \frac{1}{\sqrt{\ell_k(x_i)}+\sqrt{\ell_k(x_j)}}$.*

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 $\mu = \frac{\sqrt{\ell_k(x_i)}}{\sqrt{\ell_k(x_j)}+\sqrt{\ell_k(x_j)}}$. Then we get

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

□

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

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

Then we can have an approximation of CkNN's filter function, as shown in Theorem A.5.

Theorem A.5. *$\forall \epsilon > 0, G^f_\epsilon(X) \subset G_\epsilon(X) \subset G^g_\epsilon(X)$, where*

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

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

$f(x) = \min_{i\in[n]} \max_{j\in[n]}(\frac{\ell(x,x_i)}{\sqrt{\ell_k(x_i)}}w(i,j))$, $g(x) = \min_{i\in[n]} \min_{j\in[n]}(\frac{\ell(x,x_i)}{\sqrt{\ell_k(x_i)}}w(i,j))$, $\mathcal{X}^f_\epsilon = \{x \in \mathcal{X} | f(x) \leq \epsilon\} = \cup_{i=1}^n B(x_i, r^f_i(\epsilon))$, $\mathcal{X}^g_\epsilon = \{x \in \mathcal{X} | g(x) \leq \epsilon\} = \cup_{i=1}^n B(x_i, r^g_i(\epsilon))$, $r^f_i(\epsilon) = \frac{\sqrt{\ell_k(x_i)}}{max_{p\in[n]} w(i,p)}\epsilon$ and $r^g_i(\epsilon) = \frac{\sqrt{\ell_k(x_i)}}{min_{p\in[n]} w(i,p)}\epsilon$. B is defined in Section 6, and $w(\cdot,\cdot)$ is defined in lemma A.4.

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^f_i to each x_i under given $r^f_i(\epsilon) = \frac{\sqrt{\ell_k(x_i)}}{max_{p\in[n]} w(i,p)}\epsilon$.

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

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

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

So according to Lemma A.4, we can get $\frac{\ell(x_i,x_j)}{\sqrt{\ell_k(x_i)\ell_k(x_j)}} \leq \epsilon$, 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 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:

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

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

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

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

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. Then we have

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

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

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

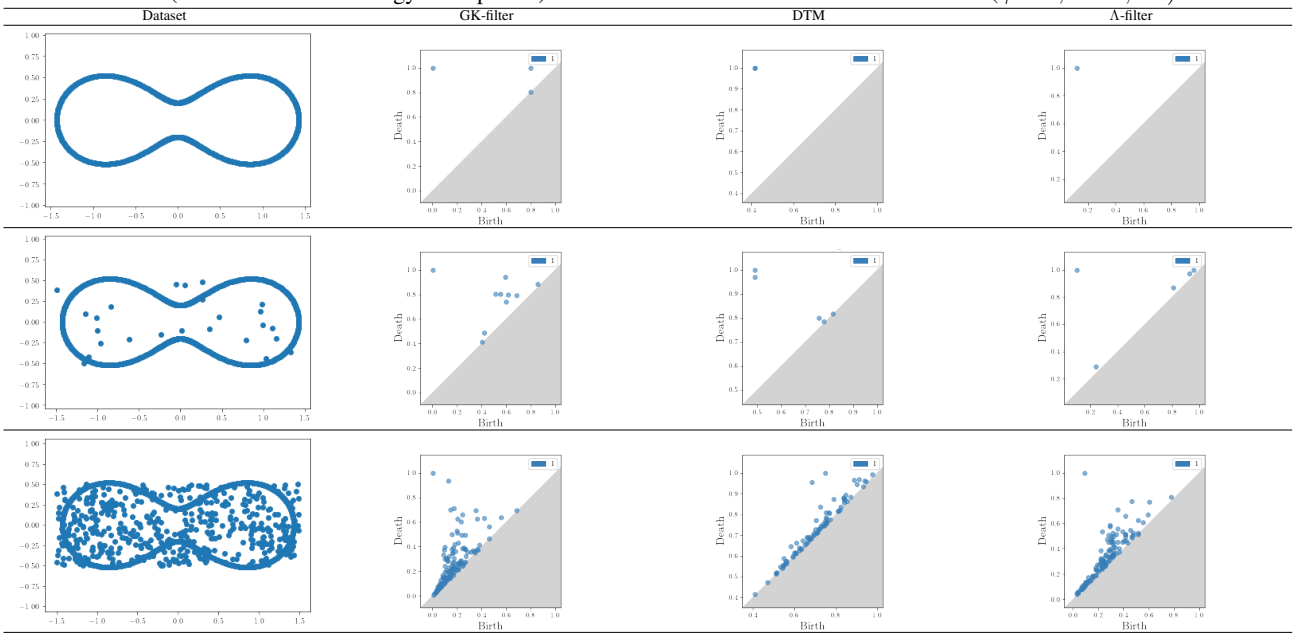
B. Additional Experiments

B.1. Robustness to noise

This section provides the details of Figure 3 described in Section 7.1. It presents the advantages of Λ-filter over other methods in terms of robustness to noise.

Table 4 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.

Table 4. The PDs (1-dimensional homology are reported) of the data with noise of three different ratios ($\gamma = 0, 0.025, 0.5$).



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. Experiment that demonstrates d_Λ has Property 5.8

This section provides an example to show that d_Λ has the ability of preserve the distance after stretching or shrinking (Property 5.8 in Section 5).

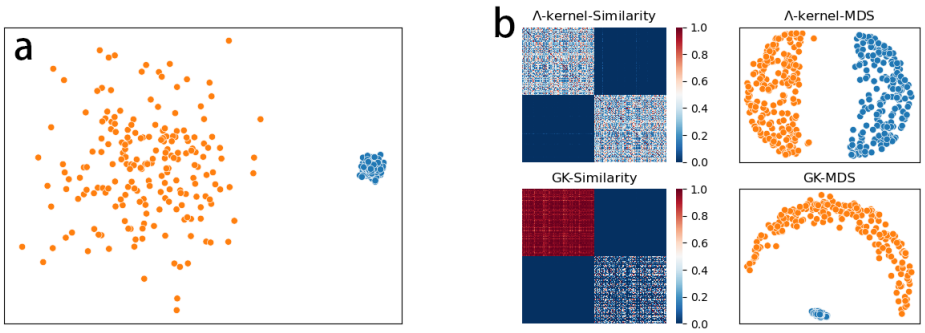


Figure 10. (a) Original data with sparse and dense clusters. (b) Similarity matrices and MDS visualizations of Λ-kernel and GK.

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 10, where the dense cluster T is obtained by a tenfold shrinking of sparse cluster S.

In Figure 10, 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 10(b).

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 5.8.

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 4.3 experimentally.

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

B.3. Details of Table 3 : Growth of sublevel sets

This section gives the corresponding growth of sublevel sets of the Persistence Diagrams in Table 3 from Section 7.3. For each filter function, 3 snippets of the growth of sublevel sets are provided in Table 5.

Table 5 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 8.

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 classification of the bone scripts dataset

This section describes the experiment setting of the bone scripts classification from Section 8.2.

We use a dataset which contains ten classes of bone scripts, which are referred to as ten 'heavenly stems' in Chinese culture, as shown in Figure 7(a). 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 9(a), 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 11 in the form of confusion matrices. These results demonstrate the superiority of PDs produced by Λ-filter, where Λ-filter has the highest accuracy.

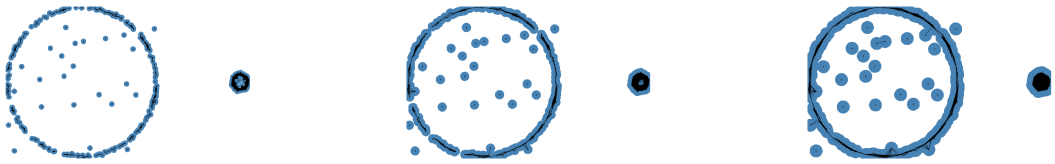
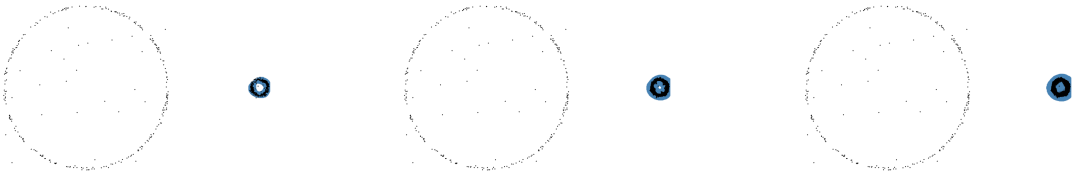
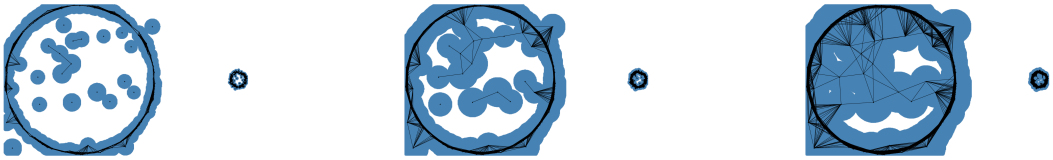
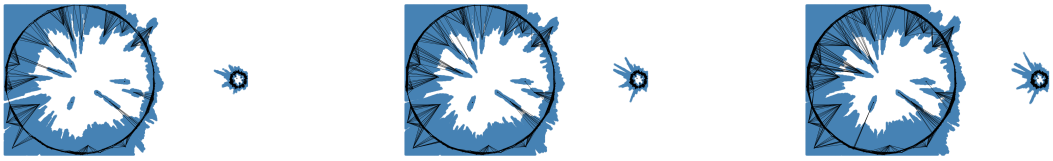
In Figure 9(b), for each (η, ψ) pair, we select the best PI bandwidth to explore the effect of η, ψ in Λ-filter.

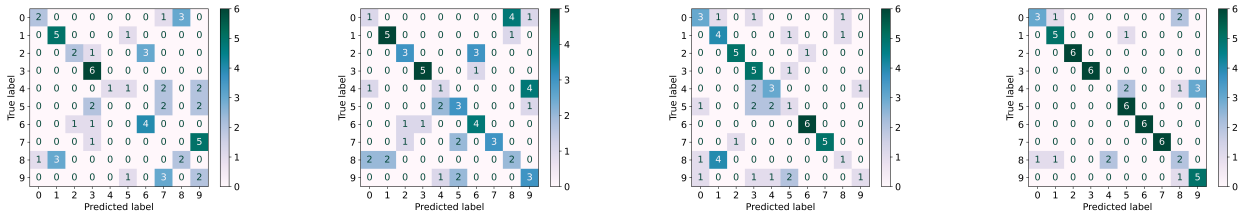
B.6. Multi-parameter Persistent Homology

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

Multi-parameter Persistent Homology (MPH) (Blumberg & 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 sparse and dense

Table 5. 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	$\epsilon = 0.125$	$\epsilon = 0.150$	$\epsilon = 0.175$



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

Figure 11. Confusion matrices

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 4, where there is one dense ring, one sparse ring, and some noise. The result is shown in Figure 12. MPH 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.

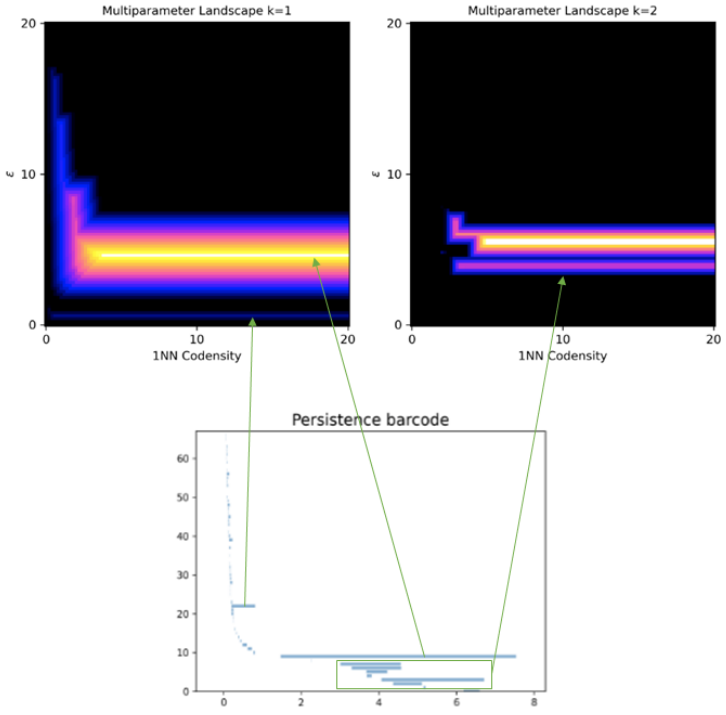


Figure 12. First MPL and second MPL, and PD from the Rips filtration.