Persistent homology and its applications

Yuzhou He

Prepared under the supervision of Prof. Martin Frankland

October 22, 2025

Contents

Persistent homology has become a fundamental tool in topological data analysis, providing a robust way to extract multi-scale topological features from complex data; see for instance [EH10; Oud15; CV22]. In this work, we first review the basic concepts of persistent homology, including the construction of filtered complexes, barcodes and persistence diagrams. We then explore its applications in two distinct settings: time series analysis and knot theory.

In Section 3, following Perea and Harer's work on sliding windows and persistence [PH15], we develop a MATLAB implementation to verify the periodicity of time series data via sliding window embedding and one-dimensional persistence. In Section 4, following the work of Celoria and Mahler [CM22], we apply persistent homology to study geometric and statistical properties of knots.

Our main contribution lies in the practical realization of these methods through MATLAB programs that enable visualization and verification of topological features in both temporal and geometric datasets. All related code and implementation details can be found at:

`https://github.com/wiliiiii/Persistence_Homology`

Acknowledgments

This work was supported by Mitacs as part of a Globalink Research Internship held at the University of Regina.

1 Filtered simplicial complexes

Given a simplicial complex K, a subcomplex of K is a subset of its simplices that is closed under the face relation. A filtration of K is a nested sequence of subcomplexes that starts with an empty complex and ends with a complete complex

$$\emptyset = K_0 \subset K_1 \subset \cdots \subset K_m = K.$$

1.1 Different complexes

Suppose that $X = \{p_1, \cdots, p_n\}$ is a finite point cloud in the Euclidean space $\mathbb{R}^d$.

Definition 1.1.1. The **Čech complex** of X is the filtered simplicial complex

$$C(X) = \{C_r(X) : r > 0\}$$

where

$$C_r(X) = \{\sigma \subset X : \bigcap_{x \in \sigma} B(x, r) \neq \emptyset\}.$$

Definition 1.1.2. The **Vietoris–Rips** complex of X is the filtered simplicial complex

$$VR(X) = \{VR_r(X) : r > 0\}$$

where

$$VR_r(X) = \{\sigma \subset X : \text{diam}(\sigma) \leq r\}.$$

If $r = 0$, $VR_r(X)$ is discrete and 0-dimensional, that is, only has 0-simplices. If $r >> 0$, $VR_r(X)$ is the full simplex.

Before introducing the alpha complex (subcomplexes of the Delaunay complex), we begin with some necessary background. More details can be found in these notes by Edelsbrunner: `https://courses.cs.duke.edu/fall06/cps296.1/Lectures/sec-III-4.pdf`

Denote

$$\text{Union}(r) = \{x \in \mathbb{R}^d : \exists p \in X, ||x - p|| \leq r\}.$$

For each $p \in X$, we can create its Voronoi cell:

$$V_p = \{x \in \mathbb{R}^d : ||x - p|| \leq ||x - q||, \forall q \in X\}.$$

We can see that the Voronoi cell is convex. Let $H_{i,j}$ be the half-space:

$$H_{i,j} = \{x \in \mathbb{R}^d : ||x - p_i|| \leq ||x - p_j||\}.$$

Clearly, $V_{p_i} = \bigcap_{j \neq i} H_{i,j}$. Since each $H_{i,j}$ is closed and convex, V_p is also closed and convex. To decompose the union, we intersect each ball with the corresponding Voronoi cell, $R_p(r) = B_p(r) \cap V_p$. Since balls and Voronoi cells are convex, $R_p(r)$ is also convex. Any two of them are disjoint or overlap along a common piece of their boundaries. If $R_{p_1}(r) \cap R_{p_2} \neq \emptyset$, then

$$\exists x \in R_{p_1}(r) = B_{p_1}(r) \cap V_{p_1}, x \in R_{p_2}(r) = B_{p_2}(r) \cap V_{p_2}$$

so $x \in V_{p_1} \cap V_{p_2}$. From the definition of the Voronoi cell, x can only fall on the boundary. Together the $R_p(r)$ cover the entire union, as in the Figure 1.

Definition 1.1.3. The **alpha complex** of X is

$$\text{Alpha}(X) = \{\text{Alpha}_r(X) : r > 0\}$$

where

$$\text{Alpha}_r(X) = \{\sigma \subset X : \bigcap_{p \in \sigma} R_p(r) \neq \emptyset\}.$$

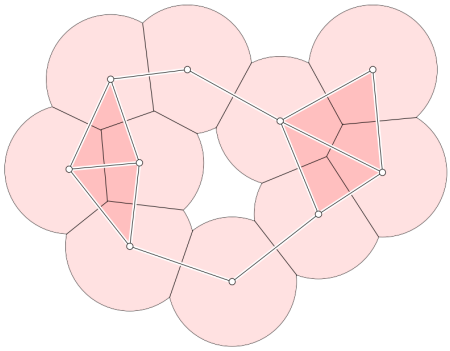


Figure 1: The union of disks is decomposed into convex regions by the Voronoi cells. The corresponding alpha complex is superimposed. Image source: Herbert Edelsbrunner.

Next, we introduce the relationships among these three different simplicial complexes.

Proposition 1.1.4. *Suppose that $r > 0$, then*

$$VR_r(X) \subset C_r(X) \subset VR_{2r}(X).$$

Since $R_p(r) = V_p(r) \cap B_p(r)$, we have the following property.

Proposition 1.1.5. *Suppose that $r > 0$, then* $\text{Alpha}_r(X) \subset C_r(X)$.

Then we need to use the nerve theorem to prove the homotopy relation.

Definition 1.1.6. Let I be a set of indices and C be a family of sets $\{U_i\}_{i \in I}$. The **nerve** of C is a set of finite subsets of the index set I defined as follows. It contains all finite subsets $J \subset I$ such that the intersection of the U_i whose subindices are in J is non-empty:

$$N(C) := \{J \subset I : \bigcap_{j \in J} U_j \neq \emptyset, J \text{ finite set}\}.$$

Theorem 1.1.7. *If any intersection of sets in $N(C)$ is contractible, then $N(C)$ is homotopy equivalent to $\bigcup C$.*

Since each ball is convex, hence there finite intersection is convex and contractible. Applying the nerve theorem to the Čech complex, we obtain the following proposition.

Proposition 1.1.8. *Suppose that $r > 0$, $C_r(X) \simeq \text{Union}(r)$.*

Similarly, the intersection of finite $R_p(r)$ is also contractible. Apply the nerve theorem to the alpha complex, and notice that homotopy equivalence is an equivalence relation.

Proposition 1.1.9. *Suppose that $r > 0$, $C_r(X) \simeq \text{Alpha}_r(X)$.*

Since homology is a homotopy invariant, we can obtain the homology of the Čech complex by calculating the homology of the alpha complex.

1.2 Persistent homology

We next define persistence, persistent homology, and the persistence diagram for a simplicial complex K. We have obtained a filtered complex $VR(X) = \{VR_r\}_{r\geq 0}$, which can be regarded as a functor:

$$VR : \mathbb{R}_{\geq 0} \to \mathbf{Top}(\mathbf{Simp}).$$

If we apply $H_i(-; k)$, where k is a field, then:

$$\mathbb{R}_{\geq 0} \to \mathbf{Top} \to \mathbf{Vect}_k.$$

This also holds for the Čech complex and alpha complex.

Definition 1.2.1. A **persistence module** is a diagram of vector spaces indexed by $\mathbb{N}$ or $\mathbb{R}_{\geq 0}$.

A filtration of a simplicial complex K is:

$$\emptyset = K_0 \subset K_1 \subset \cdots \subset K_m = K.$$

A homology class α is born at K_i if it is not in the image of map induced by the inclusion $K_{i-1} \subset K_i$. If α is born at K_i, then we say that it dies entering K_j if the image of the map induced by $K_{i-1} \subset K_{j-1}$ does not contain the image of α but the image of the map induced by $K_{i-1} \subset K_j$ does. The persistence of *alpha* is $j - i$.

We encode birth and death information in persistence diagrams, one for each dimension. The diagram $\mathbf{dgm}(q)$ has a point (i, j) for every $q-$homology class that is born at K_i and dies entering K_j.

2 Persistence modules

2.1 Structure theorem

Theorem 2.1.1. *Any diagram of finite-dimensional $k-$vector spaces:*

$$V : \mathbb{R}_{r\geq 0} \to \mathbf{Vect}_k$$

is a direct sum of the interval modules:

$$I_{[a,b)}(t) = \begin{cases} k & t \in [a, b) \\ 0 & otherwise \end{cases}, \quad I_{[a,b)}(s \leq t) = \begin{cases} \text{id}_k & s, t \in [a, b) \\ 0 & otherwise. \end{cases}$$

Example 2.1.2. For the vertex set $\{a, b, c\}$, consider the filtered simplicial complex

$$K_0 \subset K_1 \subset K_2 \subset K_3.$$

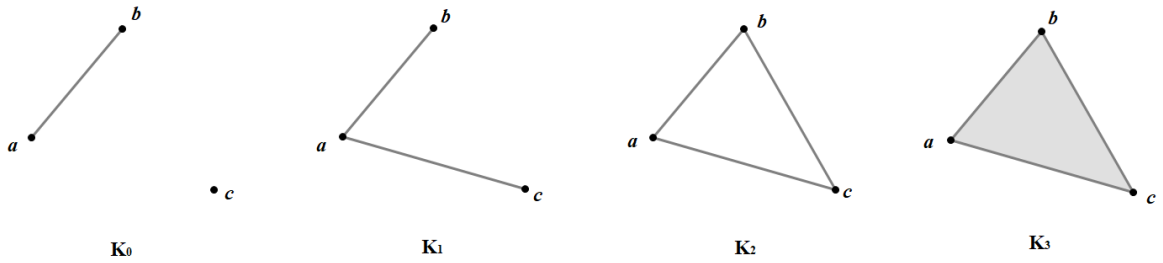


Figure 2: The filtration on a simplicial complex

Definition 2.1.3. A **barcode** is a multiset of intervals. The **barcode of a persistence module** is the list of intervals appearing in the interval decomposition.

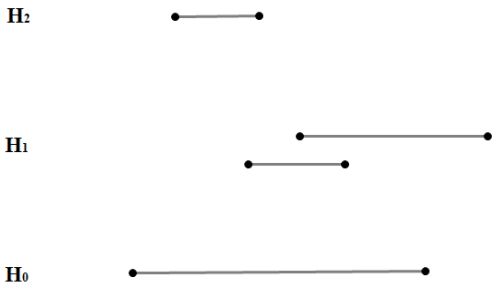


Figure 3: A barcode

2.2 Coefficient field

In this section, we follow [OY23].

When we compute a persistence diagram, we need to select a coefficient field before computation. If all coefficient fields k yield the same persistence diagram, this represents the most ideal case. However, this is not practical, because the dimensions of homology vector spaces for the space topological space are different when $\mathbb{Z}$-homology group of the spaces has non-zero torsion. Let's first consider the example of the Möbius strip.

Example 2.2.1 (The Möbius strip)**.**

Let M be a Möbius strip and ∂M be its boundary. Both $H_1(\partial M; \mathbb{Z})$ and $H_1(M; \mathbb{Z})$ are isomorphic to $\mathbb{Z}$, and the homomorphism

$$H_1(\partial M; \mathbb{Z}) \to H_1(M; \mathbb{Z})$$

is isomorphic to

$$n \mapsto 2n.$$

Consider the following filtration:

$$\mathbb{X} : \emptyset = X_0 \subset X_1 = \partial M \subset X_2 = M$$

the decomposition on $\mathbb{Z}_2$ is

$$H_1(\mathbb{X}; \mathbb{Z}_2) = [1, 2) \oplus [2, \infty).$$

However, the decomposition on $\mathbb{R}$ is

$$H_1(\mathbb{X}; \mathbb{R}) = [1, \infty).$$

In this example, both $H_1(\partial M; \mathbb{Z})$ and $H_1(M; \mathbb{Z})$ are free, but $H_1(M, \partial M; \mathbb{Z}) \cong \mathbb{Z}_2$ and this is not free. We may guess this is the key to the different diagrams.

Independent We use $\mathbb{X}$ to denote a filtration of a complex

$$\mathbb{X} : \emptyset = X_0 \subset X_1 \subset \cdots \subset X_N = X.$$

Use $D_q(\mathbb{X}; k)$ to denote the qth persistence diagram.

Theorem 2.2.2. *The qth persistence diagram $D_q(\mathbb{X}; k)$ is independent of the choice of k if $H_q(X_n, X_m; \mathbb{Z})$ is free for any $0 \leq m < n \leq N$ and $H_{q-1}(X_n; \mathbb{Z})$ is free for any $0 \leq n \leq N$.*

Since $X_0 = \emptyset$, the first assumption includes the freeness of $H_q(X_n; \mathbb{Z}) = H_q(X_n, X_0; \mathbb{Z})$.

Corollary 2.2.3. *$D_0(\mathbb{X}; k)$ is always independent of the choice of k.*

Corollary 2.2.4. *When $\mathbb{X}$ is a filtration of finite cell/simplicial/cubical complexes embedded in $\mathbb{R}^d$, the $(d-1)$th persistent homology gives the same persistence diagram among any fields k.*

The above two corollaries ensure that if a filtration is embedded in $\mathbb{R}^2$, all non-trivial persistence diagrams D_0 and D_1 do not depend on the choice of the coefficient field.

Dependent A natural question is that when $D_q(\mathbb{X}, k)$ changes depending of the choice of k, how does it change?

In the above example about a Möbius strip, a long interval $[1, \infty)$ is split into two shorter intervals, $[1, 2)$ and $[2, \infty)$, when k changes from $\mathbb{R}$ to $\mathbb{Z}_2$. From the example, we expect that a long interval indecomposable tends to be split into shorter intervals when k changes from $\mathbb{R}$ to $\mathbb{Z}_p$.

Theorem 2.2.5. *Let q be a positive integer and we consider an $\mathbb{R}$-parameterized filtration $\mathbb{X} = \{X_t\}_{t \in \mathbb{R}}$. Assume that X_t is finite, that is, for sufficiently large t, X_t is a finite simplicial complex. We also assume that $X_t = \varnothing$ for sufficiently small t and $H_q(X_t) = 0$ for sufficiently large t. Let f be a C^2 convex function on $[0, \infty)$ with $f(0) = 0$. Then the following inequality holds if $H_q(X_t; \mathbb{Z})$ and $H_{q-1}(X_t; \mathbb{Z})$ are free for all t:*

$$\sum_{(b,d) \in D_q(\mathbb{X}; \mathbb{R})} f(d - b) \;\geq \sum_{(b,d) \in D_q(\mathbb{X}; \mathbb{Z}_p)} f(d - b).$$

When f is strictly convex, the equality holds if and only if $D_q(\mathbb{X}; \mathbb{R}) = D_q(\mathbb{X}; \mathbb{Z}_p)$.

Therefore, $D_q(\mathbb{X}; \mathbb{R})$ contains richer information than $D_q(\mathbb{X}; \mathbb{Z}_p)$ under the condition of the theorem. However, this is not true for higher dimensional filtrations in general.

3 Application to time series

In this section, we follow [PH15].

For a discrete variable t, we have a point s_t in a metric space $(\mathbb{R}^2, d)$. We may view the sequence $\{s_i\}_{i=0}^n$ as sampled from a real-valued function f by cubic spline interpolation.

Supposed that $\{s_i = (x_i, y_i)\}_{i=0}^n$ are $n+1$ knots of the spline interpolation. There'll be a cubic polynomial $q_i(x) = y$ between each successive knots (x_{i-1}, y_{i-1}) and (x_i, y_i) connecting to both of them, where $i = 1, 2, \ldots, n$. There will be n polynomials, with the first one starting at (x_0, y_0), and the last one ending at (x_n, y_n).

We already know that the curvature of any curve $y = y(x)$ is

$$\kappa = \frac{y''}{\left(1 + (y')^2\right)^{3/2}}.$$

We will define both y' and y'' to be continuous everywhere, including at the knots, which is to say that

$$\begin{cases} q_i(x_i) = q_{i+1}(x_i) = y_i \\ q_i'(x_i) = q_{i+1}'(x_i) \\ q_i''(x_i) = q_{i+1}''(x_i) \end{cases} \qquad 1 \le i \le n-1.$$

We can require f has degree 3, then these equations can be achieved. Moreover, $f \in C^2$.

Definition 3.0.1. Choose an integer M and a real number τ, both greater than 0. The **sliding-window embedding** of f based at $t \in \mathbb{R}$ into $\mathbb{R}^{M+1}$ is the point

$$SW_{M,\tau} f(t) = \begin{bmatrix} f(t) \\ f(t+\tau) \\ \vdots \\ f(t+M\tau) \end{bmatrix}.$$

Choosing different values of t gives a collection of points called a **sliding-window point cloud** for f. A critical parameter for this embedding is the window size $M\tau$.

It turns out that in general, periodic functions trace out ellipses in $\mathbb{R}^{M+1}$, so it is very meaningful to study its persistent homology.

3.1 Cosine function

Let $n \in \mathbb{N}$ and $f(t) = \cos(nt)$. Then

$$\begin{aligned} SW_{M,\tau} f(t) &= \begin{bmatrix} \cos(nt) \\ \cos(nt + n\tau) \\ \vdots \\ \cos(nt + nM\tau) \end{bmatrix} \\ &= \cos(nt) \begin{bmatrix} 1 \\ \cos(n\tau) \\ \vdots \\ \cos(nM\tau) \end{bmatrix} - \sin(nt) \begin{bmatrix} 0 \\ \sin(n\tau) \\ \vdots \\ \sin(nM\tau) \end{bmatrix} \\ &:= \cos(nt)\mathbf{u}_n - \sin(nt)\mathbf{v}_n. \end{aligned}$$

Similarly, $g(t) = \sin(nt)$. Then

$$\begin{aligned} SW_{M,\tau} g(t) &= \begin{bmatrix} \sin(nt) \\ \sin(nt + n\tau) \\ \vdots \\ \sin(nt + nM\tau) \end{bmatrix} \\ &= \sin(nt) \begin{bmatrix} 1 \\ \cos(n\tau) \\ \vdots \\ \cos(nM\tau) \end{bmatrix} + \cos(nt) \begin{bmatrix} 0 \\ \sin(n\tau) \\ \vdots \\ \sin(nM\tau) \end{bmatrix} \\ &= \sin(nt)\mathbf{u}_n - \cos(nt)\mathbf{v}_n. \end{aligned}$$

Hence the sliding-window embeddings of cosine and sine are linear combination of $\mathbf{u}$ and $\mathbf{v}$, satisfying:

$$\begin{aligned} \langle \mathbf{u}_n, \mathbf{v}_n \rangle &= \frac{1}{2} \sum_{m=0}^{M} \sin(2nm\tau) = \frac{\sin(n(M+1)\tau)\sin(nM\tau)}{2\sin(n\tau)}, \\ \|\mathbf{u}_n\|^2 - \|\mathbf{v}_n\|^2 &= \sum_{m=0}^{M} \cos(2nm\tau) = \frac{\sin(n(M+1)\tau)\cos(nM\tau)}{\sin(n\tau)}, \\ \|\mathbf{u}_n\|^2 + \|\mathbf{v}_n\|^2 &= M + 1. \end{aligned}$$

3.2 Approximation structure

In this subsection, we show that one can study $SW_{M,\tau} f$ and the persistence of the point cloud it generates for a generic function $f \in L^2(\mathbb{T} = \mathbb{R}/2\pi\mathbb{Z})$ by using its Fourier series approximation.

Let $C(X, Y)$ denote the set of continuous functions from X to Y equipped with the sup norm. The sliding-window embedding includes a mapping

$$SW_{M,\tau} : C(\mathbb{T}, \mathbb{R}) \to C(\mathbb{T}, \mathbb{R}^{M+1}).$$

Proposition 3.2.1. *For all $M \in \mathbb{N}$ and $\tau > 0$, the map $SW_{M,\tau}$ is a bounded linear operator with norm $\|SW_{M,\tau}\| \leq \sqrt{M+1}$.*

We now consider approximating a function f by its Fourier polynomials. In particular, let

$$f(t) = S_N f(t) + R_N f(t)$$

where

$$S_N f(t) = \sum_{n=0}^{N} a_n \cos(nt) + b_n \sin(nt).$$

We can easily compute that

$$SW_{M,\tau} f(t) = \sum_{n=0}^{N} \cos(nt)\,(a_n \mathbf{u}_n + b_n \mathbf{v}_n) + \sin(nt)\,(b_n \mathbf{u}_n - a_n \mathbf{v}_n) + SW_{M,\tau} R_N f(t).$$

The vector $\mathbf{u}_n$ and $\mathbf{v}_n$ form a fundamental basis. We introduce the notation

$$\phi_\tau(t) = \sum_{n=0}^{N} \cos(nt)\,(a_n \mathbf{u}_n + b_n \mathbf{v}_n) + \sin(nt)\,(b_n \mathbf{u}_n - a_n \mathbf{v}_n).$$

Proposition 3.2.2. *Let $M\tau < 2\pi$. Then $\mathbf{u}_0 = \mathbf{1}, \mathbf{u}_1, \mathbf{v}_1, \ldots, \mathbf{u}_N, \mathbf{v}_N$ are linearly independent if and only if $M \geq 2N$.*

Given $N \in \mathbb{N}$, we will always set $M = 2N$ and require $\tau > 0$ to be such that $M\tau < 2\pi$.

Definition 3.2.3. We say that a function f is ***L*-periodic** on $[0, 2\pi]$, $L \in \mathbb{N}$, if

$$f(t + \frac{2\pi}{L}) = f(t)$$

for all t.

If we let $a_n + ib_n = r_n e^{i\alpha_n}$, with $\alpha_n = 0$ whenever $r_n = 0$, then $r_n \neq 0$ implies $n \equiv 0 \pmod{L}$. This is because if we set $g(t) = f(\frac{t}{L})$, then $g(t + 2\pi) = g(t)$ and therefore has a Fourier series expansion

$$g(t) = \sum_{r=0}^{\infty} a'_r \cos(rt) + b'_r \sin(rt)$$

with equality almost everywhere. Thus,

$$f(t) = g(tL) = \sum_{r=0}^{\infty} \big(a'_r \cos(rLt) + b'_r \sin(rLt)\big) = \sum_{n=0}^{\infty} \big(a_n \cos(nt) + b_n \sin(nt)\big).$$

for almost every t, and the result follows from the uniqueness of the Fourier expansion in $L^2(\mathbb{T})$.

Proposition 3.2.4. *Let f be L-periodic, and let $\tau = \dfrac{2\pi}{L(M+1)}$. Then the vectors in*

$$\{\mathbf{u}_n, \mathbf{v}_n \;\mid\; 0 \leq n \leq N,\ n \equiv 0 \pmod{L}\}$$

are mutually orthogonal, and we have $\|\mathbf{u}_n\| = \|\mathbf{v}_n\| = \sqrt{\frac{M+1}{2}}$ for $n \equiv 0 \pmod{L}$.

We then define the centering map $C : \mathbb{R}^{M+1} \to \mathbb{R}^{M+1}$:

$$C(\mathbf{x}) = \mathbf{x} - \frac{\langle \mathbf{x}, \mathbf{1} \rangle}{\|\mathbf{1}\|^2}\,\mathbf{1} \quad \text{where} \quad \mathbf{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^{M+1}.$$

Later we will set after centering and normalizing, the persistence diagram will have good properties.

3.3 Convergence

First we can control the pointwise distance of $SW_{M,\tau}f(t)$ and $\phi_\tau(t)$ by the following proposition.

Proposition 3.3.1. *Let $k \in \mathbb{N}$. If $f \in C^k(\mathbb{T}, \mathbb{R})$, then for all $t \in \mathbb{T}$*

$$\left\|SW_{M,\tau}f(t) - \phi_\tau(t)\right\|_{\mathbb{R}^{M+1}} \;\le\; \sqrt{4k-2}\left\|R_N f^{(k)}\right\|_2 \cdot \frac{\sqrt{M+1}}{(N+1)^{k-\frac{1}{2}}}.$$

From the stability theorem:

$$d_B(\mathrm{dgm}(X), \mathrm{dgm}(Y)) \;\le\; 2d_{GH}(X,Y) \;\le\; 2d_H(X,Y).$$

Let X, Y be the images of $T \subset \mathbb{T}$ through $SW_{M,\tau}f$ and ϕ_τ, respectively. It follows that if $f \in C^k(\mathbb{T}, \mathbb{R})$,

$$d_{\mathrm{B}}\big(\mathrm{dgm}(X), \mathrm{dgm}(Y)\big) \;\le\; 2\sqrt{4k-2}\left\|R_N f^{(k)}\right\|_2 \cdot \frac{\sqrt{M+1}}{(N+1)^{k-\frac{1}{2}}}.$$

Moreover, if M and N satisfy some conditions, the basis $\{\mathbf{u}_n\}$ and $\{\mathbf{v}_m\}$ are linearly independent. Then $S_N f$ can be recovered from $\phi_\tau(t)$, that is, there is no loss of information.

Then we will consider the bottleneck distance of respective persistence diagrams. We require $T \subset \mathbb{T}$ to be finite since the persistence diagram with bottleneck distance is complete under the finite set.

Proposition 3.3.2. *Let f be L-periodic, $N < N'$, $M = 2N$, $M' = 2N'$, and*

$$\tau = \frac{2\pi}{L(M+1)}, \qquad \tau' = \frac{2\pi}{L(M'+1)}.$$

If $T \subset \mathbb{T}$ is finite, $Y = SW_{M,\tau}S_N f(T)$ and $Y' = SW_{M,\tau'}S_{N'}f(T)$, then

$$d_{\mathrm{B}}\left(\frac{\mathrm{dgm}(Y)}{\sqrt{M+1}}, \frac{\mathrm{dgm}(Y')}{\sqrt{M'+1}}\right) \;\le\; 2\left\|S_N f - S_{N'} f\right\|_2,$$

where $\lambda \cdot \mathrm{dgm}(Z)$ is defined as $\{(\lambda x, \lambda y) | (x,y) \in \mathrm{dgm}(Z)\}$ for $\lambda \ge 0$.

Let $\bar{Y}_N$ be the set resulting from pointwise centering and normalizing the point cloud $SW_{2N,\tau_N}S_N f(T)$. Then we have following proposition.

Proposition 3.3.3. *For any field of coefficients, the sequence $\mathrm{dgm}(\bar{Y}_N)$ of persistence diagrams is Cauchy with respect to d_{B}.*

Completeness of the set, allows us to state the convergence of this Cauchy sequence. Let $w = \dfrac{2\pi}{L}$, and denote by $\mathrm{dgm}_\infty(f, T, w)$ the limit in the bottleneck distance of the sequence $\mathrm{dgm}(\bar{Y}_N)$. Here, the window size $M\tau = \dfrac{M}{M+1}w \to w$, as $N = \dfrac{1}{2}M \to \infty$ We have following convergence theorem.

Theorem 3.3.4. *Let $f \in C^1(\mathbb{T})$ be an L-periodic function, $N \in \mathbb{N}$, $\tau_N = \dfrac{2\pi}{L(2N+1)}$, $T \subset \mathbb{T}$ finite and let $\bar{Y}_N$ be as before. Let $\bar{X}_N$ be the set resulting form pointwise centering and normalizing the point cloud $SW_{2N,\tau_N}f(T) \subset \mathbb{R}^{2N+1}$. Then for any field of coefficients, the sequence $\mathrm{dgm}(\bar{X}_N)$ of persistence diagrams is Cauchy with respect to d_{B}, and*

$$\lim_{N\to\infty} \mathrm{dgm}(\bar{X}_N) = \lim_{N\to\infty} \mathrm{dgm}(\bar{Y}_N) = \mathrm{dgm}_\infty(f, T, w).$$

Next we will show that there is also convergence when T tends to $\mathbb{T}$ with respect to the Hausdorff distance on subspace of $\mathbb{T}$.

Theorem 3.3.5. *Let $T, T' \subset \mathbb{T}$ be finite, and let $f \in C^1(\mathbb{T})$ be L-periodic with modules of continuity $\omega : [0, \infty] \to [0, \infty]$. If $w = \dfrac{2\pi}{L}$, then*

$$d_{\mathrm{B}}(\mathrm{dgm}_\infty(f, T, w), \mathrm{dgm}_\infty(f, T', w)) \;\leq\; 2\,\|f - \hat{f}(0)\|_2\,\omega\big(d_H(T, T')\big)$$

and thus there exists a persistence diagram $\mathrm{dgm}_\infty(f, w)$ so that

$$\lim_{T \to \mathbb{T}} \mathrm{dgm}_\infty(f, T, w) = \mathrm{dgm}_\infty(f, w).$$

3.4 Classification

We can use various methods to classify whether a time series is periodic or not. In this subsection, we use `Maincodes.py`.

3.4.1 Giotto-tda

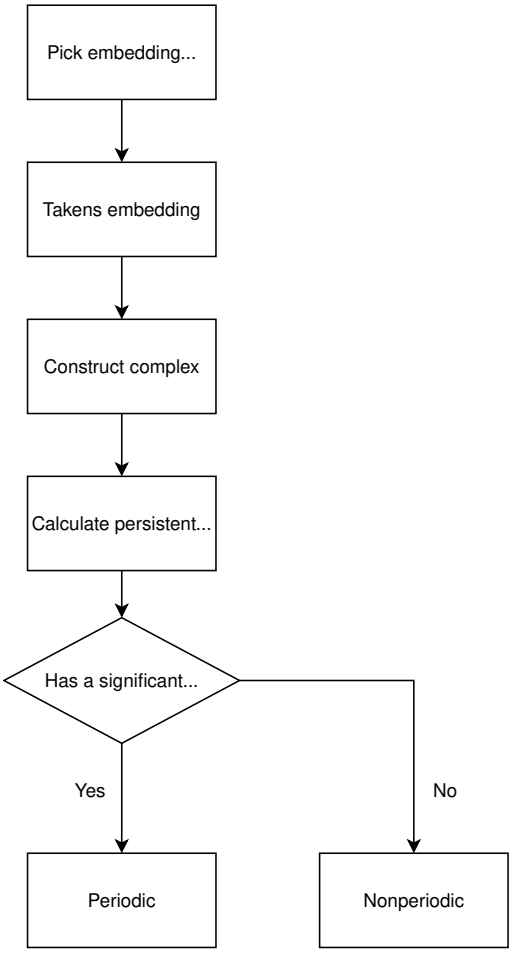


Figure 4: Steps of giotto-tda

The Giotto tutorials can be found here: `https://giotto-ai.github.io/gtda-docs/latest/notebooks/tutorials.html`

Step 1: Pick embedding dimension and time delay

We use mutual information to determineτ, and false nearest neighbours to determine d.

To determine an optimal value for τ, we first calculate the maximum $x_{\max}$ and minimum $x_{\min}$ values of the time series, and divide the interval $[x_{\min}, x_{\max}]$ into a large number of bins. We let p_k be the probability that an element of the time series is in the kth bin and let $p_{j,k}$ be the probability that x_i is in the jth bin while $x_{i+\tau}$ is in the kth bin. Then the mutual information is defined as:

$$I(\tau) = -\sum_{j=1}^{n_{\text{bins}}} \sum_{k=1}^{n_{\text{bins}}} p_{j,k}(\tau) \log \frac{p_{j,k}(\tau)}{p_j p_k}.$$

The first minimum of $I(\tau)$ gives the optimal time delay since there we get the most information by adding $t_{i+\tau}$.

The false nearest neighbours algorithm is based on the assumption that "unfolding" or embedding a deterministic system into successively higher dimensions is smooth. In other words, points which are close in one embedding dimension should be close in a higher one. More formally, if we have a point p_i and neighbour p_j, we check if the normalised distance R_i for the next dimension is greater than some threshold R_{th}:

$$R_i = \frac{|x_{i+m\tau} - x_{j+m\tau}|}{\|p_i - p_j\|} > R_{\text{th}}.$$

If $R_i > R_{\text{th}}$ then we have a "false nearest neighbour" and the optimal embedding dimension is obtained by minimising the total number of such neighbours.

Step 2: Takens embedding

Given a time series $f(t)$, we first extract a sequence of vectors of the form

$$SW_{d-1,\tau} f(t) = [f(t), f(t+\tau), \cdots, f(t+(d-1)\tau)] \in \mathbb{R}^d$$

where d is the embedding dimension and τ is the time delay. Since t is a discrete variable, denoted as $\{t_i\}_{i=0}^n$. We can obtain a sequence of vector in $\mathbb{R}^d$:

$$[f(t_i), f(t_i+\tau), \cdots, f(t_i+(d-1)\tau)], \quad 0 \le i \le n.$$

Step 3: Construct complex

Then we denote the finite point cloud in $\mathbb{R}^d$ by X. We create the Vietoris–Rips complex from X, denoted as $VR(X)$.

Step 4: Calculate persistence diagram

Use gitto-tda, we can easily compute the persistence diagram of $VR(X)$.

Example 3.4.1 (Periodic function)**.** Let's begin with a periodic function

$$f(t) = \cos(5t), \{t_i = \frac{i}{1000}\}_{i=0}^{1001}.$$

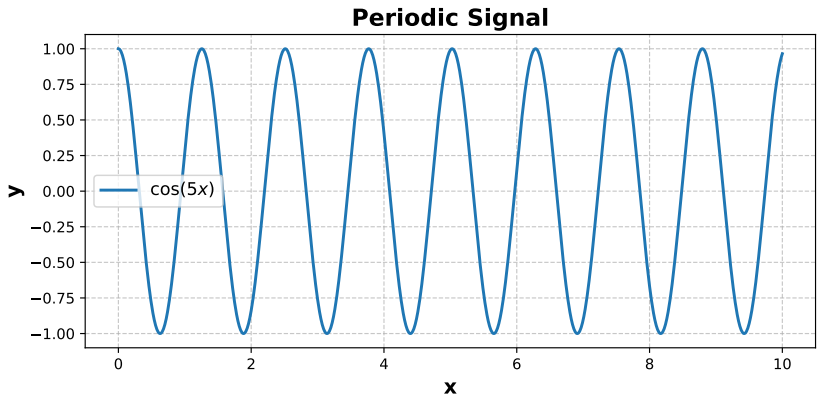


Figure 5: the periodic function

Use the search functions in **Step1**, optimal embedding dimension is 6 and time delay is 28. So $\tau = 28$, $d = 6$ and let stride to be 3, which means we have following vectors in $\mathbb{R}^6$:

$$[f(t_0), f(t_{28}), \cdots, f(t_{140})],$$

$$[f(t_3), f(t_{31}), \cdots, f(t_{143})],$$

$$\vdots$$

To visualize the Takens embedding obtained in higher dimensions, we applied Principal Component Analysis (PCA) to project the embedding into three dimensions. The first three principal components (PC1, PC2, PC3) capture most of the variance of the embedded trajectory, allowing us to inspect its geometry in a 3D plot.

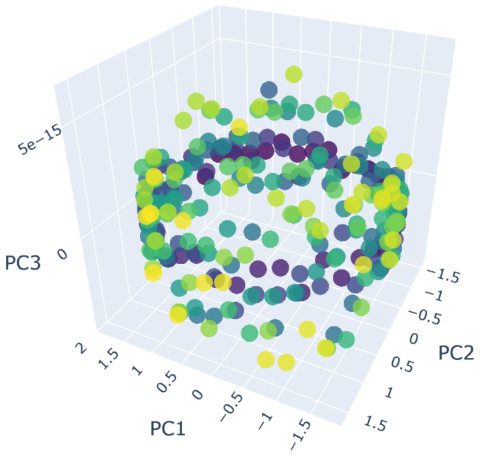


Figure 6: First figure of the point cloud

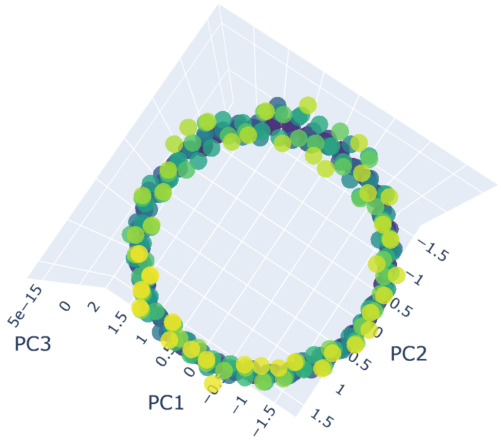


Figure 7: Second figure of the point cloud

The next step is to calculate the persistence diagrams associated with the point cloud. Let's first set the coefficient field to be $\mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z}$.

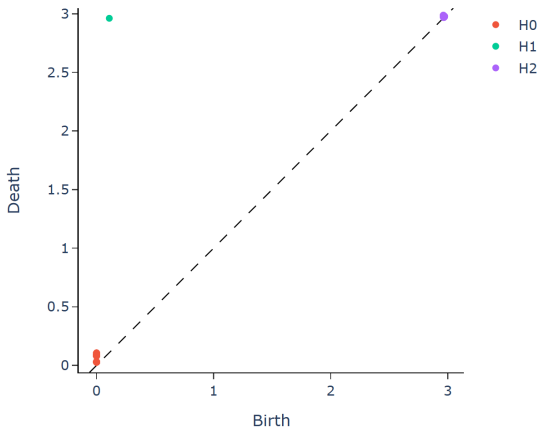


Figure 8: The persistence diagram for a periodic function

We can see that this persistence diagram has an important 1-class, which is away from the diagonal line. And there's no important 2-class, which means that there's no void in the complex.

Example 3.4.2 (Nonperiodic function)**.** Then we consider a nonperiodic function

$$g(t) = \cos(t) + \cos(\pi t), \ \{t_i = \frac{i}{20}\}_{i=0}^{1001}.$$

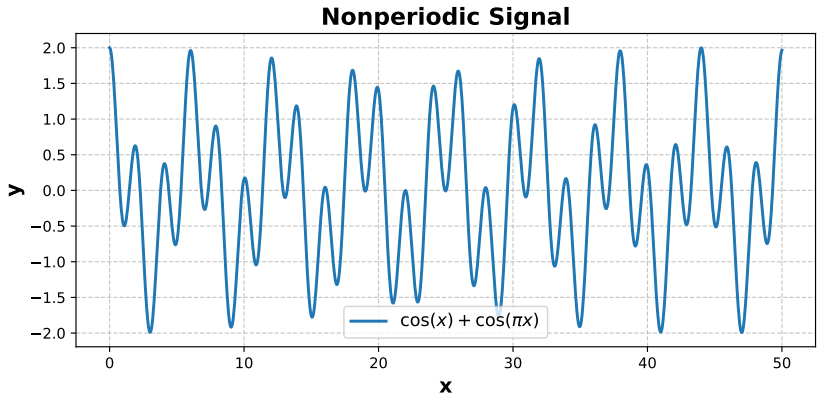


Figure 9: The nonperiodic function $g(t)$

Use the search functions in **Step1**, optimal embedding dimension is 6 and time delay is 30. So $\tau = 30$, $d = 6$ and let stride to be 3.

Then the point cloud has the following shape:

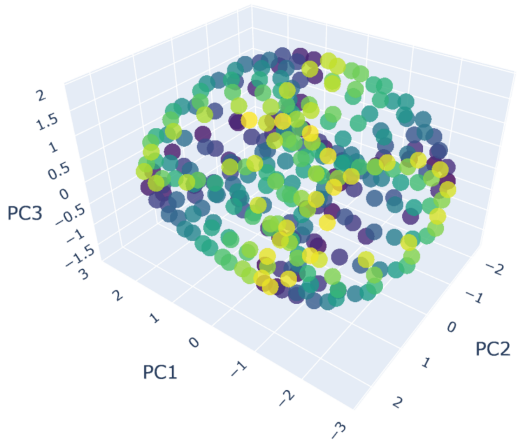


Figure 10: First figure of the point cloud

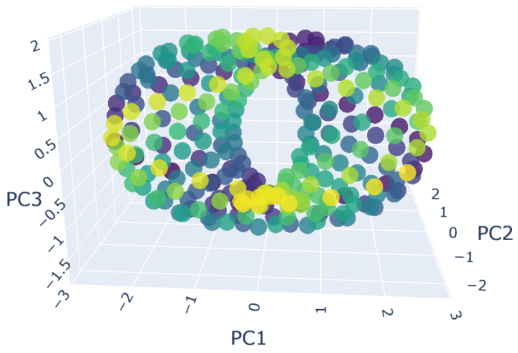


Figure 11: Second figure of the point cloud

In the second figure, we can see that the point cloud looks like a hypertorus. We verify this by computing its persistent homology. Let's first set the coefficient field to be $\mathbb{Z}_2 = \mathbb{Z}/2\mathbb{Z}$.

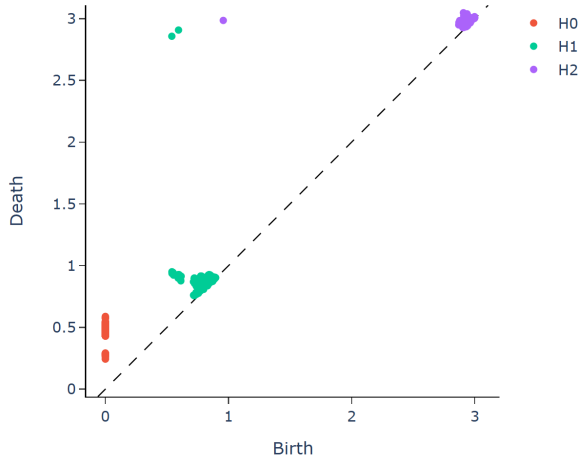


Figure 12: The persistence diagram for a nonperiodic function

The non-periodic signal reveals two 1-classes and one 2-class, which is the signature of a hypertorus!

At this point, it is natural to investigate how different choices of stride and the coefficient field may influence the results, as these parameters play distinct roles in the embedding and homological computation processes.

Time delay and embedding dimension We may feel curious about the importance of choosing the optimal time delay τ and embedding dimension d. Let's consider the periodic function:

$$h(t) = \cos(5t) + \sin(7t), \ \{t_i = \frac{i}{20}\}_{i=0}^{1001}$$

with the stride fixed to be 4 and coefficient field set to $\mathbb{Z}_2$.

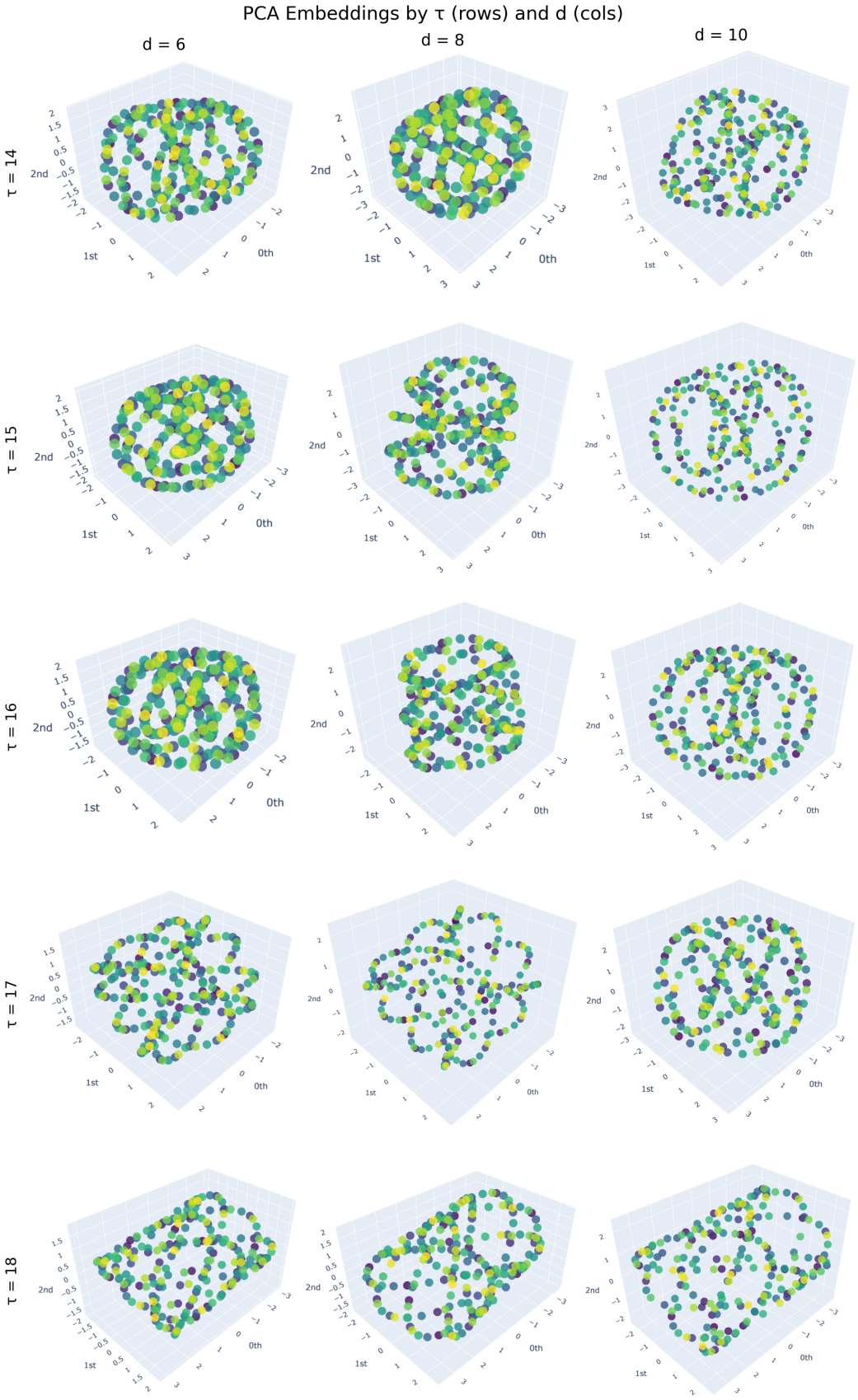


Figure 13: The PCA embedding

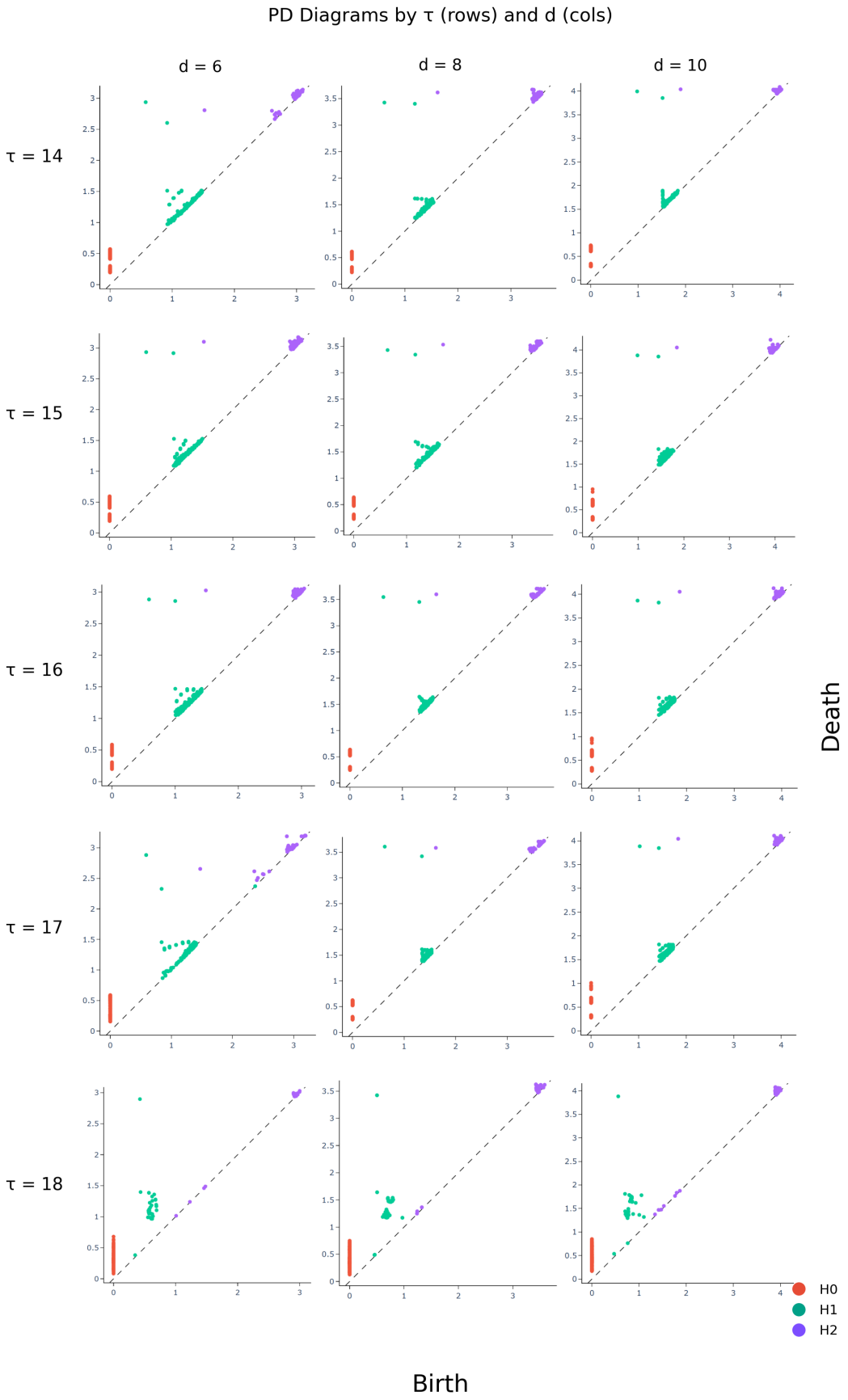


Figure 14: The persistence diagrams

From the search function, the best parameters for $h(t)$ are $\tau = 16$ and $d = 8$. Figures 13 and 14 consistently show that τ is the key hyperparameter: at $\tau = 16$ the dominant H_1 (and H_2) classes are relatively longest-lived and near-diagonal clutter is reduced, and the delay-embedding clouds are the cleanest and least folded. By contrast, varying the embedding dimension from $d = 6$ to $d = 10$ has only a minor effect once d is moderately large, indicating reconstruction stability. Moving τ away from 16 (e.g., to 14 or 18) weakens persistence and yields more tangled PCA projections. Thus, d only needs to be sufficiently large, while τ governs the topological and geometric clarity. This tells us picking the optimal embedding dimension and time delay is necessary.

Next we try to quantify the signals in the persistence diagram, using `Quantities.py`. We first quantify the signal strength in the q-th persistence diagram dgm_q by the L^2 norm of the barcode lengths. Let $\mathrm{pers}(x) = d - b$ for $x = [d, b) \in \mathrm{dgm}_q$. Then

$$\| \mathrm{dgm}_q \|_2 \coloneqq \left(\sum_{x \in D_q(\mathbb{X}) \text{ finite}} \mathrm{pers}(x)^2 \right)^{1/2}.$$

Table 1: The L^2 norm of the barcodes

	H_1			H_2		
	$d = 6$	$d = 8$	$d = 10$	$d = 6$	$d = 8$	$d = 10$
$\tau = 14$	3.154	3.726	3.965	1.369	2.110	2.201
$\tau = 15$	3.151	3.721	3.916	1.670	1.933	2.303
$\tau = 16$	3.111	3.721	3.930	1.624	2.035	2.308
$\tau = 17$	3.032	3.737	3.910	1.283	2.040	2.333
$\tau = 18$	3.881	4.603	5.154	0.248	0.485	0.449

Next, we quantify the signal strength by the L^3 norm of the barcode lengths.

$$\| \mathrm{dgm}_q \|_3 \coloneqq \left(\sum_{x \in D_q(\mathbb{X}) \text{ finite}} \mathrm{pers}(x)^3 \right)^{1/3}.$$

Table 2: The L^3 norm of the barcodes

	H_1			H_2		
	$d = 6$	$d = 8$	$d = 10$	$d = 6$	$d = 8$	$d = 10$
$\tau = 14$	2.649	3.219	3.429	1.292	2.005	2.140
$\tau = 15$	2.700	3.189	3.381	1.576	1.844	2.213
$\tau = 16$	2.659	3.253	3.384	1.545	1.955	2.207
$\tau = 17$	2.524	3.288	3.359	1.195	1.970	2.216
$\tau = 18$	2.747	3.245	3.665	0.159	0.308	0.292

The third one is the maximum persistence, which can be regarded as the signal strength by the L^∞

norm of the barcode lengths.

$$\max(\mathrm{dgm}_q) \coloneqq \max_{x \in D_q(\mathbb{X}) \text{ finite}} |\,\mathrm{pers}(x)\,|.$$

Table 3: The maximum persistence of the barcodes

	H_1			H_2		
	$d=6$	$d=8$	$d=10$	$d=6$	$d=8$	$d=10$
$\tau=14$	2.364	2.812	3.013	1.288	1.999	2.138
$\tau=15$	2.339	2.789	2.900	1.572	1.840	2.209
$\tau=16$	2.298	2.909	2.903	1.542	1.952	2.202
$\tau=17$	2.295	2.982	2.862	1.183	1.968	2.211
$\tau=18$	2.471	2.921	3.324	0.096	0.198	0.185

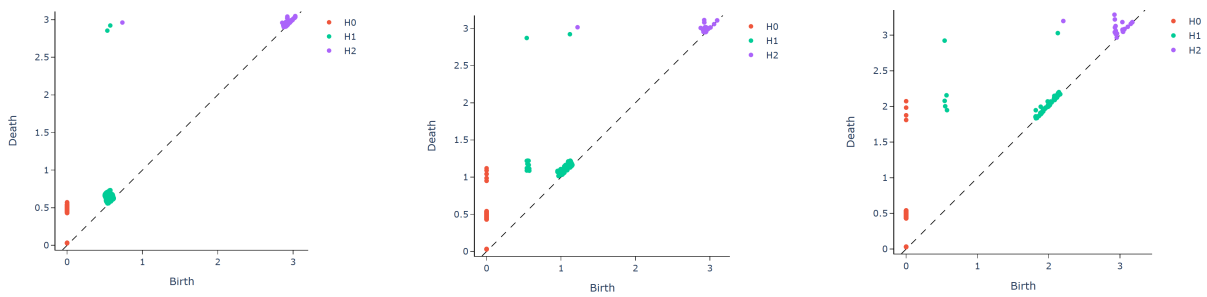
From Table 1, the L^2 norms change little with the embedding dimension: for $\tau \in \{14, 15, 16, 17\}$, $\|\,\mathrm{dgm}_1\,\|_2$ varies only slightly across $d \in \{6, 8, 10\}$, and $\|\,\mathrm{dgm}_2\,\|_2$ stays in a similar range (about 1.9–2.3). By contrast, the time delay has a pronounced effect: at $\tau = 18$ the H_1 norm increases markedly while the H_2 norm collapses (to < 0.5), indicating that 2D classes are no longer significant. Overall, these trends support our earlier conclusion that τ is the critical hyperparameter, whereas d only needs to be moderately large. We can draw the same conclusion from Table 2 and Table 3.

We may notice that there is a huge jump between when $\tau = 17$ and $\tau = 18$. The reason may be the ratio $\dfrac{2\pi}{18 * 0.05} = 6.98 \approx 7$. The same phenomonen occurs when $\tau = 21$, then the ratio becomes $\dfrac{2\pi}{21 * 0.05} = 5.98 \approx 6$. However, there are also some cases strange. For example, when $\tau = 12$, the ratio is not close to a integer, but the quantities of H_2 are super low.

Stride Actually, if the stride is not large enough, it has little effect on the persistence diagram. However, if the stride is not so small, the calculation will be much faster. We'll show this by choosing different stride in the nonperiodic function(since it's more complex). For the nonperiodic function

$$g(t) = \cos(t) + \cos(\pi t),\ \{t_i = \frac{i}{20}\}_{i=0}^{1001}.$$

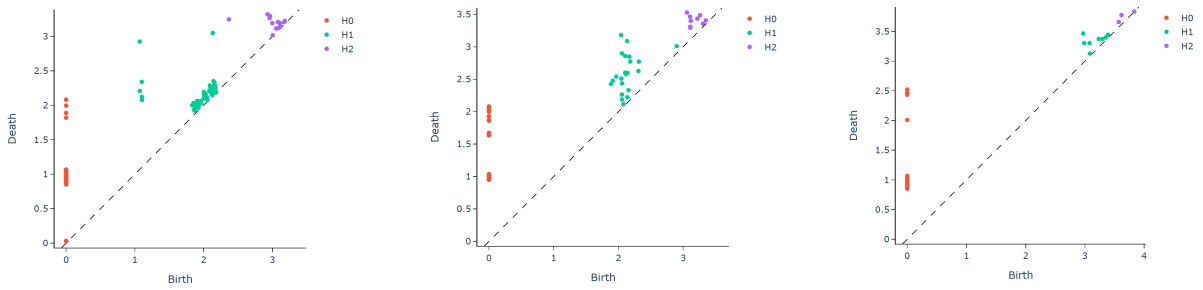
We already know that the optimal embedding dimension is 6 and time delay is 14.



(a) stride=2 (b) stride=4 (c) stride=8

Figure 15: The persistence diagram when the stride is small

From Figure 15, we observe that all three cases exhibit two prominent 1-dimensional homology classes and one prominent 2-dimensional homology class. Nevertheless, the 1-dimensional classes in the cases with stride 2 and 4 appear to be more dominant compared to those obtained with stride 8.



(a) stride=16 (b) stride=32 (c) stride=48

Figure 16: The persistence diagram when the stride is large

Comparing Figure 15 and Figure 16, we observe that although both exhibit two significant 1-dimensional homology classes and one 2-dimensional class when the stride is small, the persistence of these features is less pronounced in Figure 15a. In particular, as the stride increases, the topological signatures become weaker and gradually vanish; at stride 48, the 1-dimensional and 2-dimensional classes are barely visible, indicating that an excessively large stride results in a severe loss of topological information.

Next we use the L^2 norm of the barcodes and the maximum persistence to to quantify how persistence changes as the stride varies.

Table 4: The L^2 norm of the barcodes($d = 6, \tau = 14, \text{coffe} = \mathbb{Z}_2$)

stride	2	3	4	5	6	7	8
$\| \text{dgm}_1 \|_2$	3.848	3.786	3.596	3.640	3.545	3.187	3.935
$\| \text{dgm}_2 \|_2$	2.273	2.047	1.873	1.688	1.437	1.523	1.191

Table 5: The maximum persistence($d = 6, \tau = 14, \text{coffe} = \mathbb{Z}_2$)

stride	2	3	4	5	6	7	8
$\max(\text{dgm}_1)$	2.949	2.316	2.348	2.373	1.928	1.914	2.383
$\max(\text{dgm}_2)$	2.231	1.986	1.809	1.598	1.326	1.421	0.996

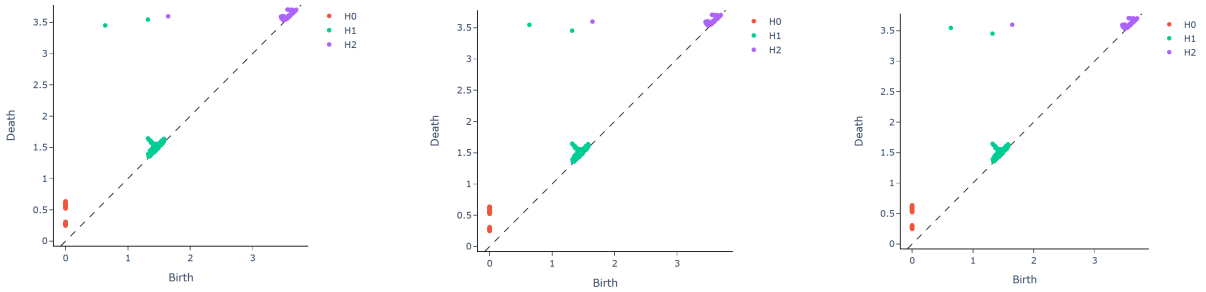
From Tables 4 and 5, for strides $s \in \{2, 3, 4\}$ the L^2 norm and the maximum persistence in dimensions 1 and 2 vary only slightly. However, as the stride increases to $s \in \{6, 7, 8\}$, the H_2 signal collapses—bars in the 2D persistence diagram become negligible—so the diagram carries little information.

Coefficient field In giotto-tda, the default coefficient field for homology computation is $\mathbb{Z}_2 = \mathbb{Z}/2\mathbb{Z}$. A natural question is whether the resulting persistence diagrams exhibit substantial changes when com-

puted over different fields. To investigate this, using `VerifyCoeffiDependence.py`, we first consider the periodic function

$$h(t) = \cos(5t) + \sin(7t),\ \{t_i = \frac{i}{20}\}_{i=0}^{1001}$$

fix the stride parameter at 4 and choose the optimal embedding dimension 8 and time delay 16. Compute the corresponding persistence diagrams under several choices of coefficient fields.



(a) coeff=5 (b) coeff=7 (c) coeff=11

Figure 17: The persistence diagram of a periodic function under different coefficient fields

For the periodic function, the persistence diagrams computed with different coefficient fields(e.g.,$\mathbb{Z}_5, \mathbb{Z}_7, \mathbb{Z}_{11}$)are almost identical. In all cases, we observe one significant 1-class and one significant 2-class, which are stable across different fields. This indicates that the topological features of periodic signals are robust and their detection does not depend on the choice of the coefficient field.

Table 6: The L^2 norm of the barcodes($d = 8, \tau = 16, \text{stride} = 4$)

characteristic	2	3	5	7	11	13	17
$\| \text{dgm}_1 \|_2$	3.721	3.721	3.704	3.721	3.721	3.721	3.721
$\| \text{dgm}_2 \|_2$	2.035	2.035	2.035	2.035	2.035	2.035	2.035

Table 7: The maximum persistence($d = 8, \tau = 16, \text{stride} = 4$)

characteristic	2	3	5	7	11	13	17
$\max(\text{dgm}_1)$	2.909	2.909	2.815	2.909	2.909	2.909	2.909
$\max(\text{dgm}_2)$	1.952	1.951	1.951	1.951	1.951	1.951	1.951

We find that except $\mathbb{Z}_5$, the other quantities are same under different coefficient field. Moreover, the difference of different quantities under field $\mathbb{Z}_2$ and $\mathbb{Z}_5$ is very small:

Table 8: The different quantities of the barcodes

	$\|\operatorname{dgm}_1\|_2$	$\|\operatorname{dgm}_1\|_3$	$\max(\operatorname{dgm}_1)$	$\|\operatorname{dgm}_2\|_2$	$\|\operatorname{dgm}_2\|_3$	$\max(\operatorname{dgm}_2)$
characteristic=2	3.720851	3.253383	2.909238	2.035045	1.954654	1.951770
characteristic=3	3.720851	3.253383	2.909238	2.035045	1.954654	1.951770
characteristic=5	3.703549	3.222506	2.815436	2.035045	1.954654	1.951770
characteristic=7	3.720851	3.253383	2.909238	2.035045	1.954654	1.951770

From Table 8, we find the 1th persistence diagram is independent of the choice of k, except $k = \mathbb{Z}_5$. The 2th persistence diagram is independent of the choice of k. Recall **Theorem2.2.2.**: The qth persistence diagram $D_q(\mathbb{X}; k)$ is independent of the choice of k if $H_q(X_n, X_m; \mathbb{Z})$ is free for any $0 \leq m < n \leq N$ and $H_{q-1}(X_n; \mathbb{Z})$ is free for any $0 \leq n \leq N$. We consider the independence of dgm_1 here. Since $H_0(X_n; \mathbb{Z})$ are always free, the only dangerous is the torsion of $Hq(X_n, X_m; \mathbb{Z})$.

We can verify this claim computationally by implementing in Python the procedure specified by the pseudocode below: construct the delay embedding, optionally subsample the embedded points, compute Vietoris–Rips persistent homology over each prime field $\mathbb{Z}_p (p \leq 13)$, and compare the resulting H_1 persistence diagrams against a baseline to detect any field dependence.

Algorithm 1: Field-dependence for H_1 persistence

Input : Scalar time series $y(1{:}N)$; Takens parameters (τ, d, stride); prime set $P = \{2, 3, 5, 7, 11, 13\}$; tolerance $\varepsilon > 0$; user-chosen cap $m \in \mathbb{N}$ (maximum number of embedded points used for homology).

Output: Set $\Delta \subseteq P$ of primes where the H_1 persistence diagram differs from the baseline, and a per-prime summary.

(1) Takens embedding. Construct the delay embedding $X_t \in \mathbb{R}^d$ from y with time delay τ, dimension d, and stride. This yields a point cloud $X = \{X_t\}_{t=1}^n \subset \mathbb{R}^d$.

(2) Optional subsampling. If $n > m$, choose indices $1 \leq t_1 < \cdots < t_m \leq n$ uniformly at random without replacement, and set $\tilde{X} = \{X_{t_i}\}_{i=1}^m \subset X$; else set $\tilde{X} := X$.

(3) Diagram on a field. For $p \in P$, compute the Vietoris–Rips persistent homology over $\mathbb{F}_p$ on $\tilde{X}$ and extract the H_1 diagram

$$\operatorname{dgm}_{1,\mathbb{F}_p} = \{(b_i^{(p)}, d_i^{(p)})\}_{i=1}^{N_p},$$

removing points with $d_i^{(p)} = +\infty$, and sorting by (b, d) lexicographically.

(4) Baseline comparison. Fix a baseline $p_0 \in P$ (e.g. $p_0 = 2$). For every $p \in P \setminus \{p_0\}$:

1. If $N_p \neq N_{p_0}$, record $p \in \Delta$ with summary "bar count differs".
2. Else compute $\delta_p = \max_{1 \leq i \leq N_p} \|(b_i^{(p)}, d_i^{(p)}) - (b_i^{(p_0)}, d_i^{(p_0)})\|_\infty$. If $\delta_p > \varepsilon$, record $p \in \Delta$ with summary "same count, max diff $= \delta_p$".

Return Δ and the associated summaries.

Using Algorithm 1, we find that—at the optimal embedding dimension and time delay—the 1-dimensional persistence diagram is independent of the coefficient field k for all tested fields except

$k = \mathbb{Z}_5$. For the time series

$$x_1(t) = \cos(5t) + \sin(3t), \quad t_i = \tfrac{i}{20},\ i = 0, \ldots, 1001$$

the 1-dimensional persistence of the Takens embedding is independent of the choice of k (except $k = \mathbb{Z}_3$). We verified this conclusion by comparing the L^2- and L^3-norms of the barcodes and the maximum persistence across fields, which agree with the output of the code. However, for the time series

$$x_2(t) = \cos(5t) + \sin(11t), \quad t_i = \tfrac{i}{20},\ i = 0, \ldots, 1001$$

the 1-dimensional persistence of the Takens embedding is independent of the choice of k.

It is worth noting that even a slight variation in the coefficients can alter the dependency behavior. For instance, for

$$x_2(t) = \cos(5t) + \sin(11t), \quad t_i = \tfrac{i}{20},\ i = 0, \ldots, 1001$$

the 1-dimensional persistence of the Takens embedding is independent of the choice of the coefficient field. However, for

$$x_3(t) = 0.6\cos(5t) + 0.8\sin(11t), \quad t_i = \tfrac{i}{20},\ i = 0, \ldots, 1001$$

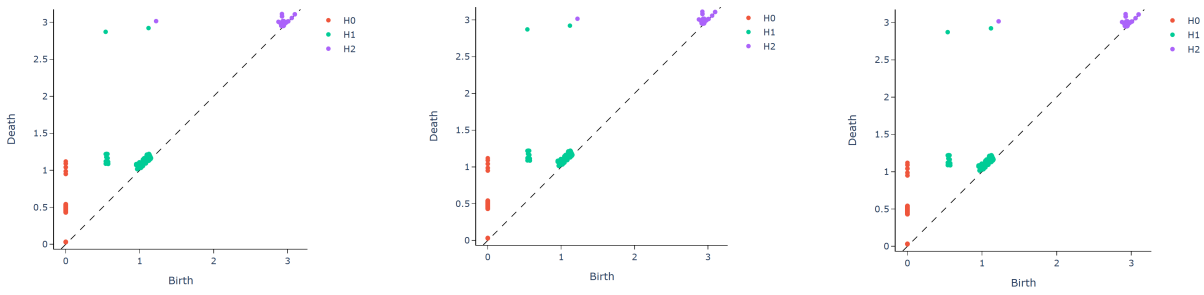
the persistence becomes dependent on the coefficient field (with the special characteristic being 3). In contrast, for

$$x_4(t) = \tfrac{1}{2}\cos(5t) + \tfrac{\sqrt{3}}{2}\sin(7t), \quad t_i = \tfrac{i}{20},\ i = 0, \ldots, 1001$$

the 1-dimensional persistence remains independent of the coefficient field. For nonperiodic function

$$g(t) = \cos(t) + \cos(\pi t), \quad t_i = \tfrac{i}{20},\ i = 0, \ldots, 1001,$$

fix the stride parameter at 4, and compute the corresponding persistence diagrams under several choices of coefficient fields.



(a) coeff=5 (b) coeff=7 (c) coeff=11

Figure 18: The persistence diagram of a nonperiodic function under different coefficient fields

In the case of the nonperiodic function $g(t) = \cos(t) + \cos(\pi t)$, the persistence diagrams still exhibit prominent and relatively stable one- and two-dimensional homology classes, reflecting the quasi-periodic structure of the signal. However, when we change the coefficient field(e.g.,$\mathbb{Z}_5, \mathbb{Z}_7, \mathbb{Z}_{11}$), the resulting diagrams remain largely consistent. This indicates that, although nonperiodic, the signal gives rise

to robust topological features whose detection is not sensitive to the choice of coefficient field. Using Algorithm 1, we verified computationally that the 1-dimensional persistence diagram is independent of the choice of coefficient field k (up to numerical tolerance).

Advantages and disadvantages It's noticeable that the optimal time delay and embedding dimension depend on the stride, while the dependence of coefficient field seems unrelated to the stride.

Giotto-tda makes it easy to construct complexes and compute the corresponding persistence diagrams. However, it becomes challenging to determine whether a function is periodic when the function has a more complicated form (not simply cosine-like). Let's consider the following periodic function:

$$h(t) = \cos(5t) + \sin(7t),\ \{t_i = \frac{i}{20}\}_{i=0}^{1001}$$

use the search functions, the optimal embedding dimension is 8 and the time delay is 16. Let stride be 4.

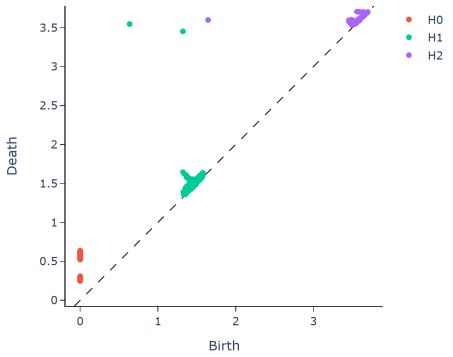


Figure 19: The persistence diagram for a periodic function

In comparison with Figure 12, it is difficult to distinguish whether the corresponding function is periodic.

3.4.2 SW1Pers

We now introduce a way to quantify the periodicity of sampled signals. To begin with, we associate to each sampled signal $S = [s_1, s_2, \cdots, s_J]$ a real-valued function f_S by cubic spline interpolation, construct its centered and normalized sliding-window point cloud X_S. We denote the maximum persistence of a 1D-persistence diagram dgm by $mp(\text{dgm})$, and let

$$\frac{mp(\text{dgm}(X_S))}{\sqrt{3}} = \text{Score}(S).$$

be its periodicity score. In the original MATLAB implementation of SW1PerS, the score is defined as

$$\text{Score}_{\text{matlab}} = 1 - \frac{mp(\text{dgm}(X_S))}{\sqrt{3}}.$$

This is an inverted version of the theoretical score defined. Hence, in the MATLAB code, smaller values indicate stronger periodicity, while in the theoretical definition, larger values indicate stronger periodicity. Both conventions are equivalent up to this inversion.

In MATLAB, we set the Fourier degree $N = 7$, the coefficient field$k = \mathbb{Z}_{11}$, the embedding dimension $M = 2N + 1$, and the time delay $\tau = \dfrac{2\pi}{(M+1) \cdot \text{num\_clycles}}$.

Algorithm 2: SW1PerS: Periodicity Scoring Procedure for Time Series (Simplified Version)

Input: Time series $\mathbf{s}$; number of cycles n_{cyc}; feature type `feature_type`.

Output: Score `score` $\in [0, 1]$, where a higher value indicates stronger periodicity.

Step 1. Signal smoothing and preprocessing: Optionally perform exponential moving average, moving average, or spline smoothing to remove noise and trends.

Step 2. Sliding window embedding: Map $\mathbf{s}$ into an $(M+1)$-dimensional point cloud $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_p}\}$, where each $\mathbf{x}_i$ is a time-delay vector; then apply mean-centering and ℓ_2 normalization.

Step 3. Post-processing of the point cloud: If mean-shift is enabled, smooth the point cloud using cosine distance to further reduce noise.

Step 4. Persistent homology computation: Construct the Vietoris–Rips complex of $\mathbf{X}$ over the coefficient field $\mathbb{Z}/p\mathbb{Z}$, and compute persistence intervals for H_0 and H_1.

Step 5. Periodicity scoring:

If H_1 is empty, set `score` $= 1$; otherwise, depending on the chosen feature type, compute the normalized persistence from the 1-dimensional intervals $\Delta_\ell = d_\ell^{(1)} - b_\ell^{(1)}$ to obtain the final score.

return `score`

User Guide We illustrate how to use the provided MATLAB code to analyze the periodicity of time series. Run the following command in the MATLAB Command Window:

Listing 1: Launching the main GUI (TDA_TimeSeries_GUI.m)

```
TDA_TimeSeries_GUI
```

The left panel of the interface serves as the input and parameter control area, while the right panel is used for visualization and result display. The overall layout mainly consists of the following parts:

(1) **Top-left:** `giotto-tda` input panel

Responsible for importing time series data and setting embedding parameters, including the file path, number of points, value range, maximum embedding dimension `max_dim`, maximum time delay `max_delay`, and stride `stride`.

(2) **Middle-left:** `SW1PerS` parameter panel

Controls the options related to sliding-window embedding, signal smoothing, and periodicity scoring.

(3) **Bottom-left:** Operation panel

Provides three main buttons:

1. **Generate Random Time Series:** Creates a random signal and displays it in the upper-left plot;
2. **Compute Persistence Diagram (Takens→VR):** Performs sliding-window embedding and constructs the Vietoris–Rips complex, showing the combined $H_0/H_1/H_2$ persistence diagram in the upper-right plot;
3. **SW1PerS Score:** Computes the periodicity score for the current signal.

(4) **Right panel:** Displays the time series and the corresponding persistent homology results.

(5) **Bottom status bar:** Shows the current computation status (e.g., "Ready", "Processing", etc.). After computing the SW1PerS score, the result is also displayed here.

Table 9: Description of input parameters for `SW1PerS`

Parameter	**Type**	**Description**
`num_cycles`	int	Number of cycles to detect; for example, 2 means detecting two complete periods.
`feature_type`	int (1–5)	Type of feature extraction method; different values correspond to different persistence-based scoring formulas. Usually, 3 is recommended.
`num_points`	int	Number of sampled points in the point cloud; larger values capture more topological features.
`movingavg_win`	int	Window size for moving average smoothing; a larger value produces stronger smoothing.
`allow_trending`	bool	Whether to allow mean-centering to remove linear trends.
`use_meanshift`	bool	Whether to enable mean-shift smoothing to reduce noise perturbation.
`ms_epsilon`	float	Radius parameter for mean-shift; defines the similarity range of neighboring points.
`expavg`	bool	Whether to apply exponential moving average filtering.
`expavg_alpha`	float	Smoothing factor α for exponential moving average.
`movingavg`	bool	Whether to enable standard moving average filtering.
`smoothspline`	bool	Whether to apply smoothing spline fitting.
`smoothspline_sigma`	float	Smoothing parameter for spline fitting, controlling the smoothness level.

Users can perform topological analysis and periodicity scoring of time series according to the following steps:

Step 1: Import or Generate a Time Series

Click the "Browse" button next to the "TXT Path" field to select an existing time series file, or click "1 Generate Random Time Series" to automatically create a random signal. The generated signal will be displayed in the upper-left plot window.

Step 2: Set Embedding Parameters

In the "`giotto-tda` Input" panel, specify the number of points, range, maximum embedding dimension `max_dim`, maximum time delay `max_delay`, and sliding step `stride`, which determine the time-delay structure for the Takens embedding.

Step 3: Compute Persistent Homology

Click the "2 Compute Persistence Diagram (Takens→VR)" button to automatically perform sliding-window embedding and Vietoris–Rips complex construction. The combined $H_0/H_1/H_2$ persistence diagram will be shown in the upper-right plot.

Step 4: Adjust SW1PerS Parameters

In the "`SW1PerS` Input" panel, adjust parameters such as the number of cycles `num_cycles`, feature type `feature_type`, and smoothing parameters (e.g., `movingavg_win`). Optional filters like exponential moving average, mean-shift, or spline smoothing can be enabled as needed. In most cases, the default parameter settings work best.

Step 5: Compute the Periodicity Score

Click the "SW1PerS Score" button to calculate the periodicity score under the current signal and parameter settings. The score ranges within $[0, 1]$, where values closer to 1 indicate stronger periodicity.

Step 6: View and Save Results

The output includes: (1) the time-series curve; (2) the corresponding persistence diagram; and (3) the SW1PerS numerical score. Users can take screenshots or export the results for further analysis or publication.

Example 3.4.3. Click Button 1 to generate a random time series consisting of 800 values within the range $[-7, 1]$. Keep the parameters max_dim=30,max_tau=30, and stride=4.

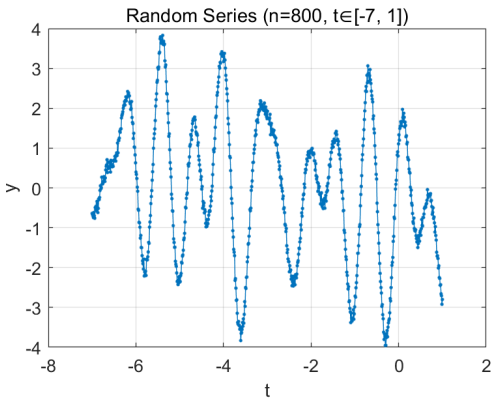


Figure 20: Example of random time series

We can make a preliminary observation that this random time series exhibits a certain degree of periodicity. Next, let us verify this by clicking Button 2 to compute its persistence diagram.

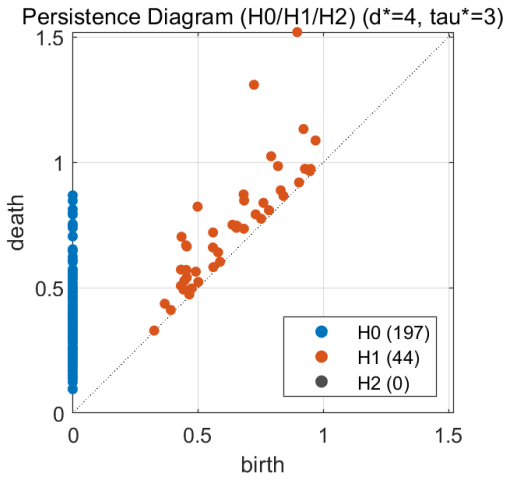


Figure 21: Persistence Diagram

From Figure 21, we can observe that the corresponding persistence diagram shows a prominent one-dimensional homology class, while no significant two-dimensional homology classes appear. This indicates that the time series possesses noticeable periodicity. By clicking Button 3, the computed SW1PerS score is 0.7406, which confirms that the time series exhibits a very strong periodicity, validating our initial hypothesis.

4 Application to knot theory

Persistence homology can also be used in knot theory. For example, we consider that does the topology of a random knot influence how it 'occupies' space. Similar ideas have previously been considered often with the main goal of understanding the mechanism of DNA.

We call a knot K in the image of a PL embedding of S^1 in S^3. Use the notation $\ell(K)$ to denote the length of a knot K. In this section, we follow [CM22].

We denote by $v_t(K)$ the metric neighborhood of K of radius $t > 0$. The injectivity radius $\mathrm{IR}(K)$ is the supremum among all radii $t \geq 0$ for which the tubular neighborhood $v_t(K)$ is regular. In other words, $\mathrm{IR}(K)$ is the smallest value of the neighborhood's radius such that $v_t(K)$ comes into contact with itself. We use $\mathrm{RS}(K)$ to denote the radius of the circumscribing sphere of K. It's easy to see that $\mathrm{IR}(K)$ depends on the embedding way of the knot K, so it's not an invariant under the equivalence class of knot type. However, for the length over diameter ratio of K:

$$L/D(K) = \frac{\ell(K)}{2\mathrm{IR}(K)}.$$

We use $\mathcal{K}$ to denote the knot type of K, then $\inf_{K\in\mathcal{K}} L/D(K)$ is a knot invariant. We denote that an ideal knot is an embedded knot that minimizes the L/D ratio within its knot types.

4.1 Correlation

To obtain random polygonal knots used in this study, we employed the `Topoly` library to generate equilateral random closed polygons of a prescribed number of vertices. Specifically, we used the function `topoly.generate_loop(N, 1, bond_length)` to sample random loops in three–dimensional space, where each edge has fixed length and the closure condition is automatically enforced by `Topoly`.

For each parameter setting (e.g., $N = 40$), we generated multiple independent realizations using the following command:

```
python GenRandomKnot.py --length 40 --count 500
```

Each polygonal chain was saved in `.xyz` format under the `output_coordinate/` directory, with coordinates centered and without any topological filtering or classification.

Hence, the resulting dataset consists of 500 equilateral random polygonal loops of length $N = 40$, representing unbiased random knots generated purely by stochastic geometric sampling.

4.1.1 Random PL knots

Generation and Calculation The random polygonal knots were generated using `GenRandomKnot.py` For each prescribed length ranging from 5 to 100 (in increments of 5), we generated 500 random equilateral polygonal knots.

For each knot, several geometric and topological quantities were computed, including the volume of its minimal enclosing sphere, the volume of its convex hull, the total curvature, torsion, and radius of gyration. We then computed the persistence barcodes corresponding to the Alpha filtrations associated with the knots using the efficient library `gudhi`.

To approximate the topology of the neighbourhoods of the generated embeddings as closely as possible, we interpolated each unit segment of the polygonal embedding with ten equidistant points and computed the Alpha filtration on the resulting dense point cloud. Using ten points per linear segment

provides a sufficiently fine sampling of each embedding, yielding an Alpha filtration that closely approximates the growth of the knot's tubular neighbourhood for most cases. From the resulting persistence diagrams, we extracted the maximum bar lengths in dimensions 1 and 2, denoted by $H_{1,\max}$ and $H_{2,\max}$, as well as the integrals of the first and second Betti curves, denoted by I_1 and I_2, respectively. All these quantities were calculated using `ComputeQuantities.py`.

The average crossing number (ACN) is computed by `ACN.py`, using `pyknotid`.

We then used the Pearson correlation coefficient to quantify the relationships between the geometric structures of the embeddings and the quantities derived from persistent homology, and plotted the corresponding correlation figures.

Result It can be observed that the correlations of $H_{1,\max}$ with the curvature, and of I_1 with the radius of gyration, as well as with the volumes of the convex hull and the circumscribing sphere, depend on the length of the knots. In contrast, the correlations between other geometric and topological quantities appear to be independent of knot length.

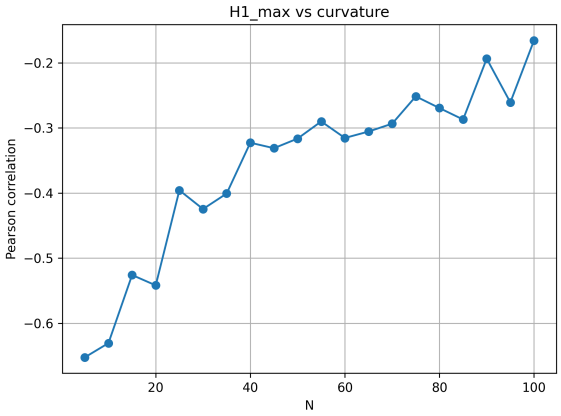


Figure 22: H1_max vs curvature

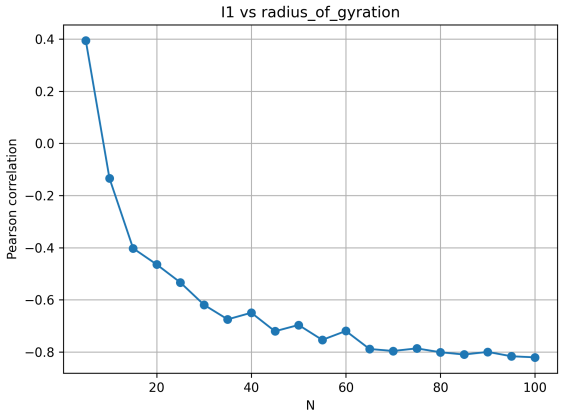


Figure 23: I1 vs radius of gyration

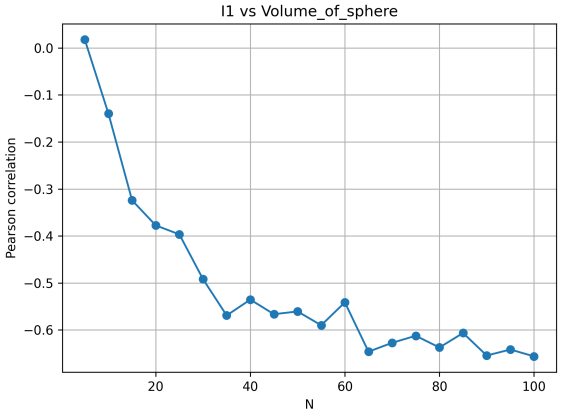


Figure 24: I1 vs Volume of sphere

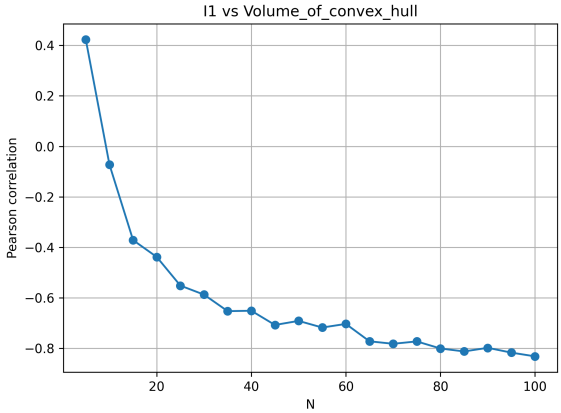


Figure 25: I1 vs Volume of convex hull

However, we may observe that $H_{1,\max}$ is strongly positively correlated with the radius of gyration.

This can be explained by the fact that a long H_1 bar in the persistence diagram indicates a simpler topological structure, and thus the knot tends to be more spatially extended, resulting in a larger radius of gyration. By contrast, when I_1 is relatively large, the knot is topologically more complex, exhibiting a greater number of H_1 bars; consequently, the embedding becomes more compact and the radius of gyration is smaller. Therefore, I_1 is negatively correlated with the radius of gyration. Similar reasoning can be applied to interpret the sign and magnitude of the correlation coefficients for other pairs of quantities.

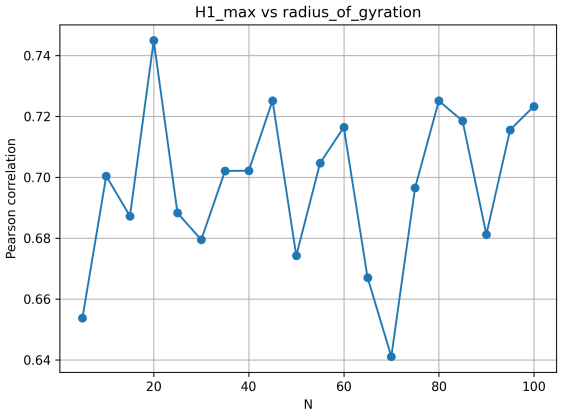


Figure 26: H1_max vs radius of gyration

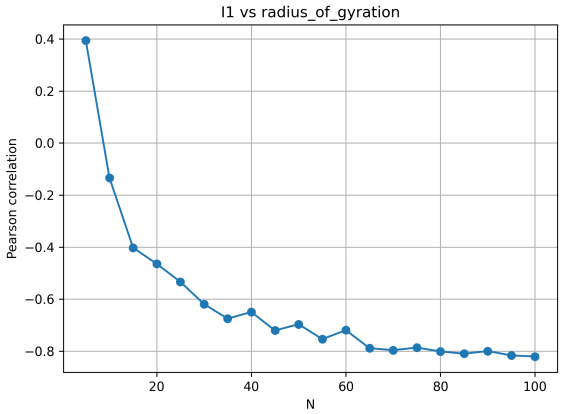
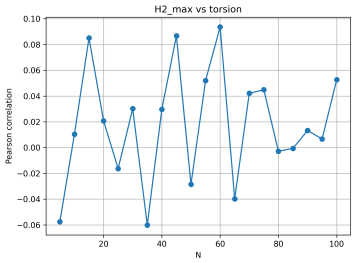


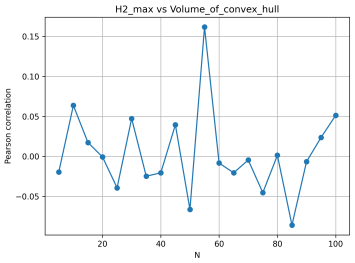
Figure 27: I1 vs radius of gyration

Unfortunately, the correlations between the second-dimensional topological quantities and the geometric structures of the embeddings are generally weak.

(a) H2_max vs radius of gyration

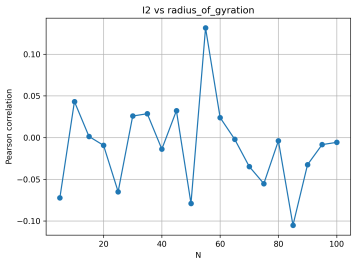


(b) H2_max vs torsion

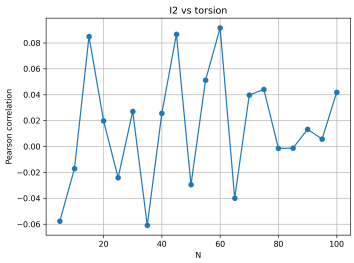


(c) H2_max vs Vol of convex hull

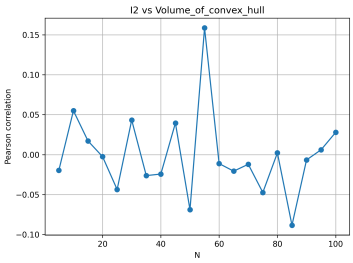
Figure 28: H2_max vs some geometric structures



(a) I2 vs radius of gyration



(b) I2 vs torsion



(c) I2 vs Vol of convex hull

Figure 29: I2 vs some geometric structures

4.1.2 Specifc PL knots

We aimed to study the correlations between the quantities derived from persistent homology and the geometric descriptors of specific types of knots.

Generation and Calculation To generate knots of prescribed types, we first produced millions of random polygonal knots using `Topoly`, and then classified them according to their Jones and Alexander polynomials, using `GenSpecificKnot.py`.

Since these two invariants are complete for knots with relatively small crossing numbers, we restricted our attention to simpler knot types, including the unknot (0_1), the trefoil (3_1), the figure-eight knot (4_1), the torus knot $K_{5,2}$ (5_1), and the knots 5_2, 6_1, 6_2, and 6_3.

We display the values of the average integral for the various knot types as a function of length, using `Compute_AvgBetti_Stats.py`.

Result In the first two figures, the relationship appears to be almost perfectly linear.

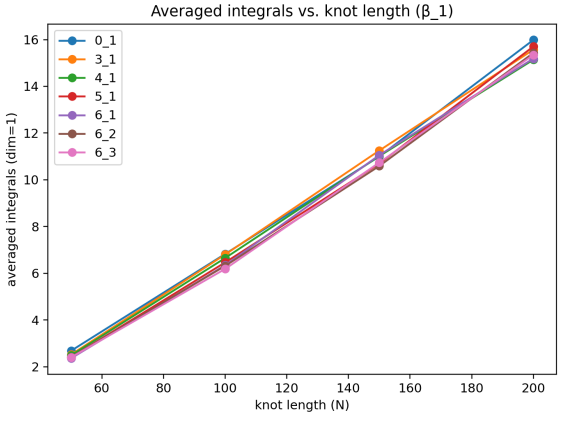


Figure 30: average integral(dim1)

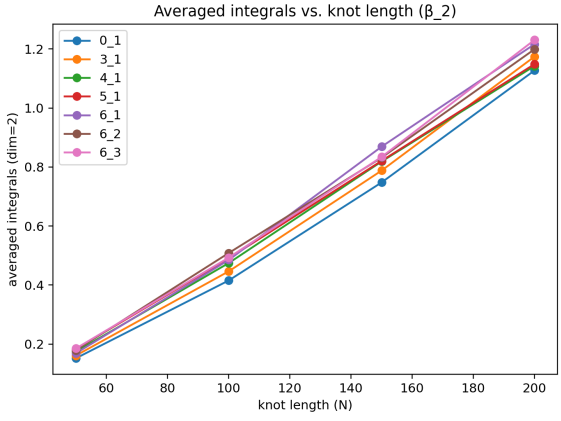


Figure 31: average integral(dim2)

However, the average maximum Betti numbers in dimensions 1 and 2 show a clear approximate trend, rather than an exact linear dependence.

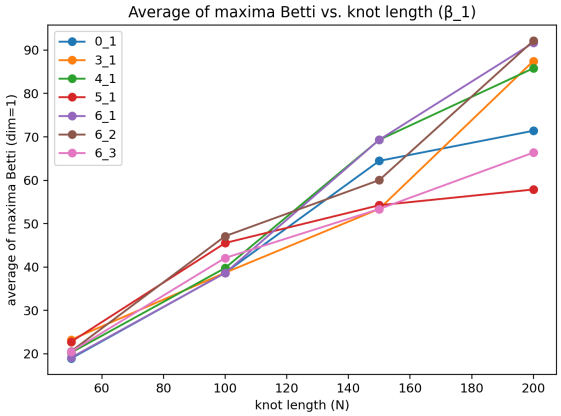


Figure 32: average of maxima Betti number(dim1)

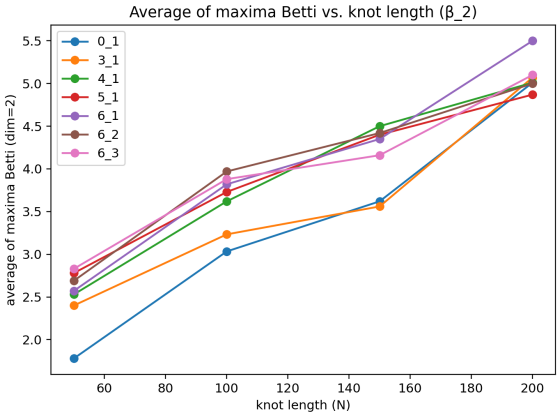


Figure 33: average of maxima Betti number(dim2)

4.2 Deviation

We may consider that for a given knot, 'how much' it deviates from being ideal.

Persistent homology Persistent homology can be effectively used to compute IR. To be more precisely, given a knot K, for small enough t, the homology of the embedded neighborhood is of rank 1 in dimension 1. The topology changes as soon as we get to $t = \mathrm{IR}(K)$, where the rank of the homology of $v_t(K)$ increases by one. Note that for values of t greater than $\mathrm{RS}(K)$, the radius of the circumscribing sphere, $H_1(v_t(K))$ vanishes, and the only non-trivial homology is of rank 1 in degree 0.

We use the alpha filtration on a point cloud $P(K)$ constructed from a PL knot embedding K to approximate its metric neighborhood for growing radii, and define a Betti curve on the rank of the homology of the alpha filtration on a point cloud. We compute the integrals of such Betti curves $\int_0^\infty \beta_1(P(K))\mathrm{d}t$ by summing the lengths of all bars in the corresponding barcode, denoted by $\mathcal{I}(K)$. Since $\beta_1(v_t(K))$ vanishes for $t \geq \mathrm{RS}(K)$, $\mathcal{I}(K)$ is well-defined.

Table 10: Notation

Symbol	Meaning
$\ell(K)$	length of K
$\mathrm{IR}(K)$	injectivity radius of K
$v_t(K)$	metric neighborhood of K of radius $t > 0$
$\mathrm{RS}(K)$	radius of the circumscribing sphere of K
$\mathcal{I}(K)$	the interval of Betti curves
$L/D(K)$	the length over diameter ratio of K

We can see that for two knots K_1, K_2 with the same length, if $\mathrm{IR}(K_1) \leq \mathrm{IR}(K_1)$, then $L/D(K_1) \leq L/D(K_2)$, that is, K_1 deviates less from being ideal.

Quantification We aim to quantify the deviation from the ideal configuration. For knots of fixed length, we observe that a larger injectivity radius corresponds to a smaller ratio, indicating that the embedding is closer to the ideal knot. It is important to note that our proposed measure captures the deviation of a given embedding from all possible ideal embeddings, rather than restricting the comparison to those within the same knot type.

Denote by $S(K) = \max\{t \geq 0 : \beta_1(K) \neq 0\}$, and let $\varepsilon > 0$ be a small and real number. Call

$$f_{R,\varepsilon} : [0, R] \to \mathbb{R}_{\geq 0}$$

the unique function by considering the linear function taking value 1 on 0, and value 0 on $R - \varepsilon$, and defined to be identically 0 after $R - \varepsilon$. Then define

$$\delta_\varepsilon(K) = \frac{1}{S(K)} \int_0^{S(K)} f_{S(K),\varepsilon}(t) \cdot \max\{\beta_1(K)(t) - 1, 0\} \mathrm{d}t.$$

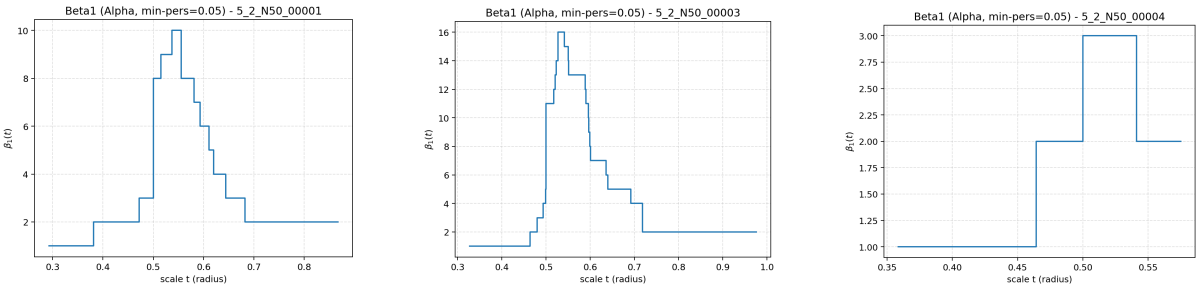
We claim that the $\delta_\varepsilon(K)$ defines a sensible quantification of the 'distance' between the given embedding and the ideal ones. For sufficiently small choice of the ε threshold, if $0 < \delta_\varepsilon(K_1) < \delta_\varepsilon(K_2)$, K_1 deviates less to the ideal ones than K_2.

Example 4.2.1. We input 100 fixed length PL knots with specific types, generated by `GenSpecifcKnot.py`, plot their Betti number curves and calculate $\delta_\varepsilon(K)$, by `deviation.py`.

Table 11 lists several representative values of the deviation measure $\delta_\varepsilon(K)$ computed for embeddings of the 5_2 knot with length $N = 50$, using $\varepsilon = 0.02$ and min_pers $= 0.05$.

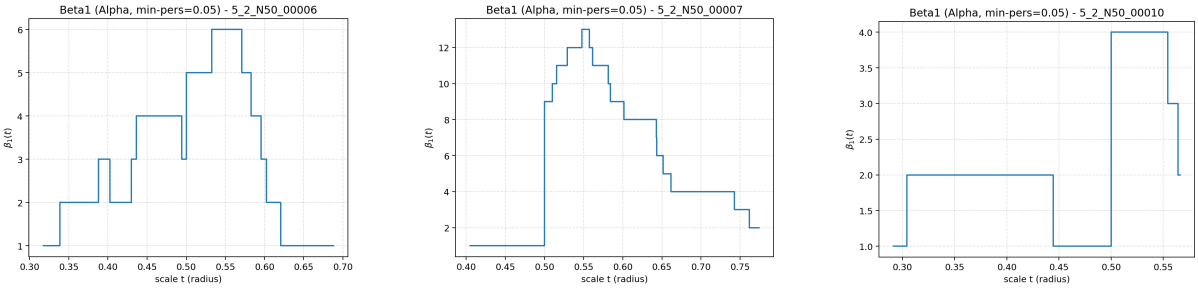
Table 11: Quantified deviation values $\delta_\varepsilon(K)$ for the 5_2 knot ($N = 50$, $\varepsilon = 0.02$, min_pers $= 0.05$).

Knot sample	**Label**	$\boldsymbol{\delta_\varepsilon(K)}$
1	5_2_N50_00001	0.3970
3	5_2_N50_00003	0.6683
4	5_2_N50_00004	0.0153
6	5_2_N50_00006	0.2279
7	5_2_N50_00007	0.4221
10	5_2_N50_00010	0.0405



(a) 5_2_N50_00001 (b) 5_2_N50_00003 (c) 5_2_N50_00004

Figure 34: Samples 1, 3, 4



(a) 5_2_N50_00006 (b) 5_2_N50_00007 (c) Knot-5_2_N50_00010

Figure 35: Samples 6, 7, 10

Although sample 7 attains the largest injectivity radius within the knot type 5_2—hence deviates least from the ideal knot in that class—sample 10 achieves the smallest δ_ε overall, indicating the least deviation from ideal knots across all knot types.

References

[CM22] Daniele Celoria and Barbara I. Mahler. "A statistical approach to knot confinement via persistent homology". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 478.2261 (2022). Received 6 September 2021; Accepted 8 April 2022, p. 20210709. ISSN: 1364-5021. DOI: 10.1098/rspa.2021.0709.

[CV22] Gunnar Carlsson and Mikael Vejdemo-Johansson. *Topological data analysis with applications*. Cambridge University Press, Cambridge, 2022, pp. xi+220. ISBN: 978-1-108-83865-8. DOI: 10.1017/9781108975704. URL: https://doi.org/10.1017/9781108975704.

[EH10] Herbert Edelsbrunner and John L. Harer. *Computational topology*. An introduction. American Mathematical Society, Providence, RI, 2010, pp. xii+241. ISBN: 978-0-8218-4925-5. DOI: 10.1090/mbk/069. URL: https://doi.org/10.1090/mbk/069.

[Oud15] Steve Y. Oudot. *Persistence theory: from quiver representations to data analysis*. Vol. 209. Mathematical Surveys and Monographs. American Mathematical Society, Providence, RI, 2015, pp. viii+218. ISBN: 978-1-4704-2545-6. DOI: 10.1090/surv/209. URL: https://doi.org/10.1090/surv/209.

[OY23] I. Obayashi and M. Yoshiwaki. "Field choice problem in persistent homology". In: *Discrete Comput. Geom.* 70.3 (2023), pp. 645–670. DOI: 10.1007/s00454-023-00544-7.

[PH15] J. A. Perea and J. Harer. "Sliding windows and persistence: an application of topological methods to signal analysis". In: *Found. Comput. Math.* 15.3 (2015), pp. 799–838. DOI: 10.1007/s10208-014-9206-z.