Delay Parameter Selection in Permutation Entropy Using Topological Data Analysis

A Preprint

Audun D. Myers
Department of Mechanical Engineering
Michigan State University
East Lansing, MI
myersau3@msu.edu

Firas A. Khasawneh
Department of Mechanical Engineering
Michigan State University
East Lansing, MI
khasawn3@egr.msu.edu

May 14, 2019

Abstract

Permutation Entropy (PE) is a powerful tool for quantifying the predictability of a sequence which includes measuring the regularity of a time series. Despite its successful application in a variety of scientific domains, PE requires a judicious choice of the delay parameter τ. While another parameter of interest in PE is the motif dimension n, Typically n is selected between 4 and 8 with 5 or 6 giving optimal results for the majority of systems. Therefore, in this work we focus solely on choosing the delay parameter. Selecting τ is often accomplished using trial and error guided by the expertise of domain scientists. However, in this paper, we show that persistent homology, the flag ship tool from Topological Data Analysis (TDA) toolset, provides an approach for the automatic selection of τ. We evaluate the successful identification of a suitable τ from our TDA-based approach by comparing our results to a variety of examples in published literature.

1 Introduction

Shannon entropy, which was introduced in 1948 [30], is a measurement of how much uncertainty there is in future data given the current dataset. Since the first introduction of information entropy, several new forms of entropy have been popularized. Some examples include approximate entropy [23], sample entropy [26], and permutation entropy (PE) [5]. While all of these methods measure the predictability of a sequence, PE also considers the order in which the data was received, which is critical for time series analysis. PE was first introduced by Bandt and Pompe [5] in 2002. Similar to Shannon Entropy, PE is quantified as the summation of the probabilities of a data type (see Eq. 1), where the data types for PE are motifs (see Fig. 1), which we represent as π. The PE parameters n and τ are used when selecting the motif size and spacing, respectively. More specifically, τ is the embedding delay lag applied to the series and n is a natural number that describes the dimension of the motif. In this study we focus on selecting τ since the dimension is typically chosen in the range $3 < n \leq 7$ for most applications [27].

Currently, the most common method for selecting PE parameters is to adopt the values suggested by domain scientists. For example, Li et al. [17] suggest using $\tau = 3$ and $n = 3$ for electroencephalographic (EEG) data, Zhang and Liu [33] suggest $\tau = 3$ or 5 and $n \in [3, 5]$ for logistic maps, and Frank et al. [9] suggest $\tau = 2$ or 3 and $n \in [3, 7]$ for heart rate applications. One main disadvantage of using suggested parameter values for an application is the high dependence of PE on the sampling frequency. As an example, Popov et al. [24] showed the importance of considering the sampling frequency when selecting τ for an EEG signal. Another limitation is the need for application expertise in order to determine the needed parameters. This can hinder using PE in new applications that have not been sufficiently explored. Consequently, there is a need for an automatic, application-independent parameter selection algorithm for PE.

To some extent, some studies have attempted to find robust and application-independent methods for determining τ. Many of these methods are based on phase space reconstruction using Takens embedding [31], which has corresponding parameters. Some of the common approaches for determining τ are mutual information [10], autocorrelation [12], and phase space methods [6]. However, due to the origin of these methods in phase space reconstruction and not PE, they may result in a poor estimate of τ.

In this paper, we present two novel approaches for finding τ based on tools from Topological Data Analysis (TDA). Specifically in the first approach we compute the 0-D sublevel set persistence of the Fourier spectrum to obtain a summary of how the different components in the spectrum are connected. We then use the modified z-score from statistics to acquire a threshold for a cutoff frequency which captures the most dominant frequencies in the spectrum in an automatic way. We then utilize Nyquist's sampling theorem to find an appropriate τ value. Our second approach utilizes sliding window and 1-D persistence scoring (SW1PerS) [21] to measure how periodic or significant the circular shape is for the embedded point cloud. Using SW1Pers, we search for the window size that maximizes a periodicity score, and then use the resulting window size to suggest an embedding delay for PE. To determine the viability of our two methods, PE parameters generated based on them are compared to expert suggested parameters.

The paper is organized as follows. In Section 1.1 we describe PE and show its computation using a simple example. Section 2 provides an overview of the tools that we use from TDA. Section 3.2 describes our first approach for finding τ which is a frequency based approach that applies 0-D sublevel set persistence and the modified z-score to the Fourier spectrum. Section 3.3 details the second TDA approach for finding τ which is a time-domain approach that combines sliding window embedding of the time series with 1-D persistence. The results are then presented in Section 4, and the concluding remarks are made in Section 5.

1.1 Permutation Entropy Example

Permutation entropy $H(n)$ for motif dimension n is calculated according to [5]

$$H(n) = -\sum p(\pi_i) \log p(\pi_i), \tag{1}$$

where $p(\pi_i)$ is the probability of a permutation π_i, and $H(n)$ has units of bits when the logarithm is of base 2. The permutation entropy parameters τ and n are used when selecting the motif size: τ is the number of time steps between two consecutive points in a uniformly sub-sampled time series, and n is the permutation length or motif dimension. Using a set X and an element of the set $x_i \in X$, we can define the vector $v_i = [x_i, x_{i+\tau}, x_{i+2\tau}, \ldots, x_{i+(n-1)\tau}]$, which has the permutation π_i. To better understand the possible permutations, consider an example with third degree ($n = 3$) permutations. This results in six possible motifs as shown in Fig. 1.

Next, to further demonstrate PE with an example, consider the sequence $X = [4, 7, 9, 10, 6, 11, 3]$ with third order permutations $n = 3$ and time delay $\tau = 1$. The sequence can be broken down into the following permutations: two $(0, 1, 2)$ permutations, one $(1, 0, 2)$ permutation, and two $(1, 2, 0)$ permutations for a total of 5 permutations. Applying Eq. (1) yields

$$H(3) = -\frac{2}{5} \log \frac{2}{5} - \frac{2}{5} \log \frac{2}{5} - \frac{1}{5} \log \frac{1}{5} = 1.522 \text{ bits}.$$
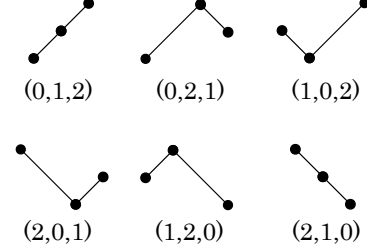

Figure 1: All possible permutation configurations (motifs) for n = 3, where $[\pi_1 \ldots \pi_6] = [(0, 1, 2) \ldots (2, 1, 0)]$.

The permutation distribution can be visually understood by illustrating the probabilities of each permutation as separate bins. To accomplish this, Fig. 2 was created by taking the same series X (Fig. 2 a) and placing the abundance of each permutation into its respective bin (Fig. 2 b). PE is at a
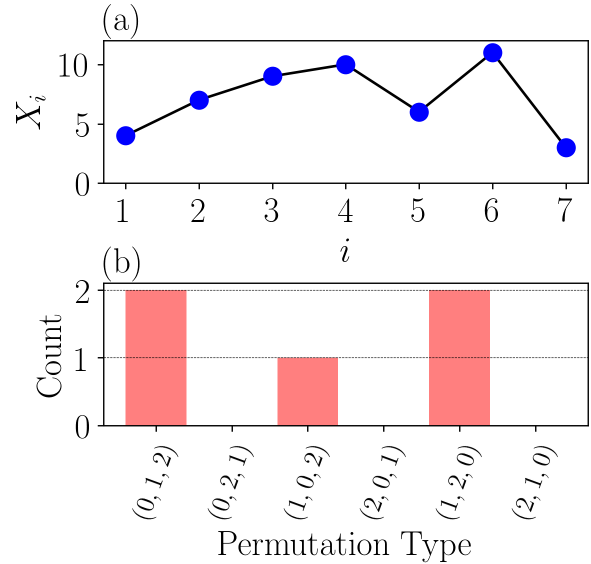

Figure 2: Abundance of each permutation from example dataset X.

maximum when all $n!$ possible permutations are evenly distributed or, equivalently, when the permutations are equiprobable with $p(\pi_i) = p(\pi_1), p(\pi_2), \ldots, p(\pi_{n!}) = \frac{1}{n!}$. From this, the maximum permutation entropy $H_{\max}$ is quantified as

$$H_{\max}(n) = -\sum p(\pi_i) \log p(\pi_i)$$
$$= -\log \frac{1}{n!} = \log n!. \tag{2}$$

Applying Eq. (2) for $n = 3$ yields a maximum PE of approximately 2.585 bits. Using the maximum possible entropy $\log_2 n!$, the normalized permutation entropy is calculated as

$$h_n = -\frac{1}{\log_2 n!} \sum p(\pi_i) \log_2 p(\pi_i). \tag{3}$$

Applying Eq. (3) to the example series X results in $h_3 \approx 0.5888$.

2 An overview of tools from TDA

Our TDA-based approaches for finding the delay dimension employ two types of persistence applied to two different types of data. Specifically, in the first approach we combine the 0-D sublevel persistence of one-dimensional time series with the z-score, while in the second approach we utilize 1-D persistent homology on embedded time series as part of the SW1PerS framework in Section 3.3. This section provides a basic background of the topics needed to at least intuitively understand the subsequent analysis. More specifics can be found in [7, 8, 11, 20].

2.1 Simplicial complexes

An abstract k-simplex σ is defined as a set of $k + 1$ indices where $\dim(\sigma) = k$. If we apply a geometric interpretation to a k-simplex, we can think of it as a set V of $k + 1$ vertices. Using this interpretation, a 0-simplex is a point, a 1-simplex is an edge, a 2-simplex is a triangle, and higher dimensional versions can be similarly obtained.

A simplicial complex K is a set of simplices $\sigma \subseteq V$ such that for every $\sigma \in K$, all the faces of σ, i.e., all the lower dimensional component simplices $\sigma' \subset \sigma$ are also in K. For example, if a triangle (2-simplex) is in a simplicial complex K, then so are the edges of the triangle (1-simplices) as well as all the nodes in the triangle (0-simplices). The dimension of the resulting simplicial complex is given by the largest dimension of its simplices according to $\dim(K) = \max_{\sigma \in K} \dim(\sigma)$. The n-skeleton of a simplicial complex $K^{(n)}$ is the restriction of the latter to its simplices of degree at most n, i.e., $K^{(n)} = \{\sigma \in K \mid \dim(\sigma) \leq n\}$.

Given an undirected graph $G = (V, E)$ where V are the vertices and E are the edges, we can construct the clique (or flag) complex

$$K(G) = \{\sigma \subseteq V \mid uv \in E \text{ for all } u \neq v \in \sigma\}.$$

2.2 Homology

If we fix a simplicial complex K, then homology groups can be used to quantify the holes of the structure in different dimensions. For example, in dimension 0, the rank of the 0 dimensional homology group $H_0(K)$ is the number of connected components. The rank of the 1-dimensional homology group $H_1(K)$ is the number of loops, while the rank of $H_2(K)$ is the number of voids, and so on. The homology groups are constructed using linear transformations termed boundary operators.

To describe boundary operators, we first define the oriented simplicial complex as the simplicial complex obtained by an ordered set of vertices $\sigma = [v_1, \ldots, v_k]$. Permuting two indices in σ gives the opposite simplex according to

$[v_1, \ldots, v_i, \ldots, v_j, \ldots, v_k] = -[v_1, \ldots, v_j, \ldots, v_i, \ldots, v_k]$. Now let $\{\alpha_\sigma\}$ be coefficients in a field F (in this paper we choose $F = \mathbb{Z}_2$). Then $K^{(n)}$, the n-skeleton of K, can be used as a generating set of the F-vector space $\Delta_n(K)$. In this representation, any element of $\Delta_n(K)$ can be written as a linear combination $\sum_{\sigma \in K^{(n)}} \alpha_\sigma \sigma$. Further, elements in $\Delta_n(K)$ are added by adding their coefficients. A finite formal sum of the n-simplices in K is called an n-chain, and the group of all n-chains is the nth chain group $\Delta_n(K)$, which is a vector space.

Given a simplicial complex K, the boundary map $\partial_n : \Delta_n(K) \to \Delta_{n-1}(K)$ is defined by

$$\partial_n([v_0, \ldots, v_n]) = \sum_{i=0}^{n} (-1)^i [v_0, \ldots, \hat{v}_i, \ldots, v_n],$$

where $\hat{v}_i$ denotes the absence of element v_i from the set. This linear transformation maps any n-simplex to the sum of its codimension 1 (codim-1) faces. The geometric interpretation of the boundary operator is that it yields the orientation-preserved boundary of a chain.

By combining boundary operators, we obtain the chain complex

$$\ldots \xrightarrow{\partial_{n+1}} \Delta_n(K) \xrightarrow{\partial_n} \ldots \xrightarrow{\partial_1} \Delta_1(K) \xrightarrow{\partial_0} 0,$$

where the composition of any two subsequent boundary operators is zero, i.e., $\partial_n \circ \partial_{n+1} = 0$. A n-chain $\alpha \in \Delta_n(K)$ is a cycle if $\partial_n(\alpha) = 0$; it is a boundary if there is an $n + 1$-chain β such that $\partial_{n+1}(\beta) = \alpha$. Define the kernel of the boundary map ∂_n using $Z_n(K) = \{c \in \Delta_n(K) : \partial_n c = 0\}$, and the image of ∂_{n+1} $B_n(K) = \{c \in \Delta_n(K) : c = \delta_{n+1} c', c' \in \Delta_{n+1}(K)\}$. Consequently, we have $B_k(K) \subseteq Z_k(K)$. Therefore, we define the nth homology group of K as the quotient group $H_n(K) = Z_n(K)/B_n(K)$. In this paper, we only need 0- and 1-dimensional persistent homology, and we always assume homology with $\mathbb{Z}_2$ coefficients which removes the need to keep track of orientation. In the case of 0-dimensional homology, there is a unique class in $H_0(K)$ for each connected component of K. For 1-dimensional homology, there is one homology class in $H_1(K)$ for each *hole* in the complex.

2.3 Filtration of a simplicial complex

Now, we are interested in studying the structure of a changing simplicial complex. We introduce a real-valued filtration function on the simplicies of K such that $f(\tau) \leq f(\sigma)$ for all $\tau \leq \sigma$ simplices in K. If we let $\{y_1 < y_2 < \ldots < y_\ell\}$ be the set of the sorted range of f for any $y \in \mathbb{R}$, then the filtration of K with respect to f is the ordered sequences of its subcomplexes

$$\emptyset \subseteq K(y_1) \subseteq K(y_2) \subseteq \cdots \subseteq K(y_\ell) = K.$$

The sublevel set of K corresponding to y is defined as

$$K(y) = \{\sigma \in K \mid f(\sigma) \leq y\}, \tag{4}$$

where each of the resulting $K(y)$ is a simplicial complex, and for any $y_1 \leq y_2$, we have $K(y_1) \subseteq K(y_2)$.

The filtration of K enables the investigation of the topological space under multiple scales of the output value of the filtration function f. In this paper we consider two different filtration functions where each of these functions is applied to a different type of data: 0-D persistence applied to 1-D time series, and 1-D persistence applied to point clouds embedding in $\mathbb{R}^n$.

0-D persistence applied to 1-D time series: Let χ be the time-ordered set of the critical values of a time series. Here, we can think of the simplicial complex $K = G(V, E)$ containing a number of vertices $|V|$ equal to the number of critical values in the time series and only the edges E that connect adjacent vertices. i.e., vertices $\{v_i \mid 1 \leq i \leq n\}$, and edges $\{v_i v_{i+1} \mid 1 \leq i \leq n-1\}$. Therefore, we have a one-to-one correspondence between the critical values in χ and the vertices of the simplicial complex V. We define the filtration function for every face σ in K according to

$$f(\sigma_i) = \begin{cases} \chi_i & \text{if } \sigma_i \in V, \\ \max(u, v) & \text{if } \sigma_i = uv \in E. \end{cases}$$

Using this filtration function in Eq. (4), we can define an ordered sequence of subcomplexes where $y \in [\min(\chi), \max(\chi)]$.

1-D persistence applied to point clouds in $\mathbb{R}^m$: The second set of data points we work with is a point cloud $P \subset \mathbb{R}^m$. There are several methods to embed a time series; however, here we embed the time series using delay reconstruction. For each point $p \in P$, let $B(p, r)$ be the ball centered at p and of radius r. Now, for some radius r, we can build a simplicial complex where, for example, the intersection of any two balls adds an edge, the intersection of three balls adds a triangle, and higher dimensional analogs are added similarly, see the construction for r_1 through r_4 in Fig. 4. The result of the union of all the balls $\cup_{p \in P} B(p, r)$ is called the Čech complex; although in practice we work with Rips complexes which are the clique complexes of the function f and they are easier to obtain computationally while still capturing the interesting features of the space. The filtration function in this case is given by

$$f(\sigma) = \max_{\{u, v\} \subseteq \sigma} d(u, v),$$

where $d(u, v)$ is the distance between vertices u and v, and the distance between a vertex and itself is zero.

2.4 Persistent homology

Persistent homology is a tool from topological data analysis which can be used to quantify the shape of data. The main idea behind persistent homology is to watch how the homology changes over the course of a given filtration.

Fix a dimension n, then for a given filtration

$$K_1 \subseteq K_2 \subseteq \cdots \subseteq K_N$$

we have a sequence of maps on the homology

$$H_n(K_1) \rightarrow H_n(K_2) \rightarrow \cdots \rightarrow H_n(K_N).$$

We say that a class $[\alpha] \in H_n(K_i)$ is born at i if it is not in the image of the map $H_n(K_{i-1}) \rightarrow H_n(K_i)$. The same class dies at j if $[\alpha] \neq 0$ in $H_n(K_{j-1})$ but $[\alpha] = 0$ in $H_n(K_j)$.

This information can be used to construct a persistence diagram as follows. A class that is born at i and dies at j is represented by a point in $\mathbb{R}^2$ at (i, j). The collection of the points in the persistence diagram, therefore, give a summary of the topological features that persists over the defined filtration. See the example of Fig. 3 for $n = 0$ and time series data, and Fig. 4 for $n = 1$ and point cloud data.
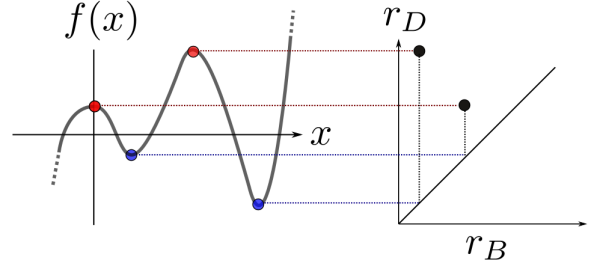

Figure 3: Example formulation of a persistence diagram based on 0-D sublevel sets.

3 Embedding Delay Parameter Selection

Delay embedding is a uniform subsampling of the original time series according to the embedding parameter τ. For example, the subsampled sequence X with elements $\{x_i : i \in \mathbb{N} \cup 0\}$ subject to the delay τ is defined as $X(\tau) = [x_0, x_\tau, x_{2\tau}, \ldots]$. Riedl et al. [27] showed that PE is sensitive to the time delay, which prompts the need for a robust method for determining an appropriate value for τ. For estimating the optimal τ, we will be investigating the following methods in the subsequent sections: Mutual Information (MI) in Section 3.1, combining Fourier analysis with 0-D persistence and the modified z-score (Section 3.2), and Sliding Window for One-Dimensional Persistence Scoring (SW1PerS) (Section 3.3). We recognize, but do not investigate, some other commonly used methods for finding τ. These include the autocorrelation function [12] and the phase space expansion [6].

3.1 Mutual Information

Mutual information is a measurement of how much information is shared between two sequences, and it was first realized by Shannon et al. [29] as

$$I(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}, \qquad (5)$$

where X and Y are separate sequences, $p(x)$ and $p(y)$ are the probability of the element x and y separately, and $p(x, y)$ is the joint probability of x and y. Fraser and Swinney [10] showed that for a chaotic time series the mutual information between the sequence $x(t)$ and $x(t + \tau)$
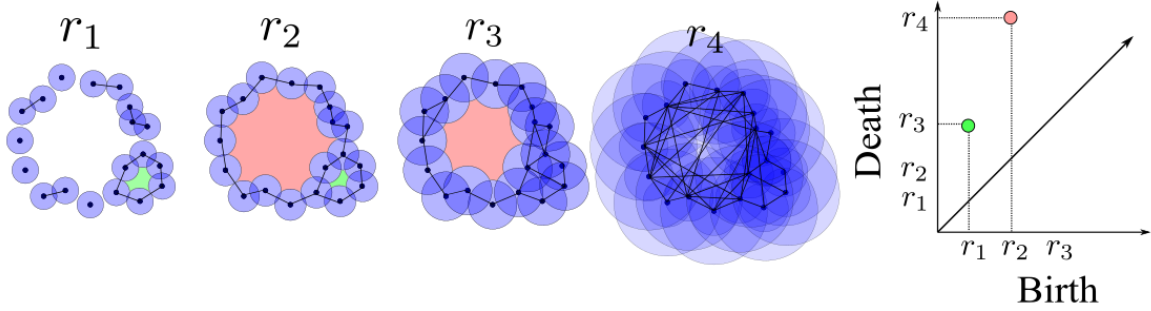
Figure 4: Example persistence diagram (right most figure) for 1-D persistent homology. The first 1-D rips complex or topological feature is born at radius r_1 and dies at r_3, while the second complex is born at r_2 and dies at a much larger radius r_4.

will decrease as τ increases until reaching a minimum. At this delay τ, the individual data points share a minimum amount of information, thus indicating that the data points are sufficiently separated. While this delay value was specifically developed for phase space reconstruction from a single time series, it is also commonly used for the PE parameter τ. We would like to point out that in general there is no guarantee that minima exist for the mutual information function, which is a serious limitation for computing τ using this method.

3.2 Fourier Spectrum Noise Cutoff: 0-D Persistence and Modified z-score

In this section we present a novel topological data analysis (TDA) based approach for finding the noise floor in the Fourier spectrum. Specifically, we show how the 0-dimensional sublevel persistence, a tool from TDA discussed in Section 2, can be used to find the lifetime of the maxima and minima in a signal. We describe a fast and simple algorithm (Algorithm 1) for computing the lifetimes using the peaks and valleys of the signal. We then separate the noise lifetimes from significant lifetimes through the use of the modified z-score, which allows us to find the noise floor and frequency cutoff. This frequency cutoff will later be used to find an embedding delay τ for PE. The process for finding the cutoff frequency is summarized in Fig. 5. The following paragraphs give an overview of the modified z-score and the threshold and cutoff analysis.

Modified z-score The modified z-score z_m is essential to understanding the techniques used for isolating noise from a signal [28]. The standard score, commonly known as the z-score, uses the mean and the standard deviation of a dataset to find an associated z-score for each data point and is defined as

$$z = \frac{x - \mu}{\sigma},\qquad(6)$$

where x is the dataset, μ is the mean, and σ is the standard deviation of the dataset, respectively. The z-score value is commonly used to identify outliers in the dataset by rejecting points that are above a set threshold, which is set in terms of how many standard deviations away from

the mean are acceptable. Unfortunately, the z-score is susceptible to outliers itself because both the mean and the standard deviation are not robust against outliers [16]. This led Hampel [13] to develop the modified z-score as an outlier detection method that is insensitive to outliers. The logic behind the modified z-score or median absolute deviation (MAD) method is grounded on the use of the median for the mean. The MAD is calculated as

$$\text{MAD} = \text{median}(|x - \tilde{x}|),\qquad(7)$$

where x is the dataset and $\tilde{x}$ is the median of the dataset. The MAD is substituted for the standard deviation in Eq. (6). To complete the modified z-score, Iglewicz and Hoaglin [14] suggested substituting the mean with the median. The resulting equation for the modified z-score is

$$z_m = 0.6745 \frac{x - \tilde{x}}{\text{MAD}}.\qquad(8)$$

Using z_m allows for outlier detection with a breakdown point of 50% outliers. A threshold for separating noise in the persistence domain is discussed in the following paragraph.

Threshold and Cutoff Analysis To determine the noise floor in the normalized Fast Fourier Transform (FFT) spectrum, we compute the 0-dimensional persistence of the FFT. This provides relatively short lifetimes for the GWN, while the prominent peaks, which represent the actual signal, have comparatively long lifetimes or high persistence. To separate the noise from the outliers, we apply the modified z-score for each lifetime in the persistence diagram. We determine the modified z-score threshold based on the need to account for approximately all of the noise in the FFT. To do this, we use a signal of pure GWN with a standard deviation of 1.0 and length of 10^5. Next, we calculate the FFT of the GWN signal and its 0-D sublevel set persistence to get a set of lifetimes from the persistence diagram. We then apply modified z-score thresholds ranging from 0 to 5 on the lifetimes from the persistence diagram to determine the percent of noise as a function of the threshold. By setting a threshold requirement that accounts for approximately all persistence diagram noise, while not being excessively large, a threshold of approximately 4.8 was found as shown in Fig. 6. This threshold was rounded up to 5 for simplicity.
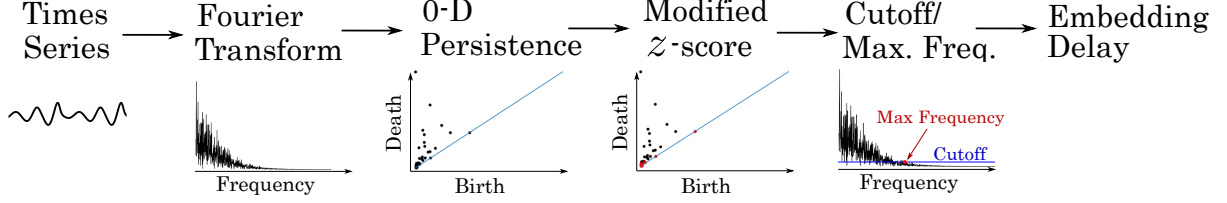
Figure 5: Overview of procedure for finding maximum significant frequency using 0-dimensional sublevel set persistence and the modified z-score for a signal contaminated with GWN.

With a suitable modifed z-score threshold, our next goal in finding a the noise floor cutoff begins by investigating the distribution of the FFT of GWN in the persistence domain. To do this, the same GWN signal is analyzed. Again, the FFT and the 0-D persistence were calculated to determine the lifetimes as well as the births and deaths of the persistence diagram. Figure 7 shows the histogram obtained from the resulting persistence diagram.
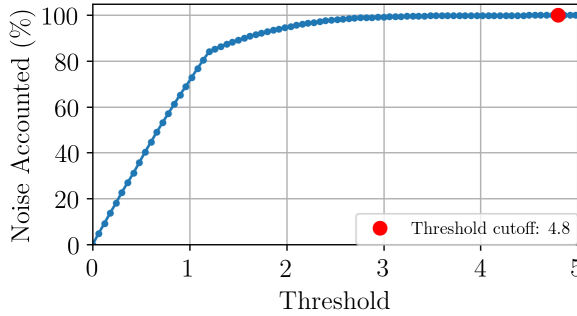

Figure 6: Percent of the persistence points from 0-D sublevel set persistence of the FFT of GWN using the modified z-score with the provided threshold ranging from 0 to 5.

Although there have been studies on pushing forward probability distributions into the persistence domain [1, 2, 15], it is difficult to obtain a theoretical cutoff value in persistence space. Therefore, we proceed by computing the histograms for the 0-D sublevel set and establishing a heuristic cutoff value that seems to perform well. The theoretical distribution of GWN in the Fourier spectrum has a Rayleigh distribution [25]. Approximately the same distribution appears in the histogram of noise for the birth times of the persistence diagram with the following differences: the death times tend to show a slightly shifted normal distribution, while the lifetimes show approximately a folded-normal distribution. Without doing an in depth statistical analysis of the distributions, we calculate a suitable cutoff using the median of the births and maximum of the lifetimes as

$$\text{Cutoff}_{\text{noise}} = \max\left(\text{lifetime}_{\text{noise}}\right) + \text{median}\left(\text{birth}_{\text{noise}}\right),$$
(9)

where $\text{Cutoff}_{\text{noise}}$ is the suggested noise cutoff and $\text{birth}_{\text{noise}}$ and $\text{lifetime}_{\text{noise}}$ is the set of the births and lifetimes that were considered noise from the modified z-scores.
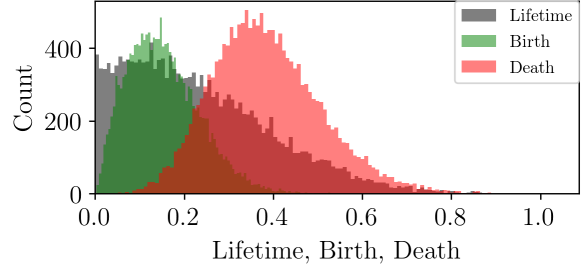

Figure 7: Distributions of the lifetimes, deaths, and births of 0-D sublevel set persistence for the FFT of GWN with a length of 10^5, mean $\mu = 0.0$, and standard deviation $\sigma = 1.0$.

We then find the maximum significant frequency $f_{\max}$ as the highest frequency in the Fourier spectrum with an amplitude greater than the specified cutoff from Eq. (9). Using $f_{\max}$, we suggest a delay following Nyquist's sampling rate as $\tau = \frac{f_s}{2f_{\max}}$, where f_s is the sampling frequency.

3.3 Finding τ using SW1Pers

Perea and Harer [22] developed Sliding Window for One-Dimensional Persistence Scoring (SW1PerS) as a TDA method for determining the most significant period of a time series. SW1PerS is also useful for finding the window in which a chaotic time series has the highest periodicity. In this section we develop an algorithm that combines the period of the main oscillation derived from SW1PerS with frequency criterion developed by [18] to determine a suitable delay.

SW1PerS uses 1-D persistent homology from Section 2 to measure how periodic or significant the circular shape in an embedded dimension (point cloud) is as the embedding window size increases. The sliding window SW is defined as

$$SW_{m,\tau_s} f(t) = [f(t_0), f(t_0 + \tau_s), \dots, f(t_0 + m\tau_s)], \quad (10)$$

where $f(t)$ is the time series being analyzed and τ_s and m are, respectively, SW1PerS embedding delay and dimension. Applying Eq. (10) across the range of the time series n_T times results in a collection of vectors known as a set of point clouds, which live in an m-dimensional Euclidean space. The number of sliding windows n_T was set to 200 to be sufficiently high for forming 1-D topological shapes or

simplices. Specifically, we first linearly map the entire time series onto the domain $[0, 2\pi]$. Next, we set the desired window size as

$$w = \frac{2\pi m}{L(m+1)}, \qquad (11)$$

where L is the division parameter that segments the entire time series into seperate windows of size w. For our analysis, we varied the L parameter from Eq. (11) to find the window size which yielded the highest periodicity. Figure 8 shows an example sliding window using the Lorenz system described in Section A.2 with $\rho = 28$. The
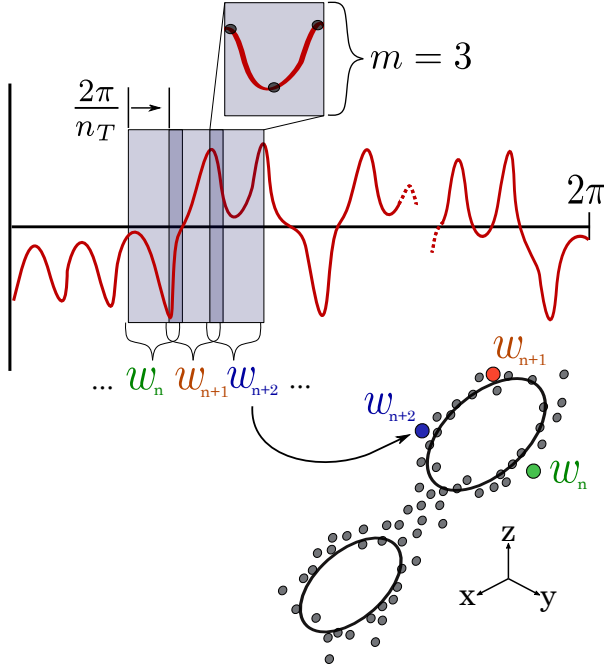

Figure 8: SW1PerS applied on Lorenz example defined in Section A.2 showing sliding window for time series linearly scaled to 2π and embedded in dimension $m = 3$.

sliding window example uses the scaled signal with an embedding dimension $m = 3$. Normally, m is determined based on the theory developed by Perea et al. [22], which showed the necessary value of m for reconstruction is bounded by $m \geq 2N$, where N is the number of Fourier terms necessary for reconstructing the signal. For our application, we will also set $m = 2N$. Perea et al. [21] did not choose N based on the data, but rather picked a large number assuming it would be sufficient to faithfully represent the data. In contrast, in this work we automate choosing N by approximating the Fourier series using the discrete Fourier transform, and then computing the ℓ_2 norm between the approximation and the signal to obtain the value of N that yields an error within a desired threshold, see Eq. (13). Specifically, if we let the time series X be a discrete time sampling of a piecewise smooth signal $x(t)$, then the N-partial sum of the Fourier series of $x(t)$ can be

approximated according to

$$f_N(t) = \frac{1}{|X|} \sum_{k=0}^{N} \Big(\sum_{j=0}^{|X|} X(j) e^{-2\pi i j k / T} \Big) e^{2\pi i k t / T}, \qquad (12)$$

where $|X|$ is the length of the time series X. As a rule of thumb, $N \approx |X|/8$ yields an accurate reconstruction of $x(t)$ [4]. The relative ℓ_2 norm that measures the error between time series X and its reconstruction $f_N(t)$ is given by

$$\ell_2(N) = \frac{\left(\sum_{j=0}^{|X|} \left[X(j) - f_N(j) \right]^2 \right)^{1/2}}{\left(\sum_{j=0}^{|X|} X^2(j) \right)^{1/2}}. \qquad (13)$$

For our application, we cosider $f_N(t)$ as sufficiently close to $x(t)$ when we find a value of N for which $\ell_2(N) < 0.10$. Using m determined as $m = 2N$, we then calculate the sliding window delay τ_s based on the relationship $\tau_s = w/m$. To apply this desired time delay, we interpolate the time series using a cubic spline fit. We can then determine the window size that provides the greatest periodicity using the periodicity score

$$s = 1 - \frac{r_B^2 - r_D^2}{3}, \qquad (14)$$

Equation (15) normalizes the scores so that $s = 1$ corresponds to no periodicity and $s = 0$ corresponds to perfectly periodic. We determine the period P of the time series using the L parameter corresponding to the lowest periodicity score s to calculate P as

$$P = \frac{mT}{(m+1)L_P}, \qquad (15)$$

where T is the time span of the original time series and L_P is the L value corresponding to the the lowest periodicity score s. To find L_P, L was incremented from 1 to $L_{\max}$, where $L_{\max}$ is the maximum number of periods to consider in the time series. Figure 9 demonstrates how the shape of the data changes with the time delay τ_s, which changes the window size w. Additionally, Fig. 9 shows how a minimum periodicity score occurs when the embedded data forms a 1-D simplex with the longest lifetime. Using the period of the time series P, we compute a suggested embedding delay according to

$$\tau = \frac{f_s P}{\alpha}, \qquad (16)$$

where f_s is the original sampling frequency and $\alpha \in [2, 4]$ is based on the frequency criteria from [18]. For this paper we set $\alpha = 3$.

4 Results and Discussion

To verify our TDA-based methods for determining τ, Table 1 compares our results to the values suggested by experts using a variety of systems including the ones listed by Riedl et al. [27]. The table highlights the automatically computed τ which best matches the expert-identified values for τ.
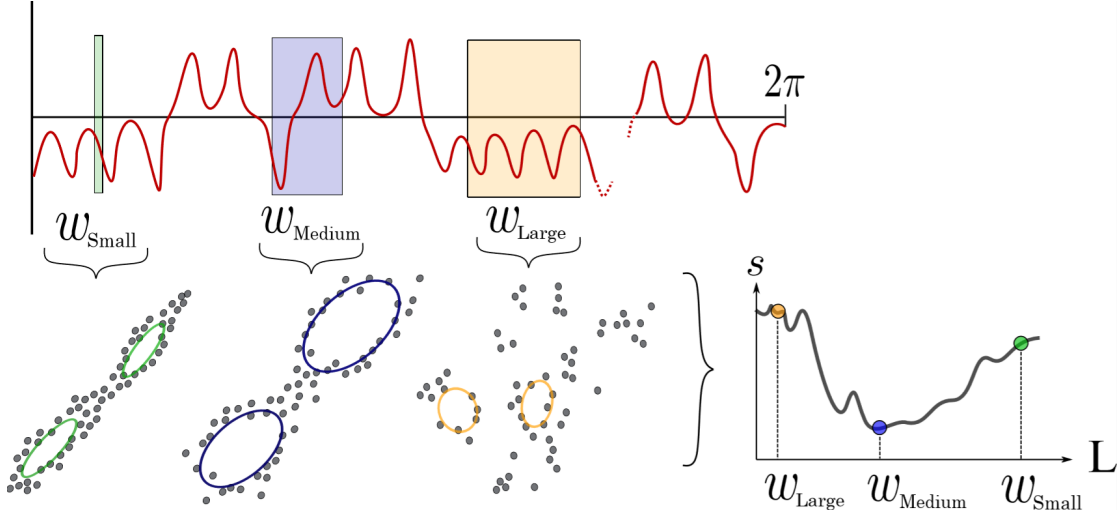
Figure 9: Example showing the shape of the embedded time series as a function of the window size: (left) w_{Small} is too small revealing elliptical shape and high periodicity score s, (middle) w_{Medium} is properly sized and results in a minimum periodicity score s, and (right) w_{Medium} is too large and results in a high periodicity score s.

The tables shows that for GWN, all the considered methods accurately calculate an embedding delay of 1. However, none of the methods gave reliable results for both difference equations (Henon and the logistic map). Specifically, the frequency approach based on 0-D persistence provided the best results with $\tau = 1$, while the MI method yielded the best result for the Logistic map. While the 0-D persistence approach slightly underestimated τ for these systems, we do not suggest using SW1PerS for difference equations since it results in considerably inaccurate suggestions for the delay parameter.

Although Table 1 shows that there is no one method that performs well for all types of signals, in the following we provide recommendations for which method to use with chaotic signals, periodic signals, and EEG/ECG data. For chaotic differential equations, we suggest using the delay calculated using the frequency approach with a 0-D-persistence noise filter as this method consistently provided the most accurate time delays. For periodic systems, we suggest using the τ found using either SW1PerS or the 0-D persistence approach, but to avoid using the MI results. For EEG/ECG data, we suggest using either SW1PerS or the 0-D persistence approach as they consistently provide more accurate values of τ in comparison to the MI method.

5 Conclusion

In this paper we describe two novel TDA-based methods for automatically determining the PE delay parameter τ given a sufficiently sampled/oversampled time series. One of the methods is a frequency domain approach based on the modified z-score and the 0-D sublevel set persistence. The other methods is based on SW1PerS, and it operates on a sliding window embedding of the time-domain signal.

Both of these methods were compared to the standard MI approach of Fraser and Swinney [10].

For the frequency approach, we developed an automatic algorithm for finding the maximum significant frequency using a cutoff greater than the additive Gaussian noise floor. We find the noise floor using 0-D sublevel set persistence with noise separated using the modified z-score.

We applied these methods to various categories including difference equation, chaotic differential equations, periodic systems, EEG/ECG data, and Gaussian noise. We then compared the generated parameters to values suggested by experts to determine which methods consistently found accurate values for τ, see Table 1. We showed that no single method provides accurate values of τ for every application, but each method showed better performance depending on the type of the studied system.

Specifically, for pure white noise, all the methods produced a viable estimate for τ. However for chaotic systems, the 0-D persistence approach yielded the most consistent results. For periodic time series, both TDA-based method outperforms their MI counterparts.

For EEG/ECG data, the lowest bound for either SW1PerS or the 0-D gave better estimates for τ in comparison to the MI approach. However, for difference equations, no one method consistently provided accurate values for τ with the SW1PerS approach giving large overestimates. For these systems, although the frequency approach based on 0-D persistence underestimated τ, it came closest to providing the best results for Hénon map. On the other hand, the MI method yielded the best result for the Logistic map. Therefore, if TDA-methods are used for finding the delay parameter for difference equations, we recommend avoiding the SW1PerS approach in favor of the 0-D persistence approach.

Table 1: A comparison between the calculated and suggested values for the delay parameter τ. The shaded cells highlight the methods that yielded the closest match to the suggested delay.

Catagory	System	Method			Suggested Delay τ	Ref.
		MI	SW1PerS	0-D Persistence of FFT		
Noise	White Noise	1	1	1	1	[27]
Chaotic Differential Equation	Lorenz	10	9	8	10	[27]
	Rossler	13	22	7	9	[32]
	Bi-directional Rossler	11	7	12	15	[27]
	Mackey-Glass	7	10	3	1-to 700	[27]
Periodic System	Sine Wave	2	22	21	20	[32]
Nonlinear Difference Eq.	Logistic Map	4	15	1	1 to 5	[33]
	Henon Map	9	90	1	1 to 2	[27]
Medical Data	ECG	5	11	12	1 to 4	[27]
	EEG	6	2	5	1 to 3	[27]

Acknowledgment

References

[1] Robert J. Adler, Omer Bobrowski, Matthew S. Borman, Eliran Subag, and Shmuel Weinberger. Persistent homology for random fields and complexes. In *Institute of Mathematical Statistics Collections*, pages 124–143. Institute of Mathematical Statistics, 2010.

[2] Robert J. Adler, Omer Bobrowski, and Shmuel Weinberger. Crackle: The homology of noise. *Discrete & Computational Geometry*, 52(4):680–704, aug 2014.

[3] Ralph G Andrzejak, Klaus Lehnertz, Florian Mormann, Christoph Rieke, Peter David, and Christian E Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, 64(6):061907, 2001.

[4] Nakhlé H Asmar. *Partial differential equations with Fourier series and boundary value problems*. Courier Dover Publications, 2016.

[5] Christoph Bandt and Bernd Pompe. Permutation entropy: a natural complexity measure for time series. *Physical review letters*, 88(17):174102, 2002.

[6] Th Buzug and G Pfister. Optimal delay time and embedding dimension for delay-time coordinates by analysis of the global static and local dynamical behavior of strange attractors. *Physical review A*, 45(10):7073, 1992.

[7] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, January 2009. Survey.

[8] Herbert Edelsbrunner and John L. Harer. *Computational topology: an introduction*. American Mathematical Society, 2009.

[9] Birgit Frank, Bernd Pompe, Uwe Schneider, and Dirk Hoyer. Permutation entropy improves fetal behavioural state classification based on heart rate analysis from biomagnetic recordings in near term fetuses. *Medical and Biological Engineering and Computing*, 44(3):179, 2006.

[10] Andrew M Fraser and Harry L Swinney. Independent coordinates for strange attractors from mutual information. *Physical review A*, 33(2):1134, 1986.

[11] Robert Ghrist. Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society*, 45:61–75, 2008. Survey.

[12] Peter Grassberger and Itamar Procaccia. Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, 9(1-2):189–208, 1983.

[13] Frank R Hampel. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393, 1974.

[14] Boris Iglewicz and David Hoaglin. *Volume 16: how to detect and handle outliers, The ASQC basic references in quality control: statistical techniques, Edward F. Mykytka*. PhD thesis, Ph. D., Editor, 1993.

[15] Matthew Kahle and Elizabeth Meckes. Limit theorems for betti numbers of random simplicial complexes. *Homology, Homotopy and Applications*, 15(1):343–374, 2013.

[16] Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4):764–766, 2013.

[17] Duan Li, Zhenhu Liang, Yinghua Wang, Satoshi Hagihira, Jamie W Sleigh, and Xiaoli Li. Parameter selection in permutation entropy for an electroencephalographic measure of isoflurane anesthetic drug effect. *Journal of clinical monitoring and computing*, 27(2):113–123, 2013.

[18] Michał Melosik and W Marszalek. On the 0/1 test for chaos in continuous systems. *Bulletin of the Polish*

Academy of Sciences Technical Sciences, 64(3):521–528, 2016.

[19] George B Moody and Roger G Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.

[20] S. Y. Oudot. *Persistence theory: from quiver representations to data analysis*, volume 209 of *AMS Mathematical Surveys and Monographs*. American Mathematical Society, 2015.

[21] Jose A Perea, Anastasia Deckard, Steve B Haase, and John Harer. Sw1pers: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. *BMC bioinformatics*, 16(1):257, 2015.

[22] Jose A Perea and John Harer. Sliding windows and persistence: An application of topological methods to signal analysis. *Foundations of Computational Mathematics*, 15(3):799–838, 2015.

[23] Steven M Pincus. Approximate entropy as a measure of system complexity. *Proceedings of the National Academy of Sciences*, 88(6):2297–2301, 1991.

[24] Anton Popov, Oleksii Avilov, and Oleksii Kanaykin. Permutation entropy of eeg signals for different sampling rate and time lag combinations. In *Signal Processing Symposium (SPS), 2013*, pages 1–4. IEEE, 2013.

[25] Mark A Richards. The discrete-time fourier transform and discrete fourier transform of windowed stationary white noise. *Georgia Institute of Technology, Tech. Rep*, 2013.

[26] Joshua S Richman and J Randall Moorman. Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology*, 278(6):H2039–H2049, 2000.

[27] Müller Riedl, A Müller, and N Wessel. Practical considerations of permutation entropy. *The European Physical Journal Special Topics*, 222(2):249–262, 2013.

[28] Songwon Seo. *A review and comparison of methods for detecting outliers in univariate data sets*. PhD thesis, University of Pittsburgh, 2006.

[29] Claude E Shannon, Warren Weaver, and Arthur W Burks. The mathematical theory of communication. 1951.

[30] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

[31] Floris Takens. Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980*, pages 366–381. Springer, 1981.

[32] Mei Tao, Kristina Poskuviene, Nizar Alkayem, Maosen Cao, and Minvydas Ragulskis. Permutation entropy based on non-uniform embedding. *Entropy*, 20(8):612, 2018.

[33] Hong Zhang and Xuncheng Liu. Analysis of parameter selection for permutation entropy in logistic chaotic series. In *Intelligent Transportation, Big Data & Smart City (ICITBS), 2018 International Conference on*, pages 398–402. IEEE, 2018.

A Summary of the used data and models

A.1 Gaussian White Noise

The white noise was generated using NumPy's random.normal function with a standard deviation of 1.0 and a length of 2000 points.

A.2 Lorenz System

The Lorenz system used is defined as

$$\frac{dx}{dt} = \sigma(y - x), \ \frac{dy}{dt} = x(\rho - z) - y, \ \frac{dz}{dt} = xy - \beta z. \quad (17)$$

The Lorenz system had a sampling rate of 100 Hz with parameters $\sigma = 10.0$, $\beta = 8.0/3.0$, and $\rho = 95$. This system was solved for 100 seconds and the last 24 seconds were used.

A.3 Rössler System

The Rössler system used was defined as

$$\frac{dx}{dt} = -y - z, \ \frac{dy}{dt} = x + ay, \ \frac{dz}{dt} = b + z(x - c), \quad (18)$$

with parameters of $a = 0.1$, $b = 0.1$, $c = 14$, which was solved over 400 seconds with a sampling rate of 100 Hz. Only the last 2000 data points of the x-solution were used in the analysis.

A.4 Bi-Directional Coupled Rössler System

The Bi-directional Rössler system is defined as

$$
\begin{aligned}
\frac{dx_1}{dt} &= -w_1 y_1 - z_1 + k(x_2 - x_1), \\
\frac{dy_1}{dt} &= w_1 x_1 + 0.165 y_1, \\
\frac{dz_1}{dt} &= 0.2 + z_1(x_1 - 10), \\
\frac{dx_2}{dt} &= -w_2 y_2 - z2 + k(x_1 - x_2), \\
\frac{dy_2}{dt} &= w_2 x_2 + 0.165 y_2, \\
\frac{dz_2}{dt} &= 0.2 + z_2(x_2 - 10),
\end{aligned}
\quad (19)
$$

with $w_1 = 0.99$, $w_2 = 0.95$, and $k = 0.05$. This was solved for 4000 seconds with a sampling rate of 10 Hz. Only the last 400 seconds of the x-solution were used in the analysis.

A.5 Mackey-Glass Delayed Differential Equation

The Mackey-Glass Delayed Differential Equation is defined as

$$x(t) = -\gamma x(t) + \beta \frac{x(t - \tau)}{1 + x(t - \tau)^n} \quad (20)$$

with $\tau = 2$, $\beta = 2$, $\gamma = 1$, and $n = 9.65$. This was solved for 400 seconds with a sampling rate of 100 Hz. Only the last 300 seconds of the x-solution were used in the analysis.

A.6 EEG Data

The EEG signal was taken from andrzejak et al. [3]. Specifically, the first 2000 data points from the EEG data of a healthy patient from set A, file Z-093 was used.

A.7 ECG Data

The Electrocardoagram (ECG) data was taken from SciPy's misc.electrocardiogram data set. This ECG data was originally provided by the MIT-BIH Arrhythmia Database [19]. We used data points 3000 to 4500 during normal sinus rhythm.

A.8 Logistic Map

The logistic map was generated as

$$x_{n+1} = rx_n(1 - x_n), \tag{21}$$

with $x_0 = 0.5$ and $r = 3.95$. Equation 21 was solved for the first 500 data points.

A.9 Hénon Map

The Hénon map was solved as

$$
\begin{aligned}
x_{n+1} &= 1 - ax_n^2 + y_n, \\
y_{n+1} &= bx_n,
\end{aligned}
\tag{22}
$$

where $b = 0.3$, $x_0 = 0.1$, $y_0 = 0.3$, and $a = 1.4$. This system was solved for the first 500 data points of the x-solution.

B Algorithms

B.1 Fast 0D persistence algorithm for 1D data

B.2 SW1PerS Algorithm for the Embedding Delay τ

The procedure we described in Section 3.3 is summarized in algorithm 2. This algorithm can be used to automatically determine a suitable delay τ using SW1PerS.

Result: Persistence Diagram: Birth and Deaths Times
begin
    initialize time series;
    find locations and values of minima and maxima;
    store locations and values in matrix (M_{minmax});
    while M_{minmax} *has values* **do**
        increment i;
        difference = maxima values - minima values;
        find index of smallest difference (I_{min});
        peak value (P_v) = M_{minmax} of $I_{\min}$ for max value column;
        peak index (P_i) = M_{minmax} of $I_{\min}$ for max index column;
        valley value (V_v) = M_{minmax} of $I_{\min}$ for min value column;
        valley index (V_i) = M_{minmax} of $I_{\min}$ for min index column;
        persistence diagram point[i] = [valley value, peak value];
        remove P_v, P_i, V_v, V_i from M_{minmax};
    end
    return persistence diagram points;
end

Algorithm 1: Zero Dimensional Persistence Algorithm.

Result: τ
begin
    set $n_T = 200$;
    calculate $L_{\max}$ using max significant frequency;
    calculate m using FFT reconstruction to 10% error threshold;
    linearly map time series onto the domain $[0, 2\pi]$;
    set $L = 1$;
    while $L < L_{\max}$ **do**
        calculate window size as $w = \frac{2\pi m}{L(m+1)}$;
        calculate τ_s as $\tau_s = w/m$;
        generate interpolated function for time series using cubic spline at τ_s intervals;
        create point cloud using n_T sliding windows as $SW_{m,\tau}f(t)$;
        apply persistence homology on SW point cloud;
        calculate periodicity score as $s(L) = 1 - \frac{r_B^2 - r_D^2}{3}$;
        increment L;
    end
    calculate L at minimum score as L_P;
    Find length of series as T;
    calculate main period of time series as $P = \frac{mT}{(m+1)L_P}$;
    set α between 2 and 4;
    calculate $\tau = \frac{f_s P}{\alpha}$;
    return τ;
end

Algorithm 2: SW1PerS algorithm for τ.