AIMS Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 9(10): 27028–27057.
DOI: 10.3934/math.20241315
Received: 03 July 2024
Revised: 19 August 2024
Accepted: 09 September 2024
Published: 18 September 2024

*Research article*

# Persistent homology of featured time series data and its applications

**Eunwoo Heo and Jae-Hun Jung***

Department of Mathematics, and Mathematical Institute for Data Science, Pohang University of Science and Technology, Pohang 37673, Korea

* **Correspondence:** Email: jung153@postech.ac.kr.

**Abstract:** Recent studies have actively employed persistent homology (PH), a topological data analysis technique, to analyze the topological information in time series data. Many successful studies have utilized graph representations of time series data for PH calculation. Given the diverse nature of time series data, it is crucial to have mechanisms that can adjust the PH calculations by incorporating domain-specific knowledge. In this context, we introduce a methodology that allows the adjustment of PH calculations by reflecting relevant domain knowledge in specific fields. We introduce the concept of featured time series, which is the pair of a time series augmented with specific features such as domain knowledge, and an influence vector that assigns a value to each feature to fine-tune the results of the PH. We then prove the stability theorem of the proposed method, which states that adjusting the influence vectors grants stability to the PH calculations. The proposed approach enables the tailored analysis of a time series based on the graph representation methodology, which makes it applicable to real-world domains. We consider two examples to verify the proposed method's advantages: anomaly detection of stock data and topological analysis of music data.

**Keywords:** topological data analysis; persistent homology; time series analysis; featured time series; graph representation; stability theorem
**Mathematics Subject Classification:** 00A69, 37M10, 55N31, 91B84

## 1. Introduction

Data that can be represented as one-dimensional variables is called time series data; this data format is simple and widely prevalent across various fields. Time series data analysis has been a long-standing field, including anomaly detection [1,2], forecasting [3–6], classification [7–10], and clustering [11–13]. Despite being extensively researched, time series analysis remains a challenging problem. Recent advancements in topological data analysis (TDA) have led to a surge in research on analyzing time series data using topological information. A prominent method within TDA is persistent homology (PH), wherein the duration for which a $p$-dimensional homology persists is calculated as the given data undergoes sequential transformations, thereby revealing the underlying topological structure. Research analyzing time series data using PH can be broadly classified into two categories: studies utilizing delay embedding and those utilizing graph representation.

The delay embedding $\phi_\tau$ for a time series $T$, as described by Packard et al. [14], is a function into $\mathbb{R}^m$ defined by $\phi_\tau(t) = (T(t), T(t-\tau), \ldots, T(t-(d-1)\tau))$, where $\tau$ is the delay lag and $m$ is the embedding dimension. This embedding method is employed to transform time series data into point clouds (PCs) in Euclidean space $\mathbb{R}^m$ for PH analysis. Takens' embedding theorem [15] ensures that this transformation preserves the topological information of the data sampled from hidden dynamical systems. The initial approach was used to analyze dynamical systems in [16], where the time series sampled from these systems were transformed into PCs for PH analysis. Subsequent research [17, 18] expanded this foundation by affirming the effectiveness of PH in identifying the periodicity within a time series and extending its application to quasi-periodic systems. Research on the early detection of critical transitions [19] based on [20] and analysis of the stability of dynamical systems [21] was conducted using PH and delay embedding. Moreover, PH and delay embedding methodologies were adopted for clustering [22] and classification [23] of time series data. The effectiveness of PH in multivariate time series analysis for machine learning tasks such as room occupancy detection was demonstrated in [24]. The versatility of PH has been further highlighted by demonstrating its broad applicability to diverse fields, e.g., identifying wheezes in signals [25], detecting financial crises [26, 27], classifying motions using motion-capture data [28], and distinguishing periodic biological signals from nonperiodic synthetic signals [29].

On the other hand, in addition to delay embedding methods, several studies have successfully used graph representation of time series data with PH. The graph representation $G = (V, E)$ of a time series $T$ consists of a vertex set $V$ and an edge set $E$, which are derived from the data points and their relationships within $T$. This graph is then analyzed using PH to study the topological features of $T$. Notably, functional magnetic resonance imaging (fMRI) data of the brain was converted into graph data, called functional networks in [30], which was then analyzed using PH. In [31], the early signs of critical transitions in financial time series were detected by converting the series into graph data, called correlation networks, for PH analysis. In text mining, PH has been applied to graphs representing the main characters in the text time series [32]. A study proposed the construction of graphs, called music networks, for classifying Turkish makam music through PH analysis [33]. These music networks were used in [34] to analyze traditional Korean music using PH and in [35] to study composition by integrating machine learning. In addition, the time series data has been transformed into Tonnetz, proposed by Euler, for music classification through PH analysis [36]. The authors of [37] addressed the classification of music style by creating data called intervallic transition graphs from music time series and applying PH. Graph representation of time series is also crucial in recent artificial intelligence-based time series analysis utilizing graph neural networks (GNNs). The graph representation of time series data was employed for forecasting [38], anomaly detection [39], and classification [40] with GNNs.

Given the extensive and diverse nature of time series data, in addition to delay embedding, various other embedding methods such as derivative embedding [14], integral-differential embedding [41], and global principal value embedding [42] have been used. Recently, selecting appropriate embeddings from these various embeddings using PH was investigated in [43]. Similarly, choosing the appropriate graph representation for each research field is also crucial. However, further research is still needed on how to adjust and select suitable graph representations. In this context, our study makes the following contributions:

(1) We introduce a novel concept of featured time series data created by adding domain knowledge (features) to the time series data and influence vectors that assign a value to each feature. This allows the adjustment of graph representations to ensure suitability to the specific domain.

(2) We prove that adjusting the graph representations via the influence vectors provides stability to the PH calculation, thereby demonstrating the robustness of the proposed method.

The remainder of this paper is organized as follows. Section 2 describes the frequency-based graph representation and methods for calculating the PH of the graphs. We provide an example of potential information loss that can occur during the PH computation process. In Section 3, we introduce the novel concepts of featured time series data and influence vectors to analyze a time series by incorporating domain knowledge. Furthermore, we show that the examples of information loss presented in Section 2 can be addressed by adjusting the influence vectors. In Section 4, we state the theorem that asserts the stability property of the influence vectors when subjected to such adjustments. This theorem underscores the reliability and robustness of the influence vector adjustments in mitigating information loss during the graph representation process. In Section 5, we explore the effect of variations in the influence vectors on the analysis of real-world time series data based on previous research, including anomaly detection of stock data and topological analysis of music data. All codes used in this study are available at `https://github.com/AI-hew-math/Feat_time_series`. In Section 6, we provide a detailed proof of the main theorem introduced in Section 4. In Section 7, a brief concluding remark is provided.

## 2. Frequency-based PH analysis of time series data

In this study, we addressed the graph representation of time series data using the frequency of occurrence of observations in the time series data.

### 2.1. Frequency-based graph construction

Consider a time series $T : \mathbb{T} \to \mathcal{X}$, where $\mathcal{X}$ represents the set of all possible values that the time series data can take, and $\mathbb{T}$ is assumed finite. Define the vertex or node set $V$ as

$$V = \{\{T(t)\} \mid t \in \mathbb{T}\}.$$

Enumerating $\mathbb{T}$ as $\{t_1, t_2, \ldots, t_n\}$, the edge set $E$ is defined as

$$E = \{\{T(t_i), T(t_{i+1})\} \mid 1 \leq i \leq n-1, T(t_i) \neq T(t_{i+1})\}.$$

Let $f_e$ represent *the frequency of an edge $e \in E$* in $T$ given by

$$f_e = \Big|\{t_i \in \mathbb{T} \mid e = \{T(t_i), T(t_{i+1})\}\}\Big|.$$

Here, $f_e$ measures the total number of co-occurrences of the two nodes appearing side by side in $T$ associated with the edge $e$. Note that when defining the frequency $f_e$ as above, we did not consider these two nodes' specific order of appearance. The frequency of the edge $e$ increases whenever the two nodes are positioned adjacent to each other in $T$. Define a weight function $W_E$ on $E$ as $W_E(e) = f_e$ for any edge $e \in E$. This weight measures how strongly the associated nodes are connected to each other in $e \in E$.

### 2.2. *Distance on weighted graphs*

To compute the PH over the graph, we define the distance between two nodes in the graph. In this study, we considered the definition of natural distance, which is given below. However, note that the distance definition is not unique but might be the best definition depending on the problem.

**Definition 2.1.** *(Distance)* Let $G = (V, E, W_E)$ be a connected weighted graph. Define the distance between $v$ and $w$ in $V$ such that $d(v, w) = 0$ if $v = w$; otherwise,

$$d(v, w) = \min_{p} \left\{ \sum_{e \in p} (W_E(e))^{-1} \middle| p \text{ is a path in } G \text{ connecting } v \text{ and } w \right\}.$$

Note that the distance $d$ satisfies the metric conditions. The proof is provided in Proposition 3.1, where a more general case is presented. For $d$, we utilize the reciprocal of the edge weight function $W_E$. This implies that we consider the vertices $v$ and $w$ connected by the edge $e$ as closer when the frequency $f_e$ of the edge is higher. Define a *distance matrix* $A$ as $A = (a_{ij})$, where $a_{ij} = d(v_i, v_j)$ for every pair of vertices $v_i, v_j \in V$.

### 2.3. *Persistent homology of metric spaces*

We now consider computing PH given a metric space $(V, d)$. Suppose we have a metric space $(V, d)$ from graph $G = (V, E, W_E)$. The power set $\mathcal{P}(V)$ of the vertex set $V$ is an abstract simplicial complex, denoted as $\mathbb{X}$. We define the Vietoris-Rips (Rips) filtration function $h : \mathbb{X} \to \mathbb{R}$ as

$$h(\sigma) = \max\{d(v_i, v_j) \mid \tau \subseteq \sigma, 1\text{-simplex } \tau = \{v_i, v_j\}\},$$

whenever $p \geq 1$ for $p$-simplex $\sigma \in \mathbb{X}$. For any 0-simplex $v$, we define $h(v) = 0$. Denote $h^{-1}((-\infty, \epsilon])$ as $\mathbb{X}_\epsilon$ for $\epsilon \in \mathbb{R}$. Then, for $\epsilon_n \geq \max_\sigma(h(\sigma))$ and $0 = \epsilon_0 \leq \epsilon_1 \leq \epsilon_2 \leq \cdots \leq \epsilon_n$,

$$\mathbb{X}_{\epsilon_0} \subseteq \mathbb{X}_{\epsilon_1} \subseteq \cdots \subseteq \mathbb{X}_{\epsilon_n} = \mathbb{X}$$

forms the Rips filtration for an abstract simplicial complex $\mathbb{X}$ (see [44, 45]). A simplicial complex can be associated with a sequence of abelian groups, termed chain groups. For each dimension $p$, the $p$-chains $C_p(X)$ are the formal sums of the $p$-simplices. For each dimension $p$, there exists a boundary map $\partial_p : C_p(\mathbb{X}) \to C_{p-1}(\mathbb{X})$ that maps each $p$-simplex to its boundary. A crucial property is that the boundary of the boundary of the simplices is always null, that is, $\partial_{p-1} \circ \partial_p = 0$. Given the chain groups and boundary maps, the *pth homology group* is defined as the quotient group

$$H_p(\mathbb{X}) = \frac{\ker(\partial_p)}{\operatorname{Im}(\partial_{p+1})}.$$

This quotient captures the $p$-dimensional holes in a given topological space. As the filtration progresses, these homology groups also change. The PH serves as an effective tool for tracing such changes. For each dimension $p$ and filtration value $\epsilon_i$, we compute $H_p(\mathbb{X}_{\epsilon_i})$. As $\epsilon_i$ changes, certain homological features (e.g., loops) may emerge or disappear. A persistence diagram [46] (PD) is a multi-set that visualizes the coordinates of these features on a plane. The $x$-coordinate of a point in the PD indicates the birth of such a homological feature, and the $y$-coordinate indicates its death. The longer the persistence of the considered feature, the greater the deviation of the corresponding point from the diagonal on the plane of the PD. Note that the computation of PH depends on the filtration choice, but Rips filtration is one of the most popular filtration methods. Open-source libraries used for computing the PH based on Rips filtration include JavaPlex [47] written in Java but easily usable in MATLAB, GUDHI [48] written in C++ but also accessible in Python, and the recently developed Ripser [49], which significantly reduces the computation time and memory usage when compared with the other methods. Rip filtration allows the computation of PH not only on point clouds but also on general abstract simplicial complexes, especially graphs. In [50], the authors demonstrated that Rips filtration could approximate the topological shapes of the data's manifolds, as discussed in [51].

### *2.4. Discussion of information loss through an example*

Let us examine a simple example to see how PH computation for time series analysis of a specific field can result in information loss. Here, we discuss a simple case of anomaly detection in a time series. Consider a time series $T$ consisting of 27 timestamps for temperature changes, as shown in Figure 1. We can construct a graph from $T$, as shown in Figure 2. In the graph, each vertex represents a temperature value, and the value associated with the edge represents the co-occurring frequency of the two adjacent temperature values in $T$. From the definition of the distance $d$, the distance matrix is obtained as shown in Figure 2. The matrix's rows and columns correspond to the temperature values associated with the vertices $21, 22, 23, 24$, and 25. For example, the element at the position of $(2, 3)$ in the matrix represents the distance between the vertices 22 and 23. The complexity of computing the distance matrix depends on the complexity of the pathfinding process $O(|E| \log|V|)$, which can be efficiently calculated using the Dijkstra algorithm [52].

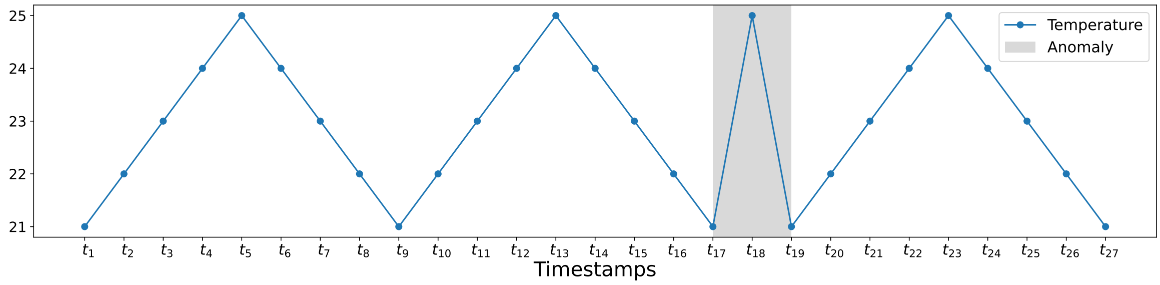
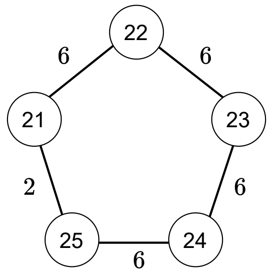


**Figure 1.** A time series of temperature with the anomaly region shaded.



$$\begin{pmatrix} 0 & \frac{1}{6} & \frac{1}{3} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{6} & 0 & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{3} & \frac{1}{6} & 0 \end{pmatrix}$$

**Figure 2.** Weighted graph $G = (V, E, W_E)$ of the time series $T$ in Figure 1 (left) and the corresponding distance matrix (right).

We construct the Rips filtration for the distance matrix and compute the persistence barcode. The upper row of Figure 3 illustrates how the Rips complex changes in the Rips filtration as the filtration value $\epsilon$ varies. The bottom row shows the corresponding 0-dimensional persistence barcode. The corresponding persistence barcode shown in Figure 3 illustrates the occurrence of information loss during filtration. The time series $T$ in Figure 1 undergoes abnormal temperature changes in the shaded area, where temperatures (nodes) 21 and 25 are connected twice. As shown in Figure 3, the information regarding the frequency between nodes 21 and 25 in the anomaly region and its distance $\frac{1}{2}$ are not reflected in the computation of homology. The birth and death of the 0-dimensional homology occur at 0, $\frac{1}{6}$, and $\infty$, with no occurrence at $\frac{1}{2}$. Moreover, the 1-dimensional homology is not detected either. This is because the interior of the pentagon is filled in advance at $\frac{1}{3}$ preceding $\frac{1}{2}$. Therefore, if the change in temperature from 21 to 25 is considered an anomaly, then it must be reflected in this loss of information. For example, instead of simply increasing the frequency by 1 at each appearance of 21 and 25 in $T$, we can assign a larger value. This can generate 1-dimensional homology because if the distance $d(21, 25)$ becomes smaller than $\frac{1}{3}$, a 1-cycle is formed before any interior edge of the pentagon is created. In addition, if the distance $d(21, 25)$ becomes less than $\frac{1}{6}$, the edge $\{21, 25\}$ will be the first to emerge in the filtration; thus, $d(21, 25)$ becomes the death time in the 0-dimensional barcode. This is discussed in detail in Example 3.5 later. These examples underscore the critical need for domain-specific adjustments in analysis to accurately perform PH computations for time series data within particular fields.

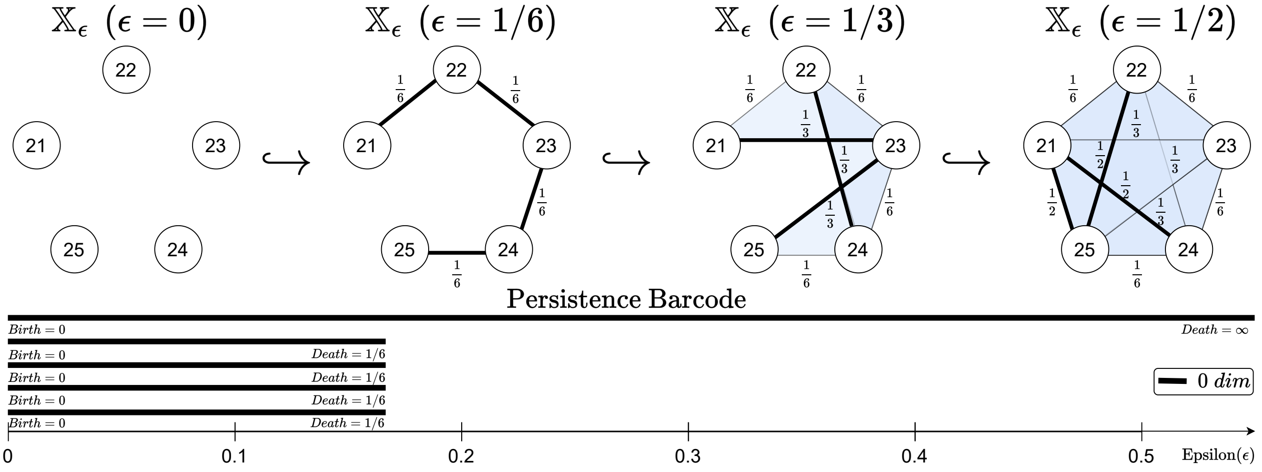

$\mathbb{X}_\epsilon$ ($\epsilon = 0$)
$\mathbb{X}_\epsilon$ ($\epsilon = 1/6$)
$\mathbb{X}_\epsilon$ ($\epsilon = 1/3$)
$\mathbb{X}_\epsilon$ ($\epsilon = 1/2$)
21
22
23
24
25
$\frac{1}{6}$
$\frac{1}{3}$
$\frac{1}{2}$
Persistence Barcode
Birth = 0
Death = ∞
Death = 1/6
0 dim
0
0.1
0.2
0.3
0.4
0.5
Epsilon($\epsilon$)

**Figure 3.** Rips filtration (Top) and its persistence barcode (Bottom).

## 3. Introduction of featured time series data

We propose the featured time series data below to enable adjustments in the PH calculation that reflect each field's domain knowledge.

### 3.1. Featured time series data

**Definition 3.1.** *(Feature set)* Let $F^p$ denote finite sets, where $p = 0, 1$. The Cartesian product of the power set of $F^p$ is called the feature set, denoted by $\mathcal{F}$. That is, $\mathcal{F} = \mathcal{P}(F^0) \times \mathcal{P}(F^1)$. We call $F^p$ the $p$th feature set. The elements of $F^p$ are called the $p$th features.

As demonstrated below, the $p$th feature is associated with the $p$-simplex in a graph in our proposed method.

**Definition 3.2.** *(Influence vectors)* Consider a nonnegative real value function $g : F^0 \cup F^1 \cup \{\emptyset^0, \emptyset^1\} \to \mathbb{R}$, where the symbols $\emptyset^0$ and $\emptyset^1$ represent the states of the given time series without any feature. This function $g$ is called the influence vector with the size of $|F^0| + |F^1| + 2$. For any element $z$ in $F^0 \cup F^1 \cup \{\emptyset^0, \emptyset^1\}$, the value $g(z)$ is referred to as the influence of $z$.

**Definition 3.3.** *(Featured time series)* Let $T : \mathbb{T} \to \mathcal{X}$ be a time series, $\mathcal{F}$ be a feature set, and $g$ be an influence vector. Consider a function $\widehat{T} : \mathbb{T} \to \mathcal{X} \times \mathcal{F}$ such that $\widehat{T}(t) = (T(t), T_f(t)) \in \mathcal{X} \times \mathcal{F}$ for $t \in \mathbb{T}$, where $T_f : \mathbb{T} \to \mathcal{F}$ represents any function. The pair $(\widehat{T}, g)$ is called a featured time series of $T$ by $g$. The function $T_f$ is termed the feature component of the $\widehat{T}$.

For example, let $F^0 = \{r_1, r_2, r_3\}$ and $F^1 = \{s_1, s_2\}$ represent the zeroth and first feature sets, respectively. Consider a time series $T : \mathbb{T} \to \mathcal{X}$ such that $T(t_i) = x_{t_i}$ , where $\mathbb{T} = \{t_i \mid i \in \mathbb{N}\}$. The following sequence illustrates an example of $\widehat{T}$:

$$(x_{t_1}, \{r_1, r_3\}, \{s_1, s_2\}),\ (x_{t_2}, \{r_2, r_3\}, \emptyset^1),\ \ldots,\ (x_{t_i}, A_{t_i}, B_{t_i}),\ \ldots,$$

where $A_{t_i} \subseteq \{r_1, r_2, r_3\}$ and $B_{t_i} \subseteq \{s_1, s_2\}$. For any influence vector $g$, $(\widehat{T}, g)$ is a featured time series.

### *3.2. Graph representation of featured time series data*

Consider a featured time series $(\widehat{T}, g)$ as defined in Definition 3.3. A connected graph $G = (V, E, W_E)$ is constructed using $T$ as explained in Section 2.1. The main goal of the proposed method is to generalize the weight function $W_E$ in $G$, yielding a more flexible and adaptive PH analysis by fully utilizing the given information in the time series. For such generalization, we use $T_f$ and $g$.

Consider two time series $T_f^0 : \mathbb{T} \to \mathcal{P}(F^0)$ and $T_f^1 : \mathbb{T} \to \mathcal{P}(F^1)$ such that $T_f(t) = (T_f^0(t), T_f^1(t))$ for any $t \in \mathbb{T}$. Write $F^0 = \{r_1, \ldots, r_m\}$ and $V = \{v_1, v_2, \ldots, v_{|V|}\}$ as ordered sets. Define the *zeroth count matrix* $C_0 = (c^0_{ij})$ using $F^0$ such that

$$c^0_{ij} := \begin{cases} \left|\{t \in \mathbb{T} \mid v_i = T(t), r_j \in T_f^0(t)\}\right| & \text{if } j = 1, \ldots, m, \\ \left|\{t \in \mathbb{T} \mid v_i = T(t), T_f^0(t) = \emptyset^0\}\right| & \text{if } j = 0, \end{cases}$$

where $\left|\cdot\right|$ denotes the cardinality of the set. The $c^0_{ij}$ represents the total number of timestamps $t \in \mathbb{T}$ when the vertex $v_i$ in $V$ and feature $r_j$ in $F^0$ are encountered together in $\widehat{T}$. The $i$th element of the first column of $C_0$ represents the total number of timestamps when $v_i$ appears without any zeroth features.

Similarly, write $F^1 = \{s_1, \ldots, s_l\}$ and $E = \{e_1, e_2, \ldots, e_{|E|}\}$ as ordered sets and define the *first count matrix* $C_1 = (c^1_{ij})$ using $F^1$ such that

$$c^1_{ij} := \begin{cases} \left|\{t_k \in \mathbb{T} \mid e_i = \{T(t_k), T(t_{k+1})\}, s_j \in T_f^1(t_k)\}\right| & \text{if } j = 1, \ldots, l, \\ \left|\{t_k \in \mathbb{T} \mid e_i = \{T(t_k), T(t_{k+1})\}, T_f^1(t_k) = \emptyset^1\}\right| & \text{if } j = 0. \end{cases}$$

The $c^1_{ij}$ represents the total number of timestamps $t \in \mathbb{T}$ when the edge $e_i$ and feature $s_j$ in $F^1$ are encountered together in $\widehat{T}$. The $i$th element of the first column of $C_1$ represents the total number of the timestamps when $e_i$ appears without any first features.

Set $\vec{g_0} = (g(\emptyset^0), g(r_1), g(r_2), \ldots, g(r_m))$ and $\vec{g_1} = (g(\emptyset^1), g(s_1), g(s_2), \ldots, g(s_l))$ for an influence vector $g$. Consider two vectors $C_0 \cdot \vec{g_0}$ and $C_1 \cdot (\vec{g_1} + \vec{1})$. Here, $\cdot$ denotes matrix multiplication, and $\vec{1}$ means the vector whose elements are all 1. The use of $\vec{1}$ in $C_1 \cdot (\vec{g_1} + \vec{1})$ is to ensure that when $\vec{g_1} = 0$, the value $(C_1 \cdot \vec{1})_i$ aligns with the frequency $f_e$ defined in Section 2.1 for edge $e$ in $E$.

Define a vertex weight function $\widehat{W}_V : V \to \mathbb{R}$ as $\widehat{W}_V(v_i) = (C_0 \cdot \vec{g_0})_i$ for $v_i \in V$ and an edge weight function $\widehat{W}_E : E \to \mathbb{R}$ as $\widehat{W}_E(e_i) = (C_1 \cdot (\vec{g_1} + \vec{1}))_i$ for $e_i \in E$. Write $\widehat{W}_E(e_i)$ as $\widehat{f_{e_i}}$. The weight $\widehat{f_e}$ is called the weighted frequency of an edge $e$. It is a generalization of the frequency explained in Section 2.1. The weight $\widehat{W}_V(v)$ is called the weighted frequency of a vertex $v$ associated with the zeroth features, $r_1, r_2, \ldots, r_m$. When $\vec{g_0} = \vec{1}$ , the $\widehat{W}_V(v)$ is simply the frequency of appearance of vertex $v$ in $\widehat{T}$.

We then obtain the vertex- and edge-weighted graph $\widehat{G^g} = (V, E, \widehat{W}_V, \widehat{W}_E)$ from the featured time series $(\widehat{T}, g)$. These weight functions are used to define the metric space in the following section.

### 3.3. Distance on vertex-edge weighted graphs

Suppose that we have a graph $\widehat{G^g} = (V, E, \widehat{W}_V, \widehat{W}_E)$. Let $\alpha = \min_{e\in E}(\widehat{W}_E(e))^{-1}$. Take an increasing function $\rho : \mathbb{R} \to (-1, 1)$ into the open interval $(-1, 1)$ such that $\rho(0) = 0$. We call $\rho$ an activation function. Define the length function $L^g : E \to \mathbb{R}$ of the graph $\widehat{G^g}$ as

$$L^g(e) = (\widehat{W}_E(e))^{-1} - \alpha\left(\rho(\widehat{W}_V(a) + \widehat{W}_V(b))\right) \text{ for any edge } e = \{a, b\} \in E.$$

Here, $\alpha$ ensures that the length $L^g(e) = (\widehat{W}_E(e))^{-1} - \alpha\left(\rho(\widehat{W}_V(a) + \widehat{W}_V(b))\right)$ remains positive for any edge $e$. The distance $\widehat{d}$ of the graph $\widehat{G^g}$ is defined below.

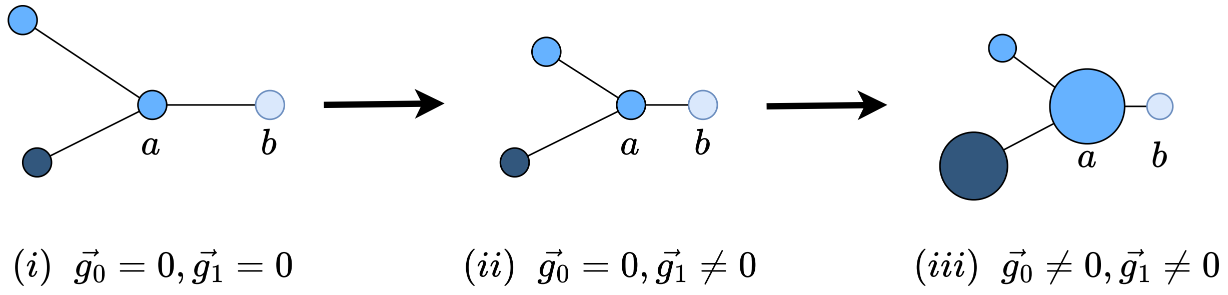
**Definition 3.4.** (Distance) Let $v, w \in V$ be any vertices. If $v = w$, define $\widehat{d}(v, w) = 0$. Otherwise, the distance $\widehat{d}$ between $v$ and $w$ is defined as

$$\widehat{d}(v, w) = \min\left\{\sum_{e\in p} L^g(e) \middle| p \text{ is a path in } G \text{ from } v \text{ to } w\right\}.$$

The motivation for defining distance $\widehat{d}$ is as follows. Consider an edge $e = \{a, b\}$ in the graph $\widehat{G^g}$. As discussed in Section 2.2, we fundamentally use the reciprocal of $\widehat{W}_E$ for the distance $\widehat{d}$. The stronger the relevance between $a$ and $b$, the closer they should be positioned. Compared with $W_E$, in particular, $\widehat{W}_E$ is a function that further assigns the sum of the influences on the features, meaning the more frequent the appearance of the first feature, which has a significant influence on the featured time series $\widehat{T}$, the stronger the connection becomes. Additionally, we used the vertex weight function $\widehat{W}_V$ for the distance $\widehat{d}$. This is designed to reflect the influence of a single node $a$ on the other nodes in the entire network. The greater the vertex weight assigned to $a$, the shorter the distance from $a$ to all other nodes. For example, consider a social network $G$ in which node $a$ is a famous influencer in the network and node $b$ is connected to $a$. Even if there is no direct interaction between nodes $a$ and $b$, it is natural to consider the distance between them to be small if $a$'s influence is strong. To achieve this, in the definition of the length function $L^g$, we subtract the sum of the influences of nodes $a$ and $b$ from $\widehat{W}_E(e)$, resulting in $\widehat{W}_E(e) - (\widehat{W}_V(a) + \widehat{W}_V(b))$. However, as $\widehat{W}_V$ and $\widehat{W}_E$ are independent functions, they can be negative. Therefore, we ultimately define the length $L^g$ for any edge $e = \{a, b\} \in E$ as

$$L^g(e) = (\widehat{W}_E(e))^{-1} - \alpha\left(\rho(\widehat{W}_V(a) + \widehat{W}_V(b))\right).$$

Figure 4 presents a schematic illustration of how the graph $\widehat{G}^g = (V, E, \widehat{W}_V, \widehat{W}_E)$ changes as we consider more general vertex weights $\widehat{W}_V$ and edge weights $\widehat{W}_E$.



$(i)$ $\vec{g_0} = 0, \vec{g_1} = 0$ $(ii)$ $\vec{g_0} = 0, \vec{g_1} \neq 0$ $(iii)$ $\vec{g_0} \neq 0, \vec{g_1} \neq 0$

**Figure 4.** Schematic illustrations of the weighted graph $\widehat{G}^g = (V, E, \widehat{W}_V, \widehat{W}_E)$ for various influence vectors $g$. (i) represents the case where $\widehat{W}_V = 0$ and $\widehat{W}_E = W_E$, (ii) shows the changes when only the edge weight is varied, and (iii) illustrates the changes in the vertex weights from those in (ii).

The distance $\widehat{d}$ satisfies the metric conditions, making it a more appropriate distance measure that naturally approximates the actual data space more closely than general distance measures without metric conditions.

**Proposition 3.1.** *Let $\widehat{d}$ be the distance as defined in Definition 3.4. Then, $\widehat{d}$ is a metric.*

*Proof.* Let $v, w, z \in V$ be any vertices. If $v = w$, we have $\widehat{d}(v, w) = 0$ by Definition 3.4. In addition, the length function $L^g$ is positive; therefore, $\widehat{d}(v, w) = 0$ implies $v = w$. Suppose that $v \neq w$. For any path $p$ from $v$ to $w$, we can create a path $p^{-1}$ from $w$ to $v$ by simply reversing the order of the vertices in $p$. Therefore, the distance $\widehat{d}$ satisfies $\widehat{d}(v, w) = \widehat{d}(w, v)$. Finally, let us prove the condition of the triangle inequality. Denote the path satisfying the minimality in the definition of $\widehat{d}(v, z)$ as $p_1$, and the path in $\widehat{d}(z, w)$ as $p_2$. Consider a path $p_1 \cup p_2$, which is obtained by concatenating $p_1$ and $p_2$. Sum all values $L^g(e)$ according to any edge $e$ in path $p_1 \cup p_2$, then it is $\widehat{d}(v, z) + \widehat{d}(z, w)$. Because $p_1 \cup p_2$ is a path from $v$ to $w$, by the property of minimality in the definition of $\widehat{d}(v, w)$, we have $\widehat{d}(v, w) \leq \widehat{d}(v, z) + \widehat{d}(z, w)$. □

**Corollary 3.1.** *$(V, \widehat{d})$ is a metric space for graph $\widehat{G}^g = (V, E, \widehat{W}_V, \widehat{W}_E)$.*

The distance $\widehat{d}$ defined on $\widehat{G}^g$ can be considered a generalization of the distance $d$ from Definition 2.1 from the perspective of the following proposition:

**Proposition 3.2.** *If the influence vector $g$ is zero, then the two distances $d$ in Definition 2.1 and $\widehat{d}$ in Definition 3.4 are identical on $V$.*

*Proof.* Because $\vec{g_0}$ is zero, the vertex weight function $\widehat{W}_V$ is zero. From $\rho(0) = 0$, we get

$$\sum_{e \in p} L^g(e) = \sum_{\substack{e = \{a,b\} \\ e \in p}} \left( (\widehat{W}_E(e))^{-1} - \alpha\rho(\widehat{W}_V(a) + \widehat{W}_V(b)) \right) = \sum_{e \in p} (\widehat{W}_E(e))^{-1}.$$

As $\vec{g_1}$ is zero, it follows that $\widehat{f_{e_i}} = (C_1 \cdot (\vec{g_1} + \vec{1}))_i = (C_1 \cdot \vec{1})_i = f_e$, leading to $\widehat{W}_E(e) = \widehat{f_e} = f_e = W_E(e)$.

Consequently, by the definition of $\widehat{d}$, we obtain

$$\begin{aligned}\widehat{d}(v,w) &= \min\left\{\sum_{e\in p} L^{g}(e) \middle| p \text{ is a path between } v \text{ and } w\right\}\\ &= \min\left\{\sum_{e\in p}(W_E(e))^{-1} \middle| p \text{ is a path between } v \text{ and } w\right\} = d(v,w).\end{aligned}$$

□

### 3.4. Activation function

The role of $\rho$ is to ensure that the values of the sum of any two vertex weights exist between 0 and 1 in the ascending order. If $\rho$ is not appropriately chosen, all the elements of $\rho(\widehat{W}_V(a) + \widehat{W}_V(b))$ in Definition 3.4 may become nearly identical for any edge $e = (a, b)$. It is crucial to select $\rho$ so that the distribution of the elements of $\rho(\widehat{W}_V(a) + \widehat{W}_V(b))$ facilitates effective analysis. Although the choice of the activation function can vary depending on the problem, this process can also be efficiently automated. We consider the following activation function:

$$\rho(z) = \begin{cases} 1 - e^{-z^2} & \text{if } z \geq 0, \\ 0 & \text{if } z < 0. \end{cases}$$

Let $M$ be the maximum value of $\{\widehat{W}_V(v_i) + \widehat{W}_V(v_j) \mid i \neq j\}$, where $M \geq 0$. Define the automatic activation function $\rho_*(z)$ as $\rho_*(z) = \rho\left(\frac{2z}{M+1}\right)$.

**Proposition 3.3.** *The function $\rho_*(z)$ is Lipschitz continuous with a constant that is independent of the influence vector $g$, although $M$ depends on $\widehat{W}_V$.*

*Proof.* We know that $\rho$ is Lipschitz continuous owing to the boundness of its derivative $\rho'$. There exists a constant $k$, independent of $\widehat{W}_V$, such that we have $|\rho'(z)| \leq k$ for any $z$. We can infer that $|\rho'_*(z)| = \frac{2}{M+1}|\rho'(\frac{2z}{M+1})| \leq 2k$. □

In the stability theorem proved in Section 6, the independence of the Lipschitz constant of the activation function from the influence vectors $g$ becomes crucial. Proposition 3.3 means that using the automatic activation function $\rho_*$ is permissible.

### 3.5. Example: addressing information loss in PH calculation

Consider the time series $T$, as shown in Figure 1. We define the zeroth feature set $F_0$ as $\{L, H\}$, representing humidity levels including low (L) and high (H), and the first feature set $F_1$ as $\{|T(t+1) - T(t)| \mid t \in \mathbb{T}\}$, representing the magnitude of the temperature changes. As shown in Figure 5, $T$ is complemented by a tuple of features $(f_1, f_2) \in F_0 \times F_1$, providing additional information. This is denoted as $\widehat{T}$.

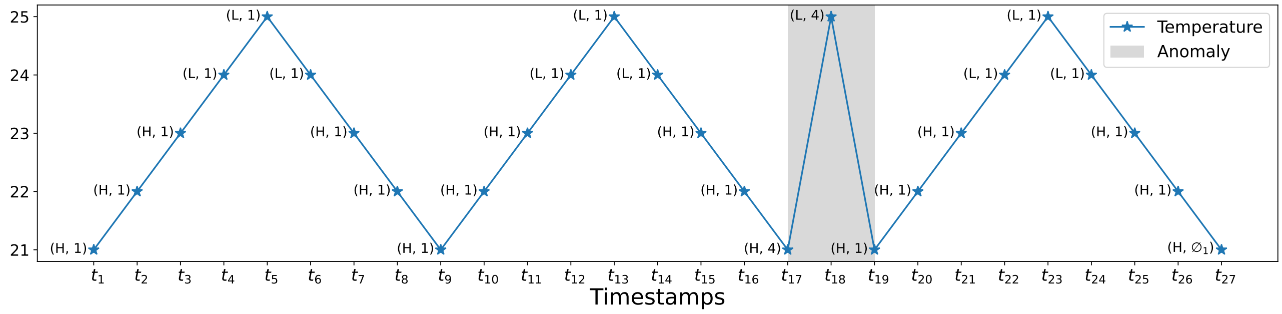


**Figure 5.** A time series $\widehat{T}$ with features added from the original time series $T$ shown in Figure 1.

To emphasize the sudden changes in temperature, define the influence vector $g$ such that $g(4) = 5$ for $4 \in F_1$ and $g(x) = 0$ for all other features of $x$. In other words, $\vec{g_0}$ and $\vec{g_1}$ are expressed as

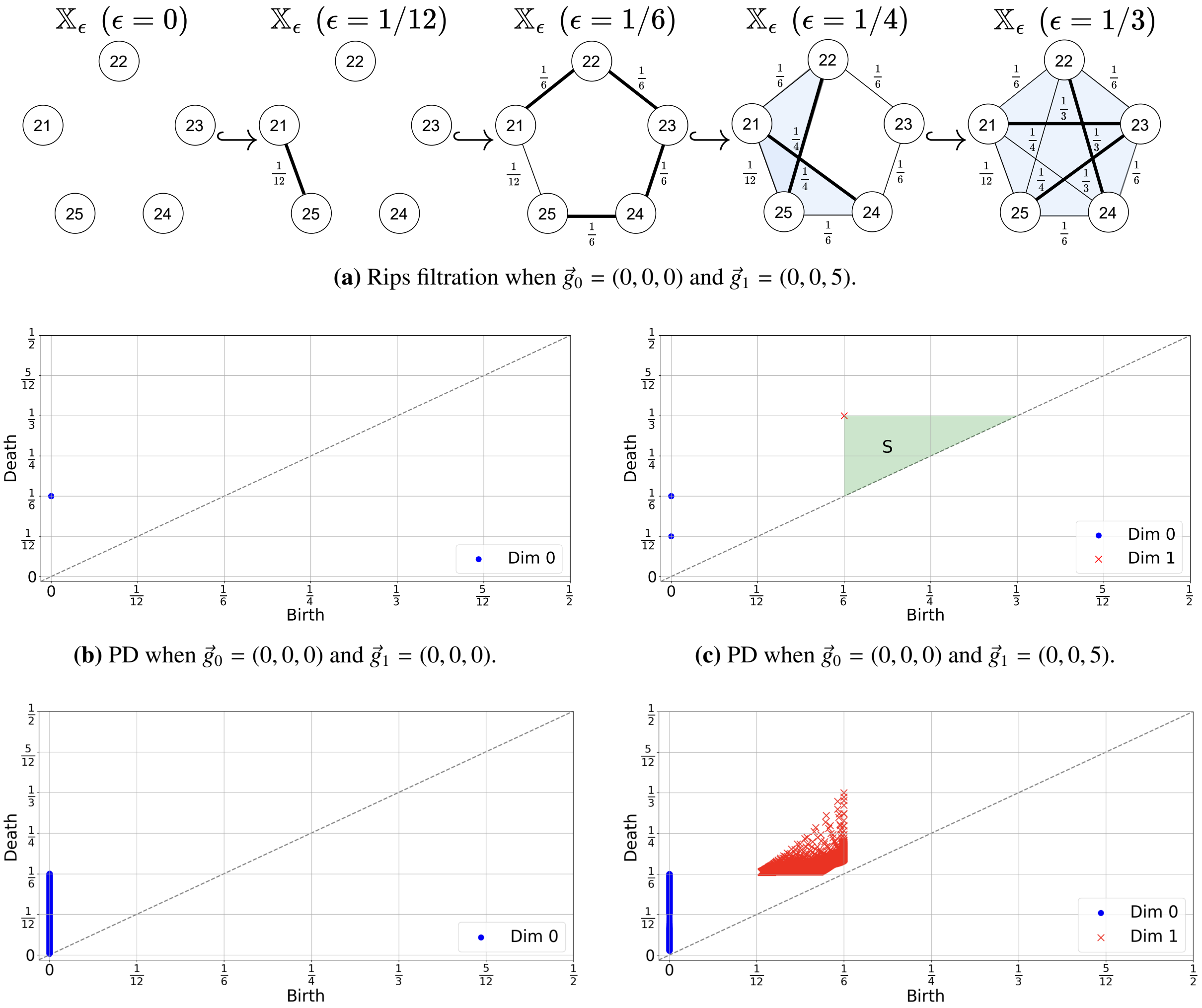
$$\vec{g_0} = \begin{array}{c} \begin{matrix} \emptyset^0 & H & L \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T \end{array} \text{ and } \vec{g_1} = \begin{array}{c} \begin{matrix} \emptyset^1 & 1 & 4 \end{matrix} \\ \begin{pmatrix} 0 & 0 & 5 \end{pmatrix}^T \end{array}.$$

For the featured time series $(\widehat{T}, g)$, the vertex weight function $\widehat{W}_V$ is zero, as $\vec{g_0} = \vec{0}$. The edge weight function $\widehat{W}_E$ can be obtained using the first count matrix $C_1$ as follows:

$$\begin{array}{c|ccc|} & \emptyset^1 & 1 & 4 \\ (21,22) & 0 & 6 & 0 \\ (22,23) & 0 & 6 & 0 \\ (23,24) & 0 & 6 & 0 \\ (24,25) & 0 & 6 & 0 \\ (21,25) & 0 & 0 & 2 \end{array} \times \begin{array}{c} \vec{g_1} + \vec{1} \\ \begin{pmatrix} 1 \\ 1 \\ 6 \end{pmatrix} \end{array} = \begin{array}{c} C_1 \cdot (\vec{g_1} + \vec{1}) \\ \begin{pmatrix} 6 \\ 6 \\ 6 \\ 6 \\ 12 \end{pmatrix} \end{array}.$$

Figure 6a illustrates the Rips filtration for the metric space $(V, \widehat{d})$, which is constructed from the graph $\widehat{G^g} = (V, E, \widehat{W}_V, \widehat{W}_E)$. In the filtration, the edge $\{21, 25\}$ first appears in $\mathbb{X}_\epsilon$ when $\epsilon = d(21, 25) = \frac{1}{12}$. Unlike the example discussed in Section 2.4, by assigning an influence value of 5 to $4 \in F_1$, we emphasize the importance of the edge $\{21, 25\}$. This not only results in the formation of a point in the 1-dimensional PD, as observed in Figure 6a, but also leads to the emergence of a point in the 0-dimensional PD with a death time of $d(21, 25) = \frac{1}{12}$. These developments are illustrated in Figures 6a and 6c, which contrast with the scenario in Figures 3 and 6b, where the latter displays a single point representing four 0-dimensional points. This example illustrates that by adjusting the influence vector $g$, previously unrepresented domain-specific information can now be effectively preserved in the computation of PH.

In this example, the humidity was used as the zeroth feature. Figures 6d and 6e show that the influence on the zeroth features affects the persistence (death time-birth time) of each point rather than the existence of 1-dimensional points. Although not extensively covered in this example, the zeroth feature is considered significant in the stock data, as discussed in Section 5.1. Furthermore, in Section 5.1, the area of the shaded region $S$ in Figure 6c is used for anomaly detection on the stock data.



**(a)** Rips filtration when $\vec{g}_0 = (0, 0, 0)$ and $\vec{g}_1 = (0, 0, 5)$.

**(b)** PD when $\vec{g}_0 = (0, 0, 0)$ and $\vec{g}_1 = (0, 0, 0)$.

**(c)** PD when $\vec{g}_0 = (0, 0, 0)$ and $\vec{g}_1 = (0, 0, 5)$.

**(d)** All points of the PDs for $\vec{g}_0 = (0, i, j)$ with $i, j \in \mathcal{I} = \left\{ \frac{n}{100} \mid 0 \le n \le 500, n \in \mathbb{N} \right\}$ when $\vec{g}_1 = (0, 0, 0)$.

**(e)** All points of the PDs for $\vec{g}_0 = (0, i, j)$ with $i, j \in \mathcal{I} = \left\{ \frac{n}{100} \mid 0 \le n \le 500, n \in \mathbb{N} \right\}$ when $\vec{g}_1 = (0, 0, 5)$.

**Figure 6.** Rips filtration and persistence diagrams (PDs) for various influence vectors for $\widehat{G^g}$.

### *3.6. Feature selection*

Feature selection is problem-dependent, and features should be properly categorized for analysis. Suppose that we have a time series $T$ and a collection $O$ of objects, including the various domain knowledge that we want to consider as features. To classify $O$ into the zeroth and first features, it is important to understand that the $i$th features of the featured time series data determine the weights of $i$-simplices in graphs. The vertex set of graphs consists of single observations $\{T(t)\}$, making it sensible to associate any feature $s_0 \in O$ closely related to a single observation with the zeroth feature. In addition, because the first features influence the weights of edges $\{T(t), T(t + 1)\}$ for some $t \in \mathbb{T}$, if any feature $s_1 \in O$ relates to any two observations simultaneously, it would be proper to regard it as the first feature.

In Example 3.5, humidity is considered the zeroth feature because it pertains to the information from a single observation $T(t)$ at each timestamp $t \in \mathbb{T}$. Furthermore, the magnitude of the temperature change is classified as the first feature because it is derived from two consecutive observations $T(t)$ and

$T(t+1)$ at each timestamp $t \in \mathbb{T}$.

## 4. Stability theorem for influence vectors

The stability theorem for persistence diagrams is crucial for ensuring reliable topological inferences.

**Lemma 4.1.** *(Cohen-Steiner et al. [53]) Let $\mathbb{X}$ be a finite abstract simplicial complex. For any two functions $h_1, h_2 : \mathbb{X} \to \mathbb{R}$, the persistence diagrams $dgm_p(h_1)$ and $dgm_p(h_2)$ satisfy*

$$D_B(dgm_p(h_1), dgm_p(h_2)) \leq \|h_1 - h_2\|_\infty$$

*for any dimension p, where $D_B$ is the bottleneck distance.*

If the change in the function $h : \mathbb{X} \to \mathbb{R}$ is small, then the change in the PD is also tiny. We prove that the PD remains stable when adjusting the influence vector $g$ for the featured time series $(\widehat{T}, g)$.

**Theorem 4.1.** *(Stability of the proposed method) Let $\widehat{T}$ be a time series with augmented feature information derived from a times series $T$. If the activation function $\rho$, which defines $\widehat{d}$ in Definition 3.4, is Lipschitz continuous, then there exists a constant $C = C(\widehat{T}, \rho) > 0$ such that for any two influence vectors $g$ and $g'$, the persistence diagrams $dgm_p(g)$ and $dgm_p(g')$ satisfy*

$$D_B(dgm_p(g), dgm_p(g')) \leq C\|g - g'\|_\infty$$

*for any dimension p, where $D_B$ is the bottleneck distance.*

The function $\Phi : g \mapsto \mathrm{dgm}_p(g)$ is Lipschitz continuous. For the count matrices $C_0 = (c^0_{ij})$ and $C_1 = (c^1_{ij})$, define $C^{\max}_0$ and $C^{\min}_1$, respectively, as follows:

$$C^{\max}_0 := \max_{v\in V} \sum_j c^0_{i_v j} \quad \text{and} \quad C^{\min}_1 := \min_{e\in E} \sum_j c^1_{i_e j}.$$

By the Lipschitz continuity of the activation function $\rho$, we can find a constant $k$ such that

$$|\rho(x) - \rho(y)| \leq k\,|x - y| \ \text{ for any } x, y.$$

The constant $C$ of Theorem 4.1 is

$$C = (kC^{\max}_0 + 1)\,\frac{2}{C^{\min}_1}(|V| - 1).$$

This theorem guarantees the robustness of the proposed method. This robustness ensures that small perturbations in the influence vectors, commonly encountered in real-world scenarios, do not lead to significant changes in the resulting persistence diagrams. Consequently, the stability property underscored by this theorem provides a reliable foundation for utilizing featured time series data in various applications, particularly in conditions where data input may be uncertain or noisy.

## 5. Applications and experiments

This section aims to demonstrate the potential applications of adjusting the influence vector in PH analysis of actual time series based on the main theorem introduced in Section 4. In Sections 5.1 and 5.2, we explore the analysis of stock and music data, respectively.

However, these analyses are not the main focus of this study. Instead, we aim to demonstrate that by using our proposed method of adjusting the appropriate influence vector when employing graph representations, it is possible to achieve outcomes comparable to the previous methods for some cases. Moreover, we highlight the possibility of conducting various other analyses that were not attempted in the previous research. The first example aims to improve the time series analysis by adjusting the influence vector and determining its value suitable for analysis. For this purpose, we apply our methodology to the stock data analyzed in previous PH studies. The second example focuses on analyzing the effect of features on the time series data. For this purpose, we apply the proposed method to the music data and observe how the PH results change when the influence assigned to each feature varies.

### *5.1. Application to stock data*

Recent studies in finance explore a wide range of topics. As [54] examined the interconnectedness between financial assets during the COVID-19 pandemic, similar research has studied the spillover effects between assets. Moreover, predicting financial assets remains a key area of research, where [55] introduces an enhanced artificial electric field algorithm (EOAEFA) that improves higher-order neural networks (HONNs) for more accurate financial time series prediction. Additionally, research utilizing high-frequency data to analyze rapidly evolving key trends is critical, and [56] provides a bibliometric analysis of high-frequency data in cryptocurrency research, identifying key trends and major research streams from 2015 to 2022. Conversely, studying signals of global financial changes over longer periods is also significant. Recently, [57] shows that TDA can detect early warning signals of financial bubbles, focusing on Bitcoin's historical price data using the log-periodic power law singularity (LPPLS) model. In this research, our goal is to enhance the precision through our proposed method for global anomaly detection. Specifically, this section aims to enhance the precision by adjusting the influence vector $g$ based on the stability described in Theorem 4.1, as applied to the global anomaly detection analysis performed in [26]. A previous study [26] employed four time series, specifically the Standard and Poor's 500 (S&P 500), the Dow Jones Industrial Average (DJIA), the National Association of Securities Dealers Automated Quotations (NASDAQ), and Russell 2000, and treated them as the 4-dimensional time series. In [26], the time series was transformed into a point cloud using the sliding window, and the persistence landscape [58] was computed. Calculating the $L^p$-norm of the persistence landscape yields a real number value for anomaly scores. Curves were created by advancing the sliding window stepwise and repeating the same process. In [26], these curves were used to detect anomalies such as the collapse of the Dotcom bubble on 03/10/2000 and the bankruptcy of Lehman Brothers on 09/15/2008.

For a given date $t$, define $P(t)$ as the closing price of a stock. Consider a time series $X$ defined as $X(t) = \log P(t+1) - \log P(t)$ for each date $t$. Because the closing price $P$ is already aggregated over time $t$, we further perform an aggregation on the price by partitioning the values of $X$ into 30 equal intervals, denoted as $\mathcal{I} = \{I_1, I_2, \ldots, I_{30}\}$. This aggregation process enhances the robustness of our analysis, ensuring reliable results even in the presence of noisy data. We obtain an aggregated time series $T : \mathbb{T} \to \mathcal{I}$ defined by $T(t) = I_{k_t}$ such that $X(t) \in I_{k_t}$ for any $t \in \mathbb{T}$. The zeroth feature ($T_f^0$) is chosen as the day of the week. For the first feature ($T_f^1$), we partition the set $\{X(t+1) - X(t) \mid t \in \mathbb{T}\}$ into 4 intervals of equal length, denoted as $\{J_1, J_2, J_3, J_4\}$. The time series $\widehat{T} = (T, T_f)$ is characterized as

(1) $T(t) = I_{k_t}$ such that $X(t) \in I_{k_t}$ for each date $t$
(2) The zeroth feature set $F_0$ used in $T_f$ is $\{\emptyset^0, \text{Mon}, \text{Tue}, \text{Wed}, \text{Thu}, \text{Fri}\}$
(3) The first feature set $F_1$ used in $T_f$ is $\{\emptyset^1, J_1, J_2, J_3, J_4\}$.

Let $\widehat{T^i}$ represent the time series obtained from the index $P^i$, where $P^i$ represents S&P 500, DJIA, NASDAQ, and Russell 2000, for $i = 1, 2, 3, 4$, respectively. For each influence vector $g$, we obtain the featured time series $(\widehat{T^i}, g)$. For a fixed date $t$, define $\widehat{T^i}|_t^w$ as the function obtained by slicing $\widehat{T^i}$ from $t$ to $t + w$, where $w$ is the window size. Let $\text{dgm}_1(\widehat{T^i}|_t^w)$ denote the one-dimensional PD obtained for the sliced time series $\widehat{T^i}|_t^w$.

We use the persistence landscape [58] as a tool to measure the PD, defined as follows: Let $\text{dgm}_p$ be a $p$-dimensional PD consisting of points $(b, d)$ where $b$ and $d$ represent the birth and death times of the homological features, respectively. To begin, transform each point $(b_m, d_m)$ in $\text{dgm}_p$ into a tent function, $\phi_m(x) = [\min\{x - b_m, d_m - x\}]^+$, where $[z]^+ = \max\{z, 0\}$ denotes the positive part of $z$. The persistence landscape of $\text{dgm}_p$ is defined as the sequence of functions $\lambda_k$, $k \geq 1$, where

$$\lambda_k(x) = k^{th} \text{ largest value of } \{\phi_m(x)\}_{m\geq 1}.$$

If there are fewer than $k$ functions $\phi_m(x)$ at a given point $x$, we set $\lambda_k(x) = 0$.

We define the anomaly score curve (ASC) of a single time series $P^i$ as follows: for each date $t$,

$$ASC(P^i)(t) = \frac{1}{M^i} \sum_{k=1}^{\infty} \|\lambda_k(\text{dgm}_1(\widehat{T^i}|_t^w))\|_\infty,$$

where $M^i$ is the normalization factor such that the maximum value of $ASC(P^i)(t)$ is 1 and $\|\cdot\|_\infty$ denotes the $L^\infty$-norm. The norm of the persistence landscape in the definition of the ASC curve is related to the area of the shaded region $S$ mentioned in Figure 6c. If $\text{dgm}_1$ contains only one point $(b, d)$, then we have $\sum_{k=1}^{\infty} \|\lambda_k(\text{dgm}_1(\cdot)\|_\infty = (d - b)/2 = \sqrt{\text{Area}(S)/2}$ because the area of $S$ is given by $(d - b)^2/2$.

The total anomaly score curve (TASC) of four time series $P^i$ is defined as follows: for each date $t$,

$$TASC(t) = \frac{1}{M} \prod_{i=1}^{4} ASC(P^i)(t),$$

where $M$ is the normalization factor such that the maximum value of $TASC(t)$ is 1. The rationale behind taking the product of the four $ASC(P^i)$ is to ensure that the total score approaches zero if any of them are close to zero. This is a cautious approach to minimize the error in the anomaly score. In the ASC and TASC curves, values closer to 1 indicate the presence of anomalies. Note that in the computation of TASC, the anomaly scores for each sliced time series $\widehat{T^i}|_t^w$ are computed independently for any $t$, facilitating parallel processing. This capability enables the efficient analysis of large datasets.

Experiments were conducted to identify the influence vector $g$ that enhances the anomaly detection performance. According to the stability theorem for influence vectors introduced in Section 4, slight variations in the influence vector $g$ cause subtle changes in the ASC and TASC. Therefore, the components of $g$ were altered to 0, 10, and 20 to observe significant changes. Because the zeroth and first features are always present in the stock data for each date $t$, the influence related to $\emptyset^0$ and $\emptyset^1$ was set to 0. In Figure 7, the date 03/10/2000 (denoted as $t_{Dot}$) is marked as the starting point of the collapse of the Dotcom bubble, and the shaded areas are colored according to the window size from the starting point. Similarly, 09/15/2008 (denoted as $t_{Leh}$) corresponds to the bankruptcy of Lehman Brothers. The values of $\vec{g}_0$ represent the influence values for $(\emptyset^0, \text{Mon}, \text{Tue}, \text{Wed}, \text{Thu}, \text{Fri})$, whereas those of $\vec{g}_1$ correspond to $(\emptyset^1, J_1^i, J_2^i, J_3^i, J_4^i)$.

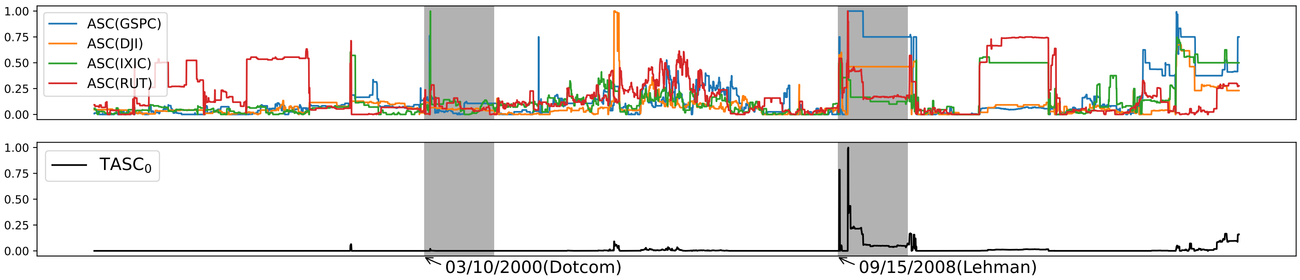


**Figure 7.** The first row presents the ASC for each of the S&P 500 (GSPC), DJIA (DJI), NASDAQ (IXIC), and Russell 2000 (RUT) indices, with $\vec{g}_0 = \vec{0}$, $\vec{g}_1 = \vec{0}$ and the window size $w = 360$. The second row shows the TASC$_0$ derived from these four ASCs.

To identify the optimal influence vector, the experiment employed a grid search strategy with a window size of $w = 360$ over $\vec{g}_0 = (0, a_1, 0, a_2, 0, a_3)$ and $\vec{g}_1 = (0, a_4, a_5, a_6, a_7)$, where $a_i \in \{0, 1, 2\}$ for all $i$. Table 1 summarizes the detection ratios for identifying the Dotcom and Lehman crises among the $3^7 = 2187$ possible $(\vec{g}_0, \vec{g}_1)$ pairs. In the table, Dot represents the set of $(\vec{g}_0, \vec{g}_1)$ pairs where the max value of the TASC curve exceeds the threshold $\epsilon = 0.9999$ within the interval $[t_{Dot}, t_{Dot} + w]$ and is less than 0.25 in all other regions. This can be formally expressed as $\max(TASC\ |_{[t_{Dot}, t_{Dot}+w]}) > \epsilon$, $\max(TASC\ |_{[-\infty, t_{Dot}]}) < 0.25$, and $\max(TASC\ |_{[t_{Dot}+w, t_{Dot}+2w]}) < 0.25$. Similarly, Leh represents the set of $(\vec{g}_0, \vec{g}_1)$ pairs where the max value of the TASC curve exceeds the threshold $\epsilon = 0.9999$ within the interval $[t_{Leh}, t_{Leh} + w]$ and is less than 0.25 in all other regions except the interval $[t_{Dot}, t_{Dot} + w]$. This is formally expressed as $\max(TASC\ |_{[t_{Leh}, t_{Leh}+w]}) > \epsilon$ and $\max(TASC\ |_{[-\infty, t_{Dot}]}) < 0.25$, $\max(TASC\ |_{[t_{Dot}+w, t_{Leh}]}) < 0.25$, and $\max(TASC\ |_{[t_{Leh}+w, t_{Leh}+2w]}) < 0.25$.

**Table 1.** Detection ratios for Dotcom and Lehman crises.

| | Dot | Leh | Dot ∩ Leh | Dot \ Leh | Leh \ Dot |
|---|---|---|---|---|---|
| Detect ratio (%) | 20.58 | 17.88 | 6.22 | 14.36 | 11.66 |

Dot: The set of $(\vec{g}_0, \vec{g}_1)$ pairs detecting the collapse of the Dotcom bubble as an anomaly; Leh: The set of $(\vec{g}_0, \vec{g}_1)$ pairs detecting the bankruptcy of Lehman Brothers as an anomaly, where $\vec{g}_0 = (0, a_1, 0, a_2, 0, a_3)$ and $\vec{g}_1 = (0, a_4, a_5, a_6, a_7)$, with $a_i \in \{0, 1, 2\}$.

As shown in the second row of Figure 7, the TASC$_0$ successfully detects the Lehman crisis but fails to detect the Dotcom bubble collapse. The pair $(\vec{g}_0, \vec{g}_1) \in Dot \cap Leh$, where $\vec{g}_0 = (0, 20, 0, 0, 0, 0)$ and $\vec{g}_1 = (0, 0, 10, 0, 20)$, successfully detects both the Dotcom bubble collapse and the Lehman Brothers crisis as anomalies. Figure 8 illustrates the normalized one-dimensional persistence diagrams for four stock indices at the peak anomaly day, denoted as $t_{\text{peak}}$, of TASC with $\vec{g}_0 = (0, 20, 0, 0, 0, 0)$ and

$\vec{g_1} = (0, 0, 10, 0, 20)$ in the Dotcom collapse region. Here, the normalized one-dimensional persistence diagram refers to $c \cdot \mathrm{dgm}_1$ with $c = 1/(M^i \sqrt[4]{M})$, where $M^i$ and $M$ are normalization factors defined for ASC and TASC, respectively, ensuring the maximum values are scaled to 1. This normalization stems from the definition of TASC, which is given by:

$$\begin{aligned} TASC(t) &= \frac{1}{M} \prod_{i=1}^{4} ASC(P^i)(t) = \prod_{i=1}^{4} \left( \frac{1}{M^i \sqrt[4]{M}} \sum_{k=1}^{\infty} \|\lambda_k(\mathrm{dgm}_1(\widehat{T^i}|_t^w))\|_\infty \right) \\ &= \prod_{i=1}^{4} \left( \sum_{k=1}^{\infty} \left\| \lambda_k \left( \frac{1}{M^i \sqrt[4]{M}} \mathrm{dgm}_1(\widehat{T^i}|_t^w) \right) \right\|_\infty \right). \end{aligned}$$

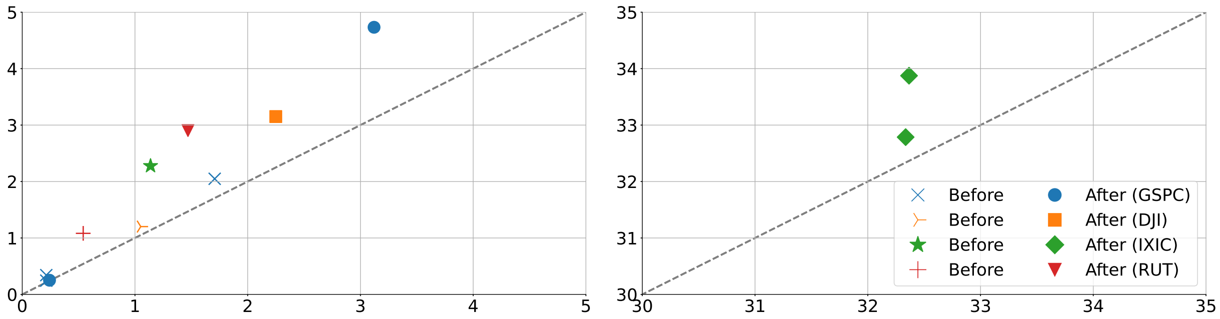


**Figure 8.** Normalized one-dimensional persistence diagrams for four stock indices at $t_{\text{peak}}$. (Before) Points when $\vec{g_0}$ and $\vec{g_1}$ are both zero vectors. (After) Points when $\vec{g_0} = (0, 20, 0, 0, 0, 0)$ and $\vec{g_1} = (0, 0, 10, 0, 20)$.

In Figure 8, the label 'Before' refers to the case where $(\vec{g_0}, \vec{g_1})$ are both zero vectors, while the label 'After' refers to the case where $\vec{g_0} = (0, 20, 0, 0, 0, 0)$ and $\vec{g_1} = (0, 0, 10, 0, 20)$. In the before case, the one-dimensional points for the four stock indices are close to the $y = x$ line, indicating a lower anomaly score, whereas in the after case, the points diverge further from the $y = x$ line, suggesting a higher anomaly score.

Figure 9 shows the TASC results for various window sizes $w$. For window sizes $w = 180, 210, 240,$ and 270, the TASCs fail to detect the Dotcom bubble. With larger window sizes of $300, 330,$ and $360,$ the TASCs identify the Dotcom bubble area. For window sizes of $w = 180, 210,$ and 240, high anomaly scores are observed in the regions starting at 11/26/2004 and 11/14/2014. Increasing the window size to $270, 300, 330,$ and 360 results in the disappearance of high anomaly scores in these regions. This suggests that the Dotcom and Lehman crises, which have led to a sustained global economic crisis, are predominantly measured when the window size is extended.

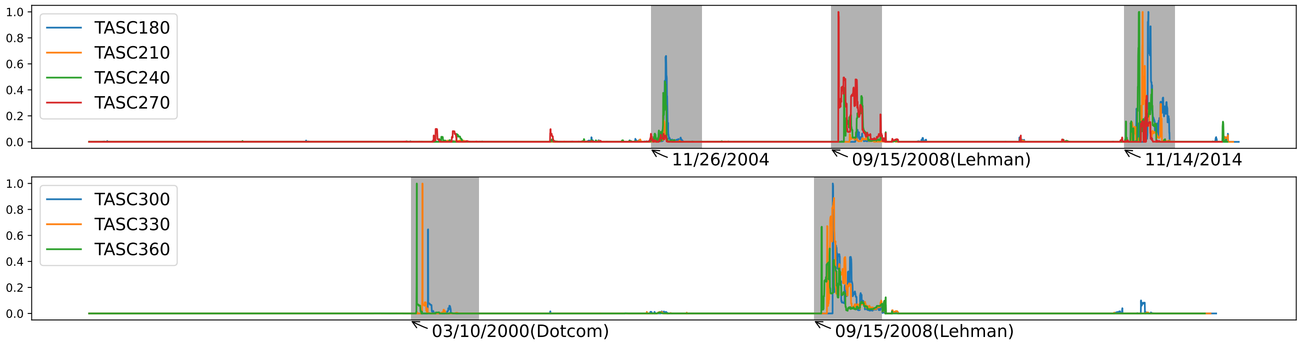


**Figure 9.** TASCs for varying window sizes $w$ with $\vec{g_0} = (0, 20, 0, 0, 0, 0)$ and $\vec{g_1} = (0, 0, 10, 0, 20)$ for each $w \in \{180, 210, 240, 270, 300, 330, 360\}$.

In the experiment discussed in Section 3.5, the adjustment of the zeroth feature did not significantly influence the birth of points, as shown in Figure 6d. However, the zeroth feature is crucial for anomaly detection in this section. Figure 10 illustrates the outcomes of the same experiment as in Figure 9 but with $\vec{g_0}$ set as the zero vector. The anomaly scores in the shaded areas resemble those in Figure 9, but noisier anomalies were observed in the mid-region, as shown in Figure 10. One approach to analyze the stock volatility is to examine the prices only for a specific day of the week and analyze them weekly. This approach reduces short-term, repetitive noise and focuses more on long-term trends and patterns. In our methodology, by setting $\vec{g_0} = (0, 20, 0, 0, 0, 0)$, we emphasize the information for Monday to enable a focused analysis of the effect of this particular day of the week. Similar results are observed when focusing on Tuesday, Wednesday, Thursday, and Friday. As demonstrated here, the proposed method generates various domain-specific and interpretable results by adjusting both $g_0$ and $g_1$.

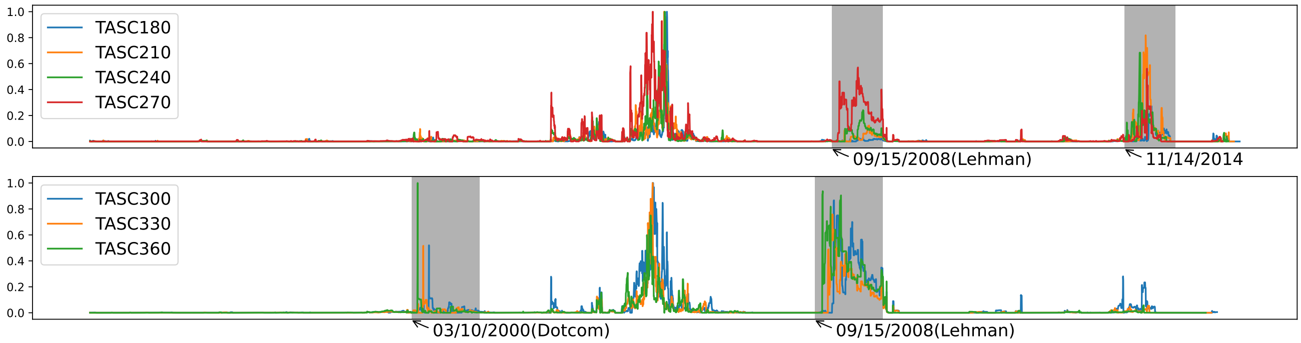


**Figure 10.** TASCs for varying window sizes $w$ with $\vec{g_0} = \vec{0}$ and $\vec{g_1} = (0, 0, 10, 0, 20)$ for each $w \in \{180, 210, 240, 270, 300, 330, 360\}$.

### 5.2. *Application to music data*

Music data is time series data containing information beyond pitch and rhythm. Although music is a challenging form of art to quantify, the authors of [34] analyzed the hidden topological information in traditional Korean music. We analyze the Celebrated Chop Waltz as a simple example, where each note is played simultaneously with both the right and left hands. Define the pair of pitches played simultaneously using the left and right hands as a *pitch pair*. We define the time series $T$ as $T(t_m) = (P(t_m), D(t_m))$, where $P(t_m)$ is a pitch pair and $D(t_m)$ is the playing length of each note played at the $m$th timestamp $t_m$. For instance, in the left part of Figure 11, the pitch pair $(F, G)$ is repeated six times with an eighth note duration.

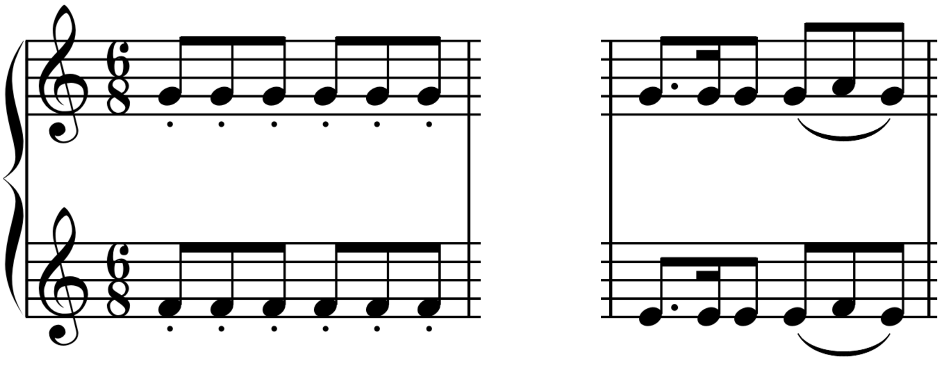


**Figure 11.** Musical notations used in the Celebrated Chop Waltz. (Left) When a staccato symbol is attached, the note is played briefly as if it were popping. (Right) If multiple notes are tied together with a slur symbol, these notes are played smoothly and connectedly.

As shown in Figure 11, the Celebrated Chop Waltz contains staccato and slur musical notations. We aim to understand how the changes in the influence of these musical notations affect the results of PH calculations. Staccato is a technique in which notes are played very briefly. Slur indicates that two distinct notes should be played as if they are connected smoothly. We define staccato related to a single note as the zeroth feature and slur related to two or more notes as the first feature. The time series $\widehat{T} = (T, T_f)$ is characterized as

(1) $T(t_m) = (P(t_m), D(t_m))$ at the timestamp $t_m$.
(2) The zeroth feature set $F_0$ used in $T_f$ is $\{\emptyset^0,\ \text{staccato}\}$.
(3) The first feature set $F_1$ used in $T_f$ is $\{\emptyset^1,\ \text{slur}\}$.

We conducted experiments to investigate the effects of changes in influence values on the resulting PDs. Figure 12 shows the topological information about the one-dimensional PD as influence values for the staccato and slur vary. In Figure 12, the $x$-axis represents the cases where $\vec{g_0}$ is $(0, x)$, which corresponds to the influence of the staccato being $x$. The $y$-axis signifies when $\vec{g_1}$ is $(0, y)$, indicating the value of the slur as $y$. Figures 12a and 12b plot the longest and shortest persistence among all points in $\mathrm{dgm}_1$, respectively. Here, the persistence is defined as $d - b$ for each point $(b, d) \in \mathrm{dgm}_1(\vec{g_0}, \vec{g_1})$. The function $z = z(x, y)$ in Figure 12a is

$$z(x, y) = \max\{d - b |\ (b, d) \in \mathrm{dgm}_1(\vec{g_0}, \vec{g_1})\} \text{ with } \vec{g_0} = (0, x) \text{ and } \vec{g_1} = (0, y)$$

and that in Figure 12b is

$$z(x, y) = \min\{d - b |\ (b, d) \in \mathrm{dgm}_1(\vec{g_0}, \vec{g_1})\} \text{ with } \vec{g_0} = (0, x) \text{ and } \vec{g_1} = (0, y).$$

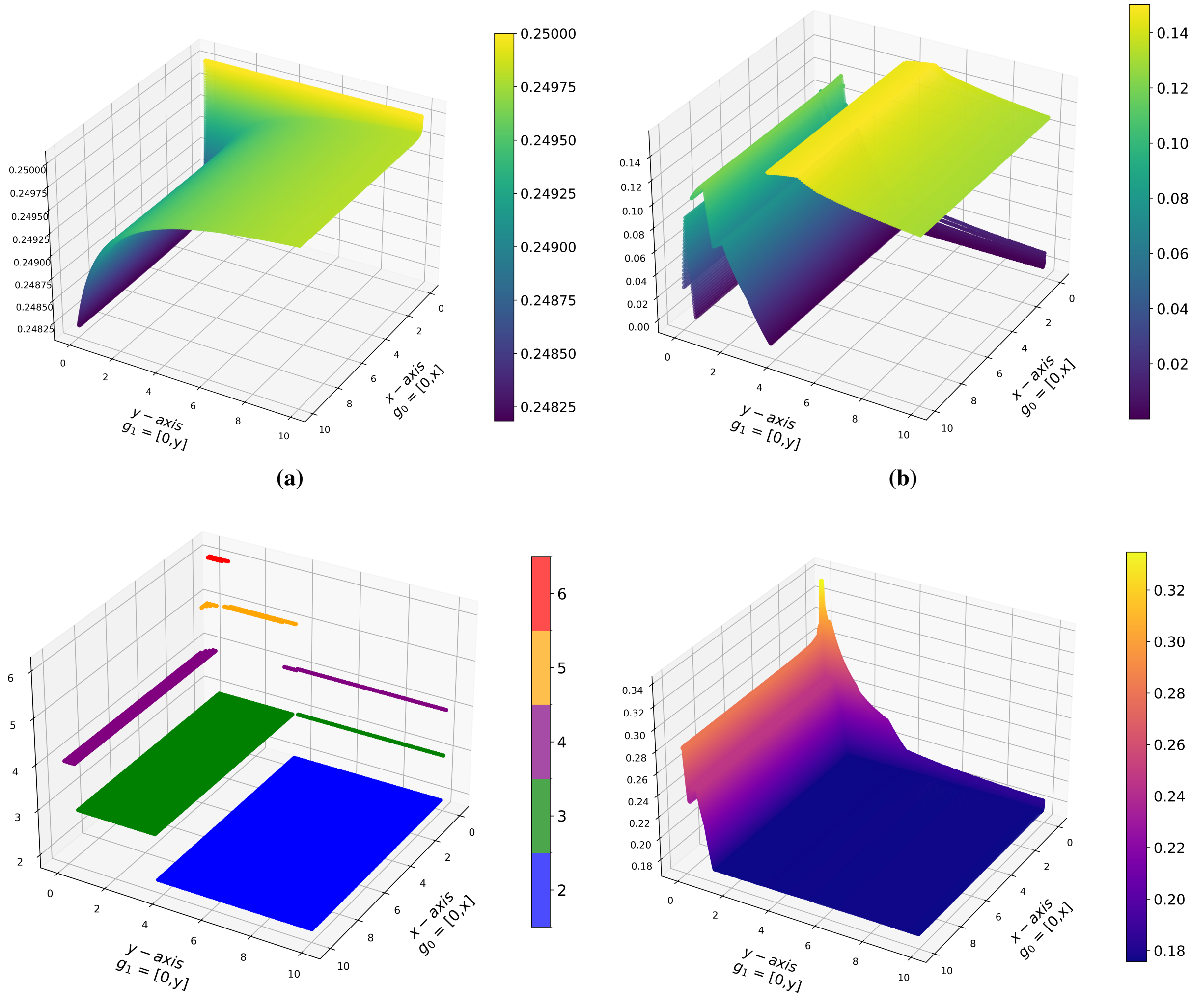
Figure 12c shows the total number of points in $\mathrm{dgm}_1(\vec{g_0}, \vec{g_1})$. The function $z = z(x, y)$ in Figure 12c is defined as

$$z(x, y) = |\mathrm{dgm}_1(\vec{g_0}, \vec{g_1})| \text{ with } \vec{g_0} = (0, x) \text{ and } \vec{g_1} = (0, y),$$

where $|\cdot|$ represents the number of elements in the multi-set. Figure 12d illustrates the sum of the $L^\infty$-norm of the $k$-th persistence landscape $\lambda_k$ of $\mathrm{dgm}_1(\vec{g_0}, \vec{g_1})$, which can be considered as information related to the overall total persistence. The function $z = z(x, y)$ in Figure 12d is

$$z(x,y) = \sum_{k=1}^{\infty} \|\lambda_k(\mathrm{dgm}_1(\vec{g_0}, \vec{g_1}))\|_\infty \text{ with } \vec{g_0} = (0, x) \text{ and } \vec{g_1} = (0, y).$$

The continuity of the surface observed in Figure 12d is a consequence of Theorem 4.1.



**(a)**

**(b)**

**(c)**

**(d)**

**Figure 12.** Topological information about the one-dimensional persistence diagrams (PDs) while the influence vectors $g$ of the staccato and slur are varied. (a) The longest persistence among the elements of the one-dimensional PD as $g$ is varied. (b) The shortest persistence among the elements of the one-dimensional PD as $g$ is varied. (c) The total number of points in the one-dimensional PD as $g$ is varied. (d) The $L^\infty$-norm of the landscape of the one-dimensional PD as $g$ is varied.

From Figure 12c, we observe that when $(x, y) = (0, 0)$, the number of points in $\mathrm{dgm}_1$ is the highest. However, as $x$ and $y$ increase, the number of points gradually decreases. If $x$ and $y$ are sufficiently large, only the two points with longer persistence remain. The disappearing points with smaller persistence cause the discontinuities shown in Figure 12b. In Figures 12a and 12b, the variation in the shortest persistence is greater than that in the longest persistence. Therefore, the changes in Figure 12d are primarily due to the variation of the points with shorter persistence. This suggests that points with shorter persistence can provide important information about the overall topological structure of the music. The authors of [34] also considered all points representing the music's hidden structure as important, regardless of the persistence. From this perspective, when the influences of staccato and slur are zero, the highest number of hidden topological structures of the music are observed. However, when the influences of staccato and slur are sufficiently large, only two structures remain. In summary, starting with six points when $(x, y) = (0, 0)$, changes in the $x$ and $y$ values cause points with relatively shorter persistence to vary dramatically, with up to four disappearances. This illustrates how the overall PH outcome changes with variations in the influence of staccato and slur.

In [34], the authors analyzed the hidden topological structures in music using the cycle representatives of each point in $\mathrm{dgm}_1$. They introduced the overlapping percentage to quantify the topological repetitiveness in music. Assume that $\mathrm{dgm}_1$ contains $m$ points $[b_1, d_1], \ldots, [b_m, d_m]$. For each point $[b_i, d_i]$, we fix a corresponding one-dimensional cycle representative $\sigma_i \subseteq V$. Although $\sigma_i$ is generally not unique, the authors of [34] used the JavaPlex algorithm [47] to select $\sigma_i$. The chosen cycle representatives $\sigma_i$ correspond closely with the 2-simplex, which directly results in the disappearance of the 1-cycle. For a music time series $T : \mathbb{T} \to \mathcal{X}$, define

$$N_c = \left|\{t \in \mathbb{T} \mid T(t) \in \bigcup_{i=1}^{m} \sigma_i\}\right| \text{ and } N_s = \left|\{t \in \mathbb{T} \mid T(t) \in \bigcup_{1 \le i < j \le m} (\sigma_i \cap \sigma_j)\}\right|.$$

Then, the overlapping percentage is defined as $(N_s/N_c) \times 100(\%)$. This value represents the percentage of timestamps in a time series that belongs simultaneously to multiple cycle representatives. Figure 13 shows the overlapping percentage obtained by varying the influence vector $g$. In Figure 13, the overlapping percentage increases for the staccato influence $x$ when $x > 0$ rather than $x = 0$. This suggests that the presence of staccato adds greater topological repetitiveness than its absence; however, increasing the value does not necessarily enhance it further. Moreover, as the influence of the slur $y$ increases, the overlapping percentage decreases. Slurs appear less frequently in scores than staccato. Emphasizing the slurs gradually decreases the overlapping percentage, reducing the topological repetitiveness.

In this section, we described the effect of adjusting the influence vector $g$ on the overall PH outcome. This analysis allows us to examine the impact of features on the time series. As shown in this section, the proposed method yields a more flexible analysis by varying $g$ from the music data.

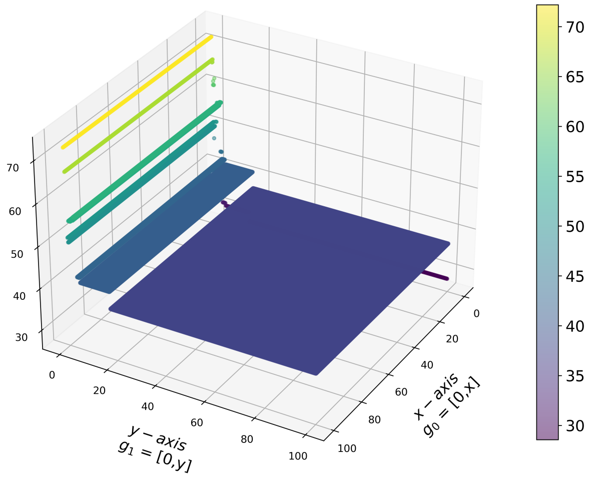


**Figure 13.** The z-axis represents the overlapping percentage for each influence vector.

## 6. Proof of the stability theorem for influence vectors

This section is devoted to the proof of Theorem 4.1.

*Proof.* Consider any featured time series $(\widehat{T}, g)$. We append $g$ to all notations if they are variables dependent on $g$. Consider the connected graph $\widehat{G}^g = (V, E, \widehat{W}_V^g, \widehat{W}_E^g)$ and the metric space $(V, \widehat{d}^g)$. Recall that in Section 2.3, we defined the filtration value function $h_g : \mathbb{X} \to \mathbb{R}$, where $\mathbb{X} = \mathcal{P}(V)$. By Lemma 4.1, we have $D_B\big(\mathrm{dgm}_p(g), \mathrm{dgm}_p(g')\big) \le \|h_g - h_{g'}\|_\infty$ for any two influence vectors $g$ and $g'$. Therefore, it suffices to show that there exists a constant $C$ independent of $g$ and $g'$, satisfying $\|h_g - h_{g'}\|_\infty \le C\|g - g'\|_\infty$.

Since $\mathbb{X}$ is finite, we have $\|h_g - h_{g'}\|_\infty = |(h_g - h_{g'})(\sigma_*)|$ for a $p$-simplex $\sigma_* \in \mathbb{X}$. By the maximality of $h_g$ and $h_{g'}$, $h_g(\sigma_*) = \widehat{d}^g(v_g, w_g)$ and $h_{g'}(\sigma_*) = \widehat{d}^{g'}(v_{g'}, w_{g'})$ for edges $(v_g, w_g), (v_{g'}, w_{g'}) \subseteq \sigma_*$. Consider an interpolation map $q(t) = (1-t)g + tg'$ for a real number $t$, where $q(0) = g$ and $q(1) = g'$. Define a function $H(t) = \widehat{d}^{q(t)}(v_g, w_g) - \widehat{d}^{q(t)}(v_{g'}, w_{g'})$, where $\widehat{d}^{q(t)}$ is the distance in the graph $\widehat{G}^{q(t)}$, as defined in Definition 3.4. By the maximality of the functions $h_g$ and $h_{g'}$, we have $H(0) \ge 0$ and $H(1) \le 0$. Because $H$ is continuous on the closed interval $[0, 1]$, by the intermediate value theorem, there exists a real number $t_0 \in [0, 1]$ such that $H(t_0) = 0$. If we denote $q(t_0)$ as $g^*$, we have

$$\begin{aligned}
|(h_g - h_{g'})(\sigma_*)| &= |\widehat{d}^g(v_g, w_g) - \widehat{d}^{g'}(v_{g'}, w_{g'})| \\
&\le |\widehat{d}^g(v_g, w_g) - \widehat{d}^{g^*}(v_g, w_g)| + |\widehat{d}^{g^*}(v_{g'}, w_{g'}) - \widehat{d}^{g'}(v_{g'}, w_{g'})| \\
&= |h_g - h_{g^*}|((v_g, w_g)) + |h_{g^*} - h_{g'}|((v_{g'}, w_{g'})).
\end{aligned}$$

If we can prove there exists a constant $C$ such that $|h_{g_1} - h_{g_2}|(e) \le C\|g_1 - g_2\|_\infty$ for any influence vectors $g_1, g_2$ and edge $e \in E$, then we have

$$\begin{aligned}
|h_g - h_{g^*}|((v_g, w_g)) + |h_{g^*} - h_{g'}|((v_{g'}, w_{g'})) &\le C\|g - g^*\|_\infty + C\|g' - g^*\|_\infty \\
&= Ct_0\|g - g'\|_\infty + C(1 - t_0)\|g - g'\|_\infty \\
&= C\|g - g'\|_\infty,
\end{aligned}$$

which completes the proof. Therefore, it suffices to show that for any influence vectors $g$, $g'$ and edge $e \in E$, there exists a constant $C$ such that $|h_g - h_{g'}|(e) \leq C\|g - g'\|_\infty$.

Take any influence vectors $g$, $g'$ and edge $e_* = (v, w) \in E$. Consider an interpolation map $q(t) = (1-t)g + tg'$ for the real number $t$ such that $q(0) = g$ and $q(1) = g'$. For any influence vector $q(t)$, we have $\widehat{d}^{q(t)}(v, w) = \sum_{e \in p^{q(t)}} L^{q(t)}(e)$ for some path $p^{q(t)}$ from $v$ to $w$, as described in Definition 3.4. Define a function $K(t) = \sum_{e \in p^g} L^{q(t)}(e) - \sum_{e \in p^{g'}} L^{q(t)}(e)$. By the minimality of paths $p^g$ and $p^{g'}$, we deduce $K(0) \leq 0$ and $K(1) \geq 0$. By the intermediate value theorem, there exists $t_1 \in [0, 1]$ such that $K(t_1) = 0$. If we denote $q(t_1)$ as $g^*$, then

$$\begin{aligned}
|(h_g - h_{g'})(e_*)| =& \left|\widehat{d^g}(v, w) - \widehat{d^g}'(v, w)\right| = \Big|\sum_{e \in p^g} L^g(e) - \sum_{e \in p^{g'}} L^{g'}(e)\Big| \\
=& \Big|\sum_{e \in p^g} L^g(e) - \sum_{e \in p^g} L^{g*}(e) + \sum_{e \in p^{g'}} L^{g*}(e) - \sum_{e \in p^{g'}} L^{g'}(e)\Big| \\
\leq& \sum_{e \in p^g} |L^g(e) - L^{g*}(e)| + \sum_{e \in p^{g'}} |L^{g*}(e) - L^{g'}(e)|.
\end{aligned} \tag{6.1}$$

Consider the first summation in inequality (6.1) as follows:

$$\begin{aligned}
&\sum_{e \in p^g} |L^g(e) - L^{g*}(e)| \\
&= \sum_{e \in p^g} \left|(\widehat{W}^g_E(e))^{-1} - (\widehat{W}^{g^*}_E(e))^{-1} - \alpha^g(\rho(\widehat{W}^g_V(a) + \widehat{W}^g_V(b)) + \alpha^{g^*}(\rho(\widehat{W}^{g^*}_V(a) + \widehat{W}^{g^*}_V(b))\right| \\
&\leq \sum_{e \in p^g} \left|(\widehat{W}^g_E(e))^{-1} - (\widehat{W}^{g^*}_E(e))^{-1}\right| + \left|\alpha^g(\rho(\widehat{W}^g_V(a) + \widehat{W}^g_V(b)) - \alpha^{g^*}(\rho(\widehat{W}^{g^*}_V(a) + \widehat{W}^{g^*}_V(b))\right|.
\end{aligned}$$

We can express $(\widehat{W}^g_E(e))^{-1} = 1/\left(\sum_j (g_1(s_j) + 1)c^1_{i_e j}\right)$, where $i_e$ designates the position of $e$ in the rows of the first count matrix $C_1 = (c^1_{ij})$. Observe that

$$\begin{aligned}
\sum_{e \in p^g} \left|(\widehat{W}^g_E(e))^{-1} - (\widehat{W}^{g^*}_E(e))^{-1}\right| =& \sum_{e \in p^g} \left|\frac{1}{\sum_j (g_1(s_j) + 1)c^1_{i_e j}} - \frac{1}{\sum_j (g^*_1(s_j) + 1)c^1_{i_e j}}\right| \\
=& \sum_{e \in p^g} \left|\frac{\sum_j c^1_{i_e j}(g_1(s_j) - g^*_1(s_j))}{(\sum_j (g_1(s_j) + 1)c^1_{i_e j})(\sum_j (g^*_1(s_j) + 1)c^1_{i_e j})}\right| \\
\leq& \sum_{e \in p^g} \left|\frac{\sum_j c^1_{i_e j}(g_1(s_j) - g^*_1(s_j))}{(\sum_j c^1_{i_e j})(\sum_j c^1_{i_e j})}\right| \\
\leq& \sum_{e \in p^g} \left|\frac{(\sum_j c^1_{i_e j})}{(\sum_j c^1_{i_e j})(\sum_j c^1_{i_e j})}\right| \|g - g^*\|_\infty \\
\leq& t_1 \sum_{e \in p^g} \left(\frac{1}{(\sum_j c^1_{i_e j})}\right) \|g - g'\|_\infty \quad \text{by } g - g^* = t_1(g - g') \\
\leq& t_1 \sum_{e \in p^g} \left(\frac{1}{(\sum_j c^1_{i_e j})}\right) \|g - g'\|_\infty
\end{aligned}$$

$$
\begin{aligned}
&\leq t_1 \sum_{e\in p^g} \left(\frac{1}{C_1^{\min}}\right) \|g-g'\|_\infty \quad \text{by } C_1^{\min} = \min_{e\in E} \sum_j c^1_{i_e j} \\
&\leq t_1 \frac{1}{C_1^{\min}} \|g-g'\|_\infty (|V|-1).
\end{aligned} \tag{6.2}
$$

In the last inequality (6.2), we utilize the fact that the number of edges in a path $p$, without repeated vertices, is precisely $|V_p| - 1$, where $|V_p|$ denotes the number of vertices in $p$. The path $p^g$ does not contain any repeated vertices owing to its minimality in the definition of the distance $d$.

We demonstrate that $\alpha^g$ is bounded for any given $g$; recall that $\alpha^g = \min_{e\in E}(\widehat{W}^g_E(e))^{-1}$. Given that all elements of an influence vector are nonnegative,

$$
\sum_j (g_1(s_j)+1)c^1_{i_e j} \geq \sum_j c^1_{i_e j} \geq \min_{e\in E}\{\sum_j c^1_{i_e j}\} = C_1^{\min}
$$

holds for any edge $e \in E$. Therefore, we can assert that $\alpha^g \leq \frac{1}{C_1^{\min}}$ for any $g$.

**Lemma 6.1.** *Given any pair of influence vectors g and g′, it holds true that* $|\alpha^g - \alpha^{g'}| \leq \frac{1}{C_1^{min}} \|g-g'\|_\infty$ *.*

*Proof.* Observe that for any edge $e \in E$,

$$
\begin{aligned}
|(\widehat{W}^g_E(e))^{-1} - (\widehat{W}^{g'}_E(e))^{-1}| &= \left| \frac{\sum_j c^1_{i_e j}(g_1 - g'_1)(t_j)}{\left(\sum_j (g_1(s_j)+1)c^1_{i_e j}\right)\left(\sum_j (g'_1(t_j)+1)c^1_{i_e j}\right)} \right| \\
&\leq \left| \frac{\left(\sum_j c^1_{i_e j}\right)\|g-g'\|_\infty}{\left(\sum_j c^1_{i_e j}\right)\left(\sum_j c^1_{i_e j}\right)} \right| = \frac{1}{\left(\sum_j c^1_{i_e j}\right)} \|g-g'\|_\infty \\
&\leq \frac{1}{C_1^{\min}} \|g-g'\|_\infty .
\end{aligned} \tag{6.3}
$$

Assume that $|\alpha^g - \alpha^{g'}| > \frac{1}{C_1^{\min}}\|g-g'\|_\infty$. Without a loss of generality, suppose $\alpha^g > \alpha^{g'}$. Because the set of edges $E$ is finite, there exist edges $e_1, e_2 \in E$ such that $\alpha^g = (\widehat{W}^g_E(e_1))^{-1}$ and $\alpha^{g'} = (\widehat{W}^{g'}_E(e_2))^{-1}$. From the inequality 6.3, we derive that $|\alpha^{g'} - (\widehat{W}^g_E(e_2))^{-1}| \leq \frac{1}{C_1^{\min}}\|g-g'\|_\infty$, leading to $(\widehat{W}^g_E(e_2))^{-1} \leq \alpha^{g'} + \frac{1}{C_1^{\min}}\|g-g'\|_\infty$. However, by assumption, we obtain $\alpha^{g'} + \frac{1}{C_1^{\min}}\|g-g'\|_\infty < \alpha^g$. This implies $(\widehat{W}^g_E(e_2))^{-1} < \alpha^g$, which contradicts the definition of $\alpha^g$ as the minimum value across all edges $e \in E$ for a given $g$. □

For any edge $e = (a, b)$, if we write $\rho(\widehat{W}^g_V(a) + \widehat{W}^g_V(b))$ as $D(g, e)$, then

$$
\sum_{e\in p^g} \left| \alpha^g(\rho(\widehat{W}^g_V(a) + \widehat{W}^g_V(b)) - \alpha^{g^*}(\rho(\widehat{W}^{g^*}_V(a) + \widehat{W}^{g^*}_V(b)) \right| = \sum_{e\in p^g} \left| \alpha^g D(g,e) - \alpha^{g^*} D(g^*, e) \right|.
$$

The vertex weight $\widehat{W}^g_V(v)$ can be expressed as $\widehat{W}^g_V(v) = \sum_j g_0(r_j)c^0_{i_v j}$, where $i_v$ designates the position of vertex $v$ in the rows of the zeroth count matrix $C_0 = (c^0_{ij})$. Because $\rho$ is Lipschitz continuous, there exists a positive constant $k$ such that $|\rho(z) - \rho(z')| \leq k|z - z'|$ for any two real numbers $z$ and $z'$. Hence,

$$
|D(g,e) - D(g^*, e)| = \left| \rho(\widehat{W}^g_V(a) + \widehat{W}^g_V(b)) - \rho(\widehat{W}^{g^*}_V(a) + \widehat{W}^{g^*}_V(b)) \right|
$$

$$
\begin{aligned}
&\le k\left|(\widehat{W}_V^g(a)+\widehat{W}_V^g(b))-(\widehat{W}_V^{g^*}(a)+\widehat{W}_V^{g^*}(b))\right|\\
&= k\left|\left(\sum_j c_{i_a j}^0(g_0-g_0^*)(s_j)+\sum_j c_{i_b j}^0(g_0-g_0^*)(s_j)\right)\right|\\
&\le k\|g-g^*\|_\infty \sum_j (c_{i_a j}^0 + c_{i_b j}^0)\\
&\le 2kC_0^{\max}\|g-g^*\|_\infty \quad \text{by } C_0^{\max} = \max_{v\in V}\sum_j c_{i_v j}^0.
\end{aligned}
$$

Therefore, we have

$$
\begin{aligned}
\sum_{e\in p^g}\left|\alpha^g D(g,e)-\alpha^{g^*}D(g^*,e)\right| &= \sum_{e\in p^g}\left|\alpha^g D(g,e)-\alpha^g D(g^*,e)+\alpha^g D(g^*,e)-\alpha^{g^*}D(g^*,e)\right|\\
&\le \sum_{e\in p^g}|\alpha^g|\,|D(g,e)-D(g^*,e)| + \sum_{e\in p^g}|\alpha^g-\alpha^{g^*}|\,|D(g^*,e)|\\
&\le \sum_{e\in p^g}\frac{1}{C_1^{\min}}\,|D(g,e)-D(g^*,e)| + \sum_{e\in p^g}|\alpha^g-\alpha^{g^*}|\\
&\le \sum_{e\in p^g}\frac{2}{C_1^{\min}}kC_0^{\max}\|g-g^*\|_\infty + \sum_{e\in p^g}\frac{1}{C_1^{\min}}\|g-g^*\|_\infty\\
&= (2kC_0^{\max}+1)\frac{\|g-g^*\|_\infty}{C_1^{\min}}\sum_{e\in p^g}1\\
&\le (2kC_0^{\max}+1)\frac{\|g-g^*\|_\infty}{C_1^{\min}}(|V|-1)\\
&= t_1(2kC_0^{\max}+1)\frac{\|g-g'\|_\infty}{C_1^{\min}}(|V|-1) \quad \text{by } g-g^* = t_1(g-g'). \qquad (6.4)
\end{aligned}
$$

If we set a constant $C := (2kC_0^{\max}+1)\frac{1}{C_1^{\min}}(|V|-1)$, then by the inequalities (6.2) and (6.4), we have

$$
\begin{aligned}
&\sum_{e\in p^g}|L^g(e)-L^{g*}(e)|\\
&\le \sum_{e\in p^g}\left|(\widehat{W}_E^g(e))^{-1}-(\widehat{W}_E^{g^*}(e))^{-1}\right| + \left|\alpha^g(\rho(\widehat{W}_V^g(a)+\widehat{W}_V^g(b))-\alpha^{g^*}(\rho(\widehat{W}_V^{g^*}(a)+\widehat{W}_V^{g^*}(b))\right|\\
&\le t_1\frac{1}{C_1^{\min}}\|g-g'\|_\infty(|V|-1) + t_1(2kC_0^{\max}+1)\frac{\|g-g'\|_\infty}{C_1^{\min}}(|V|-1)\\
&= t_1C\|g-g'\|_\infty.
\end{aligned}
$$

Similarly, we have

$$\sum_{e\in p^g}\left|L^{g*}(e)-L^{g'}(e)\right| \le (1-t_1)C\|g-g'\|_\infty.$$

Therefore, by the inequality (6.1), we conclude

$$|(h_g-h_{g'})(e_*)| \le t_1C\|g-g'\|_\infty + (1-t_1)C\|g-g'\|_\infty = C\|g-g'\|_\infty,$$

which completes the proof.

□

## 7. Conclusions

In this paper, we introduced a novel concept of the featured time series data and proposed a method based on persistent homology with varying influence vectors. With the featured time series and influence vectors, the proposed method allows customizing PH calculations to reflect domain-specific knowledge effectively. Our approach enhances the adaptability of domain-specific PH analysis across different types of time series data and ensures the stability of the PH calculations. The application of our methodology to real-world datasets, such as stock data for anomaly detection and musical data for feature impact analysis, has shown promising results. These applications demonstrate the practical feasibility and effectiveness of our proposed method. However, there remain limitations within applications that necessitate further improvements. The selection of an optimal influence vector depends on a grid search method, as a specialized algorithm for determining the optimal influence vector has not yet been developed. Additionally, the computational complexity of persistent homology calculations presents significant challenges for real-time analysis. Future research will focus on extending this framework to other applications and exploring the optimization method of the influence vectors that can further refine the accuracy and applicability of PH under various settings.

## Author contributions

The first author conceived the proposed idea and conducted analytical and numerical calculations. The second author supervised the project. The first and second authors contributed to the final version of the manuscript.

## Acknowledgments

This work was supported by NRF of Korea under the grant number 2021R1A2C3009648 and POSTECH Basic Science Research Institute under the NRF grant number NRF2021R1A6A1A1004294412. The work is also supported partially by NRF grant by the Korea government (MSIT) (RS-2023-00219980).

## Conflict of interest

The authors declare that they have no conflicts of interest.

## References

1. M. Braei, S. Wagner, Anomaly detection in univariate time-series: A survey on the state-of-the-art, 2020. https://doi.org/10.48550/arXiv.2004.00433
2. A. Blázquez-García, A. Conde, U. Mori, J. A Lozano, A review on outlier/anomaly detection in time series data, *ACM Comput. Surv.*, **54** (2021), 1–33. https://doi.org/10.1145/3444690
3. B. Lim, S. Zohren, Time-series forecasting with deep learning: a survey, *Philos. T. R. Soc. A*, **379** (2021), 20200209. https://doi.org/10.1098/rsta.2020.0209

4. O. B. Sezer, M. U. Gudelek, A. M. Ozbayoglu, Financial time series forecasting with deep learning: A systematic literature review: 2005–2019, *Appl. soft comput.*, **90** (2020), 106181. https://doi.org/10.1016/j.asoc.2020.106181

5. J. F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, A. Troncoso, Deep learning for time series forecasting: a survey, *Big Data*, **9** (2021), 3–21. https://doi.org/10.1089/big.2020.0159

6. P. Lara-Benítez, M. Carranza-García, J. C. Riquelme, An experimental review on deep learning architectures for time series forecasting, *Int. J. Neural Syst.*, **31** (2021), 2130001. https://doi.org/10.1142/S0129065721300011

7. A. Bagnall, J. Lines, A. Bostrom, J. Large, E. Keogh, The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances, *Data Min. Knowl. Disc.*, **31** (2017), 606–660. https://doi.org/10.1007/s10618-016-0483-9

8. G. A. Susto, A. Cenedese, M. Terzi, Time-series classification methods: Review and applications to power systems data, *Big Data Appl. Power Syst.*, **2018** (2018), 179–220. https://doi.org/10.1016/B978-0-12-811968-6.00009-7

9. H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, P. A. Muller, Deep learning for time series classification: a review, *Data Min. Knowl. Disc.*, **33** (2019), 917–963. https://doi.org/10.1007/s10618-019-00619-1

10. A. Abanda, U. Mori, J. A Lozano, A review on distance based time series classification, *Data Min. Knowl. Dis.*, **33** (2019), 378–412. https://doi.org/10.1007/s10618-018-0596-4

11. S. Aghabozorgi, A. S. Shirkhorshidi, T. Y. Wah, Time-series clustering-a decade review, *Inform. syst.*, **53** (2015), 16–38. https://doi.org/10.1016/j.is.2015.04.007

12. E. A. Maharaj, P. D'Urso, J. Caiado, *Time series clustering and classification*, Chapman and Hall/CRC, 2019. https://doi.org/10.1201/9780429058264

13. M. Ali, A. Alqahtani, M. W. Jones, X. Xie, Clustering and classification for time series data in visual analytics: A survey, *IEEE Access*, **7** (2019), 181314–181338. https://doi.org/10.1109/ACCESS.2019.2958551

14. N. H. Packard, J. P. Crutchfield, J. D. Farmer, R. S. Shaw, Geometry from a time series, *Phys. Rev. Lett.*, **45** (1980), 712. https://doi.org/10.1103/PhysRevLett.45.712

15. F. Takens, Detecting strange attractors in turbulence, In: *Dynamical Systems and Turbulence, Warwick 1980*, Berlin: Springer, 1981. https://doi.org/10.1007/BFb0091924

16. P. Skraba, V. De Silva, M. Vejdemo-Johansson, Topological analysis of recurrent systems, In: *NIPS 2012 Workshop on Algebraic Topology and Machine Learning*, 2012, 1–5.

17. J. A. Perea, J. Harer, Sliding windows and persistence: An application of topological methods to signal analysis, *Found. Comput. Math.*, **15** (2015), 799–838. https://doi.org/10.1007/s10208-014-9206-z

18. J. A. Perea, Persistent homology of toroidal sliding window embeddings, In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, 6435–6439. https://doi.org/10.1109/ICASSP.2016.7472916

19. J. Berwald, M. Gidea, Critical transitions in a model of a genetic regulatory system, 2013. https://doi.org/10.48550/arXiv.1309.7919

20. M. Scheffer, J. Bascompte, W. A. Brock, V. Brovkin, S. R. Carpenter, V. Dakos, et al., Early-warning signals for critical transitions, *Nature*, **461** (2009), 53–59. https://doi.org/10.1038/nature08227

21. F. A. Khasawneh, E. Munch, Stability determination in turning using persistent homology and time series analysis, In: *Proceedings of the ASME 2014 International Mechanical Engineering Congress and Exposition*, 2014, 14–20. https://doi.org/10.1115/IMECE2014-40221

22. C. M. M. Pereira, R. F. de Mello, Persistent homology for time series and spatial data clustering, *Expert Syst. Appl.*, **42** (2015), 6026–6038. https://doi.org/10.1016/j.eswa.2015.04.010

23. Q. H. Tran, Y. Hasegawa, Topological time-series analysis with delay-variant embedding, *Phys. Rev. E*, **99** (2019), 032209. https://doi.org/10.1103/PhysRevE.99.032209

24. C. Wu, C. A. Hargreaves, Topological machine learning for multivariate time series, *J. Exp. Theor. Artif. In.*, **34** (2022), 311–326. https://doi.org/10.1080/0952813X.2021.1871971

25. S. Emrani, T. Gentimis, H. Krim, Persistent homology of delay embeddings and its application to wheeze detection, *IEEE Signal Proc. Let.*, **21** (2014), 459–463. https://doi.org/10.1109/LSP.2014.2305700

26. M. Gidea, Y. Katz, Topological data analysis of financial time series: Landscapes of crashes, *Physica A*, **491** (2018), 820–834. https://doi.org/10.1016/j.physa.2017.09.028

27. M. S. Ismail, M. S. Md Noorani, M. Ismail, F. A. Razak, M. A. Alias, Early warning signals of financial crises using persistent homology, *Physica A*, **586** (2022), 126459. https://doi.org/10.1016/j.physa.2021.126459

28. V. Venkataraman, K. N. Ramamurthy, P. Turaga, Persistent homology of attractors for action recognition, In: *2016 IEEE international conference on image processing (ICIP)*, 2016, 4150–4154.. https://doi.org/10.1109/ICIP.2016.7533141

29. J. A Perea, A. Deckard, S. B. Haase, J. Harer, Sw1pers: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data, *BMC Bioinformatics*, **16** (2015), 257. https://doi.org/10.1186/s12859-015-0645-6

30. B. J. Stolz, H. A. Harrington, M. A. Porter, Persistent homology of time-dependent functional networks constructed from coupled time series, *Chaos*, **27** (2017), 047410. https://doi.org/10.1063/1.4978997

31. M. Gidea, Topological data analysis of critical transitions in financial networks, In: *3rd International Winter School and Conference on Network Science*, 2017, 47–59. https://doi.org/10.1007/978-3-319-55471-6_5

32. S. Gholizadeh, A. Seyeditabari, W. Zadrozny, Topological signature of 19th century novelists: Persistent homology in text mining, *Big Data Cogn. Comput.*, **2** (2018), 33. https://doi.org/10.3390/bdcc2040033

33. M. E. Aktas, E. Akbas, J. Papayik, Y. Kovankaya, Classification of turkish makam music: a topological approach, *J. Math. Music*, **13** (2019), 135–149. https://doi.org/10.1080/17459737.2019.1622810

34. M. L. Tran, C. Park, J. H. Jung, Topological data analysis of korean music in jeongganbo: a cycle structure, *J. Math. Music*, **17** (2023), 403–432. https://doi.org/10.1080/17459737.2022.2164626

35. M. L. Tran, D. Lee, J. H. Jung, Machine composition of korean music via topological data analysis and artificial neural network, *J. Math. Music*, **18** (2024), 20–41. https://doi.org/10.1080/17459737.2023.2197905

36. M. G Bergomi, A. Baratè, Homological persistence in time series: an application to music classification., *J. Math. Music*, **14** (2020),204–221. https://doi.org/10.1080/17459737.2020.1786745

37. M. Mijangos, A. Bravetti, P. Padilla-Longoria, Musical stylistic analysis: a study of intervallic transition graphs via persistent homology, *J. Math. Music*, **18** (2024), 89–108. https://doi.org/10.1080/17459737.2023.2232811

38. D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, et al., Spectral temporal graph neural network for multivariate time-series forecasting, *Adv. Neural Inform. Proc. Syst.*, **33** (2020), 17766–17778.

39. A. Deng, B. Hooi, Graph neural network-based anomaly detection in multivariate time series, In: *Proceedings of the AAAI conference on artificial intelligence*, 2021, 4027–4035. https://doi.org/10.1609/aaai.v35i5.16523

40. D. Zha, K. Lai, K. Zhou, X. Hu, Towards similarity-aware time-series classification, In: *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, 2022, 199–207. https://doi.org/10.1137/1.9781611977172.23

41. R. Gilmore, Topological analysis of chaotic dynamical systems, *Rev. Mod. Phys.*, **70** (1998), 1455. https://doi.org/10.1103/RevModPhys.70.1455

42. D. S. Broomhead, G. P. King, Extracting qualitative dynamics from experimental data, *Physica D*, **20** (1986), 217–236. https://doi.org/10.1016/0167-2789(86)90031-X

43. E. Tan, S. Algar, D. Corrêa, M. Small, T. Stemler, D. Walker, Selecting embedding delays: An overview of embedding techniques and a new method using persistent homology, *Chaos*, **33** (2023), 032101. https://doi.org/10.1063/5.0137223

44. L. Vietoris, Über den höheren zusammenhang kompakter räume und eine klasse von zusammenhangstreuen abbildungen, *Math. Ann.*, **97** (1927), 454–472. https://doi.org/10.1007/BF01447877

45. M. Gromov, Hyperbolic groups, In: *Essays in group theory*, 1987. https://doi.org/10.1007/978-1-4613-9586-7_3

46. Edelsbrunner, Letscher, Zomorodian, Topological persistence, simplification, *Discrete Comput. Geom.*, **28** (2002), 511–533. https://doi.org/10.1007/s00454-002-2885-2

47. H. Adams, A. Tausz, M. Vejdemo-Johansson, Javaplex: A research software package for persistent (co) homology, In: *Mathematical Software-ICMS 2014*, 2014, 129–136. https://doi.org/10.1007/978-3-662-44199-2_23

48. C. Maria, J. Boissonnat, M. Glisse, M. Yvinec, The gudhi library: Simplicial complexes and persistent homology, In: *Mathematical Software-ICMS 2014*, 2014, 167–174. https://doi.org/10.1007/978-3-662-44199-2_28

49. U. Bauer, Ripser: efficient computation of vietoris-rips persistence barcodes, *J. Appl. Comput. Topology*, **5** (2021), 391–423. https://doi.org/10.1007/s41468-021-00071-5

50. D. Attali, A. Lieutier, D. Salinas, Vietoris-rips complexes also provide topologically correct reconstructions of sampled shapes, In: *Proceedings of the twenty-seventh annual symposium on Computational geometry*, 2011, 491–500. https://doi.org/10.1145/1998196.1998276

51. T. K. Dey, Y. Wang, *Computational topology for data analysis*, Cambridge University Press, 2022. https://doi.org/10.1017/9781009099950

52. E. W. Dijkstra, A note on two problems in connexion with graphs, *Numer. Math.*, **1** (1959), 269–271.

53. D. Cohen-Steiner, H. Edelsbrunner, J. Harer, Stability of persistence diagrams, In: *Proceedings of the twenty-first annual symposium on Computational geometry*, 2005, 263–271. https://doi.org/10.1145/1064092.1064133

54. M. M. Ghazani, A. A. M. Malekshah, R. Khosravi, Analyzing time–frequency connectedness between cryptocurrencies, stock indices, and benchmark crude oils during the covid-19 pandemic, *Financ. Innova.*, **10** (2024), 119. https://doi.org/10.1186/s40854-024-00645-z

55. S. C. Nayak, S. Dehuri, S. B. Cho, Elitist-opposition-based artificial electric field algorithm for higher-order neural network optimization and financial time series forecasting, *Financ. Innov.*, **10** (2024), 5. https://doi.org/10.1186/s40854-023-00534-x

56. M. Anas, S. J. H. Shahzad, L. Yarovaya, The use of high-frequency data in cryptocurrency research: A meta-review of literature with bibliometric analysis, *Financ. Innov.*, **10** (2024), 90. https://doi.org/10.1186/s40854-023-00595-y

57. S. W. Akingbade, M. Gidea, M. Manzi, V. Nateghi, Why topological data analysis detects financial bubbles? *Commun. Nonlinear Sci.*, **128** (2024), 107665. https://doi.org/10.1016/j.cnsns.2023.107665

58. P. Bubenik, Statistical topological data analysis using persistence landscapes, *J. Mach. Learn. Res.*, **16** (2015), 77–102.

AIMS Press