Explainable Framework for Time-series Analysis via Topological Data Analysis

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

We propose an explainable framework for TDA-based time-series analysis, which characterizes time-series signals through time-delay embedding and persistent diagrams. Given the persistence diagram corresponding to a target class, our method continuously deforms an input signal into a signal whose diagram is close to the target diagram. We formulate this problem as a minimization of Wasserstein distance between persistence diagrams. The potential of this method is illustrated on some synthetic and real examples.

1 Introduction

Machine learning has been widely applied in many domains such as healthcare, finance, and social science, where there is increasing interest in explainability so that the users can better understand and trust the results of the algorithms. Several explainable techniques have been proposed, such as visualization of feature contribution to a model's prediction [1, 6].

In this paper, we focus on explainable frameworks for time-series analysis. Time-series data commonly appear and play an important role in many real world applications. According to the previous study [11], practical explainable frameworks for time-series analysis is required to give information not only about which parts in a given time-series contribute to the output, but also about how it differs from signals in another class. To provide such information, Karlsson *et al.* [11] proposed the following problem: given a classifier and a signal, find a transformation of signals that changes the predicted class to a desired one with minimal distortion.

For time-series analysis, Topological Data Analysis (TDA) has been successfully used in the past decade [15, 16, 18, 19]. Here, TDA is a topic in data analysis, which characterizes the shape of data such as high dimensional point clouds. When TDA is applied to time-series analysis, a time-series signal is converted into a point cloud through time-delay embedding and then into a persistence diagram. With persistence diagrams, we can effectively apply machine learning methods to chaotic time-series signals [16, 18].

In this work, we focus on TDA for time-series analysis and propose a novel explainable framework. To show how a given time-series signal differs from one in another class, we consider deforming the input signal continuously so that the resultant persistent diagram is close to a target diagram. For this purpose, we propose to minimize the 2-Wasserstein distance between persistence diagrams, which can be solved by gradient-based method.

Submitted to the Topological Data Analysis and Beyond Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

Notation

In this work, a time-series signal is represented as a sequence of length N: $\{x_t\}_{t=0}^{N-1}, x_t \in \mathbb{R}$. For this time-series, we define a point cloud $e(\{x_t\}_t)$ embedded in an E-dimensional space as

$$\begin{aligned} e(\{x_t\}_t) &= \{u_0, u_1, \ldots, u_{n-1}\}, \\ u_i &= \left[x_i, \ldots, x_{i+(E-1)\tau}\right] \in \mathbb{R}^E, \end{aligned} \tag{1}$$

where E is a positive integer and τ is the time-delay parameter. This point set $e(\{x_t\}_t)$ is called the *time-delay embedding* of the signal $\{x_t\}_t$. With the right choice of E and τ it is possible to reconstruct the phase space of a dynamical system according to Takens' embedding theorem [17]. The correspondence from $\{x_t\}_t$ to $e(\{x_t\}_t)$ can be regarded as a linear map $e\colon \mathbb{R}^N \to \mathbb{R}^L$, where $L = n \cdot E$. Constructing the associated VR-filtration or DTM-filtration [2], we can compute the persistence diagram $D(\{x_t\}_t)$ from the time-delay embedding $e(\{x_t\}_t)$.

2 Methodology

Now we present our method in a more formal way. Given an input signal $\{x_t\}_t$ and a target persistence diagram D^{tgt}, our aim is to continuously deform $\{x_t\}_t$ so that the persistence diagram of the resulting signal is close to D^{tgt}. We formulate this goal in the form of the following optimization problem:

$$\underset{\{w_t\}_{t=0}^{N-1}, w_t \in \mathbb{R}}{\text{minimize}} \quad W_2^2(D(\{w_t\}_t), D^{\text{tgt}}), \tag{2}$$

where W_2^2 is the squared 2-Wasserstein distance and we initialize the problem with $\{x_t\}_t$. The definition of W_2^2 is shown in Appendix.

Since, the correspondence from a time-series to its persistence diagram $\{w_t\}_t \mapsto D(\{w_t\}_t)$ is piecewise differentiable, the objective function $W_2^2(D(\{w\}_t), D^{\text{tgt}})$ is piecewise differentiable. However, it rarely happens that a point generated by an iterative method lands exactly on a non-smooth configuration due to limited numerical precision [8]. Therefore, we can apply gradient-based methods to the problem (2). The piecewise differentiability of the objective function is described in Appendix. See also [5, 8, 12] for the differentiability of persistence maps.

Our approach can be used for any input signal and any target persistence diagram. In the context of explainability for time-series analysis, a target persistence diagram D^{tgt} needs to be defined in a suitable way. The target is expected to represent the desired behavior. One possible way is to define the target to be the persistence diagram of any signal that belongs to the desired class. Another way is to define the target by modifying the persistence diagram of an input signal according to our purpose. For example, we modify a persistence diagram according to a rule related to a TDA-based algorithm whose results we would like to analyze.

3 Experimental results

In this section, we show the usefulness of our algorithm based on both artificial and real-world time-series datasets. In each experiment, the algorithm continuously deformed a given time-series by solving the problem (2) with some target diagram. In the experiment with an artificial dataset, we set a target diagram by a rule associated with a classifier. For EEG and motion sensor datasets, we set our target diagram to be the diagram of a reference time-series in each case. We compared the input and the deformed signals to understand which behavior contributes to the difference.

Throughout our experiments, we set the embedding dimension $E = 3$ and the delay parameter $\tau = 1$. The algorithm was implemented in Python with the use of the GUDHI package [13], which is available at [9]. In order to minimize the optimal transport based cost in Equation (2), we used the fast implementation given in [10]. Also, we used the sequential quadratic programming (SQP) solver [14], which is available in SciPy.

3.1 Artificial datasets

First we show some results for an artificial dataset. We considered two types of anomalous time-series as show in Figure 1: (a) mean shifts, (b) spike noise, where the normal signal was generated from a Gaussian with standard derivation 0.5.

(a) mean shift

(b) spike noise

Figure 1: Examples of anomalous signals

To define a target diagram for each signal, we set a classification rule that detects anomaly based on the persistence diagram of a given signal. More concretely, we regarded a signal as anomalous if there is a point in its persistence diagram outside the region $\{(b,d) \in \mathbb{R}^2 \mid b \leq 1.4, d-b \leq 0.3\}$ (see Figure 1). According to this classification rule, we set a target diagram as follows. For a point (b,d) in the persistence diagram of an input signal with $b \leq 1, d-b > 0.3$ we replaced it by $(b, b+0.3)$, and for a point (b,d) with $b > 1.4$ we replaced it by $(1.4, 1.4)$, to obtain D^{tgt}. Here, we used the DTM-filtration with DTM parameter $k = 20$ to obtain a persistence diagram.

By solving the optimization problem given in Equation (2), we generated a new signal, which can be compared with the anomalous signal. Figure 2 (resp. Figure 3) shows a result for a signal with mean shifts (with spike noise) and its persistence diagram. In Figures 2 and 3, we could see that the generated signals exhibited normal behavior, while still being close to the input signals. By comparing the new signal with the anomalous signal, we could better understand how the TDA-based algorithm recognized anomalies.

In such applications, constant offsets could appear in the deformed signal, as we see in Figure 2. This is because two signals which differ by a constant offset could still have time-delay embedded point clouds of the same shape.

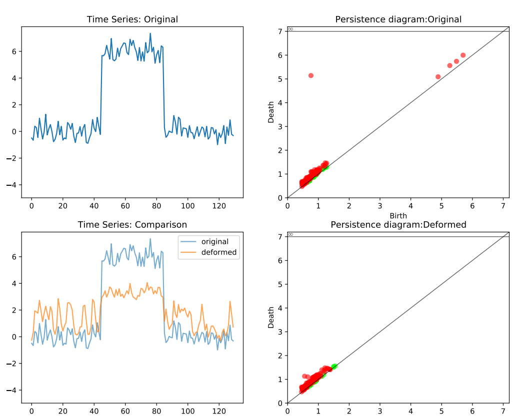

Figure 2: Result for a signal with mean shifts

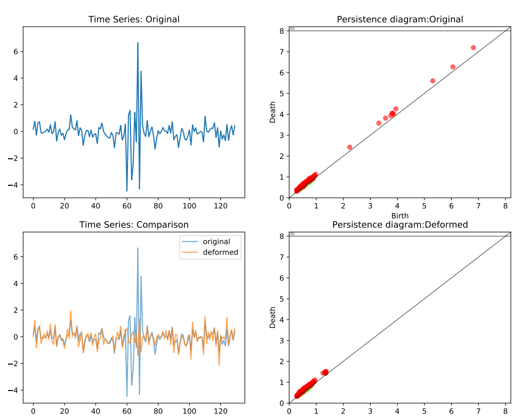

Figure 3: Result for a signal with spike noise

3.2 Real datasets

In this experiment, we use two real datasets: an electroencephalogram (EEG) dataset [3] and the daily and sports activities dataset [4].

EEG dataset contains two types of EEG signals, one is for the state of eyes open and the other is for the state of eyes closed. We set k in the DTM-filtration to be 5 and defined D^{tgt} to be the persistence diagram of a signal when eyes are closed and deformed a signal when eyes are open. Figure 4 shows the original signal, the target signal, the deformed signal, and their persistence diagrams. In the results shown in Figure 4, we noticed that the output signal behaved like one where eyes are closed.

By this result, we could see that the amplitude and frequency made the difference between the states of eyes open and closed, which agreed with a widely-accepted theory. Note that our framework deformed the signal by only referring to the target persistence diagram that corresponds to a signal obtained when eyes are closed.

Next we present a result for the daily and sports activities dataset [4]. We created an input time-series, where the first and third parts correspond to walking, while the middle part corresponds to running. The reference time-series consisted of walking only. Note that the walking samples were taken from different time windows. In the experiment, we set k in the DTM-filtration to be 10, and applied our algorithm to the input signal and the persistence diagram of the reference signal. From the results shown in Figure 5, we can see that the amplitude of the part corresponding to running was deformed towards that of walking. Thus, our framework succeeded in showing how the part corresponding to running differs from that of walking.

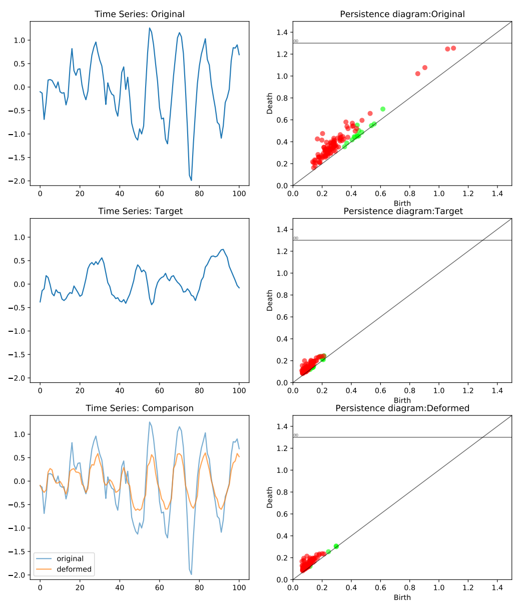

Figure 4: Result for a signal in the EEG dataset [3]

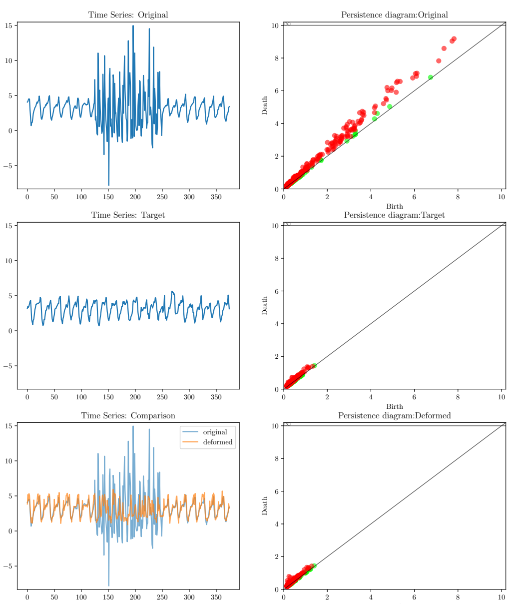

Figure 5: Result for a signal in the dataset [4]

Through these experiments, we verified that our proposed algorithm could bridge the difference between persistence diagrams into that between time-series, which can explain how two classes differ as time-series data. Each of the broad range of situations described so far could be addressed by dedicated methods derived from a priori knowledge. However, an advantage of our approach is that it can work without requiring special knowledge about a given problem.

4 Conclusion

In this paper, we have proposed a TDA-based explainable framework to better understand the output of a machine learning algorithm for time-series data. Our proposed method can continuously deform an input signal so that the persistence diagram of the resulting signal is close to a target one that describes behavior of a specific class. This gives further information about which behavior of the signal led to it being classified in a particular way. We have formulated the problem of finding such a deformation as a minimization problem of the distance between persistence diagrams, which can be solved by a gradient-based approach. With artificial and real-world datasets, we have experimentally shown that our method could explain the difference between two classes of time-series data through the difference in their persistence diagrams.

To the best of our knowledge, our work is the first application of a TDA-based method towards explainability for time-series data. We believe that our method provides a new fundamental tool for time-series analysis. For further research, defining a desired target persistence diagram is one of the interesting topics.

References

[1] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.

[2] Hirokazu Anai, Frédéric Chazal, Marc Glisse, Yuichi Ike, Hiroya Inakoshi, Raphaël Tinarrage, and Yuhei Umeda. DTM-Based Filtrations. In Gill Barequet and Yusu Wang, editors, *35th International Symposium on Computational Geometry, SoCG 2019, June 18-21, 2019, Portland, Oregon, USA.*, volume 129 of *LIPIcs*, pages 58:1–58:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2019.

[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] Billur Barshan and Murat Cihan Yüksek. Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units. *The Computer Journal*, 57(11):1649–1667, 2014.

[5] Frédéric Chazal and Vincent Divol. The density of expected persistence diagrams and its kernel based estimation. *arXiv preprint arXiv:1802.10457*, 2018.

[6] Filip Karlo Dosilovic, Mario Brcic, and Nikica Hlupic. Explainable artificial intelligence: A survey. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, May 2018.

[7] Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

[8] Marcio Gameiro, Yasuaki Hiraoka, and Ippei Obayashi. Continuation of point clouds via persistence diagrams. *Physica D: Nonlinear Phenomena*, 334:118–132, 2016.

[9] GUDHI. GUDHI Library - Geometry Understanding in Higher Dimensions. https://github.com/GUDHI/gudhi-devel, 2020. Accessed: 2020-01-21.

[10] Christoph Heindl. Linear Assignment Problem Solver. https://github.com/cheind/py-lapsolver, 2019. Accessed: 2019-07-26.

[11] Isak Karlsson, Jonathan Rebane, Panagiotis Papapetrou, and Aristides Gionis. Explainable time series tweaking via irreversible and reversible temporal transformations. *2018 IEEE International Conference on Data Mining (ICDM)*, pages 207–216, 2018.

[12] Jacob Leygonie, Steve Oudot, and Ulrike Tillmann. A framework for differential calculus on persistence barcodes. *arXiv preprint arXiv:1910.00960*, 2019.

[13] Clément Maria, Jean-Daniel Boissonnat, Marc Glisse, and Mariette Yvinec. The gudhi library: Simplicial complexes and persistent homology. In *International Congress on Mathematical Software*, pages 167–174. Springer, 2014.

[14] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[15] 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.

[16] Lee M Seversky, Shelby Davis, and Matthew Berger. On time-series topological data analysis: New data and opportunities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 59–67, 2016.

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

[18] Yuhei Umeda. Time series classification via topological data analysis. *Information and Media Technologies*, 12:228–239, 2017.

[19] Vinay Venkataraman, Karthikeyan Natesan Ramamurthy, and Pavan Turaga. Persistent homology of attractors for action recognition. In *2016 IEEE international conference on image processing (ICIP)*, pages 4150–4154. IEEE, 2016.

A Persistence diagram and Wasserstein distance

In this section, we explain how to construct a persistence diagram from a given point cloud. We also recall the Wasserstein distance on the space of persistence diagrams.

Let P be a point cloud in $\mathbb{R}^E$: $P = \{u_i \in \mathbb{R}^E \mid i = 1, \dots, n\}$. We regard a subset of P as a simplex, for example, a point, a line segment connecting two points, a triangle, and so on. For $r \in \mathbb{R}_{\geq 0}$, we define a set of simplices or a simplicial complex $\Sigma(r)$ by

$$\begin{aligned} &\sigma \in \Sigma(r) \iff r \geq r_{u_i,u_j} \text{ for any } u_i, u_j \in \sigma, \\ &\text{where } r_{u_i,u_j} = \frac{1}{2}\|u_i - u_j\|. \end{aligned} \tag{3}$$

In other words, the value r_σ when a simplex σ appears in $\Sigma(r)$ is equal to $\max_{u_i,u_j \in \sigma} r_{u_i,u_j}$. By considering increasing values of r, we obtain a series of simplicial complexes $\{\Sigma(r)\}_r$, which is called a filtration. The above definition leads to the Vietoris-Rips filtration.

Defining r_σ in other ways could give us other filtrations. Here we present DTM-filtration [2], which is base on the k-nearest neighbors structure around each point and proven to be more robust to noise. We fix a positive integer k and define the DTM-function $f\colon P \to \mathbb{R}_{\geq 0}$ on the point cloud P as follows,

$$f(u_i) = \sqrt{\frac{1}{k} \sum_{u \in \mathrm{NN}(u_i,k)} \|u - u_i\|^2}, \tag{4}$$

where $\mathrm{NN}(u_i, k)$ denotes the set of k-nearest neighbors of u_i. By replacing r_{u_i,u_j} in (3) with

$$r_{u_i,u_j} = \frac{1}{2}\left(\|u_i - u_j\| + f(u_i) + f(u_j)\right), \tag{5}$$

we obtain the DTM-filtration. Note that in the case $k = 1$ the DTM-filtration is equivalent to the Vietoris-Rips filtration.

As r increases, more and more simplices appear in $\Sigma(r)$ and the topology of the resulting simplicial complex would change. Changing topology takes the form of birth or death of various topological features. Here, 0-dimensional topological features are connected components, 1-dimensional ones are loops, 2-dimensional ones are voids, etc. Each ℓ-dimensional topological feature appears (resp. vanishes) by the introduction of a particular simplex and the value r which leads to the creation of the simplex is called the birth (resp. death) time of the feature. The pair of the birth and death times of an ℓ-dimensional feature can be described as a point (b, d) in $\bar{\mathbb{R}}^2$, where $\bar{\mathbb{R}} = \mathbb{R} \cup \{\infty\}$. Here if a feature never vanishes, its death time is considered to be ∞. For each dimension ℓ, the collection of such points forms a multiset $D_\ell(P)$ of $\bar{\mathbb{R}}^2$ and is called the ℓth *persistence diagram*. In our application, we simultaneously consider persistence diagrams corresponding to more than one dimension. Hence, it is useful to introduce the notion of a total persistence diagram: $D(P) := \{D_\ell(P)\}_{\ell=0,1,2,\dots}$.

A distance based on optimal transport, such as the bottleneck distance or the p-Wasserstein distance, endows the space of persistence diagrams with a partial matching metric [7]. For $p \geq 1$, the p-Wasserstein metric between two ℓth persistence diagrams D_ℓ^1 and D_ℓ^2 is defined as

$$W_p(D_\ell^1, D_\ell^2) := \left(\inf_{\gamma \in \Gamma\left(\overline{D_\ell^1}, \overline{D_\ell^2}\right)} \sum_{q \in \overline{D_\ell^1}} \|q - \gamma(q)\|_p^p \right)^{\frac{1}{p}} \tag{6}$$

where $\overline{D} = D \cup \Delta$, Δ being the diagonal $\{(x, x) \in \bar{\mathbb{R}}^2 \mid x \in \bar{\mathbb{R}}\}$ with infinite multiplicity, and $\Gamma\left(\overline{D_\ell^1}, \overline{D_\ell^2}\right)$ is the set of all bijections $\gamma\colon \overline{D_\ell^1} \to \overline{D_\ell^2}$. This means that a point in one persistence diagram can be either matched to another point in the other diagram or its own projection to the diagonal. In this paper, we work with $p = 2$. Moreover, for two total persistence diagrams D^1 and D^2, we define the squared 2-Wasserstein distance between them by

$$W_2^2(D^1, D^2) := \sum_{\ell \geq 0} \left(W_2(D_\ell^1, D_\ell^2)\right)^2. \tag{7}$$

In practical use of the DTM-filtration [2], points in 0th persistence diagrams could be close to the diagonal. When we minimize the 2-Wasserstein distance between two 0th persistence diagrams, a point could get matched to its own projection on the diagonal, instead of to a point in the other diagram. Thus, we will not be able to move the input persistence diagram towards the target diagram, when we try to minimize the objective in Equation (2). To overcome this problem, under the condition that two 0th persistence diagrams have the same number of points, we will use complete matching instead of partial matching. That is, for 0th persistence diagrams D_0^1 and D_0^2 with the same number of points, we compute

$$W_2(D_0^1, D_0^2) \ := \ \left(\inf_{\gamma \in \Gamma\left(D_0^1, D_0^2\right)} \sum_{q \in D_0^1} \|q - \gamma(q)\|_2^2 \right)^{\frac{1}{2}}, \tag{8}$$

where unlike Equation (6), $\Gamma\left(D_0^1, D_0^2\right)$ is the set of bijections between D_0^1 and D_0^2. In our applications, the assumption on the number of points in 0th persistence diagrams is satisfied and we use (8) for the 0th part in the squared Wasserstein distance (7). In case the target 0th persistence diagram has a different number of points compared to the input 0th diagram, we can interpolate and resample the input signal, so that this condition is satisfied. This will encourage points in the input diagram to move towards those in the target.

B Piecewise differentiability of the objective function

In this section, we explain the piecewise differentiability of the objective function.

We identify $\{u_i \in \mathbb{R}^E \mid i = 1, \ldots, n\}$ with the vector $u \in \mathbb{R}^L, L = n \cdot E$ obtained by stacking the points u_i. For $u \in \mathbb{R}^L$, we obtain the persistence diagram $D(u)$ of the corresponding point cloud, through the Vietoris-Rips filtration or the DTM-filtration [2]. We can represent the persistence diagram $D(u)$ as a vector, by ordering the birth-death pairs as described in [8], for example. It can be aligned as the following form:

$$D(u) = (b_1, d_1, \ldots, b_{m'}, d_{m'}, b_{m'+1}, \ldots, b_{m'+n'}) \in \mathbb{R}^M, \tag{9}$$

where $M = 2m' + n'$. Note that the dimension M depends on the configuration of a point cloud u. One can prove that there exists a decomposition $\mathbb{R}^L = S \sqcup O_1 \sqcup \cdots \sqcup O_m$ satisfying

1. S is of measure zero and each O_i is an open subset,
2. on each O_i the dimension M of $D(u)$ is constant, and
3. the correspondence $O_i \ni u \mapsto D(u) \in \mathbb{R}^M$ is diffentiable.

Each element of $D(u)$ is of the filtration value r_σ associated with some simplex σ, which is explicitly written as a function of u. Thus on each O_i, we can compute the derivative of the map $u \mapsto D(u)$ with respect to u. This leads to a piecewise differentiable map that assigns a point cloud to a persistence diagram, which is called the persistence map.

By composing the persistence map with the time-delay embedding map e, we obtain a piecewise differentiable map from a time-series to its persistence diagram, whose derivative can be computed by the chain rule. As a result, the objective function $W_2^2(D(\{w_t\}_t), D^{\text{tgt}})$ is a piecewise differentiable function with respect to $\{w_t\}_t \in \mathbb{R}^N$ and its gradient can be explicitly computed.