Time-Aware Knowledge Representations of Dynamic Objects with Multidimensional Persistence

Baris Coskunuzer*[1], Ignacio Segovia-Dominguez*[2], Yuzhou Chen*[3], Yulia R. Gel[1,4]

[1]University of Texas at Dallas, Department of Mathematical Sciences
[2]West Virginia University, School of Mathematical & Data Sciences
[3]Temple University, Department of Computer and Information Sciences
[4]National Science Foundation
coskunuz@utdallas.edu, Ignacio.SegoviaDominguez@mail.wvu.edu, yuzhou.chen@temple.edu, ygl@utdallas.edu

Abstract

Learning time-evolving objects such as multivariate time series and dynamic networks requires the development of novel knowledge representation mechanisms and neural network architectures, which allow for capturing implicit time-dependent information contained in the data. Such information is typically not directly observed but plays a key role in the learning task performance. In turn, lack of time dimension in knowledge encoding mechanisms for time-dependent data leads to frequent model updates, poor learning performance, and, as a result, subpar decision-making. Here we propose a new approach to a time-aware knowledge representation mechanism that notably focuses on implicit time-dependent topological information along multiple geometric dimensions. In particular, we propose a new approach, named *Temporal MultiPersistence* (TMP), which produces multidimensional topological fingerprints of the data by using the existing single parameter topological summaries. The main idea behind TMP is to merge the two newest directions in topological representation learning, that is, multi-persistence which simultaneously describes data shape evolution along multiple key parameters, and zigzag persistence to enable us to extract the most salient data shape information over time. We derive theoretical guarantees of TMP vectorizations and show its utility, in application to forecasting on benchmark traffic flow, Ethereum blockchain, and electrocardiogram datasets, demonstrating the competitive performance, especially, in scenarios of limited data records. In addition, our TMP method improves the computational efficiency of the state-of-the-art multipersistence summaries up to 59.5 times.

1 Introduction

Over the last decade, the field of topological data analysis (TDA) has demonstrated its effectiveness in revealing concealed patterns within diverse types of data that conventional methods struggle to access. Notably, in cases where conventional approaches frequently falter, tools such as persistent homology (PH) within TDA have showcased remarkable capabilities in identifying both localized and overarching patterns. These tools have the potential to generate a distinctive topological signature, a trait that holds great promise for a range of ML applications. This inherent capacity of PH becomes particularly appealing for capturing implicit temporal traits of evolving data, which might hold the crucial insights underlying the performance of learning tasks.

In turn, the concept of multiparameter persistence (MP) introduces a groundbreaking dimension to machine learning by enhancing the capabilities of persistent homology. Its objective is to analyze data across multiple dimensions concurrently, in a more nuanced manner. However, due to the complex algebraic challenges intrinsic to its framework, MP has yet to be universally defined in all contexts (Botnan and Lesnick 2022; Carrière and Blumberg 2020).

In response, we present a novel approach designed to effectively harness MP homology for the dual purposes of time-aware learning and the representation of time-dependent data. Specifically, the temporal parameter within time-dependent data furnishes the crucial dimension necessary for the application of the slicing concept within the MP framework. Our method yields a distinctive topological MP signature for the provided time-dependent data, manifested as multidimensional vectors (matrices or tensors). These vectors are highly compatible with ML applications. Notably, our findings possess broad applicability and can be tailored to various forms of PH vectorization, rendering them suitable for diverse categories of time-dependent data.

Our key contributions can be summarized as follows:

- We bring a new perspective to use TDA for time-dependent data by using multipersistence approach.
- We introduce TMP vectorizations framework which provides a multidimensional topological fingerprint of the data. TMP framework expands many known single persistence vectorizations to multidimensions by utilizing time dimension effectively in PH machinery.
- The versatility of our TMP framework allows its application to diverse types of time-dependent data. Furthermore, we show that TMP enjoys many important stability guarantees as most single persistence summaries.
- Rooted in computational linear algebra, TMP vectorizations generate multidimensional arrays (i.e., matrices or tensors) which serve as compatible inputs for various ML models. Notably, our proposed TMP approach boasts a speed advantage, performing up to 59.5 times faster than

*These authors contributed equally.

the cutting-edge MP methods.

- Through successful integration of the latest TDA techniques with deep learning tools, our TMP-Nets model consistently and cohesively outperforms the majority of state-of-the-art deep learning models.

2 Related Work

Time Series Forecasting

Recurrent Neural Networks (RNNs) are the most successful deep learning techniques to model datasets with time-dependent variables (Lipton, Berkowitz, and Elkan 2015). Long-Short-Term Memory networks (LSTMs) addressed the prior RNN limitations in learning long-term dependencies by solving known issues with exploding and vanishing gradients (Yu et al. 2019), serving as basis for other improved RNN, such as Gate Recurrent Units (GRUs) (Dey and Salem 2017), Bidirectional LSTMs (BI_LSTM) (Wang, Yang, and Meinel 2018), and seq2seq LSTMs (Sutskever, Vinyals, and Le 2014). Despite the widespread adoption of RNNs in multiple applications (Xiang, Yan, and Demir 2020; Schmidhuber 2017; Shin and Kim 2020; Shewalkar, Nyavanandi, and Ludwig 2019; Segovia-Dominguez et al. 2021; Bin et al. 2018), RNNs are limited by the structure of the input data and can not naturally handle data-structures from manifolds and graphs, i.e. non-Euclidean spaces.

Graph Convolutional Networks

New methods on graph convolution-based methods overcome prior limitations of traditional GCN approaches, e.g. learning underlying local and global connectivity patterns (Veličković et al. 2018; Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2017). GCN handles graph-structure data via aggregation of node information from the neighborhoods using graph filters. Lately, there is an increasing interest in expanding GCN capabilities to the time series forecasting domain. In this context, modern approaches have reached outstanding results in COVID-19 forecasting, money laundering, transportation forecasting, and scene recognition (Pareja et al. 2020; Segovia Dominguez et al. 2021; Yu, Yin, and Zhu 2018; Yan, Xiong, and Lin 2018; Guo et al. 2019; Weber et al. 2019; Yao et al. 2018). However, a major drawback of these approaches is the lack of versatility as they assume a fixed graph-structure and rely on the existing correlation among spatial and temporal features.

Multiparameter Persistence

Multipersistence (MP) is a highly promising approach to significantly improve the success of single parameter persistence (SP) in applied TDA, but the theory is not complete yet (Botnan and Lesnick 2022). Except for some special cases, the MP theory tends to suffer from the problem of the nonexistence of barcode decomposition because of the partially ordered structure of the index set $\{(\alpha_i, \beta_j)\}$. The existing approaches remedy this issue via the slicing technique by studying one-dimensional fibers of the multiparameter domain. However, this approach tends to lose most of the information the MP approach produces. Another idea along these lines is to use several such directions (vineyards), and produce a vectorization summarizing these SP vectorizations (Carrière and Blumberg 2020). However, again choosing these directions suitably and computing restricted SP vectorizations are computationally costly which restricts these approaches in many real-life applications. There are several promising recent studies in this direction (Botnan, Oppermann, and Oudot 2022; Vipond 2020), but these techniques often do not provide a topological summary that can readily serve as input to ML models. In this paper, we develop a highly efficient way to use the MP approach for time-dependent data and provide a multidimensional topological summary with TMP Vectorizations. We discuss the current fundamental challenges in the MP theory and the contributions of our TMP vectorizations in Appendix D.

3 Background

We start by providing the basic background for our machinery. While our techniques are applicable to any type of time-dependent data, here we mainly focus on the dynamic networks since our primary motivation comes from time-aware learning of time-evolving graphs as well as time series and spatio-temporal processes, also represented as graph structures. (For discussion on other types of data see Appendix D.)

Notation Table: All the notations used in the paper are given in Table 12 in the appendix.

Time-Dependent Data: Throughout the paper, by *time-dependent data*, we mean the data which implicitly or explicitly has time information embedded in itself. Such data include but are not limited to multivariate time series, spatio-temporal processes, and dynamic networks. Since our paper is primarily motivated by time-aware graph neural networks and their broader applications to forecasting, we focus on dynamic networks. Let $\{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_T\}$ be a sequence of weighted graphs for time steps $t = \{1, \dots, T\}$. In particular, $\mathcal{G}_t = \{\mathcal{V}_t, \mathcal{E}_t, W_t\}$ with node set $\mathcal{V}_t$, and edge set $\mathcal{E}_t$. Let $|\mathcal{V}_t| = N_t$ be the cardinality of the node set. W_t represents the edge weights for $\mathcal{E}_t$ as a nonnegative symmetric $N_t \times N_t$-matrix with entries $\{\omega^t_{rs}\}_{1 \leq r,s \leq N_t}$, i.e. the adjacency matrix of $\mathcal{G}_t$. In other words, $\omega^t_{rs} > 0$ for any $e^t_{rs} \in \mathcal{E}_t$ and $\omega^t_{rs} = 0$, otherwise. In the case of unweighted networks, let $\omega^t_{rs} = 1$ for any $e^t_{rs} \in \mathcal{E}_t$ and $\omega^t_{rs} = 0$, otherwise.

Background on Persistent Homology

Persistent homology (PH) is a mathematical machinery to capture the hidden shape patterns in the data by using algebraic topology tools. PH extracts this information by keeping track of the evolution of the topological features (components, loops, cavities) created in the data while looking at it using different resolutions. Here, we give basic background for PH in the graph setting. For further details, see (Dey and Wang 2022; Edelsbrunner and Harer 2010).

For a given graph $\mathcal{G}$, consider a nested sequence of subgraphs $\mathcal{G}^1 \subseteq \ldots \subseteq \mathcal{G}^N = \mathcal{G}$. For each $\mathcal{G}^i$, define an abstract simplicial complex $\widehat{\mathcal{G}}^i$, $1 \leq i \leq N$, yielding a filtration of complexes $\widehat{\mathcal{G}}^1 \subseteq \ldots \subseteq \widehat{\mathcal{G}}^N$. Here, clique complexes are among the most common ones, i.e., clique complex $\widehat{\mathcal{G}}$ is obtained by assigning (filling with) a k-simplex to each complete $(k+1)$-complete subgraph in $\mathcal{G}$, e.g., a 3-clique, a complete 3-subgraph, in $\mathcal{G}$ will be filled with a 2-simplex (triangle). Then, in this sequence of simplicial complexes, one can systematically keep track of the evolution of the topological patterns mentioned above. A k-dimensional topological feature (or k-hole) may represent connected components (0-hole), loops (1-hole) and cavities (2-hole). For each k-hole σ, PH records its first appearance in the filtration sequence, say $\widehat{\mathcal{G}}^{b_\sigma}$, and first disappearance in later complexes, $\widehat{\mathcal{G}}^{d_\sigma}$ with a unique pair (b_σ, d_σ), where $1 \leq b_\sigma < d_\sigma \leq N$ We call b_σ *the birth time* of σ and d_σ *the death time* of σ. We call $d_\sigma - b_\sigma$ *the life span* of σ. PH records all these birth and death times of the topological features in *persistence diagrams*. Let $0 \leq k \leq D$ where D is the highest dimension in the simplicial complex $\widehat{\mathcal{G}}^N$. Then k^{th} persistence diagram $\mathrm{PD_k}(\mathcal{G}) = \{(b_\sigma, d_\sigma) \mid \sigma \in H_k(\widehat{\mathcal{G}}^i) \text{ for } b_\sigma \leq i < d_\sigma\}$. Here, $H_k(\widehat{\mathcal{G}}^i)$ represents the k^{th} homology group of $\widehat{\mathcal{G}}^i$ which keeps the information of the k-holes in the simplicial complex $\widehat{\mathcal{G}}^i$. With the intuition that the topological features with a long life span (persistent features) describe the hidden shape patterns in the data, these persistence diagrams provide a unique topological fingerprint of $\mathcal{G}$.

As one can easily notice, the most important step in the PH machinery is the construction of the nested sequence of subgraphs $\mathcal{G}^1 \subseteq \ldots \subseteq \mathcal{G}^N = \mathcal{G}$. For a given unweighted graph $\mathcal{G} = (\mathcal{V}, E)$, the most common technique is to use a filtering function $f : \mathcal{V} \rightarrow \mathbb{R}$ with a choice of thresholds $\mathcal{I} = \{\alpha_i\}_1^N$ where $\alpha_1 = \min_{v\in\mathcal{V}} f(v) < \alpha_2 < \ldots < \alpha_N = \max_{v\in\mathcal{V}} f(v)$. For $\alpha_i \in \mathcal{I}$, let $\mathcal{V}_i = \{v_r \in \mathcal{V} \mid f(v_r) \leq \alpha_i\}$. Let $\mathcal{G}^i$ be the induced subgraph of $\mathcal{G}$ by $\mathcal{V}_i$, i.e. $\mathcal{G}^i = (\mathcal{V}_i, \mathcal{E}_i)$ where $\mathcal{E}_i = \{e_{rs} \in \mathcal{E} \mid v_r, v_s \in \mathcal{V}_i\}$. This process yields a nested sequence of subgraphs $\mathcal{G}^1 \subset \mathcal{G}^2 \subset \ldots \subset \mathcal{G}^N = \mathcal{G}$, called *the sublevel filtration* induced by the filtering function f. Choice of f is crucial here, and in most cases, f is either an important function from the domain of the data, e.g. amount of transactions or volume transfer, or a function defined from intrinsic properties of the graph, e.g. degree, betweenness. Similarly, for a weighted graph, one can use sublevel filtration on the weights of the edges and obtain a suitable filtration reflecting the domain information stored in the edge weights. For further details on different filtration types of networks, see (Aktas, Akbas, and El Fatmaoui 2019; Hofer et al. 2020).

Multidimensional Persistence

In the previous section, we discussed the single-parameter persistence theory. The reason for the term ”single” is that we filter the data in only one direction $\mathcal{G}^1 \subset \cdots \subset \mathcal{G}^N = \mathcal{G}$. Here, the choice of direction is the key to extracting the hidden patterns from the observed data. For some tasks and data types, it is sufficient to consider only one dimension (or filtering function $f : \mathcal{V} \rightarrow \mathbb{R}$) (e.g., atomic numbers for protein networks) in order to extract the intrinsic data properties. However, often the observed data may have more than one direction to be analyzed (for example, in the case of money laundering detection on bitcoin, we may need to use both transaction amounts and numbers of transactions between any two traders). With this intuition, multiparameter persistence (MP) theory is suggested as a natural generalization of single persistence (SP).

In simpler terms, if one uses only one filtering function, sublevel sets induce a single parameter filtration $\widehat{\mathcal{G}}^1 \subset \cdots \subset \widehat{\mathcal{G}}^N = \widehat{\mathcal{G}}$. Instead, if one uses two or more functions, then it would enable us to study finer substructures and patterns in the data. In particular, let $f : \mathcal{V} \rightarrow \mathbb{R}$ and $g : \mathcal{V} \rightarrow \mathbb{R}$ be two filtering functions with very valuable complementary information of the network. Then, MP idea is presumed to produce a unique topological fingerprint combining the information from both functions. These pair of functions f, g induces a multivariate filtering function $F : \mathcal{V} \mapsto \mathbb{R}^2$ with $F(v) = (f(v), g(v))$. Again, we can define a set of nondecreasing thresholds $\{\alpha_i\}_1^m$ and $\{\beta_j\}_1^n$ for f and g respectively. Then, $\mathcal{V}^{ij} = \{v_r \in V \mid f(v_r) \leq \alpha_i, g(v_r) \leq \beta_j\}$, i.e. $\mathcal{V}^{ij} = F^{-1}((-\infty, \alpha_i] \times (-\infty, \beta_j])$. Then, let $\mathcal{G}^{ij}$ be the induced subgraph of $\mathcal{G}$ by $\mathcal{V}^{ij}$, i.e. the smallest subgraph of $\mathcal{G}$ containing $\mathcal{V}^{ij}$. Then, instead of a single filtration of complexes, we get a bifiltration of complexes $\{\widehat{\mathcal{G}}^{ij} \mid 1 \leq i \leq m, 1 \leq j \leq n\}$. See Figure 2 (Appendix) for an explicit example.

As illustrated in Figure 2, we can imagine $\{\widehat{\mathcal{G}}^{ij}\}$ as a rectangular grid of size $m \times n$ such that for each $1 \leq i_0 \leq m$, $\{\widehat{G}^{i_0 j}\}_{j=1}^n$ gives a nested (horizontal) sequence of simplicial complexes. Similarly, for each $1 \leq j_0 \leq n$, $\{\widehat{G}^{i j_0}\}_{i=1}^m$ gives a nested (vertical) sequence of simplicial complexes. By computing the homology groups of these complexes, $\{H_k(\mathcal{G}^{ij})\}$, we obtain the induced bigraded persistence module (a rectangular grid of size $m \times n$). Again, the idea is to keep track of the k-dimensional topological features via the homology groups $\{H_k(\widehat{\mathcal{G}}^{ij})\}$ in this grid. As detailed in Appendix D, because of the technical issues related to commutative algebra coming from the partially ordered structure of the multipersistence module, this MP approach has not been completed like SP theory yet, and there is a need to facilitate this promising idea effectively in real-life applications.

In this paper, for time-dependent data, we overcome this problem by using the naturally inherited special direction in the data: Time. By using this canonical direction in the multipersistence module, we bypass the partial ordering problem and generalize the ideas from single parameter persistence to produce a unique topological fingerprint of the data via MP. Our approach provides a general framework to utilize various vectorization forms defined for single PH and gives a multidimensional topological summary of the data.

Utilizing Time Direction - Zigzag Persistence: While our intuition is to use time direction in MP for forecasting purposes, the time parameter is not very suitable to use in PH construction in its original form. This is because PH construction needs nested subgraphs to keep track of the existing topological features, while time-dependent data do not come nested, i.e. $\mathcal{G}_{t_1} \not\subseteq \mathcal{G}_{t_2}$ in general for $t_1 \leq t_2$. However, a generalized version of PH construction helps us to overcome this problem. We want to keep track of topological features which exist in different time instances. Zigzag homology (Carlsson and Silva 2010) bypasses the requirement of the nested sequence by using the "zigzag scheme". We provide the details of zigzag persistent homology in Appendix C.

4 TMP Vectorizations

We now introduce a general framework to define vectorizations for multipersistence homology on time-dependent data. First, we recall the single persistence vectorizations which we will expand as multidimensional vectorizations with our TMP framework.

Single Persistence Vectorizations

While PH extracts hidden shape patterns from data as persistence diagrams (PD), PDs being a collection of points $\{(b_\sigma, d_\sigma)\}$ in $\mathbb{R}^2$ by itself are not very practical for statistical and machine learning purposes. Instead, the common techniques are by accurately representing PDs as kernels (Kriege, Johansson, and Morris 2020) or vectorizations (Ali et al. 2023). *Single Persistence Vectorizations* transform obtained PH information (PDs) into a function or a feature vector form which are much more suitable for ML tools. Common single persistence (SP) vectorization methods are Persistence Images (Adams et al. 2017), Persistence Landscapes (Bubenik 2015), Silhouettes (Chazal et al. 2014), and various Persistence Curves (Chung and Lawson 2022). These vectorizations define a single variable or multivariable functions out of PDs, which can be used as fixed size $1D$ or $2D$ vectors in applications, i.e $1 \times m$ vectors or $m \times n$ vectors. For example, a Betti curve for a PD with m thresholds can be written as $1 \times m$ size vectors. Similarly, persistence images is an example of $2D$ vectors with the chosen resolution (grid) size. See the examples below and in Appendix D for further details.

TMP Vectorizations

Finally, we define our Temporal MultiPersistence (TMP) framework for time-dependent data. In particular, by using the existing single-parameter persistence vectorizations, we produce multidimensional vectorization by effectively using the time direction in the multipersistence module. The idea is to use zigzag homology in the time direction and consider d-dimensional filtering for the other directions. This process produces $(d+1)$-dimensional vectorizations of the dataset. While the most common choice would be $d = 1$ for computational purposes, we restrict ourselves to $d = 2$ to give a general idea. The construction can easily be generalized to higher dimensions. Below and in Appendix D, we provide explicit examples of TMP Vectorizations. While we mainly focus on network data in this part, we give how to generalize TMP vectorizations to other types of data (e.g., point clouds, images) in Appendix D.

Again, let $\widetilde{\mathcal{G}} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_T\}$ be a sequence of weighted (or unweighted) graphs for time steps $t = 1, \ldots, T$ with $\mathcal{G}_t = \{\mathcal{V}_t, \mathcal{E}_t, W_t\}$ as defined in Section 3. By using a filtering function $F_t : \mathcal{V}_t \to \mathbb{R}^d$ or weights, define a bifiltration for each t_0, i.e. $\{\mathcal{G}_{t_0}^{ij}\}$ for $1 \leq i \leq m$ and $1 \leq j \leq n$. For each fixed i_0, j_0, consider the sequence $\{\mathcal{G}_1^{i_0 j_0}, \mathcal{G}_2^{i_0 j_0}, \ldots, \mathcal{G}_T^{i_0 j_0}\}$. This sequence of subgraphs induces a zigzag sequence of clique complexes as described in Appendix C:

$$\widehat{\mathcal{G}}_1^{i_0 j_0} \hookrightarrow \widehat{\mathcal{G}}_{1.5}^{i_0 j_0} \hookleftarrow \widehat{\mathcal{G}}_2^{i_0 j_0} \hookrightarrow \widehat{\mathcal{G}}_{2.5}^{i_0 j_0} \hookleftarrow \widehat{\mathcal{G}}_3 \hookrightarrow \cdots \hookleftarrow \widehat{\mathcal{G}}_T^{i_0 j_0}.$$

Now, let $ZPD_k(\widetilde{\mathcal{G}}^{i_0 j_0})$ be the induced zigzag persistence diagram. Let φ represent an SP vectorization as described above, e.g. Persistence Landscape, Silhouette, Persistence Image, Persistence Curves. This means if $PD(\mathcal{G})$ is the persistence diagram for some filtration induced by $\mathcal{G}$, then we call $\varphi(\mathcal{G})$ is the corresponding vectorization for $PD(\mathcal{G})$ (see Figure 1 in Appendix F7). In most cases, $\varphi(\mathcal{G})$ is represented as functions on the threshold domain (Persistence curves, Landscapes, Silhouettes, Persistence Surfaces). However, the discrete structure of the threshold domain enables us to interpret the function $\varphi(\mathcal{G})$ as a $1D$-vector $\vec{\varphi}(\mathcal{G})$ (Persistence curves, Landscapes, Silhouettes) or $2D$-vector $\vec{\varphi}(\mathcal{G})$ (Persistence Images). See examples given below and in the Appendix D for more details.

Now, let $\vec{\varphi}(\widetilde{\mathcal{G}}^{ij})$ be the corresponding vector for the zigzag persistence diagram $ZPD_k(\widetilde{\mathcal{G}}^{ij})$. Then, for any $1 \leq i \leq m$ and $1 \leq j \leq n$, we have a ($1D$ or $2D$) vector $\vec{\varphi}(\widetilde{\mathcal{G}}^{ij})$. Now, define the induced TMP Vectorization $\mathbf{M}_\varphi$ as the corresponding tensor $\mathbf{M}_\varphi^{ij} = \vec{\varphi}(\widetilde{\mathcal{G}}^{ij})$ for $1 \leq i \leq m$ and $1 \leq j \leq n$.

In particular, if $\vec{\varphi}$ is a $1D$-vector of size $1 \times k$, then $\mathbf{M}_\varphi$ would be a $3D$-vector (rank-3 tensor) with size $m \times n \times k$. if $\vec{\varphi}$ is a $2D$-vector of size $k \times r$, then $\mathbf{M}_\varphi$ would be a rank-4 tensor with size $m \times n \times k \times r$. In the examples below, we provide explicit constructions for $\mathbf{M}_\varphi$ for the most common SP vectorizations φ.

Examples of TMP Vectorizations

While we describe TMP Vectorizations for $d = 2$, in most applications, $d = 1$ would be preferable for computational purposes. Then if the preferred single persistence (SP) vectorization φ produces $1D$-vector (say size $1 \times r$), then induced TMP Vectorization would be a $2D$-vector M_φ (a matrix) of size $m \times r$ where m is the number of thresholds for the filtering function used, e.g. $f : \mathcal{V}_t \to \mathbb{R}$. These $m \times r$ matrices provide unique topological fingerprints for each time-dependent dataset $\{\mathcal{G}_t\}_{t=1}^T$. These multidimensional fingerprints are produced by using persistent homology with two-dimensional filtering where the first dimension is the natural direction time t, and the second dimension comes from the filtering function f.

Here, we discuss explicit constructions of two examples of TMP vectorizations. As we mentioned above, the framework is very general, and it can be applied to various vectorization methods. In Appendix D, we provide details of further examples of TMP Vectorizations for time-dependent data, i.e., TMP Silhouettes, and TMP Betti Summaries.

TMP Landscapes Persistence Landscapes λ are one of the most common SP vectorization methods introduced in (Bubenik 2015). For a given persistence diagram

$PD(\mathcal{G}) = \{(b_i, d_i)\}$, λ produces a function $\lambda(\mathcal{G})$ by using generating functions Λ_i for each $(b_i, d_i) \in PD(\mathcal{G})$, i.e. $\Lambda_i : [b_i, d_i] \to \mathbb{R}$ is a piecewise linear function obtained by two line segments starting from $(b_i, 0)$ and $(d_i, 0)$ connecting to the same point $(\frac{b_i+d_i}{2}, \frac{b_i-d_i}{2})$. Then, the *Persistence Landscape* function $\lambda(\mathcal{G}) : [\epsilon_1, \epsilon_q] \to \mathbb{R}$ is defined as $\lambda(\mathcal{G})(t) = \max_i \Lambda_i(t)$ for $t \in [\epsilon_1, \epsilon_q]$, where $\{\epsilon_k\}_1^q$ represents the thresholds for the filtration used.

Considering the piecewise linear structure of the function, $\lambda(\mathcal{G})$ is completely determined by its values on $2q - 1$ points, i.e. $\frac{b_i \pm d_i}{2} \in \{\epsilon_1, \epsilon_{1.5}, \epsilon_2, \epsilon_{2.5}, \ldots, \epsilon_q\}$ where $\epsilon_{k.5} = (\epsilon_k + \epsilon_{k+1})/2$. Hence, a vector of size $1 \times (2q-1)$ whose entries the values of this function would suffice to capture all the information needed, i.e. $\vec{\lambda} = [\lambda(\epsilon_1)\ \lambda(\epsilon_{1.5})\ \lambda(\epsilon_2)\ \lambda(\epsilon_{2.5})\ \lambda(\epsilon_3)\ \ldots\ \lambda(\epsilon_q)]$.

Now, for the time-dependent data $\widetilde{\mathcal{G}} = \{\mathcal{G}_t\}_{t=1}^T$, to construct our induced TMP Vectorization $\mathbf{M}_\lambda$, TMP Landscapes, we use λ for time direction, $t = 1, \ldots, T$. For zigzag persistence, we have $2T - 1$ thresholds steps. Hence, by taking $q = 2T - 1$, we would have $4T - 3$ length vector $\vec{\lambda}(\widetilde{\mathcal{G}})$.

For the other multipersistence direction, by using a filtering function $f : \mathcal{V}_t \to \mathbb{R}$ with the threshold set $\mathcal{I} = \{\alpha_j\}_1^m$, we obtain *TMP Landscape* $\mathbf{M}_\lambda$ as follows: $\mathbf{M}_\lambda^j = \vec{\lambda}(\widetilde{\mathcal{G}}^j)$ where $\mathbf{M}_\lambda^j$ represents j^{th}-row of the $2D$-vector $\mathbf{M}_\lambda$. Here, $\widetilde{\mathcal{G}}^j = \{\mathcal{G}_t^j\}_{t=1}^T$ is induced by the sublevel filtration for $f : \mathcal{V}_t \to \mathbb{R}$ as described in the paper, i.e. $\mathcal{G}_t^j$ is the induced subgraph by $\mathcal{V}_t^j = \{v_r \in \mathcal{V}_t \mid f(v_r) \leq \alpha_j\}$.

Hence, for a time-dependent data $\widetilde{\mathcal{G}} = \{\mathcal{G}_t\}_{t=1}^T$, TMP Landscape $\mathbf{M}_\lambda(\widetilde{\mathcal{G}})$ is a $2D$-vector of size $m \times (4T - 3)$ where T is the number of time steps.

TMP Persistence Images Next SP vectorization in our list is persistence images (Adams et al. 2017). Different than most SP vectorizations, persistence images produce $2D$-vectors. The idea is to capture the location of the points in the persistence diagrams with a multivariable function by using the $2D$ Gaussian functions centered at these points. For $PD(\mathcal{G}) = \{(b_i, d_i)\}$, let ϕ_i represent a $2D$-Gaussian centered at the point $(b_i, d_i) \in \mathbb{R}^2$. Then, one defines a multivariable function, *Persistence Surface*, $\widetilde{\mu} = \sum_i w_i \phi_i$ where w_i is the weight, mostly a function of the life span $d_i - b_i$. To represent this multivariable function as a $2D$-vector, one defines a $k \times l$ grid (resolution size) on the domain of $\widetilde{\mu}$, i.e. threshold domain of $PD(\mathcal{G})$. Then, one obtains the *Persistence Image*, a $2D$-vector $\vec{\mu} = [\mu_{rs}]$ of size $k \times l$, where $\mu_{rs} = \int_{\Delta_{rs}} \widetilde{\mu}(x, y)\, dxdy$ and Δ_{rs} is the corresponding pixel (rectangle) in the $k \times l$ grid.

Following a similar route, for our TMP vectorization, we use time as one direction, and the filtering function in the other direction, i.e. $f : \mathcal{V}_t \to \mathbb{R}$ with threshold set $\mathcal{I} = \{\alpha_j\}_1^m$. Then, for time-dependent data $\widetilde{\mathcal{G}} = \{\mathcal{G}_t\}_{t=1}^T$, in the time direction, we use zigzag PDs and their persistence images. Hence, for each $1 \leq j \leq m$, we define *TMP Persistence Image* as $\mathbf{M}_\mu^j(\widetilde{\mathcal{G}}) = \vec{\mu}(\widetilde{\mathcal{G}}^j)$ where $2D$-vector $\mathbf{M}_\mu^j$ is j^{th}-floor of the $3D$-vector $\mathbf{M}_\mu$. Then, TMP Persistence Image $\mathbf{M}_\mu^j(\widetilde{\mathcal{G}})$ is a $3D$-vector of size $m \times k \times l$.

More details for TMP Persistence Surfaces and TMP Silhouettes are provided in Appendix D.

Stability of TMP Vectorizations

We now prove that when the source single parameter vectorization φ is stable, then so is its induced TMP vectorization $\mathbf{M}_\varphi$. We discuss the details of the stability notion in persistence theory and examples of stable SP vectorizations in Appendix C.

Let $\widetilde{\mathcal{G}} = \{\mathcal{G}_t\}_{t=1}^T$ and $\widetilde{\mathcal{H}} = \{\mathcal{H}_t\}_{t=1}^T$ be two time sequences of networks. Let φ be a stable SP vectorization with the stability equation

$$\mathrm{d}(\varphi(\widetilde{\mathcal{G}}), \varphi(\widetilde{\mathcal{H}})) \leq C_\varphi \cdot \mathcal{W}_{p_\varphi}(PD(\widetilde{\mathcal{G}}), PD(\widetilde{\mathcal{H}}))$$

for some $1 \leq p_\varphi \leq \infty$. Here, $\mathcal{W}_p$ represents Wasserstein-p distance as defined in Appendix C.

Now, consider the bifiltrations $\{\widehat{\mathcal{G}}_t^{ij}\}$ and $\{\widehat{\mathcal{H}}_t^{ij}\}$ for each $1 \leq t \leq T$. We define the induced matching distance between the multiple persistence diagrams (See Remark 2) as $\mathbf{D}(\{ZPD(\widetilde{\mathcal{G}})\}, \{ZPD(\widetilde{\mathcal{H}})\}) = \max_{i,j} \mathcal{W}_{p_\varphi}(ZPD(\widetilde{\mathcal{G}}^{ij}), ZPD(\widetilde{\mathcal{H}}^{ij}))$

Now, define the distance between TMP Vectorizations as $\mathbf{D}(\mathbf{M}_\varphi(\widetilde{\mathcal{G}}), \mathbf{M}_\varphi(\widetilde{\mathcal{H}})) = \max_{i,j} \mathrm{d}(\varphi(\widetilde{\mathcal{G}}^{ij}), \varphi(\widetilde{\mathcal{H}}^{ij}))$.

Theorem 1. *Let φ be a stable vectorization for single parameter PDs. Then, the induced TMP Vectorization $\mathbf{M}_\varphi$ is also stable, i.e. With the notation above, there exists $\widehat{C}_\varphi > 0$ such that for any pair of time-aware network sequences $\widetilde{\mathcal{G}}$ and $\widetilde{\mathcal{H}}$, we have the following inequality.*

$$\mathbf{D}(\mathbf{M}_\varphi(\widetilde{\mathcal{G}}), \mathbf{M}_\varphi(\widetilde{H})) \leq \widehat{C}_\varphi \cdot \mathbf{D}(\{ZPD(\widetilde{\mathcal{G}})\}, \{ZPD(\widetilde{\mathcal{H}})\})$$

The proof of the theorem is given in Appendix E.

5 TMP-Nets

To fully take advantage of the extracted signatures by TMP vectorizations, we propose a GNN-based module to track and learn significant temporal and topological patterns. Our Time Aware Multiparameter Persistence Nets (TMP-Nets) capture spatio-temporal relationships via trainable node embedding dictionaries in a GDL-based framework.

Graph Convolution on Adaptive Adjacency Matrix

To model the hidden dependencies among nodes in the spatio-temporal graph, we define the spatial graph convolution operation based on the adaptive adjacency matrix and given node feature matrix. Inspired by (Wu et al. 2019), to investigate the beyond pairwise relations among nodes, we use the adaptive adjacency matrix based on trainable node embedding dictionaries, i.e., $Z_{t,\text{Spatial}}^{(\ell)} = L Z_{t,\text{Spatial}}^{(\ell-1)} W^{(\ell-1)}$, where $L = \text{Softmax}(\text{ReLU}(E_\theta E_\theta^\top))$ (here $E_\theta \in \mathbb{R}^{N \times d_c}$ and $d_c \geq 1$), $Z_{\text{Spatial}}^{(\ell-1)}$ and $Z_{\text{Spatial}}^{(\ell)}$ are the input and output of $(\ell-1)$-th layer, and $Z_{\text{Spatial}}^{(0)} = X \in \mathbb{R}^{N \times F}$ (here F represents the number of features for each node), and $W^{(\ell-1)}$ is the trainable weights.

Topological Signatures Representation Learning

In our experiments, we use CNN based model to learn the TMP topological features. Given the TMP topological features of resolution p, i.e., $\text{TMP}_t \in \mathbb{R}^{p \times p}$, we employ CNN-based model and global max pooling to obtain the image-level local topological feature $Z_{t,\text{TMP}}$ as

$$Z_{t,\text{TMP}} = f_{\text{GMP}}(f_\theta(\text{TMP}_t)),$$

where f_{GMP} is the global max pooling, f_θ is a CNN based neural network with parameter set θ_i, and $Z_{t,\text{TMP}} \in \mathbb{R}^{d_c}$ is the output for TMP representation.

Lastly, we combine the two embeddings to obtain the final embedding Z_t:

$$Z_t = Concat(Z_{t,\text{Spatial}}, Z_{t,\text{TMP}}).$$

To capture both spatial and temporal correlations in time-series, we feed the final embedding Z_t into Gated Recurrent Units (GRU) for future time points forecasting.

6 Experiments

Datasets: We consider three types of data: two widely used benchmark datasets on California (CA) **traffic** (Chen et al. 2001) and **electrocardiography** (ECG5000) (Chen et al. 2015a), and the newly emerged data on Ethereum **blockchain** tokens (Shamsi et al. 2022). (The results on the ECG5000 are presented in Appendix A). More details descriptions of datasets can be found in Appendix B.

Experimental Results

We compare our TMP-Nets with 6 state-of-the-art baselines. We use three standard performance metrics Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean absolute percentage error (MAPE). We provide additional experimental results in Appendix A. In Appendix B, we provide further details on the experimental setup and empirical evaluation. Our source code is available at the link [1].

Results on Blockchain Datasets: Table 1 shows performance on Bytom, Decentraland, and Golem. Table 1 suggests the following phenomena: (i) TMP-Nets achieves the best performance on Bytom and Decentraland, and the relative gains of TMP-Nets over the best baseline (i.e., Z-GCNETs) are 7.89% and 3.66% on Bytom and Decentraland respectively; (ii) compared with Z-GCNETs, the size of TMP topological features used in this work is much smaller than the zigzag persistence image utilized in Z-GCNETs.

An interesting question is why TMP-Nets performs differently on Golem vs. Bytom and Decentraland. Success on each network token depends on the diversity of connections among nodes. In cryptocurrency networks, we expect nodes/addresses to be connected with other nodes with similar transaction connectivity (e.g. interaction among whales) as well as with nodes with low connectivity (e.g. terminal nodes). However, the assortativity measure of Golem (-0.47) is considerably lower than Bytom (-0.42) and Decentraland (-0.35), leading to disassortativity patterns (i.e., repetitive isolated clusters) in the Golem network, which, in turn, downgrade the success rate of forecasting.

Results on Traffic Datasets: For traffic flow data PeMSD4 and PeMSD8, we evaluate Z-GCNETs' performance on varying lengths. This allows us to further explore the learning capabilities of our Z-GCNETs as a function of sample size. In particular, in many real-world scenarios, there exists only a limited number of temporal records to be used in the training stage, and the learning problem with lower sample sizes becomes substantially more challenging. Tables 2 and 3 show that under the scenario of limited data records for both PeMSD4 and PeMSD8 (i.e., $\mathcal{T} = 1,000$ and $\mathcal{T}' = 2,000$), our TMP-Nets always outperforms three representative baselines in MAE and RMSE. For example, TMP-Nets significantly outperform SOTA baselines, where we achieve relative gains of 1.79% and 4.36% in RMSE on $\text{PeMSD4}_{\mathcal{T}=1,000}$ and $\text{PeMSD8}_{\mathcal{T}=1,000}$, respectively. Overall, the results demonstrate that our proposed TMP-Nets can accurately capture the hidden complex spatial and temporal correlations in the correlated time series datasets and achieve promising forecasting performances under the scenarios of limited data records. Moreover, we conduct experiments on the whole PeMSD4 and PeMSD8 datasets. As Table 6 (Appendix) indicates, our TMP-Nets still achieve competitive performances on both datasets.

Finally, we applied our approach in a different domain with a benchmark electrocardiogram dataset, ECG5000. Again, our model gives highly competitive results with the SOTA methods (Appendix A).

Ablation Studies: To better evaluate the importance of different components of TMP-Nets, we perform ablation studies on two traffic datasets, i.e., PeMSD4 and PeMSD8 by using only (i) $Z^{(\ell)}_{t,\text{Spatial}}$ or (ii) $Z_{t,\text{TMP}}$ as input. Table 4 report the forecasting performance of (i) $Z^{(\ell)}_{t,\text{Spatial}}$, (ii) $Z_{t,\text{TMP}}$, and (iii) TMP-Nets (our proposed model). We find that our TMP-Nets outperforms both $Z^{(\ell)}_{t,\text{Spatial}}$ and $Z_{t,\text{TMP}}$ on two datasets, yielding highly statistically significant gains. Hence, we can conclude that (i) TMP vectorizations help to better capture global and local hidden topological information in the time dimension, and (ii) spatial graph convolution operation accurately learns the inter-dependencies (i.e., spatial correlations) among spatio-temporal graphs. We provide further ablation studies comparing the effect of slicing direction and the MP vectorization methods in the Appendix A.

Computational Complexity: One of the key issues why MP has not propagated widely into practice yet is its high computational costs. Our method improves on the state-of-the-art MP (ranging from 23.8 to 59.5 times faster than Multiparameter Persistence Image (MP-I) (Carrière and Blumberg 2020), and from 1.2 to 8.6 times faster than Multiparameter Persistence Kernel (MP-L) (Corbet et al. 2019)) and, armed with a computationally fast vectorization method (e.g., Betti summary (Lesnick and Wright 2022)), TMP yields competitive computational costs for a lower number of filtering functions (See Appendix A). Nevertheless, scaling for really large scale-problems is still a challenge. In the

[1] https://www.dropbox.com/sh/h28f1cf98t9xmzj/AACBavvHc_ctCB1FVQNyf-XRa?dl=0

Model	Bytom	Decentraland	Golem
DCRNN (Li et al. 2018)	35.36±1.18	27.69±1.77	23.15±1.91
STGCN (Yu, Yin, and Zhu 2018)	37.33±1.06	28.22±1.69	23.68±2.31
GraphWaveNet (Wu et al. 2019)	39.18±0.96	37.67±1.76	28.89±2.34
AGCRN (Bai et al. 2020)	34.46±1.37	26.75±1.51	22.83±1.91
Z-GCNETs (Chen, Segovia, and Gel 2021)	31.04±0.78	23.81±2.43	**22.32±1.42**
StemGNN (Cao et al. 2020)	34.91±1.04	28.37±1.96	22.50±2.01
TMP-Nets	**28.77±3.30**	**22.97±1.80**	29.01±1.05

Table 1: Experimental results on Bytom, Decentraland, and Golem on MAPE and standard deviation.

Model	PeMSD4			PeMSD8		
	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
AGCRN	110.36±0.20	150.37±0.15	208.36±0.20	87.12±0.25	109.20±0.33	277.44±0.26
Z-GCNETs	112.65±0.12	153.47±0.17	**206.09±0.33**	69.82±0.16	95.83±0.37	**102.74±0.53**
StemGNN	112.83±0.07	150.22±0.30	209.52±0.51	65.16±0.36	89.60±0.60	108.71±0.51
TMP-Nets	**108.38±0.10**	**147.57±0.23**	208.66±0.27	**59.82±0.82**	**85.86±0.64**	109.88±0.65

Table 2: Forecasting performance on (first 1,000 networks) of PeMSD4 and PeMSD8 benchmark datasets.

Model	PeMSD4			PeMSD8		
	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
AGCRN	90.36±0.10	122.61±0.13	176.90±0.35	55.20±0.19	83.01±0.53	167.39±0.25
Z-GCNETs	89.57±0.11	117.94±0.15	180.11±0.26	47.11±0.20	80.25±0.24	98.15±0.33
StemGNN	93.27±0.16	131.49±0.21	189.18±0.30	53.86±0.39	82.00±0.52	**97.78±0.30**
TMP-Nets	**85.15±0.12**	**115.00±0.16**	**170.97±0.22**	**50.20±0.37**	**80.17±0.26**	100.31±0.58

Table 3: Forecasting performance on (first 2,000 networks) PeMSD4 and PeMSD8 benchmark datasets.

Model	PeMSD4	PeMSD8
TMP-Nets	**147.57±0.23**	**85.86±0.64**
$Z_{t,\text{TMP}}$	165.67±0.30	90.23±0.15
$Z^{(\ell)}_{t,\text{Spatial}}$	153.75±0.22	88.38±1.05

Table 4: Ablation Study on PeMSD4 and PESMD8 (RMSE results for first 1000 networks).

future we will explore TMP constructed only on the landmark points, that is, TMP will be constructed not on all but on the most important *landmark* nodes, which would lead to substantial sparsification of the graph representation.

Comparison with Other Topological GNN Models for Dynamic Networks: The two existing time-aware topological GNNs for dynamic networks are TAMP-S2GCNets (Chen et al. 2021) and Z-GCNETs (Chen, Segovia, and Gel 2021). The pivotal distinction between our model and these counterparts lies in the fact that our model serves as a comprehensive extension of both, applicable across diverse data types encompassing point clouds and images (see Section D). Z-GCNETs employs single persistence approach, rendering it unsuitable for datasets that encompass two or more significant domain functions. In contrast, TAMP-S2GCNets employs multipersistence; however, its Euler-Characteristic surface vectorization fails to encapsulate lifespan information present in persistence diagrams. Notably, in scenarios involving sparse data, barcodes with longer lifespans signify main data characteristics, while short barcodes are considered as topological noise. The limitation of Euler-Characteristic Surfaces, being simply a linear combination of bigraded Betti numbers, lies in its inability to capture this distinction. In stark contrast, our framework encompasses all forms of vectorizations, permitting practitioners to choose their preferred vectorization technique while adapting to dynamic networks or time-dependent data comprehensively. For instance, compared to TAMP-S2GCNets model, our TMP-Nets achieves a better performance on the Bytom dataset, i.e., TMP-Nets (MAPE: 28.77±3.30) vs. TAMP-S2GCNets (MAPE: 29.26±1.06). Furthermore, from the computational time perspective, the average computation time of TMP and Dynamic Euler-Poincaré Surface (which is used in TAMP-S2GCNets model) are 1.85 seconds and 38.99 seconds respectively, i.e., our TMP is more efficient.

7 Discussion

We have proposed a new highly computationally efficient summary for multidimensional persistence for time-dependent objects, Temporal MultiPersistence (TMP). By successfully combining the latest TDA methods with deep learning tools, our TMP approach outperforms many popular state-of-the-art deep learning models in a consistent and unified manner. Further, we have shown that TMP enjoys important theoretical stability guarantees. As such, TMP makes an important step toward bringing the theoretical concepts of multipersistence from pure mathematics to the machine learning community and to the practical problems of time-aware learning of time-conditioned objects, such as dynamic graphs, time series, and spatio-temporal processes.

Still, scaling for ultra high-dimensional processes, especially in modern data streaming scenarios, may be infeasible for TMP. In the future, we will investigate algorithms such as those based on landmarks or pruning, with the goal to advance the computational efficiency of TMP for streaming applications.

Acknowledgements

Supported by the NSF grants DMS-2220613, DMS-2229417, ECCS 2039701, TIP-2333703, Simons Foundation grant # 579977, and ONR grant N00014-21-1-2530. Also, the paper is based upon work supported by (while Y.R.G. was serving at) the NSF. The views expressed in the article do not necessarily represent the views of NSF or ONR.

References

Adams, H.; et al. 2017. Persistence images: A stable vector representation of persistent homology. *JMLR*, 18.

Akcora, C. G.; Gel, Y. R.; and Kantarcioglu, M. 2022. Blockchain networks: Data structures of bitcoin, monero, zcash, ethereum, ripple, and iota. *Wiley Int. Reviews: Data Mining and Knowledge Discovery*, 12(1): e1436.

Aktas, M. E.; Akbas, E.; and El Fatmaoui, A. 2019. Persistence homology of networks: methods and applications. *Applied Network Science*, 4(1): 1–28.

Ali, D.; et al. 2023. A survey of vectorization methods in topological data analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Atienza, N.; et al. 2020. On the stability of persistent entropy and new summary functions for topological data analysis. *Pattern Recognition*, 107: 107509.

Bai, L.; Yao, L.; Li, C.; Wang, X.; and Wang, C. 2020. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. *NeurIPS*, 33.

Bin, Y.; et al. 2018. Describing video with attention-based bidirectional LSTM. *IEEE transactions on cybernetics*, 49(7): 2631–2641.

Botnan, M. B.; and Lesnick, M. 2022. An introduction to multiparameter persistence. *arXiv preprint arXiv:2203.14289*.

Botnan, M. B.; Oppermann, S.; and Oudot, S. 2022. Signed barcodes for multi-parameter persistence via rank decompositions. In *SoCG*.

Bubenik, P. 2015. Statistical Topological Data Analysis using Persistence Landscapes. *JMLR*, 16(1): 77–102.

Cao, D.; et al. 2020. Spectral Temporal Graph Neural Network for Multivariate Time-series Forecasting. In *NeurIPS*, volume 33, 17766–17778.

Carlsson, G.; De Silva, V.; and Morozov, D. 2009. Zigzag persistent homology and real-valued functions. In *SoCG*.

Carlsson, G.; and Silva, V. 2010. Zigzag Persistence. *Found. Comput. Math.*, 10(4): 367–405.

Carrière, M.; and Blumberg, A. 2020. Multiparameter persistence image for topological machine learning. *NeurIPS*.

Chazal, F.; Fasy, B. T.; Lecci, F.; Rinaldo, A.; and Wasserman, L. 2014. Stochastic convergence of persistence landscapes and silhouettes. In *SoCG*.

Chen, C.; Petty, K.; Skabardonis, A.; Varaiya, P.; and Jia, Z. 2001. Freeway performance measurement system: mining loop detector data. *Transportation Research Record*, 1748(1): 96–102.

Chen, Y.; Keogh, E.; Hu, B.; Begum, N.; Bagnall, A.; Mueen, A.; and Batista, G. 2015a. The UCR time series classification archive.

Chen, Y.; Segovia, I.; and Gel, Y. R. 2021. Z-GCNETs: time zigzags at graph convolutional networks for time series forecasting. In *ICML*, 1684–1694. PMLR.

Chen, Y.; Segovia-Dominguez, I.; Coskunuzer, B.; and Gel, Y. 2021. TAMP-S2GCNets: coupling time-aware multipersistence knowledge representation with spatio-supra graph convolutional networks for time-series forecasting. In *ICLR*.

Chen, Y.; et al. 2015b. A general framework for never-ending learning from time series streams. *Data mining and knowledge discovery*, 29: 1622–1664.

Chung, Y.-M.; and Lawson, A. 2022. Persistence curves: A canonical framework for summarizing persistence diagrams. *Advances in Computational Mathematics*, 48(1): 6.

Corbet, R.; Fugacci, U.; Kerber, M.; Landi, C.; and Wang, B. 2019. A kernel for multi-parameter persistent homology. *Computers & graphics: X*, 2: 100005.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *NeurIPS*, volume 29, 3844–3852.

Dey, R.; and Salem, F. M. 2017. Gate-variants of Gated Recurrent Unit (GRU) neural networks. In *MWSCAS*.

Dey, T. K.; and Wang, Y. 2022. *Computational Topology for Data Analysis*. Cambridge University Press.

di Angelo, M.; and Salzer, G. 2020. Tokens, Types, and Standards: Identification and Utilization in Ethereum. In *2020 IEEE DAPPS*, 1–10.

Edelsbrunner, H.; and Harer, J. 2010. *Computational topology: an introduction*. American Mathematical Soc.

Eisenbud, D. 2013. *Commutative algebra: with a view toward algebraic geometry*, volume 150. Springer Science & Business Media.

Guo, S.; Lin, Y.; Feng, N.; Song, C.; and Wan, H. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *AAAI*.

Hofer, C.; Graf, F.; Rieck, B.; Niethammer, M.; and Kwitt, R. 2020. Graph filtration learning. In *ICML*, 4314–4323. PMLR.

Jiang, W.; and Luo, J. 2022. Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications*, 207: 117921.

Johnson, M.; and Jung, J.-H. 2021. Instability of the Betti Sequence for Persistent Homology. *J. Korean Soc. Ind. and Applied Math.*, 25(4): 296–311.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. *ICLR*.

Kriege, N. M.; Johansson, F. D.; and Morris, C. 2020. A survey on graph kernels. *Applied Network Science*, 1–42.

Lesnick, M.; and Wright, M. 2022. Computing minimal presentations and bigraded betti numbers of 2-parameter persistent homology. *SIAGA*.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *ICLR*.

Lipton, Z. C.; Berkowitz, J.; and Elkan, C. 2015. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.

Pareja, A.; Domeniconi, G.; Chen, J.; Ma, T.; Suzumura, T.; Kanezashi, H.; Kaler, T.; Schardl, T.; and Leiserson, C. 2020. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *AAAI*.

Schmidhuber, J. 2017. LSTM: Impact on the world's most valuable public companies. http://people.idsia.ch/~juergen/impact-on-most-valuable-companies.html. Accessed: 2020-03-19.

Segovia Dominguez, I.; et al. 2021. Does Air Quality Really Impact COVID-19 Clinical Severity: Coupling NASA Satellite Datasets with Geometric Deep Learning. *KDD*.

Segovia-Dominguez, I.; et al. 2021. TLife-LSTM: Forecasting Future COVID-19 Progression with Topological Signatures of Atmospheric Conditions. In *PAKDD (1)*, 201–212.

Shamsi, K.; Victor, F.; Kantarcioglu, M.; Gel, Y.; and Akcora, C. G. 2022. Chartalist: Labeled Graph Datasets for UTXO and Account-based Blockchains. *NeurIPS*.

Shewalkar, A.; Nyavanandi, D.; and Ludwig, S. A. 2019. Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU. *J. Artificial Intelligence and Soft Computing Research*, 9(4): 235–245.

Shin, S.; and Kim, W. 2020. Skeleton-Based Dynamic Hand Gesture Recognition Using a Part-Based GRU-RNN for Gesture-Based Interface. *IEEE Access*, 8: 50236–50243.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, 3104–3112. Cambridge, MA, USA: MIT Press.

Vassilevska, V.; Williams, R.; and Yuster, R. 2006. Finding the Smallest H-Subgraph in Real Weighted Graphs and Related Problems. In *Automata, Languages and Programming*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph attention networks. *ICLR*.

Vipond, O. 2020. Multiparameter Persistence Landscapes. *J. Mach. Learn. Res.*, 21: 61–1.

Wang, C.; Yang, H.; and Meinel, C. 2018. Image Captioning with Deep Bidirectional LSTMs and Multi-Task Learning. *ACM Trans. Multimedia Comput. Commun. Appl.*, 14(2s).

Weber, M.; et al. 2019. Anti-Money Laundering in Bitcoin: Experimenting with GCNs for Financial Forensics. In *KDD*.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; and Zhang, C. 2019. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *IJCAI*.

Xiang, Z.; Yan, J.; and Demir, I. 2020. A rainfall-runoff model with LSTM-based sequence-to-sequence learning. *Water resources research*, 56(1): e2019WR025326.

Yan, S.; Xiong, Y.; and Lin, D. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*.

Yao, H.; Wu, F.; Ke, J.; Tang, X.; Jia, Y.; Lu, S.; Gong, P.; Ye, J.; and Li, Z. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *AAAI*.

Yu, B.; Yin, H.; and Zhu, Z. 2018. Spatio-temporal GCNs: a deep learning framework for traffic forecasting. In *IJCAI*.

Yu, Y.; Si, X.; Hu, C.; and Zhang, J. 2019. A Review of Recurrent Neural Networks: Lstm Cells and Network Architectures. *Neural Comput.*, 31(7): 1235–1270.