Z-GCNETs: Time Zigzags at Graph Convolutional Networks for Time Series Forecasting

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

Abstract

There recently has been a surge of interest in developing a new class of deep learning (DL) architectures that integrate an explicit time dimension as a fundamental building block of learning and representation mechanisms. In turn, many recent results show that topological descriptors of the observed data, encoding information on the shape of the dataset in a topological space at different scales, that is, persistent homology of the data, may contain important complementary information, improving both performance and robustness of DL. As convergence of these two emerging ideas, we propose to enhance DL architectures with the most salient time-conditioned topological information of the data and introduce the concept of zigzag persistence into time-aware graph convolutional networks (GCNs). Zigzag persistence provides a systematic and mathematically rigorous framework to track the most important topological features of the observed data that tend to manifest themselves over time. To integrate the extracted time-conditioned topological descriptors into DL, we develop a new topological summary, zigzag persistence image, and derive its theoretical stability guarantees. We validate the new GCNs with a time-aware zigzag topological layer (Z-GCNETs), in application to traffic forecasting and Ethereum blockchain price prediction. Our results indicate that Z-GCNET outperforms 13 state-of-the-art methods on 4 time series datasets.

[1]Department of Statistical Science, Southern Methodist University, TX, USA [2]Energy Storage & Distributed Resources Division, Lawrence Berkeley National Laboratory, CA, USA [3]Department of Mathematical Sciences, University of Texas at Dallas, TX, USA [4]NASA Jet Propulsion Laboratory, CA, USA. Correspondence to: Yuzhou Chen <yuzhouc@smu.edu>, Ignacio Segovia Dominguez <ignacio.segoviadominguez@utdallas.edu>, Yulia R. Gel <ygl@utdallas.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

1. Introduction

Many real world phenomena are intrinsically dynamic by nature, and ideally neural networks, encoding the knowledge about the world should also be based on more explicit time-conditioned representation and learning mechanisms. However, most currently available deep learning (DL) architectures are inherently static and do not systematically integrate time-dimension into the learning process. As a result, such model architectures often cannot reliably, accurately and on time learn many salient time-conditioned characteristics of complex interdependent systems, resulting in outdated decisions and requiring frequent model updates.

In turn, in the last few years we observe an increasing interest to integrate deep neural network architectures with persistent homology representations of the learned objects, typically in a form of some topological layer in DL (Hofer et al., 2019; Carrière et al., 2020; Carlsson & Gabrielsson, 2020). Such persistent homology representations allow us to extract and learn descriptors of the object *shape*. (By shape here we broadly understand data characteristics that are invariant under continuous transformations such as bending, stretching, and compressing.) Such interest in combining persistent homology representations with DL is explained by the complementary multi-scale information topological descriptors deliver about the underlying objects, and higher robustness of these salient object characterisations to perturbations.

Here we take the first step toward merging the two directions. To enhance DL with the most salient *time-conditioned topological* information, we introduce the concept of zigzag persistence into time-aware DL. Building on the fundamental results on quiver representations, zigzag persistence studies properties of topological spaces which are connected via inclusions going in both directions (Carlsson & Silva, 2010; Tausz & Carlsson, 2011; Carlsson, 2019). Such generalization of ordinary persistent homology allows us to track topological properties of time-conditioned objects by extracting salient *time-aware topological features* through time-ordered inclusions. We propose to summarize the extracted time-aware persistence in a form of zigzag persistence images and then to integrate the resulting information as a learnable time-aware zigzag layer into GCN.

The key novelty of our paper can be summarized as follows:

- This is the first approach bridging time-conditioned DL with time-aware persistent homology representations of the data.
- We propose a new vectorized summary for time-aware persistence, namely, *zigzag persistence image* and discuss its theoretical stability guarantees.
- We introduce the concepts of time-aware zigzag persistence into learning time-conditioned graph structures and develop a zigzag topological layer (Z-GCNET) for time-aware graph convolutional networks (GCNs).
- Our experiments on application Z-GCNET to traffic forecasting and Ethereum blockchain price prediction show that Z-GCNET surpasses 13 state-of-the-art methods on 4 benchmark datasets, both in terms of accuracy and robustness.

2. Related Work

Zigzag Persistence is yet an emerging tool in applied topological data analysis, but many recent studies have already shown its high utility in such diverse applications as brain sciences (Chowdhury et al., 2018), imagery classification (Adams et al., 2020), cyber-security of mobile sensor networks (Adams & Carlsson, 2015; Gamble et al., 2015), and characterization of flocking and swarming behavior in biological sciences (Corcoran & Jones, 2017; Kim et al., 2020). An alternative to zigzag but a closely related approach to assess properties of time-varying data with persistent homology, namely, crocker stacks, has been recently suggested by Xian et al. (2020), though the crocker stacks representations are not learnable in DL models. While zigzag has been studied in conjunction with dynamic systems (Tymochko et al., 2020) and time-evolving point clouds (Corcoran & Jones, 2017), till now, the utility of zigzag persistence remains untapped not only in conjunction with GCNs but with any other DL tools.

Time series forecasting From a deep learning perspective, Recurrent Neural Networks (RNNs) are natural methods to model time-dependent datasets (Yu et al., 2019). In particular, the stable architecture of Long Short Term Memories (LSTMs), and its variant called Gate Recurrent Unit (GRU), solves the gradient instability of predecessors and adds extra flexibility due to their memory storage and forget gates. The ability of LSTM and GRU to selectively learn historical patterns led to their wide spread adoption as one of the main DL tools for time-dependant objects (Schmidhuber, 2017; Shin & Kim, 2020; Segovia-Dominguez et al., 2021). In general, GRU models tend to have fewer parameters than LSTMs but exhibit similar forecasting performance (Greff et al., 2017; Gao et al., 2020). However, applications of RNN are limited by the underlying structure of the input data, for instance, these methods are not designed to handle objects from non-Euclidean spaces, such as graphs and manifolds.

Graph convolutional networks To overcome the limitations of traditional convolution on graph structured data, graph convolution-based methods (Defferrard et al., 2016; Kipf & Welling, 2017; Veličković et al., 2018) are proposed to explore both global and local structures. GCNs usually consists of graph convolution layers which extract the edge characteristics between neighbor nodes and aggregate feature information from neighborhood via graph filters. In addition to convolution, there has been a surge of interest in applying GCNs to time series forecasting tasks (Yu et al., 2018; Yao et al., 2018; Yan et al., 2018; Guo et al., 2019; Weber et al., 2019; Pareja et al., 2020; Segovia Dominguez et al., 2021). Although these methods have achieved state-of-the-art performance in traffic flow forecasting, human action recognition, and anti-money laundering regulation, the design of spatial temporal graph convolution network framework is mostly based on modeling spatial-temporal correlation in terms of feature-level and pre-defined graph structure.

3. Time-Aware Topological Signatures of Graphs

Spatio-temporal Data as Graph Structures The spatial-temporal networks can be represented as a sequence of discrete snapshots, $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_T\}$, where $\mathcal{G}_t = \{\mathcal{V}_t, \mathcal{E}_t, W_t\}$ is the graph structure at time step t, $t = 1, \ldots, T$. In $\mathcal{G}_t$, $\mathcal{V}_t$ is a node set with cardinality $|\mathcal{V}_t|$ of N_t and $\mathcal{E}_t \subseteq \mathcal{V}_t \times \mathcal{V}_t$ is an edge set. A nonnegative symmetric $N_t \times N_t$-matrix W_t with entries $\{\omega_{ij}^t\}_{1\leq i,j\leq N_t}$ represents the adjacency matrix of $\mathcal{G}_t$, that is, $\omega_{ij}^t > 0$ for any $e_{ij}^t \in \mathcal{E}_t$ and $\omega_{ij}^t = 0$, otherwise. Let $F, F \in \mathbb{Z}_{>0}$ be the number of different node features associated each node $v \in \mathcal{V}_t$. Then, a $N_t \times F$ feature matrix $\boldsymbol{X}_t$ serves as an input to the framework of time series process modeling. Throughout the paper we suppress subscript t, for the sake of notations, unless dependency over time is emphasized in the particular context.

Background on Persistent Homology Persistent homology is a mathematical machinery to extract the intrinsic shape properties of graph $\mathcal{G}$ that are invariant under continuous transformations such as bending, stretching, and twisting. The key idea is, based on some appropriate scale parameter, to associate $\mathcal{G}$ with a graph filtration $\mathcal{G}^1 \subseteq \ldots \subseteq \mathcal{G}^n = \mathcal{G}$ and then to equip each $\mathcal{G}^i$ with an abstract simplicial complex $\mathscr{C}(\mathcal{G}^i)$, $1 \leq i \leq n$, yielding a filtration of complexes $\mathscr{C}(\mathcal{G}^1) \subseteq \ldots \subseteq \mathscr{C}(\mathcal{G}^n)$. Now, we can systematically and efficiently track evolution of various patterns such as connected components, cycles, and voids throughout this hierarchical sequence of complexes. Each topological feature, or p-hole (e.g., number of connected components and voids), $0 \leq p \leq \mathbb{D}$, is represented by a unique pair (i_b, j_d), where birth i_b and death j_d are the scale parameters at which the feature first appears and disappears, respectively, and $\mathbb{D}$ is the highest dimension of the simplicial complexes. The lifespan of the feature is defined as $j_d - i_b$. The extracted topological information can be then summarized as a persistence diagram $\mathrm{Dgm} = \{(i_b, j_d) \in \mathbb{R}^2 | i_b < j_d\}$. Multiplicity of a point $(i_b, j_d) \in \mathcal{D}$ is the number of p-dimensional topological features (p-holes) that are born and die at i_b and j_b, respectively. Points at the diagonal Dgm are taken with infinite multiplicity. The idea is then to evaluate topological features that persist (i.e., have longer lifespan) over the complex filtration and, hence, are likelier to contain important structural information on the graph.

Finally, filtration of the weighted graph $\mathcal{G}$ can be constructed in multiple ways. For instance, (i) we can select a scale parameter as a shortest weighted path between any two nodes; then as an abstract simplicial complex $\mathscr{C}$ on $\mathcal{G}$, consider a Vietoris–Rips (VR) complex $\mathscr{V}\mathscr{R}^{\nu_*}(\mathcal{G}) = \{\mathcal{G}^{'} \subseteq \mathcal{G} | diam(\mathcal{G}^{'}) \leq \nu_*\}$. That is, Vietoris-Rips complex $\mathscr{V}\mathscr{R}^{\nu_*}(\mathcal{G})$ is generated by subgraphs $\mathcal{G}^{'}$ of bounded diameter ν_* (i.e. any subgraph $\mathcal{G}^{'}$ of k-nodes with $diam(\mathcal{G}^{'}) \leq \nu_*$ generates a $(k-1)$-simplex in $\mathscr{V}\mathscr{R}^{\nu_*}(\mathcal{G})$). Hence, for a set of scale thresholds $\nu_1 \leq \ldots \leq \nu_n$, we obtain a VR filtration $\mathscr{V}\mathscr{R}^1 \subseteq \ldots \subseteq \mathscr{V}\mathscr{R}^n$. Alternatively, (ii) we can consider a sublevel filtration induced by a continuous function f defined on the nodes set $\mathcal{V}$ of $\mathcal{G}$. Let $f: \mathcal{V} \to \mathbb{R}$ and $\nu_1 < \nu_2 < \ldots < \nu_n$ be a sequence of sorted filtered values, then $\mathscr{C}^i = \{\sigma \in \mathscr{C} : \max_{v \in \sigma} f(v) \leq \nu_i\}$. Note that a VR filtration (i) is a subcase of sublevel filtration (ii) with f being the diameter function (Adams et al., 2017; Bauer, 2019).

Time-Aware Zigzag Persistence Since our primary aim is to assess interconnected evolution of multiple time-conditioned objects, the developed methodology for tracking topological and geometric properties of these objects shall ideally account for their intrinsically dynamic nature. We address this goal by introducing the concept of *zigzag* persistence into GCN. Zigzag persistence is a generalization of persistent homology proposed by (Carlsson & Silva, 2010) and provides a systematic and mathematically rigorous framework to track the most important topological features of the data persisting over time.

Let $\{\mathcal{G}_t\}_1^T$ be a sequence of networks observed over time. The key idea of zigzag persistence is to evaluate pairwise compatible topological features in this time-ordered sequence of networks. First, we define a set of network inclusions over time

$$\begin{tikzcd} \mathcal{G}_1 \arrow[dr] & & \mathcal{G}_2 \arrow[dl] \arrow[dr] & & \mathcal{G}_3 \arrow[dl] & \ldots \\ & \mathcal{G}_1 \cup \mathcal{G}_2 & & \mathcal{G}_2 \cup \mathcal{G}_3 & & \end{tikzcd},$$

where $\mathcal{G}_k \cup \mathcal{G}_{k+1}$ is defined as a graph with a node set $V_k \cup V_{k+1}$ and an edge set $E_k \cup E_{k+1}$. Second, we fix a scale parameter ν_* and build a zigzag diagram of simplicial complexes for the given ν_* over the constructed set of network inclusions

$$\begin{tikzcd} \mathscr{C}(\mathcal{G}_1, \nu_*) \arrow[dr] & & \mathscr{C}(\mathcal{G}_2, \nu_*) \arrow[dl] \arrow[dr] & \\ & \mathscr{C}(\mathcal{G}_1 \cup \mathcal{G}_2, \nu_*) & & \ldots \end{tikzcd}$$

Using the zigzag filtration for the given ν_*, we can track birth and death of each topological feature over $\{\mathcal{G}_t\}_1^T$ as time points t_b and t_d, $1 \leq t_b \leq t_d \leq T$, respectively. Similarly to a non-dynamic case, we can extend the notion of persistence diagram for the analysis of topological characteristics of time-varying data delivered by the zigzag persistence.

Definition 3.1 (Zigzag Persistence Diagram (ZPD))**.** Let t_b and t_d be time points, when a topological feature first appears (i.e., is born) and disappears (i.e., dies) in the time period $[1, T]$ over the zigzag diagram of simplicial complexes for a fixed scale parameter ν_*, respectively. If the topological feature first appears in $\mathscr{C}(\mathcal{G}_k, \nu_*)$, $t_b = k$; if it first appears in $\mathscr{C}(\mathcal{G}_k \cup \mathcal{G}_{k+1}, \nu_*)$, $t_b = k+1/2$. Similarly, if a topological feature last appears in $\mathscr{C}(\mathcal{G}_k, \nu_*)$, $t_d = k$; and if it last appears in $\mathscr{C}(\mathcal{G}_k \cup \mathcal{G}_{k+1}, \nu_*)$, $t_d = k+1/2$. A multiset of points in $\mathbb{R}^2$, $\mathrm{DgmZZ}_{\nu_*} = \{(\mathrm{t_b}, \mathrm{t_d}) \in \mathbb{R}^2 | \mathrm{t_b} < \mathrm{t_b}\}$, for a fixed ν_* is called a *zigzag persistence diagram (ZPD)*.

Inspired by the notion of a persistent image as a summary of ordinary persistence (Adams et al., 2017), to input topological information summarized by ZPD into a GCN, we propose a representation of ZPD as *zigzag persistence image* (ZPI). ZPI is a finite-dimensional vector representation of a ZPD and can be computed through the following steps:

- *Step 1:* Map a zigzag persistence diagram DgmZZ_{ν_*} to an integrable function $\rho_{\mathrm{DgmZZ}_{\nu_*}}: \mathbb{R}^2 \to \mathbb{R}^2$, called a *zigzag persistence surface*. The zigzag persistence surface is given by sums of weighted Gaussian functions that are centered at each point in DgmZZ_{ν_*}, i.e.

$$\rho_{\mathrm{DgmZZ}_{\nu_*}} = \sum_{\mu \in \mathrm{DgmZZ}'_{\nu_*}} g(\mu) \, e^{\left\{-\frac{||z-\mu||^2}{2\vartheta^2}\right\}}.$$

  Here DgmZZ'_{ν_*} is the transformed multi-set in DgmZZ_{ν_*}, i.e., $\mathrm{DgmZZ}'_{\nu_*}(\mathrm{x}, \mathrm{y}) = (\mathrm{x}, \mathrm{y} - \mathrm{x})$; $g(\mu)$ is a weighting function with mean $\mu = (\mu_x, \mu_y) \in \mathbb{R}^2$ and variance ϑ^2, which depends on the distance from the diagonal.

- *Step 2:* Perform a discretization of a subdomain of zigzag persistence surface $\rho_{\mathrm{DgmZZ}_{\nu_*}}$ in a grid.

- *Step 3:* The ZPI, i.e., a matrix of pixel values, can be obtained by subsequent integration over each grid box.

The value of each pixel $z \in \mathbb{R}^2$ within a ZPI is then defined as:

$$\mathrm{ZPI}_{\nu_*}(\mathrm{z}) = \iint\limits_{\mathrm{z}} \sum_{\mu \in \mathrm{DgmZZ}'_{\nu_*}} \mathrm{g}\left(\mu\right) \mathrm{e}^{\left\{-\frac{||\mathrm{z}-\mu||^2}{2\vartheta^2}\right\}} \mathrm{dz_x dz_y}.$$

Proposition 3.1. *Let $g : \mathbb{R}^2 \to \mathbb{R}$ be a non-negative continuous and piece-wise differentiable function. Let DgmZZ_{ν_*} be a zigzag persistence diagram for some fixed scale parameter ν_*, and let ZPI_{ν_*} be its corresponding zigzag persistence image. Then, ZPI_{ν_*} is stable with respect to the Wasserstein-1 distance between zigzag persistence diagrams.*

Derivations of the proposition can be found in Appendix D of the supplementary material.

Tracking evolution of topological patterns in these sequences of time-evolving graphs allows us to glean insights into which properties of the observed time-conditioned objects, e.g., traffic data or Ethereum transaction graphs, tend to persist over time and, hence, are likelier to play a more important role in predictive tasks.

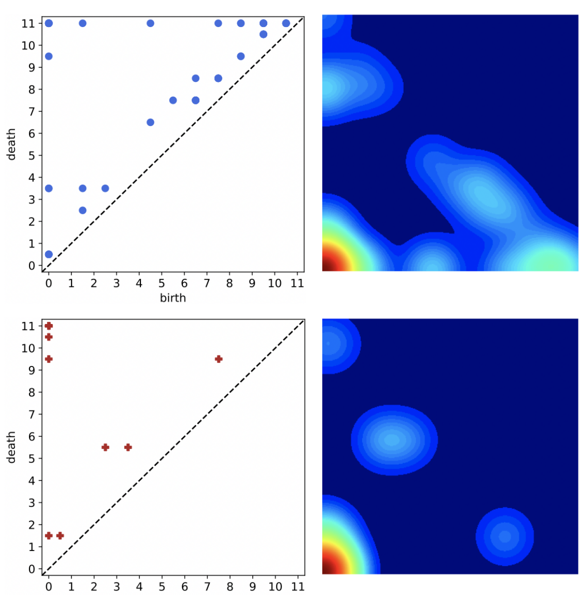

Figure 1. Illustrations of 0- and 1-dimensional ZPD and 0- and 1-dimensional ZPI for PeMSD4 dataset using the sliding window size $\tau = 12$, i.e., dynamic network with 12 graphs. Upper part shows the 0-dimensional ZPD and ZPI whilst the lower part is the 1-dimensional ZPD and ZPI.

4. Z-GCNETs

Given the graph $\mathcal{G}$ and graph signals $\boldsymbol{X}^\tau = \{\boldsymbol{X}_{t-\tau}, \ldots, \boldsymbol{X}_{t-1}\} \in \mathbb{R}^{\tau \times N \times F}$ of τ past time periods (i.e. window size τ; where $\boldsymbol{X}_i \in \mathbb{R}^{N \times F}$ and $i \in \{t-\tau, \ldots, t-1\}$), we employ a model targeted on multi-step time series forecasting. That is, given the windows size τ of past graph signals and the ahead horizon size h, our goal is to learn a mapping function which maps the historical data $\{\boldsymbol{X}_{t-\tau}, \ldots, \boldsymbol{X}_{t-1}\}$ into the future data $\{\boldsymbol{X}_t, \ldots, \boldsymbol{X}_{t+h}\}$.

Laplacianlink In spatial-temporal domain, the topology of graph may have different structure at different points in time. In this paper, we use the self-adaptive adjacency matrix (Wu et al., 2019) as the normalized Laplacian by trainable node embedding dictionaries $\boldsymbol{\phi} \in \mathbb{R}^{N \times c}$, i.e., $\boldsymbol{L} = softmax(ReLU(\boldsymbol{\phi}\boldsymbol{\phi}^\top))$, where the dimension of embedding $c \geq 1$. Although introducing node embedding dictionaries allows capture hidden spatial dependence information, it cannot sufficiently capture the global graph information and the similarity between nodes. To overcome the limits and explore neighborhoods of nodes at different depths, we define a new polynomial representation for Laplacian based on positive powers of the Laplacian matrix. Laplacianlink $\tilde{\boldsymbol{L}}$ is then formulated as:

$$\tilde{\boldsymbol{L}} = \left[\boldsymbol{I}, \boldsymbol{L}, \boldsymbol{L}^2, \ldots, \boldsymbol{L}^K\right] \in \mathbb{R}^{N \times N \times (K+1)}, \quad (1)$$

where $K \geq 1$, $\boldsymbol{I} \in \mathbb{R}^{N \times N}$ represents the identity matrix, and $\boldsymbol{L}^k \in \mathbb{R}^{N \times N}$, with $0 \leq k \leq K$, denotes the power series of normalized Laplacian.

By *linking* (i.e., stacking) the power series of normalized Laplacian, we build a diffusion formalism to accumulate neighbors' information of different power levels. Hence, each node will successfully exploit and propagate spatial-temporal correlations after spatial and temporal graph convolutional operations.

Spatial graph convolution To model the spatial network $\mathcal{G}_t$ at timestamp t with its node feature matrix $\boldsymbol{X}_t$, we define the spatial graph convolution as multiplying the input of each layer with the Laplacianlink $\tilde{\boldsymbol{L}}$, which is then fed into the trainable projection matrix $\boldsymbol{\Theta} = \boldsymbol{\phi}\boldsymbol{V}$ (where $\boldsymbol{V}$ stands for the trainable weight). In spatial-temporal graph modeling, we prefer to use weight sharing in matrix factorization rather than directly assigning a trainable weight matrix in order not only to avoid the risk of over-fitting but also to reduce the computational complexity. We compute the transformation in spatial domain, in each layer, as follows:

$$\boldsymbol{H}^{(\ell)}_{i,\mathcal{S}} = (\tilde{\boldsymbol{L}}\boldsymbol{H}^{(\ell-1)}_{i,\mathcal{S}})^\top \boldsymbol{\phi}\boldsymbol{V}, \quad (2)$$

where $\boldsymbol{\phi} \in \mathbb{R}^{N \times c}$ is the node embedding and $\boldsymbol{V} \in \mathbb{R}^{c \times (K+1) \times C_{in} \times (C_{out}/2)}$ is the trainable weight (C_{in} and

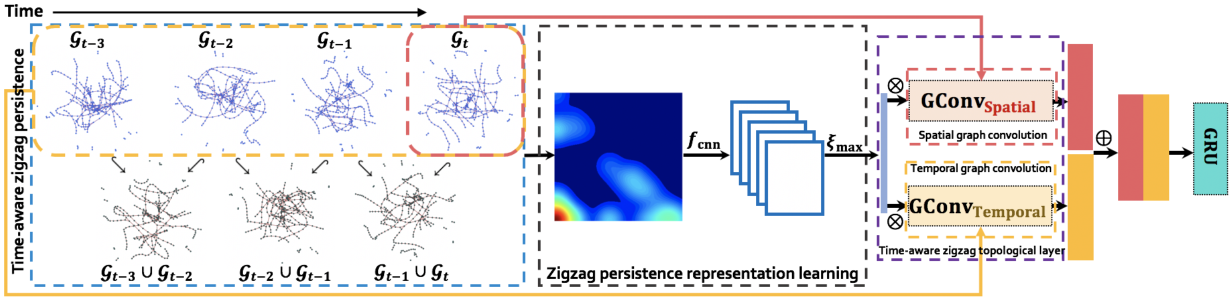

Figure 2. The architecture of Z-GCNETs. Given a sliding window, e.g. $(\mathcal{G}_{t-3}, \ldots, \mathcal{G}_t)$, we extract zigzag persistence image (ZPI) based on zigzag filtration. For the ZPI $\in \mathbb{R}^2$ with the shape of $p \times p$, Z-GCNETs first learn the topological features of ZPI through CNN-based framework, and then apply global max-pooling to obtain the maximum values among pooled activation maps. The output of zigzag persistence representation learning is decoded into spatial graph convolution and temporal graph convolution, where the inputs of spatial graph convolution and temporal graph convolution are current timestamps, e.g., $\mathcal{G}_t$ in red dashed box, and the sliding window (i.e., $(\mathcal{G}_{t-3}, \ldots, \mathcal{G}_t)$ in yellow dashed box) respectively. After graph convolution operations, features from time-aware zigzag topological layer are combined and moved to GRU to perform forecasting. Symbol $\otimes$ represents dot product whilst $\oplus$ denotes combination.

C_{out} are the number of channels in input and output, respectively). Finally, $\boldsymbol{H}_{i,\mathcal{S}}^{(\ell-1)} \in \mathbb{R}^{N\times(C_{in}/2)}$ is the matrix of activations of spatial graph convolution to the ℓ-th layer and $\boldsymbol{H}_{i,\mathcal{S}}^{(0)} = \boldsymbol{X}_i$. As a result, all information regarding the ℓ-layered input at time i are reflected in the latest state variable.

Temporal graph convolution In addition to spatial domain, the nature of spatial-temporal networks includes temporal relationships among multiple spatial networks. To extract temporal dependency patterns, we choose longer window size (i.e., by using the entire sliding window as input) and apply temporal graph convolution to graph signals in sliding window $\boldsymbol{X}^{\tau}$. The mechanism has several excellent properties: (i) there is no need to select a particular size of the nested sliding window, (ii) temporal dependency patterns can be well captured and evaluated by (longer) window sizes, whereas shorter window sizes (i.e., nested sliding window) are likelier to be biased and noisy, and (iii) the sliding window $\boldsymbol{X}^{\tau}$ enhances efficiency of estimating temporal dependencies. The temporal graph convolution is given by:

$$\boldsymbol{H}_{i,\mathcal{T}}^{(\ell)} = \left((\tilde{\boldsymbol{L}}\boldsymbol{H}_{i,\mathcal{T}}^{(\ell-1)})^{\top}\phi\boldsymbol{U}^{(\ell-1)}\right)\boldsymbol{Q}, \tag{3}$$

where $\boldsymbol{U}^{(\ell)} \in \mathbb{R}^{c\times C_{in}\times(C_{out}/2)}$ is trainable weight and $\boldsymbol{U}^{(0)} \in \mathbb{R}^{c\times F\times(C_{out}/2)}$, $\boldsymbol{Q} \in \mathbb{R}^{\tau\times 1}$ is the trainable projection vector in temporal graph convolutional layer, and $\boldsymbol{H}_{i,\mathcal{T}}^{(\ell-1)} \in \mathbb{R}^{\tau\times N\times(C_{in}/2)}$ is the hidden matrix fed to the ℓ-th layer and $\boldsymbol{H}_{i,\mathcal{T}}^{(0)} = \boldsymbol{X}^{\tau} \in \mathbb{R}^{\tau\times N\times F}$.

Time-aware zigzag topological layer To learn the topological features across a range of spatial and temporal scales, we extend the CNN model to be used along with ZPI. In this paper, we present a framework to aggregate the topological persistent features into the feature representation learned from GCN. Let ZPI^{τ} denote the ZPI based on the sliding window $\boldsymbol{X}^{\tau}$. (Here for brevity we suppress dependence of ZPI on a scale parameter ν_*.) We design the time-aware zigzag topological layer to (i) extract and learn the spatial-temporal topological features contained in ZPI, (ii) aggregate transformed information from (spatial or temporal) graph convolution and spatial-temporal topological information from zigzag persistence module, and (iii) mix spatial-temporal and spatial-temporal topological information. The information's extraction, aggregation, and combination processes are expressed as:

$$\begin{aligned}
\boldsymbol{Z}^{(\ell)} &= \xi_{\text{max}}\left(f_{\text{cnn}}^{(\ell)}(\text{ZPI}^{\tau})\right), \\
\boldsymbol{S}_i^{(\ell)} &= \boldsymbol{H}_{i,\mathcal{S}}^{(\ell)}\boldsymbol{Z}^{(\ell)}, \\
\boldsymbol{T}_i^{(\ell)} &= \boldsymbol{H}_{i,\mathcal{T}}^{(\ell)}\boldsymbol{Z}^{(\ell)}, \\
\boldsymbol{H}_{i,out}^{(\ell)} &= \text{COMBINE}^{(\ell)}(\boldsymbol{S}_i^{(\ell)}, \boldsymbol{T}_i^{(\ell)}),
\end{aligned} \tag{4}$$

where $f_{\text{cnn}}^{(\ell)}$ represents the convolutional neural network (CNN) in the ℓ-th layer, $\xi_{\text{max}}(\cdot)$ denotes global max-pooling operation, $\boldsymbol{Z}^{(\ell)} \in \mathbb{R}^{(C_{out}/2)}$ is the learned zigzag persistence representation from CNN, $\boldsymbol{S}_i^{(\ell)} \in \mathbb{R}^{N\times(C_{out}/2)}$ is the aggregated spatial^2-temporal representation, $\boldsymbol{T}_i^{(\ell)} \in \mathbb{R}^{N\times(C_{out}/2)}$ is the aggregated spatial-temporal^2 representation, and the output of time-aware zigzag topological layer $\boldsymbol{H}_{i,out}^{(\ell)} \in \mathbb{R}^{N\times C_{out}}$ combines hidden states $\boldsymbol{S}_i^{(\ell)}$ and $\boldsymbol{T}_i^{(\ell)}$ at time i.

GRU with time-aware zigzag topological layer GRU is a variant of the LSTM network. Compared with LSTM, GRU has a simpler structure, fewer training parameters, and more easily overcome vanishing and exploding gradient problems. The feed forward propagation of GRU with time-aware

Table 1. Summary of datasets used in time series forecasting tasks. [‡] means the average number of edges in transportation networks under threshold ν_*.

Dataset	# Nodes	Avg # edges	Time range
Bytom	100	9.98	27/07/2017 - 07/05/2018
Decentraland	100	16.94	14/10/2017 - 07/05/2018
PeMSD4	307	316.10‡	01/01/2018 - 28/02/2018
PeMSD8	170	193.53‡	01/07/2016 - 31/08/2016

zigzag topological layer is recursively conducted as:

$$\begin{aligned} \boldsymbol{z}_i &= \varphi\left(\boldsymbol{W}_z\left[\boldsymbol{O}_{i-1}, \boldsymbol{H}_{i,out}\right] + \boldsymbol{b}_z\right), \\ \boldsymbol{r}_i &= \varphi\left(\boldsymbol{W}_r\left[\boldsymbol{O}_{i-1}, \boldsymbol{H}_{i,out}\right] + \boldsymbol{b}_r\right), \\ \tilde{\boldsymbol{O}}_i &= \tanh\left(\boldsymbol{W}_o\left[\boldsymbol{r}_i \odot \boldsymbol{O}_{i-1}, \boldsymbol{H}_{i,out}\right] + \boldsymbol{b}_o\right), \\ \tilde{\boldsymbol{O}}_i &= \boldsymbol{z}_t \odot \boldsymbol{O}_{i-1} + (1 - \boldsymbol{z}_t) \odot \tilde{\boldsymbol{O}}_i, \end{aligned} \tag{5}$$

where $\varphi(\cdot)$ is a non-linear function, i.e., the ReLU function; $\odot$ is the elementwise product; $\boldsymbol{z}_i$ and $\boldsymbol{r}_i$ are update gate and reset gate, respectively; $\boldsymbol{b}_z$, $\boldsymbol{b}_r$, $\boldsymbol{b}_o$, $\boldsymbol{W}_z$, $\boldsymbol{W}_r$, and $\boldsymbol{W}_o$ are trainable parameters; $[\boldsymbol{O}_{i-1}, \boldsymbol{H}_{i,out}]$ and $\boldsymbol{O}_i$ are the input and output of GRU model, respectively. In this way, Z-GCNETs contains structural, temporal, and topological information.

Figure 2 depicts the framework of our proposed Z-GCNETs model. As illustrated in Figure 2, Z-GCNETs contains the four major steps. First, we use CNN base model (f_{cnn}) to learn the topological features of an ZPI and then employ global max-pooling (ξ_{max}) to the corresponding feature maps to obtain image-level feature representation (see the *black* dashed box). Second, *purple* dashed box, in the spatial and temporal dimensions, i.e., (i) we use spatial graph convolution to capture spatial correlations between nodes and get $\boldsymbol{H}_S$ (Equation 2); (ii) we use temporal graph convolution to capture temporal correlations between features in different time slices and get $\boldsymbol{H}_T$ (Equation 3). Third, we can get hidden states S (*red* dashed block) and T (*yellow* dashed block) by applying the $\boldsymbol{H}_S$ and $\boldsymbol{H}_T$ to ZPI representation, respectively; then combine S and T and get output of time-aware zigzag topological layer $\boldsymbol{H}_{out}$ (Equation 4). Fourth, we pass $\boldsymbol{H}_{out}$ to the GRU for modeling the temporal dependency.

5. Experiments

5.1. Datasets

We consider two types of networks (i) traffic network and (ii) Ethereum token network. Statistical overview of all datasets is given in Table 1. We now describe the detailed construction of traffic and Ethereum transaction networks as follows (i) The freeway Performance Measurement System (PeMS) data sources (i.e., PeMSD4 and PeMSD8) (Chen et al., 2001) collects real time traffic data in California. Both PeMSD4 and PeMSD8 datasets are aggregated to 5 minutes, therefore there are overall 16,992 and 17,856 data points in PeMSD4 and PeMSD8, respectively. In the traffic network, the node is represented by the loop detector which can detect real time measurement of traffic conditions and the edge is a freeway segment between two nearest nodes. Hence, the node set $\mathcal{V}_t \equiv \mathcal{V}$ and the node feature matrix of traffic network $X_t \in \mathbb{R}^{N \times 3}$ denotes that each node has 3 features (i.e., flow rate, speed, and occupancy) at time t. To capture both spatial and temporal dependencies, we reconstruct the traffic graph structure $\mathcal{G}_t = \{\mathcal{V}, \mathcal{E}_t, W_t^{\nu_*}\}$ at time t. Here, we define the *right censoring* weight $W_t^{\nu_*}$

$$\omega_{t,uv}^{\nu_*} = \begin{cases} w_{t,uv} & (u,v) \in \mathcal{E} \text{ and } w_{t,uv} \leq \nu_* \\ 0 & (u,v) \in \mathcal{E} \text{ and } w_{t,uv} > \nu_* \\ 0 & (u,v) \notin \mathcal{E} \end{cases}, \tag{6}$$

where $w_{t,uv} = e^{-||x_{t,u} - x_{t,v}||^2/\gamma}$ is based on the Radial Basis Function (RBF). To investigate how the traffic graph structure evolves over time, at each time point t we keep only edges with weights $\omega_{t,uv}^{\nu_*}$ which are no greater than some positive threshold ν_*. Hence, the resulting graph is dynamic, that is, its edge set changes over the considered time period. In our experiments, we assign parameter $\gamma = 1.0$ to RBF and set the thresholds in PeMSD4 and PeMSD8 to $\nu_* = 0.5$ and $\nu_* = 0.3$, respectively. (ii) The Ethereum blockchain was developed in 2014 to implement Smart Contracts, which are used to create and sell digital assets on the network[1]. In particular, token assets are specially valuable because each token naturally represents a network layer with the same nodes, i.e., addresses of users, appearing in the networks, i.e., layers, of multiple tokens (Akcora et al., 2021; Li et al., 2020; di Angelo & Salzer, 2020). For our experiments, we extract two token networks with more than \$100M in market value[2], Bytom and Decentraland tokens, from the publicly available Ethereum blockchain. We focus our analysis on the dynamic network generated by the daily transactions on each token network, and historical daily closed prices[2]. Since each token has different creation date[3], Bytom dynamic network contains 285 nets whilst Decentraland dynamic network has 206 nets. Ethereum's token networks have an average of 442788/1192722 nodes/edges. To maintain a reasonable computation time, we obtain a subgraph via the maximum weight subgraph approximation method of (Vassilevska et al., 2006), which allows us to reduce the dynamic network size considering only most active nodes and its corresponding nodes. Let $\mathcal{G}_t = \{\mathcal{V}_t, \mathcal{E}_t, \tilde{W}_t\}$ denotes the reduced Ethereum blockchain network on day t and $X_t \in \mathbb{R}^{N_t \times 1}$ be the node feature matrix, we assume a solely node feature: the node degree. Each node in $\mathcal{V}_t$ is a

[1] Ethereum.org

[2] EtherScan.io

[3] End date: May 7, 2018

Table 2. Forecasting performance comparison of different approaches on PeMSD4 and PeMSD8 datasets. The baselines with † marks the results from rerunning the published source codes. Z-GCNETs uses the weight rank clique filtration.

Model	PeMSD4			PeMSD8		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HA	38.03	59.24	27.88%	34.86	52.04	24.07%
VAR (Hamilton, 2020)	24.54	38.61	17.24%	19.19	29.81	13.10%
FC-LSTM† (Sutskever et al., 2014)	26.77	40.65	18.23%	23.09	35.17	14.99%
GRU-ED (Cho et al., 2014)	23.68	39.27	16.44%	22.00	36.23	13.33%
DSANet (Huang et al., 2019)	22.79	35.77	16.03%	17.14	26.96	11.32%
DCRNN† (Li et al., 2018)	21.20	37.23	14.15%	16.83	26.35	10.90%
STGCN† (Yu et al., 2018)	21.16	35.69	13.83%	17.52	27.07	11.23%
GraphWaveNet† (Wu et al., 2019)	28.15	39.88	18.52%	20.30	30.82	13.84%
ASTGCN† (Guo et al., 2019)	22.81	34.33	16.60%	17.98	28.00	11.66%
MSTGCN† (Guo et al., 2019)	23.96	37.21	14.33%	19.00	29.15	12.38%
STSGCN† (Song et al., 2020)	21.23	33.69	13.90%	17.13	26.86	10.96%
AGCRN† (Bai et al., 2020)	19.83	32.30	12.97%	15.95	25.22	10.09%
LSGCN (Huang et al., 2020)	21.53	33.86	13.18%	17.73	26.76	11.20%
Z-GCNETs (ours)	**19.50**	**31.61**	**12.78%**	**15.76**	**25.11**	**10.01%**

buyer/seller and edges in $\mathcal{E}_t$ represent transactions in the network. To construct the similarity matrix $\tilde{W}_t$, the normalized number of transactions between node pairs (u, v) serves as the edge weight value $w_{t,uv} \in \tilde{W}_t$. In our experiments, we consider a set of 100 most actively trading nodes over the whole time period. That is, on each particular day t, both the node and edge sets vary. If a particular node v_i does not trade on day t, then v_i is entered as isolated node. As such, $N = 100$.

5.2. Experiment Settings

For multi-step time series forecasting, we evaluate the performances of Z-GCNETs on 4 time series datasets versus 13 state-of-the-art baselines (SOAs). Among them, Historical Average (HA) and Vector Auto-Regression (VAR) (Hamilton, 2020) are the statistical time series models. FC-LSTM (Sutskever et al., 2014) and GRU-ED (Cho et al., 2014) are RNN-based neural networks. DSANet (Huang et al., 2019) is the self-attention networks. DCRNN (Li et al., 2018), STGCN (Yu et al., 2018), GraphWaveNet (Wu et al., 2019), ASTGCN (Guo et al., 2019), MSTGCN (Guo et al., 2019), STSGCN (Song et al., 2020) are the spatial-temporal GCNs. AGCRN (Bai et al., 2020) and LSGCN (Huang et al., 2020) are the GRU-based GCNs. We conduct our experiments on NVIDIA GeForce RTX 3090 GPU card with 24GB memory. The PeMSD4 and PeMSD8 are split in chronological order with 60% for training sets, 20% for validation sets, and 20% for test sets. For PeMSD4 and PeMSD8, Z-GCNETs contains 2 layers, with each layer has 64 hidden units. We consider the window size $\tau = 12$ and horizon $h = 12$ for Z-GCNETs on both PeMSD4 and PeMSD8 datasets. Besides, the inputs of PeMSD4 and PeMSD8 are normalized by min-max normalization approach. We split Bytom and Decentraland with 80% for training sets and 20% for test sets. For token networks, Z-GCNETs contains 2 layers, where each layer has 16 hidden units. We use one week historical data to predict the next week's data, i.e., window size $\tau = 7$ and horizon $h = 7$ over Bytom and Decentralnad datasets. All reported results are based on the weight rank clique filtration (Stolz et al., 2017). More detailed description of the experimental settings can be found in Appendix A, while the analysis of sensitivity with respect to the choice of filtration is in Appendix B. The code is available at `https://github.com/Z-GCNETs/Z-GCNETs.git`.

Table 3. Computational costs for generation of zigzag persistence images (ZPI) and a single training epoch of Z-GCNETs.

Dataset	Window Size	Average Time Taken (sec)	
		ZPI	Z-GCNETs (epoch)
Decentraland	7	0.03	2.09
Bytom	7	0.03	2.06
PeMSD4	12	0.86	30.12
PeMSD8	12	0.65	36.76

Table 3 reports the average running time of ZPI generation and training time per epoch of our Z-GCNETs model on all datasets.

5.3. Comparison with the Baseline Methods

Table 2 shows the comparison of our proposed Z-GCNETs and SOAs for traffic flow forecasting tasks. We assess model performance with Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) on PeMSD4 and PeMSD8. From Table 2, we find that our proposed model Z-GCNETs consistently outperforms SOAs on PeMSD4 and PeMSD8. The improvement gain of Z-GCNETs over the next most accurate methods ranges from 0.44% to 2.06% in RMSE for PeMSD4 and PeMSD8. Table 4 summarizes the forecasting performance on Bytom and Decentraland in terms of RMSE.

We find that Z-GCNETs, based on the power filtration with transaction volume as edge weight, outperforms AGCRN by margins of 3.67% and 4.60%. (See the results in Appendix B for other filtrations. For all filtration types, Z-GCNETs yields better forecasting performance than all SOAs.). In contrast with SOAs, Z-GCNETs fully leverages the topological information by incorporating zigzag topological features via CNN on topological space. Given the dynamic nature of the considered data, our experiments show that establishing a connection between the time-indexed zigzag pairs can deliver substantial gains for learning and forecasting time-evolving objects.

Table 4. Forecasting results (MAPE) on Ethereum token networks. Z-GCNETs uses power filtration with transaction volume as edge weight.

Model	**Bytom**	**Decentraland**
FC-LSTM (Sutskever et al., 2014)	40.72%	33.46%
DCRNN (Li et al., 2018)	35.36%	27.69%
STGCN (Yu et al., 2018)	37.33%	28.22%
GraphWaveNet (Wu et al., 2019)	39.18%	37.67%
ASTGCN (Guo et al., 2019)	34.49%	27.43%
AGCRN (Bai et al., 2020)	34.46%	26.75%
LSGCN (Huang et al., 2020)	34.91%	28.37%
Z-GCNETs	**30.79%**	**22.15%**

5.4. Ablation Study

To better understand the importance of the different components in Z-GCNETs, we conduct ablation studies on PeMSD4 and PeMSD8 and the results are presented in Table 5. The results show that Z-GCNETs have better performance over Z-GCNETs without zigzag persistence representation learning (zigzag learning), spatial graph convolution ($\text{GCN}_{\text{Spatial}}$), or temporal graph convolution ($\text{GCN}_{\text{Temporal}}$). Specifically, we observe that when removing $\text{GCN}_{\text{Temporal}}$, the multi-step forecasting is affected significantly, i.e., Z-GCNETs outperforms Z-GCNETs without temporal graph convolution with relative gain 6.46% on RMSE for PeMSD4. Comparison results on PeMSD8, w/o zigzag learning and w/o show the necessity for encoding topological information and modeling spatial structural information in multi-step forecasting over spatial-temporal time series datasets. Additional results for the ablation study on Ethereum tokens are presented in Appendix C.

Table 5. Ablation study of the network architecture. [*] means the $\text{GCN}_{\text{Spatial}}$ is only applied to the most recent time point in the sliding window.

	Architecture	**MAE**	**RMSE**	**MAPE**
PeMSD4	**Z-GCNETs**	**19.50**	**31.61**	**12.78%**
	W/o Zigzag learning	19.65	31.94	13.01%
	W/o $\text{GCN}_{\text{Spatial}}{}^*$	19.86	31.96	13.19%
	W/o $\text{GCN}_{\text{Temporal}}$	20.76	33.18	13.60%
PeMSD8	**Z-GCNETs**	**15.76**	**25.11**	**10.01%**
	W/o Zigzag learning	17.16	27.06	10.77%
	W/o $\text{GCN}_{\text{Spatial}}{}^*$	16.92	26.86	10.33%
	W/o $\text{GCN}_{\text{Temporal}}$	16.66	26.44	10.39%

5.5. How does time-aware zigzag persistence help?

To track the importance of p-dimensional topological features in Z-GCNETs (i.e., 0-dimensional and 1-dimensional holes), we evaluate the performance of Z-GCNETs on two different aspects: (i) the sensitivity of Z-GCNETs to different dimensional topological features and (ii) the effects of threshold ν_* in constructed input networks along with zigzag persistence. Table 6 summarizes the results using different dimensional topological features and different thresholds on PeMSD4 and PeMSD8. Under the same scale parameter ν_*, we find that 1-dimensional topological features consistently outperform 0-dimensional terms on both datasets. Furthermore, the forecasting results on PeMSD4 are not significantly affected by varying ν_*. However, on PeMSD8, 1-dimensional topological features constructed under ν_* of 0.3 yield better results than 1-dimensional summaries constructed under $\nu_* = 0.5$.

5.6. Robustness Study

To assess robustness of Z-GCNETs under noisy conditions, we consider adding Gaussian noise into 30% of training sets. The added noise follows zero-mean i.i.d Gaussian density with fixed variance ς^2, i.e., $\mathcal{N}(0, \varsigma^2)$, where $\varsigma \in \{2, 4\}$. In Table 7, we report comparisons with two competitive baselines (AGCRN and LSGCN) on Decentraland and PeMSD4 using two different noise levels. Table 7 shows performance of Z-GCNETs and two SOAs under described noisy conditions. We find that performance of all methods slowly decays as variance of noise increases. Nevertheless, we notice that Z-GCNETs is still consistently more robust than

Table 6. Results of zigzag persistence on the dynamic network with different dimensional features and threshold values (ν_*).

	Zigzag module	PeMSD4		
		MAE	RMSE	MAPE
Z-GCNETs +	0-th $\mathrm{ZPI}_{\nu_*=0.3}$	19.73	32.04	12.93%
	1-st $\mathrm{ZPI}_{\nu_*=0.3}$	19.47	31.66	12.75%
	0-th $\mathrm{ZPI}_{\nu_*=0.5}$	19.78	32.20	12.98%
	1-st $\mathrm{ZPI}_{\nu_*=0.5}$	19.50	31.61	12.78%
	Zigzag module	**PeMSD8**		
		MAE	RMSE	MAPE
Z-GCNETs +	0-th $\mathrm{ZPI}_{\nu_*=0.3}$	17.14	27.24	10.66%
	1-st $\mathrm{ZPI}_{\nu_*=0.3}$	15.76	25.11	10.01%
	0-th $\mathrm{ZPI}_{\nu_*=0.5}$	17.22	27.41	10.77%
	1-st $\mathrm{ZPI}_{\nu_*=0.5}$	16.77	26.62	10.39%

SOAs on both Decentraland and PeMSD4.

Table 7. Robustness study on Decentraland and PeMSD4 (RMSE).

	Noise	AGCRN	LSGCN	Z-GCNETs (ours)
Decentraland	$\mathcal{N}(0,2)$	27.69	36.10	**24.12**
	$\mathcal{N}(0,4)$	28.12	36.79	**25.03**
PeMSD4	$\mathcal{N}(0,2)$	32.24	34.16	**31.95**
	$\mathcal{N}(0,4)$	32.67	34.75	**32.18**

6. Conclusion

Inspired by the recent call for developing time-aware deep learning mechanisms of the US Defense Advanced Research Projects Agency (DARPA), we have proposed a new time-aware zigzag topological layer (Z-GCNETs) for time-conditioned GCNs. Our idea is based on the concepts of zigzag persistence whose utility remains unexplored not only in conjunction with time-aware GCN but DL in general. The new Z-GCNETs layer allows us to track the salient time-aware topological characterizations of the data persisting over time. Our results on spatio-temporal graph structured data have indicated that integration of the new time-aware zigzag topological layer into GCNs results both in enhanced forecasting performance and robustness gains.

7. Acknowledgements

The project has been supported in part by the grants NSF DMS 1925346, NSF ECCS 2039701, the UTSystem-CONACYT ConTex program, and the grant N000142112226 from the Department of the Navy, Office of Naval Research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Office of Naval Research. The authors are grateful to Baris Coskunuzer for insightful discussions.

References

Adams, H. and Carlsson, G. Evasion paths in mobile sensor networks. *The International Journal of Robotics Research*, 34(1):90–104, 2015.

Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F., and Ziegelmeier, L. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18, 2017.

Adams, H., Bush, J., Carr, B., Kassab, L., and Mirth, J. A torus model for optical flow. *Pattern Recognition Letters*, 129:304–310, 2020.

Akcora, C. G., Kantarcioglu, M., and Gel, Y. R. Blockchain networks: Data structures of bitcoin, monero, zcash, ethereum, ripple and iota. *arXiv:2103.08712*, 2021.

Bai, L., Yao, L., Li, C., Wang, X., and Wang, C. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in Neural Information Processing Systems*, 33, 2020.

Bauer, U. Ripser: efficient computation of vietoris-rips persistence barcodes. *arXiv:1908.02518*, 2019.

Carlsson, G. Persistent homology and applied homotopy theory. *Handbook of Homotopy Theory*, 2019.

Carlsson, G. and Gabrielsson, R. B. Topological approaches to deep learning. In *Topological Data Analysis*, pp. 119–146. Springer, 2020.

Carlsson, G. and Silva, V. Zigzag persistence. *Found. Comput. Math.*, 10(4):367–405, August 2010. ISSN 1615-3375.

Carrière, M., Chazal, F., Ike, Y., Lacombe, T., Royer, M., and Umeda, Y. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In *AISTATS*, pp. 2786–2796, 2020.

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

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Chowdhury, S., Dai, B., and Mémoli, F. The importance of forgetting: Limiting memory improves recovery of topological characteristics from neural data. *PloS one*, 13 (9):e0202561, 2018.

Corcoran, P. and Jones, C. B. Modelling topological features of swarm behaviour in space and time with persistence landscapes. *IEEE Access*, 5:18534–18544, 2017.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, volume 29, pp. 3844–3852, 2016.

di Angelo, M. and Salzer, G. Tokens, types, and standards: Identification and utilization in ethereum. In *2020 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*, pp. 1–10, 2020. doi: 10.1109/DAPPS49028.2020.00001.

Gamble, J., Chintakunta, H., and Krim, H. Coordinate-free quantification of coverage in dynamic sensor networks. *Signal Processing*, 114:1–18, 2015.

Gao, S., Huang, Y., Zhang, S., Han, J., Wang, G., Zhang, M., and Lin, Q. Short-term runoff prediction with gru and lstm networks without requiring time step optimization during sample generation. *Journal of Hydrology*, 589:125188, 2020. ISSN 0022-1694. doi: https://doi.org/10.1016/j.jhydrol.2020.125188. URL https://www.sciencedirect.com/science/article/pii/S002216942030648X.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232, 2017. doi: 10.1109/TNNLS.2016.2582924.

Guo, S., Lin, Y., Feng, N., Song, C., and Wan, H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 922–929, 2019.

Hamilton, J. D. *Time series analysis*. Princeton university press, 2020.

Hofer, C. D., Kwitt, R., and Niethammer, M. Learning representations of persistence barcodes. *JMLR*, 20(126): 1–45, 2019.

Huang, R., Huang, C., Liu, Y., Dai, G., and Kong, W. Lsgcn: Long short-term traffic prediction with graph convolutional networks. In Bessiere, C. (ed.), *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 2355–2361, 2020.

Huang, S., Wang, D., Wu, X., and Tang, A. Dsanet: Dual self-attention network for multivariate time series forecasting. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 2129–2132, 2019.

Kim, W., Mémoli, F., and Smith, Z. Analysis of dynamic graphs and dynamic metric spaces via zigzag persistence. In *Topological Data Analysis*, pp. 371–389. Springer, 2020.

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

Li, Y., Yu, R., Shahabi, C., and Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *International Conference on Learning Representations*, 2018.

Li, Y., Islambekov, U., Akcora, C., Smirnova, E., Gel, Y. R., and Kantarcioglu, M. Dissecting ethereum blockchain analytics: What we learn from topology and geometry of the ethereum graph? In *SDM*, pp. 523–531, 2020.

Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T., and Leiserson, C. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5363–5370, 2020.

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

Segovia Dominguez, I., Lee, H., Chen, Y., Garay, M., Gorski, K., and Gel, Y. Does air quality really impact covid-19 clinical severity: Coupling nasa satellite datasets with geometric deep learning. *KDD*, 2021.

Segovia-Dominguez, I., Zhen, Z., Wagh, R., Lee, H., and Gel, Y. R. Tlife-lstm: Forecasting future covid-19 progression with topological signatures of atmospheric conditions. In *PAKDD (1)*, pp. 201–212, 2021.

Shin, S. and Kim, W. Skeleton-based dynamic hand gesture recognition using a part-based gru-rnn for gesture-based interface. *IEEE Access*, 8:50236–50243, 2020. doi: 10.1109/ACCESS.2020.2980128.

Song, C., Lin, Y., Guo, S., and Wan, H. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 914–921, 2020.

Stolz, B. J., Harrington, H. A., and Porter, M. A. Persistent homology of time-dependent functional networks constructed from coupled time series. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(4):047410, 2017.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27:3104–3112, 2014.

Tausz, A. and Carlsson, G. Applications of zigzag persistence to topological data analysis. *arXiv:1108.3545*, 2011.

Tymochko, S., Munch, E., and Khasawneh, F. A. Hopf bifurcation analysis using zigzag persistence. *Algorithms*, 13(11):278, 2020.

Vassilevska, V., Williams, R., and Yuster, R. Finding the smallest h-subgraph in real weighted graphs and related problems. In *Automata, Languages and Programming*, pp. 262–273. Springer Berlin Heidelberg, 2006.

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

Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Bellei, C., Robinson, T., and Leiserson, C. E. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591*, 2019.

Wu, Z., Pan, S., Long, G., Jiang, J., and Zhang, C. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 1907–1913, 2019.

Xian, L., Adams, H., Topaz, C. M., and Ziegelmeier, L. Capturing dynamics of time-varying data via topology. *arXiv:2010.05780*, 2020.

Yan, S., Xiong, Y., and Lin, D. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Ye, J., and Li, Z. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Yu, B., Yin, H., and Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 3634–3640, 2018.

Yu, Y., Si, X., Hu, C., and Zhang, J. A review of recurrent neural networks: Lstm cells and network architectures. *Neural Comput.*, 31(7):1235–1270, 2019.