Time-Conditioned Dances with Simplicial Complexes: Zigzag Filtration Curve based Supra-Hodge Convolution Networks for Time-series Forecasting

Yuzhou Chen
Department of Computer and Information Sciences
Temple University
yuzhou.chen@temple.edu

Yulia R. Gel
Department of Mathematical Sciences
University of Texas at Dallas
National Science Foundation
ygl@utdallas.edu

H. Vincent Poor
Department of Electrical and Computer Engineering
Princeton University
poor@princeton.edu

Abstract

Graph neural networks (GNNs) offer a new powerful alternative for multivariate time series forecasting, demonstrating remarkable success in a variety of spatio-temporal applications, from urban flow monitoring systems to health care informatics to financial analytics. Yet, such GNN models pre-dominantly capture only lower order interactions, that is, pairwise relations among nodes, and also largely ignore intrinsic time-conditioned information on the underlying topology of multivariate time series. To address these limitations, we propose a new time-aware GNN architecture which amplifies the power of the recently emerged simplicial neural networks with a time-conditioned topological knowledge representation in a form of zigzag persistence. That is, our new approach, Zigzag Filtration Curve based Supra-Hodge Convolution Networks (ZFC-SHCN) is built upon the two main components: (i) a new highly computationally efficient zigzag persistence curve which allows us to systematically encode time-conditioned topological information, and (ii) a new temporal multiplex graph representation module for learning higher-order network interactions. We discuss theoretical properties of the proposed time-conditioned topological knowledge representation and extensively validate the new time-aware ZFC-SHCN model in conjunction with time series forecasting on a broad range of synthetic and real-world datasets: traffic flows, COVID-19 biosurveillance, Ethereum blockchain, surface air temperature, wind energy, and vector autoregressions. Our experiments demonstrate that the ZFC-SHCN achieves the state-of-the-art performance with lower requirements on computational costs.

1 Introduction

Over the last few years, graph neural networks (GNNs) have emerged as a new powerful alternative to traditional statistical and machine learning models in conjunction with univariate and multivariate time series forecasting tasks [31, 5, 44, 44, 32]. Such successful applications of GNNs range from urban traffic analytics to forecasting COVID-19 hospitalizations to electrocardiogram monitoring [4, 40, 59, 11, 21]. However, most GNNs remain inherently static and do not explicitly incorporate the inherent time characteristics of the encoded knowledge [62, 45]. In turn, limitations in capturing the

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

time dimension in the knowledge representation and learning mechanisms for time-evolving data results in GNNs becoming less relevant over time and, hence, requiring frequent updates.

Furthermore, GNNs tend to pre-dominantly focus only on information propagation among nodes and also be limited in their ability to describe polyadic relationships among multiple substructures of multivariate time series or multi-node interactions in dynamics graphs. However, as recently shown by [7, 22], such higher-order interactions might be the key toward better understanding of the underlying mechanisms of many real-world graph-structured phenomena. This challenge on polyadic graph interactions has been recently addressed by [26, 9, 8] who propose to model higher order substructures as simplices. Then, by borrowing the concepts of the Hodge theory, these approaches allow for generalization of the ideas of the combinatorial graph Laplacian which describes a diffusion from node to node via edges to a case of diffusion over simplices. Such Hodge Laplacian construction allows for extending the notion of convolution operation to simplicial convolution, and the resulting simplicial neural networks (SNNs) are arguably one of the frontlines in graph learning today. However, these ideas have never been yet applied in conjunction with knowledge representation and learning of time-evolving objects.

Our goal here is to bridge the emerging concept of time-aware learning with the recent notions of simplicial convolution, with a particular focus on explicitly integrating the core time-conditioned topological characteristics. In particular, we amplify the power of SNNs with a time-conditioned topological knowledge representation in a form of zigzag persistence for time-indexed data and, more specifically, its new highly computationally efficient summary, Zigzag Filtration Curve. As a result, our new approach, Zigzag Filtration Curve based Supra-Hodge Convolution Networks (ZFC-SHCN) enables us to systematically learn the most intrinsic time-conditioned information both on the underlying topology of the time-evolving data and higher-order interactions among various substructures.

Significance of our contributions can be summarized as follows:

- ZFC-SHCN is the first approach bringing the concepts of simplicial convolution and SNNs to time-aware learning.
- We propose a new highly computationally efficient summary of persistence for time-indexed data, Zigzag Filtration Curve, and derive its theoretical stability guarantees.
- We validate the utility of ZFC-SHCN in conjunction with forecasting multivariate time series from diverse application domains such as traffic networks, COVID-19 biosurveillance, surface air temperature, token prices on Ethereum blockchain, wind energy, and vector autoregressions. Our findings indicate that ZFC-SHCN delivers the state-of-the-art forecasting performance, with a significant margin and demonstrates higher computational efficiency.

2 Related Work

Time-series Forecasting and Spatio-temporal Graph Convolutional Networks Time-series forecasting is one of the core subfields in statistical sciences [16, 10]. Most recently, there have appeared a number of unconventional machine learning approaches to time-series forecasting. In particular, graph convolutional network (GCN)-based models for spatio-temporal network data have emerged as a promising forecasting tool. For instance, DCRNN [45] introduces spectral graph convolution into spatio-temporal network data prediction, which can capture spatio-temporal dependencies. STGCN [62] uses convolutional neural networks (CNNs) to model temporal correlations. Moreover, to infer hidden inter-dependencies between different traffic variables, [60, 4, 11] conduct a convolution operation in spatial dimension through adaptive adjacency matrices. Recent Z-GCNETs [21] develops a zigzag topological layer equipped with a zigzag persistence image into a GCN framework to model temporal correlations. Another promising recent direction for time series forecasting beyond GCN is a fractional-order dynamical model proposed by [31]. This approach offers an alternating scheme to determine the best estimate of the model parameters and unknown stimuli. In turn, [32] proposes a Padé approximation based exponential neural operator (Padé Exp), aiming to improve time-series forecasting with exponential operators in neural operator learning schemes. However, all of the above methods only focus on node-level representations. In contrast, in this paper, we focus on both higher-order structure representation and topological information learning.

Topological Data Analysis for Graph Learning Persistent homology [27, 65] is a suite of tools within topological data analysis (TDA) that provides a way for measuring topological features of shapes and functions. The extracted topological features have been recently shown to provide invaluable insights into hidden mechanisms behind the organization and functionality of graph structured data. In particular, topological features have been actively used for node classification [64, 18], link prediction [61], and graph classification [35, 36, 15, 34]. For instance, [35] is one of the first approaches to integrate topological features into neural networks for graph classification, while [15] proposes a versatile framework for learning multiple vectorizations of persistent diagrams on graphs. In turn, [64, 18, 61] apply topological features to GNNs to understand and improve the message passing between nodes. Finally, [37] proposes a topological graph layer with learnable filtration functions for graph and node classification tasks, while [14] advances the ideas of multipersistence to graph learning.

Zigzag Persistent Homology Despite its promise, regular persistent homology does not explicitly model the geometric and topological information from a sequence of topological spaces. To address this limitation, a generalization of ordinary persistence, i.e., zigzag persistent homology, based on the theory of quiver representation, has been proposed by [13]. Zigzag persistence allows us to systematically describe how the homology changes over a sequence of spaces. Despite its high potential, especially in conjunction with analysis of time-evolving data, zigzag persistence still remains largely a theoretical concept, and its applications are yet scarce. The recent results for time-dependent data studies include, for example, zigzag-based clustering [41], bifurcation analysis of dynamic systems [57], and time series forecasting [21]. The memory and computational efficiency of zigzag persistence is one of the daunting challenges. Inspired by [47], we propose a novel highly computationally efficient representation of zigzag persistence for learning time-evolving data, that is, zigzag filtration curve.

Simplicial Neural Networks Modeling higher-order interactions on graphs is an emerging direction in graph representation learning. While the role of higher-order structures for graph learning has been documented for a number of years [1, 38] and involves such diverse applications as graph signal processing in image recognition [25], dynamics of disease transmission and biological networks, integration of higher-order graph substructures into deep learning on graphs has emerged only in 2020. As shown by [7, 53], higher-order network structures can be leveraged to boost graph learning performance. Indeed, several recent approaches [26, 52, 9, 19] propose to leverage simplicial information to perform neural networks on graphs. However, neither of these Simplicial Neural Networks (SNNs) are integrated with a topology-based graph convolution layer allowing us to learn both time-aware persistent topological features and simplicial geometry of graphs. In this paper, we propose ZFC-SHCN to address this limitation.

3 Time-Aware Topological Learning with Zigzag Curves

Spatio-temporal Graph Construction A spatio-temporal graph is a collection of snapshots at different time steps, denoted by $\boldsymbol{\mathcal{G}} = \{\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_{\mathcal{T}}\}$, where $\mathcal{T}$ is the maximum timestamp. Here $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \boldsymbol{\mathcal{A}}_t, \boldsymbol{X}_t)$ is the graph observed at time step $t \in [1, \mathcal{T}]$, where $\mathcal{V}_t$ is a finite set of $|\mathcal{V}| = N$ nodes, $\mathcal{E}_t$ is a set of edges, $\boldsymbol{\mathcal{A}}_t \in \mathbb{R}^{N \times N}$ is the adjacency matrix, and $\boldsymbol{X}_t \in \mathbb{R}^{N \times d}$ is the node feature matrix. Specifically, each row of $\boldsymbol{X}_t$ is a d-dimensional feature vector of the corresponding node. For sake of notations, wherever applicable below, we omit the subscript t and denote graph $\mathcal{G}_t$ at time t as $\mathcal{G}$.

Background on Ordinary Persistence Tools of ordinary persistence, or persistent homology (PH), allow us to study salient data shape patterns along various dimensions. By shape here we broadly understand data properties that are invariant under continuous transformations, that is, transformations that do not alter "holes" in the data, for example, bending, twisting, and stretching. The key idea is to choose some suitable scale parameter ν and then to study a graph $\mathcal{G}$ not as a single object but as a nested sequence of graphs, or *graph filtration* $\mathcal{G}_1 \subseteq \ldots \subseteq \mathcal{G}_n = \mathcal{G}$, which is induced by monotonic changes of scale ν. For example, if $\mathcal{G}$ is an edge-weighted graph $(\mathcal{V}, \mathcal{E}, w)$ with weight function $w : \mathcal{E} \mapsto \mathbb{R}$, then for each ν_j, $j = 1, \ldots, n$, we set $\mathcal{G}_{\leq \nu_j} = (\mathcal{V}, \mathcal{E}, w^{-1}(-\infty, \nu_j])$, yielding the induced edge-weighted filtration. We can also consider only induced subgraphs of $\mathcal{G}$ with maximal degree of ν_j for each $j = 1, \ldots, n$, resulting in the degree sublevel set filtration. (For more discussion on graph filtrations see [34].) Armed with this construction, we can track which shape patterns, for example, independent components, loops, and voids, emerge as the scale ν varies. To make the

process of pattern counting more systematic and efficient, we build an abstract simplicial complex $\mathscr{K}(\mathcal{G}_j)$ on each $\mathcal{G}_j$. We also record complex indices j_b (birth) and j_d (death) at which we first or last observe each shape feature. Topological features with longer lifespans are said to persist and are likelier to yield important information on the structural organization of $\mathcal{G}$.

Learning Shapes of Time-Conditioned Data with Zigzag Persistence This construction enables us to extract the key topological descriptors from a single graph $\mathcal{G}$. However, in our case, we observe not a single graph but a sequence of time-evolving graphs $\{\mathcal{G}^1, \ldots, \mathcal{G}^{\mathcal{T}}\}$. How can we track shape signatures which are not just individualistic for each time stamp but characterize intrinsic properties of the observed object over time? One approach to how we can bring PH tools to analysis of time-conditioned objects is zigzag persistence. Based on the theory of quiver representations, zigzag persistence generalizes ordinary persistence to track characteristics of graphs (or other topological spaces) with inclusions going in different directions [13, 12]. In particular, given a time-indexed sequence of graphs $\{\mathcal{G}^1, \ldots, \mathcal{G}^{\mathcal{T}}\}$, we first form a set of graph inclusions over time

$$\begin{array}{ccccccccc} & \mathcal{G}_1 \cup \mathcal{G}_2 & & \mathcal{G}_2 \cup \mathcal{G}_3 & & \mathcal{G}_3 \cup \mathcal{G}_4 & & \ldots \\ \nearrow & & \nwarrow & & \nearrow & & \nwarrow & & \nearrow \\ \mathcal{G}_1 & & \mathcal{G}_2 & & \mathcal{G}_3 & & \mathcal{G}_4 & & \end{array}$$

and then assess the compatibility of persistent topological features across unions of graphs. That is, we record indices at which topological features (dis)appear, for some given scale ν_*. If for a given ν_* topological feature ρ (i.e., p-dimensional hole, $0 \leq p \leq \mathcal{K}$, where $\mathcal{K}$ is the dimension of the simplicial complex $\mathbb{K}(\mathcal{G})$) is first recorded in $\mathbb{K}(\mathcal{G}^j)$, we say that the feature's birth is j, and if ρ first appears in $\mathbb{K}(\mathcal{G}^j \cup \mathcal{G}^{j+1})$, we record its birth as $j + 1/2$. In turn, if ρ is last seen in $\mathbb{K}(\mathcal{G}^j)$, we record its death as j, while if it is last seen in $\mathbb{K}(\mathcal{G}^j \cup \mathcal{G}^{j+1})$, we say that its death is at $j + 1/2$. Let $\mathfrak{J}$ be the set of all observed topological features for a given ν_*. Collecting then births and deaths over $\mathfrak{J}$, we summarize all extracted information as a multiset $\mathcal{D}_{\nu_*} = \{(b_\rho, d_\rho) \in \mathbb{R}^2 | b_\rho < d_\rho, \rho \in \mathfrak{J}\}$, called a zigzag persistent diagram (ZPD) (where b_ρ and d_ρ are the birth and death of the topological feature ρ respectively).

This makes zigzag persistence particularly attractive for the analysis of dynamic objects which are naturally indexed by time. However, the idea of zigzag persistence is applicable far beyond learning time-evolving objects. Nevertheless, zigzag persistence still remains largely a theoretical concept, with yet only a handful of applications, and one of the roadblocks hindering a broader proliferation of zigzag-based methods in practice is their computational costs. Here we take a step toward bringing a more computationally efficient summary of zigzag persistence to real-world applications.

Time-Aware Zigzag Filtration Curves Consider a sequence of time intervals associated with a zigzag filtration over a time period $[t_1, t_{\mathcal{N}}]$

$$\left(t_1, t_1 + \frac{1}{2}\right), \left(t_1 + \frac{1}{2}, t_2\right), \left(t_2, t_2 + \frac{1}{2}\right), \ldots, \left(t_{\mathcal{N}-1} + \frac{1}{2}, t_{\mathcal{N}}\right).$$

Let DgmZZ_{ν_*} be the resulting ZPD for a given ν_* and $\mathcal{M}$ be the number of off-diagonal topological features in ZPD, i.e., DgmZZ_{ν_*}. Inspired by the recent results on stabilized Betti sequences by [39] and filtration curves by [47] for ordinary persistence, we propose a new simple and computationally efficient summary of zigzag persistence, called a *Zigzag Filtration Curve*.

Definition 3.1 (Zigzag Filtration Curve (ZFC))**.** The zigzag filtration curve evaluated at $\Delta t_i^- = (t_{i-1} + \frac{1}{2}, t_i)$, $i = \{1, 2, \ldots, \mathcal{N}\}$, for a given ν_*, is defined as

$$\text{ZFC}^p_{\nu_*}(\Delta t_i^-) = \sum_{j=1}^{\mathcal{M}} \xi_i(t_{b_j}, t_{d_j})\omega_i,$$

where $(t_{b_j}, t_{d_j}) \in \mathbb{R}^2$ is a vector containing the birth and death of the j-th off-diagonal p-dimensional topological feature in DgmZZ_{ν_*} (as such, $t_{b_j} < t_{d_j}$), $j = \{1, 2, \ldots, \mathcal{M}\}$, $0 \leq p \leq \mathcal{K}$; $\xi_i : \mathbb{R}^2 \mapsto \mathbb{R}$ is some suitable Lipschitz continuous function with Lipschitz constant L_i, for example, a Gaussian density; and $\omega_i > 0$, $i = \{1, 2, \ldots, \mathcal{N}\}$ are weights such that $\sum_i \omega_i = 1$. Zigzag filtration curve at $\Delta t_i^+ = (t_i, t_i + \frac{1}{2})$ is defined analogously. (For the sake of notational simplicity, wherever applicable in the further exposition we suppress the index p in ZFC.)

Motivated by [39], here as the Lipschitz continuous function ξ_i for intervals Δt_i^-, we use a Gaussian density f with mean $(t_{i-1} + 1/2, t_i)$, while for intervals Δt_i^+, we set the mean of f to $(t_i, t_i + 1/2)$,

$i = 1, 2, \ldots \mathcal{N}$. For both Δt_i^- and Δt_i^+, we choose the 2×2-variance-covariance matrix Σ to be the identity matrix. (See Appendix C.7 for more discussion on sensitivity analysis.) Another suitable choice of ξ is the arctan function.

As we show below, the proposed ZPC also enjoys important theoretical stability guarantees in terms of Wasserstein-1 distance.

Proposition 3.2 (Stability of Zigzag Filtration Curve)**.** *Let $DgmZZ_{\nu_*}$ be a zigzag persistence diagram and $DgmZZ'_{c_*}$ be its perturbed copy such that $\mathcal{W}_1\big(DgmZZ_{\nu_*}, DgmZZ'_{\nu_*}\big) < \epsilon$, where $\mathcal{W}_1$ is Wasserstein-1 distance. Then, ZFC is stable with respect to Wasserstein-1 distance.*

In practice topological features of various dimensions p, $p = 0, 1, \ldots, \mathcal{K}$, may play different roles in the learning task performance, and these roles are not known a-priori. Hence, to harness time-conditioned information encoded in ZFC corresponding to different dimensions p, we propose Multi-Zigzag Filtration Curves (M-ZFCs) M-ZFCs$_{\nu_*} \in \mathbb{R}^{K \times \frac{\mathcal{N}-1}{2}}$ by stacking ZFC^0, ZFC^1, $\ldots$, ZFC$^{\mathcal{K}}$. Figure 1 in Appendix B shows the both 0- and 1-dimensional ZFCs obtained from the proposed ZFC. In the following section, we demonstrate how ZFC can be integrated into neural network architectures for graph learning tasks.

4 Zigzag Filtration Curve Based Supra-Hodge Convolution Networks

Given a graph $\mathcal{G}$ and its historical ω step graph signals $\boldsymbol{X}^\omega = \{\boldsymbol{X}_{t-\omega+1}, \ldots, \boldsymbol{X}_t\} \in \mathbb{R}^{\omega \times N \times F}$ (F is the node feature dimensionality), the time-series forecasting problem is to learn a mapping function f that maps the historical data $\{\boldsymbol{X}_{t-\omega+1}, \ldots, \boldsymbol{X}_t\}$ into the next h step data $\{\boldsymbol{X}_{t+1}, \ldots, \boldsymbol{X}_{t+h}\}$. The mapping relation is represented as follows: $\{\boldsymbol{X}_{t-\omega+1}, \ldots, \boldsymbol{X}_t\} \xrightarrow{f} \{\boldsymbol{X}_{t+1}, \ldots, \boldsymbol{X}_{t+h}\}$.

4.1 Graph convolution in the spatial dimension

Given the node embedding dictionary $\boldsymbol{W}^\phi = (w_1^\phi, w_2^\phi, \ldots, w_N^\phi) \in \mathbb{R}^{N \times d_c}$ (where $x_u^\phi \in \mathbb{R}^{d_c}$ and d_c is the dimension of node embedding), we aim to seek a non-negative function $\boldsymbol{S}_{u,v} = \mathscr{G}(w_u^\phi, w_v^\phi)$ which represents the pairwise similarity between any two nodes u and v. Concretely, the multiplication between $\boldsymbol{W}^\phi$ and $(\boldsymbol{W}^\phi)^\top$ can (i) give a sum pooling of second-order features from the outer product of all the embedding vector pairs (w_u^ϕ, w_v^ϕ) and (ii) infer the hidden spatial dependencies of nodes

$$\boldsymbol{S}_{uv} = \mathscr{G}(w_u^\phi, w_v^\phi) = \frac{\exp\left(\text{ReLU}(w_u^\phi (w_v^\phi)^\top\right)}{\sum_{u=1}^N \exp\left(\text{ReLU}(w_u^\phi (w_v^\phi)^\top\right)},$$

where $\text{ReLU}(\cdot) = \max(0, \cdot)$ is a nonlinear activation function, which is used to eliminate weak connections proactively, and the role of the softmax function is applied to normalize the learned graph $\boldsymbol{S}$. Inspired by the recent advancements in random walk-based graph embedding learning [50, 29], we make a graph convolution in spatial dimension, feeding a power series of the learned graph $\boldsymbol{S}$ with varying random walk steps $\{1, 2, \cdots, r\}$ ($r \in \mathbb{Z}_+$), as follows:

$$\boldsymbol{H}_{t,\text{GC}}^{(\ell+1)} = \sigma(\text{Stack}(\boldsymbol{I}, \boldsymbol{S}, \cdots, \boldsymbol{S}^r)\boldsymbol{H}_{t,\text{GC}}^{(\ell)}\boldsymbol{\Theta}_{\text{GC}}^{(\ell)}), \tag{1}$$

where $\sigma(\cdot)$ stands for a nonlinear activation function, Stack$(\cdot)$ is the function which stacks r powered learned graphs, $\boldsymbol{H}_{t,\text{GC}}^{(\ell)}$ and $\boldsymbol{H}_{t,\text{GC}}^{(\ell+1)}$ are the input and output activations for layer ℓ (where $\boldsymbol{H}_{t,\text{GC}}^{(0)} = \boldsymbol{X}_t \in \mathbb{R}^{N \times F}$), and $\boldsymbol{\Theta}_{\text{GC}}^{(\ell)} \in \mathbb{R}^{d_\ell^{\text{GC}} \times d_{\ell+1}^{\text{GC}}}$ is the ℓ-th layer's trainable weights. Next, we introduce representation learning of the higher-order graph (sub)structures using the supra-Hodge Laplacian which allows us to systematically leverage the underlying topological information.

4.2 Supra-Hodge convolution in temporal dimension

Time-evolving data such as multivariate time series, spatio-temporal processes, and dynamic networks, often exhibit a highly complex dependency among its substructures that goes far beyond what can be described by dyadic (or pairwise) interactions among nodes. Instead, such higher-order polyadic interactions can be systematically addressed using the Hodge theory. In particular, the discrete Hodge theory allows us to generalize the notion of a standard combinatorial graph Laplacian which describes diffusion on graph $\mathcal{G}$ from node to node via edges to diffusion over higher-order substructures of

$\mathcal{G}$ [46, 7]. In turn, higher-order substructures can be modeled as k-simplices of $\mathcal{G}$. (See Appendix B for background information on Hodge Laplacians.) Convolutional architectures on simplicial complexes based on the associated concepts of the Hodge theory have emerged as a recent direction in graph neural networks but have not yet been applied to learning time-evolving data. Our goal here is to introduce the notion of simplicial convolution and the ideas of Hodge-Laplacians to time-aware learning.

In particular, to capture time-conditioned higher-order interactions on $\mathcal{G}$ and to describe diffusion of information over simplices along the temporal dimension, we build a supra-Hodge convolution operation, based on the multiplex network representation learning. (In the following for simplicity, notation without sub/superscript k stands for node-level quantities and in our experiments we always consider $k \in \mathbb{Z}_+$). First, given the historical spatio-temporal network series $\boldsymbol{\mathcal{G}}_{t-\omega+1:t} = \{\mathcal{G}_{t-\omega+1}, \mathcal{G}_{t-\omega+2}, \ldots, \mathcal{G}_t\}$, we consider a directed connected node-aligned multiplex network, which is made up of ω layers with N nodes on each layer. That is, the adjacency matrix $\boldsymbol{\mathcal{A}}^{\alpha} = \{a_{uv}^{\alpha}\}_{N\times N}$ (where $\alpha \in \{t-\omega+1, \ldots, t\}$) defines the intra-connection between nodes u and v in layer α and a distance matrix $\boldsymbol{\mathcal{D}}^{\alpha\beta} = \{d_{uu}^{\alpha\beta}\}_{N\times N}$ quantifies the transition probability of moving from node u of layer α to node u of layer β. (Here $\beta > \alpha$, since we consider information diffusion procedures only along the temporal dimension). Next, based on the discrete Hodge theory, we propose a new Hodge k-Laplacian for multiplex graphs, called the *supra-Hodge k-Laplacian* $\boldsymbol{\mathcal{L}}_k^{Sup} \in \mathbb{R}^{\phi_k\omega\times\phi_k\omega}$

$$\boldsymbol{\mathcal{L}}_k^{\text{Sup}} = \begin{pmatrix} (\boldsymbol{\mathcal{L}}_k^{11})^r & \boldsymbol{D}_{k+1}^{12} & \cdots & \boldsymbol{D}_{k+1}^{1\omega} \\ \mathbf{0} & (\boldsymbol{\mathcal{L}}_k^{22})^r & \cdots & \boldsymbol{D}_{k+1}^{2\omega} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & (\boldsymbol{\mathcal{L}}_k^{\omega\omega})^r \end{pmatrix}, \tag{2}$$

where $\boldsymbol{\mathcal{L}}_k^{\alpha\alpha}$ is the Hodge k-Laplacian in layer α, $\boldsymbol{D}_{k+1}$ is the diagonal matrix of degrees of each k-simplex, i.e., $\boldsymbol{D}_{k+1} = \max\left(\text{diag}(|\boldsymbol{B}_{k+1}|\mathbf{1}, \boldsymbol{I})\right)$ and $\boldsymbol{B}_{k+1}$ is the k-simplex-to-$(k+1)$-simplex incidence matrix, and the r-th power of $\boldsymbol{\mathcal{L}}_k^{\alpha\alpha}$ represents r-step random walk on the Hodge k-Laplacian of layer α which will allow every k-simplex to accumulate information from its neighbors. Hence, when $k = 1$, we can infer the spatial dependencies between each pair of edges and capture meaningful edge information in both spatial and temporal dimensions – through the lens of the supra-Hodge 1-Laplacian. For instance, in molecule networks, each node represents an atom and each edge is a bond connecting two atoms; the bond (i.e., edge) features include bond type, ring status, and molecular charge which are closely related to atom (i.e., node) features (such as atomic total and partial charges).

Since the goal of the forecasting task is to predict node (i.e., 0-simplex) attribute(s) in the next few time steps, we propose a novel diffusion supra-Hodge convolution on the sliding window $\boldsymbol{\mathcal{G}}_{t-\omega+1:t}$. We then update nodes' representations by transforming the multiplex k-simplex embedding to nodes via incidence matrices

$$\boldsymbol{H}_{t,k,\text{SH}}^{(\ell+1)} = \sigma(\boldsymbol{\mathcal{L}}_k^{\text{Sup}} \boldsymbol{H}_{t,k,\text{SH}}^{(\ell)} \boldsymbol{\Theta}_{k,\text{SH}}^{(\ell)}), \tag{3}$$

$$\boldsymbol{H}_{t,\text{SH}}^{(\ell+1)} = (\boldsymbol{B}_1^{\top} \cdots \boldsymbol{B}_k^{\top}) \boldsymbol{H}_{t,k,\text{SH}}^{(\ell+1)}, \tag{4}$$

where (i) in Equation 3: $\boldsymbol{\Theta}_{k,\text{SH}}^{(\ell)} \in \mathbb{R}^{d_{k;\ell}^{\text{SH}} \times d_{k;\ell+1}^{\text{SH}}}$ is a learnable filter matrix for layer ℓ (here $d_{k;\ell}^{\text{SH}}$ and $d_{k;\ell+1}^{\text{SH}}$ are the intermediate and output dimensions to the ℓ-th layer), $\boldsymbol{H}_{t,k,\text{SH}}^{(\ell)}$ and $\boldsymbol{H}_{t,k,\text{SH}}^{(\ell+1)}$ are the input and output activations for layer ℓ (where $\boldsymbol{H}_{t,k,\text{SH}}^{(0)} = \bar{\boldsymbol{X}}_{k;t-\omega+1:t} \in \mathbb{R}^{\phi_k\omega\times d_k^{\text{in}}}$ and the historical k-simplex features of the spatio-temporal networks $\boldsymbol{X}_{k;t-\omega+1:t} = \{\boldsymbol{X}_{k;t-\omega+1}, \boldsymbol{X}_{k;t-\omega+2}, \ldots, \boldsymbol{X}_{k;t}\} \in \mathbb{R}^{\phi_k\times\omega\times d_k^{\text{in}}}$ is reshaped as a matrix $\bar{\boldsymbol{X}}_{k;t-\omega+1:t}$ with shape $\phi_k\omega \times d_k^{\text{in}}$) and (ii) in Equation 4: we transform the k-simplex embedding $\boldsymbol{H}_{t,k,\text{SH}}^{(\ell+1)}$ to node embedding $\boldsymbol{H}_{t,\text{SH}}^{(\ell+1)} \in \mathbb{R}^{N\times d_{k;\ell+1}^{\text{SH}}}$ through incidence matrices.

4.3 ZFC convolution: a bridge between spatial and time dimensions

Armed with the representation learning of graph (sub)structures at each timestamp, we now discuss the ZFC convolution which allows us to preserve and propagate both spatial and time-aware topological information simultaneously. The intuition behind ZFC convolution is that it learns a strong connection between two dimensions via two 1D convolution layers, i.e., time-wise and node-wise. ZFC

convolution consists of three key components: (i) a linear embedding on M-ZFCs, which can learn the importance of time-aware topological features for each node to form a time-dimension-specific node embedding; (ii) a time-wise 1D convolution layer, where it gathers time-aware topological features from the entire space into a compact set; (iii) a node-wise 1D convolution layer, which can capture relations between different nodes. The resulted ZFC convolution operation over a M-ZFCs$^{\omega}$ is defined as

$$\boldsymbol{H}_{t,\text{M-ZFC}} = \mathscr{F}_{\theta}(\mathscr{F}_{\psi}(\boldsymbol{\Theta}_{\text{M-ZFC}}\text{M-ZFCs}^{\omega})^{\top})^{\top}, \tag{5}$$

where ω is the size of the window for sequence learning, M-ZFCs$^{\omega}$ denotes the M-ZFCs feature extracted from the time window with size ω, $\boldsymbol{\Theta}_{\text{M-ZFC}} \in \mathbb{R}^{N \times d_q}$ is a weight matrix to be learned, $\mathscr{F}_{\theta}$ and $\mathscr{F}_{\psi}$ are 1D convolutional layers, and $\boldsymbol{H}_{t,\text{M-ZFC}} \in \mathbb{R}^{N \times d_{\text{out}}^{\text{M-ZFC}}}$ is the $d_{\text{out}}^{\text{M-ZFC}}$-dimensional output. We then combine the embeddings from graph convolution, M-ZFCs convolution, and supra-Hodge convolution to get the final embedding $\boldsymbol{H}_{t,\text{out}}^{(\ell+1)}$

$$\boldsymbol{H}_{t,\text{out}}^{(\ell+1)} = [\boldsymbol{H}_{t,\text{GC}}^{(\ell+1)}, \boldsymbol{H}_{t,\text{M-ZFCs}}, \boldsymbol{H}_{t,\text{SH}}^{(\ell+1)}], \tag{6}$$

where $[\cdot,\cdot,\cdot]$ denotes the concatenation of the outputs from three convolution operations, and $\boldsymbol{H}_{t,\text{out}}^{(\ell+1)} \in \mathbb{R}^{N \times d_{\text{out}}}$ (where $d_{\text{out}} = d_{\ell+1}^{\text{GC}} + d_{\text{out}}^{\text{ZFC}} + d_{\ell+1}^{\text{SH}}$).

4.4 Gate Recurrent Unit with ZFC-SHCN

To describe the complex spatio-temporal dependencies among time series and assess a hidden state of nodes at a future timestamp, we feed the final embedding $\boldsymbol{H}_{t,\text{out}}^{(\ell+1)}$ into Gated Recurrent Units (GRUs). Formally, we set the forward propagation equations of the GRUs as

$$\begin{aligned}
\boldsymbol{\Re}_t &= \eta\left(\boldsymbol{W}_{\Re}\left[\boldsymbol{\Psi}_{t-1}, \boldsymbol{H}_{t,\text{out}}^{(\ell+1)}\right] + \boldsymbol{b}_{\Re}\right), \qquad \boldsymbol{\Im}_t = \eta\left(\boldsymbol{W}_{\Im}\left[\boldsymbol{\Psi}_{t-1}, \boldsymbol{H}_{t,\text{out}}^{(\ell+1)}\right] + \boldsymbol{b}_{\Im}\right), \\
\boldsymbol{\Psi}_t &= \tanh\left(\boldsymbol{W}_{\Psi}\left[\boldsymbol{\Im}_t \odot \boldsymbol{\Psi}_{t-1}, \quad \boldsymbol{H}_{t,\text{out}}^{(\ell+1)}\right] + \boldsymbol{b}_{\Psi}\right), \quad \tilde{\boldsymbol{\Psi}}_t = \boldsymbol{\Re}_i \odot \boldsymbol{\Psi}_{t-1} + (1 - \boldsymbol{\Re}_t) \odot \boldsymbol{\Psi}_t,
\end{aligned}$$

where $\eta(\cdot)$ is an activation function (e.g., ReLU, LeakyReLU), $\odot$ is the elementwise product, $\boldsymbol{\Re}_t$ is the update gate and $\boldsymbol{\Im}_i$ is the reset gate. Here $\boldsymbol{b}_{\Re}$, $\boldsymbol{b}_{\Im}$, $\boldsymbol{b}_{\Psi}$, $\boldsymbol{W}_{\Re}$, $\boldsymbol{W}_{\Im}$, and $\boldsymbol{W}_{\Psi}$ are learnable parameters, while $\left[\boldsymbol{\Psi}_{t-1}, \boldsymbol{H}_{t,\text{out}}^{(\ell+1)}\right]$ and $\boldsymbol{\Psi}_t$ are the input and output of GRU model, respectively. We then obtain $\tilde{\boldsymbol{\Psi}}_t$ which contains both the spatio-temporal and time-aware information.

5 Experiments

5.1 Datasets

We validate our ZFC-SHCN model on six diverse data types: (i) COVID-19 datasets [54]: CA, PA, and TX represent the number of COVID-19 hospitalizations in California (CA), Pennsylvania (PA), and Texas (TX) respectively; (ii) traffic datasets [17]: PeMSD4 and PeMSD8 are two real-time traffic datasets from California; (iii) synthetic multivariate time-series (MTS) datasets based on vector autoregression (VAR) [33, 48] (where the VAR model is a generalization of the univariate AR process with more than one time-evolving component); (iv) daily surface air temperature in CA, PA, and TX over 02/01/2020–12/31/2020; (v) Bytom token prices of Ethereum blockchain over 07/27/2017–05/07/2018; and (vi) wind speed data of 57 stations on the East Coast. The results on (i)–(iii) are presented in the main body, and the analysis of (iv) and (v) is in Appendix C.2 and C.6. The detailed description of each dataset is in Appendix C.1. We also report results on the wind speed dataset in Appendix C.3.

5.2 Baselines

We compare our proposed ZFC-SHCN with 14 types of state-of-the-art baselines (SOAs), including FC-LSTM [56], SFM [63], N-BEATS [49], DCRNN [45], LSTNet [42], STGCN [62], TCN [5], DeepState [51], GraphWaveNet [60], DeepGLO [55], LRGCN [43] AGCRN [4], StemGNN [11], and Z-GCNETs [21].

5.3 Experimental settings

We implement ZFC-SHCN within a Pytorch framework on NVIDIA GeForce RTX 3090 GPU. We optimize all the models using an Adam optimizer for a maximum of 200 epochs. The learning rate is searched in $\{0.001, 0.003, 0.005, 0.01, 0.05\}$ and the embedding dimension is searched in $\{1, 2, 3, 5, 10\}$. Our ZFC-SHCN is trained with batch sizes of 64 and 8 on PeMSD4 and PeMSD8, respectively. On both COVID-19 and surface air temperature datasets (i.e., CA, PA, and TX), we set the batch size to be 8. We train two 1D convolutional layers for ZFC representation learning with the same hidden layer dimension $nhid$ where $nhid \in \{8, 16, 32, 64, 128\}$. For PeMSD4 and PeMSD8, we consider the window size $\omega = 12$ and the horizon $h = 3$; for both COVID-19 and surface air temperature datasets, we consider a window size $\omega = 5$ and horizon $h = 15$; for two simulated VAR datasets VAR_{T_1} and VAR_{T_2}, we set the window size as $\omega = 10$ and horizon as $h = 5$, and set the batch size as 8; for Bytom, we consider the window size $\omega = 7$ and horizon $h = 7$, and set the batch size as 8; for the wind speed dataset, we consider the window size $\omega = 12$ and horizon $h = 12$, and set the batch size as 8. All models are evaluated in terms of Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). The best results are shown in **bold** font and the results shown with *dotted underlines* are the second-best results. We also perform a one-sided two-sample t-test between the best result and the best performance achieved by the runner-up, where *, **, *** denote p-value < 0.1, 0.05, 0.01 (i.e., denote significant, statistically significant, and highly statistically significant results, respectively. Code is available at `https://github.com/zfcshcn/ZFC-SHCN.git`.

5.4 Experimental results

Real datasets The experimental results on PeMSD4 and PeMSD8 traffic data are reported in Table 2. As Table 2 shows, ZFC-SHCN achieves the best MAE, RMSE, and MAPE compared with SOAs on both PeMSD4 and PeMSD8. Compared to the RNN-based methods such as FC-LSTM, SFM, N-BEATS, LSTNet, and TCN, ZFC-SHCN achieves relative gains in RMSE over the runner-ups, ranging from 17.68% to 65.41% for both PeMSD4 and PeMSD8. In turn, DCRNN, STGCN, GraphWaveNet, AGCRN, and StemGNN only focus on learning node-level representations. Compared to them, ZFC-SHCN captures interactions and encodes higher-order structure correlations beyond pairwise relations among nodes and yields a relative gain from 2.06% to 5.63% in RMSE on the traffic datasets. In addition, we compare ZFC-SHCN with the method based on the zigzag persistence image, i.e., Z-GCNETs, and find that ZFC-SHCN outperforms Z-GCNETs by 1.75% on PeMSD4 and 5.36% on PeMSD8 in terms of RMSE. Table 3 presents COVID-19 hospitalization prediction results (RMSE) in CA, PA, and TX, and we observe the following findings. First, our proposed ZFC-SHCN achieves state-of-the-art performance on all three datasets. For instance, ZFC-SHCN yields 3.61%, 1.47%, 65.55% relative gains in RMSE over the runner-ups (including both GCN-based and zigzag persistence image-based methods) on three biosurveillance datasets. These results indicate that the ZFC mechanism and higher-order representation learning module play significant roles in capturing both topological information and higher-order structures. Second, as shown in Figure 2 in Appendix C, we find that, compared to the runner-up (i.e., Z-GCNETs), the predicted value of COVID-19 hospitalizations is more consistent with the ground-truth. Finally, Tables 6 and 10 in Appendix C present the overall prediction performances of ZFC-SHCN and representative baselines on surface air temperature and Ethereum blockchain datasets. We find that our proposed ZFC-SHCN consistently outperforms all baselines with either a significant or (highly) statistically significant margin across all data, except surface air temperature in TX, where ZFC-SHCN still yields the best performance across all models.

Table 1: Forecasting performance (RMSE) of ZFC-SHCN and top three baselines on synthetic time series following vector autoregressions (VAR).

Model	**VAR_{T_1}**	**VAR_{T_2}**
AGCRN [4]	0.50±0.03	0.44±0.02
StemGNN [11]	0.51±0.02	*0.42±0.02*
Z-GCNETs [21]	*0.49±0.02*	0.44±0.01
ZFC-SHCN (ours)	***0.45±0.01**	***0.38±0.01**

Synthetic datasets The evaluation results on two VAR datasets are summarized in Table 1. Compared to the three strongest baselines (i.e., AGCRN, StemGNN, and Z-GCNETs), we observe that our proposed ZFC-SHCN consistently yields the best performance for all synthetic datasets. More precisely, ZFC-SHCN outperforms the runner-ups from 8.89% to 10.52% for VAR_{T_1} and VAR_{T_2}. Furthermore, to assess the time-wise and high network interactions, we use the global clustering coefficient (GCC) and Euler-Poincaré characteristic (EPC) as measures of higher order substructures [6]. We find that

Table 2: Performance comparison of all methods on PEMSD4 and PEMSD8 traffic data.

Model	**PeMSD4**			**PeMSD8**		
	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
FC-LSTM [56]	27.14	41.59	18.20	22.20	34.06	14.20
SFM [63]	24.36	37.10	17.20	16.01	27.41	10.40
N-BEATS [49]	25.56	39.90	17.18	19.48	28.32	13.50
DCRNN [45]	24.70	38.12	17.12	17.86	27.83	11.45
LSTNet [42]	24.04	37.38	17.01	20.26	31.96	11.30
STGCN [62]	22.70	35.50	14.59	18.02	27.83	11.40
TCN [5]	26.31	36.11	15.62	15.93	25.69	16.50
DeepState [51]	26.50	33.00	15.40	19.34	27.18	16.00
GraphWaveNet [60]	26.85	39.70	17.29	19.13	28.16	12.68
DeepGLO [55]	25.45	35.90	12.20	15.12	25.22	13.20
AGCRN [4]	17.78	29.17	11.79	14.59	23.06	9.29
StemGNN [11]	20.20	31.83	12.00	15.83	24.93	9.26
Z-GCNETs [21]	18.05	29.08	11.79	14.52	23.00	9.28
ZFC-SHCN (ours)	*****17.58**	*****28.58**	*****11.68**	*****13.86**	*****21.83**	*****8.92**

Table 3: Performance on COVID-19 hospitalizations in CA, PA, and TX.

Model	**CA**			**PA**			**TX**		
	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
FC-LSTM [56]	167.86±3.25	502.29±4.16	90.71±7.17	47.60±2.36	108.74±2.28	69.37±2.38	20.75±0.50	71.66±2.15	89.16±8.96
SFM [63]	103.98±1.46	475.66±3.23	66.91±5.70	45.91±2.42	106.12±1.87	70.15±5.60	17.49±0.58	70.18±1.91	85.85±8.25
N-BEATS [49]	105.99±1.51	476.10±2.89	67.55±5.16	46.91±2.82	105.86±1.95	15.82±4.39	21.25±0.56	69.26±1.61	85.11±8.77
DCRNN [45]	107.20±1.00	492.10±2.96	69.83±5.57	47.49±2.13	107.21±1.63	67.15±3.94	17.15±0.55	70.47±2.28	88.95±8.01
LSTNet [42]	105.63±1.82	480.61±3.91	67.72±2.17	46.01±2.70	105.67±1.76	72.33±4.21	13.06±0.53	72.93±2.20	84.23±9.01
STGCN [62]	102.88±1.11	470.52±3.06	69.73±5.69	52.69±2.40	106.78±1.87	69.36±4.59	12.69±0.49	63.26±2.25	60.47±8.37
TCN [5]	110.82±1.35	492.82±3.54	70.00±6.92	49.80±2.36	105.07±1.90	69.86±4.91	13.70±0.40	67.08±1.89	65.23±8.61
DeepState [51]	100.57±1.58	469.15±3.71	68.24±5.79	48.46±2.70	107.61±2.31	67.69±4.21	13.24±0.46	59.47±1.93	62.38±8.60
GraphWaveNet [60]	89.64±1.80	394.83±3.35	67.61±5.36	48.08±2.18	109.41±1.91	69.39±4.27	12.66±0.56	58.98±1.22	60.33±8.87
DeepGLO [55]	95.58±1.94	455.80±3.18	67.35±5.99	47.66±2.57	103.74±2.07	68.71±4.09	12.02±0.60	53.09±1.97	62.88±8.83
AGCRN [4]	87.24±1.77	448.27±2.78	66.30±5.17	44.69±2.49	103.79±3.08	63.45±4.00	10.93±0.49	52.96±3.92	59.92±8.99
StemGNN [11]	82.36±2.23	377.25±3.91	67.90±5.94	42.98±2.80	103.15±1.87	63.47±4.11	9.57±0.80	51.00±2.60	59.60±8.81
Z-GCNETs [21]	81.22±2.15	356.35±3.20	62.81±5.75	43.52±2.20	106.22±1.27	65.89±4.66	9.37±0.50	46.04±1.78	59.21±8.73
ZFC-SHCN (ours)	**79.33±1.70**	****343.92±3.66**	**61.25±5.80**	**42.74±2.24**	**101.65±1.53**	**61.90±4.07**	****9.20±0.39**	*****27.81±1.72**	**55.34±8.60**

for GCC for VAR_{T_1} and VAR_{T_2} are 4.96 and 5.87, respectively; while the average EPC for VAR_{T_1} and VAR_{T_2} are 7.47 and 6.91, respectively. Interestingly (although it could be expected), higher GCC and lower EPC tend to be associated with higher relative gains delivered by ZFC-SHCN. Finally, in Appendix C.7, we present the sensitivity analysis for ZFC as a function of the covariance matrix in VAR models.

5.5 Ablation studies

To evaluate the performance of different components in our ZFC-SHCN model, we perform an expansive ablation study. The ablation study is conducted with three setups: (i) ZFC-SHCN without graph convolution in spatial dimension (W/o Graph convolution in spatial dimension), (ii) ZFC-SHCN without ZFC convolution (W/o ZFC convolution), and (iii) ZFC-SHCN without supra-Hodge convolution (W/o Supra-Hodge convolution). The experimental results are shown in Table 4 and prove the validity of each component. As Table 4 indicates, compared to ZFC-SHCN w/o ZFC convolution, the zigzag homological feature is vital for capturing the topological structure of spatio-temporal graph and our proposed graph convolution operation on ZFC significantly improves forecasting performance. By comparing to ZFC-SHCN w/o supra-

Table 4: Ablation study on PeMSD4 and COVID-19 hospitalizations in TX.

	Architecture	**RMSE**
PeMSD4	**ZFC-SHCN**	*****28.58±0.15**
	W/o Graph convolution in spatial dimension	28.82±0.28
	W/o ZFC convolution	28.66±0.20
	W/o Supra-Hodge convolution	28.69±0.19
TX-COVID-19	**ZFC-SHCN**	****27.81±1.72**
	W/o Graph convolution in spatial dimension	29.17±1.73
	W/o ZFC convolution	29.75±1.96
	W/o Supra-Hodge convolution	30.11±1.77

Hodge convolution, we illustrate the significance of higher-order structure representation learning for guiding the model to how to capture information on higher-order interactions. Also, ZFC-SHCN w/o graph convolution in spatial dimension demonstrates that the learned graph obtained from trainable weights can learn hidden information and enhance (multivariate) time-series representation learning.

5.6 Computational complexity

For higher-order simplices, the incidence matrices $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ can be calculated efficiently with complexity $\mathcal{O}(N+M)$ and $\mathcal{O}(M+Q)$ respectively, where N is the number of 0-simplices (i.e., nodes), M is the number of 1-simplices (i.e., edges), and Q is the number of 2-simplices (i.e., filled triangles). The computational complexity of ZFC is $\mathcal{O}(\Upsilon^{\delta})$ [3, 23], where Υ represents the number of points in time interval and $\delta \in [2, 2.373)$. The computational complexity of the overall approach is $\mathcal{O}(N^2 + \Upsilon^{\delta} + \Xi_k \omega F_k d_{out} + \Xi_k \omega^2 d_{out}/2 + d_{out} \sum_{\ell=t-w}^{t-1} \Xi_{k+1}^{(\ell)} + W_{GRU})$, including (i) graph convolution in spatial dimension: $\mathcal{O}(N^2)$, (ii) zigzag filtration curve: $\mathcal{O}(\Upsilon^{\delta})$, (iii) supra-Hodge convolution in temporal dimension: $\mathcal{O}(\Xi_k \omega F_k d_{out} + \Xi_k \omega^2 d_{out}/2 + d_{out} \sum_{\ell=t-w}^{t-1} \Xi_{k+1}^{(\ell)})$ (where F_k is the number of k-simplex attribute features, ω is the sliding window size, d_{out} is the output dimension of the supra-Hodge convolution layer, and $\Xi_{k+1}^{(\ell)}$ is the number of (k+1)-simplex Ξ_{k+1} at the ℓ-th layer), and (iv) GRU: $\mathcal{O}(W_{GRU})$. We also compare our ZFC-SHCN with the most recent approach based on multipersistence-GNN [20] (i.e., TAMP-S2GCNets). We find that ZFC-SHCN yields either on-par or more competitive performance than TAMP-S2GCNets, while our proposed ZFC-SHCN significantly improves the computational efficiency (see Appendix C.5 for more details). More details about running time comparison can be found in Appendix C.6.

6 Conclusion

We have proposed a novel framework for time-aware deep learning of time-evolving objects which takes advantages of both the higher-order interactions among the data substructures, described as simplices, and the most intrinsic time-conditioned topological information exhibited by the object, characterized via zigzag persistent homology. By leveraging the power of simplicial convolution operation and zigzag persistence for time-indexed data, ZFC-SHCN has been shown to demonstrate capabilities to yield the most competitive forecasting performance, while requiring fewer computational resources than its closest competitors. Still, computational complexity and limited theoretical results on statistical inference for zigzag persistence remain one of the major existing limitations of ZFC and, more generally, all topological methods for time dependent processes. In the future, we plan to investigate these theoretical and methodological challenges and will extend the ZFC-SHCN idea to anomaly detection in streaming time-dependent processes.

Acknowledgments

This work was partially supported by the National Science Foundation (NSF) under awards # ECCS 2039701 and # ECCS-2039716, the Department of the Navy, Office of Naval Research (ONR) under ONR award # N00014-21-1-2530, C3.ai Digital Transformation Institute, and NASA AIST grant 21-AIST21_2-0059. Part of this material is also based upon work supported by (while serving at) the NSF. 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 NSF, ONR, C3.ai DTI, or NASA.

References

[1] S. Agarwal, K. Branson, and S. Belongie. Higher order learning with graphs. In *Proceedings of the International Conference on Machine Learning*, pages 17–24, 2006.

[2] C. G. Akcora, M. Kantarcioglu, and Y. R. Gel. Blockchain networks: Data structures of bitcoin, monero, zcash, ethereum, ripple and iota. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1436, 2022.

[3] J. Alman and V. V. Williams. A refined laser method and faster matrix multiplication. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 522–539. SIAM, 2021.

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

[5] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[6] F. Battiston, G. Cencetti, I. Iacopini, V. Latora, M. Lucas, A. Patania, J. Young, and G. Petri. Networks beyond pairwise interactions: structure and dynamics. *Physics Reports*, 874:1–92, 2020.

[7] A. R. Benson, R. Abebe, M. T. Schaub, A. Jadbabaie, and J. Kleinberg. Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences*, 115:E11221–30, 2018.

[8] C. Bodnar, F. Frasca, N. Otter, Y. G. Wang, P. Lio, G. F. Montufar, and M. Bronstein. Weisfeiler and Lehman go cellular: Cw networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 2625–2640, 2021.

[9] C. Bodnar, F. Frasca, Y. G. Wang, N. Otter, G. Montúfar, P. Lio, and M. Bronstein. Weisfeiler and Lehman go topological: Message passing simplicial networks. In *Proceedings of the International Conference on Machine Learning*, 2021.

[10] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015.

[11] D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, Y. Tong, B. Xu, J. Bai, J. Tong, and Q. Zhang. Spectral temporal graph neural network for multivariate time-series forecasting. In *Advances in Neural Information Processing Systems*, volume 33, pages 17766–17778, 2020.

[12] G. Carlsson, V. De Silva, S. Kališnik, and D. Morozov. Parametrized homology via zigzag persistence. *Algebraic & Geometric Topology*, 19(2):657–700, 2019.

[13] G. Carlsson, V. De Silva, and D. Morozov. Zigzag persistent homology and real-valued functions. In *Proceedings of the Twenty-fifth Annual Symposium on Computational Geometry*, pages 247–256, 2009.

[14] M. Carrière and A. Blumberg. Multiparameter persistence image for topological machine learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 22432–22444, 2020.

[15] M. Carrière, F. Chazal, Y. Ike, T. Lacombe, M. Royer, and Y. Umeda. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 2786–2796, 2020.

[16] C. Chatfield. *The Analysis of Time Series: An Introduction*. Chapman and hall/CRC, 2003.

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

[18] Y. Chen, B. Coskunuzer, and Y. R. Gel. Topological relational learning on graphs. In *Advances in Neural Information Processing Systems*, volume 34, pages 27029–27042, 2021.

[19] Y. Chen, Y. R. Gel, and H. V. Poor. BScNets: Block simplicial complex neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6333–6341, 2022.

[20] Y. Chen, I. Segovia-Dominguez, B. Coskunuzer, and Y. R. Gel. TAMP-S2GCNets: coupling time-aware multipersistence knowledge representation with spatio-supra graph convolutional networks for time-series forecasting. In *Proceedings of the International Conference on Learning Representations*, 2022.

[21] Y. Chen, I. Segovia-Dominguez, and Y. R. Gel. Z-GCNETs: Time zigzags at graph convolutional networks for time series forecasting. In *Proceedings of the International Conference on Machine Learning*, pages 1684–1694, 2021.

[22] A. K. Dey, Y. R. Gel, and H. V. Poor. What network motifs tell us about resilience and reliability of complex networks. *Proceedings of the National Academy of Sciences*, 116(39):19368–19373, 2019.

[23] T. K. Dey and T. Hou. Fast computation of zigzag persistence. *arXiv preprint arXiv:2204.11080*, 2022.

[24] M. di Angelo and G. Salzer. Tokens, types, and standards: Identification and utilization in ethereum. In *Proceedings of the IEEE International Conference on Decentralized Applications and Infrastructures*, pages 1–10, 2020.

[25] X. Dong, D. Thanou, M. Rabbat, and P. Frossard. Learning graphs from data: A signal representation perspective. *IEEE Signal Processing Magazine*, 36(3):44–63, 2019.

[26] S. Ebli, M. Defferrard, and G. Spreemann. Simplicial neural networks. In *Advances in Neural Information Processing Systems, Workshop on TDA and Beyond*, 2020.

[27] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 454–463, 2000.

[28] A. Ghaderi, B. M. Sanandaji, and F. Ghaderi. Deep forecast: Deep learning-based spatio-temporal forecasting. In *Proceedings of the International Conference on Machine Learning, Time Series Workshop*, 2017.

[29] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864, 2016.

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

[31] G. Gupta, S. Pequito, and P. Bogdan. Dealing with unknown unknowns: Identification and selection of minimal sensing for fractional dynamics with unknown inputs. In *Proceedings of the Annual American Control Conference*, pages 2814–2820. IEEE, 2018.

[32] G. Gupta, X. Xiao, R. Balan, and P. Bogdan. Non-linear operator approximations for initial value problems. In *Proceedings of the International Conference on Learning Representations*, 2021.

[33] J. D. Hamilton. *Time Series Analysis*. Princeton University Press, 2020.

[34] C. Hofer, F. Graf, B. Rieck, M. Niethammer, and R. Kwitt. Graph filtration learning. In *Proceedings of International Conference on Machine Learning*, pages 4314–4323, 2020.

[35] C. Hofer, R. Kwitt, M. Niethammer, and A. Uhl. Deep learning with topological signatures. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[36] C. D. Hofer, R. Kwitt, and M. Niethammer. Learning representations of persistence barcodes. *Journal of Machine Learning Research*, 20(126):1–45, 2019.

[37] M. Horn, E. De Brouwer, M. Moor, Y. Moreau, B. Rieck, and K. Borgwardt. Topological graph neural networks. In *Proceedings of International Conference on Learning Representations*, 2021.

[38] J. L. Johnson and T. Goldring. Discrete hodge theory on graphs: a tutorial. *IEEE Computing in Science & Engineering*, 15(5):42–55, 2012.

[39] M. Johnson and J. Jung. Instability of the betti sequence for persistent homology and a stabilized version of the betti sequence. *arXiv preprint arXiv:2109.09218*, 2021.

[40] A. Kapoor, X. Ben, L. Liu, B. Perozzi, M. Barnes, M. Blais, and S. O'Banion. Examining covid-19 forecasting using spatio-temporal graph neural networks. *arXiv preprint arXiv:2007.03113*, 2020.

[41] W. Kim and F. Memoli. Stable signatures for dynamic graphs and dynamic metric spaces via zigzag persistence. *arXiv preprint arXiv:1712.04064*, 2017.

[42] G. Lai, W. C. Chang, Y. Yang, and H. Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *Proceedings of the International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 95–104, 2018.

[43] J. Li, Z. Han, H. Cheng, J. Su, P. Wang, J. Zhang, and L. Pan. Predicting path failure in time-evolving graphs. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1279–1289, 2019.

[44] J. Li, F. Yang, M. Tomizuka, and C. Choi. Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning. *Advances in Neural Information Processing Systems*, 33:19783–19794, 2020.

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

[46] L. H. Lim. Hodge laplacians on graphs. *SIAM Review*, 2020.

[47] L. O'Bray, B. Rieck, and K. Borgwardt. Filtration curves for graph representation. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1267–1275, 2021.

[48] D. Ofori-Boateng, Y. R. Gel, and I. Cribben. Nonparametric anomaly detection on time series of graphs. *Journal of Computational and Graphical Statistics*, 30(3):756–767, 2021.

[49] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *Proceedings of the International Conference on Learning Representations*, 2019.

[50] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 701–710, 2014.

[51] S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski. Deep state space models for time series forecasting. In *Advances in Neural Information Processing Systems*, volume 31, pages 7785–7794, 2018.

[52] T. M. Roddenberry, N. Glaze, and S. Segarra. Principled simplicial neural networks for trajectory prediction. In *Proceedings of the International Conference on Machine Learning*, pages 9020–9029, 2021.

[53] M. T. Schaub, A. R. Benson, P. Horn, G. Lippner, and A. Jadbabaie. Random walks on simplicial complexes and the normalized hodge 1-laplacian. *SIAM Review*, 62(2):353–391, 2020.

[54] I. Segovia Dominguez, H. Lee, Y. Chen, M. Garay, K. M. Gorski, and Y. R. Gel. Does air quality really impact covid-19 clinical severity: Coupling nasa satellite datasets with geometric deep learning. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3540–3548, 2021.

[55] R. Sen, H. F. Yu, and I. S. Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[56] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27, pages 3104–3112, 2014.

[57] S. Tymochko, E. Munch, and F. Khasawneh. Bifurcation analysis using zigzag persistence. In *Advances in Neural Information Processing Systems, Workshop on Topological Data Analysis and Beyond*, 2020.

[58] V. Vassilevska, R. Williams, and R. Yuster. Finding the smallest h-subgraph in real weighted graphs and related problems. In *Automata, Languages and Programming*, pages 262–273, 2006.

[59] L. Wang, X. Ben, A. Adiga, A. Sadilek, A. Tendulkar, S. Venkatramanan, A. Vullikanti, G. Aggarwal, A. Talekar, J. Chen, et al. Using mobility data to understand and forecast covid19 dynamics. *medRxiv*, 2020.

[60] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2019.

[61] Z. Yan, T. Ma, L. Gao, Z. Tang, and C. Chen. Link prediction with persistent homology: An interactive view. In *Proceedings of International Conference on Machine Learning*, pages 11659–11669, 2021.

[62] B. Yu, H. Yin, and Z. Zhu. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 3634–3640, 2018.

[63] L. Zhang, C. Aggarwal, and G. J. Qi. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2141–2149, 2017.

[64] Q. Zhao, Z. Ye, C. Chen, and Y. Wang. Persistence enhanced graph neural network. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 2896–2906, 2020.

[65] A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See the Conclusion Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See the Broader Impact Section in Appendix D.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 3 and Appendix A.
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Section 5.3.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5.3.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 5.3.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes]

   (b) Did you mention the license of the assets? [No] They use public-domain-equivalent license.

   (c) Did you include any new assets either in the supplemental material or as a URL? [No]

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

A Proof of Proposition 3.2

Let $\eta_0 : \mathrm{DgmZZ}_{\nu_*} \mapsto \mathrm{DgmZZ}'_{\nu_*}$ be the bijective map that yields the Wasserstein-1 distance between DgmZZ_{ν_*} and its perturbed copy DgmZZ'_{ν_*}. That is, Wasserstein-1 distance is defined as

$$\mathcal{W}_1(\mathrm{DgmZZ}_{\nu_*}, \mathrm{DgmZZ}'_{\nu_*}) = \inf_{\eta}\Big(\sum_{\mathrm{x}\in \mathrm{DgmZZ}_{\nu_*}\cup\mathcal{L}} ||\mathrm{x} - \eta(\mathrm{x})||^1_\infty \Big),$$

where $||\cdot||_\infty$ is a norm in $\mathcal{L}^\infty$, i.e., $||z||_\infty = \max_i |z_i|$, $\Gamma = \{(t,t)|t\in\mathbb{R}\}$ and ν ranges over all bijections between $\mathrm{DgmZZ}_{\nu_*} \cup \Gamma$ and $\mathrm{DgmZZ}'_{\nu_*} \cup \Gamma$. Inclusion of the diagonal Γ ensures that the set of bijections is non-empty even under different cardinalities of DgmZZ_{ν_*} and DgmZZ'_{ν_*}.

Let $\mathrm{ZFC}_{\nu_*}(\Delta t_i^-)$ and $\mathrm{ZFC}'_{\nu_*}(\Delta t_i^-)$ be Zigzag Filtration Curves corresponding to the zigzag persistence diagrams DgmZZ_{ν_*} and DgmZZ'_{ν_*}, respectively, and evaluated at Δt_i^-.

Then, similarly to [39], for each $i = \{1, 2, \ldots, \mathcal{N}\}$, we have

$$\begin{aligned}\Big|\mathrm{ZFC}_{\nu_*}(\Delta t_i^-) - \mathrm{ZFC}'_{\nu_*}(\Delta t_i^-)\Big| &= \Bigg[\sum_{j=1}^{\mathcal{M}} \xi_i(t_{b_j}, t_{d_j})\omega_i - \sum_{j=1}^{\mathcal{M}} \xi_i(\eta_0(t_{b_j}, t_{d_j}))\omega_i\Bigg] \\ \leq \omega_i \sum_{j=1}^{M}\Big[\xi_i(t_{b_j}, t_{d_j}) - \xi_i(\eta_0(t_{b_j}, t_{d_j}))\Big] &\leq \omega_i L_i \sum_{j=1}^{\mathcal{M}}\Big[(t_{b_j}, t_{d_j}) - \eta_0(t_{b_j}, t_{d_j})\Big] \\ &= \omega_i L_i \mathcal{W}_1\big(\mathrm{DgmZZ}_{\nu_*}, \mathrm{DgmZZ}'_{\nu_*}\big) < \omega_i L_i \epsilon.\end{aligned}$$

Here in the second inequality, we take into account that ξ_i is a Lipschitz continuous function with constant L_i. A similar result holds for $\mathrm{ZFC}_{\nu_*}(\Delta t_i^+)$ and $\mathrm{ZFC}'_{\nu_*}(\Delta t_i^+)$, $i = \{1, 2, \ldots, \mathcal{N}\}$.

Hence,

$$||\mathrm{ZFC}_{\nu_*} - \mathrm{ZFC}'_{\nu_*}||_\infty \leq C W_1\big(\mathrm{DgmZZ}_{\nu_*}, \mathrm{DgmZZ}'_{\nu_*}\big),$$

where $C = \max_{\{i=1,2,\ldots,\mathcal{N}\}} \omega_i L_i$ and ZFC_{ν_*} and ZFC'_{ν_*} are viewed as $(2n-1)$-dimensional vectors (i.e., ZFC is evaluated at each interval Δt_i^+ and Δt_i^-, $i = 1, 2, \ldots, \mathcal{N}$), which concludes the proof.

B Background on the Hodge Theory and Hodge-Laplacians

Let C^k be a real-valued vector space which is endowed with basis from the oriented k-simplices. (By orientation of simplices, we mean selecting some (arbitrary) order for its nodes, where two orderings are said to be equivalent if they differ by an even permutation.) A linear map $\partial_k : C^k \to C^{k-1}$ is called a *boundary* operator. The adjoint of the boundary map induces the *co-boundary* operator $\partial_k^T : C^k \to C^{k+1}$. We define matrix representations of ∂_k and $\partial_k^\top$ as $\boldsymbol{B}_k$ and $\boldsymbol{B}_k^\top$, respectively.

Definition B.1. An operator over oriented k-simplices $\boldsymbol{\mathcal{L}}_k : C^k \to C^k$ is called the *k-Hodge Laplacian*, and its matrix representation is given by

$$\boldsymbol{\mathcal{L}}_k = \boldsymbol{B}_k^\top \boldsymbol{B}_k + \boldsymbol{B}_{k+1}\boldsymbol{B}_{k+1}^\top, \tag{7}$$

where $\boldsymbol{B}_k^\top \boldsymbol{B}_k$ and $\boldsymbol{B}_{k+1}\boldsymbol{B}_{k+1}^\top$ are often referred to $\boldsymbol{\mathcal{L}}_k^{down}$ and $\boldsymbol{\mathcal{L}}_k^{up}$, respectively.

That is, the standard graph Laplacian $\boldsymbol{\mathcal{L}}_0 = \boldsymbol{B}_1\boldsymbol{B}_1^\top \in \mathbb{R}^{N\times N}$ is a special case of the above k-th combinatorial Hodge Laplacian and the matrix $\boldsymbol{\mathcal{L}}_1 \in \mathbb{R}^{M\times M}$ is the Hodge 1-Laplacian.

C Additional Experiments and Running Time

C.1 Datasets

Detailed data description is as follows:

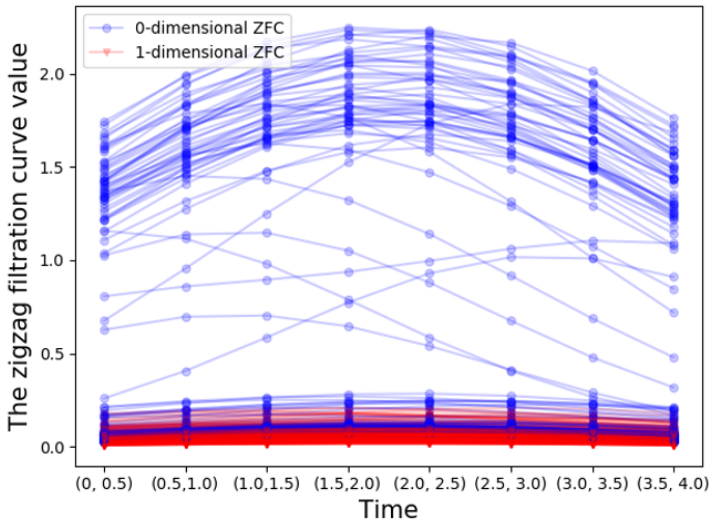


Figure 1: Illustration of 0- and 1- dimensional ZFCs for COVID-19 dataset on Texas (TX). For COVID-19 hospitalization forecasting on TX, we set the sliding window size to be 5 and thus $\mathcal{N} = 8$ over the time period $[t_1, t_5]$.

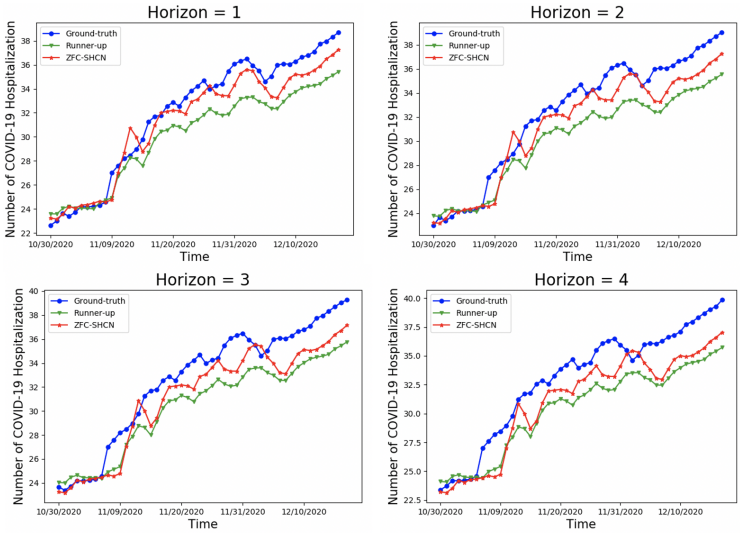


Figure 2: Comparison of forecasting performance at four different prediction horizons for COVID-19 hospitalization in TX.

1. The **COVID-19** datasets include the daily number of hospitalizations from February 1, 2020 to December 31, 2021 on a county level in three US states (i.e., CA, PA, and TX), where each node denotes a county and each edge represents a border connection between two adjacent counties. In our experiments, we split the COVID-19 datasets into training and test sets in chronological order with the split ratio $8:2$ for all three states.
2. The **traffic** datasets, i.e., PeMSD4 and PeMSD8 consist of the traffic flow data in California from January 1, 2018 to February 28, 2018 and from January 7, 2016 to August 31, 2016, respectively. Moreover, both PeMSD4 and PeMSD8 are aggregated to 5 minutes, which means there are 12 time points in the flow data per hour, and we split the traffic datasets with ratio $6:2:2$ into training, validation, and test sets by following the same setting of [30].
3. The **synthetic MTS** datasets include two MTS following vector autoregressions VAR_{T_1}: $\{\mathbf{\Xi}_t\}_{t=1}^{T_1} = \{\xi_{1t}, \ldots, \xi_{pt}\}_{t=1}^{T_1}$ and VAR_{T_2}: $\{\mathbf{\Xi}_t\}_{t=1}^{T_2} = \{\xi_{1t}, \ldots, \xi_{pt}\}_{t=1}^{T_2}$ with the graph node size $p = 20$ for two different time series lengths ($T_1 = 200$ and $T_2 = 300$). Note that

Table 5: Summary of the time series forecasting datasets. † represents the average number of edges in simulated VAR datasets under threshold $\omega_{\text{VAR}} = 0.1$.

Dataset	# Nodes	# Edges	Time range	# Timesteps
CA	55	129	02/01/2020 - 12/31/2020	335
PA	60	139	02/01/2020 - 12/31/2020	335
TX	251	709	02/01/2020 - 12/31/2020	335
PeMSD4	307	340	01/01/2018 - 02/28/2018	16,992
PeMSD8	170	274	01/07/2016 - 08/31/2016	17,856
VAR_{T_1}	20	$85^\dagger$	-	100
VAR_{T_2}	20	$73^\dagger$	-	200
Bytom	100	10	07/27/2017 - 05/07/2018	285

for two VAR datasets we split into training and test sets with the ratio $8:2$. We define the ω-lag vector autoregressive (VAR(ω)) $\boldsymbol{\Xi}_t = \{\xi_{1t}, \ldots, \xi_{pt}\}^\top$ as follows

$$\boldsymbol{\Xi}_t = \boldsymbol{a} + \boldsymbol{\Pi}_1 \boldsymbol{\Xi}_{t-1} + \boldsymbol{\Pi}_2 \boldsymbol{\Xi}_{t-2} + \cdots + \boldsymbol{\Pi}_w \boldsymbol{\Xi}_{t-w} + \boldsymbol{\epsilon}_t,$$

where $\boldsymbol{\Pi}_i \in \mathbb{R}^{q \times q}$ is an coefficient matrix and $\boldsymbol{\epsilon}_t \in \mathbb{R}^{q \times 1}$ denotes an unobservable mean white noise vector process with time invariant covariance matrix $\boldsymbol{\Sigma}$. In our experiments, we set $q = 20$ and $\omega = 10$.

4. The **surface air temperature** datasets include the daily surface air temperature from February 1, 2020 to December 31, 2021 on a county level in three US states (i.e., CA, PA, and TX). Besides, we use the same experimental settings as the COVID-19 datasets.
5. The **Ethereum blockchain Bytom networks** are compound of addresses of users (nodes) and daily transactions among users (edges) [24, 2]. Since original token networks have an average of 442788/1192722 nodes/edges, we compute a subgraph via a maximum weight subgraph approximation [58] using the amount of transactions as weight. The dynamic networks contain different number of networks since every token was created at different days. Hence, for each timestamp t (from July 27, 2017 to May 07, 2018), given the dynamic network $\mathcal{G}_t = \{\mathcal{V}_t, \mathcal{E}_t, \tilde{W}_t\}$ and its corresponding node feature matrix $X_t \in \mathbb{R}^{N \times F}$, where F represents the number of features (e.g., node degree, node betweenness, etc), we test our algorithm with both node and edge features and use the set of most active nodes, i.e., $N = 100$.

C.2 Surface Temperature Prediction Performance

We further compare our ZFC-SHCN with the 3 strongest SOA baselines for surface temperature forecasting on CA, PA, and TX in Table 6. The results show that ZFC-SHCN achieves relative gains of 39.55%, 6.02%, and 15.59% over runner-ups (see the results with *dotted underline*) on CA, PA, and TX respectively.

Table 6: Forecasting results (MAPE (%)) on surface temperature in CA, PA, and TX.

Model	CA	PA	TX
AGCRN	4.21±1.05	7.07±1.21	12.97±1.39
StemGNN	4.59±1.21	6.61±1.10	11.94±1.60
Z-GCNETs	4.34±0.87	6.52±1.05	12.00±1.65
ZFC-SHCN (ours)	****3.11±0.65**	**6.15±1.15**	***10.33±1.57**

C.3 Wind Speed Prediction Performance

To assess the utility of our approach for other tasks, we now also conducted spatio-temporal forecasting for graph values (i.e., on graph-level) on the wind speed dataset [28]. In this experiment, we use hourly wind speed data to predict the total wind speed from 57 stations in East Coast including

Massachusetts, Connecticut, New York, and New Hampshire. The evaluation results are summarized in Table 3. We find that our ZFC-SHCN outperforms the next best competitor, with up to 39.20% of improvement, and the results are highly statistically significant.

Table 7: Forecasting results of ZFC-SHCH and baselines on wind speed dataset.

	Model	RMSE
Wind Speed	**ZFC-SHCN**	***$\mathbf{1.07 \pm 0.02}$
	ZPI-SHCN	1.30 ± 0.02
	Z-GCNETs	1.68 ± 0.05
	AGCRN	1.52 ± 0.03
	StemGNN	1.76 ± 0.03

C.4 Additional Ablation Studies

We conduct the additional experiments by (i) replacing ZFC with zigzag persistence image (ZPI), i.e., ZPI-SHCN and (ii) replacing Supra-Hodge convolution with GCN (which is temporal graph convolution in temporal domain), i.e., ZFC-GCN, on both PeMDS4 and TX-COVID-19 (in Texas state) datasets. First, Table 8 demonstrates that for both datasets, the ZFC representation (i.e., ZFC-SHCN) significantly outperforms zigzag persistence image (i.e., ZPI-SHCN). These findings might be explained by the fact that, especially for small- and medium-size datasets, ZFC tends to better capture the topological signal than ZPI which, being a 2-d representation, might be more noisy. Second, Table 8 indicates that compared to ZFC-GCN, ZFC-SHCN always achieves better forecasting performance. We attribute these findings to the fact that supra-Hodge convolution operation integrates knowledge on important interactions among higher-order structures (i.e., edges, filled triangles - beyond the node-space) into graph learning. For TX-COVID-19 dataset, the relative gain of ZFC-SHCN over the runner-up (i.e., ZFC-GCN) in RMSE is 3.13%; for PeMSD4 dataset, the relative gain of ZFC-SHCN over the runner-up (i.e., ZFC-GCN) in RMSE is 1.11%.

Table 8: Additional ablation study of ZFC convolution and Supra-Hodge convolution on PeMSD4 and COVID-19 hospitalizations in TX.

	Architecture	RMSE
PeMSD4	**ZFC-SHCN**	***$\mathbf{28.58 \pm 0.15}$
	ZPI-SHCN	29.00 ± 0.17
	ZFC-GCN	28.90 ± 0.26
TX-COVID-19	**ZFC-SHCN**	**$\mathbf{27.81 \pm 1.72}$
	ZPI-SHCN	29.26 ± 1.86
	ZFC-GCN	28.71 ± 1.90

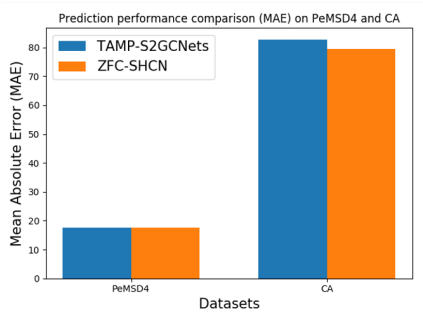
Regarding the choice of simplicial complexes, we have also conducted additional experiments between (i) ZFC-SH^{1}CN based on Supra-Hodge 1-Laplacian (i.e., utilizing information of 0-simplices, 1-simplices, and 2-simplices) and (ii) ZFC-SH^{2}CN based on Supra-Hodge 2-Laplacian (i.e., utilizing information of 1-simplices, 2-simplices, and 3-simplices) on PeMSD4 and COVID-19 in TX dataset. The evaluation results are summarized in Table 9. Experimental results show that ZFC-SH^{1}CN consistently outperforms ZFC-SH^{2}CN. We tend to attribute such findings to the fact that we observe a very low number of 3-simplices. For instance, in PeMSD4 at each timestamp we observe around 29 2-simplices (filled triangles) but we observe only around 2 3-simplices (tetrahedron). Furthermore, it is worth to mention that: for PeMSD4 dataset, ZFC-SH^{2}CN is still very competitive compared to other baselines (i.e., slightly worse than the runner-up, i.e., Z-GCNETs); for COVID-19 in TX, the performance of ZFC-SH^{2}CN is much better the runner-up. Nevertheless, we believe that higher-order simplices, if observed in the targeted network, has a potential to boost the model performance.

Table 9: Additional ablation study of simplices on PeMSD4 and COVID-19 hospitalizations in TX.

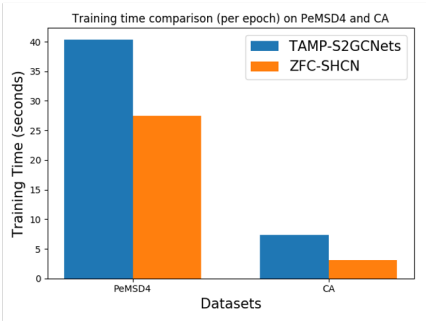
	Architecture	RMSE
PeMSD4	**ZFC-SH^{1}CN**	***$\mathbf{28.58 \pm 0.15}$**
	ZFC-SH^{2}CN	29.15 ± 0.30
TX-COVID-19	**ZFC-SH^{1}CN**	**$\mathbf{27.81 \pm 1.72}$**
	ZFC-SH^{2}CN	29.39 ± 1.93

C.5 Comparison to Multiparameter Persistence

The development of multiparameter persistence (MP) has recently witnessed a rapid progress (such as [14, 20]) due to in many applications, particularly, involving spatio-temporal processes, the data exhibit richer structures which cannot be well encoded with a single parameter persistence. Although there exist some studies in the literature that have shown promising performances for spatial temporal prediction tasks [20], the computation of MP is labor-intensive and very time-consuming. To better demonstrate the power of our proposed ZFC-SHCN, here we compare our method with one representative MP-based GNNs, i.e., TAMP-S2GCNets [20] from both prediction performance and running time perspectives. We conduct comparison experiments on PeMSD4, CA, and Bytom, and we find that ZFC-SHCN yields highly competitive performance against TAMP-S2GCNets (see Figures 3a and 4a); that is, for **PeMSD4**, ZFC-SHCN: {MAE: 17.58 ± 0.17, RMSE: 28.58 ± 0.15, MAPE (%): 11.68 ± 0.07} vs. TAMP-S2GCNets: {MAE: 17.58 ± 0.20 RMSE: 28.56 ± 0.28, MAPE (%) 11.01}; for **CA**: ZFC-SHCN: {MAE: 79.33 ± 1.70, RMSE: 343.92 ± 3.66, MAPE (%): 61.25 ± 5.80} vs. TAMP-S2GCNets: {MAE: 82.75 ± 1.93, RMSE: 371.60 ± 2.68, MAPE (%): 62.43 ± 5.61}; for **Bytom**, ZFC-SHCN: {MAPE: 29.51 ± 0.77} vs. TAMP-S2GCNets: {MAPE: 29.26 ± 1.06}). Furthermore, Figures 3b and 4b present visual comparison of running time of ZFC-SHCN and TAMP-S2GCNets (per epoch in seconds). We find that training time consumed by ZFC-SHCN is substantially lower than that of TAMP-S2GCNets on all datasets. These findings suggest that ZFC-SHCN may be the preferred forecasting choice, in terms of the balance between accuracy and computational efficiency.



(a) Prediction performance comparison.



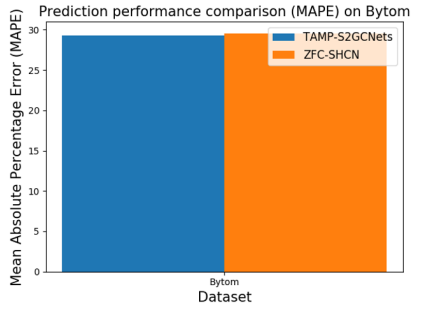
(b) Training time comparison.

Figure 3: Prediction performance (RMSE) and training time comparisons on PeMSD4 and CA.

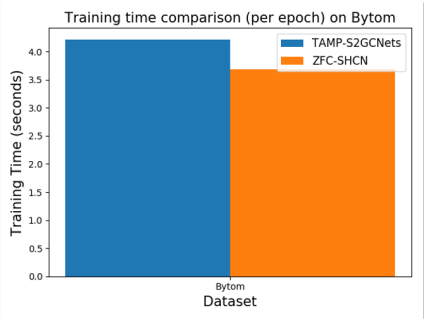
C.6 Overall Comparison with Runner-ups

We present comparison of computational time of our ZFC-SHCN with the 3 strongest SOA baselines on PeMSD4, CA, and Bytom in Table 10. The results show the superior computational efficiency of ZFC-SHCN. Note, that while AGCRN is faster than ZFC-SHCN, ZFC-SHCN significantly outperforms AGCRN in terms of forecasting accuracy, e.g., with a relative gain of up to 48.81% on MAPE for CA. Furthermore, from Tables 11 and 12, we find both the running times on zigzag-based topological feature generation and per-epoch training time of ZFC-SHCN are less than the runner-up model (i.e., Z-GCNETs).

Furthermore, we have conducted comparison experiments between our ZFC-SHCN and two baselines, i.e., fractional-order dynamical model [31] and Padé Exp [32] on COVID-19 (TX) dataset. Table 13 shows that ZFC-SHCN outperforms both fractional-order dynamical model and Padé Exp.



(a) Prediction performance comparison.



(b) Training time comparison.

Figure 4: Prediction performance (MAPE) and training time comparisons on Bytom.

Table 10: Comparisons of prediction performance and computation time (in ( )) on PeMSD4, CA, and Bytom datasets.

Dataset	**RMSE/MAPE (Average Training Time per Epoch (sec))**		
	PeMSD4 (*RMSE*)	CA (*RMSE*)	Bytom (*MAPE*)
AGCRN	29.17±0.09 (25.74 s)	448.27±2.78 (3.01 s)	34.46±1.37 (2.47 s)
StemGNN	31.83±0.11 (33.86 s)	377.25±3.91 (6.63 s)	34.50±1.53 (3.05 s)
Z-GCNETs	29.08±0.19 (30.12 s)	356.35±3.20 (6.56 s)	31.04±0.78 (2.83 s)
ZFC-SHCN (ours)	******28.58±0.15** (27.53 s)	****343.92±3.66** (3.15 s)	******29.51±0.77** (2.82 s)

C.7 Sensitivity analysis of the covariance matrix

To further investigate the effect of covariance matrix in ZFC's representational power, we perform the sensitivity analysis of covariance matrix Σ of ZFC, which specifies the patterns of variability as well as the shape of the multivariate Gaussian distribution. We show that the variation of RMSE for COVID-19 hospitalization prediction task with Σ (note that, we assume Σ is the same for every persistent point (t_{b_j}, t_{d_j})). Table 14 shows variations of RMSE with different diagonal and off-diagonal elements in Σ. As demonstrated in Table 14, (i) RMSE increases with increasing the off-diagonal element (i.e., the correlation coefficient) when fixing diagonal element; (ii) RMSE first decreases and then increases via increasing diagonal element when setting the correlation coefficient to be 0 (i.e., there is no linear relationship between the t_{b_j} and t_{d_j}). In addition, we find evidence of a non-monotonic relationship between prediction performance (in RMSE) and diagonal element of Σ.

D Broader Impact

The key **positive** impact of this project constitutes in the simultaneous enhancement of both predictive performance and computational efficiency in forecasting tasks associated with complex multivariate time series, including but not limited to spatio-temporal processes. Such tasks are widely met in a diverse range of applications, from finance to atmospheric sciences to biosurveillance. Furthermore, topological methods and graph representations of multivariate time series and spatio-temporal processes, in conjunction with forecasting, remains yet nascent in both machine learning and statistics, and has a high potential, especially for analysis of multivariate time series with a complex dependence structure, such as disease surveillance, wildfire monitoring, and cryptomarket analytics. However, the current lack of proper statistical inferential methods for topological characteristics of multivariate

Table 11: Computational costs for generation of zigzag persistence images (ZPI) and a single training epoch of Z-GCNETs.

Dataset	**Average Time Taken (sec)**	
	ZPI	Z-GCNETs (epoch)
CA	0.23 s	6.56 s
PeMSD4	0.86 s	30.12 s

Table 12: Computational costs for generation of zigzag filtration curve (ZFC) and a single training epoch of ZFC-SHCN.

Dataset	**Average Time Taken (sec)**	
	ZFC	ZFC-SHCN (epoch)
CA	0.10 s	3.15 s
PeMSD4	0.52 s	27.53 s

Table 13: Comparison of RMSE of ZFC-SHCN and fractional-order dynamical model on COVID-19 in TX.

	Architecture	**RMSE**
TX-COVID-19	**ZFC-SHCN**	**27.81±1.72**
	Fractional-order dynamical model	28.60±1.88
	Padé Exp	28.21±1.69

Table 14: Sensitivity analysis of the covariance matrix Σ of ZFC on COVID-19 hospitalizations in CA and TX.

Covariance Matrix	**CA**	**TX**
$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	346.66±2.78	30.16±3.61
$\Sigma = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix}$	349.88±2.67	30.22±3.02
$\Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$	350.94±1.95	32.50±3.47
$\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$	352.76±2.95	32.83±2.83
$\Sigma = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$	347.88±2.89	31.32±2.55
$\Sigma = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$	**343.92±3.66**	**27.81±1.72**

time series makes it questionable how **robust** these methods and the conclusions derived from them are, especially under adverse scenarios. This **negative** impact of the developed predictive analytics shall be necessarily accounted in the proper risk management, particularly, when targeted attacks are expected, and shall be further investigated with rigours inferential tools of mathematical statistics.