Simplicial Vector Autoregressive Models

Krishnan, Joshin; Money, Rohan; Beferull-Lozano, Baltasar; Isufi, Elvin

DOI
10.1109/TSP.2024.3503063

Publication date
2024

Document Version
Final published version

Published in
IEEE Transactions on Signal Processing

Simplicial Vector Autoregressive Models

Joshin Krishnan, Rohan Money, Baltasar Beferull-Lozano, *Senior Member, IEEE*, and Elvin Isufi, *Senior Member, IEEE*

***Abstract*—The vector autoregressive (VAR) model is extensively employed for modelling dynamic processes, yet its scalability is challenged by an overwhelming growth in parameters when dealing with several hundred time series. To overcome this issue, data relations can be leveraged as inductive priors to tackle the curse of dimensionality while still effectively modelling the time series. In this paper, we study the role of simplicial complexes as inductive biases when modelling time series defined on higher-order network structures such as edges and triangles. First, we propose two simplicial VAR models: one that models time series defined on a single simplicial level, such as edge flows, and another that jointly models multiple time series defined across different simplicial levels, ultimately capturing their spatio-temporal interdependencies. The proposed models use simplicial convolutional filters to facilitate parameter sharing and capture structure-aware spatio-temporal dependencies in a multiresolution manner. Second, we develop a joint simplicial-temporal Fourier transform to study the spectral characteristics of the models, depicting them as simplicial-temporal filters. Third, targeting streaming signals, we develop an online algorithm for learning simplicial VAR models. We prove this online learner attains a sublinear dynamic regret bound, ensuring convergence under reasonable assumptions. Finally, we corroborate the proposed approach through experiments on synthetic networks, water distribution networks, and collaborating agents. Our findings show that the proposed models attain competitive signal modelling accuracy with orders of magnitude fewer parameters than the state-of-the-art alternatives.**

***Index Terms*—Simplicial convolution, simplicial complex, simplicial vector autoregressive model, Hodge Laplacians.**

I. Introduction

THE proliferation of applications across social, biological, financial, water, power, communication, and transportation networks motivates models for the dynamic evolution of processes associated to them [2], [3], [4], [5], [6]. For example, in a water network where vertices and edges represent respectively junctions and pipes, we may be interested in modelling the water demand based on past observations and the underlying water distribution system [7]. Among the successful and tractable approaches, the vector autoregressive model (VAR) and its variations are prominent [8], [9], [10], [11], [12], [13], [14], [15], [16], [17]. A VAR model characterizes the spatio-temporal dependencies among the time-varying processes as a linear combination of their past realizations.

Standard VAR models disregard any inherent structure in the data. Consequently, they suffer the curse of dimensionality, particularly when dealing with tens or hundreds of time series. Several approaches have been proposed to overcome this challenge, including factor models [18], shrinkage estimators [19], and low-rank data representations [20]. While these approaches reduce the data demand, they neglect the underlying network structure of the data and, consequently, do not grasp the inductive biases inherent in them [21]. Given that the network affecting the process is typically sparse, any model ignoring this structure fails to capitalize on the sparsity of interactions between the time series. Furthermore, when a new time series is added to the system or when the underlying topology changes slightly, we must retrain the model completely. However, this is not the case for network-based VAR models such as the graph VAR model (G-VAR) [14], which accounts for the structure and captures the sparsity in the data via the so-called graph convolutional filters [22]. The G-VAR considers the time series as processes over the vertices of a graph and models their evolution as a sparse linear combination of the time series in the adjacent vertices. The graph convolutional filter enables parameter sharing across vertices, facilitating its transferability. Furthermore, the computational complexity of a G-VAR model is linear in the number of time series, while that of standard approaches is quadratic [8], [11], [18], [19], [20].

The G-VAR overcomes the challenges of standard VAR models, but it models only time series on the nodes of a graph. However, real-world networks often contain time series on various higher-order structures, including edges (pair of nodes), triplets, and beyond [23], [24] that influence each other in a non-trivial manner. For instance, in a water network, the water flow in the pipes can be better modelled as a process evolving over the edges of the network, whereas the pressure is a process over the nodes. The mutual influence between these edge and vertex signals underscores the need for a model that can capture their dependencies based on the network structure.

Received 1 May 2024; revised 9 October 2024; accepted 9 November 2024. Date of publication 20 November 2024; date of current version 10 December 2024. This work was supported in part by the IKTPLUSS DISCO under Grant 338740, in part by the TU Delft AI Labs Programme, in part by the NWO OTP GraSPA proposal under Grant 19497, and in part by the NWO Project VENI under Grant 19052. An earlier version of this paper was presented in part at the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) [DOI: 10.1109/ICASSP49357.2023.10096095]. The associate editor coordinating the review of this article and approving it for publication was Prof. Che Lin. *(Corresponding author: Rohan Money.)*

Joshin Krishnan, Rohan Money, and Baltasar Beferull-Lozano are with SIGIPRO Department, Simula Metropolitan Center for Digital Engineering, 0167 Oslo, Norway (e-mail: joshin@simula.no; rohan@simula.no; baltasar@simula.no).

Elvin Isufi is with the Intelligent Systems Department, Delft University of Technology, 2628 XE Delft, The Netherlands (e-mail: e.isufi-1@tudelft.nl).

Digital Object Identifier 10.1109/TSP.2024.3503063

Recent literature has proposed methodologies to process data on higher-order topologies accounting for their underlying structure via simplicial complexes [23], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34]. A simplicial complex serves as a mathematical representation of the higher-order connectivities within networks, offering mathematical tractability reminiscent of graphs employed for pairwise similarities [35]. Subsequently, [36] extends signal processing concepts, such as the Fourier transform and sampling, to the simplex by drawing analogies with graph signal processing (GSP), and [37] introduces simplicial convolutional filters. However, all these approaches are tailored to process *time-invariant* data.

The overarching contribution of this paper is to develop a simplicial-aware VAR model for time series defined over higher-order networks. Leveraging the Hodge Laplacian representation of simplicial complexes and simplicial convolution filters, the proposed model efficiently captures temporal interactions among signals across different higher-order network structures, while attaining orders of magnitude less parameters than conventional VAR models. Our model assumes that the true topology and corresponding Hodge Laplacians are given, and its novelty lies in exploiting this known topology as an inductive bias. In addition, we propose an online learning strategy aimed at updating the time-varying parameters of the proposed model from streaming time series. We organize the contributions of this paper into four components:

1) We propose a simplicial complex VAR (SC-VAR) model to capture sparse spatio-temporal dependencies in time-varying processes. The SC-VAR model captures dependencies between signals defined on distinct simplicial levels, such as expressing the process at an edge as a linear combination of processes at adjacent edges, vertices, and triangles. A particular instance of the SC-VAR is the simplicial VAR (S-VAR) model [1] that focuses on modelling time series in a single simplicial level.
2) We show that the SC-VAR model is simplicially local, shares parameters across simplices, and is equivariant to both vertex permutations and edge orientations. These properties play a central role in addressing the curse of dimensionality and enabling seamless transfer of the learned model to arbitrary simplices without re-training, thereby giving our model a clear advantage over the standard VAR model.
3) We present an online algorithm for learning time-varying SC-VAR model parameters from streaming time series, featuring convergence guarantees. Our theoretical analysis establishes an upper bound for the dynamic regret, attaining a sublinear dynamic regret under reasonable assumptions suitable for real-world applications.
4) We introduce a joint simplicial-temporal Fourier transform by extending the recent advances of simplicial Fourier transform [36] and analyze the SC-VAR model in the spectral domain to investigate the learned frequency responses of the model. Unlike the analogous G-VAR analysis [14], we show a coupled two-dimensional frequency response between different types of simplicial frequencies.

The rest of this paper is organized as follows. Section II introduces two families of simplicial VAR models and discusses their simplicial properties. Section III formulates the parameter learning strategy. Section IV offers an in-depth spectral analysis of the models. Numerical results are presented in Section V[1], and the paper concludes in Section VI.

II. VAR Models in the Simplex

A VAR model expresses a multivariate time-varying process $\mathbf{x}_t \in \mathbb{R}^N$ as a linear combination of its P preceding realizations $\mathbf{x}_{t-1}, \ldots, \mathbf{x}_{t-P}$:

$$\mathbf{x}_t = \sum_{p=1}^{P} \mathbf{W}_p \mathbf{x}_{t-p} + \boldsymbol{\varepsilon}_t, \tag{1}$$

where $\mathbf{W}_p \in \mathbb{R}^{N\times N}$ encompasses the parameters for the temporal lag p, in which $[\mathbf{W}_p]_{i,j}$ denotes the dependency between $[\mathbf{x}_{t-p}]_j$ and $[\mathbf{x}_t]_i$. The variable $\boldsymbol{\varepsilon}_t$ represents the noise, assumed to be a temporally white zero-mean Gaussian process with covariance $\sigma^2\mathbf{I}_N$. For future reference, we write model (1) as:

$$\begin{aligned} \mathbf{x}_t &:= \Phi_{\text{VAR}}(\mathbf{x}_{t-1}, \ldots, \mathbf{x}_{t-P}; \mathcal{H}_{\text{VAR}}), \\ \mathcal{H}_{\text{VAR}} &= \{\mathbf{W}_1, \ldots, \mathbf{W}_P\}, \end{aligned} \tag{2}$$

where set $\mathcal{H}_{\text{VAR}}$ collects all the N^2P parameters. The VAR model has been widely used in a variety of disciplines, including statistical signal processing [38], finance [39], and neuroscience [40]. However, it suffers from training instabilities when the number of time series N becomes comparable with the number of training data [38]. Additionally, for network-based time series, the introduction or removal of a node requires a complete model retraining, along with a change in the dimension of the parameter matrix, ultimately lacking inductive capabilities. To address these limitations, we seek for inductive biases from the data structure to effectively reduce the degrees of freedom of the VAR model and aid learning [14]. Next, we show how this can be achieved for data with a simplicial structure [35].

A. Simplicial Complex and Signals

We are interested in processing time-varying signals whose underlying support is represented by a simplicial complex. Let $\mathcal{V} = \{1, \ldots, N\}$ be a set of vertices. A $k-$simplex $\mathcal{S}^k$ is a subset of $\mathcal{V}$ containing $k+1$ distinct elements. In a Euclidean embedding, simplices of orders 0, 1, and 2, can be depicted as nodes, edges, and triangles, respectively. A simplicial complex (SC) of order K, denoted as $\mathcal{X}^K$, comprises a set of k-simplices $\mathcal{S}^k$, where $k = 0, 1, \ldots, K$. It adheres to an inclusivity criterion stating that a simplex $\mathcal{S}^k$ belongs to $\mathcal{X}^K$ iff all of its subsets are also constituents of $\mathcal{X}^K$ [35]. Two simplices of different orders $\mathcal{S}_i^{k-1}$ and $\mathcal{S}_j^k$ are *vertical adjacent* (or *vertical neighbours*) if $\mathcal{S}_i^{k-1} \subset \mathcal{S}_j^k$. And two simplices of the same order $\mathcal{S}_i^k$ and $\mathcal{S}_j^k$ are *horizontal adjacent* (or *horizontal neighbours*) if they have in common a vertical neighbour simplex $\mathcal{S}_l^{k-1}$ or $\mathcal{S}_l^{k+1}$. To ease exposition, we focus on SC of order up to $K = 2$, which

[1]Codes are publicly available at https://github.com/rohantmoney/Simplicial-vector-autoregressive-model

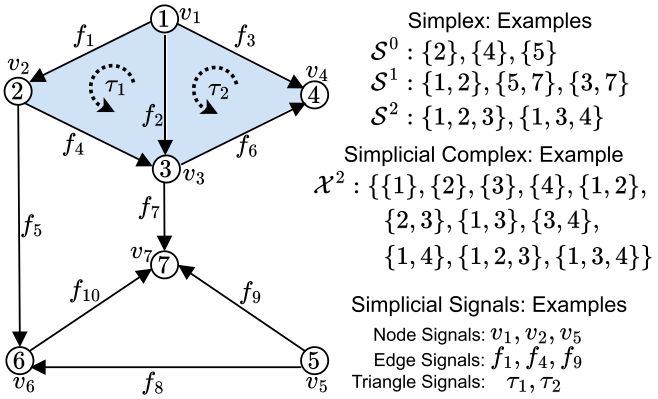


Figure 1. A simplicial complex and associated simplicial signals.

includes nodes, edges, and triangles. An example is provided in Figure 1, where the node $\{3\}$ and the edge $\{1,3\}$ are vertical neighbours since $\{3\} \subset \{1,3\}$, and the triangles $\{1,2,3\}$ and $\{1,3,4\}$ are horizontal neighbours with the common edge $\{1,3\}$.

Adjacencies in a SC can be represented via the incidence matrices and Hodge Laplacians [35]. Let N_k be the number of $k-$simplices in a SC. The incidence matrix $\mathbf{B}_k \in \{-1,0,1\}^{N_{k-1} \times N_k}$ employs $(k-1)-$ simplices as the row indices and $k-$simplices as the column indices, capturing the adjacencies between them. For instance, the matrix $\mathbf{B}_1$ represents the incidence from nodes to edges, and $\mathbf{B}_2$ represents the incidence from edges to triangles. These incidence matrices adhere to the boundary condition $\mathbf{B}_1\mathbf{B}_2 = \mathbf{0}$, implying that there is no interaction among simplices of different orders that are not vertical neighbours. The structure of a SC is described by Hodge Laplacians, which are high-order combinatorial Laplacian matrices crafted from the incidence matrices. Hodge Laplacians representing the structure of $\mathcal{X}^2$ are given by:

$$\begin{aligned} \mathbf{L}_0 &= \mathbf{B}_1\mathbf{B}_1^\top, \\ \mathbf{L}_1 &= \mathbf{L}_{1,d} + \mathbf{L}_{1,u} := \mathbf{B}_1^\top\mathbf{B}_1 + \mathbf{B}_2\mathbf{B}_2^\top, \\ \mathbf{L}_2 &= \mathbf{B}_2^\top\mathbf{B}_2. \end{aligned} \tag{3}$$

Here, the 0^{th} Hodge Laplacian $\mathbf{L}_0$ denotes the well-known graph Laplacian [41] that encapsulates horizontal adjacencies between vertices based on common edges. The 1^{st} Hodge Laplacian $\mathbf{L}_1$ expresses horizontal adjacencies between edges based on two vertical adjacencies: $i)$ based on common vertices via the lower-Laplacian $\mathbf{L}_{1,d} = \mathbf{B}_1^\top\mathbf{B}_1$ and $ii)$ based on common triangles via the upper-Laplacian $\mathbf{L}_{1,u} = \mathbf{B}_2\mathbf{B}_2^\top$. The 2^{nd} Hodge Laplacian $\mathbf{L}_2$ expresses proximities between triangles via lower edge adjacencies. Drawing parallels with the graph signal processing literature [22], the Hodge Laplacian emerges as a natural choice for a shift operator for signals defined on a simplicial complex [26], [36].

Simplicial signals. Simplicial signals are functions from any $k-$ simplices to the set of real numbers. For example, when considering the $1-$simplices (edges), the edge flow $\mathbf{x}_t^1 := \mathbf{f}_t = [f_{1,t}, \ldots, f_{N_1,t}]^\top \in \mathbb{R}^{N_1}$ can be defined, with $f_{e,t}$ denoting the flow of the edge e in $\mathcal{S}^1$ at time t. Similarly, we define the vertex signals over $\mathcal{S}^0$ as $\mathbf{x}_t^0 := \mathbf{v}_t \in \mathbb{R}^{N_0}$ and triangle signal over $\mathcal{S}^2$ as $\mathbf{x}_t^2 := \boldsymbol{\tau}_t \in \mathbb{R}^{N_2}$; see Figure 1. We also define simplicial complex signals encompassing vertex, edge, and triangle signals as $\mathbf{x}_t = [\mathbf{x}_t^{0\top}, \mathbf{x}_t^{1\top}, \mathbf{x}_t^{2\top}]^\top \in \mathbb{R}^{N_0+N_1+N_2}$. As it follows from (3), proximities in a SC are translated into proximities between data points defined over the different simplices. The objective therefore is to exploit such proximities to process signals $\mathbf{v}_t, \mathbf{f}_t$, and $\boldsymbol{\tau}_t$ [36].

B. Simplicial VAR (S-VAR)

The S-VAR model of order P for a $k-$simplicial time series is defined as

$$\mathbf{x}_t^k = \sum_{p=1}^{P} \mathbf{H}_p^k(\mathbf{L}_k)\mathbf{x}_{t-p}^k + \boldsymbol{\varepsilon}_t, \quad \text{for } k = 0,1,2, \tag{4}$$

where $\{\mathbf{H}_p^k(\mathbf{L}_k)\}_{p=1}^P$ are simplicial convolutional filters [1] defined by:

$$\mathbf{H}_p^k(\mathbf{L}_k) = \begin{cases} \sum_{n=0}^{L-1} \beta_{p,n}\mathbf{L}_k^n, & \text{for } k = 0,2, \\ \underbrace{\sum_{n=0}^{L_d-1} \beta_{p,n}^d \mathbf{L}_{1,d}^n}_{\mathbf{H}_p^{1,d}(\mathbf{L}_{1,d})} + \underbrace{\sum_{n=0}^{L-L_d-1} \beta_{p,n}^u \mathbf{L}_{1,u}^n}_{\mathbf{H}_p^{1,u}(\mathbf{L}_{1,u})}, & \text{for } k = 1. \end{cases} \tag{5}$$

Here L is the convolutional filter order, and $\{\beta_{p,n}\}_{n=0}^{L-1}$ are the filter coefficients weighting $\mathbf{x}_{t-p}^k$ shifted n times by $\mathbf{L}_k$. For simplicity, we have omitted the time index t for the filter coefficients, even though they may vary over time. In $\mathcal{X}^2$, the convolution operations for the vertex $(k=0)$ and triangle $(k=2)$ signals involve simplicial shifting operations using $\mathbf{L}_0$ and $\mathbf{L}_2$, respectively. However, the convolution of edge signals $(k=1)$ involves simplicial shifting by both lower and upper components of the first-order Hodge Laplacian [cf. (5)] as shown in Figure 2.

Upon grouping all filters defining (4) in set $\mathcal{H}_{\text{S-VAR}}^k = \{\mathbf{H}_1^k(\mathbf{L}_k), \ldots, \mathbf{H}_P^k(\mathbf{L}_k)\}$, the S-VAR model can be seen as the particular case of (2) (resp. (1)), i.e.,

$$\mathbf{x}_t^k = \Phi_{\text{S-VAR}}(\mathbf{x}_{t-1}^k, \ldots, \mathbf{x}_{t-P}^k; \mathcal{H}_{\text{S-VAR}}^k). \tag{6}$$

Here, the convolution filter $\mathbf{H}_p^k(\mathbf{L}_k)$ captures the spatial interactions among the components of the simplicial signal $\mathbf{x}_{t-p}^k$ that are within a radius $L-1$ by leveraging the adjacencies defined by the Hodge Laplacian $\mathbf{L}_k$, whereas the VAR regression captures the temporal interactions among the time-lagged signals $\mathbf{x}_{t-1}^k, \cdots, \mathbf{x}_{t-P}^k$. Model (4) uses the simplicial structure as the inductive bias by leveraging the simplicial convolution filters. With this inductive bias, the model gains parameter-sharing capabilities, allowing it to exploit structure-aware proximities within the network and achieve a notably low parameter count. The S-VAR model for a $k-$simplicial signal involves only LP parameters, irrespective of the number of $k-$simplicials N_k. This positions S-VAR as distinctly advantageous in real-world networks with a few hundred or more time series, as compared to the standard VAR (1) with $N_k^2 P$ parameters. Furthermore, S-VAR model effortlessly transfers the learned model across

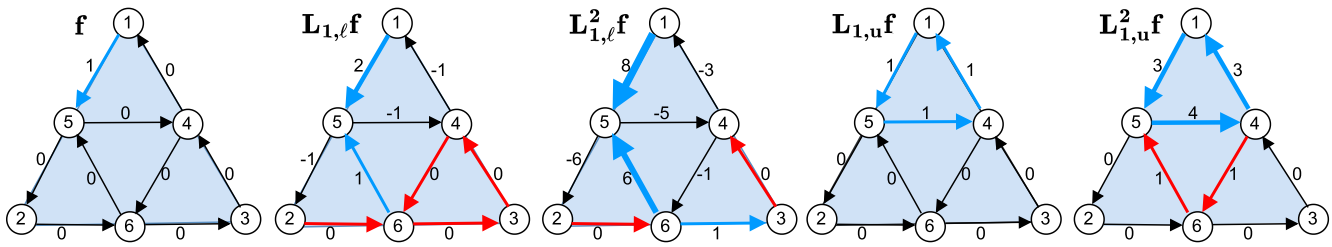


Figure 2. Simplicial shifting of an indicator flow, with directed arrows indicating arbitrarily edge orientations. The colours blue, red, and black respectively indicate high, intermediate, and low flow values, distinguished also by edge thickness variations.

close-by simplices (i.e., deploying a fixed model on another simplex), obviating the need for model retraining when adding (or removing) a time series to the network.

C. Simplicial Complex VAR (SC-VAR)

The S-VAR model (4) neglects the vertical adjacencies, thereby overlooking the dependencies across distinct simplicial levels, which may pose drawbacks in real-world networks. For instance, in water networks, the hydraulic pressure at the nodes (junctions) influences flows in the pipes. When such information is available, exploring it may further improve our understanding of SC signal inter-dependencies as well as the model performance, e.g., in terms of prediction or parameter learning. This vertical adjacency is represented by the incidence matrices [cf. (3)] and leads to different types of signal couplings [37]:

• **Divergence flow.** The incidence matrix $\mathbf{B}_1$ captures node-to-edge proximities. Applying such a matrix to an edge flow $\mathbf{f}_t$ leads to the vertex signal:

$$\mathbf{v}_{\text{div},t} = \mathbf{B}_1\mathbf{f}_t, \tag{7}$$

in which the n^{th} entry is the net flow through vertex n; i.e., the difference between the total inflow and outflow. A nonzero divergence indicates that a node is generating or absorbing flows, which corresponds to a source or sink node, respectively.

• **Gradient flow.** By applying the transpose matrix $\mathbf{B}_1^\top$ to a vertex flow $\mathbf{v}_t$, we obtain the so-called gradient flow:

$$\mathbf{f}_{\text{grd},t} = \mathbf{B}_1^\top\mathbf{v}_t, \tag{8}$$

i.e., the difference of node signals at the extremes of an edge.

• **Curl flow.** Matrix $\mathbf{B}_2$ represents edge-to-triangle proximities, and applying it to a triangle flow $\boldsymbol{\tau}_t$ yields the curl flow:

$$\mathbf{f}_{\text{crl},t} = \mathbf{B}_2\boldsymbol{\tau}_t. \tag{9}$$

Such a flow corresponds to an edge signal induced by the local cycles of the triangles.

• **Cyclic flow.** Finally, we can induce triangle flows from edge flows as:

$$\boldsymbol{\tau}_{\text{cyc},t} = \mathbf{B}_2^\top\mathbf{f}_t, \tag{10}$$

which are flows circulating along the edges of triangles.

By leveraging these vertical couplings, we can process a $k-$simplicial signal $\mathbf{x}_{t-p}^k$ by first filtering it with a filter $\mathbf{H}_p(\mathbf{L}_k)$ of the form (5), then transforming it into its adjacent simplex via the incidence matrix as in (7)-(10), and finally filtering the transformed signal via another filter $\mathbf{G}_p(\mathbf{L}_{k+1})$ or $\mathbf{G}_p(\mathbf{L}_{k-1})$ of the form (5), e.g., $\mathbf{G}_p(\mathbf{L}_{k+1})\mathbf{B}_k^\top\mathbf{H}_p(\mathbf{L}_k)\mathbf{x}_{t-p}^k$. Following, this *convolve-transform-convolve* principle, we define a SC-VAR model of order P as:

$$\begin{aligned}
\mathbf{v}_t &= \sum_{p=1}^{P}\mathbf{H}_p^{00}(\mathbf{L}_0)\mathbf{v}_{t-p} + \mathbf{G}_p^{01}(\mathbf{L}_0)\mathbf{B}_1\mathbf{H}_p^{01}(\mathbf{L}_1)\mathbf{f}_{t-p} + \boldsymbol{\varepsilon}_{vt}\\
\mathbf{f}_t &= \sum_{p=1}^{P}\mathbf{G}_p^{10}(\mathbf{L}_1)\mathbf{B}_1^\top\mathbf{H}_p^{10}(\mathbf{L}_0)\mathbf{v}_{t-p} + \mathbf{H}_p^{11}(\mathbf{L}_1)\mathbf{f}_{t-p}\\
&\quad + \mathbf{G}_p^{12}(\mathbf{L}_1)\mathbf{B}_2\mathbf{H}_p^{12}(\mathbf{L}_2)\boldsymbol{\tau}_{t-p} + \boldsymbol{\varepsilon}_{ft}\\
\boldsymbol{\tau}_t &= \sum_{p=1}^{P}\mathbf{G}_p^{21}(\mathbf{L}_2)\mathbf{B}_2^\top\mathbf{H}_p^{21}(\mathbf{L}_1)\mathbf{f}_{t-p} + \mathbf{H}_p^{22}(\mathbf{L}_2)\boldsymbol{\tau}_{t-p} + \boldsymbol{\varepsilon}_{\tau t}
\end{aligned} \tag{11}$$

The SC-VAR expression in (11) consists of three $k-$process equations, each for $k = 0, 1,$ and, 2, delineating the processes related to vertices, edges, and triangles. The indices m and n of filters $\mathbf{G}_p^{mn}(\cdot)$ and $\mathbf{H}_p^{mn}(\cdot)$ represent the simplicial levels after and before the convolve-transform-convolve operations, i.e., we are processing a $n-$simplicial signal on the $m-$process equation. When processing a $k-$signal on the $k-$process equation, the *transform* operation is not required, and hence such terms in (11) do not require the post-transform filters $\mathbf{G}_p^{kk}(\cdot)$.

Upon collecting in vector $\mathbf{x}_t = [\mathbf{v}_t^\top, \mathbf{f}_t^\top, \boldsymbol{\tau}_t^\top]^\top \in \mathbb{R}^{N_0+N_1+N_2}$, the simplicial processes at time t, and defining the block filtering matrix:

We can write (11) in the compact form:

$$\mathbf{x}_t = \sum_{p=1}^{P}\boldsymbol{\mathcal{H}}_p(\boldsymbol{\mathcal{L}})\mathbf{x}_{t-p} + \boldsymbol{\varepsilon}_t, \tag{13}$$

where $\boldsymbol{\mathcal{L}} = \{\mathbf{L}_0, \mathbf{L}_1, \mathbf{L}_2\}$ stands for the Hodge Laplacians defining the $2-$SC and $\boldsymbol{\varepsilon}_t = [\boldsymbol{\varepsilon}_{vt}^\top, \boldsymbol{\varepsilon}_{ft}^\top, \boldsymbol{\varepsilon}_{\tau t}^\top]^\top$.

$$\boldsymbol{\mathcal{H}}_p(\boldsymbol{\mathcal{L}}) = \begin{pmatrix} \mathbf{H}_p^{00}(\mathbf{L}_0), & \mathbf{G}_p^{01}(\mathbf{L}_0)\mathbf{B}_1\mathbf{H}_p^{01}(\mathbf{L}_1), & \mathbf{0} \\ \mathbf{G}_p^{10}(\mathbf{L}_1)\mathbf{B}_1^\top\mathbf{H}_p^{10}(\mathbf{L}_0), & \mathbf{H}_p^{11}(\mathbf{L}_1), & \mathbf{G}_p^{12}(\mathbf{L}_1)\mathbf{B}_2\mathbf{H}_p^{12}(\mathbf{L}_2) \\ \mathbf{0}, & \mathbf{G}_p^{21}(\mathbf{L}_2)\mathbf{B}_2^\top\mathbf{H}_p^{21}(\mathbf{L}_1), & \mathbf{H}_p^{22}(\mathbf{L}_2) \end{pmatrix}, \tag{12}$$

By grouping further all filters defining (11) in $\mathcal{H}_{\text{SC-VAR}} = \{\boldsymbol{\mathcal{H}}_1(\boldsymbol{\mathcal{L}}), \ldots, \boldsymbol{\mathcal{H}}_P(\boldsymbol{\mathcal{L}})\}$, the SC-VAR model is seen as a particular case of (2) (resp. (1)), i.e.,

$$\mathbf{x}_t = \Phi_{\text{SC-VAR}}(\mathbf{x}_{t-1}, \cdots, \mathbf{x}_{t-P}; \mathcal{H}_{\text{SC-VAR}}). \tag{14}$$

The SC-VAR model accommodates the spatio-temporal memory of simplicial signals in three different ways: it is temporal up to P past realizations, simplex horizontal due to the convolutional operation, and SC vertical through the different transformation operations. These intertwined SC memories allow the SC-VAR to capture signal dependencies via the different convolve-transform-convolve blocks. The S-VAR model (4) can be derived from (11) by deactivating the off-diagonal filter blocks in (12), leading us to primarily concentrate on the SC-VAR model in the ensuing discussions.

The SC-VAR model benefits from the simplicial inductive bias, allowing for parameter sharing and model transferability to close-by simplicial complexes, similar to the S-VAR model. It has $11LP$ parameters, and notably, the parameter count is independent of the process dimensions N_0, N_1, and N_2. In large-scale networks, the advantage of the SC-VAR model becomes apparent over the standard VAR model, which requires $(N_0 + N_1 + N_2)^2 P$ parameters. This reduction in parameter count stems from the fact that SC-VAR models signal dependencies in a fundamentally different way than standard VAR. Instead of considering all pair-wise interactions between signals, as in a standard VAR, SC-VAR captures interactions among neighboring simplices in the simplicial complex through simplicial convolution operations. In real-world networks, signals often interact more strongly with other signals within their topological neighbourhood. The simplicial convolution operation captures these localized, network-aware interactions, allowing the model to maintain a low parameter count. Our modelling assumes that the true topology, and consequently the Hodge Laplacians, are provided. The main novelty of our model lies in its ability to leverage the known topology as an inductive bias. While recovering the unknown topology from the data by estimating the Hodge Laplacian matrices would be an interesting pursuit, it is beyond the scope of the proposed framework.

Instead of the proposed SC-VAR model, an alternative is to employ three distinct S-VAR models for each simplicial level, resulting in a reduced parameter count of $3LP$ but disregarding dependencies among vertical neighbours. Denoting by $N_{\max} = \max\{N_0, N_1, N_2\}$ the maximum number of simplices involved and exploiting the sparsities in $\mathbf{B}_1, \mathbf{B}_2, \mathbf{L}_k$ as in [26], we can compute the outputs of the SC-VAR model with a cost of order at most $\mathcal{O}(LPN_{\max})$, which is substantially smaller than the cost associated to the standard VAR model $\mathcal{O}(P(N_0 + N_1 + N_2)^2)$.

D. Simplicial Properties

We now study how the SC-based inductive biases contribute to the equivariance properties [42] of simplicial-based models and examine their impact on the vertical locality of the SC-VAR model.

Proposition 1 (Equivariance Property): The SC-VAR model (11) exhibits equivariance under permutations of vertex labels and reorientations of edge flow directions.

Proof: See Appendix C. □

The equivariance property signifies that it accommodates arbitrary vertex labelling and edge flow orientations without being influenced by different choices of arbitrariness. That is, if we were to relabel the vertices (or reorient the flows) in a simplicial complex and then apply the SC-VAR, the resulting model output would be a relabeled (or reoriented) version of the output obtained by applying the SC-VAR model before the modifications. These findings generalize the concept of equivariances seen for Euclidean and graph data [43] to the simplex. In fact, they are the main reason why the parameter sharing works in the SC-VAR models as it allows them to use the simplices and the simplicial complexes as inductive biases for identifying patterns in the data.

Proposition 2 (Vertical locality): Consider a simplicial complex process $\mathbf{x}_t = [\mathbf{v}_t^\top, \mathbf{f}_t^\top, \boldsymbol{\tau}_t^\top]^\top$ composed of the vertex, edge, and triangle processes whose evolution is modelled via the SC-VAR model (11). Then, any simplicial process is influenced only by its one-hop vertical adjacent simplicial processes.

Proof: See Appendix D. □

The consequence of **Proposition** 2 is that, in a SC-VAR model, a vertex process is not influenced by triangle flows, implying that any modelling error in $\mathbf{v}_t$ is not propagated into the triangle flow model and vice-versa. It also implies, for instance, that the performance of vertex signal modelling can be improved by relying on edge signals but not on triangle signals. This vertical locality has further interesting consequences when analyzing the SC-VAR model from a simplicial spectral perspective, detailed in Section IV.

III. Parameter Estimation

Learning model parameters faces a substantial challenge due to the time-varying nature of real-world systems. In such instances, traditional offline learning strategies prove inadequate, prompting the exploration of online parameter updates as new data samples appear. We first formulate an offline learning strategy, which we subsequently adapt into an online algorithm.

To begin, we rephrase the SC-VAR model in (11) into a more compact form. Consider a representative convolve-transform-convolve term $\mathbf{G}_p^{01}(\mathbf{L}_0)\mathbf{B}_1\mathbf{H}_p^{01}(\mathbf{L}_1)\mathbf{f}_{t-p}$ in (11). Following Appendix A, this term can be reduced to a compact form:

$$\mathbf{G}_p^{01}(\mathbf{L}_0)\mathbf{B}_1\mathbf{H}_p^{01}(\mathbf{L}_1)\mathbf{f}_{t-p} = \widetilde{\mathbf{V}}_{t-p}^{01}\boldsymbol{\beta}_p^{01}, \tag{15}$$

where $\widetilde{\mathbf{V}}_{t-p}^{01} \in \mathbb{R}^{N_0 \times L^2}$ collects the features in its columns. These features are obtained via three steps: $i)$ applying simplicial shift operations on $\mathbf{f}_{t-p}$ using the Hodge Laplacian $\mathbf{L}_1$ to get L distinct simplicial shifted signals, $ii)$ projecting each shifted signal onto the vertex domain using (7), and $iii)$ performing again simplicial shift operations on each projected signal using the Hodge Laplacian $\mathbf{L}_0$. The vector $\boldsymbol{\beta}_p^{01} \in \mathbb{R}^{L^2}$ collects the filter coefficients of the convolve-transform-convolve operations. The sum over time lags 1 to P in (11)

can be written as:

$$\sum_{p=1}^{P} \mathbf{G}_p^{01}(\mathbf{L}_0)\mathbf{B}_1\mathbf{H}_p^{01}(\mathbf{L}_1)\mathbf{f}_{t-p} = \sum_{p=1}^{P} \widetilde{\mathbf{V}}_{t-p}^{01}\boldsymbol{\beta}_p^{01} = \mathbf{V}_t^{01}\boldsymbol{\beta}^{01}, \tag{16}$$

where $\mathbf{V}_t^{01} = [\widetilde{\mathbf{V}}_{t-1}^{01}, \ldots, \widetilde{\mathbf{V}}_{t-P}^{01}] \in \mathbb{R}^{N_0 \times L^2 P}$ and $\boldsymbol{\beta}^{01} = \left[\boldsymbol{\beta}_1^{01\top}, \ldots, \boldsymbol{\beta}_P^{01\top}\right]^\top \in \mathbb{R}^{L^2 P}$. Following the same procedure for all the other terms in (11), the SC-VAR can be expressed as:

$$\begin{aligned}
\mathbf{v}_t &= \mathbf{V}_t^{00}\boldsymbol{\beta}^{00} + \mathbf{F}_t^{01}\boldsymbol{\beta}^{01} + \boldsymbol{\varepsilon}_{vt} = \mathbf{S}_t^0\boldsymbol{\beta}^0 + \boldsymbol{\varepsilon}_{vt} \\
\mathbf{f}_t &= \mathbf{V}_t^{10}\boldsymbol{\beta}^{10} + \mathbf{F}_t^{11}\boldsymbol{\beta}^{11} + \mathbf{T}_t^{12}\boldsymbol{\beta}^{12} + \boldsymbol{\varepsilon}_{ft} = \mathbf{S}_t^1\boldsymbol{\beta}^1 + \boldsymbol{\varepsilon}_{ft} \\
\boldsymbol{\tau}_t &= \mathbf{F}_t^{21}\boldsymbol{\beta}^{21} + \mathbf{T}_t^{22}\boldsymbol{\beta}^{22} + \boldsymbol{\varepsilon}_{\tau t}^2 = \mathbf{S}_t^2\boldsymbol{\beta}^2 + \boldsymbol{\varepsilon}_{\tau t},
\end{aligned} \tag{17}$$

where

$$\begin{aligned}
&\mathbf{S}_t^0 = [\mathbf{V}_t^{00}, \mathbf{F}_t^{01}], \mathbf{S}_t^1 = [\mathbf{V}_t^{10}, \mathbf{F}_t^{11}, \mathbf{T}_t^{12}], \mathbf{S}_t^2 = [\mathbf{F}_t^{21}, \mathbf{T}_t^{22}], \\
&\boldsymbol{\beta}^0 = \left[\boldsymbol{\beta}^{00\top}, \boldsymbol{\beta}^{01\top}\right]^\top, \boldsymbol{\beta}^1 = \left[\boldsymbol{\beta}^{10\top}, \boldsymbol{\beta}^{11\top}, \boldsymbol{\beta}^{12\top}\right]^\top, \\
&\boldsymbol{\beta}^2 = \left[\boldsymbol{\beta}^{21\top}, \boldsymbol{\beta}^{22\top}\right]^\top.
\end{aligned}$$

Offline (batch) estimation. An effective approach for estimating VAR parameters involves minimizing the empirical mean square error (MSE) combined with a regularization term that incorporates priors, such as Bayesian priors [44] or sparsity priors [8], [11], to reduce overfitting. We estimate SC-VAR parameters by minimizing the empirical MSE over T_0 samples using a set of regularizers derived using Hodge Laplacians in addition to the Tikhonov regularization:

$$\boldsymbol{\beta}^{k*} = \arg\min_{\boldsymbol{\beta}^k} \frac{1}{2} \sum_{t=P+1}^{T_0} h_t(\boldsymbol{\beta}^k) + \omega(\boldsymbol{\beta}^k), \quad \text{for } k = 0, 1, 2, \tag{18}$$

where

$$h_t(\boldsymbol{\beta}^k) = \|\mathbf{x}_t^k - \mathbf{S}_t^k\boldsymbol{\beta}^k\|_2^2 + \|\mathbf{S}_t^k\boldsymbol{\beta}^k\|_{\mathbf{R}_k}^2, \tag{19}$$

$$\omega(\boldsymbol{\beta}^k) = \lambda\|\boldsymbol{\beta}^k\|_2^2. \tag{20}$$

In (18), $\omega(\boldsymbol{\beta}^k)$ is a regularizer with a hyperparameter $\lambda \geq 0$. The operation $\|\cdot\|_{\mathbf{R}_k}$ denotes the Mahalanobis distance given by $\|\mathbf{y}\|_{\mathbf{R}_k} = \sqrt{\mathbf{y}^\top\mathbf{R}_k\mathbf{y}}$, where we compose the matrix $\mathbf{R}_k$ using appropriate Hodge Laplacians as:

$$\mathbf{R}_k = \begin{cases} \mu_0\mathbf{L}_0 & \text{for } k = 0, \\ \mu_{1,d}\mathbf{L}_{1,d} + \mu_{1,u}\mathbf{L}_{1,u}, & \text{for } k = 1, \\ \mu_2\mathbf{L}_2, & \text{for } k = 2. \end{cases} \tag{21}$$

Here, the regularization terms involving $\mathbf{L}_0$, $\mathbf{L}_{1,d}$, $\mathbf{L}_{1,u}$, and $\mathbf{L}_2$ are respectively deployed to regulate the gradient, divergence, cyclic, and curl flows [cf. (3), (7)-(10)] along with the non-negative hyperparameters μ_0, $\mu_{1,d}$, $\mu_{1,u}$, and μ_2.

The optimization problem (18) is convex with solution:

$$\boldsymbol{\beta}^{k*} = \left(\sum_{t=P+1}^{T_0} \mathbf{S}_t^{k\top}(\mathbf{I} + \mathbf{R}_k)\mathbf{S}_t^k + \lambda\mathbf{I}\right)^{-1} \sum_{t=P+1}^{T_0} \mathbf{S}_t^{k\top}\mathbf{x}_t^k. \tag{22}$$

Online estimation. Note that (18) is an offline (batch) strategy, meaning that the entire batch of data $\mathbf{x}_t^k, t = 1, \ldots, T_0$ is required to compute the solution. In a data streaming setting, we do not have access to all the data and need to adapt our algorithm parameters on the fly. Offline learning produces time-invariant parameter estimates, whereas online learning is capable of estimating parameters that evolve over time. We can develop an online strategy by replacing the cumulative loss $\sum_{t=P+1}^{T_0} h_t(\boldsymbol{\beta}^k)$ with a running average loss using an exponential window [8]:

$$\ell_t(\boldsymbol{\beta}^k) = \frac{1}{2}\delta \sum_{t'=P+1}^{t} \gamma^{t-t'} h_{t'}(\boldsymbol{\beta}^k). \tag{23}$$

Here, $\gamma \in (0, 1)$ is the forgetting factor of the window, and $\delta = 1 - \gamma$ is set to normalize the exponential weighting window. Note that (23) resembles the typical loss in a recursive least square (RLS) problem. We expand (23) using (19) as

$$\ell_t(\boldsymbol{\beta}^k) = \frac{1}{2}\delta \sum_{t'=P}^{t-1} \gamma^{t-t'}\|\mathbf{x}_{t'}^k\|_2^2 + \frac{1}{2}\boldsymbol{\beta}^{k\top}\boldsymbol{\Phi}_t^k\boldsymbol{\beta}^k - \mathbf{r}_t^{k\top}\boldsymbol{\beta}^k, \tag{24}$$

where

$$\boldsymbol{\Phi}_t^k = \delta \sum_{t'=P+1}^{t} \gamma^{t-t'} \mathbf{S}_{t'}^{k\top}(\mathbf{I} + \mathbf{R}_k)\mathbf{S}_{t'}^k, \tag{25}$$

$$\mathbf{r}_t^k = \delta \sum_{t'=P+1}^{t} \gamma^{t-t'} \mathbf{S}_{t'}^{k\top}\mathbf{x}_{t'}^k. \tag{26}$$

In more lenient terms, we refer to variables $\boldsymbol{\Phi}_t^k$ and $\mathbf{r}_t^k$ as the regularized weighted sample autocorrelation matrix and weighted sample cross-correlation matrix. Then, the SC-VAR parameters can be estimated online by solving the convex optimization problem:

$$\boldsymbol{\beta}_t^{k*} = \arg\min_{\boldsymbol{\beta}^k} \ell_t(\boldsymbol{\beta}^k) + \omega(\boldsymbol{\beta}^k). \tag{27}$$

The inclusion of the time index t in the SC-VAR parameters indicates that they may evolve over time. To solve (27) online, we use the stochastic gradient descent (SGD), resulting in the online recursive update formula

$$\widehat{\boldsymbol{\beta}}_{t+1}^k = \widehat{\boldsymbol{\beta}}_t^k - \eta_t(\boldsymbol{\Phi}_t^k\widehat{\boldsymbol{\beta}}_t^k - \mathbf{r}_t^k + \lambda\widehat{\boldsymbol{\beta}}_t^k). \tag{28}$$

In an online setting, $\boldsymbol{\Phi}_t^k$ and $\boldsymbol{r}_t^k$ in (28) can be updated recursively as

$$\boldsymbol{\Phi}_t^k = \gamma\boldsymbol{\Phi}_{t-1}^k + \delta\mathbf{S}_t^{k\top}(\mathbf{I} + \mathbf{R}_k)\mathbf{S}_t^k, \tag{29}$$

$$\mathbf{r}_t^k = \gamma\mathbf{r}_{t-1}^k + \delta\mathbf{S}_t^{k\top}\mathbf{x}_t^k, \tag{30}$$

which follow from (25) and (26).

Number of parameters and complexity. Consider an SC-VAR model described by (17), with convolutional filters of order L and $k-$simplicial process having dimension N_k for $k = 0, 1, 2$. Its total number of parameters is $P(4L^2 + 3L)$, while a standard VAR model [cf. (1)] requires N^2P parameters, where $N = N_0 + N_1 + N_2$. In large real-world networks with N in the order of hundreds, the SC-VAR model with the choice $L \ll N$, enjoys a distinct advantage in parameter

count over standard VAR models. This choice is reasonable since we are often interested in capturing the local interactions in the networks using $L-$hop convolution filters, where the condition $L \ll N$ holds true. The computational complexity of the algorithm is mainly contributed by (28) and (29), and for a $k-$simplex SC-VAR model, it is of order $\mathcal{O}\big(L^2P^2 + N_k^2LP\big)$.

A. Dynamic Regret Analysis

We analyze the performance of the SC-VAR online learning algorithm using the dynamic regret [8], [11]. Our analysis is based on the following standard assumptions:

- ***A1***. Bounded simplicial signals: there exists a finite $B > 0$ such that $\left\{|[\mathbf{x}_t^k]_n|^2\right\}_{k,t,n} \leq B$.
- ***A2***. Bounded minimum and maximum eigenvalues of the autocorrelation matrix $\mathbf{\Phi}_t^k$ in (25) : there exist $\rho_{\min}, \rho_{\max} > 0$ such that $\Lambda_{\min}(\mathbf{\Phi}_t^k) \geq \rho_{\min}$ and $\Lambda_{\max}(\mathbf{\Phi}_t^k) < \rho_{\max} < \infty$, for all $t \geq P$, where $\Lambda_{\min}(\cdot)$ and $\Lambda_{\max}(\cdot)$ are the minimum and maximum eigenvalue operators.
- ***A3***. Bounded spectral norm of the feature matrix $\mathbf{S}_t^k$: there exists a finite $C > 0$ such that $\sqrt{\Lambda_{\max}(\mathbf{S}_t^{k\top}\mathbf{S}_t^k)} < C$.

A1 is reasonable as the signals from real-world applications are bounded. The lower bound on the minimum eigenvalue in ***A2*** always holds true since $\mathbf{\Phi}_t^k$ is a positive definite matrix, which can be easily verified from (25). The upper bound on the maximum eigenvalue in ***A2*** can be obtained by combining ***A1*** and the fact that the sum of eigenvalues of $\mathbf{\Phi}_t^k$ is equal to its trace. To justify ***A3***, recall that the feature matrix $\mathbf{S}_t^k$ collects simplicial signals resulting from convolve-transform-convolve operations in its columns. ***A3*** holds true since the maximum eigenvalue of $\mathbf{S}_t^{k\top}\mathbf{S}_t^k$ is bounded for simplicial complexes with a finite number of vertices and bounded simplicial signals.

Dynamic regret. Dynamic regret is the cumulative sum of the difference between the cost function evaluated at the estimated and the optimal values of the model parameters over all time instants. Let $c_t(\boldsymbol{\beta}_t^k) = \ell_t(\boldsymbol{\beta}_t^k) + \omega(\boldsymbol{\beta}_t^k)$ be the cost function of the optimization problem (27) at time t. Then, the dynamic regret is defined as:

$$R^k[T] \triangleq \sum_{t=P}^{T}\left[c_t(\hat{\boldsymbol{\beta}}_t^k) - c_t(\boldsymbol{\beta}_t^{k^*})\right], \tag{31}$$

where $\hat{\boldsymbol{\beta}}_t^k$ are the SGD parameter estimates obtained by (28), and $\boldsymbol{\beta}_t^{k^*}$ are the optimal parameters of the problem (27). A prevalent approach to bounding the dynamic regret involves considering the cumulative distance between two consecutive instantaneous optimal solutions, known as *path length*:

$$W^k[T] \triangleq \sum_{t=P+1}^{T}\left\|\boldsymbol{\beta}_t^{k^*} - \boldsymbol{\beta}_{t-1}^{k^*}\right\|_2. \tag{32}$$

Next, we derive an upper bound for the dynamic regret $R^k[T]$.

Theorem 1: Under assumptions ***A1*** – ***A3***, and with a constant step-size $\eta \in (0, 1/\rho_{\max}]$, if there exists a $\sigma > 0$ such that:

$$\left\|\boldsymbol{\beta}_t^{k^*} - \boldsymbol{\beta}_{t-1}^{k^*}\right\|_2 \leq \sigma, \text{ for all } t \geq P+1,$$

then the dynamic regret $R^k(T)$ is upper-bounded by:

$$R^k[T] \leq CB\sqrt{N_k}\left(1 + \frac{\lambda + \rho_{\max}}{\lambda + \rho_{\min}}\right)\left(\|\boldsymbol{\beta}_P^{k^*}\|_2 + W^k[T]\right). \tag{33}$$

Proof: See Appendix B. □

An online algorithm should aim for a sublinear dynamic regret, implying that $R^k[T]/T \to 0$ as $T \to \infty$. From **Theorem 1**, the dynamic regret of the SC-VAR learning algorithm is sublinear if the path length $W^k[T]$ is sublinear, which holds true for slowly varying systems, thereby assuring the convergence guarantees. Additionally, the upper bound in (33) becomes tighter as C [cf. ***A3***] decreases and the values of $\rho_{\min}$ and $\rho_{\max}$ [cf. ***A2***] become closer. This condition is met when the eigenvalues of the feature matrix $\mathbf{S}_t^k$ are both small and approach each other.

IV. Simplex-Temporal Filtering

VAR models enjoy a spectral interpretation as a temporal multidimensional filter [38]. When data exhibit spatial variability, as seen in the graph-VAR model (i.e., zero-simplex S-VAR) [14], leveraging a joint graph and temporal Fourier transform enables analyzing its behaviour as a joint graph-time filtering [45], [46]. We extend this concept to the simplicial domain by introducing a joint simplicial-temporal Fourier transform.

A. Joint Simplicial-Temporal Fourier Transform

Definition 1 (Simplicial-temporal Fourier transform (STFT)): Let $\mathbf{x}_t^k$ be a $k-$simplex process with matrix $\mathbf{X}^k = [\mathbf{x}_1^k, \ldots, \mathbf{x}_T^k] \in \mathbb{R}^{N_k \times T}$ collecting T realizations of this process. The STFT is defined as

$$\widetilde{\mathbf{X}}^k = \text{STFT}(\mathbf{X}^k) := \mathbf{U}_k^\top \mathbf{X}^k \mathbf{V}_T^*, \tag{34}$$

where $\mathbf{V}_T$ is the DFT matrix, symbol * denotes the complex conjugate operation, and $\mathbf{U}_k$ is the matrix collecting eigenvectors of the k^{th} Hodge Laplacian in its column.

To investigate **Definition 1**, we express the STFT operation in terms of the discrete Fourier transform (DFT) operator [47] and the simplicial Fourier transform operator (SFT) [36]:

$$\text{STFT}(\mathbf{X}^k) := \text{SFT}(\text{DFT}(\mathbf{X}^k)) = \text{DFT}(\text{SFT}(\mathbf{X}^k)). \tag{35}$$

The DFT consists of right-multiplying matrix $\mathbf{X}^k$ with the complex conjugate of the DFT matrix $\mathbf{V}_T$:

$$\text{DFT}(\mathbf{X}^k) = \mathbf{X}^k \mathbf{V}_T^*, \tag{36}$$

where $[\mathbf{V}_T^*]_{t,m} = \frac{1}{\sqrt{T}} e^{-j\omega_m(t-1)}$, with $j = \sqrt{-1}, \omega_m = \frac{2\pi(m-1)}{T}$, and $t, m = 1, \cdots, T$. The DFT is applied row-wise to each time series and captures each process variability across time. Likewise, we can take the eigendecomposition of the k^{th} Hodge Laplacian $\mathbf{L}_k = \mathbf{U}_k \mathbf{\Lambda}_k \mathbf{U}_k^\top$ with eigenvectors $\mathbf{U}_k = [\mathbf{u}_{k1}, \ldots, \mathbf{u}_{kN_k}]$ and eigenvalue matrix $\mathbf{\Lambda}_k = \text{diag}(\lambda_{k1}, \ldots, \lambda_{kN_k})$. The simplicial Fourier transform (SFT), applied individually to any simplicial signal $\mathbf{x}_t^k$, is given by [36]

$$\tilde{\mathbf{x}}_t^k = \text{SFT}(\mathbf{x}_t^k) = \mathbf{U}_k^\top \mathbf{x}_t^k, \text{ for all } t = 1, \ldots, T. \tag{37}$$

Vector $\tilde{\mathbf{x}}_t^k$ collects the SFT coefficients, and each of its entries captures the contribution of the respective eigenvector in $\mathbf{U}_k$ to represent signal $\mathbf{x}_t^k$. The inverse SFT is defined as $\mathbf{x}_t^k = \mathbf{U}_k \tilde{\mathbf{x}}_t^k$. We shall discuss in the rest of the section that these SFT coefficients are related to different signal variability over the simplex.

Definition 1 can be seen as the result of substituting (36) and (37) in (35).

Simplicial frequencies. As the eigenvalues of $\mathbf{L}_0 = \mathbf{U}_0 \mathbf{\Lambda}_0 \mathbf{U}_0^T$ carry the notion of frequency for graph signals, so do the eigenvalues of $\mathbf{L}_k = \mathbf{U}_k \mathbf{\Lambda}_k \mathbf{U}_k^T$ for a generic simplex $\mathcal{S}^k$. I.e., they can be ordered as $0 \leq \lambda_{k1} \leq \ldots \leq \lambda_{kN_k}$. But differently from the notion of frequency in the temporal and graph domain, in the simplex $(k \geq 1)$, we have three types of frequencies, each with its own characteristics. To understand these frequencies, we rely on the Hodge decomposition that states that any $k-$simplicial process space $\mathbb{R}^{N_k}$ can be decomposed into the three orthogonal subspaces:

$$\mathbb{R}^{N_k} = \text{im}(\mathbf{B}_k^\top) \oplus \text{im}(\mathbf{B}_{k+1}) \oplus \ker(\mathbf{L}_k) \tag{38}$$

where $\text{im}(\cdot)$ and $\ker(\cdot)$ are the image and kernel spaces of a matrix and $\oplus$ is the direct sum of vector spaces. Hence, any $k-$simplicial process $\mathbf{x}_t^k$ can be decomposed into three signals $\mathbf{x}_{d,t}^{k-1}$, $\mathbf{x}_{h,t}^k$, and $\mathbf{x}_{u,t}^{k+1}$ of orders $k-1$, k, and $k+1$, respectively, such that we can write $\mathbf{x}_t^k$ as the sum of three orthogonal components:

$$\mathbf{x}_t^k = \mathbf{B}_k^\top \mathbf{x}_{d,t}^{k-1} + \mathbf{B}_{k+1} \mathbf{x}_{u,t}^{k+1} + \mathbf{x}_{h,t}^k, \tag{39}$$

where, once again, we can see the role played by the incidence matrices $\mathbf{B}_k$ and $\mathbf{B}_{k+1}$ to couple vertical adjacent signals [cf. (7)-(10)]. Furthermore, the eigenvectors of $\mathbf{L}_k$ span $\mathbb{R}^{N_k}$ and there exists an inherent coupling among $\mathbf{U}_k$, the three subspaces in (39), and the SFT in (37) [36], [37]. For the edge signal space, such a coupling carries the following frequency interpretation:

- **Gradient frequencies.** Space $\text{im}(\mathbf{B}_1^\top)$ is known as the gradient space. It is spanned by the eigenvectors $\mathbf{U}_\text{g} \subseteq \mathbf{U}_k \in \mathbb{R}^{N_k \times N_g}$ of $\mathbf{L}_{1,d}$ associated to positive eigenvalues. For an eigenvector $\mathbf{u}_\text{g} \in \mathbf{U}_\text{g}$, the corresponding eigenvalue $\lambda_\text{g} = \mathbf{u}_\text{g}^\top \mathbf{L}_{1,d} \mathbf{u}_\text{g} = \|\mathbf{B}_1 \mathbf{u}_\text{g}\|_2^2$ is the ℓ_2-norm of the divergence of $\mathbf{u}_\text{g}$ [cf. (7)]. Thus, eigenvectors $\mathbf{u}_\text{g}$ corresponding to a large eigenvalue λ_g have a large divergence. Consequently, for an edge flow $\mathbf{f}_t$ with a large SFT projection on such eigenvectors, we say that it has a high gradient frequency. We refer to eigenvalues λ_g as gradient frequencies and collect them in set $\mathcal{Q}_\text{g} = \{\lambda_{\text{g}1}, \ldots . \lambda_{\text{g}\,N_g}\}$.
- **Curl frequencies.** Space $\text{im}(\mathbf{B}_2)$ is known as the curl space. It is spanned by the eigenvectors $\mathbf{U}_\text{c} \subseteq \mathbf{U}_k \in \mathbb{R}^{N_k \times N_c}$ of $\mathbf{L}_{1,u}$ associated to positive eigenvalues. For an eigenvector $\mathbf{u}_\text{c} \in \mathbf{U}_\text{c}$, the corresponding eigenvalue $\lambda_\text{c} = \mathbf{u}_\text{c}^\top \mathbf{L}_1 \mathbf{u}_\text{c} = \|\mathbf{B}_2^\top \mathbf{u}_\text{c}\|_2^2$ is the ℓ_2-norm of the curl of $\mathbf{u}_\text{g}$ [cf. (10)]. Thus, eigenvectors $\mathbf{u}_\text{c}$ corresponding to a large eigenvalue λ_c have a large curl. We refer to eigenvalues λ_c as curl frequencies and collect them in set $\mathcal{Q}_\text{c} = \{\lambda_{\text{c}1}, \ldots . \lambda_{\text{c}\,N_c}\}$.
- **Harmonic frequencies.** Space $\ker(\mathbf{L}_k)$ is called the harmonic space. It is spanned by eigenvectors $\mathbf{U}_\text{h} \subseteq \mathbf{U}_k \in \mathbb{R}^{N_k \times N_h}$ that are the solution of $\mathbf{L}_1 \mathbf{u}_\text{h} = \mathbf{0}$. Hence, the corresponding eigenvalues are all zeros. We refer to $\lambda_\text{h} = 0$ as the harmonic frequencies and collect them in set $\mathcal{Q}_\text{h} = \{\lambda_{\text{h}1} = 0, \ldots . \lambda_{\text{h}N_h} = 0\}$.

With these notions of frequencies, the SFT of an edge process $\mathbf{f}_t$ [cf. (37)] can be seen as building three simplicial embeddings at time t: (i) the gradient embedding $\tilde{\mathbf{f}}_\text{g} = \mathbf{U}_\text{g}^\top \mathbf{f}$; (ii) the curl embedding $\tilde{\mathbf{f}}_\text{c} = \mathbf{U}_\text{c}^\top \mathbf{f}$; and (iii) the harmonic embedding $\tilde{\mathbf{f}}_\text{h} = \mathbf{U}_\text{h}^\top \mathbf{f}$. The n^th entry of the gradient embedding $[\tilde{\mathbf{f}}_\text{g}]_n$ is the SFT coefficient associated to the gradient frequency $\lambda_{\text{g}n}$, and similarly $[\tilde{\mathbf{f}}_\text{c}]_n$ is associated with the curl frequency $\lambda_{\text{c}n}$. For the harmonic embedding, each $[\tilde{\mathbf{f}}_\text{h}]_n$ entry is associated with the zero harmonic frequency $\lambda_{\text{h}n} = 0$. Simplicial frequencies allows us to dissect the two-dimensional STFT spectrum into its gradient, curl, and harmonic components, allowing analysis in each component.

B. Two-Dimensional Filtering

Leveraging the STFT, we show that the simplicial-based VAR models achieve a two-dimensional filtering. The subsequent frequency analyses are grounded in edge processes, as they represent the most general case in an order-2 SC.

S-VAR filtering. Consider the S-VAR edge process model:

$$\mathbf{f}_t - \sum_{p=1}^{P} \left(\sum_{k=0}^{L_d-1} \beta_{p,k}^d \mathbf{L}_{1,d}^k + \sum_{k=0}^{L-L_d-1} \beta_{p,k}^u \mathbf{L}_{1,u}^k \right) \mathbf{f}_{t-p} = \boldsymbol{\varepsilon}_t.$$

Upon applying the STFT to both sides, we obtain:

$$\left(1 - \sum_{p=1}^{P} h_p^1(\lambda_n) e^{-\jmath \omega_m p}\right) \tilde{f}_n(e^{\jmath \omega_m}) = \tilde{\varepsilon}_n(e^{\jmath \omega_m}), \tag{40}$$

where $\tilde{f}_n(e^{\jmath \omega_m})$ and $\tilde{\varepsilon}_n(e^{\jmath \omega_m})$ are the $n-$th entries of the STFT of $\mathbf{f}_t$ and $\boldsymbol{\varepsilon}_t$, respectively, and $h_p^1(\lambda_n) = h_p^{1,d}(\lambda_n) + h_p^{1,u}(\lambda_n)$ is the simplicial frequency response of the p^th filter $\mathbf{H}_p^1(\mathbf{L}_1)$ at frequency λ_n. The components of this frequency response are:

$$h_p^{1,d}(\lambda_n) := \sum_{k=0}^{L_d-1} \beta_{p,k}^d \lambda_n^k \text{ and } h_p^{1,u}(\lambda_n) := \sum_{k=0}^{L-L_d-1} \beta_{p,k}^u \lambda_n^k, \tag{41}$$

which correspond to the frequency responses of subfilters $\mathbf{H}_p^{1,d}(\mathbf{L}_{1,d})$ and $\mathbf{H}_p^{1,u}(\mathbf{L}_{1,u})$ [cf. (5)], respectively. Then, for parameters $\beta_{p,k}^d$ and $\beta_{p,k}^u$ that satisfy the stability condition

$$1 - \sum_{p=1}^{P} h_p^1(\lambda_n) e^{-\jmath \omega_m p} \neq 0, \tag{42}$$

the S-VAR model implements the two-dimensional simplicial-time frequency response:

$$h^1(\lambda_n, e^{\jmath \omega_m}) = \frac{1}{1 - \sum_{p=1}^{P} h_p^1(\lambda_n) e^{-\jmath \omega_m p}}, \tag{43}$$

which is a rational two-dimensional function in the frequency variables λ_n and ω_m. Since the simplicial frequencies carry a

different meaning, the frequency response (43) has the form

$$h^1(\lambda_n, e^{\jmath\omega_m}) = \begin{cases} \big(1+\sum_{p=1}^{P}(\beta_{p,0}^{d}+\beta_{p,0}^{u})e^{-\jmath\omega_m p}\big)^{-1} & \text{for } \lambda_n \in \mathcal{Q}_{\mathrm{h}}, \\ \big(1+\sum_{p=1}^{P}(h_p^{1,d}(\lambda_n)+\beta_{p,0}^{u})e^{-\jmath\omega_m p}\big)^{-1} & \text{for } \lambda_n \in \mathcal{Q}_{\mathrm{g}}, \\ \big(1+\sum_{p=1}^{P}(\beta_{p,0}^{d}+h_p^{1,u}(\lambda_n))e^{-\jmath\omega_m p}\big)^{-1} & \text{for } \lambda_n \in \mathcal{Q}_{\mathrm{c}}. \end{cases} \tag{44}$$

That is, we have a two-dimensional simplex-temporal frequency response, whose behaviour on the different types of simplicial frequencies is governed by a different set of parameters. We remark that the filter indeed cannot differentiate between signal components that lie within the subspace spanned by eigenvectors corresponding to an eigenvalue with multiplicity greater than one. For instance, it cannot provide distinct responses to multiple harmonic components.

SC-VAR filtering. The SC-VAR model possesses a greater challenge for frequency analysis compared to the S-VAR, due to vertical signal transformations between adjacent simplices in the convolve-transform-convolve operations. To ease exposition, we discuss the input-output frequency behaviour of a simplified SC-VAR model involving only one convolution operation[2]. This simplified transform-then-convolve version is given by:

$$\begin{aligned} \mathbf{f}_t &= \sum_{p=1}^{P} \mathbf{G}_p^{10}(\mathbf{L}_1)\mathbf{B}_1^{\top}\mathbf{v}_{t-p} + \mathbf{G}_p^{11}(\mathbf{L}_1)\mathbf{f}_{t-p} \\ &\quad + \mathbf{G}_p^{12}(\mathbf{L}_1)\mathbf{B}_2\boldsymbol{\tau}_{t-p} + \boldsymbol{\varepsilon}_{ft} \\ &= \sum_{p=1}^{P} \mathbf{G}_p^{10}(\mathbf{L}_1)\mathbf{f}_{t-p}^{\mathrm{grd}} + \mathbf{G}_p^{11}(\mathbf{L}_1)\mathbf{f}_{t-p} + \mathbf{G}_p^{12}(\mathbf{L}_1)\mathbf{f}_{t-p}^{\mathrm{crl}} + \boldsymbol{\varepsilon}_{ft} \end{aligned} \tag{45}$$

where in the second equality, we use the gradient transformation (8) to write $\mathbf{f}_{t-p}^{\mathrm{grd}} = \mathbf{B}_1^{\top}\mathbf{v}_{t-p}$ and the curl transformation (9) to write $\mathbf{f}_{t-p}^{\mathrm{crl}} = \mathbf{B}_2\boldsymbol{\tau}_{t-p}$. Expanding then the filters in (45) as $\mathbf{G}_p^{\mathrm{kl}}(\mathbf{L}_1) = \mathbf{G}_p^{\mathrm{kl},d}(\mathbf{L}_{1,d}) + \mathbf{G}_p^{\mathrm{kl},u}(\mathbf{L}_{1,u})$ similar to (5) and applying **Proposition** 2 leads to:

$$\begin{aligned} \mathbf{f}_t = \sum_{p=1}^{P} &\big(\mathbf{G}_p^{10,d}(\mathbf{L}_{1,d}) + \beta_{p,0}^{10,u}\big)\mathbf{f}_{t-p}^{\mathrm{grd}} + \mathbf{G}_p^{11}(\mathbf{L}_1)\mathbf{f}_{t-p} \\ &+ \big(\beta_{p,0}^{12,d} + \mathbf{G}_p^{12,u}(\mathbf{L}_{1,u})\big)\mathbf{f}_{t-p}^{\mathrm{crl}} + \boldsymbol{\varepsilon}_{ft} \end{aligned} \tag{46}$$

where $\beta_{p,i}^{\mathrm{kl},d}, i = 0, \ldots, L_d - 1$ are the coefficients of the lower-Laplacian filter $\mathbf{G}_p^{\mathrm{kl},d}(\mathbf{L}_{1,d})$ and $\beta_{p,i}^{\mathrm{kl},u}, i = 0, \ldots, L - L_d - 1$ are the coefficients of the upper-Laplacian filter $\mathbf{G}_p^{\mathrm{kl},u}(\mathbf{L}_{1,u})$. Applying then the STFT (35) on both sides of (46), the input-output relation in the spectral domain is:

Gradient frequencies ($\lambda_n \in \mathcal{Q}_g$):

$$\begin{aligned} \tilde{f}_n(e^{\jmath\omega_m}) = \sum_{p=1}^{P} \Big(&\big(g_p^{10,d}(\lambda_n) + \beta_{p,0}^{10,u}\big)\tilde{f}_n^{\mathrm{grd}}(e^{\jmath\omega_m}) \\ &+ g_p^{11}(\lambda_n)\tilde{f}_n(e^{\jmath\omega_t}) \Big) e^{\jmath\omega_m p} + \tilde{\varepsilon}_{fn}^{grd}(e^{\jmath\omega_m}); \end{aligned} \tag{47}$$

Curl frequencies ($\lambda_n \in \mathcal{Q}_c$):

$$\begin{aligned} \tilde{f}_n(e^{\jmath\omega_m}) = \sum_{p=1}^{P} \Big(&g_p^{11}(\lambda_n)\tilde{f}_n(e^{\jmath\omega_m}) \\ &+ \big(g_p^{12,u}(\lambda_n) + \beta_{p,0}^{12,d}\big)\tilde{f}_n^{\mathrm{crl}}(e^{\jmath\omega_m}) \Big) e^{\jmath\omega_m p} + \tilde{\varepsilon}_{fn}^{crl}(e^{\jmath\omega_m}); \end{aligned} \tag{48}$$

Harmonic frequencies ($\lambda_n \in \mathcal{Q}_h$):

$$\begin{aligned} \tilde{f}_n(e^{\jmath\omega_m}) = \sum_{p=1}^{P} \Big(&\big(\beta_{p,0}^{10,d} + \beta_{p,0}^{10,u}\big)\tilde{f}_n^{\mathrm{grd}}(e^{\jmath\omega_m}) \\ &+ \big(\beta_{p,0}^{11,d} + \beta_{p,0}^{11,u}\big)\tilde{f}_n(e^{\jmath\omega_m}) \\ &+ \big(\beta_{p,0}^{12,d} + \beta_{p,0}^{12,u}\big)\tilde{f}_n^{\mathrm{crl}}(e^{\jmath\omega_m}) \Big) e^{\jmath\omega_m p} + \tilde{\varepsilon}_{fn}^{h}(e^{\jmath\omega_m}); \end{aligned} \tag{49}$$

where $g_p^{11}(\lambda_n)$, $g_p^{10,d}(\lambda_n)$, and $g_p^{12,u}(\lambda_n)$ are the simplicial convolutional frequency responses of filters $\mathbf{G}_p^{11}(\mathbf{L}_1)$, $\mathbf{G}_p^{10,d}(\mathbf{L}_{1,d})$, and $\mathbf{G}_p^{12,u}(\mathbf{L}_{1,u})$, respectively. As it follows from (47)-(49), we have different types of input spectral processes, which, combined with the different simplicial filters, yield an edge flow. At the gradient frequencies, the STFT output spectra do not have any contributions from the curl component, and at the curl frequencies, they lack contributions from the gradient component. Similar to the S-VAR frequency response (44), we can see an intrinsic coupling of the frequency responses at different simplicial frequencies via the shared filter parameters. However, contrasting (47)-(49) with (44), we note the increased number of SC-VAR parameters yields a two-dimensional frequency response with higher degrees of freedom.

V. Numerical Results

This section conducts comprehensive numerical experiments to assess the performance of the proposed online SC-VAR algorithm. The model's efficacy is benchmarked against three online algorithms: TIRSO [8], RFNL-TIRSO [11], and the moving average algorithm. Both TIRSO and RFNL-TIRSO are state-of-the-art online topology estimation algorithms. TIRSO is constructed utilizing a standard linear VAR model represented by (1), requiring N^2P parameters, where N is the number of time series. In contrast, RFNL-TIRSO embraces a kernel-based nonlinear VAR model with random feature approximation, demanding $2PN^2D$ parameters, where D denotes the number of random features. Both of these algorithms present formidable competition for SC-VAR, as they take into consideration all pairwise interactions among time series with a high parameter count. We measure the performance of all models based on their prediction accuracy, measured by the normalized mean squared error (NMSE), which for a $k-$ simplicial signal is defined as:

$$\mathrm{NMSE}(t) = \frac{1}{N}\sum_{n=1}^{N} \frac{\sum_{t'=1}^{t}([\mathbf{x}_{t'+t_{step}}^{k}]_n - [\hat{\mathbf{x}}_{t'+t_{step}}^{k}]_n)^2}{\sum_{t'=1}^{t}[\mathbf{x}_{t'+t_{step}}^{k}]_n^2}, \tag{50}$$

where $[\hat{\mathbf{x}}_{t'+t_{step}}^{k}]_n$ is the n^{th} component of the $t_{step}-$ahead prediction at time t' and $[\mathbf{x}_{t'+t_{step}}^{k}]_n$ is the ground truth.

[2]Deriving a spectral input-output relation for the general SC-VAR model (11) presents a significant challenge due to filter blocks that include two convolution operations with distinct Hodge Laplacians.

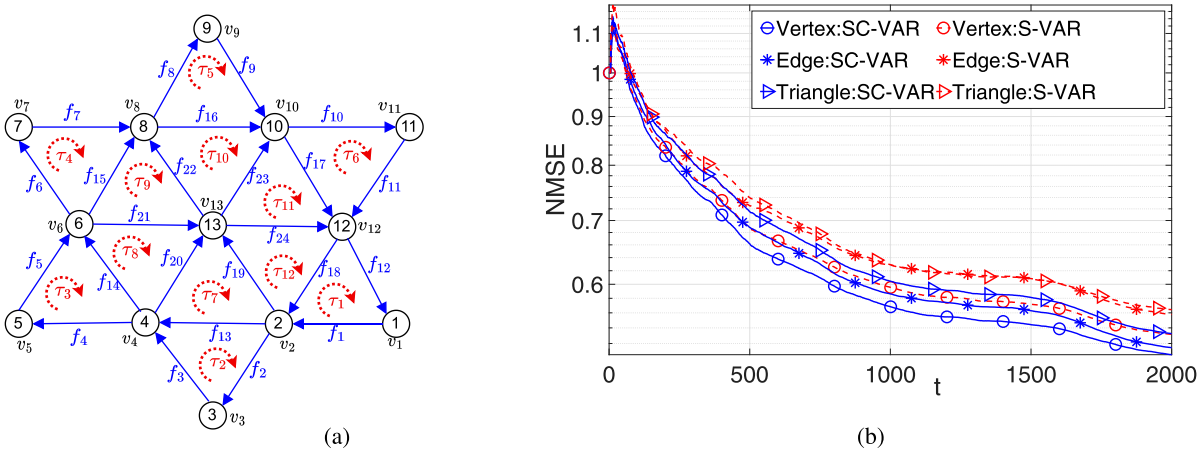


Figure 3. (a) Topology used to create the synthetic dataset. All the orientations are selected arbitrarily. (b) Comparative evaluation of NMSE (in log scale) for 1-step ahead prediction between S-VAR and SC-VAR.

A. Synthetic Data

We create a synthetic dataset that includes vertex, edge, and triangle signals with spatio-temporal interactions based on a specified topology (see, Figure 3(a)), having $N_0 = 13$ vertices, $N_1 = 24$ edges, and $N_2 = 12$ triangles. First, we generated three distinct VAR processes, $\mathbf{V}' \in \mathbb{R}^{N_0 \times T}$, $\mathbf{F}' \in \mathbb{R}^{N_1 \times T}$, and $\boldsymbol{\mathcal{T}}' \in \mathbb{R}^{N_2 \times T}$ using the standard VAR model (1) with zero-mean i.i.d. Gaussian noise and variance 0.1. The VAR coefficients are randomly sampled from $\mathcal{N}(0, 1)$, and to make the setting more dynamic, they are completely changed at every 400 time step. Subsequently, we modify these signals by inducing interactions among them based on the topology in Figure 3(a) using the following procedure:

$$\begin{aligned} \mathbf{V} &= \sin(\mathbf{V}') + \mathbf{B}_1 \mathbf{F}', \\ \mathbf{F} &= \mathbf{B}_1^\top \mathbf{V}' + \sin(\mathbf{F}') + \mathbf{B}_2 \boldsymbol{\mathcal{T}}', \\ \boldsymbol{\mathcal{T}} &= \mathbf{B}_2^\top \mathbf{F}' + \sin(\boldsymbol{\mathcal{T}}'). \end{aligned} \tag{51}$$

We generated a dataset comprising 110 realizations of $\mathbf{V}, \mathbf{F}, \text{and } \boldsymbol{\mathcal{T}}$ with $T = 2000$. A grid search, based on the lowest NMSE, is performed across all algorithms to optimize regularization hyperparameters using a validation subset of 10 randomly chosen realizations. The NMSE values reported in the experiments are averaged over the remaining 100 realizations. The hyperparameters for SC-VAR are $P = 4, L = 8, \lambda = 0.001, \text{and } \gamma = 0.98$. We assign $\mu_0 = \mu_{1,d} = \mu_{1,u} = \mu_2 = 0$ since the model (51) does not introduce any regulations to the gradient, divergence, curl, and cyclic components. Following the analysis in [8], we choose the step size $\eta_t = 1/\Lambda_{max}(\boldsymbol{\Phi}_t)$, where $\Lambda_{max}(\cdot)$ represents the maximum eigenvalue operation.

First, we compare the S-VAR and SC-VAR models through a $1-$step prediction. Figure 3(b) illustrates that the SC-VAR consistently outperforms the S-VAR in predicting all three signals. This is because the S-VAR model relies solely on information from a specific signal level, while the SC-VAR model incorporates information from all three signals.

In Figure 4(a)–4(c), we compare SC-VAR with the moving average, TIRSO, and RFNL-TIRSO algorithms for vertex, edge, and triangle signal. For TIRSO and RFNL-TIRSO, we combine all signals, resulting in a signal dimension of $N = N_0 + N_1 + N_2 = 49$. SC-VAR demonstrates highly competitive results with the best performer RFNL-TIRSO, despite having two orders of magnitude fewer parameters. The parameter counts of all the algorithms are shown in Figure 5. SC-VAR's superior performance is attributed to its effective utilization of topology information through the simplicial convolution operation, a distinctive feature absent in other algorithms.

Due to its minimal parameter usage, SC-VAR is also anticipated to exhibit outstanding performance with limited training data in an offline learning scenario. To demonstrate this, we present an offline experiment where we use the offline SC-VAR learning strategy given by (22). For this experiment, we use the offline version of the TIRSO algorithm and an LSTM-based algorithm [48] as benchmarks. The LSTM algorithm comprises 15 hidden states with a total of 3900 parameters. We use the Adam optimizer, 500 epochs, and a learning rate of 0.0004. Using varying batch sizes of $40, 60, \text{and } 80$, we train all models and perform $1-$ to $4-$step ahead prediction. From Table I, we deduce the following: i) performance of all algorithms exhibit a gradual degradation from $1-$ to $4-$step ahead predictions; ii) at larger batch sizes, all algorithms perform equally well with LSTM being slightly better than the others; iii) at small batch size, the SC-VAR algorithms outperforms the others. SC-VAR demonstrates a distinct advantage in performing well with limited training data, attributed to its low parameter count.

B. Water Networks

We present experiments with the Cherry Hills water network, consisting of 36 nodes (0-simplices), 40 pipes (1-simplices), and 2 triangles (2-simplices) [49]. Flow signals (in GPM) along the edges and pressure signals (in PSI) at the vertices constitute our dataset. Triangle signals were excluded from this experiment because the Cherryhills topology consists of just two triangles, without any substantial features associated with them.

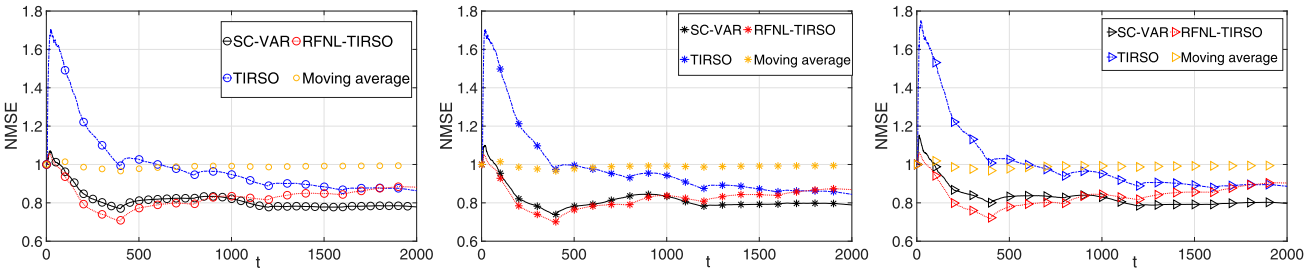


Figure 4. NMSE comparison based on 1-step ahead prediction: (a) for vertex, (b) for edge, and (c) for triangle signals.

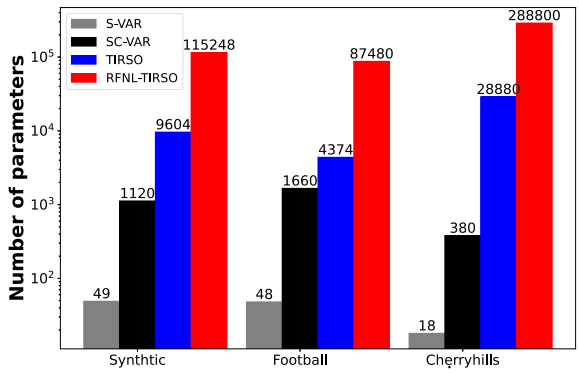


Figure 5. Parameter comparison of various algorithms (in log scale).

TABLE I
NMSE PERFORMANCE OF ALGORITHMS ACROSS DIFFERENT PREDICTION HORIZONS AND BATCH SIZES

Batch size	Signals	Algorithm	Prediction NMSE (Different prediction horizons) 1-step	2-step	3-step	4-step
40	Vertex	SC-VAR	0.80	0.82	0.84	0.86
		VAR	3.37	3.82	4.26	4.84
		LSTM	0.87	0.87	0.87	0.88
	Edge	SC-VAR	0.77	0.78	0.79	0.79
		VAR	3.36	3.79	4.20	4.82
		LSTM	0.86	0.86	0.86	0.87
	Triangle	SC-VAR	0.82	0.82	0.83	0.84
		VAR	1.32	3.61	4.55	5.21
		LSTM	0.87	0.87	0.87	0.89
60	Vertex	SC-VAR	0.81	0.83	0.85	0.87
		VAR	3.86	3.74	3.67	3.64
		LSTM	0.83	0.83	0.84	0.85
	Edge	SC-VAR	0.76	0.77	0.78	0.80
		VAR	3.91	3.74	3.69	3.68
		LSTM	0.83	0.83	0.84	0.85
	Triangle	SC-VAR	0.81	0.82	0.83	0.84
		VAR	3.93	3.76	3.70	3.68
		LSTM	0.84	0.84	0.84	0.85
80	Vertex	SC-VAR	0.79	0.81	0.83	0.86
		VAR	2.75	2.65	2.61	2.62
		LSTM	0.81	0.82	0.83	0.83
	Edge	SC-VAR	0.73	0.74	0.75	0.77
		VAR	2.76	2.65	2.63	2.62
		LSTM	0.80	0.80	0.82	0.83
	Triangle	SC-VAR	0.77	0.79	0.80	0.81
		VAR	2.82	2.69	2.67	2.67
		LSTM	0.82	0.82	0.82	0.84

Assuming a reference flow direction as in Figure 6(a), we generate time-evolving flow and pressure signals using the EPANET software with a demand-driven model such that the water flows meet the time-varying water demands at the nodes. Grounded in hydrological principles, we expect that the pressure signals will enhance the prediction of the flow signals, as pressure has a significant impact on flows. We use the initial 20% data samples to tune the hyperparameters (via grid search for the best NMSE), resulting in $P = 5, L = 4, \lambda = 0.0001, \mu_0 = 0.5, \mu_{1,d} = 0.5, \mu_{1,u} = \mu_2 = 0$, and $\gamma = 0.99$. A similar tuning of hyperparameters for all benchmark algorithms is also performed using the same data samples.

The NMSEs for $1-$step and $2-$step predictions of flow signals are plotted in Figure 6(b) and 6(c), respectively. The RFNL-TIRSO ($D = 5$) algorithm performs best, followed by SC-VAR, S-VAR, and TIRSO. SC-VAR surpasses S-VAR by leveraging additional information from the pressure signals, a capability that S-VAR lacks. This observation aligns with hydrological principles, highlighting the impact of pressure on flows. We underscore that SC-VAR achieves competitive performance compared to RFNL-TIRSO, with three orders of magnitude fewer parameters (see Figure 5).

C. Collaborative Agents

In collaborative environments such as SWARM robots, team sports, and wind farms, agent dynamics are mostly influenced by the local interactions with nearby agents. SC-VAR stands out as an effective model for capturing these dynamics, given its ability to incorporate proximity-aware sparse interactions. We demonstrate how SC-VAR can be used to predict the behaviour of collaborative agents by using a real football dataset [50]. The dataset includes the coordinates, speed, and pairwise distances between 10 players, sampled at 0.05 seconds intervals over 40 minutes. We calculate the mean coordinates of the players for the first minute and build a $2-$nearest neighbour topology with an arbitrary orientation, resulting in 10 vertices, 14 edges, and 3 triangles, as shown in Figure 7(a). The rationale for this choice is that the topology created from the initial match position reflects the teams' formation strategy. It is therefore likely that the players will frequently return to positions closer to their assigned formation strategy. The topology in this experiment differs from the previous two, as it is an abstract topology. One might also explore a dynamic topology during the game, but such an approach would entail time-varying Hodge Laplacians, which falls outside the scope of SC-VAR. The speeds of the players and the distance between connected players are

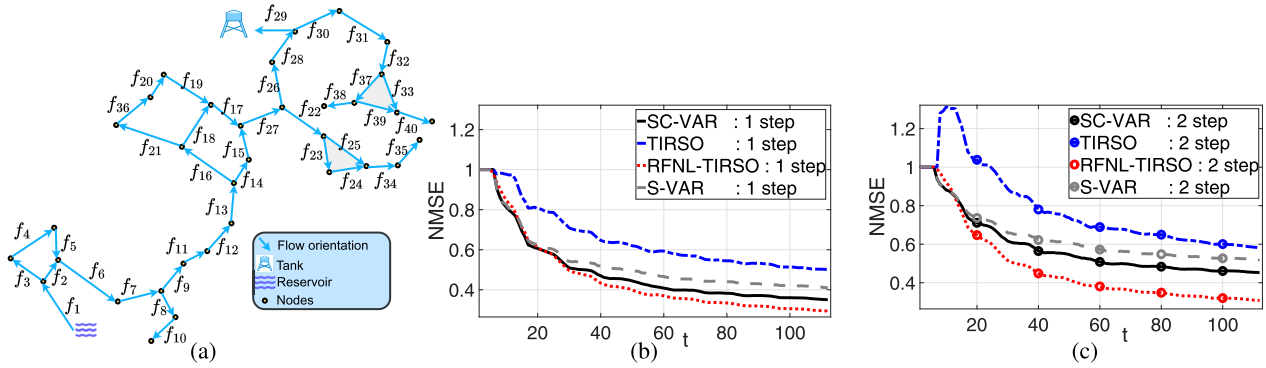


Figure 6. (a) Cherry Hills water network, (b) $1-$step and (c) $2-$step ahead prediction of water flows.

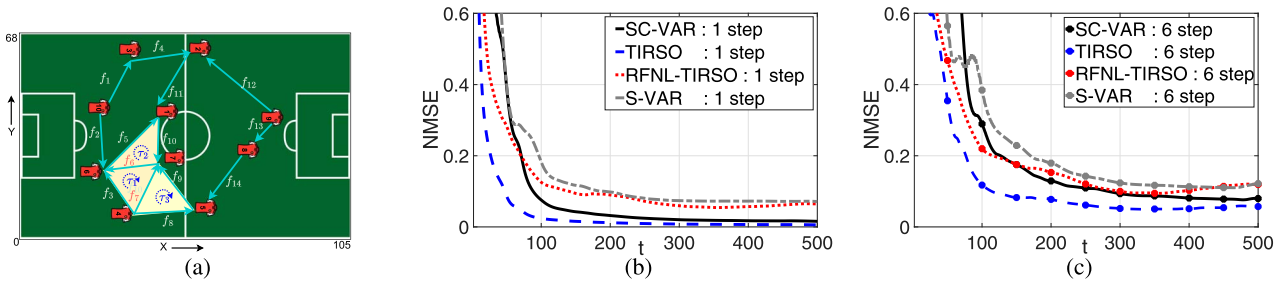


Figure 7. (a) Topology created with mean positions of football players. Comparative evaluation of NMSE for (b) 1-step ahead and (c) 6-step ahead prediction of player speed.

considered as the vertex and the edge signals, respectively. The triangle signals, defined by the areas enclosed by the triangles, represents how spread out the players are on the field. In this experiment, we use $P = 6$ for all algorithms and $L = 8$ for the simplicial models (chosen via grid search for the best NMSE). The initial 20% data samples are used to tune the hyperparameters, resulting in $\lambda = 0.01, \mu_0 = 0.01, \mu_{1,d} = \mu_{1,u} = \mu_2 = 0$, and$\gamma = 0.9$. A similar tuning of hyperparameters for all benchmark algorithms is also performed using the same data samples.

In Figure 7(b) and 7(c), we plot the NMSEs for $1-$step and $6-$step predictions of players' speeds. Despite having a markedly lower parameter count (see Figure 5), SC-VAR outperforms RFNL-TIRSO and achieves a prediction accuracy that is highly competitive, on par with the top-performing TIRSO algorithm. On the other hand, S-VAR trails behind other algorithms, highlighting the essential role of information from edges and triangles in predicting the speed. This experiment stands as a testament to the efficiency of the proposed SC-VAR model, adeptly capturing sparse higher-order interactions within collaborative environments and opening up opportunities for diverse real-world applications.

VI. Conclusion

We propose simplicial complex VAR models, which capture the intricate spatio-temporal dependencies among signals defined over higher-order network structures. Leveraging simplicial convolution filters, our models capitalize on the underlying simplicial structure as an inductive bias and feature parameter sharing across simplices to achieve a significant reduction in parameters relative to the standard VAR model. Targeting streaming signals from real-world time-varying networks, we develop an online learning algorithm featuring a sublinear dynamic regret bound. We further characterize the proposed methods in the spectral domain via a joint simplicial-temporal Fourier transform. Our experiments with both real and synthetic data, demonstrate that the proposed model achieves highly competitive performance compared to standard VAR-based models.

Appendix A

SC-VAR feature formulation strategy. Consider a representative convolve-transform-convolve term $\mathbf{G}_p^{01}(\mathbf{L}_0)\mathbf{B}_1\mathbf{H}_p^{01}(\mathbf{L}_1)\mathbf{f}_{t-p}$ in (11). The first convolution operation can be written as:

$$\mathbf{H}_p^{01}(\mathbf{L}_1)\mathbf{f}_{t-p} = \mathcal{L}_1\tilde{\mathbf{F}}_p\boldsymbol{\alpha}_p^{01}, \tag{52}$$

where

$$\mathcal{L}_1 = [\mathbf{L}_{1,d}^0, \dots \mathbf{L}_{1,d}^{(L_d-1)}, \mathbf{L}_{1,u}^0, \dots \mathbf{L}_{1,u}^{(L-L_d-1)}] \in \mathbb{R}^{N_1 \times LN_1},$$
$$\boldsymbol{\alpha}_p^{01} = [\alpha_{p,0}^{01,d}, \dots, \alpha_{p,(L_d-1)}^{01,d}, \alpha_{p,0}^{01,u}, \dots, \alpha_{p,(L-L_d-1)}^{01,u}]^\top \in \mathbb{R}^L,$$
$$\tilde{\mathbf{F}}_p = \mathbf{I}_L \otimes \mathbf{f}_{t-p} \in \mathbb{R}^{LN_1 \times L}.$$

Here, $\boldsymbol{\alpha}_p^{01}$ are the filter coefficients associated to $\mathbf{H}_p^{01}(\mathbf{L}_1)$, operation $\otimes$ denotes the Kronecker product, and $\tilde{\mathbf{F}}_p$ is a block diagonal matrix with $\mathbf{f}_{t-p}$ as the diagonal blocks, repeated L times. Using (52) and applying the same strategy for the second

convolution operation, the representative convolve-transform-convolve term can be reformulated as:

$$\mathbf{G}_p^{01}(\mathbf{L}_0)\mathbf{B}_1\mathbf{H}_p^{01}(\mathbf{L}_1)\mathbf{f}_{t-p} = \mathbf{G}_p^{01}(\mathbf{L}_0)\mathbf{B}_1\mathcal{L}_1\tilde{\mathbf{F}}_p\boldsymbol{\alpha}_p^{01} \tag{53}$$

$$= \mathcal{L}_0\left[\mathbf{I}_L \otimes (\mathbf{B}_1\mathcal{L}_1\tilde{\mathbf{F}}_p\boldsymbol{\alpha}_p^{01})\right]\boldsymbol{\delta}_p^{01} \tag{54}$$

$$= \mathcal{L}_0\left[\mathbf{I}_L \otimes (\mathbf{B}_1\mathcal{L}_1\tilde{\mathbf{F}}_p)\right](\mathbf{I}_L \otimes \boldsymbol{\alpha}_p^{01})\boldsymbol{\delta}_p^{01}, \tag{55}$$

where $\boldsymbol{\delta}_p^{01}$ is the filter coefficients associated with $\mathbf{G}_p^{01}(\mathbf{L}_0)$ and the equality (55) follows from the mixed-product property of the Kronecker product. In (55), $\mathbf{I}_L \otimes \boldsymbol{\alpha}_p^{01} \in \mathbb{R}^{L^2\times L}$ has L distinctive components, each one repeated L times, and $\boldsymbol{\delta}_p^{01} \in \mathbb{R}^{L\times 1}$ also has L distinctive components. Hence, the total number of variables to optimize is $2L$. Treating $\boldsymbol{\alpha}_p^{01}$ and $\boldsymbol{\delta}_p^{01}$ as optimization variables causes nonconvexity issues in the learning process since a pair-wise multiplication of the components of $\boldsymbol{\alpha}_p^{01}$ and $\boldsymbol{\delta}_p^{01}$ occurs. But we do not have to estimate $\boldsymbol{\alpha}_p^{01}$ and $\boldsymbol{\delta}_p^{01}$ separately, and hence, we define a new optimization parameter vector of length L^2:

$$\boldsymbol{\beta}_p^{01} := (\mathbf{I}_L \otimes \boldsymbol{\alpha}_p^{01})\boldsymbol{\delta}_p^{01} \in \mathbb{R}^{L^2} \tag{56}$$

Let

$$\widetilde{\mathbf{V}}_{t-p}^{01} := \mathcal{L}_0\left[\mathbf{I}_L \otimes (\mathbf{B}_1\mathcal{L}_1\tilde{\mathbf{F}}_p)\right] \in \mathbb{R}^{N_0\times L^2} \tag{57}$$

Substitute (56) and (57) in (55) to get

$$\mathbf{H}_p^{01}(\mathbf{L}_0)\mathbf{B}_1\mathbf{H}_p^{01}(\mathbf{L}_1)\mathbf{f}_{t-p} = \widetilde{\mathbf{V}}_{t-p}^{01}\boldsymbol{\beta}_p^{01} \tag{58}$$

We follow a similar procedure for all other convolve-transform-convolve terms in (11) to generate the feature matrices $\widetilde{\mathbf{V}}_{t-p}^{mn}$, $\widetilde{\mathbf{F}}_{t-p}^{mn}$, and $\widetilde{\mathbf{T}}_{t-p}^{mn}$, for $m, n \in \{0, 1, 2\}$.

APPENDIX B

Proof of Theorem 1. The dynamic regret can be expanded as:

$$R^k[T] \triangleq \sum_{t=P}^{T}\left[c_t(\hat{\boldsymbol{\beta}}_t^k) - c_t(\boldsymbol{\beta}_t^{k*})\right] \tag{59}$$

$$\leq \sum_{t=P}^{T}\nabla c_t(\hat{\boldsymbol{\beta}}_t^k)^\top(\hat{\boldsymbol{\beta}}_t^k - \boldsymbol{\beta}_t^{k*}) \tag{60}$$

$$\leq \sum_{t=P}^{T}\|\nabla c_t(\hat{\boldsymbol{\beta}}_t^k)\|_2\|\hat{\boldsymbol{\beta}}_t^k - \boldsymbol{\beta}_t^{k*}\|_2, \tag{61}$$

where ∇ denotes the gradient operation, (60) follows from the convexity of $c_t(\cdot)$, and (61) follows from the Cauchy Schwarz inequality. Next, we bound $\|\nabla c_t(\hat{\boldsymbol{\beta}}_t^k)\|_2$ in (61).

Bounding $\|\nabla c_t(\hat{\boldsymbol{\beta}}_t^k)\|_2$: Using the gradient in (28):

$$\|\nabla c_t(\hat{\boldsymbol{\beta}}_t^k)\|_2 = \|\boldsymbol{\Phi}_t^k\widehat{\boldsymbol{\beta}}_t^k - \mathbf{r}_t^k + \lambda\widehat{\boldsymbol{\beta}}_t^k\|_2 \tag{62}$$

$$\leq \|\boldsymbol{\Phi}_t^k\widehat{\boldsymbol{\beta}}_t^k\|_2 + \|\mathbf{r}_t^k\|_2 + \|\lambda\widehat{\boldsymbol{\beta}}_t^k\|_2 \tag{63}$$

$$\leq \rho_{\max}\|\widehat{\boldsymbol{\beta}}_t^k\|_2 + \|\mathbf{r}_t^k\|_2 + \lambda\|\widehat{\boldsymbol{\beta}}_t^k\|_2 \tag{64}$$

$$= (\rho_{\max} + \lambda)\|\widehat{\boldsymbol{\beta}}_t^k\|_2 + \|\mathbf{r}_t^k\|_2, \tag{65}$$

where (63) follows from the triangle inequality and (64) follows from the spectral norm properties and assumption ***A2***.

Bounding $\|\mathbf{r}_t^k\|_2$ in (65): Using (26),

$$\|\mathbf{r}_t^k\|_2 = \|\delta\sum_{t'=P+1}^{t}\gamma^{t-t'}\mathbf{S}_{t'}^{k\top}\mathbf{x}_{t'}^k\|_2 \tag{66}$$

$$\leq \delta\sum_{t'=P+1}^{t}\gamma^{t-t'}\|\mathbf{S}_{t'}^{k\top}\mathbf{x}_{t'}^k\|_2 \tag{67}$$

$$\leq \delta\sum_{t'=P+1}^{t}\gamma^{t-t'}\sqrt{\Lambda_{\max}(\mathbf{S}_{t'}^{k\top}\mathbf{S}_{t'}^k)}\|\mathbf{x}_{t'}^k\|_2 \tag{68}$$

$$\leq \delta\sum_{t'=P+1}^{t}\gamma^{t-t'}C\|\mathbf{x}_{t'}^k\|_2 \tag{69}$$

$$\leq \delta\sum_{t'=P+1}^{t}\gamma^{t-t'}CB\sqrt{N_k} \tag{70}$$

$$\leq CB\sqrt{N_k}(1-\gamma^{(t-P)}) \tag{71}$$

$$\leq CB\sqrt{N_k}, \tag{72}$$

where (67) follows from the triangle inequality, (68) follows from the spectral norm properties, (69) follows from assumption ***A3***, and (72) follows from the fact that $\gamma < 1$.

Bounding $\|\widehat{\boldsymbol{\beta}}_t\|_2$ in (65): Using (28),

$$\|\widehat{\boldsymbol{\beta}}_{t+1}^k\|_2 = \|\widehat{\boldsymbol{\beta}}_t^k - \eta_t(\boldsymbol{\Phi}_t^k\widehat{\boldsymbol{\beta}}_t^k - \mathbf{r}_t^k + \lambda\widehat{\boldsymbol{\beta}}_t^k)\|_2 \tag{73}$$

$$= \|(\mathbf{I} - \eta_t\lambda\mathbf{I} - \eta_t\boldsymbol{\Phi}_t^k)\widehat{\boldsymbol{\beta}}_t^k + \eta_t\mathbf{r}_t^k\|_2 \tag{74}$$

$$\leq \Lambda_{\max}(\mathbf{I} - \eta_t\lambda\mathbf{I} - \eta_t\boldsymbol{\Phi}_t^k)\|\widehat{\boldsymbol{\beta}}_t^k\|_2 + \eta_t\|\mathbf{r}_t^k\|_2 \tag{75}$$

$$\leq (1 - \eta_t\lambda - \eta_t\Lambda_{\min}(\boldsymbol{\Phi}_t^k))\|\widehat{\boldsymbol{\beta}}_t^k\|_2 + \eta_t\|\mathbf{r}_t^k\|_2 \tag{76}$$

$$\leq (1 - \eta_t\lambda - \eta_t\rho_{\min})\|\widehat{\boldsymbol{\beta}}_t^k\|_2 + \eta_t\|\mathbf{r}_t^k\|_2, \tag{77}$$

where (75) follows from the spectral norm properties and (77) follows from assumption ***A2***. Substituting (72) in (77), gives:

$$\|\widehat{\boldsymbol{\beta}}_{t+1}^k\|_2 \leq (1 - \eta_t\lambda - \eta_t\rho_{\min})\|\widehat{\boldsymbol{\beta}}_t^k\|_2 + \eta_t CB\sqrt{N_k} \tag{78}$$

Let $\zeta = 1 - \eta_t\lambda - \eta_t\rho_{\min}$ in (78). Perform $t - P + 1$ recursions of the expression (78), obtaining:

$$\|\widehat{\boldsymbol{\beta}}_{t+1}^k\|_2 \leq \zeta^{t-P+1}\|\widehat{\boldsymbol{\beta}}_P^k\|_2 + \eta_t CB\sqrt{N_k}\sum_{i=0}^{t-P}\zeta^i \tag{79}$$

$$= \eta_t CB\sqrt{N_k}\frac{1-\zeta^{t-P+1}}{1-\zeta} \tag{80}$$

$$\leq \frac{CB\sqrt{N_k}}{\lambda + \rho_{\min}}. \tag{81}$$

In (80), we assume that the filter coefficients are initialized with zeros, i.e., $\widehat{\boldsymbol{\beta}}_P = \mathbf{0}$. Also in (81), we assume that $\zeta \leq 1$. In order to guarantee this assumption in our implementation, we choose the value of the learning rate η_t to be

$$\eta_t \leq \frac{1}{\lambda + \rho_{\max}}, \tag{82}$$

such that the condition $\zeta \leq 1$ holds true (since $\rho_{\max} \geq \rho_{\min}$).

Now, substitute (72) and (81) in (65) to get

$$\|\nabla c_t(\hat{\boldsymbol{\beta}}_t^k)\|_2 \leq CB\sqrt{N_k}\left(1 + \frac{\lambda + \rho_{\max}}{\lambda + \rho_{\min}}\right). \tag{83}$$

Bounding $\|\hat{\boldsymbol{\beta}}_t^k - \boldsymbol{\beta}_t^{k^*}\|_2$ in (61): Loss function $c_t(.)$ is Lipschitz smooth and strong convex. Following the equation (E.71) in [51], it can be proved that

$$\sum_{t=P}^{T} \|\hat{\boldsymbol{\beta}}_t^k - \boldsymbol{\beta}_t^{k^*}\|_2 \leq \|\boldsymbol{\beta}_P^{k^*}\|_2 + W^k[T]. \tag{84}$$

Substitute (83) and (84) in (61) to upper bound the dynamic regret, leading to:

$$R^k[T] \leq CB\sqrt{N_k}\left(1 + \frac{\lambda + \rho_{\max}}{\lambda + \rho_{\min}}\right)\left(\|\boldsymbol{\beta}_P^{k^*}\|_2 + W^k[T]\right). \tag{85}$$

APPENDIX C

Equivariance property. Consider the following sets of permutation matrices $\mathcal{P}$ and orientation matrices $\mathcal{D}$.

$$\mathcal{P} = \{\mathbf{P}_k \in \{0,1\}^{N_k \times N_k} : \mathbf{P}_k\mathbf{1} = \mathbf{P}_k^\top\mathbf{1} = \mathbf{1}, k = 0,1,2\}, \tag{86}$$

$$\mathcal{D} = \{\mathbf{D}_k = \text{diag}(\mathbf{d}_k) : \mathbf{d}_k \in \{\pm 1\}^{N_k}, k = 1,2; \mathbf{d}_0 = \mathbf{1}\}. \tag{87}$$

Permutation matrices are such that products $\check{\mathbf{x}}^k = \mathbf{P}_k^\top\mathbf{x}^k$ are reorderings of the entries of $\mathbf{x}^k$ and that the permuted Hodge Laplacian $\check{\mathbf{L}}_k = \mathbf{P}_k^k\mathbf{L}_k\mathbf{P}_k$ is a reordering of the rows and columns of $\mathbf{L}_k$. Orientation matrices are such that products $\check{\mathbf{x}}^k = \mathbf{D}_k\mathbf{x}^k$ are reorientations of the flow directions in vectors $\mathbf{x}^k$ (edges or triangles flows) and that the reoriented Hodge Laplacians are $\check{\mathbf{L}}_k = \mathbf{D}_k\mathbf{L}_k\mathbf{D}_k$.

Permutation equivariance. Consider a simplicial $\mathcal{X}^2$ with Hodge Laplacians $\boldsymbol{\mathcal{L}} = \{\mathbf{L}_0, \mathbf{L}_1, \mathbf{L}_2\}$ [cf. (3)] and its permuted version with the respective Hodge Laplacians

$$\check{\boldsymbol{\mathcal{L}}} := \{\mathbf{P}_0^\top\mathbf{L}_0\mathbf{P}_0, \mathbf{P}_1^\top\mathbf{L}_1\mathbf{P}_1, \mathbf{P}_2^\top\mathbf{L}_2\mathbf{P}_2\}$$

for any permutation matrix $\mathbf{P} = \text{diag}(\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2)$ such that $\mathbf{P}_k \in \mathcal{P}$. Given also a set of parameters $\mathcal{H}_{\text{SC-VAR}}$. Then, for any pair of a simplicial complex signal $\mathbf{x}_t$ and its permuted version $\check{\mathbf{x}}_t := \mathbf{P}^\top\mathbf{x}_t$, the SC-VAR output satisfies

$$\begin{aligned}\check{\mathbf{x}}_t :&= \Phi_{\text{SC-VAR}}(\{\mathbf{P}^\top\mathbf{x}_{t-p}\}; \check{\boldsymbol{\mathcal{L}}}; \mathcal{H}_{\text{SC-VAR}}) \\ &= \mathbf{P}^\top\Phi_{\text{SC-VAR}}(\{\mathbf{x}_{t-p}\}; \boldsymbol{\mathcal{L}}; \mathcal{H}_{\text{SC-VAR}}).\end{aligned} \tag{88}$$

Proof: Recall the SC-VAR model from (11). We enhance the readability of the proof by assuming $P = 1$ and removing the index p without loss of generality. Consider a permuted system, where a $k-$simplex is permuted using a permutation matrix $\mathbf{P}_k \in \mathcal{P}$ [cf. (86)]. After permutation, we have the simplicial signals of the form $\check{\mathbf{x}}_t^k = \mathbf{P}_k^\top\mathbf{x}_t^k, k = 0,1,2$, with Hodge Laplacians $\mathbf{P}_k^\top\mathbf{L}_k\mathbf{P}_k, k = 0,1,2$ and incidence matrices $\mathbf{P}_m^\top\mathbf{B}_m\mathbf{P}_m, m = 1,2$. The vertex process equation of the permuted system can be written as:

$$\begin{aligned}\check{\mathbf{v}}_t = &\mathbf{H}^{00}(\mathbf{P}_0^\top\mathbf{L}_0\mathbf{P}_0)\mathbf{P}_0^\top\mathbf{v}_{t-p} \\ &+ \mathbf{G}^{01}(\mathbf{P}_0^\top\mathbf{L}_0\mathbf{P}_0)\mathbf{P}_0^\top\mathbf{B}_1\mathbf{P}_1\mathbf{H}^{01}(\mathbf{P}_1^\top\mathbf{L}_1\mathbf{P}_1)\mathbf{P}_1^\top\mathbf{f}_{t-p}\end{aligned} \tag{89}$$

In (89), any simplicial convolutional filter of the form $\mathbf{H}(\mathbf{P}_k^\top\mathbf{L}_k\mathbf{P}_k)$ can be written as:

$$\begin{aligned}\mathbf{H}(\mathbf{P}_k^\top\mathbf{L}_k\mathbf{P}_k) &= \sum_{n=0}^{L-1}\beta_n(\mathbf{P}_k^\top\mathbf{L}_k\mathbf{P}_k)^n \\ &= \sum_{n=0}^{L-1}\beta_n\mathbf{P}_k^\top\mathbf{L}_k^n\mathbf{P}_k = \mathbf{P}_k^\top\mathbf{H}(\mathbf{L}_k)\mathbf{P}_k,\end{aligned} \tag{90}$$

where the second equality follows from the orthogonal property $\mathbf{P}\mathbf{P}^\top = \mathbf{I}$ of the permutation matrix. Using the result (90) and the orthogonal property of $\mathbf{P}$, (89) can be simplified as:

$$\check{\mathbf{v}}_t = \mathbf{P}_0^\top[\mathbf{H}^{00}(\mathbf{L}_0)\mathbf{v}_{t-p} + \mathbf{G}^{01}(\mathbf{L}_0)\mathbf{B}_1\mathbf{H}^{01}(\mathbf{L}_1)\mathbf{f}_{t-p}] \tag{91}$$

Similar results can be derived for edge and triangle processes, hence, proving SC-VAR's permutation equivariance. □

Orientation equivariance. Consider a simplicial $\mathcal{X}^2$ with Hodge Laplacians $\boldsymbol{\mathcal{L}} = \{\mathbf{L}_0, \mathbf{L}_1, \mathbf{L}_2\}$ [cf. (3)] and its reoriented version

$$\check{\boldsymbol{\mathcal{D}}} := \{\mathbf{D}_0\mathbf{L}_0\mathbf{D}_0, \mathbf{D}_1\mathbf{L}_1\mathbf{D}, \mathbf{D}_2\mathbf{L}_2\mathbf{D}_2\}$$

for any orientation matrice $\mathbf{D} = \text{diag}(\mathbf{D}_0, \mathbf{D}_1, \mathbf{D}_2)$ such that $\mathbf{D}_k \in \mathcal{D}$. Given also a set of parameters $\mathcal{H}_{\text{SC-VAR}}$. Then, for any pair of a simplicial complex signal $\mathbf{x}_t$ and its reoriented version $\check{\mathbf{x}}_t := \mathbf{D}^\top\mathbf{x}_t$, the SC-VAR output satisfies:

$$\begin{aligned}\check{\mathbf{x}}_t :&= \Phi_{\text{SC-VAR}}(\{\mathbf{D}^\top\mathbf{x}_{t-p}\}; \check{\boldsymbol{\mathcal{L}}}; \mathcal{H}_{\text{SC-VAR}}) \\ &= \mathbf{D}^\top\Phi_{\text{SC-VAR}}(\{\mathbf{x}_{t-p}\}; \boldsymbol{\mathcal{L}}; \mathcal{H}_{\text{SC-VAR}}).\end{aligned} \tag{92}$$

Proof: Similar to the permutation matrix $\mathbf{P}_k$, a reorientation matrix $\mathbf{D}_k$ in (87) also exhibits the orthogonal property $\mathbf{D}_k\mathbf{D}_k^\top = \mathbf{I}$. The proof for orientation equivariance can be derived by adapting the proof of permutation equivariance with the replacement of $\mathbf{P}_k$ by $\mathbf{D}_k$. □

APPENDIX D

Proof of Proposition 2. The basic principle behind proving the claim is the boundary condition property between vertical simplices, i.e., $\mathbf{B}_1\mathbf{B}_2 = \mathbf{0}$. Without loss of generality, we provide a sketch of the proof for vertex processes $\mathbf{v}_t$ for $P = 1$. The SC-VAR vertex process has the form

$$\mathbf{v}_t = \mathbf{H}^{00}(\mathbf{L}_0)\mathbf{v}_{t-1} + \mathbf{G}^{01}(\mathbf{L}_0)\mathbf{B}_1\mathbf{H}^{01}(\mathbf{L}_1)\mathbf{f}_{t-1} \tag{93}$$

Likewise, the SC-VAR edge process at time $t-1$ can be expanded as:

$$\begin{aligned}\mathbf{f}_{t-1} = &\mathbf{G}^{10}(\mathbf{L}_1)\mathbf{B}_1^\top\mathbf{H}^{10}(\mathbf{L}_0)\mathbf{v}_{t-2} + \mathbf{H}^{11}(\mathbf{L}_1)\mathbf{f}_{t-2} \\ &+ \mathbf{G}^{12}(\mathbf{L}_1)\mathbf{B}_2\mathbf{H}^{12}(\mathbf{L}_2)\boldsymbol{\tau}_{t-2}.\end{aligned} \tag{94}$$

Substituting (94) into (93) yields

$$\begin{aligned}\mathbf{v}_t = &\mathbf{H}^{00}(\mathbf{L}_0)\mathbf{v}_{t-1} \\ &+ \mathbf{G}^{01}(\mathbf{L}_0)\mathbf{B}_1\mathbf{H}^{01}(\mathbf{L}_1)\mathbf{G}^{10}(\mathbf{L}_1)\mathbf{B}_1^\top\mathbf{H}^{10}(\mathbf{L}_0)\mathbf{v}_{t-2} \\ &+ \mathbf{G}^{01}(\mathbf{L}_0)\mathbf{B}_1\mathbf{H}^{01}(\mathbf{L}_1)\mathbf{H}^{11}(\mathbf{L}_1)\mathbf{f}_{t-2} \\ &+ \mathbf{G}^{01}(\mathbf{L}_0)\mathbf{B}_1\mathbf{H}^{01}(\mathbf{L}_1)\mathbf{G}^{12}(\mathbf{L}_1)\mathbf{B}_2\mathbf{H}^{12}(\mathbf{L}_2)\boldsymbol{\tau}_{t-2}.\end{aligned} \tag{95}$$

In (95), the fourth term vanishes because when expanding the expression $\mathbf{B}_1\mathbf{H}^{01}(\mathbf{L}_1)\mathbf{G}^{12}(\mathbf{L}_1)\mathbf{B}_2$, all subterms involve $\mathbf{B}_1\mathbf{B}_2$, which evaluate to 0 according to the boundary condition. Hence, the vertex process $\mathbf{v}_t$ is influenced only by past vertex and edge processes but not by the triangle signals nor by the transformation of triangle signals into curl flows [cf. (9)]. Similar arguments can be applied to prove the vertical property of the triangle signal.

REFERENCES

[1] J. Krishnan, R. Money, B. Beferull-Lozano, and E. Isufi, "Simplicial vector autoregressive model for streaming edge flows," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2023, pp. 1–5.

[2] H. Weiyu, G. Leah, W. Nicholas, G. Scott, B. Danielle, and R. Alejandro, "Graph frequency analysis of brain signals," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 7, pp. 1189-1203, Oct. 2016.

[3] D. Cheng, F. Yang, S. Xiang, and J. Liu, "Financial time series forecasting with multi-modality graph neural network," *Pattern Recognit.*, vol. 121, 2022.

[4] R. Money, J. Krishnan, and B. Beferull-Lozano, "Online joint nonlinear topology identification and missing data imputation over dynamic graphs," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, 2022.

[5] A. Carriero, T. Clark, and M. Marcellino, "Large bayesian vector autoregressions with stochastic volatility and non-conjugate priors," *J. Econ.*, vol. 212, no. 1, pp. 137–154, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030440761930079X

[6] M. Feldkircher, L. Gruber, F. Huber, and G. Kastner, "Sophisticated and small versus simple and sizeable: When does it pay off to introduce drifting coefficients in bayesian vector autoregressions?" *J. Forecasting*, vol. 43, no. 6, pp. 2126–2145, 2024. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/for.3121

[7] A. Niknam, H. Zare, H. Hosseininasab, A. Mostafaeipour, and M. Herrera, "A critical review of short-term water demand forecasting tools —What method should I use?" *Sustainability*, vol. 14, no. 9, p. 5412, 2022.

[8] B. Zaman, L. Ramos, D. Romero, and B. Beferull-Lozano, "Online topology identification from vector autoregressive time series," *IEEE Trans. Signal Process.*, vol. 69, pp. 210–225, 2021.

[9] R. Money, J. Krishnan, and B. Beferull-Lozano, "Online non-linear topology identification from graph-connected time series," in *Poroc. IEEE Data Sci. Learn. Workshop (DSLW)*, 2021, pp. 1–6.

[10] R. Money, J. Krishnan, and B. Beferull-Lozano, "Random feature approximation for online nonlinear graph topology identification," in *Proc. IEEE MLSP*, 2021, pp. 1–6.

[11] R. Money, J. Krishnan, and B. Beferull-Lozano, "Sparse online learning with kernels using random features for estimating nonlinear dynamic graphs," *IEEE Trans. Signal Process.*, vol. 71, pp. 2027–2042, 2023.

[12] Y. Shen, G. Giannakis, and B. Baingana, "Nonlinear structural vector autoregressive models with application to directed brain networks," *IEEE Trans. Signal Process.*, vol. 67, pp. 5325–5339, 2019.

[13] M. Veedu, D. Harish, and M. Salapaka, "Topology learning of linear dynamical systems with latent nodes using matrix decomposition," *IEEE Trans. Autom. Control*, vol. 67, no. 11, pp. 5746–5761, Nov. 2022.

[14] E. Isufi, A. Loukas, N. Perraudin, and G. Leus, "Forecasting time series with VARMA recursions on graphs," *IEEE Trans. Signal Process.*, vol. 67, no. 18, pp. 4870–4885, Sep. 2019.

[15] L. Gruber and G. Kastner, "Forecasting macroeconomic data with bayesian vars: Sparse or dense? It depends!" 2023, *arXiv:2206.04902*.

[16] G. Kastner and F. Huber, "Sparse bayesian vector autoregressions in huge dimensions," *J. Forecasting*, vol. 39, no. 7, pp. 1142–1165, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/for.2680

[17] G. Primiceri, "Time varying structural vector autoregressions and monetary policy," *Rev. Econ. Stud.*, vol. 72, no. 3, pp. 821–852, Jul. 2005, doi: 10.1111/j.1467-937X.2005.00353.x.

[18] C. Lam and Q. Yao, "Factor modeling for high-dimensional time series: Inference for the number of factors," *Ann. Statist.*, vol. 40, no. 2, pp. 694–726, 2012.

[19] N. Lee, H. Choi, and S. Kim, "Bayes shrinkage estimation for high-dimensional var models with scale mixture of normal distributions for noise," *Comput. Statist. & Data Anal.*, vol. 101, no. 2016, pp. 250–276, 2016.

[20] J. Chang, B. Guo, and Q. Yao, "Principal component analysis for second-order stationary vector time series," *Ann. Statist.*, vol. 46, no. 5, pp. 2094–2124, 2018.

[21] P. Battaglia et al., "Relational inductive biases, deep learning, and graph networks," 2018, *arXiv:1806.01261*.

[22] E. Isufi, F. Gama, D. Shuman, and S. Segarra, "Graph filters for signal processing and machine learning on graphs," *IEEE Trans. Signal Process.*, vol. 72, pp. 4745–4781, 2024.

[23] M. Schaub, Y. Zhu, J. Seby, T. Roddenberry, and S. Segarra, "Signal processing on higher-order networks: Livin'on the edge... and beyond," *Signal Process.*, vol. 187, 2021, Art. no. 108149.

[24] G. Leus, M. Yang, M. Coutino, and E. Isufi, "Topological volterra filters," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2021, pp. 5385–5399.

[25] G. Essl, "Topological IIR filters over simplicial topologies via sheaves," *IEEE Signal Process. Lett.*, vol. 27, pp. 1215–1219, 2020.

[26] M. Yang and E. Isufi, "Convolutional learning on simplicial complexes," 2023, *arXiv:2301.11163v1*.

[27] E. Isufi and M. Yang, "Convolutional filtering in simplicial complexes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2022, pp. 5578–5582.

[28] F. Baccini, F. Geraci, and G. Bianconi, "Weighted simplicial complexes and their representation power of higher-order network data and topology," *Phys. Rev. E*, vol. 106, Sep. 2022, Art. no. 034319.

[29] C. Battiloro, S. Sardellitti, S. Barbarossa, and P. Lorenzo, "Topological signal processing over weighted simplicial complexes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2023, pp. 1–5.

[30] R. Money, J. Krishnan, B. Beferull-Lozano, and E. Isufi, "Online edge flow imputation on networks," *IEEE Signal Process. Lett.*, vol. 30, pp. 115–119, 2023.

[31] C. Battiloro, P. Lorenzo, and S. Barbarossa, "Topological slepians: Maximally localized representations of signals over simplicial complexes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2023, pp. 1–5.

[32] C. Battiloro, P. Lorenzo, and A. Ribeiro, "Parametric dictionary learning for topological signal representation," in *Proc. 31st Eur. Signal Process. Conf. (EUSIPCO)*, 2023, pp. 1958–1962.

[33] T. Roddenberry, F. Frantzen, M. Schaub, and S. Segarra, "Hodgelets: Localized spectral representations of flows on simplicial complexes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2022, pp. 5922–5926.

[34] S. Reddy and S. Chepuri, "Sampling and recovery of signals on a simplicial complex using neighbourhood aggregation," 2023, *arXiv:2308.09333*.

[35] L. Lim, "Hodge laplacians on graphs," *Siam Rev.*, vol. 62, no. 3, pp. 685–715, 2020.

[36] S. Barbarossa and S. Sardellitti, "Topological signal processing over simplicial complexes," *IEEE Trans. Signal Process.*, vol. 68, pp. 2992–3007, 2020.

[37] M. Yang, E. Isufi, M. Schaub, and G. Leus, "Simplicial convolutional filters," *IEEE Trans. Signal Process.*, vol. 70, pp. 4633–4648, 2022.

[38] H. Lütkepohl, *New Introduction to Multiple Time Series Analysis*. Berlin, Heidelberg, Germany: Springer Science & Business Media, 2005.

[39] Y. Feng, D. Palomar et al., *A signal Processing Perspective on Financial Engineering*. Boston, MA, USA: Now, 2016, vol. 9.

[40] C. Gorrostieta, H. Ombao, P. Bédard, and J. N. Sanes, "Investigating brain connectivity using mixed effects vector autoregressive models," *NeuroImage*, vol. 59, no. 4, pp. 3347–3355, 2012.

[41] D. Shuman, S. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[42] T. Roddenberry, N. Glaze, and S. Segarra, "Principled simplicial neural networks for trajectory prediction," in *Proc. 38th Int. Conf. Mach. Learn.*, vol. 139. PMLR, Jul. 2021, pp. 9020–9029.

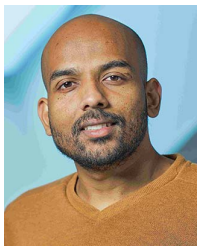
[43] M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges," 2021, *arXiv:2104.13478*.

[44] R. Litterman, "Forecasting with Bayesian vector autoregressions: Five years of experience," *J. Bus. & Econ. Statist.*, vol. 4, no. 1, pp. 25–38, 1986. [Online]. Available: http://www.jstor.org/stable/1391384

[45] E. Isufi, G. Leus, and P. Banelli, "2-dimensional finite impulse response graph-temporal filters," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP),* Piscataway, NJ, USA: IEEE Press, 2016, pp. 405–409.

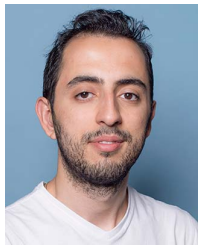
[46] F. Grassi, A. Loukas, N. Perraudin, and B. Ricaud, "A time-vertex signal processing framework: Scalable processing and meaningful representations for time-series on graphs," *IEEE Trans. Signal Process.*, vol. 66, no. 3, pp. 817–829, 2017.
[47] A. V. Oppenheim, J. R. Buck, and R. W. Schafer, *Discrete-Time Signal Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.
[48] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[49] L. Rossman, R. Clark, and W. Grayman, "Modeling chlorine residuals in drinking water distribution systems," *J. Environ. Eng.*, vol. 120, no. 4, pp. 803–820, 1994.
[50] S. Pettersen et al., "Soccer video and player position dataset," in *Proc. 5th ACM Multimedia Syst. Conf.*, 2014, pp. 18–23.
[51] B. Zaman, "Online machine learning for graph topology identification from multiple time series," *UiA*, 2019. [Online]. Available: https://uia.brage.unit.no/uia-xmlui/handle/11250/2685379

Joshin Krishnan received the B.Tech. degree in electronics and communication engineering from the College of Engineering, Thiruvananthapuram, (University of Kerala, India), in 2010, the M.E. degree in telecommunications from the Indian Institute of Science (IISc), Bengaluru, in 2014, and the Ph.D. degree in electrical and computer engineering from the Instituto Superior Técnico, University of Lisbon, Portugal, in 2019. Currently, he is working as a R&D Signal Processing Engineer with Safran Sensing Technologies, Norway. From 2015 to 2019, he was with the Instituto de Telecomunicações, Lisbon, as a Marie Curie Early-Stage Researcher of the Machine Sensing Training Network (MacSeNet). From 2019 to 2024, he served as a Postdoctoral Researcher with WISENET Research Center, UiA, Norway, and the Simula Metropolitan Center for Digital Engineering, Norway. His research interests include image inverse problems, optimization, time-series analysis, graph signal processing, and machine learning.



Rohan Money received the B.Tech. degree in electrical and electronics engineering from Rajiv Gandhi Institute of Technology, M.G. University, Kottayam, India, in 2015, the M.Tech. degree in systems and control from the Indian Institute of Technology Hyderabad, Hyderabad, India, in 2018, and the Ph.D. degree from the Department of ICT, University of Agder, Grimstad, Norway, in 2023. Currently, he is a Postdoctoral Researcher with the Simula Metropolitan Center for Digital Engineering, Norway. His research interests include optimization, time-series analysis, graph signal processing, machine learning, and wireless communication. He was the recipient of the Best Student Paper Runner-Up Award at the IEEE DSLW 2021 Conference.

Baltasar Beferull-Lozano (Senior Member, IEEE) received the M.Sc. (Hons.) degree in physics from the Universidad de Valencia, Spain, in 1995, and the M.Sc. and Ph.D. degrees in electrical engineering from University of Southern California (USC), Los Angeles, in 1999 and 2002, respectively. In 2002, he joined the AudioVisual Communications Laboratory, Department of Communication Systems, EPFL, as a Research Associate, where he spent around three years. In 2006, he joined the School of Engineering, University of Valencia as an Associate Professor. Since 2014, he is a Professor with the Department of Information and Communication Technology and (by courtesy) the Department of Engineering, University of Agder, Norway, where he leads the Center Intelligent Signal Processing and Wireless Networks (WISENET). Since 2021, he is also a Chief Research Scientist and Professor with the Simula Metropolitan Center for Digital Engineering (SimulaMet), Norway, where he leads the Department of Signal and Information Processing for Intelligent Systems (SIGIPRO). He has served as a Senior Area Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING since 2016 and has also served as a member of the Technical Program Committees for several ACM & IEEE International Conferences. His research interests include in the general areas of data science and machine learning, graph signal processing, in-network signal processing and collective intelligence, optimization, networked cyber-physical systems and artificial intelligence for next generation wireless networks. At USC, he received several awards including the Best Ph.D. Thesis Paper Award in 2002 and the Outstanding Academic Achievement Award in 1999. He has received Best Paper Awards at several international conferences, such as IEEE DCOSS and IEEE DSLW. He has also received a TOPPFORSK Grant Award from the Research Council of Norway, 2015. He isa member of the Norwegian Academy of Science and Technology.



Elvin Isufi (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electronic and telecommunication engineering from the University of Perugia, Italy, in 2012 and 2014, respectively, and the Ph.D. degree in electrical engineering from Delft University of Technology (TU Delft), The Netherlands, in 2019. In 2019, he was a Postdoctoral Researcher with the Department of Electrical and Systems Engineering, University of Pennsylvania. Currently, he is an Assistant Professor with the Faculty of Electrical Engineering, Mathematics, and Computer Science, TU Delft, where he co-founded and co-directs AIdroLab—the TU Delft AI Lab focused on fundamental advances in graph-based machine learning for water networks and flood modeling. His research interests lie at the intersection of signal processing, mathematical modeling, machine learning, and network theory, particularly in processing and learning from data on graphs and higher-order structures. He received the 2022 IEEE Signal Processing Society (SPS) Best Ph.D. Dissertation Award, along with paper recognition awards at IEEE CAMSAP 2017, DSLW 2021, DSLW 2022, and ICASSP 2023. He is a member of the IEEE SPS Technical Committee on Signal Processing for Communications and Networking, and holds editorial positions with IEEE TRANSACTIONS ON SIGNAL PROCESSING (2024–2026) and Elsevier Signal Processing (2021–2024). He is a Veni Fellow and an MSCA Co-Fund Fellow.