Topological signal processing and learning
Recent advances and future challenges

Isufi, Elvin; Leus, Geert; Beferull-Lozano, Baltasar; Barbarossa, Sergio; Di Lorenzo, Paolo

DOI
10.1016/j.sigpro.2025.109930

Publication date
2025

Document Version
Final published version

Published in
Signal Processing

Signal Processing 233 (2025) 109930

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Topological signal processing and learning: Recent advances and future challenges

Elvin Isufi [a,*], Geert Leus [a], Baltasar Beferull-Lozano [b], Sergio Barbarossa [c], Paolo Di Lorenzo [c,*]

[a] Faculty of Electrical Engineering Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands
[b] SIGIPRO, Department, Simula Metropolitan Center for Digital Engineering, Oslo, Norway
[c] Department of Information Engineering, Electronics, and Telecommunications, Sapienza University of Rome, Rome, Italy

ARTICLE INFO

Keywords:
Topological signal processing
Topological deep learning
Topological data analysis
Graph signal processing
Graph machine learning
Network science
Hodge theory

ABSTRACT

Developing methods to process irregularly structured data is crucial in applications like gene-regulatory, brain, power, and socioeconomic networks. Graphs have been the go-to algebraic tool for modeling the structure via nodes and edges capturing their interactions, leading to the establishment of the fields of graph signal processing (GSP) and graph machine learning (GML). Key graph-aware methods include Fourier transform, filtering, sampling, as well as topology identification and spatiotemporal processing. Although versatile, graphs can model only pairwise dependencies in the data. To this end, topological structures such as simplicial and cell complexes have emerged as algebraic representations for more intricate structure modeling in data-driven systems, fueling the rapid development of novel topological-based processing and learning methods. This paper first presents the core principles of topological signal processing through the Hodge theory, a framework instrumental in propelling the field forward thanks to principled connections with GSP-GML. It then outlines advances in topological signal representation, filtering, and sampling, as well as inferring topological structures from data, processing spatiotemporal topological signals, and connections with topological machine learning. The impact of topological signal processing and learning is finally highlighted in applications dealing with flow data over networks, geometric processing, statistical ranking, biology, and semantic communication.

1. Introduction and motivation

Many technological, biological, and natural systems exhibit data with inherently irregular structures, as observed in critical infrastructure networks, neuroscience, gene regulatory networks, and social interaction systems [1–4]. Such data often defy the assumptions of traditional Euclidean-based signal processing and machine learning techniques, rendering these approaches insufficient for capturing their underlying complexities. Consequently, these irregular dependencies have motivated the development of new perspectives and methodologies capable of accommodating their non-Euclidean nature. In particular, graphs have emerged as the dominant paradigm for modeling irregular data structures by representing pairwise relationships through nodes and edges. This framework has been pivotal in historical advances, such as graphical modeling [5] and network science [2,4], as well as in the more recent fields of graph signal processing (GSP) [6] and graph machine learning (GML) [7]. GSP extends classical signal processing principles to graph-structured data, enabling tasks such as filtering [8], spectral analysis [9], and signal sampling [10,11]. GML, on the other hand, leverages graph structures as *relational inductive biases* [12], to design deep learning architectures that directly capture relations among the data, thus facilitating powerful predictive models through representation learning techniques like graph neural networks (GNNs) [13]. These methods have revolutionized applications in social networks, recommendation systems, and biological networks, showcasing their potential for uncovering intricate patterns in relational data. However, the pairwise modeling of graphs imposes limitations in representing more complex relationships, necessitating a broader paradigm.

While graphs are versatile, they represent a simple example of a topological space, limited to capturing pairwise (or dyadic) relationships between data entities [14]. As data grows increasingly complex —such as in gene regulatory networks, social interactions, and neural activity— the richness of interactions among the constituent elements often exceeds the scope of simple dyadic relationships [15–17]. Moreover, graph-based techniques are predominantly designed for signals defined on nodes, making them inadequate for analyzing signals associated with higher-dimensional structures, such as flows along edges

* Corresponding authors.
E-mail address: paolo.dilorenzo@uniroma1.it (P. Di Lorenzo).

https://doi.org/10.1016/j.sigpro.2025.109930
Received 22 November 2024; Received in revised form 26 January 2025; Accepted 31 January 2025
Available online 10 February 2025

or signals defined over groups of nodes. These limitations constrain the applicability of graph signal processing and learning methods in scenarios where dependencies span multiple scales and dimensions. Therefore, to effectively capture these data structures and their intricate interdependencies, it is crucial to move beyond graphs and adopt richer topological representations.

Topological representations, such as hypergraphs, simplicial complexes, and cell complexes, provide powerful frameworks for modeling and analyzing multiway relationships [17]. Hypergraphs extend the concept of edges to encompass connections among more than two nodes, while simplicial and cell complexes introduce hierarchical structures that facilitate efficient processing across multiple dimensions [18]. Although hypergraphs are highly flexible, their generality introduces an inherent complexity in terms of algebraic representation, mainly based on tensor models [19–23], which might limit practical applicability. In contrast, simplicial and cell complexes endow the domain with hierarchical structures, enabling numerically efficient processing that can handle signals defined at various levels of higher-order structures [14]. Additionally, simplicial and cell complexes are equipped with advanced algebraic tools, such as Hodge theory and the Hodge Laplacian, which enable spectral analysis and signal decomposition across different topological levels [24]. One of the fundamental capabilities of algebraic topology is to extract *global properties* of the space, i.e. *invariants*, starting from *local relations*. Graphs, which are a simple case of topological space, can only capture global properties associated to connectivity. Higher order structures, like simplicial or cell complexes, enable the extraction of additional invariants, like the number of holes or cavities associated with covering the embedding space with higher order simplicial or cell complexes. The knowledge of these invariants, also known as *Betti numbers*, plays a fundamental role in devising signal processing tools tuned to the space where the signals live. Finally, unlike graphs, higher order structures allow a joint processing of signals defined not only on nodes and edges but also on higher-dimensional entities, such as triangles or tetrahedra, ultimately, paving the way for new opportunities in analyzing and learning from complex, multi-scale data. Recent advances in topological signal processing and learning (TSPL) have harnessed these capabilities, enabling innovative methods for filtering, reconstruction, and representation learning that are grounded in the geometric and topological properties of data.

While recent progress in TSPL has demonstrated its potential, current approaches face significant challenges such as developing methods from first principles, ensuring computational efficiency, and achieving seamless integration with machine learning frameworks. Advances in Hodge theory, topological Fourier transform, convolutional filtering, and topological neural networks, have provided a theoretical foundation for the field. However, many techniques remain fragmented across disciplines, hindering their broader adoption and development. This paper seeks to unify the latest advances in TSPL, offering a comprehensive and easily-accessible overview of the methods, challenges, and opportunities in this rapidly evolving domain. By addressing the limitations of graph-based techniques and emphasizing the promise of topological frameworks, we aim to inspire further research and applications of TSPL methods to address the increasingly complex demands of modern data analysis.

1.1. Related works and paper position

This paper serves two main purposes. First, it provides a unified, comprehensive, up-to-date, and accessible resource on topological structures and their algebraic foundations. It explores how these structures serve as a framework for signals with irregular characteristics that cannot be represented by graphs and how they connect to key signal processing concepts through Hodge theory. Second, it seeks to unify recent, scattered works in topological signal processing and topological machine learning by linking them to these foundational concepts. Ultimately, the paper aims to serve as a valuable resource for both first-time and advanced readers, helping them identify key contributions in this rapidly evolving field. To this end, we position this work in relation to advances in related areas, as well as to seminal works in topological signal processing and topological machine learning.

(1) Graph signal processing and graph machine learning: Advances in these areas primarily address pairwise dependencies between data points represented by graphs and focus on processing or learning representations from node values (commonly referred to as graph signals) [6,8,13,25,26]. Consequently, much of this work does not emphasize processing data defined on edges or other higher-order structures. These contributions have established foundational concepts, such as the graph Fourier transform, sampling and reconstruction of graph signals, graph convolutions, and graph neural networks. While some works on message-passing graph neural networks include aggregation rules for edge values, they often do so in a black-box manner, treating edge values as features without providing principled insights into how they should interact with neighboring node or edge signals. Connections to TSPL arise primarily through the Hodge Laplacian, which encodes relational locality between topological signals. For instance, the Hodge decomposition and the spectrum of the Hodge Laplacian form the foundation for a topological Fourier transform, strongly tied to GSP concepts. These tools also play a critical role in developing principled topological neural networks.

(2) Network science: This area focuses on analyzing complex systems and understanding their behavior through higher-order networks, such as simplicial complexes, cell complexes, and hypergraphs [15,17,27–30]. On the one hand, much of this research emphasizes the structure of higher-order networks rather than the data defined on them. For example, studies explore probabilistic models for their structure, algebraic representations, and the relationship between spectral properties and higher-order interactions. Connections to TSPL arise primarily from the simplicial and cell complex representations of higher-order networks and the use of the eigendecomposition of Hodge Laplacians to represent signals in the spectral domain. On the other hand, many works focus on modeling information diffusion and contagion over higher-order networks [16,27,31–33]. TSPL can complement these efforts by characterizing the evolution of such processes from a Fourier perspective, providing sampling and reconstruction strategies, and offering more powerful data-driven embedding techniques compared to higher-order random walks [34]. Since then, numerous studies have expanded on these foundational concepts, formalizing methods for convolutional filtering, sampling, and reconstruction, as well as developing the first techniques to learn topologies from signals and manage spatiotemporal topological signals.

(3) Topological data analysis (TDA): This direction focuses on extracting meaningful shapes and structures present in the data by analyzing them in high-dimensional or complex spaces [35–38]. TDA provides a suite of tools, such as persistent homology, which captures topological invariants like connected components, loops, and voids across different scales, offering a robust way to characterize data geometry [39,40]. Many TDA methods align conceptually with TSPL, as both frameworks aim to model and process data through its multi-scale and multi-dimensional relationships. Persistent homology, for example, has been used to compute topological summaries that complement signal processing techniques by providing insights into the global and local structure of data. Furthermore, advances in TDA have inspired new approaches in topological machine learning, where persistent diagrams and topological features are integrated with neural network architectures for tasks like classification, clustering, and generative modeling [41]. The core idea of TDA is to use topology as a lens to uncover patterns and relationships in data that may be invisible to traditional methods. Differently, TSPL considers the topology as the support for the data and uses principled topological tools to derive analysis tools tuned to the properties of the topological space where the data lives.

(4) Hypergraph signal processing: This field focuses on processing signals defined over general topological spaces represented by hypergraphs. The study in [19] introduced a tensor-based hypergraph signal processing (HGSP) framework to extend graph signal processing (GSP) to high-order interactions. Key advancements include the definition of the hypergraph Fourier space, analysis of hypergraph Fourier transform spectrum properties, development of hypergraph sampling theory, and the fundamentals of hypergraph filter design. This approach was later extended in [20,22], leveraging a novel tensor-tensor product algebra, a powerful tool for preserving the intrinsic structures of tensors. Notable applications of these frameworks include 3D point cloud analysis [21] and brain functional connectivity inference [42]. Furthermore, [23] introduced hypergraph neural network architectures grounded in the HGSP framework, opening new avenues for high-order data analysis.

(5) Topological signal processing: The first works on topological signal processing began to mature with [43–45]. In parallel, the first tools using tensor representations for hypergraph signal processing were introduced in [19], extending the principles of GSP to hypergraph signals. More recently, the overview article in [46] provided a unified, tutorial-style perspective on signal processing concepts for simplicial complexes and hypergraphs. Building on these early contributions, numerous studies have since formalized key methods; however, these developments remain scattered across various disciplines. In Section 4.1, we unify these scattered advancements, providing a cohesive perspective.

(6) Topological deep learning: This direction focuses on developing end-to-end representation learning methods for data defined on topological structures, building on the success of graph neural networks [47–52]. Many of the methods and analyses in these works share foundational principles with topological signal processing. For instance, they leverage Hodge theory and Hodge decomposition to construct principled architectures, utilizing approaches such as convolution [53,54] or attention mechanisms [55], among others. The core idea is to use topology as a relational inductive bias to enhance deep learning methods, enabling inductive learning solutions that can generalize across different topologies and address the limitations of graph-based neural networks. In this context, we highlight Hodge theory and related topological signal processing concepts as foundational tools for advancing topological deep learning (TDL). These concepts pave the way for principled methods that complement and extend the message-passing paradigm.

1.2. Outline of the paper

This paper is organized as follows. Section 2 introduces the key elements of topological representations of data structures and topological signals. It builds upon the concept of signals defined on graphs and graph Laplacians, extending these ideas to topological signals and Hodge Laplacians. Section 3 focuses on the spectral processing of topological signals using Hodge theory. It first revisits the graph Fourier transform and then transitions to the topological Fourier transform, highlighting their similarities as well as key differences, such as the interpretation of topological frequencies. Section 4 discusses recent advancements in methods for signal representation, sampling and reconstruction, learning topologies from data, and handling spatiotemporal topological signals. This section connects these methods to the core concepts introduced in Sections 3 and 4, and explores how these techniques can be applied to develop and enhance machine and deep learning methods for topological structures. Section 5 presents promising application areas tailored to the methods discussed. Finally, Section 6 concludes the paper and outlines key open issues that warrant further research.

2. Knowledge representation over topological spaces

In this section, we formalize the concept of signals over topological spaces, focusing specifically on signals defined over simplicial and cell complexes, alongside the algebraic characterization of their associated domains. To ease exposition, we first revisit signals on graphs in Section 2.1, before extending these concepts to topological domains in Section 2.2.

2.1. Signals on graphs

Let us consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of a set of N vertices $\mathcal{V} = \{1, 2, \dots, N\}$, along with a set of E edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. Let us also denote the $N \times N$ adjacency matrix as $\mathbf{A}$ where the element (i, j) is denoted as $A_{i,j}$, $i, j \in \mathcal{V}$. We assume $A_{i,j} > 0$, if there is a link from node j to node i, i.e., $(j, i) \in \mathcal{E}$, or $A_{i,j} = 0$, otherwise. The combinatorial Laplacian matrix for an undirected graph with a symmetric adjacency matrix $\mathbf{A}$ is defined as $\mathbf{L}_0 = \text{diag}(\mathbf{1}^\top \mathbf{A}) - \mathbf{A}$.

Alternatively, a graph can be represented by its incidence matrix that encodes the incidence relations between vertices and edges. To define the incidence matrix $\mathbf{B}_1$, even if the original graph is undirected, it is necessary to introduce an orientation of the edges. Then, for each edge we have an arrow and, in the case of a binary relation, the entries of $\mathbf{B}_1$ are defined as follows:

$$\left[\mathbf{B}_1\right]_{i,j} = \begin{cases} 0, & \text{if node } i \text{ is not incident on edge } j \\ 1, & \text{if node } i \text{ is the head of arrow } j \\ -1, & \text{if node } i \text{ is the tail of arrow } j \end{cases} \tag{1}$$

It is easy to check that the combinatorial Laplacian matrix, for undirected graphs, can be written as:

$$\mathbf{L}_0 = \mathbf{B}_1 \mathbf{B}_1^\top, \tag{2}$$

and its structure is independent from the orientation chosen for the edges.

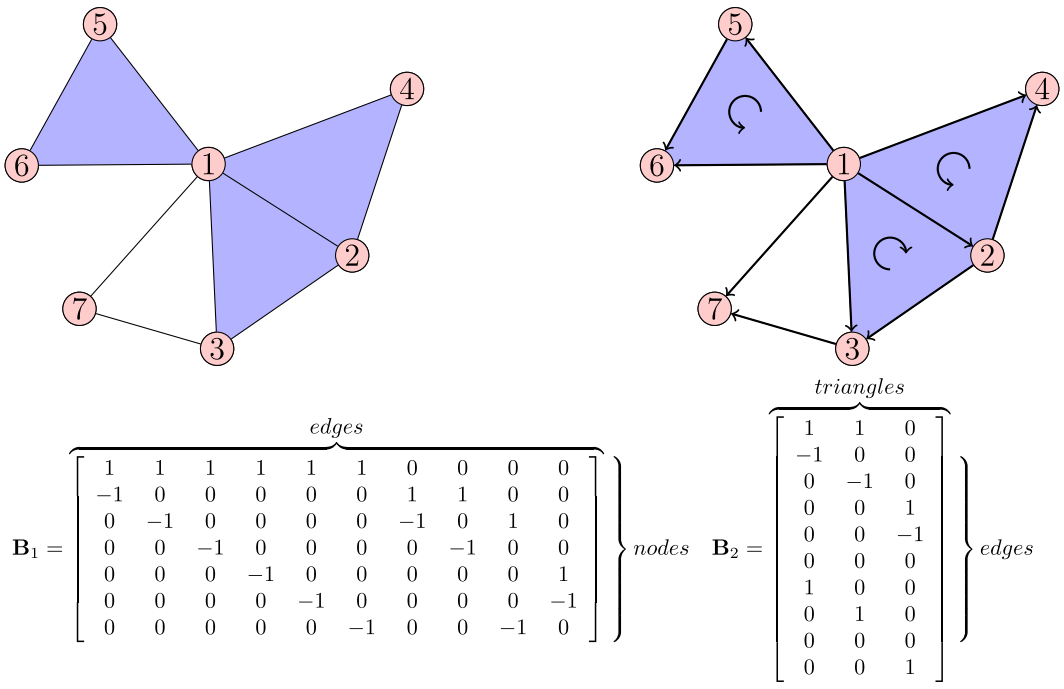
A signal $\mathbf{x} = [x_1, \dots, x_N]^\top$ over a graph $\mathcal{G}$ is a mapping from the vertex set to the set of real numbers, i.e., $\mathbf{x} : \mathcal{V} \to \mathbb{R}$. Here, entry x_i is the signal value associated to node $i \in \mathcal{V}$. Clearly, this definition can be generalized by associating vector or matrix-type data to nodes. The fundamental assumption in graph signal processing is that the algebraic proximities between nodes encoded in $\mathbf{A}$ or $\mathbf{L}_0$ translate into proximities between the respective signals [6,25]. Such a coupling can then be used for processing signal $\mathbf{x}$ by relying on neighboring information in a similar way as we process temporal and image signals based on temporal or spatial proximities [8].

2.2. Signals on topological spaces

Consider a finite set of vertices $\mathcal{V}$. A k-simplex $\mathcal{H}_{k,i}$ is a subset of $\mathcal{V}$ with cardinality $k + 1$. A face of $\mathcal{H}_{k,i}$ is a subset with cardinality k and consequently a k-simplex has $k + 1$ faces. A coface of $\mathcal{H}_{k,i}$ is a $(k + 1)$-simplex that includes $\mathcal{H}_{k,i}$ as a subset. Two simplices are called lower neighbors if they share a common face, and upper neighbors if they share a common coface. A simplicial complex $\mathcal{X}_K$ of order K, is a collection of k-simplices $\mathcal{H}_{k,i}$, $k = 0, \dots, K$ such that, if a simplex $\mathcal{H}_{k,i}$ belongs to $\mathcal{X}_K$, then all its subsets $\mathcal{H}_{k-1,i} \subset \mathcal{H}_{k,i}$ also belong to $\mathcal{X}_K$ (inclusivity property) [56]. The set of k-simplices in $\mathcal{X}_K$ is denoted by $\mathcal{D}_k := \{\mathcal{H}_{k,i} : \mathcal{H}_{k,i} \in \mathcal{X}_K\}$, with its cardinality represented as $|\mathcal{D}_k| = N_k$.

In most of the cases the focus is on complexes of order up to two $\mathcal{X}_2$, thus having a set of vertices $\mathcal{V}$ with $|\mathcal{V}| = N$, a set of edges $\mathcal{E}$ with $|\mathcal{E}| = E$, and a set of triangles $\mathcal{T}$ with $|\mathcal{T}| = T$, which result in $\mathcal{D}_0 = \mathcal{V}$ (simplices of order 0), $\mathcal{D}_1 = \mathcal{E}$ (simplices of order 1), and $\mathcal{D}_2 = \mathcal{T}$ (simplices of order 2). Fig. 1 illustrates one such example. Here, edges $\{1,2\}, \{1,3\}, \{2,3\}$ are faces of the filled triangle {1,2,3}, and this triangle is a co-face of these edges. Edges $\{1,2\}$ and $\{1,3\}$ are both lower neighbors as they share node 1 (common face), and upper-neighbors as they share triangle {1,2,3} (common co-face). Instead, edges $\{1,3\}$ and {1,7} are only lower neighbors.

To facilitate computations, it is customary to introduce a reference orientation for simplices of orders $k \geq 1$, i.e., edges, triangles and so on. This reference orientation is a matter of bookkeeping similar to the arbitrary labeling of the nodes in a graph. W.l.o.g., we fix the



$$\mathbf{B}_1 = \overbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 \end{bmatrix}}^{edges} \Big\} nodes \quad \mathbf{B}_2 = \overbrace{\begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{triangles} \Big\} edges$$

Fig. 1. *(top-left)*A simplicial complex of order $K = 2$. It is composed of 7 nodes, 10 edges, and 3 filled triangles (shaded areas). *Example of simplices:* $\mathcal{H}_{0,1} = \{1\}$, $\mathcal{H}_{0,2} = \{2\}$, $\mathcal{H}_{0,3} = \{3\}$; $\mathcal{H}_{1,1} = \{1,2\}$, $\mathcal{H}_{1,2} = \{1,3\}$, $\mathcal{H}_{1,2} = \{2,3\}$; $\mathcal{H}_{2,1} = \{1,2,3\}$, $\mathcal{H}_{2,2} = \{1,2,4\}$, $\mathcal{H}_{2,3} = \{1,5,6\}$. Notice that triangle $\{1,3,7\}$ is empty and not a part of the SC. *(top-right)* The oriented version of the SC following the lexicographi ordering of its vertices. *Example of oriented simplices:* $\mathcal{H}_{1,1} = [1,2]$, $\mathcal{H}_{1,2} = [1,3]$, $\mathcal{H}_{1,2} = [2,3]$; $\mathcal{H}_{2,1} = [1,2,3]$, $\mathcal{H}_{2,2} = [1,2,4]$, $\mathcal{H}_{2,3} = [1,5,6]$. (bottom) Incidence matrices of the oriented simplicial complex.

orientation for a simplex according to the lexicographical ordering of its vertices. See an example in Fig. 1.

A k-simplicial signal is a collection of mappings from the set of all k-simplices contained in the complex to real numbers:

$$\mathbf{x}_k = [x_k(\mathcal{H}_{k,1}), \dots, x_k(\mathcal{H}_{k,i}), \dots, x_k(\mathcal{H}_{k,N_k})]^\top \in \mathbb{R}^{N_k}, \tag{3}$$

where $x_k : \mathcal{D}_k \to \mathbb{R}$. For simplices of order $k \geq 1$, i.e. edges, triangles, etc., if the signal defined over each simplex represents a flow, then its value is positive if the flow goes in the same direction as the orientation of simplex, or negative in the opposite case. In general, we define a simplicial complex (SC) signal as the concatenation of the signals of each order:

$$\mathbf{x}_{\mathcal{X}} = \left[\mathbf{x}_0 \| \dots \| \mathbf{x}_K\right] \in \mathbb{R}^{\sum_{k=0}^{K} N_k}. \tag{4}$$

For second-order SCs, the k-simplicial signals are defined as the following mappings:

$$x_0 : \mathcal{V} \to \mathbb{R}, \qquad x_1 : \mathcal{E} \to \mathbb{R}, \qquad x_2 : \mathcal{T} \to \mathbb{R}, \tag{5}$$

representing graph, edge and triangle signals, respectively. In this case, the corresponding SC signal is given by:

$$\mathbf{x}_{\mathcal{X}} = \left[\mathbf{x}_0 \| \mathbf{x}_1 \| \mathbf{x}_2\right] \in \mathbb{R}^{N+E+T}. \tag{6}$$

Remark 1 (*Cell Complexes.*). These definitions can be generalized to the case of signals defined over cell complexes, which share most of the properties of simplicial complexes, but also some important differences. For instance, differently from simplicial complexes, the inclusivity does not hold [14], i.e., given a simplex $\mathcal{H}_{k,i} \subset \mathcal{X}_K$, not all subsets $\mathcal{H}_{k-1,i} \subset \mathcal{H}_{k,i}$ need to belong to $\mathcal{X}_k$. Interestingly, for cell complexes of order 2, this gives rise to the presence of polygon-type relationships among the data. An example is illustrated in Fig. 2, where the presence of the quadrilateral does not imply the presence of triangles or diagonal edges within the quadrilateral itself. For more details on the differences between simplicial and cell complexes refer to [14], whereas for the role such differences induce in signal representation refer to [57–59]. □

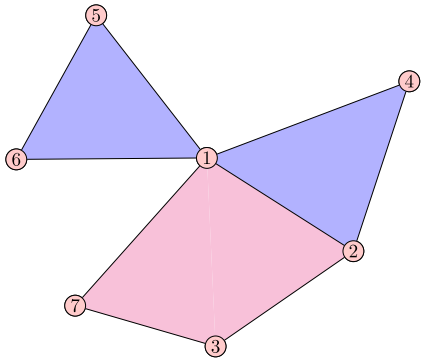


Fig. 2. A regular geometric cell complex of order $K = 2$. It is composed of 7 nodes, 9 edges, 2 filled triangles (shaded blue areas), and 1 filled polygon (shaded magenta area). *Examples of 0-cells:* nodes such as { {1}, …, {7} }. *Examples of 1-cells:* edges such as { {1,2}, …, {5,6} }. *Examples of 2-cells:* triangles { {1,2,4}, {1,5,6} } and the quadrilateral { {1,2,3,7} }. An orientation can now be set to any cell fro order $k \geq 1$ similar to the simplicial complex. The incidence matrix $\mathbf{B}_1$ will capture node-to-edge proximities, while the incidence matrix $\mathbf{B}_2$ will capture edge-to-triangle and edge-to-polygon proximities.

2.3. Algebraic representation

The structure of a simplicial complex $\mathcal{X}_K$ is completely characterized by its set of incidence matrices $\mathbf{B}_k$, $k = 1, \dots, K$. Extending the concept from graphs, the entries of the incidence matrix $\mathbf{B}_k$ specify which k-simplices are incident to which $(k-1)$-simplices. We use the notation $\mathcal{H}_{k-1,i} \sim \mathcal{H}_{k,j}$ to indicate two simplices with the same orientation, and $\mathcal{H}_{k-1,i} \nsim \mathcal{H}_{k,j}$ to indicate that they have opposite orientation. The entries of $\mathbf{B}_k$ are defined as:

$$\left[\mathbf{B}_k\right]_{i,j} = \begin{cases} 0, & \text{if } \mathcal{H}_{k-1,i} \not\subset \mathcal{H}_{k,j} \\ 1, & \text{if } \mathcal{H}_{k-1,i} \subset \mathcal{H}_{k,j} \text{ and } \mathcal{H}_{k-1,i} \sim \mathcal{H}_{k,j} \\ -1, & \text{if } \mathcal{H}_{k-1,i} \subset \mathcal{H}_{k,j} \text{ and } \mathcal{H}_{k-1,i} \nsim \mathcal{H}_{k,j} \end{cases}. \tag{7}$$

As an example, considering a simplicial complex $\mathcal{X}_2$ of order two, we have two incidence matrices: the node-to-edge incidence matrix $\mathbf{B}_1 \in \mathbb{R}^{N\times E}$, and the edge-to-triangle incidence matrix $\mathbf{B}_2 \in \mathbb{R}^{E\times T}$. The latter are illustrated in Fig. 1. From the incidence information, we can build the higher-order Hodge Laplacian matrices, of orders $k = 0, \dots, K$, as:

$$\mathbf{L}_0 = \mathbf{B}_1\mathbf{B}_1^\top, \tag{8}$$

$$\mathbf{L}_k = \underbrace{\mathbf{B}_k^\top\mathbf{B}_k}_{\mathbf{L}_k^{(d)}} + \underbrace{\mathbf{B}_{k+1}\mathbf{B}_{k+1}^\top}_{\mathbf{L}_k^{(u)}}, \ k = 1, \dots, K-1, \tag{9}$$

$$\mathbf{L}_K = \mathbf{B}_K^\top\mathbf{B}_K. \tag{10}$$

Here, $\mathbf{L}_0$ is the combinatorial graph Laplacian defined in Section 2.1 and captures node-to-node proximities. All Laplacian matrices of intermediate orders $k = 1, \dots, K-1$, contain two terms: The first term $\mathbf{L}_k^{(d)}$, known as down Laplacian, encodes the lower adjacency of k-order simplices; the second term $\mathbf{L}_k^{(u)}$, known as upper Laplacian, encodes the upper adjacency of k-order simplices. Thus, for example, two edges are lower adjacent if they share a common vertex, whereas they are upper adjacent if they are faces of a common triangle. Note that the vertices of a graph can only be upper adjacent, if they are incident to the same edge. This is why the Laplacian $\mathbf{L}_0$ contains only one term, and it corresponds to the usual graph Laplacian. Similar definitions apply also for the higher-order Hodge Laplacians of cell-complexes, upon defining the proper incidence relationships between the elements (i.e., the cells) of the domain [58,59].

A key property of the Hodge Laplacians is that they make possible to extract some fundamental global properties (invariants) of the complex. More specifically, the dimension of the kernel of the Hodge Laplacian of order k is equal to the Betti number β_k of order k and its value represents the number of connected components, for $k = 0$, the number of holes, for $k = 1$, the number of 3D cavities, for $k = 2$, and so on.

The Hodge Laplacian matrices play the equivalent role for topological signals of the graph Laplacian pays for graph signals. However, depending on the setting, we may be interested in processing solely a simplicial signal (e.g., edge flows) and have no other signal available, or process jointly the SC signal (e.g., joint processing of node, edge, and triangle signals). We shall detail the differences in these two cases in the next two sections.

2.4. Hodge decomposition

One useful property of the Hodge Laplacian is their link with the Hodge decomposition. In particular, given the kth intermediate Hodge Laplacian $\mathbf{L}_k = \mathbf{B}_k^\top\mathbf{B}_k + \mathbf{B}_{k+1}\mathbf{B}_{k+1}^\top$, the Hodge decomposition states that the signal space associated with each simplex of order k can be decomposed as the direct sum of the following three orthogonal subspaces [24]:

$$\mathbb{R}^{N_k} = \underbrace{\text{span}(\mathbf{B}_k^\top)}_{\text{(a) gradient space}} \oplus \underbrace{\text{span}(\mathbf{B}_{k+1})}_{\text{(b) curl space}} \oplus \underbrace{\text{kernel}(\mathbf{L}_k)}_{\text{(c) harmonic space}}. \tag{11}$$

This implies that the kth simplicial signal space is composed of three subspaces, namely *the gradient space* $\text{span}(\mathbf{B}_k^\top)$, *the curl space* $\text{span}(\mathbf{B}_{k+1})$, and *the harmonic space* $\text{kernel}(\mathbf{L}_k)$. In turn, the Hodge decomposition implies that every signal $\mathbf{x}_k$ of order k can be decomposed as:

$$\mathbf{x}_k = \underbrace{\mathbf{B}_k^\top\tilde{\mathbf{x}}_{k-1}}_{\text{(a) gradient component}} + \underbrace{\mathbf{B}_{k+1}\tilde{\mathbf{x}}_{k+1}}_{\text{(b) curl component}} + \underbrace{\tilde{\mathbf{x}}_k}_{\text{(c) harmonic component}}. \tag{12}$$

That is, there exists three signals $\tilde{\mathbf{x}}_{k-1}, \tilde{\mathbf{x}}_k, \tilde{\mathbf{x}}_{k+1}$ of respective orders $k-1$, k, $k+1$ that can express the signal. In other words, the decomposition in (12) shows how the inter-simplex couplings imposed by the Hodge Laplacians translate into inter-signal couplings across different levels. Such couplings yielding from the decompositions in (11) and (12) carry the following interpretation when discussing edge signals $\mathbf{x}_1$ (i.e., $k = 1$) [44,60]:

(a) *Gradient space and gradient component:* The space $\text{span}(\mathbf{B}_1^\top)$ is called the *gradient space*. An edge flow signal $\mathbf{x}_{1,g} = \mathbf{B}_1^\top\tilde{\mathbf{x}}_0$ living in this space is referred to as a *gradient flow* and it can be obtained by differentiating node signals $\tilde{\mathbf{x}}_0$ along the edges connecting them. The component of an edge signal $\mathbf{x}_1$ living in the gradient space is referred to as the *gradient component* (a.k.a., the irrotational component).

(b) *Curl space and curl component:* The space $\text{span}(\mathbf{B}_2)$ is called the *curl space*. An edge flow signal $\mathbf{x}_{1,c} = \mathbf{B}_2\tilde{\mathbf{x}}_2$ living in this space is referred to as a *curl flow* and it can be induced by some triangle signals $\tilde{\mathbf{x}}_2$. The component of an edge signal $\mathbf{x}_1$ living in the curl space is referred to as the *curl component* (a.k.a., the solenoidal component).

(c) *Harmonic space and harmonic component:* The space $\text{kernel}(\mathbf{L}_k)$ is called the *harmonic space*. An edge flow signal $\tilde{\mathbf{x}}_k$ living in this space is referred to as a *harmonic flow* and it cannot be induced from adjacent simplicial signals. The component of an edge signal $\mathbf{x}_1$ living in the harmonic space is referred to as the *harmonic component*.

The Hodge decomposition shows how the topological proximities between different simplices translate into inter-simplicial couplings. While the decomposition holds for any simplicial signal, the terminology is more intuitive when discussing edge flows but often it is used also for a more general setting. We shall see in Section 3.2 how this Hodge decomposition ties with the Fourier analysis of topological signals.

Performance-complexity tradeoff. Moving beyond graphs introduces additional computational complexity, which must be justified by a corresponding improvement in performance. The advantages of TSP become particularly evident when data exhibit multi-way relationships across various hierarchical levels, which traditional graph-based approaches fail to fully capture. For instance, in the case of edge flows, the Hodge decomposition in (12) identifies three distinct signal components: gradient, curl, and harmonic. As shown in (12), the curl and harmonic components cannot be effectively extracted without incorporating higher-order structures such as triangles or cells. When predictive tasks strongly depend on these components, graph-based methods inevitably lose valuable information, while topological methods can effectively preserve and capture it, resulting in superior performance in such contexts. This is even more critical when addressing applications where data is defined over second- or higher-order simplicial or cell structures, necessitating the development of more advanced and tailored TSP methods. Finally, the effectiveness of TSP relies heavily on the accurate definition of the topological domain. In cases where such knowledge is absent or incomplete, task-specific topology learning methods must be designed to infer the optimal topological structure that enhances performance while managing complexity. Promoting sparsity in the topological representation should serve as a key principle, as it provides a practical means to strike a balance between computational efficiency and performance gains. By addressing these challenges, TSP has the potential to unlock new opportunities for processing and learning from complex, multi-scale data in diverse applications.

2.5. Dirac operator and Dirac decomposition

The Hodge Laplacian and Hodge decomposition are conventionally used to represent, analyze, and process signals within a given simplicial order. When signals across different simplicial orders are present, a joint analysis and processing may carry useful information. Since topological complexes rely on the assumption that simplices influence each other only in consecutive orders, then a natural way is to provide a representation that enables topological signal processing across consecutive simplices. The latter is possible via the concept of the Dirac operator which we elaborate next [61,62].

Dirac operator. Focusing for simplicity on a simplicial complex $\mathcal{X}_2$ of order $K = 2$, the Dirac operator reads as:

$$\mathbf{D}_{\mathcal{X}} := \begin{bmatrix} \mathbf{0} & \mathbf{B}_1 & \mathbf{0} \\ \mathbf{B}_1^\top & \mathbf{0} & \mathbf{B}_2 \\ \mathbf{0} & \mathbf{B}_2^\top & \mathbf{0} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{B}_1 & \mathbf{0} \\ \mathbf{B}_1^\top & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{D}_{\mathcal{X}}^{(d)}} + \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_2 \\ \mathbf{0} & \mathbf{B}_2^\top & \mathbf{0} \end{bmatrix}}_{\mathbf{D}_{\mathcal{X}}^{(u)}} \tag{13}$$

where we refer to $\mathbf{D}_{\mathcal{X}}^{(d)}$ and $\mathbf{D}_{\mathcal{X}}^{(u)}$ as the down and up Dirac operator, respectively. The Dirac operator is such that its square gives a block diagonal concatenation of the Hodge Laplacians, i.e., $\mathbf{D}_{\mathcal{X}}^2 = \text{blkdiag}(\mathbf{L}_0, \mathbf{L}_1, \mathbf{L}_2)$.

Application to topological complex signals. When applied to a SC signal, the Dirac operator yields the shifted signal:

$$\mathbf{x}_{\mathcal{X}}^{(1)} = \begin{bmatrix} \mathbf{x}_0^{(1)} \\ \mathbf{x}_1^{(1)} \\ \mathbf{x}_2^{(1)} \end{bmatrix} = \mathbf{D}_{\mathcal{X}}\mathbf{x}_{\mathcal{X}} = \begin{bmatrix} \mathbf{B}_1\mathbf{x}_1 \\ \mathbf{B}_1^\top\mathbf{x}_0 + \mathbf{B}_2\mathbf{x}_2 \\ \mathbf{B}_2^\top\mathbf{x}_1 \end{bmatrix}. \tag{14}$$

The shifted node signal $\mathbf{x}_0^{(1)} = \mathbf{B}_1\mathbf{x}_1$ sums the edge flows flowing into a node and is referred to as the divergence of the edge flow; the shifted edge signal $\mathbf{x}_1^{(1)} = \mathbf{B}_1^\top\mathbf{x}_0 + \mathbf{B}_2\mathbf{x}_2$ is the linear combination of a gradient flow obtained from the nodes and a curl flow induced by triangle signals; and the shifted triangle signal $\mathbf{x}_2^{(1)} = \mathbf{B}_2^\top\mathbf{x}_1$ is the curl signal of the edge flow. The operation in (14) shows how we can translate inter-topological couplings into inter-topological signal operations. The latter has been used in a few recent works to filter topological signals on simplicial complexes in a consistent way [62], or design principled topological neural architectures [55].

Dirac decomposition. Similar to the Hodge decomposition in (11), the Dirac decomposition states that the signal space associated with a topological signal $\mathbf{x}_{\mathcal{X}}$ can also be decomposed as the direct sum of the following three orthogonal subspaces reminiscent of (13):

$$\mathbb{R}^{\sum_{k=0}^{K} N_k} = \underbrace{\text{span}(\mathbf{D}_{\mathcal{X}}^{(d)})}_{\text{(a) joint gradient space}} \oplus \underbrace{\text{span}(\mathbf{D}_{\mathcal{X}}^{(u)})}_{\text{(b) joint curl space}} \oplus \underbrace{\text{kernel}(\mathbf{D}_{\mathcal{X}})}_{\text{(b) joint harmonic space}} \tag{15}$$

where by following the same terminology as for the Hodge decomposition, we refer to (a) $\text{span}(\mathbf{D}_{\mathcal{X}}^{(d)})$ as *the joint gradient space*; (b) $\text{span}(\mathbf{D}_{\mathcal{X}}^{(u)})$ as *the joint curl space*; and (c) $\text{kernel}(\mathbf{D}_{\mathcal{X}})$ as *the joint harmonic space*. While the interpretation of these subspaces is more involved than that of the Hodge decomposition in (11), they still allow decomposing a topological signal $\mathbf{x}_{\mathcal{X}}$ into the sum of a signal $\tilde{\mathbf{x}}_{\mathcal{X}}^{(d)} \in \text{span}(\mathbf{D}_{\mathcal{X}}^{(d)})$, a signal $\tilde{\mathbf{x}}_{\mathcal{X}}^{(u)} \in \text{span}(\mathbf{D}_{\mathcal{X}}^{(u)})$, and a harmonic signal $\tilde{\mathbf{x}}_{\mathcal{X}} \in \text{kernel}(\mathbf{D}_{\mathcal{X}})$, i.e.,

$$\mathbf{x}_k = \underbrace{\tilde{\mathbf{x}}_{\mathcal{X}}^{(d)}}_{\text{(a) joint gradient component}} + \underbrace{\tilde{\mathbf{x}}_{\mathcal{X}}^{(u)}}_{\text{(b) joint curl component}} + \underbrace{\tilde{\mathbf{x}}_{\mathcal{X}}}_{\text{(c) joint harmonic component}}. \tag{16}$$

We refer the reader to [62] for more details on the Dirac decomposition of topological signals.

3. Spectral processing

In this section, we show how the Hodge and Dirac decomposition can be used for a spectral analysis of topological signals. We first recall the basic concepts of the graph Fourier transform in Section 3.1 and then define the more general topological Fourier transform in Section 3.2. We conclude by discussing the spectral duality for joint topological complex signals in Section 3.3.

3.1. Graph fourier transform

Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with graph Laplacian $\mathbf{L}$ and graph signal $\mathbf{x}$, the graph Fourier transform (GFT) of signal $\mathbf{x}$ is the signal projection onto the Laplacian eigenspace. More specifically, given the eigendecomposition $\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{H}}$, the GFT of $\mathbf{x}$ is $\hat{\mathbf{x}} = \mathbf{U}^{\mathrm{H}}\mathbf{x}$. The eigenvectors $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_N]$ serve as the spectral basis expansion for the graph signal $\mathbf{x}$, and the GFT coefficient $\hat{x}_i$ is the weight indicating how much eigenvector $\mathbf{u}_i$ contributes to represent the signal. Following the analogy with the classical Fourier transform [9], the eigenvalues in $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_N)$ contain the so-called graph frequencies.

The Fourier notion of the projection $\hat{\mathbf{x}} = \mathbf{U}^{\mathrm{H}}\mathbf{x}$ comes from the fact that we can view each eigenvector $\mathbf{u}_i = [u_{i1}, \dots, u_{iN}]^\top$ as a graph signal and analyze its variability w.r.t. the graph $\mathcal{G}$. One way to do this is via the quadratic variation $\text{QV}(\mathbf{u}_i) = \mathbf{u}_i^{\mathrm{H}}\mathbf{L}\mathbf{u}_i = \lambda_i$, which indicates how smooth $\mathbf{u}_i$ is over the graph $\mathcal{G}$. Thus, we can sort the eigenvectors based on their variability $0 = \text{QV}(\mathbf{u}_1) \leq \text{QV}(\mathbf{u}_2) \leq \dots \leq \text{QV}(\mathbf{u}_N)$, which implies an ordering of the graph frequencies $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$. As a consequence, we refer to the eigenvalues λ_i close to 0 as low frequencies and to eigenvalues $\lambda_i \gg 0$ as high frequencies. Hence, the GFT coefficient $\hat{x}_i$ indicates how much the eigenvector signal basis $\mathbf{u}_i$ contributes to the variability of the graph signal $\mathbf{x}$. We shall discuss next that a similar, yet slightly more involved, Fourier analysis can be derived also for topological signals of any order.

3.2. Topological fourier transform

As for the graph Laplacian, any Hodge Laplacian of order k enjoys an eigendecomposition of the form

$$\mathbf{L}_k = \mathbf{L}_k^{(d)} + \mathbf{L}_k^{(u)} = \mathbf{B}_k^\top\mathbf{B}_k + \mathbf{B}_{k+1}\mathbf{B}_{k+1}^\top = \mathbf{U}_k\boldsymbol{\Lambda}_k\mathbf{U}_k^\top \tag{17}$$

with orthogonal eigenvector matrix $\mathbf{U}_k = [\mathbf{u}_{k,1}, \dots, \mathbf{u}_{k,N_k}] \in \mathbb{R}^{N_k \times N_k}$ and eigenvalue matrix $\boldsymbol{\Lambda}_k = \text{diag}(\lambda_{k,1}, \dots, \lambda_{k,N_k})$. Then, the *topological Fourier transform* (TFT) of a signal $\mathbf{x}_k$ is given by the projection onto the eigenvectors $\mathbf{U}_k$, i.e., $\hat{\mathbf{x}}_k = \mathbf{U}_k^\top\mathbf{x}_k$ [44]. As for the GFT, the ith entry of $\hat{\mathbf{x}}_k$, i.e., $\hat{x}_{k,i}$ represents the weight of eigenvector $\mathbf{u}_{k,i}$ in expressing signal $\mathbf{x}_k$. The inverse TFT is given by $\mathbf{x}_k = \mathbf{U}_k\hat{\mathbf{x}}_k$. Notice that the GFT discussed in Section 3.1 is the special case of the TFT for $k = 0$. While in principle the TFT and the GFT are quite similar, they differ substantially in terms of interpretation as we elaborate in the sequel.

Interpreting the TFT. The Hodge Laplacian eigendecomposition has a correspondence with the Hodge decomposition in (11). More specifically, from [60, Proposition 4] it is possible to rearrange the eigenvectors in $\mathbf{U}_k$ and eigenvalues in $\boldsymbol{\Lambda}_k$ respectively as:

$$\mathbf{U}_k = [\mathbf{U}_{k,\text{g}}, \mathbf{U}_{k,\text{c}}, \mathbf{U}_{k,\text{h}}] \quad \text{and} \quad \boldsymbol{\Lambda}_k = \text{blkdiag}(\boldsymbol{\Lambda}_{k,\text{g}}, \boldsymbol{\Lambda}_{k,\text{c}}, \boldsymbol{\Lambda}_{k,\text{h}}). \tag{18}$$

Focusing again to edge signals, $k = 1$, we can observe the following:

(a) *Gradient space and gradient frequencies:* The eigenvectors $\mathbf{U}_{1,\text{g}} \in \mathbb{R}^{N_1 \times N_\text{g}}$ span the gradient space $\text{span}(\mathbf{B}_1^\top)$ with dimension N_g. Hence, the component $\hat{\mathbf{x}}_{1,\text{g}} = \mathbf{U}_{1,\text{g}}^\top\mathbf{x}_1$ represents the gradient component of the TFT. The eigenvectors $\mathbf{u}_\text{g} \in \mathbf{U}_{1,\text{g}}$ also carry a notion of variability via the quadratic variation w.r.t. the Hodge Laplacian $\mathbf{L}_1$, i.e.,

$$\text{QV}(\mathbf{u}_\text{g}) = \mathbf{u}_\text{g}^\top\mathbf{L}_1\mathbf{u}_\text{g} = \|\mathbf{B}_1\mathbf{u}_\text{g}\|_2^2 + \|\mathbf{B}_2^\top\mathbf{u}_\text{g}\|_2^2 = \|\mathbf{B}_1\mathbf{u}_\text{g}\|_2^2 = \lambda_\text{g} \tag{19}$$

where $\mathbf{B}_2^\top\mathbf{u}_\text{g} = \mathbf{0}$ since $\mathbf{u}_\text{g}$ is a gradient flow. Thus, the eigenvalue λ_g is the squared ℓ_2-norm of the divergence $\mathbf{B}_1\mathbf{u}_\text{g}$ of the corresponding gradient eigenvector. We can use this quadratic variation to sort the gradient eigenvectors, and correspondingly, the divergence variation of the corresponding eigenvectors in an ascending order $0 < \text{QV}(\mathbf{u}_{\text{g},i}) \leq \text{QV}(\mathbf{u}_{\text{g},j})$ implying an ordering of the eigenvalues $0 < \lambda_{\text{g},i} < \lambda_{\text{g},j}$ for $i, j \in \{1, \dots, N_\text{g}\}$. In turn, this ordering carries the same meaning of variability as for the graph frequencies but now it measures the variability of the total divergence. We refer to the eigenvalues λ_g associated to the gradient eigenvectors $\mathbf{U}_{1,\text{g}}$ as the *gradient frequencies*.

(b) *Curl space and curl frequencies:* Analogously, the eigenvectors $\mathbf{U}_{1,\text{c}} \in \mathbb{R}^{N_1 \times N_\text{c}}$ span the curl space $\text{span}(\mathbf{B}_2)$ with dimension N_c.

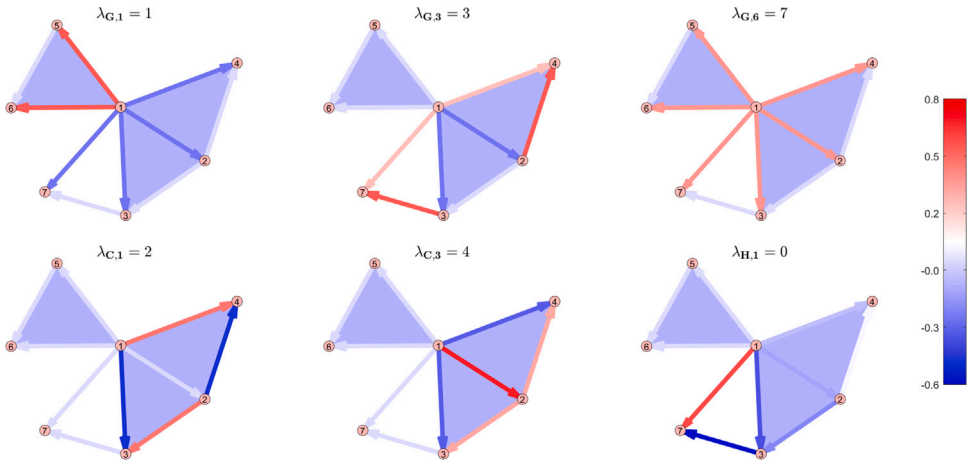


Fig. 3. Eigenvectors of the $\mathbf{L}_1$ Hodge Laplacian shown as edge flow signal along with their respective eigenvalues. (top-row) gradient eigenvectors with increasing total divergence. (bottom row) curl eigenvectors with increasing total curl variation, and the harmonic eigenvector. The latter is localized around the hole in the simplicial complex.

The projection $\hat{\mathbf{x}}_{1,\mathrm{c}} = \mathbf{U}_{1,\mathrm{c}}^{\top}\mathbf{x}_1$ represents the curl component of the TFT, where the eigenvectors $\mathbf{u}_{\mathrm{c}} \in \mathbf{U}_{1,\mathrm{c}}$ do carry a notion of variability that is different from that seen above. Using again the quadratic variation w.r.t. the $\mathbf{L}_1$ Hodge Laplacian we have

$$\mathrm{QV}(\mathbf{u}_{\mathrm{c}}) = \mathbf{u}_{\mathrm{c}}^{\top}\mathbf{L}_1\mathbf{u}_{\mathrm{c}} = \|\mathbf{B}_1\mathbf{u}_{\mathrm{c}}\|_2^2 + \|\mathbf{B}_2^{\top}\mathbf{u}_{\mathrm{c}}\|_2^2 = \|\mathbf{B}_2^{\top}\mathbf{u}_{\mathrm{c}}\|_2^2 = \lambda_{\mathrm{c}} \tag{20}$$

where $\mathbf{B}_1\mathbf{u}_{\mathrm{c}} = \mathbf{0}$ since $\mathbf{u}_{\mathrm{c}}$ is a curl flow. Now, the eigenvalue λ_{c} is the squared ℓ_2–norm of the total curl $\mathbf{B}_2^{\top}\mathbf{u}_{\mathrm{c}}$ of the corresponding curl eigenvector. Using this interpretation of the quadratic variation, we can sort separately the curl eigenvectors, and correspondingly, the total curl variation of the corresponding eigenvectors in an ascending order $0 < \mathrm{QV}(\mathbf{u}_{\mathrm{c},i}) \leq \mathrm{QV}(\mathbf{u}_{\mathrm{c},j})$ implying an ordering of the curl eigenvalues $0 < \lambda_{\mathrm{c},i} < \lambda_{\mathrm{c},j}$ for $i, j \in \{1, \dots, N_{\mathrm{c}}\}$. We refer to the eigenvalues λ_{c} associated to the curl eigenvectors $\mathbf{U}_{1,\mathrm{c}}$ as the *curl frequencies*.

(c) *Harmonic space and harmonic frequencies:* Finally, the eigenvectors $\mathbf{U}_{1,\mathrm{h}} \in \mathbb{R}^{N_1 \times N_{\mathrm{h}}}$ span the harmonic space kernel$(\mathbf{L}_k)$ with dimension N_{h}. The projection $\hat{\mathbf{x}}_{1,\mathrm{h}} = \mathbf{U}_{1,\mathrm{h}}^{\top}\mathbf{x}_1$ represents the harmonic component of the TFT, where the eigenvectors $\mathbf{u}_{\mathrm{h}} \in \mathbf{U}_{1\mathrm{h}}$ are all associated to the zero eigenvalue since

$$\mathrm{QV}(\mathbf{u}_{\mathrm{h}}) = \mathbf{u}_{\mathrm{h}}^{\top}\mathbf{L}_1\mathbf{u}_{\mathrm{h}} = \|\mathbf{B}_1\mathbf{u}_{\mathrm{h}}\|_2^2 + \|\mathbf{B}_2^{\top}\mathbf{u}_{\mathrm{h}}\|_2^2 = 0 = \lambda_{\mathrm{h}}. \tag{21}$$

Consequently, we will refer to the eigenvalues $\lambda_{\mathrm{h}} = 0$ as the *harmonic frequencies*. These harmonic frequencies correspond to a global conservative flow, i.e., a flow signal that does not have any gradient or curl component.

The above discussion shows that the notion of low and high frequency in a SC are only meaningful within a certain type. An illustration of the latter is shown in Fig. 3. This behavior is a unique characteristic of topological spaces that is not conventionally seen in graph and discrete signal processing. We refer the reader to [44,46,59,60,63] for more details on the latter.

Remark 2 (*Hodge Decomposition and GFT*)**.** Notice that the graph Laplacian can also be written as $\mathbf{L} = \mathbf{B}_1\mathbf{B}_1^{\top}$, which implies that the space of all graph signals, i.e., $k = 0$, enjoys also a Hodge decomposition of the form $\mathbb{R}^{N_0} = \mathrm{span}(\mathbf{B}_1) \oplus \mathrm{kernel}(\mathbf{L})$. Since from a topological perspective, the case $k = 0$ implies that nodes are connected only via the *upper* Laplacian, the space span$(\mathbf{B}_1)$ is the analogous of the curl space in (11), whereas kernel$(\mathbf{L})$ is the harmonic space. Consequently, the GFT accounts for the projection of the graph signals onto these subspaces, where the eigenvectors $\mathbf{U}$ associated with a non-zero eigenvalue span the curl space span$(\mathbf{B}_1)$ and those associated with a zero eigenvalue span the harmonic space kernel$(\mathbf{L})$. □

3.3. Joint spectral spaces

As for the Hodge Laplacians, the Dirac operator enjoys the eigendecomposition:

$$\mathbf{D}_{\mathcal{X}} = \mathbf{D}_{\mathcal{X}}^{(d)} + \mathbf{D}_{\mathcal{X}}^{(u)} = \mathbf{U}_{\mathcal{X}}\boldsymbol{\Lambda}_{\mathcal{X}}\mathbf{U}_{\mathcal{X}}^{\top}. \tag{22}$$

The matrix of eigenvectors $\mathbf{U}_{\mathcal{X}}$ of $\mathbf{D}_{\mathcal{X}}$ can be written as:

$$\mathbf{U}_{\mathcal{X}} = \left[\mathbf{U}_{\mathcal{X}}^{(d)}, \mathbf{U}_{\mathcal{X}}^{(u)}, \tilde{\mathbf{U}}_{\mathcal{X}}\right] \tag{23}$$

where $\mathbf{U}_{\mathcal{X}}^{(d)}, \mathbf{U}_{\mathcal{X}}^{(u)}$ are the matrices of non-zero eigenvectors of $\mathbf{D}_{\mathcal{X}}^{(d)}$ and $\mathbf{D}_{\mathcal{X}}^{(u)}$, respectively, whose columns respectively span the joint gradient space span$(\mathbf{D}_{\mathcal{X}}^{(d)})$ and the joint curl space span$(\mathbf{D}_{\mathcal{X}}^{(u)})$. The columns of $\tilde{\mathbf{U}}_{\mathcal{X}}$ span the joint harmonic space kernel$(\mathbf{D}_{\mathcal{X}})$. Since the Dirac operator matrix is a block matrix composed of the incidence matrices defining the topological complex, it is possible to relate the eigendecomposition in (23) with the singular value decompositions of the incidence matrices $\mathbf{B}_k$, as well as the harmonic eigenvectors of the Hodge-Laplacian. Likewise, also the eigenvalues in $\boldsymbol{\Lambda}_{\mathcal{X}}$ can be grouped into those related to $\mathbf{U}_{\mathcal{X}}^{(d)}, \mathbf{U}_{\mathcal{X}}^{(u)}$, and $\tilde{\mathbf{U}}_{\mathcal{X}}$ and they can in turn be related to the eigenvalues of the Hodge Laplacian and singular values of the incidence matrices. Specifically, eigenvectors associated with positive eigenvalues of $\mathbf{D}_{\mathcal{X}}^{(u)}$ and $\mathbf{D}_{\mathcal{X}}^{(d)}$ capture configurations where signals defined on the 1 and 0-simplices, and on the 2 and 1-simplices, respectively, are aligned with the action of the boundary operators. On the other hand, eigenvectors associated with negative eigenvalues, capture configurations where signals defined on the 1 and 0-simplices, and on the 2 and 1-simplices, respectively, are antialigned with the action of the boundary operators. By picking out the eigenvectors associated to positive (negative) eigenvalues of the Dirac operator, one can thus find appropriate basis vectors for aligned (anti-aligned) simplicial signals. The above theoretical framework and interpretation enables and motivates the design of filters acting on joint spectral spaces. We refer the reader to [62] for further details on the matter.

As for the topological Fourier transform, we can compute the spectral projections of simplicial signals onto the Dirac operator eigenspace. Specifically, letting $\mathbf{x} = [\mathbf{x}_0^{\top}, \mathbf{x}_1^{\top}, \mathbf{x}_2^{\top}]^{\top} \in \mathbb{R}^{N+E+T}$ be the vector collecting node, edge, and triangle signals, the topological Dirac Fourier transform reads as $\hat{\mathbf{x}} = \mathbf{U}_{\mathcal{X}}^{\top}\mathbf{x}$, where the ith entry represents the weight of the ith eigenvector of $\mathbf{U}_{\mathcal{X}}$ in expressing signal $\mathbf{x}$. The inverse topological Dirac Fourier transform readily writes as $\mathbf{x} = \mathbf{U}_{\mathcal{X}}\hat{\mathbf{x}}$.

4. Current advances

The goal of this section is twofold. First, it aims to bring together the current methodological advances on topological signal processing and

learning. Second, it highlights how these advances tie to the concepts discussed in Sections 2 and 3. More specifically, in Section 4.1 we discuss topological signal representation advances, whereas in Section 4.2 we focus on how to learn topological structures from data. Then, in Section 4.3 we discuss topological filters and in Section 4.4 sampling and reconstruction strategies that take into account the topological complex structure. Section 4.6 discusses the case of processing spatiotemporal topological signals, and finally Section 4.5 discusses how these techniques can be useful when building machine learning models for topological structures and signals.

4.1. Signal representation

A spectral theory for signals defined over simplicial complexes was first introduced in [44], utilizing algebraic topology tools such as higher-order Laplacian matrices. The framework identifies eigenvectors of these Laplacians as a natural basis for representing and analyzing these signals, and enables the Hodge decomposition of signals into meaningful components via spectral analysis. This decomposition identifies signals components like harmonic, gradient, and curl flows, which can then be manipulated or analyzed separately. Other studies extended the approach from [44] to regular cell complexes [57,58], whereas the work in [59] generalized the concept of cell complexes to include hollow cells. These approaches to filter data on simplicial complexes operate independently on each order. To formally incorporate interdependencies among simplices of different orders, the Dirac operator was introduced in [61] and has been employed as a robust and theoretically grounded approach to jointly process signals across consecutive orders of simplices [62].

A key challenge in signal processing is *sparse signal representation* [64], which focuses on creating overcomplete dictionaries of atoms to represent signals as linear combinations of only a few atoms from the dictionary. Two main approaches have proven effective for both Euclidean and graph signals. The first involves analytical dictionaries, which are structured and derived from mathematical models, designed based on the specific domain while assuming a certain class of signals, such as Fourier transforms, wavelets, or curvelets. The second approach is learnable dictionaries, which are unstructured and learned from a set of training signals [65]. A crucial trade-off exists between these methods: analytical dictionaries are typically faster to implement but are less resilient to model mismatches, whereas learnable dictionaries are more robust across different signal classes but involve higher complexity due to the training phase. In the context of topological dictionaries, analytical methods have been extended from graph signals, where the topological Fourier modes provide a natural basis for signal representation [34]. However, since Fourier modes are often non-sparse and inefficient for representing localized signals, other methods have been developed. In [66], a dictionary of basis vector maximally concentrated in the discrete vertex and frequency sets were introduced, generalizing the seminal work of Slepian from continuous time to graphs. The work in [67] proposed a family of wavelets for simplicial signals that respect the Hodge decomposition. Then, the authors of [68] introduced topological Slepians, which are signals maximally concentrated on the topological domain and perfectly localized in the spectral domain. Specifically, let us introduce two localization operators acting onto an edge concentration set, say $\mathcal{S}$, and onto a frequency concentration set, say $\mathcal{F}$, respectively. The edge-limiting operator onto the edge set $\mathcal{S}$ is defined as the matrix $\mathbf{C}_{\mathcal{S}} \in \mathbb{R}^{E\times E}$ given by:

$$\mathbf{C}_{\mathcal{S}} = \text{diag}(\mathbf{1}_{\mathcal{S}}), \tag{24}$$

where $\mathbf{1}_{\mathcal{S}} \in \mathbb{R}^{E}$ is a vector having ones in the index positions specified in $\mathcal{S}$, and zero otherwise. Clearly, from (24), an edge signal $\mathbf{x}_1$ is perfectly localized onto the set $\mathcal{S}$ if $\mathbf{C}_{\mathcal{S}}\mathbf{x}_1 = \mathbf{x}_1$. Similarly, the frequency limiting operator can be defined as:

$$\mathbf{F}_{\mathcal{F}} = \mathbf{U}_1 \,\text{diag}(\mathbf{1}_{\mathcal{F}})\,\mathbf{U}_1^{\top}, \tag{25}$$

which represents an ideal band-pass filter over the frequency set $\mathcal{F}$. Clearly, an edge signal is perfectly localized over the bandwidth $\mathcal{F}$ if $\mathbf{F}_{\mathcal{F}}\mathbf{x}_1 = \mathbf{x}_1$. The topological Slepians are the set of orthonormal vectors that are maximally concentrated over the edge set $\mathcal{S}$, and perfectly localized onto the bandwidth $\mathcal{F}$. Mathematically, topological Slepians are given by the solution of the following optimization problem:

$$\begin{aligned} \boldsymbol{\psi}_i = \underset{\boldsymbol{\psi}_i}{\text{argmax}} \;& \|\mathbf{C}_{\mathcal{S}}\boldsymbol{\psi}_i\|_2^2 \\ \text{subject to } & \|\boldsymbol{\psi}_i\| = 1, \quad \mathbf{F}_{\mathcal{F}}\boldsymbol{\psi}_i = \boldsymbol{\psi}_i, \\ & < \boldsymbol{\psi}_i, \boldsymbol{\psi}_j >= 0, \quad j = 1, \dots, i-1, \text{ if } i > 1, \end{aligned} \tag{26}$$

for all $i = 1, \dots, E$. The solution of problem (26) is given by the eigenvectors of the matrix operator $\mathbf{F}_{\mathcal{F}}\mathbf{C}_{\mathcal{F}}\mathbf{F}_{\mathcal{F}}$. Topological Slepians can be used to build localized dictionaries for signals defined over simplicial or cell complexes, which theoretically provide non-degenerate frames [68]. Other generalizations are illustrated in the work [69], which exploits multiscale basis dictionaries on simplicial complexes based on generalized Haar–Walsh and hierarchical graph Laplacian eigen transforms, to design multiscale hodge scattering neural networks.

On the other side, the paper in [70] introduced a dictionary learning algorithm for sparse representation of signals on regular cell complexes. Utilizing Hodge theory, the dictionary is structured as polynomials of Hodge Laplacians. Specifically, the dictionary consists of the concatenation of P convolutional filters, i.e.,:

$$\mathbf{H} = \{\mathbf{H}_1(\mathbf{L}_k), \mathbf{H}_2(\mathbf{L}_k), \dots, \mathbf{H}_P(\mathbf{L}_k)\} \in \mathbb{R}^{N_k \times PN_k},$$

where each sub-dictionary is expressed as:

$$\mathbf{H}_i(\mathbf{L}_k) := \sum_{t=0}^{T_d} h_{it}^{(d)} (\mathbf{L}_k^{(d)})^t + \sum_{t=0}^{T_u} h_{it}^{(u)} (\mathbf{L}_k^{(u)})^t \tag{27}$$

where $\{h_{it}^{(d)}\}, \{h_{it}^{(u)}\}$ are the filter coefficients as we discuss more in detail in Section 4.3. The dictionary learning problem, combining topological filter coefficients and sparse representation, is then efficiently solved via an alternating optimization approach.

4.2. Learning topologies from data

This section deals with discovering topological relationships from data. Within the field of graph signal processing, many results exist on edge-wise topology identification from node signals. Here, a link between the data and the graph is established based on concepts like smoothness, structural equation models, Gaussian graphical models, and spectral templates to name a few [71]. Discovering higher-order topological relationships from nodal (and potentially higher-order) observations is less obvious.

Focusing on simplicial and cell complexes, the problem statement is finding a complex that can explain the topologically data that is (fully or partially) available on different levels. In [44], a hierarchical method has been proposed to infer the structure of one layer, assuming knowledge of the lower layers. For instance, if flow data (edge signals) are available, which implicitly assumes that a graph is known (and thus $\mathbf{L}_0$ or $\mathbf{B}_1$ are known), we can infer the structure of a simplicial or cell complex of order two. Since $\mathbf{B}_1$ is already known, we know the lower Laplacian $\mathbf{L}_1^{(d)}$ and only the upper Laplacian $\mathbf{L}_1^{(u)}$ or $\mathbf{B}_2$ needs to be estimated. Referring back to the Hodge decomposition, it is clear that $\mathbf{B}_2$ is only needed to explain the data if the edge signals have a curl component. So first the flow data is projected onto the orthogonal complement of the gradient space. And if there is enough energy left, a fixed number of triangles is filled in such a way that some total variation measure (or smoothness measure) of the (gradient-free) flow data over the upper Laplacian $\mathbf{L}_1^{(u)}$ is minimized. This measure is similar to the well-known smoothness measure of node signals over the Laplacian $\mathbf{L}_0$. Extensions to cell complexes have been proposed in [59]. In [72], on the other hand, triangles are filled based on maximizing the

fit of the (gradient-free) flow data to the curl space, which is different from the smoothness measure considered in [44,59]. In [73,74], the observations over the elements of a simplicial complex are modeled as random variables and methods to infer the structure of the complex from data are proposed. The focus of [73,74] is on the edge flows, but the approach can be generalized to higher order structures. The model in [73] assumes smooth Gaussian node and triangle random variables plus a white Gaussian harmonic random variable. Since the observation of edge variables implicitly assumes that the graph skeleton is known (and thus $\mathbf{L}_0$ or $\mathbf{B}_1$ is known), the problem that maximizes the posterior probability then basically boils down to a smoothness-regularized edge flow fitting problem, which can be contrasted to the maximum likelihood style fitting problem of [72] where no smoothness priors are assumed. In [74] it was shown that, if the gradient, curl and harmonic random variables are uncorrelated, but with arbitrary covariance matrices, a principal component analysis (PCA) performed on the covariance matrix of the edge variables still satisfies the Hodge decomposition. Furthermore, simple algorithms have been proposed to infer the structure of $\mathbf{B}_2$, thus filling the triangles in the simplicial complex, based on the covariance matrix of the observed random vector projected onto the space orthogonal to the gradient space of the edge flow. A generalized version of the graphical lasso method [75] to random edge variables was also proposed, to estimate the partial correlation coefficients between edge variables and then infer conditional independencies among the edge variables.

Non-hierarchical methods have been proposed in [76–78]. The work [76] is probably closest to the previously cited papers and jointly estimates the edges and triangles, i.e., $\mathbf{B}_1$ and $\mathbf{B}_2$, using full nodal data and partial flow data (an extension to partial nodal data is straightforward). The properties that are exploited in [76] are that the node signals are smooth over the Laplacian $\mathbf{L}_0$ and the edge signals have a small curl (i.e., they are smooth over the upper Laplacian $\mathbf{L}_1^{(u)}$). Similarly, it can be considered that the edge signals have a small divergence (i.e., they are smooth over the lower Laplacian $\mathbf{L}_1^{(u)}$). Or even a weighted combination of the two edge smoothness measures can be used in the objective function. On the other hand, [77] relies on the Volterra model to estimate a second-order hypergraph using only nodal information. A least squares cost can be used to fit the nodal observations to the first- and second-order Volterra kernels. Furthermore, the constraint required to force the hypergraph to be a simplicial complex, i.e., to satisfy the inclusion property, can be relaxed to a convex constraint. A similar yet probabilistic Volterra-like model has been considered in [78] for modeling social contagion dynamics. It can be viewed as an extension of the Ising model from pairwise to higher-order interactions. From observed binary infection data, the probabilities that a node or a pair of nodes is connected to another node is estimated using the expectation–maximization algorithm. Those probabilities can then be used to recover edges and triangles of a simplicial complex.

Finally, the paper in [79] presents latent higher-order topology inference. The work introduces the differentiable cell complex module (DCM), an architecture that dynamically learns cell complexes by first inferring the 1-skeleton (a graph) and then identifying higher-order cells (polygons) to include. These steps leverage message passing and a sparse sampling technique. The DCM is trained end-to-end and achieves significant accuracy improvements on both homophilic and heterophilic datasets, demonstrating robust performance even when input graphs poorly represent the data.

4.3. Filtering

In Section 3, we discussed the spectral processing of topological signals by means of either the Hodge or Dirac decomposition. This duality between the spectrum and the topological domain opens the doors to a different signal representation and processing perspectives as discussed in Section 4.1. Alternatively, we can spectrally process topological signals via local filtering approaches without resorting to the spectral domain. The latter is achieved in two ways: first, by extending the concept of convolutional filtering to the topological domain; and second via regularization techniques.

Convolutional filtering. The spectral filtering of topological signal is computationally expensive as we need to compute the eigendecomposition of the Hodge or Dirac matrices. To avoid the latter, [60] developed topological filters by following the principle of convolution to process the kth signal $\mathbf{x}_k$ and produce the output $\mathbf{y}_k$ as:

$$\mathbf{y}_k := \mathbf{H}(\mathbf{L}_k)\mathbf{x}_k := \underbrace{\sum_{t=0}^{T_d} h_{kt}^{(d)} (\mathbf{L}_k^{(d)})^t \mathbf{x}_k}_{\mathbf{H}_k(\mathbf{L}_k^{(d)})} + \underbrace{\sum_{t=0}^{T_u} h_{kt}^{(u)} (\mathbf{L}_k^{(u)})^t \mathbf{x}_k}_{\mathbf{H}_k(\mathbf{L}_k^{(u)})} \tag{28}$$

where $h_{kt}^{(d)}$ and $h_{kt}^{(u)}$ are the filter parameters associated to the lower and upper Hodge Laplacians, respectively. Here, $(\mathbf{L}_k^{(\cdot)})^t$ is meant as the tth power of the kth of the respective Laplacian matrix which are computed up to orders T_d or T_u. Since the Hodge Laplacian matrices are sparse, operations of the form $(\mathbf{L}_k^{(\cdot)})^t \mathbf{x}_k$ can be computed recursively as $(\mathbf{L}_k^{(\cdot)})^t \mathbf{x}_k = \mathbf{L}_k^{(\cdot)} \big((\mathbf{L}_k^{(\cdot)})^{t-1} \mathbf{x}_k \big)$ to reduce the computation cost. As for the convolutional filters in time, images and graphs [8], the operation in (28) builds upon the shift-and-sum principle by propagating signal $\mathbf{x}_k$ to neighboring simplices via either the upper or lower adjacencies. This locality of information processing is the reason why the dictionaries in (27) are topological local dictionaries.

The nice property of the operation in (28) is that is preserves a spectral analogy. In fact the lower Laplacian filtering component $\mathbf{H}_k(\mathbf{L}_k^{(d)})$ processes the gradient component of the topological Fourier transform, whereas the upper Laplacian filtering component $\mathbf{H}_k(\mathbf{L}_k^{(u)})$ processes the curl component of the TFT. This processing is coupled with that of the harmonic part via the zero-th power of the Laplacian matrices. To provide an independent processing of the Harmonic part, the work in [55] modified (28) to:

$$\mathbf{y}_k := \mathbf{H}(\mathbf{L}_k)\mathbf{x}_k := \sum_{t=1}^{T_d} h_{kt}^{(d)} (\mathbf{L}_k^{(d)})^t \mathbf{x}_k + \sum_{t=1}^{T_u} h_{kt}^{(u)} (\mathbf{L}_k^{(u)})^t \mathbf{x}_k + (\mathbf{I} - \epsilon \mathbf{L}_k)^{T_h} \mathbf{x}_k \tag{29}$$

where the polynomials start from $t = 1$ and the term $(\mathbf{I} - \epsilon \mathbf{L}_k)^{T_h}$ projects the input $\mathbf{x}_k$ onto the harmonic space as the integer $T_h \to \infty$, if we set parameter $0 < \epsilon < 2/\lambda_{\max}(\mathbf{L}_k)$.

When signals across multiple topological levels are present, we can readily extend operations (28)–(29) onto filterbanks that account for the adjacent simplicial signals [80]. To illustrate the latter, let us focus on the edge flow signals $\mathbf{x}_1$ and consider available both node signal $\mathbf{x}_0$ and triangle signals $\mathbf{x}_2$. We can then compute the output edge flow as:

$$\mathbf{y}_1 = \underbrace{\mathbf{H}(\mathbf{L}_1)\mathbf{x}_1}_{\text{edge} \to \text{edge}} + \underbrace{\mathbf{H}(\mathbf{L}_1^{(d)})\mathbf{B}_1^\top \mathbf{x}_0}_{\text{node} \to \text{edge}} + \underbrace{\mathbf{H}(\mathbf{L}_1^{(u)})\mathbf{B}_2 \mathbf{x}_2}_{\text{triangle} \to \text{edge}} \tag{30}$$

where the node signal $\mathbf{x}_0$ and triangle signal $\mathbf{x}_2$ are transformed via the respective incidence relations onto edge flows and then processed via respective filters. Notice that the operation in (30) can be easily extended to processing also node and triangle signals by accounting for the edge flows in a similar manner. In all these cases, the filters have different parameters so that to allow a higher flexibility when processing the different spectral components. In case the parameters are learned from limited data, we can rely on the Dirac operator (13) and process jointly the topological signal $\mathbf{x}_{\mathcal{X}} = [\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2]$ [cf. (4)] as:

$$\mathbf{y}_{\mathcal{X}} = \mathbf{H}(\mathbf{D}_{\mathcal{X}})\mathbf{x}_{\mathcal{X}}. \tag{31}$$

Here, $\mathbf{H}(\mathbf{D}_{\mathcal{X}})$ is a matrix polynomial in the form (28) or (29) where the kth Hodge Laplacian matrix $\mathbf{L}_k$ is substituted by the Dirac matrix $\mathbf{D}_{\mathcal{X}}$. In this case, the same scalar coefficients are used to process all simplicial signals, which allows a spectral analysis of the filter frequency response

via the Dirac decomposition [cf. Section 3.3]; refer to [55, Appendix A] for more details on the latter.

Filtering by regularization. An alternative way to process topological signals with a spectral equivalence is to rely on regularization techniques in a form akin to the popular Tikhonov regularization or trend filtering for graph signals [8]. To be more specific, let us consider the task of edge flow signal $\mathbf{x}_1$ reconstruction from partial noisy measurements $\mathbf{f}_1 = \mathbf{M}(\mathbf{x}_1 + \mathbf{n})$, where $\mathbf{M} \in \{0,1\}^{M_1 \times N_1}$ is a masking matrix sampling $M_1 \ll N_1$ edge flows and $\mathbf{n}_1$ is a zero-mean Gaussian noise. Then, we can estimate the edge flow signal by solving:

$$\underset{\mathbf{x}_1 \in \mathbb{R}^{N1}}{\operatorname{argmin}} \ \|\mathbf{M}(\mathbf{f}_1 - \mathbf{x}_1)\|_2^2 + \alpha_p \|\mathbf{B}_1 \mathbf{x}_1\|_p^p + \beta_q \|\mathbf{B}_2^\top \mathbf{x}_1\|_q^q \tag{32}$$

where $\alpha_p, \beta_p > 0$ are scalars and $p, q \in \{1,2\}$. The prior information about the edge flow here is put on the regularization terms, where $\|\mathbf{B}_1 \mathbf{x}_1\|_p^p$ penalizes the divergence of the edge flow to be either of low magnitude ($p = 2$) or sparse ($p = 1$), whereas the second term $\|\mathbf{B}_2^\top \mathbf{x}_1\|_q^q$ penalizes the curl component of the edge flows. Variants of problem (32) have been discussed in [81–83]. As for the convolutional filtering, the regularizer problem can be extended to account for multiple signal levels via the Dirac operator by penalizing either smooth or sparse differences across all the simplicial complex [62].

4.4. Sampling and reconstruction

In [44], the sampling and reconstruction problem of graph signals is extended to simplicial complexes. More specifically, an edge signal $\mathbf{x}_1$ is subsampled over the edge space and is further assumed to be bandlimited in the spectrum of $\mathbf{L}_1$, i.e., it only consists of a limited number of eigenvectors of $\mathbf{L}_1$. It is then shown that perfect reconstruction of $\mathbf{x}_1$ can be obtained under a full rank condition of the product of the edge limiting and edge-frequency limiting operators. In [44], this single-layer sampling method is further extended to multi-layer sampling, where *additionally* a limited number of samples of the node signal $\tilde{\mathbf{x}}_0$ and/or the triangle signal $\tilde{\mathbf{x}}_2$ forming the edge signal $\mathbf{x}_1$ (see (12)) are available, and where furthermore $\tilde{\mathbf{x}}_0$ and $\tilde{\mathbf{x}}_2$ are assumed to be bandlimited in the spectrum of $\mathbf{L}_0$ and $\mathbf{L}_2$, respectively.

In [84], on the other hand, not only $\mathbf{x}_1$ is subsampled but also its higher-order diffusions (aggregations) over the edge Laplacian $\mathbf{L}_1$. The goal is then again to reconstruct this edge signal $\mathbf{x}_1$ but now by estimating the simplicial signals of one order lower $\tilde{\mathbf{x}}_0$ (a node signal) and one order higher $\tilde{\mathbf{x}}_2$ (a triangle signal) as well as the harmonic edge signal $\tilde{\mathbf{x}}_1$ that all three together yield the complete edge signal $\mathbf{x}_1$ through the Hodge decomposition (see again (12)). In order to do that, it is assumed that $\tilde{\mathbf{x}}_0$ and $\tilde{\mathbf{x}}_2$ are bandlimited in the spectrum of $\mathbf{L}_0$ and $\mathbf{L}_2$, respectively, and that also $\tilde{\mathbf{x}}_1$ is bandlimited over kernel($\mathbf{L}_1$). As proven though in [84], these bandlimitness conditions are equivalent to assuming bandlimitedness of $\mathbf{x}_1$ in the spectrum of $\mathbf{L}_1$. Hence, [84] is basically similar to the single-layer sampling method of [44] yet replacing edge sampling by edge aggregation sampling.

Results for a product cell structure which can be viewed as the Cartesian product of two simplicial complexes can be found in [85]. In that context, bandlimited edge signals on the product cell complex can be written as a direct sum of the Kronecker product of bandlimited edge and node signals on the factor simplicial complexes.

Gaussian processes that encapsulate the structure of a simplicial complex have been studied in [86]. Edge flows are then modeled as a function of edge features, where this function has a Gaussian process distribution with a kernel that depends on $\mathbf{L}_1$. Based on a limited number of observed edge flows and all edge features, the kernel parameters can be learned and the edge flows on the unobserved edges can be predicted. Furthermore, an online setting can be considered where the most informative edge flows (the ones with the largest uncertainty score) are sequentially sampled.

4.5. Bridging with topological machine learning

Upon defining the notions of topological support and local proximity between topological signals [cf. Section 2], as well as their spectral equivalence [cf. Section 3], it becomes fairly natural to leverage the latter as relational inductive biases [12] for developing machine learning solutions for topological signals. While different reviews [49,52] and position papers [50] are tailored to topological deep learning, we here review works that bridge the topological signal processing techniques with the broader domain of topological machine learning.

Topological neural networks (TNNs). TNNs architectures have been proposed to learn end-to-end representations for topological signals and have been inspired by three main principles: (i) message passing; (ii) convolution; and (iii) attention. Message passing architecture for simplicial and cell complexes have been developed in [87,88]. In these cases, messages within and across different simplices or cells are exchanged and update by using the incidence matrices (7) and the Hodge Laplacian. A principled analysis of the latter for edge flows has been proposed in [89] where the data invariance (such as permutation and orientation equivariance) are investigated to characterize the role of the simplicial proximities in each layer. A scalable solution inspired by the graph inception idea [90] is presented in [91].

Convolution-based architectures have a layered structure where in each layer a convolutional filter of the forms (28)–(31) nested into pointwise nonlinearity in the same spirit as the Euclidean and graph convolutional neural networks [92]. For example, the works in [53,54] proposed simplicial convolutional neural networks that operate at a single simplicial level, whereas extensions to signals present at different topological levels have been discussed in [93–95]. In these cases, the simplicial convolutional filters is used to learn both the embedding from multi-hop neighboring signal information in each layer and to link the latter with the Fourier spectral interpretation discussed in Section 3. A bridge between convolutional and message passing solutions can be established by limiting the convolutional filter order to one, which has been discussed in [96,97] in a form akin to the popular graph convolutional networks [98]. To improve the computational efficiency of these solutions, the work in [99] proposed a binarized simplicial convolutional neural network. Finally, the work in [100] introduced a convolution operation over the tangent bundle of Riemannian manifolds, defined via the exponentials of the Connection Laplacian operator, and the corresponding Tangent Bundle Neural Networks that operate on tangent bundle signals. The authors further propose a discretization procedure for Tangent Bundle Neural Network, demonstrating that their discrete implementation constitutes a principled variant of sheaf neural networks.

Self-attention mechanisms for simplicial neural networks were concurrently introduced in [101,102], with the aim of processing data components at different layers (e.g., nodes, edges, triangles, and so on), while learning how to weight both upper and lower neighborhoods of the given topological domain in a task-oriented fashion. This approach was then extended in [103] to handle a generalized higher-order domain called a combinatorial complex. Additionally, a simplicial-based attention mechanism tailored for heterogeneous graphs was presented in [104]. Finally, [55] introduces an attention-based neural architecture for processing data on simplicial complexes using masked self-attention layers, leveraging the simplicial Dirac operator to combine data from neighboring simplices of various orders (nodes, edges, triangles) in a task-specific manner, while ensuring permutation equivariance and simplicial-awareness. Interestingly, even if typically learned attentional shift operators are not Hodge Laplacians (i.e., they do not respect the Hodge decomposition), the work in [55] proposes an attentional variant enforcing a Hodge Laplacian structure over the learned shift operators.

A way to reduce the computational complexity while improving the learning performance of TNNs is to interleave the learning layers with

pooling strategies. However, to be effective and respect the topological structure of the data, these pooling layers should account for the simplicial and cell structures when coarsening the topological domain. An example is given in [105], where the authors introduced a general simplicial pooling layer consisting of three key operations: (i) a local aggregation step, which generates a meaningful summary of the input signals; (ii) a selection step, which identifies an appropriate subset of simplices; and (iii) a reduction step, which downsamples the input complex and the aggregated signals from step (i) based on the simplices selected in step (ii). By customizing steps (ii) and (iii), [105] proposed four distinct simplicial pooling layers that extend the widely used graph pooling strategies. Another interesting approach was then proposed in [106], which introduced a pooling layer for simplicial complexes that uses vertex partitions to create hierarchical representations, collapsing information through learned vertex cluster assignments and deterministically coarsening higher-dimensional simplices.

Gaussian processes (GPs). Gaussian processes are a class of statistical models that can quantify the uncertainty associated with their prediction. In essence, GPs are distribution over functions characterized by the mean function and a covariance kernel function that models the dependence between function values at two different inputs. When these inputs are topological signals, we would like to include their topological structure into the covariance matrix and define appropriate kernels to aid learning. The work in [107] discussed this case when said kernels have a spectral meaning and can distinguish between the gradient and curl components of the signal. A related discussion is present in [108] for cell complexes. Finally, topologically-aware GPs for edge flow reconstruction and simplicial closure are discussed in [86,109], respectively.

Data augmentation and self-supervised learning. The particular coupling between topological signals and their underlying structure can be used for data augmentation purposes; and in particular to train TNNs in a self-supervised manner. In this regard, the work in [110] builds upon the mixup technique [111] to generate artificial topologies that respect the simplicial complex structure. A topological-domain augmentation perspective is also discussed in [112] where random augmentations that respect the simplicial structure are used as data generation for self-supervised learning of TNNs. Differently, [113] discusses data augmentation techniques for SSL that respect spectral prior and uses the latter as an additional bias to aid learning.

4.6. Processing spatiotemporal signals

The methods discussed so far focus on time-invariant signals over topological spaces. However, in many real-world applications, observations typically consist of time series defined on various higher-order structures, including edges, triangles, and beyond [46,114] that influence each other in a non-trivial manner. As an example, in water networks, the water flow in the pipes can be better modeled as a time-varying 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 learning models that can capture their dependencies across both time and the higher-order structure. To account for the latter spatiotemporal coupling, a few works have emerged that either rely on recurrent models, adaptive filtering, or product spaces.

Recurrent models. One of the key recurrent models for multivariate data is the vector autoregressive model (VAR) [115–122]. While this model captures the temporal dependencies among the time-varying processes as a linear combination of their past realizations, it ignores the underlying network structure of the data and, consequently, do not grasp the inductive biases inherent in them. Given that the network topology affecting the process is typically sparse, a standard VAR model ignoring this structure fails to capitalize on the sparsity of interactions between the time series, suffering from the curse of dimensionality. This disadvantage can be alleviated with the introduction of network-based VAR models, such as the graph VAR (G-VAR) model [122], which 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. It accounts for the structure and captures the sparsity in the data via the so-called graph convolutional filters [123], enabling parameter sharing across vertices. This results in a computational complexity of a G-VAR model is linear in the number of time series, while that of standard approaches is quadratic [116,119,124–129]. Similarly, it is also possible to learn G-VAR models over signals defined exclusively on the edges of a graph (e.g. flows of a water network). However G-VAR models do not exploit higher-order dependencies.

In [130], SC-VAR models are introduced to represent 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. The SC-VAR model can capture spatio-temporal dependencies between signals defined on distinct simplicial levels, as we describe next.

To be more specific, let us consider a time varying vertex signal $\mathbf{x}_{0,t}$, edge signal $\mathbf{x}_{1,t}$, and triangle signal $\mathbf{x}_{2,t}$. By leveraging the adjacencies between simplices of different orders, we can process a $k-$simplicial signal $\mathbf{x}_{k,t-p}$ by first filtering it with a filter $\mathbf{H}_{k,p}(\mathbf{L}_k)$, then transforming it into its adjacent simplex via the corresponding incidence matrix, and finally filtering the transformed signal via another filter $\mathbf{G}_{k,p}(\mathbf{L}_{k+1})$ or $\mathbf{G}_{k,p}(\mathbf{L}_{k-1})$, e.g., $\mathbf{G}_{k,p}(\mathbf{L}_{k+1})\mathbf{B}_k^\top\mathbf{H}_{k,p}(\mathbf{L}_k)\mathbf{x}_{k,t-p}$. Following, this *convolve-transform-convolve* principle, we define a SC-VAR model of order P as:

$$\begin{aligned}
\mathbf{x}_{0,t} &= \sum_{p=1}^{P} \mathbf{H}_p^{00}(\mathbf{L}_0)\mathbf{x}_{0,t-p} + \mathbf{G}_p^{01}(\mathbf{L}_0)\mathbf{B}_1\mathbf{H}_p^{01}(\mathbf{L}_1)\mathbf{x}_{1,t-p} + \mathbf{n}_{0,t} \\
\mathbf{x}_{1,t} &= \sum_{p=1}^{P} \mathbf{G}_p^{10}(\mathbf{L}_1)\mathbf{B}_1^\top\mathbf{H}_p^{10}(\mathbf{L}_0)\mathbf{x}_{0,t-p} + \mathbf{H}_p^{11}(\mathbf{L}_1)\mathbf{x}_{1,t-p} \\
&\qquad + \mathbf{G}_p^{12}(\mathbf{L}_1)\mathbf{B}_2\mathbf{H}_p^{12}(\mathbf{L}_2)\mathbf{x}_{2,t-p} + \mathbf{n}_{1,t} \\
\mathbf{x}_{2,t} &= \sum_{p=1}^{P} \mathbf{G}_p^{21}(\mathbf{L}_2)\mathbf{B}_2^\top\mathbf{H}_p^{21}(\mathbf{L}_1)\mathbf{x}_{1,t-p} + \mathbf{H}_p^{22}(\mathbf{L}_2)\mathbf{x}_{2,t-p} + \mathbf{n}_{2,t}
\end{aligned} \tag{33}$$

where $\mathbf{n}_{k,t}$ denotes the model error. The SC-VAR expression in (33) 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 (33) do not require the post-transform filters $\mathbf{G}_p^{kk}(\cdot)$.

A particular instance of the SC-VAR model is the Simplicial-VAR (S-VAR) model, which neglects the vertical adjacencies, overlooking the dependencies across distinct simplicial levels:

$$\mathbf{x}_{k,t} = \sum_{p=1}^{P} \mathbf{H}_{k,p}(\mathbf{L}_k)\mathbf{x}_{k,t-p} + \mathbf{n}_t, \quad \text{for} \quad k = 0, 1, 2, \tag{34}$$

where $\{\mathbf{H}_p^k(\mathbf{L}_k)\}_{p=1}^P$ are simplicial convolutional filters [131].

An online learning method is proposed in order to update the time-varying SC-VAR model parameters of the model from streaming time series, providing convergence guarantees in terms of an upper bound for the dynamic regret, attaining a sublinear dynamic regret under reasonable assumptions suitable for real-world applications. In addition, a joint simplicial-temporal Fourier transform is also introduced by extending the simplicial Fourier transform concept [45] and analyzing the SC-VAR model in the spectral domain to investigate the

learned frequency responses of the model. Unlike the analogous G-VAR analysis [122], a coupled two-dimensional frequency response is shown between different types of simplicial frequencies.

Both SC-VAR and S-VAR have been representations that have been exploited in several relevant problems, such as online edge flow imputation, evolution backcasting of edge flows, spatio-temporal filtering and Kalman filtering for simplicial processes. An algorithm to retroactively compute the evolution of edge signals is proposed in [132]. In this case, first, this method learns an S-VAR representation and estimates missing edge signals from partially observed data using a block coordinate descent technique; and then, leveraging the learned S-VAR representation and the reconstructed signals, the algorithm backcasts the evolution of edge flows prior to the partial observations. It should be noticed that backcasting becomes more challenging when we only possess partial observations, which is common in practical scenarios (e.g. traffic or water networks) where data availability is limited due to sensor and communication failures, or due to the impracticality of placing sensors in some locations. However, the rich structural information rooted in the simplicial complex structure, enables the algorithm to address this effectively.

In [133], the simplicial-complex structure and Hodge Laplacians are exploited to incorporate the inductive bias in combination with multilinear kernel regression to perform edge imputation via manifold learning (MultiL-KRIM) [134]. MultiL-KRIM assumes that missing entries can be estimated by a set of landmark points, extracted from measurements and located around a smooth manifold, embedded in an ambient reproducing kernel Hilbert space (RKHS), which allows to perform functional approximation and perform imputation of time-varying edge flows. Similarly, in [135], an online algorithm is presented for time-varying edge flow imputation which combines learning a line graph identification via a vector autoregressive model and a group-Lasso-based optimization with a Kalman filtering-based reconstruction, leveraging a simplicial complex (SC) representation over the underlying learned causal dependencies. Another Kalman filtering approach for simplicial processes has been recently presented in [136], where it is assumed that the hidden dynamics of a system can be expressed as a simplicial process that respects the structure of the underlying network topology. These dynamics are observed through an observation matrix, which is represented using simplicial convolution filters. The combination of linear dynamics and linear observation enables the use of a Kalman filter to compute the best linear unbiased estimate of the process. Additionally, a parametric, structure-aware noise covariance model is proposed for the system dynamics, and the algorithm alternates between estimating the process state using the Kalman filter and updating the parameters through maximum likelihood estimation.

While these methods focus primary on linear models, the work in [137] proposed a recurrent neural network type TNN perspective for modeling nodal time series, where the simplicial complexes represent the higher-order interactions of the hidden states.

Adaptive filtering. Topological filtering can also be cast in the context of adaptive learning, whose aim is to infer and track the structure of an unknown system from streaming and noisy data observed over time. As an example, the work in [138] introduced a topological least-mean squares algorithm to process and learn from streaming signals defined over cell complexes. Specifically, let $\mathbf{x}_t$ be stationary edge flow signals,[1] processed over time by the linear shift-invariant graph filters according to the following model:

$$\mathbf{y}_t = \mathbf{M}_t \left[\sum_{m=1}^{T_d} h_m^{(d)} (\mathbf{L}^{(d)})^m \mathbf{x}_{t-m} + \sum_{m=0}^{T_u} h_m^{(u)} (\mathbf{L}^{(u)})^m \mathbf{x}_{t-m} + \mathbf{n}_t \right], \tag{35}$$

with $t \geq \max\{T_d, T_u\}$, where $\mathbf{n}_t$ denotes an i.i.d. zero-mean measurement noise, independent of any other signal; $\mathbf{h}$ is the vector collecting all topological filter coefficients to be estimated; and $\mathbf{M}_t = \text{diag}(m_{1t}, \ldots, m_{M_1 t}) \in \mathbb{R}^{M_1 \times M_1}$ is a sampling operator such that $m_{it} = 1$ if edge i is sampled at time t, and zero otherwise. The goal is to estimate the filter coefficients $\mathbf{h}$ in (35) using a mean-square-error criterion:

$$\min_{\mathbf{h}} \ \mathbb{E}\{\left\| \mathbf{y}_t - \mathbf{M}_t \mathbf{X}_t \mathbf{h} \right\|^2\}, \tag{36}$$

where matrix $\mathbf{X}_t$ is given by:

$$\mathbf{X}_t = \left[\mathbf{x}_t, \mathbf{L}_d \mathbf{x}_{t-1}, \ldots, \mathbf{L}_d^{T_d} \mathbf{x}_{t-T_d}, \mathbf{L}_u \mathbf{x}_{t-1}, \ldots, \mathbf{L}_u^{T_u} \mathbf{x}_{t-T_u} \right], \tag{37}$$

which collects shifted versions of the edge flows $\mathbf{x}_{t-m}$ over the upper and lower neighborhoods. Then, parameters $\mathbf{h}$ can be determined by proceeding iteratively using stochastic gradient descent:

$$\mathbf{h}_{t+1} = \mathbf{h}_t + \mu \mathbf{X}_t^\top \mathbf{M}_t \left(\mathbf{y}_t - \mathbf{X}_t \mathbf{h}_t \right), \tag{38}$$

with $\mu > 0$ denoting a (sufficiently small) step-size. Recursion (38) is referred to as the topological LMS algorithm [138], where streaming flow signals $\{\mathbf{y}_t, \mathbf{x}_t\}$ are processed taking into account the topological information coming from the cell complex domain.

Product spaces. An alternative to the above serial models, is to build upon the concept of product graphs [139,140] and enable spatio-temporal filtering of time-varying signals on SCs. The work in [141] presents a framework for signal processing on product spaces of simplicial complexes, where the time axis is interpreted as a SC, and a cellular complex is constructed to represent the spatially distributed, time-varying signal. It leverages the structure of the eigenmodes of the Hodge Laplacian of the product space to jointly filter along time and space. For this, they use a decomposition of the Hodge Laplacian of the product space, which shows how the product structure induces a decomposition of each eigenmode into a spatial and temporal component. This technique is applied to interpolate trajectories from a limited set of observed trajectories.

5. Applications

While topological signal processing and learning is an emerging research direction, they have shown promise in some key application areas or lend themselves in developments in different fields. Complementary application areas tailored to topological deep learning can be found in [50].

Processing network flow signals. Flow signals appear in a myriad of infrastructure networks such as water, power, transportation and telecommunication networks, among others. Such signals reside naturally on the edges of a network and are coupled with the node signals. One key challenge in critical infrastructure networks is that the overall network state (node+edge signals) need be estimated or forecasted by a few observations. The topological sampling techniques discussed in Section 4.4 can be used for sensor placement. One such case has been discussed in [142] for sensor placement in water distribution systems, where topological Gaussian processes have also been exploited. Additionally, topologically-aware interpolation strategies can be used for state estimation, which consists of in a node-and-edge signal interpolation task. In the water network case, a topological neural networks have been used in [143] to develop surrogate models that can transfer to unseen networks. Often, in these cases we may often require inferring edge flows from partial nodal data; in this case the physics of these models need be taken into account where the Hodge theory can be used to map between node and edge flows. The latter idea was used to develop implicit layers for simplicial neural networks in [144]. Simplicial neural networks and Hodge representation of networks has also been used in [145] for power outage detection and in [146] for false data injection attacks in smart grids. When such anomalies can be localized on particular Hodge subspaces [cf. Section 3], a mathematically tractable matched subspace detector with optimality

[1] We here drop the subscript $k = 1$ to avoid overcrowded symbols.

guarantees [147] can be used in place of neural network solutions. A further application of edge flow processing comes from the analysis of a discrete vector field, defined as a set of vectors associated to a point loud. As shown in [44], operating a Delaunay triangulation of the point cloud, a discrete vector field can be converted into a scalar field obtained by projecting the vector on each vertex onto the incident edges of the triangulation. The resulting edge flow can then be filtered using the methods described in the previous sections. The resulting scalar edge signal can then be mapped back onto a filtered discrete vector field living on the original point cloud.

Geometry processing and vector calculus. Historically, point cloud processing has been approached by graph-based techniques on the 3D mesh [148]. Such a paradigm has in fact been the early roots of developing graph signal processing techniques [149,150] till the latest progress with graph neural networks [13]. More recently, a topological-based mesh processing is taking place that combine vector calculus with mesh processing via the Hodge theory. For example, vector field-based computational processing in surfaces plays a crucial role in encoding both direction and sizing of the surfaces. By means of vector calculus and Hodge-Helmholtz decomposition it is possible to characterize and process the coupling of surface data on nodes, edges and triangular faces, ultimately, linking topological signal processing and learning with differential geometry [151,152]. For example, [153] considers a Hodge spectral processing of tangential vector fields that has strong ties with the topological Fourier analysis in Section 3. This spectral-based perspective for geometry processing has also been used in [154] to develop a spectral coarsening approach to expedite processing. Such spectral duality has also been used in end-to-end learning via tailored topological neural networks that are both geometric- and Hodge-aware in [155,156]. We hypothesize that bridging advances in geometry processing with those topological signal processing and learning can expedite the method development in both domains, in the same way, mesh-based and graph-based processing complemented each other.

Statistical ranking. Hodge theory and decompositions have shown great potential in statistical ranking of lists [157]. In particular, each vertex is an item in a list and an edge flow is considered as a ranking order; i.e., if the flow goes from vertex i to j then the ranking score is higher at i than j. Then, via the Hodge decomposition, a gradient flow shows a global consistency in ranking as they always go from higher-to-lower ranking scores, whereas a presence of an harmonic or curl flow would show local or global inconsistencies in rankings. This Hodge-based statistical ranking (HodgeRank) has been successfully applied to top-N recommender system lists and currency exchange markets [157] as well as in biomolecular data analysis [158]. The TSPL methods can further aid statistical ranking. For example, they can be used to remove ranking noise, perform ranking with missing values, or sample a minimum number of items to guarantee a certain ranking consistency. TNNs, and in particular those linked with the Hodge spectrum such as convolutional architectures [54] can be used to learn deep statistical ranking models.

Biomolecular data. A key challenge in graph-based learning is to classify molecules or even synthesize new one from a limited training set [159,160]. In fact, many of the message-passing TNN advances have been developed to overcome the limited Weisfeiler–Lehman expressivity of GNNs in graph classification tasks [87,88,91]. Molecules in particular can be seen as structurally rich graphs with (hidden) topological information that are combined their features [161]. By inducing higher-order simplicial or cell structures within a molecular graph –a technique known as lifting [162]– TNNs can leverage these topological relations to learn more expressive representations. The Hodge theory discussed in Sections 2 and 3 has been used in [158] to analyze biomolecular structures. More specifically, these biomolecular structures have been represented via simplicial complexes with certain edge flows, and the spectrum of the respective Hodge Laplacian reveals particular properties of the structure as well as characterizes folding or compactness of the biomolecule. Linking the latter with flow variabilities and the topological Fourier transform represents a promising unexplored direction that may reveal deeper insights on the relation between molecular features and structure.

Brain networks. Topological processing tools have been also largely applied to represent and extract information from brain network data. Specifically, the work in [163] addresses the challenge of identifying shared topological (group-level hole) substructures in brain networks by extending graph Laplacians to higher-order Hodge Laplacians. Then, the study in [164] employs Hodge decomposition to analyze brain networks by breaking them into gradient, curl, and harmonic flow components, effectively capturing complex topological features. Using a Wasserstein distance-based topological inference, the method reveals statistically significant differences in the topological properties of male and female brain networks from resting-state fMRI data. The works in [165,166] presents a persistent homology-based framework using the Hodge Laplacian to extract and analyze cycles in brain networks, validated through simulations and resting-state fMRI data. The authors of [167] introduced a noise-resilient method for EEG analysis using persistent homology and a Bayesian framework, enabling effective classification of noisy, nonlinear, and nonstationary signals. Other approaches have augmented GNNs with topological information [168–170]. For instance, the study [168] introduces a novel heterogeneous graph convolutional neural network using Hodge-Laplacian operators and topological graph pooling to analyze fMRI data, outperforming state-of-the-art GNNs in predicting general intelligence and providing interpretable neural circuit insights. Similarly, the work [169] proposed Hodge-Graph Neural Network, which leverages the Hodge Laplacian to directly model edgewise topological features in brain networks, outperforming traditional GNNs and graph classification models in the Alzheimer's Disease Neuroimaging Initiative study. Finally, the authors [170] introduces a dual aggregation framework for brain network analysis that simultaneously learns node and edge embeddings, capturing intricate node-edge relationships and demonstrating superior performance and interpretability.

Epidemic modeling. Modeling and predicting epidemic spreading represents one of the cornerstone applications network science [171]. This problem is conventionally seen as a dynamic process over the nodes of the graph, where the signal on the nodes indicates the state and edges represent probabilities of infection (e.g., susceptible vs. infected). Bridges of the latter with graph-based processing techniques can be found in [172–174]. More recently, contagion and spreading processes in complex systems have been better understood via higher-order networks and simplicial complexes; see e.g., [27,29,31]. While direct bridges between TSPL and epidemic spreading have yet to be established, we hypothesize that many of the advances presented in this paper can facilitate the latter and open the doors to new insights.

Semantic communication. A key challenge in semantic communications is managing the complexity of data representation while ensuring transmitted symbols effectively convey intended meanings within acceptable distortion or perceptual variation [175,176]. This demands robust methods to model, extract, and encode data semantics. Although a universal definition of semantics in communication is still lacking, it can be understood as the relationships between elements of a broadly defined language, often formalized using topological spaces—mathematical structures describing elements and their relations [175]. Defining multi-way relationships allows data to be represented as signals over a topological space, where TSP becomes crucial for semantic data representation and communication. In particular, the choice of the semantic topological space directly impacts our ability to process the data and achieve an efficient, parsimonious representation that optimally balances distortion, perception, and complexity of data representation.

6. Conclusion and open issues

This paper brought together recent methodological advances in topological signal processing and learning. Instrumental to these methods is to represent the higher-order data structure via simplicial or cell complexes, which consists of well-structured richer forms of hypergraphs. By representing these topological structures via the Hodge Laplacians, it is possible to generalize key signal processing concepts from the Euclidean and graph domain to topologies, where now we can characterize signals defined on nodes, edges and other higher-order topological structures such as triangular faces. Relying on the latter and on the spectral theory of these Hodge Laplacians, we highlighted striking differences between topological Fourier domain and Euclidean and graph Fourier domain. Subsequently, methodological advances have been revisited and linked to potential application domains where graph-based techniques remain limited.

While recent works have shown their promise for some important applications, there remain significant open issues.

- First, on the topological representation front, these methods rely predominantly on undirected and unweighted topological structures. This, however, limits drastically their representation power as dependency between higher-order signals can be better represented as a directed or a weighted one. Some recent advances about weighted representation of topological complexes have been proposed in [70,177] and the first works have emerged about algebraic representation of directed topologies [178,179] and directed topological neural networks [180]. Yet, these directions remain largely unexplored and the impact of weights and directionality on TSPL methods is little studied.
- Second, inferring the simplicial and cell structure from data poses not only the conceptual challenge of how to tie the signal with the topology, but also poses a large computational challenge. While some techniques have emerged as discussed in Section 4.2 scaling the latter to hundreds of thousands of nodes as may be the case in brain structures is non-trivial.
- Third, the spectral techniques for TSPL are meaningful when processing signals at a particular topological level. When multiple signals are present, we can rely on spectrum of the Dirac operator [cf. Section 2.5] but the latter is not as meaningful as the spectrum of the Hodge Laplacians. Especially, for spectral analysis of topological signals deeper insights on joint spectral processing of multiple topological signals remains an open issue.
- Fourth, most of the datasets and applications contain signals up to edge flows, ultimately, limiting the potential of TSPL methods. Identifying, significant application areas with data over second or higher order simplicial/cell structures may be a substantial leap forward towards developing more powerful TSPL methods. Links of the latter with topological data analysis and topological deep learning can further accelerate the impact.
- Fifth, on the application front, TSPL methods has shown the most promise for signals up to edge flows present in data sets exhibiting irregular relationships that can captured by simplicial or cell structures. However, additional research is needed to identify promising application areas where signals reside naturally on higher-order simplicial or cell structures. The latter will not only open the doors to new application domains but will also provide deeper insights on the advantages and limitations of these techniques for handling irregular data.

CRediT authorship contribution statement

Elvin Isufi: Conceptualization, Methodology, Writing – original draft, Writing – Review & Editing, Project administration. **Geert Leus:** Writing – original draft, Writing – Review & Editing. **Baltasar Beferull-Lozano:** Writing – original draft, Writing – Review & Editing. **Sergio Barbarossa:** Writing – original draft, Writing – Review & Editing. **Paolo Di Lorenzo:** Conceptualization, Methodology, Writing – original draft, Writing – Review & Editing, Project administration.

Acknowledgments

The work of E. Isufi and G. Leus has been partly supported by the Dutch Research Foundation, Open Technology Programme GraSPA proposal #19497. The work of E. Isufi has been also partly supported by the TU Delft AI Labs programme and the Dutch Research foundation VENI proposal #19052. The work of B. Beferull-Lozano has been partly supported by the IKTPLUSS DISCO grant 338740, Research Council of Norway.

The work of S. Barbarossa and P. Di Lorenzo has been partly supported by European Union under the Italian National Recovery and Resilience Plan of NextGenerationEU, partnership on Telecommunications of the Future (PE00000001 - program RESTART). The work of P. Di Lorenzo has been also partly funded the SNS JU project 6G-GOALS under the EU's Horizon program Grant Agreement No 101139232.

Data availability

No data was used for the research described in the article.

References

[1] G. Karlebach, R. Shamir, Modelling and analysis of gene regulatory networks, Nature Rev. Mol. Cell Biol. 9 (10) (2008) 770–780.
[2] M. Jackson, Social and Economic Networks, Princeton University Press, 2008.
[3] J. Richiardi, S. Achard, H. Bunke, D. Van De Ville, Machine learning with brain graphs: predictive modeling approaches for functional imaging in systems neuroscience, IEEE Signal Process. Mag. 30 (3) (2013) 58–70.
[4] M. Newman, Networks, Oxford University Press, 2018.
[5] S. Lauritzen, Graphical Models, Clarendon Press, 1996.
[6] A. Ortega, P. Frossard, J. Kovačević, J.M.F. Moura, P. Vandergheynst, Graph signal processing: Overview, challenges and applications 106 (5), 2018, pp. 808–828.
[7] M.M. Bronstein, J. Bruna, T. Cohen, P. Veličković, Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2022, arXiv:2104.13478v2. URL http://arxiv.org/abs/2104.13478.
[8] E. Isufi, F. Gama, D.I. Shuman, S. Segarra, Graph filters for signal processing and machine learning on graphs, IEEE Trans. Signal Process. (2024).
[9] A. Sandyhaila, J.M.F. Moura, Discrete signal processing on graphs: Frequency analysis 62 (12), 2014, pp. 3042–3054.
[10] M. Tsitsvero, S. Barbarossa, P. Di Lorenzo, Signals on graphs: Uncertainty principle and sampling, IEEE Trans. Signal Process. 64 (18) (2016) 4845–4860.
[11] Y. Tanaka, Y.C. Eldar, A. Ortega, G. Cheung, Sampling signals on graphs: From theory to applications, IEEE Signal Process. Mag. 37 (6) (2020) 14–30.
[12] P.W. Battaglia, J.B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al., Relational inductive biases, deep learning, and graph networks, 2018, arXiv preprint arXiv:1806.01261.
[13] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S.Y. Philip, A comprehensive survey on graph neural networks, IEEE Trans. Neural Networks Learn. Syst. 32 (1) (2020) 4–24.
[14] L.J. Grady, J.R. Polimeni, Discrete Calculus: Applied Analysis on Graphs for Computational Science, vol. 3, Springer, 2010.
[15] R. Lambiotte, M. Rosvall, I. Scholtes, From networks to optimal higher-order models of complex systems, Nat. Phys. 15 (4) (2019) 313–320.
[16] S. Majhi, M. Perc, D. Ghosh, Dynamics on higher-order networks: A review, J. R. Soc. Interface 19 (188) (2022) 20220043.
[17] C. Bick, E. Gross, H.A. Harrington, M.T. Schaub, What are higher-order networks? SIAM Rev. 65 (3) (2023) 686–731.
[18] C. Berge, Hypergraphs: Combinatorics of Finite Sets, vol. 45, Elsevier, 1984.
[19] S. Zhang, Z. Ding, S. Cui, Introducing hypergraph signal processing: Theoretical foundation and practical applications, IEEE Internet Things J. 7 (1) (2019) 639–660.
[20] K. Pena-Pena, D.L. Lau, G.R. Arce, T-HGSP: Hypergraph signal processing using t-product tensor decompositions, IEEE Trans. Signal Inf. Process. over Networks 9 (2023) 329–345.
[21] S. Zhang, S. Cui, Z. Ding, Hypergraph spectral analysis and processing in 3D point cloud, IEEE Trans. Image Process. 30 (2020) 1193–1206.

[22] K. Pena-Pena, L. Taipe, F. Wang, D.L. Lau, G.R. Arce, Learning hypergraphs tensor representations from data via t-HGSP, IEEE Trans. Signal Inf. Process. over Networks (2023).

[23] F. Wang, K. Pena-Pena, W. Qian, G.R. Arce, T-HyperGNNs: Hypergraph neural networks via tensor representations, IEEE Trans. Neural Networks Learn. Syst. (2024).

[24] L.-H. Lim, Hodge Laplacians on graphs, Siam Rev. (2020).

[25] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, P. Vandergheynst, The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains 30 (3), 2013, pp. 83–98.

[26] F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, H. Liu, Graph learning: A survey, IEEE Trans. Artif. Intell. 2 (2) (2021) 109–127.

[27] F. Battiston, G. Cencetti, I. Iacopini, V. Latora, M. Lucas, A. Patania, J.-G. Young, G. Petri, Networks beyond pairwise interactions: Structure and dynamics, Phys. Rep. 874 (2020) 1–92.

[28] S. Krishnagopal, G. Bianconi, Spectral detection of simplicial communities via Hodge Laplacians, Phys. Rev. E 104 (6) (2021) 064303.

[29] F. Battiston, E. Amico, A. Barrat, G. Bianconi, G. Ferraz de Arruda, B. Franceschiello, I. Iacopini, S. Kéfi, V. Latora, Y. Moreno, et al., The physics of higher-order interactions in complex systems, Nat. Phys. 17 (10) (2021) 1093–1098.

[30] Y. Zhang, M. Lucas, F. Battiston, Higher-order interactions shape collective dynamics differently in hypergraphs and simplicial complexes, Nat. Commun. 14 (1) (2023) 1605.

[31] I. Iacopini, G. Petri, A. Barrat, V. Latora, Simplicial models of social contagion, Nat. Commun. 10 (1) (2019) 2485.

[32] F. Malizia, A. Corso, L.V. Gambuzza, G. Russo, V. Latora, M. Frasca, Reconstructing higher-order interactions in coupled dynamical systems, Nat. Commun. 15 (1) (2024) 5184.

[33] Z. Lin, L. Han, M. Feng, Y. Liu, M. Tang, Higher-order non-Markovian social contagions in simplicial complexes, Commun. Phys. 7 (1) (2024) 175.

[34] M.T. Schaub, A.R. Benson, P. Horn, G. Lippner, A. Jadbabaie, Random walks on simplicial complexes and the normalized Hodge 1-Laplacian, SIAM Rev. 62 (2) (2020) 353–391.

[35] G. Carlsson, Topological methods for the analysis of high dimensional data sets and 3D object recognition, Bull. Am. Math. Soc. 46 (2) (2009) 255–265.

[36] H. Edelsbrunner, J. Harer, Computational Topology: An Introduction, American Mathematical Society, 2010.

[37] L. Wasserman, Topological Data Analysis, Johns Hopkins University Press, 2018.

[38] N. Otter, M. Porter, U. Tillmann, V. Vitelli, G. Carlsson, A roadmap for the computation of persistent homology, EPJ Data Sci. 6 (1) (2017) 17.

[39] R. Ghrist, Barcodes: the persistent topology of data, Bull. Am. Math. Soc. 45 (1) (2008) 61–75.

[40] H. Edelsbrunner, J. Harer, Computational Topology: An Introduction, American Mathematical Society, 2022.

[41] F. Hensel, M. Moor, B. Rieck, A survey of topological machine learning methods, Front. Artif. Intell. 4 (2021) 681108.

[42] B.C. Bispo, F.A. Santos, J.R.D.O. Neto, J.B. Lima, Emergence of higher-order functional brain connectivity with hypergraph signal processing, in: 2024 32nd European Signal Processing Conference, EUSIPCO, IEEE, 2024, pp. 1332–1336.

[43] S. Barbarossa, M. Tsitsvero, An introduction to hypergraph signal processing, in: IEEE Int. Conf. Acoust., Speech and Signal Process., 2016, pp. 6425–6429.

[44] S. Barbarossa, S. Sardellitti, Topological signal processing over simplicial complexes, IEEE Trans. Signal Process. 68 (2020) 2992–3007.

[45] S. Barbarossa, S. Sardellitti, Topological signal processing: Making sense of data building on multiway relations, IEEE Signal Process. Mag. 37 (6) (2020) 174–183.

[46] M.T. Schaub, Y. Zhu, J.-B. Seby, T.M. Roddenberry, S. Segarra, Signal processing on higher-order networks: Livin'on the edge... and beyond, Signal Process. 187 (2021) 108149.

[47] Y. Feng, H. You, Z. Zhang, R. Ji, Y. Gao, Hypergraph neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, (no. 01) 2019, pp. 3558–3565.

[48] A. Antelmi, G. Cordasco, M. Polato, V. Scarano, C. Spagnuolo, D. Yang, A survey on hypergraph representation learning, ACM Comput. Surv. 56 (1) (2023) 1–38.

[49] M. Papillon, S. Sanborn, M. Hajij, N. Miolane, Architectures of topological deep learning: A survey of message-passing topological neural networks, 2023, arXiv preprint arXiv:2304.10031.

[50] T. Papamarkou, T. Birdal, M.M. Bronstein, G.E. Carlsson, J. Curry, Y. Gao, M. Hajij, R. Kwitt, P. Lio, P. Di Lorenzo, et al., Position: Topological deep learning is the new frontier for relational learning, in: Forty-First International Conference on Machine Learning, 2024.

[51] S. Sanborn, J. Mathe, M. Papillon, D. Buracas, H.J. Lillemark, C. Shewmake, A. Bertics, X. Pennec, N. Miolane, Beyond euclid: An illustrated guide to modern machine learning with geometric, topological, and algebraic structures, 2024, arXiv preprint arXiv:2407.09468.

[52] M. Hajij, G. Zamzmi, T. Papamarkou, N. Miolane, A. Guzmán-Sáenz, K.N. Ramamurthy, T. Birdal, T.K. Dey, S. Mukherjee, S.N. Samaga, et al., Topological Deep Learning: Going Beyond Graph Data, May 2023. arXiv preprint arXiv:2206.00606.

[53] S. Ebli, M. Defferrard, G. Spreemann, Simplicial neural networks, in: NeurIPS Workshops, TDA & beyond, Preprint ArXiv:2010.03633, 2020.

[54] M. Yang, E. Isufi, G. Leus, Simplicial convolutional neural networks, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2022, pp. 8847–8851.

[55] C. Battiloro, L. Testa, L. Giusti, S. Sardellitti, P. Di Lorenzo, S. Barbarossa, Generalized simplicial attention neural networks, IEEE Trans. Signal Inf. Process. over Networks (2024).

[56] T.E. Goldberg, Combinatorial Laplacians of Simplicial Complexes (Ph.D. thesis), Bard College, 2002.

[57] S. Sardellitti, S. Barbarossa, L. Testa, Topological signal processing over cell complexes, in: 2021 55th Asilomar Conference on Signals, Systems, and Computers, IEEE, 2021, pp. 1558–1562.

[58] T.M. Roddenberry, M.T. Schaub, M. Hajij, Signal processing on cell complexes, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2022, pp. 8852–8856.

[59] S. Sardellitti, S. Barbarossa, Topological signal processing over generalized cell complexes, IEEE Trans. Signal Process. (2024).

[60] M. Yang, E. Isufi, M.T. Schaub, G. Leus, Simplicial convolutional filters, IEEE Trans. Signal Process. 70 (2022) 4633–4648.

[61] G. Bianconi, The topological Dirac equation of networks and simplicial complexes, J. Phys.: Complex. 2 (3) (2021) 035022.

[62] L. Calmon, M.T. Schaub, G. Bianconi, Dirac signal processing of higher-order topological signals, New J. Phys. 25 (9) (2023) 093013.

[63] V.P. Grande, M.T. Schaub, Disentangling the spectral properties of the Hodge Laplacian: not all small eigenvalues are equal, in: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2024, pp. 9896–9900.

[64] O. Rioul, M. Vetterli, Wavelets and signal processing, IEEE Signal Process. Mag. 8 (4) (1991) 14–38.

[65] I. Tošić, P. Frossard, Dictionary learning, IEEE Signal Process. Mag. 28 (2) (2011) 27–38.

[66] M. Tsitsvero, S. Barbarossa, On the degrees of freedom of signals on graphs, in: 2015 23rd European Signal Processing Conference, EUSIPCO, IEEE, 2015, pp. 1506–1510.

[67] T.M. Roddenberry, F. Frantzen, M.T. Schaub, S. Segarra, Hodgelets: Localized spectral representations of flows on simplicial complexes, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2022, pp. 5922–5926.

[68] C. Battiloro, P. Di Lorenzo, S. Barbarossa, Topological slepians: Maximally localized representations of signals over simplicial complexes, in: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2023, pp. 1–5.

[69] N. Saito, S.C. Schonsheck, E. Shvarts, Multiscale Hodge scattering networks for data analysis, 2023, arXiv preprint arXiv:2311.10270.

[70] C. Battiloro, S. Sardellitti, S. Barbarossa, P. Di Lorenzo, Topological signal processing over weighted simplicial complexes, in: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2023, pp. 1–5.

[71] G. Mateos, S. Segarra, A.G. Marques, A. Ribeiro, Connecting the dots: Identifying network structure via graph signal processing, IEEE Signal Process. Mag. 36 (3) (2019) 16–43.

[72] J. Hoppe, M.T. Schaub, Representing edge flows on graphs via sparse cell complexes, in: Learning on Graphs Conference, PMLR, 2024, 1–1.

[73] S. Gurugubelli, S.P. Chepuri, Simplicial complex learning from edge flows via sparse clique sampling, in: 2024 32nd European Signal Processing Conference, EUSIPCO, IEEE, 2024, pp. 2332–2336.

[74] S. Sardellitti, S. Barbarossa, Probabilistic topological models over simplicial complexes, in: 2023 57th Asilomar Conference on Signals, Systems, and Computers, IEEE, 2023, pp. 822–826.

[75] J. Friedman, T. Hastie, R. Tibshirani, Sparse inverse covariance estimation with the graphical lasso, Biostatistics 9 (3) (2008) 432–441.

[76] A. Buciulea, E. Isufi, G. Leus, A.G. Marques, Learning the topology of a simplicial complex using simplicial signals: A greedy approach, in: 2024 IEEE 13rd Sensor Array and Multichannel Signal Processing Workshop, SAM, IEEE, 2024, pp. 1–5.

[77] A. Buciulea, E. Isufi, G. Leus, A.G. Marques, Learning graphs and simplicial complexes from data, in: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2024, pp. 9861–9865.

[78] H. Wang, C. Ma, H.-S. Chen, Y.-C. Lai, H.-F. Zhang, Full reconstruction of simplicial complexes from binary contagion and ising data, Nat. Commun. 13 (1) (2022) 3043.

[79] C. Battiloro, I. Spinelli, L. Telyatnikov, M. Bronstein, S. Scardapane, P. Di Lorenzo, From latent graph to latent topology inference: Differentiable cell complex module, in: Proc. of the Twelfth International Conference on Learning Representations, ICLR, 2023, preprint arXiv:2305.16174.

[80] E. Isufi, M. Yang, Convolutional filtering in simplicial complexes, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2022, pp. 5578–5582.

[81] J. Jia, M.T. Schaub, S. Segarra, A.R. Benson, Graph-based semi-supervised & active learning for edge flows, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 761–771.

[82] M. Yang, E. Isufi, Simplicial trend filtering, in: 2022 56th Asilomar Conference on Signals, Systems, and Computers, IEEE, 2022, pp. 930–934.

[83] C. Liu, G. Leus, E. Isufi, Unrolling of simplicial ElasticNet for edge flow signal reconstruction, IEEE Open J. Signal Process. (2023).

[84] S. Reddy, S.P. Chepuri, Recovery of signals on a simplicial complex from subsampled neighbourhood aggregation, IEEE Signal Process. Lett. (2024).

[85] T.S. Reddy, S.P. Chepuri, Sampling and recovery of signals over product cell structures, in: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2024, pp. 13191–13195.

[86] S. Gurugubelli, S.P. Chepuri, Gaussian processes for edge flow prediction with active learning, in: 2023 57th Asilomar Conference on Signals, Systems, and Computers, IEEE, 2023, pp. 809–813.

[87] C. Bodnar, F. Frasca, Y. Wang, N. Otter, G.F. Montufar, P. Lio, M. Bronstein, Weisfeiler and lehman go topological: Message passing simplicial networks, in: International Conference on Machine Learning, PMLR, 2021, pp. 1026–1037.

[88] C. Bodnar, F. Frasca, N. Otter, Y. Wang, P. Lio, G.F. Montufar, M. Bronstein, Weisfeiler and lehman go cellular: Cw networks, Adv. Neural Inf. Process. Syst. 34 (2021) 2625–2640.

[89] T.M. Roddenberry, N. Glaze, S. Segarra, in: M. Meila, T. Zhang (Eds.), Principled Simplicial Neural Networks for Trajectory Prediction, vol. 139, 2021, pp. 9020–9029.

[90] E. Rossi, F. Frasca, B. Chamberlain, D. Eynard, M. Bronstein, F. Monti, Sign: Scalable inception graph neural networks, in: Proc. of ICML 2020 Workshop on Graph Representation Learning, vol. 7, 2020, p. 15, preprint arXiv:2004.11198.

[91] S. Gurugubelli, S.P. Chepuri, Sann: Simple yet powerful simplicial-aware neural networks, in: The Twelfth International Conference on Learning Representations, 2024.

[92] F. Gama, E. Isufi, G. Leus, A. Ribeiro, Graphs, convolutions, and neural networks: From graph filters to graph neural networks, IEEE Signal Process. Mag. 37 (6) (2020) 128–138.

[93] M. Yang, E. Isufi, Convolutional learning on simplicial complexes, 2023, arXiv preprint arXiv:2301.11163.

[94] C. Zhou, X. Wang, M. Zhang, Facilitating graph neural networks with random walk on simplicial complexes, Adv. Neural Inf. Process. Syst. 36 (2024).

[95] Y. Huang, Y. Zeng, Q. Wu, L. Lü, Higher-order graph convolutional network with flower-petals laplacians on simplicial complexes, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, (no. 11) 2024, pp. 12653–12661.

[96] Y. Chen, Y.R. Gel, H.V. Poor, BScNets: Block simplicial complex neural networks, in: Proceedings of the Aaai Conference on Artificial Intelligence, vol. 36, (no. 6) 2022, pp. 6333–6341.

[97] R. Yang, F. Sala, P. Bogdan, Efficient representation learning for higher-order data with simplicial complexes, in: Learning on Graphs Conference, PMLR, 2022, pp. 1–13.

[98] T.N. Kipf, M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, 2017, pp. 1–14.

[99] Y. Yan, E.E. Kuruoglu, Binarized simplicial convolutional neural networks, Neural Netw. 183 (2025) 106928.

[100] C. Battiloro, Z. Wang, H. Riess, P. Di Lorenzo, A. Ribeiro, Tangent bundle convolutional learning: from manifolds to cellular sheaves and back, IEEE Trans. Signal Process. (2024).

[101] L. Giusti, C. Battiloro, P. Di Lorenzo, S. Sardellitti, S. Barbarossa, Simplicial attention neural networks, 2022, arXiv preprint arXiv:2203.07485.

[102] C.W.J. Goh, C. Bodnar, P. Lio, Simplicial attention networks, in: ICLR 2022 Workshop on Geometrical and Topological Representation Learning, arXiv preprint arXiv:2204.09455.

[103] M. Hajij, G. Zamzmi, T. Papamarkou, N. Miolane, A. Guzmán-Sáenz, K.N. Ramamurthy, Higher-order attention networks, 2022, p. 4, arXiv preprint arXiv:2206.00606. 2 (3).

[104] S.H. Lee, F. Ji, W.P. Tay, SGAT: Simplicial graph attention network, 2022, arXiv preprint arXiv:2207.11761.

[105] D.M. Cinque, C. Battiloro, P. Di Lorenzo, Pooling strategies for simplicial convolutional networks, in: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2023, pp. 1–5.

[106] S. McGuire, E. Munch, M. Hirn, NervePool: A simplicial pooling layer, in: 2023 Joint Mathematics Meetings, JMM 2023, 2023, arXiv preprint arXiv:2305.06315.

[107] M. Yang, V. Borovitskiy, E. Isufi, Hodge-compositional edge Gaussian processes, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2024, pp. 3754–3762.

[108] M. Alain, S. Takao, B. Paige, M.P. Deisenroth, Gaussian processes on cellular complexes, in: Proc. of International Conference on Machine Learning, ICML, 2023, arXiv preprint arXiv:2311.01198.

[109] S. Gurugubelli, S.P. Chepuri, Gaussian processes for predicting simplicial closure, in: 2024 IEEE 13rd Sensor Array and Multichannel Signal Processing Workshop, SAM, IEEE, 2024, pp. 1–5.

[110] M. Navarro, S. Segarra, SC-MAD: Mixtures of higher-order networks for data augmentation, in: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2024, pp. 13446–13450.

[111] H. Zhang, Mixup: Beyond empirical risk minimization, 2017, arXiv preprint arXiv:1710.09412.

[112] H. Madhu, S.P. Chepuri, TopoSRL: topology preserving self-supervised simplicial representation learning, Adv. Neural Inf. Process. Syst. 36 (2024).

[113] A. Möllers, A. Immer, V. Fortuin, E. Isufi, Hodge-aware contrastive learning, in: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2024, pp. 9746–9750.

[114] G. Leus, M. Yang, M. Coutino, E. Isufi, Topological Volterra filters, in: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2021, pp. 5385–5399, http://dx.doi.org/10.1109/ICASSP39728.2021.9414275.

[115] H. Lütkepohl, New Introduction to Multiple Time Series Analysis, Springer Science & Business Media, 2005.

[116] B. Zaman, L. Ramos, D. Romero, B. Beferull-Lozano, Online topology identification from vector autoregressive time series, IEEE Trans. Signal Process. 69 (2021) 210–225, http://dx.doi.org/10.1109/TSP.2020.3042940.

[117] R. Money, J. Krishnan, B. Beferull-Lozano, Online non-linear topology identification from graph-connected time series, in: 2021 IEEE Data Science and Learning Workshop, DSLW, 2021, pp. 1–6, http://dx.doi.org/10.1109/DSLW51110.2021.9523399.

[118] R. Money, J. Krishnan, B. Beferull-Lozano, Random feature approximation for online nonlinear graph topology identification, IEEE MLSP (2021).

[119] R. Money, J. Krishnan, B. Beferull-Lozano, Sparse online learning with kernels using random features for estimating nonlinear dynamic graphs, IEEE Trans. Signal Process. 71 (2023) 2027–2042, http://dx.doi.org/10.1109/TSP.2023.3282068.

[120] Y. Shen, G. Giannakis, B. Baingana, Nonlinear structural vector autoregressive models with application to directed brain networks, IEEE Trans. Signal Process. 67 (2019) 5325–5339.

[121] M. Veedu, D. Harish, M. Salapaka, Topology learning of linear dynamical systems with latent nodes using matrix decomposition, IEEE Trans. Autom. Control (2021).

[122] E. Isufi, A. Loukas, N. Perraudin, G. Leus, Forecasting time series with VARMA recursions on graphs, IEEE Trans. Signal Process. 67 (18) (2019) 4870–4885, http://dx.doi.org/10.1109/TSP.2019.2929930.

[123] E. Isufi, F. Gama, D. Shuman, S. Segarra, Graph filters for signal processing and machine learning on graphs, IEEE Trans. Signal Process. (2024) 1–32, http://dx.doi.org/10.1109/TSP.2024.3349788.

[124] C. Lam, Q. Yao, Factor modeling for high-dimensional time series: Inference for the number of factors, Ann. Statist. 40 (2) (2012) 694–726.

[125] N. Lee, H. Choi, S. Kim, Bayes shrinkage estimation for high-dimensional VAR models with scale mixture of normal distributions for noise, Comput. Statist. Data Anal. 101 (2016) 250–276.

[126] J. Chang, B. Guo, Q. Yao, Principal component analysis for second-order stationary vector time series, Ann. Statist. 46 (5) (2018) 2094–2124.

[127] L. Gruber, G. Kastner, Forecasting macroeconomic data with Bayesian VARs: Sparse or dense? It depends! 2023, URL https://arxiv.org/abs/2206.04902. arXiv:2206.04902.

[128] G. Kastner, F. Huber, Sparse Bayesian vector autoregressions in huge dimensions, J. Forecast. 39 (7) (2020) 1142–1165, http://dx.doi.org/10.1002/for.2680, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/for.2680.

[129] G. Primiceri, Time varying structural vector autoregressions and monetary policy, Rev. Econ. Stud. 72 (3) (2005) 821–852, http://dx.doi.org/10.1111/j.1467-937X.2005.00353.x.

[130] J. Krishnan, R. Money, B. Beferull-Lozano, E. Isufi, Simplicial vector autoregressive models, IEEE Trans. Signal Process. (2024) 1–15, http://dx.doi.org/10.1109/TSP.2024.3503063.

[131] J. Krishnan, R. Money, B. Beferull-Lozano, E. Isufi, Simplicial vector autoregressive model for streaming edge flows, in: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2023, pp. 1–5, http://dx.doi.org/10.1109/ICASSP49357.2023.10096095.

[132] R. Money, J. Krishnan, B. Beferull-Lozano, E. Isufi, Evolution backcasting of edge flows from partial observations using simplicial vector autoregressive models, in: ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2024, pp. 9516–9520, http://dx.doi.org/10.1109/ICASSP48485.2024.10448180.

[133] D.T. Nguyen, K. Slavakis, D. Pados, Imputation of time-varying edge flows in graphs by multilinear kernel regression and manifold learning, 2024, URL https://arxiv.org/abs/2409.05135. arXiv:2409.05135.

[134] D.T. Nguyen, K. Slavakis, Multilinear kernel regression and imputation via manifold learning, IEEE Open J. Signal Process. 5 (2024) 1073–1088, http://dx.doi.org/10.1109/OJSP.2024.3444707.

[135] R. Money, J. Krishnan, B. Beferull-Lozano, E. Isufi, Online edge flow imputation on networks, IEEE Signal Process. Lett. 30 (2023) 115–119, http://dx.doi.org/10.1109/LSP.2022.3221846.

[136] R. Money, M. Sabbaqi, J. Krishnan, B. Beferull-Lonano, E. Isufi, Kalman filtering for simplicial processes, in: Asilomar 2024 - 2024 IEEE International Conference, 2024.

[137] Y. Chen, Y. Gel, H.V. Poor, Time-conditioned dances with simplicial complexes: Zigzag filtration curve based supra-hodge convolution networks for time-series forecasting, Adv. Neural Inf. Process. Syst. 35 (2022) 8940–8953.

[138] L. Marinucci, C. Battiloro, P. Di Lorenzo, Topological adaptive learning over cell complexes, in: 2024 32nd European Signal Processing Conference, EUSIPCO, IEEE, 2024, pp. 832–836.

[139] A. Sandryhaila, J.M. Moura, Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure, IEEE Signal Process. Mag. 31 (5) (2014) 80–90.

[140] M. Sabbaqi, E. Isufi, Graph-time convolutional neural networks: Architecture and theoretical analysis, IEEE Trans. Pattern Anal. Mach. Intell. (2023).

[141] T.M. Roddenberry, M.T. Schaub, M. Hajij, Signal processing on cell complexes, in: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2022, pp. 8852–8856, http://dx.doi.org/10.1109/ICASSP43922.2022.9747233.

[142] B. Kerimov, V. Pons, S. Pritsis, R. Taormina, F. Tscheikner-Gratl, Sensor placement and state estimation in water distribution systems using edge Gaussian processes, Eng. Proc. 69 (1) (2024) 150.

[143] B. Kerimov, R. Taormina, F. Tscheikner-Gratl, Towards transferable metamodels for water distribution systems with edge-based graph neural networks, Water Res. (2024) 121933.

[144] K.D. Smith, F. Seccamonte, A. Swami, F. Bullo, Physics-informed implicit representations of equilibrium network flows, Adv. Neural Inf. Process. Syst. 35 (2022) 7211–7221.

[145] Y. Chen, R.A. Jacob, Y.R. Gel, J. Zhang, H.V. Poor, Learning power grid outages with higher-order topological neural networks, IEEE Trans. Power Syst. 39 (1) (2023) 720–732.

[146] W. Xia, Y. Li, L. Yu, D. He, Locational detection of false data injection attacks in the edge space via Hodge graph neural network for smart grids, IEEE Trans. Smart Grid (2024).

[147] C. Liu, E. Isufi, Hodge-aware matched subspace detectors, in: 2024 32nd European Signal Processing Conference, EUSIPCO, IEEE, 2024, pp. 817–821.

[148] W. Hu, J. Pang, X. Liu, D. Tian, C.-W. Lin, A. Vetro, Graph signal processing for geometric data and beyond: Theory and applications, IEEE Trans. Multimed. 24 (2021) 3961–3977.

[149] G. Taubin, A signal processing approach to fair surface design, in: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, 1995, pp. 351–358.

[150] G. Taubin, T. Zhang, G. Golub, Optimal surface smoothing as filter design, in: Eur. Conf. Comput. Vision, 1996, pp. 283–292.

[151] K. Crane, F. De Goes, M. Desbrun, P. Schröder, Digital geometry processing with discrete exterior calculus, in: ACM SIGGRAPH 2013 Courses, 2013, pp. 1–126.

[152] F. De Goes, M. Desbrun, Y. Tong, Vector field processing on triangle meshes, in: ACM SIGGRAPH 2016 Courses, 2016, pp. 1–49.

[153] C. Brandt, L. Scandolo, E. Eisemann, K. Hildebrandt, Spectral processing of tangential vector fields, in: Computer Graphics Forum, vol. 36, (no. 6) Wiley Online Library, 2017, pp. 338–353.

[154] A. Keros, K. Subr, Spectral coarsening with Hodge Laplacians, in: ACM SIGGRAPH 2023 Conference Proceedings, 2023, pp. 1–11.

[155] D. Smirnov, J. Solomon, HodgeNet: Learning spectral geometry on triangle meshes, ACM Trans. Graph. 40 (4) (2021) 1–11.

[156] R. Wiersma, A. Nasikun, E. Eisemann, K. Hildebrandt, Deltaconv: anisotropic operators for geometric deep learning on point clouds, ACM Trans. Graph. 41 (4) (2022) 1–10.

[157] X. Jiang, L.-H. Lim, Y. Yao, Y. Ye, Statistical ranking and combinatorial Hodge theory, Math. Program. 127 (1) (2011) 203–244.

[158] R.K.J. Wei, J. Wee, V.E. Laurent, K. Xia, Hodge theory-based biomolecular data analysis, Sci. Rep. 12 (1) (2022) 9699.

[159] O. Wieder, S. Kohlbacher, M. Kuenemann, A. Garon, P. Ducrot, T. Seidel, T. Langer, A compact review of molecular property prediction with graph neural networks, Drug Discov. Today: Technol. 37 (2020) 1–12.

[160] M. Zhang, M. Qamar, T. Kang, Y. Jung, C. Zhang, S.-H. Bae, C. Zhang, A survey on graph diffusion models: Generative ai in science for molecule, protein and material, 2023, arXiv preprint arXiv:2304.01565.

[161] Y. Jiang, D. Chen, X. Chen, T. Li, G.-W. Wei, F. Pan, Topological representations of crystalline compounds for the machine-learning prediction of materials properties, Npj Comput. Mater. 7 (1) (2021) 28.

[162] G. Bernárdez, L. Telyatnikov, M. Montagna, F. Baccini, M. Papillon, M. Ferriol-Galmés, M. Hajij, T. Papamarkou, M.S. Bucarelli, O. Zaghen, et al., ICML topological deep learning challenge 2024: Beyond the graph domain, 2024, arXiv preprint arXiv:2409.05211.

[163] H. Lee, M.K. Chung, H. Kang, H. Choi, S. Ha, Y. Huh, E. Kim, D.S. Lee, Coidentification of group-level hole structures in brain networks via Hodge Laplacian, in: Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part IV 22, Springer, 2019, pp. 674–682.

[164] D.V. Anand, M.K. Chung, Hodge Laplacian of brain networks, IEEE Trans. Med. Imaging 42 (5) (2023) 1563–1573.

[165] S. Dakurah, D.V. Anand, Z. Chen, M.K. Chung, Modelling cycles in brain networks with the Hodge Laplacian, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2022, pp. 326–335.

[166] D.V. Anand, M.K. Chung, Hodge-decomposition of brain networks, in: 2024 IEEE International Symposium on Biomedical Imaging, ISBI, IEEE, 2024, pp. 1–5.

[167] F. Nasrin, C. Oballe, D. Boothe, V. Maroulas, Bayesian topological learning for brain state classification, in: 2019 18th IEEE International Conference on Machine Learning and Applications, ICMLA, IEEE, 2019, pp. 1247–1252.

[168] J. Huang, M.K. Chung, A. Qiu, Heterogeneous graph convolutional neural network via Hodge-Laplacian for brain functional data, in: International Conference on Information Processing in Medical Imaging, Springer, 2023, pp. 278–290.

[169] J. Park, Y. Hwang, M. Kim, M.K. Chung, G. Wu, W.H. Kim, Convolving directed graph edges via Hodge Laplacian for brain network analysis, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2023, pp. 789–799.

[170] Y. Hwang, S. Hwang, G. Wu, W.H. Kim, Multi-order simplex-based graph neural network for brain network analysis, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2024, pp. 532–541.

[171] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, A. Vespignani, Epidemic processes in complex networks, Rev. Modern Phys. 87 (3) (2015) 925–979.

[172] X. Yan, B.M. Sadler, R.J. Drost, L.Y. Paul, K. Lerman, Graph filters and the Z-Laplacian, IEEE J. Sel. Top. Signal Process. 11 (6) (2017) 774–784.

[173] A. Tomy, M. Razzanelli, F. Di Lauro, D. Rus, C. Della Santina, Estimating the state of epidemics spreading with graph neural networks, Nonlinear Dynam. 109 (1) (2022) 249–263.

[174] Z. Liu, G. Wan, B.A. Prakash, M.S. Lau, W. Jin, A review of graph neural networks in epidemic modeling, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 6577–6587.

[175] S. Barbarossa, D. Comminiello, E. Grassucci, F. Pezone, S. Sardellitti, P. Di Lorenzo, Semantic communications based on adaptive generative models and information bottleneck, IEEE Commun. Mag. 61 (11) (2023) 36–41.

[176] E.C. Strinati, P. Di Lorenzo, V. Sciancalepore, A. Aijaz, M. Kountouris, D. Gündüz, P. Popovski, M. Sana, P.A. Stavrou, B. Soret, et al., Goal-oriented and semantic communication in 6G AI-native networks: The 6G-GOALS approach, in: Proc. of EuCNC 2024, 2024, arXiv preprint arXiv:2402.07573.

[177] F. Baccini, F. Geraci, G. Bianconi, Weighted simplicial complexes and their representation power of higher-order network data and topology, Phys. Rev. E 106 (3) (2022) 034319.

[178] H. Riihimäki, Simplicial-connectivity of directed graphs with applications to network analysis, SIAM J. Math. Data Sci. 5 (3) (2023) 800–828.

[179] X. Gong, D.J. Higham, K. Zygalakis, G. Bianconi, Higher-order connection Laplacians for directed simplicial complexes, J. Phys.: Complex. 5 (1) (2024) 015022.

[180] M. Lecha, A. Cavallo, F. Dominici, E. Isufi, C. Battiloro, Higher-order topological directionality and directed simplicial neural networks, 2024, arXiv preprint arXiv:2409.08389.