Quest Journals
Journal of Research in Applied Mathematics
Volume 11 ~ Issue 12 (December 2025) pp: 135-144
ISSN (Online): 2394-0743 ISSN (Print): 2394-0735
www.questjournals.org

Research Paper

Topological Signal Processing and Machine Learning: From Spectral and Hodge Foundations to Applications

Adel Ahmed Hassan Kubba[(1)], Bastamy Mohamed Elmahadi [(2)]
[1]*Associate Professor - Nile Valley University - Faculty of Education*
[2]*Teacher Math*

Abstract:
The analysis of complex, non-Euclidean data is rapidly evolving, necessitating advanced mathematical methodologies that transcend the limitations of traditional Euclidean-based signal processing techniques. This paper provides a comprehensive survey of the field of Topological Signal Processing (TSP), which integrates Spectral Graph Theory (SGT) and Topological Data Analysis (TDA). The paper focuses on representing and processing signals defined on topological structures such as Simplicial Complexes and Cell Complexes, which generalize the concept of graphs to represent higher-order relationships (like edges and triangles). Key theoretical foundations like Persistent Homology and the Mapper Algorithm are outlined. Furthermore, core algebraic concepts are explained, including Incidence Matrices, Hodge Laplacians, and the Hodge Decomposition, which dissects signals into gradient, curl, and harmonic components. The paper also introduces the Dirac Operator and its spectral analysis, enabling joint processing of signals across different simplicial orders. Finally, the paper reviews a wide range of promising applications in areas such as: infrastructure network state estimation, geometry processing, statistical ranking, biomolecular data analysis, brain networks, epidemic modeling, and semantic communication.

Keywords: *Topological Signal Processing (TSP), Topological Learning, Higher-Order Networks, Topological Data Analysis (TDA), Persistent Homology, Hodge Decomposition, Hodge Laplacian, Dirac Operator.*

*Received 13 Dec., 2025; **R**evised 24 Dec., 2025; **A**ccepted 26 Dec., 2025 © The author(s) 2025.*
P**ublished with open access at **www.questjournas.org

I. Introduction

Spectral Graph Theory (SGT) is a field of mathematics that studies the properties of graphs through the eigenvalues and eigenvectors of matrices associated with them. It provides a bridge between graph theory and linear algebra, enabling powerful analytical tools for understanding the structure of networks. The core idea of SGT is that the spectral properties of a graph (such as its Laplacian matrix and its spectrum) can reveal key insights about connectivity, clustering, and diffusion processes within the graph. This makes SGT an essential tool in various domains, including machine learning, data science, network analysis, and physics. As data grows increasingly complex, high-dimensional, and heterogeneous, traditional analysis methods often fall short in capturing global geometric structures or relationships intrinsic to the data. This shortcoming has fueled the growth of Topological Data Analysis (TDA)-a mathematical and computational framework that uses principles from algebraic topology to study the "shape" of data. Topology, traditionally concerned with qualitative geometric features such as connectivity and holes, becomes highly relevant in modern data settings where Euclidean metrics or linear projections may obscure deeper structural patterns. 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. 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. [1]

II. Theoretical Foundations

At its core, TDA is grounded in algebraic topology, particularly the study of homology, which quantifies topological features such as connected components, loops, and voids. These features are described using Betti numbers:

- β_0: Number of connected components
- β_1: Number of one-dimensional holes (loops)
- β_2: Number of two-dimensional voids, and so on.

The key motivation is that meaningful geometric structure within data—such as cycles or clusters—can be captured and compared across scales using homological tools (Carlsson, 2009). The simplicial complex, a generalization of graphs to higher dimensions, is the data structure most commonly used to encode these features. Among the most prominent types are the Vietoris-Rips, Čech, and Alpha complexes.

Persistent Homology(2.1)[2] Persistent homology is perhaps the most widely studied and applied technique within TDA. Rather than focusing on a fixed-scale topological structure, it tracks how topological features (such as holes and connected components) appear and disappear across a filtration of simplicial complexes. This approach provides a multiscale summary of the dataset's topology, often visualized as:

- **Barcodes:** Horizontal line segments showing the lifespan of each feature
- **Persistence diagrams:** Birth vs. death coordinate plots
- **Persistence landscapes and images:** functional and vectorized representations for machine learning compatibility.

Mapper Algorithm (2.2)[2] While persistent homology extracts topological summaries, the mapper algorithm provides a tool for visualizing and clustering data using topological principles. First introduced by Singh, Mémoli, and Carlsson (2007), the mapper algorithm constructs a simplicial complex representation of data by:

i. Mapping data through a filter function (e.g., principal component, density, or eccentricity),
ii. Covering the range of the filter with overlapping intervals,
iii. Clustering the points within each interval and connecting clusters that share points.

The resulting network-like structure helps visualize the global shape of the data, revealing branches, loops, or flares indicative of important substructures. Mapper has been successfully applied in patient stratification (e.g., breast cancer survival), microbial ecology, and consumer behavior studies.

Software and Algorithmic Advances (2.3)[2] The computational challenges associated with TDA, particularly persistent homology, have spurred extensive software development.

III. Knowledge Representation over Topological Spaces

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, before extending these concepts to topological domains.

Signals on Graphs(3.1)[4] Let us consider a graph $G = (V, E)$ consisting of a set of N vertices $V = \{1,2,...,N\}$, along with a set of E edges $E \subset V \times V$. Let us also denote the $N \times N$ adjacency matrix as A where the element (i, j) is denoted as $A_{i,j}$, $i, j \in V$. We assume $A_{i,j} > 0$, if there is a link from node j to node i, i.e., $(j, i) \in E$, or $A_{i,j} = 0$, otherwise. The combinatorial Laplacian matrix for an undirected graph with a symmetric adjacency matrix A is defined as $L_0 = \text{diag}(1^{\top}A) - A$.

Alternatively, a graph can be represented by its incidence matrix that encodes the incidence relations between vertices and edges. To define the incidence matrix 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 B_1 are defined as follows:

$$[B_1]_{i,j} = \begin{cases} 0, & \text{if node i is not incedent on edge j} \\ 0, & \text{if node i is the head of arrow j} \\ -1, & \text{if node i is the tail of arrow j} \end{cases} \quad (3.1)$$

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

$$L_9 = B_1 B_1^{T}, \quad (3.2)$$

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

A signal $\mathbf{x} = [x_1,...,x_N]^{\top}$ over a graph G is a mapping from the vertex set to the set of real numbers, i.e., $x : V \rightarrow R$. Here, entry x_i is the signal value associated to node $i \in V$. 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 A or L_0 translate into proximities between the respective signals. Such a coupling can then be used for processing signal x by relying on neighboring information in a similar way as we process temporal and image signals based on temporal or spatial proximities.

Signals on Topological Spaces (3.2) [4] Consider a finite set of vertices V. A k-simplex $H_{k,i}$ is a subset of V with cardinality k + 1. A face of $H_{k,i}$ is a subset with cardinality k and consequently a k-simplex has k+1 faces. A coface of $H_{k,i}$ is a (k+1)-simplex that includes $H_{k,i}$ 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 XK of order K, is a collection of k-simplices $H_{k,i}$, k = 0,...,K such that, if a simplex $H_{k,i}$ belongs to X_K, then all its subsets $H_{k-1,i} \subset Hk,i$ also belong to XK (inclusivity property). The set of k-simplices in X_K is denoted by $D_k := \{H_{k,i} : H_{k,i} \in X_K\}$, with its cardinality represented as $|D_k| = N_k$.
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 orientation for a simplex according to the lexicographical ordering of its vertices. See an example in Figure 3.1.
A k-simplicial signal is a collection of mappings from the set of all k simplices contained in the complex to real numbers:

$$x_k = [x_k(H_{k,1}),...,x_k(H_{k,i}),...,x_k(H_k,N_k)]^T \in R^{Nk}, \quad (3.3)$$

where $x_k : D_k \rightarrow R$. 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:

$$x_X = \left[x_0 \| \ldots \| x_k\right] \in R^{\sum_{k=0}^{k} N_k}. \quad (3.4)$$

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

$$x_0 : V \rightarrow R, \quad x_1 : E \rightarrow R, \quad x_2 : T \rightarrow R, \quad (3.5)$$

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

$$x_X = \left[x_0 \| x_1 \| x_k\right] \in R^{N+E+T}. \quad (3.6)$$

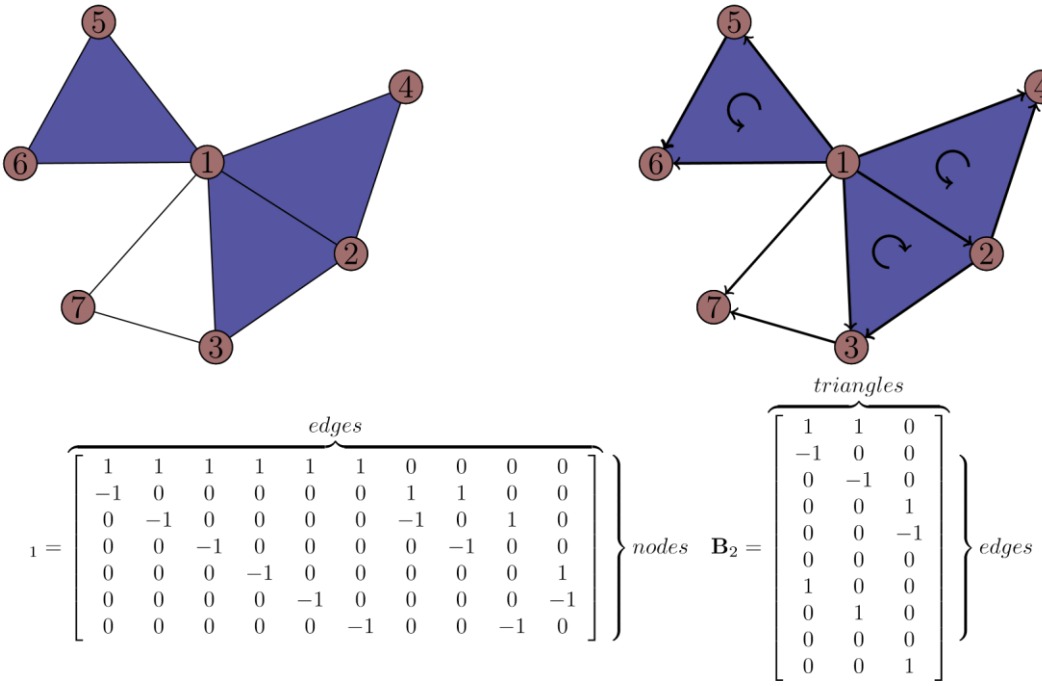

Figure 3.1: (top-left) A simplicial complex of order K = 2.

It is composed of 7 nodes, 10 edges, and 3 filled triangles (shaded areas). Example of simplices: $H_{0,1} = \{1\}$, $H0_{,2} = \{2\}$, $H0_{,3} = \{3\}$; $H1_{,1} = \{1,2\}$, $H1_{,2} = \{1,3\}$, $H_{1,2} = \{2,3\}$; $H_{2,1} = \{1,2,3\}$, $H_{2,2} = \{1,2,4\}$, $H_{2,3} = \{1,5,6\}$. 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 lexicography ordering of its vertices. Example of oriented simplices:
$H_{1,1} = [1,2]$, $H_{1,2} = [1,3]$, $H_{1,2} = [2,3]$; $H_{2,1} = [1,2,3]$, $H_{2,2} = [1,2,4]$, $H_{2,3} = [1,5,6]$. (bottom) Incidence matrices of the oriented simplicial complex.

Remark (3.3)[5] (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, i.e., given a simplex $H_{k,i} \subset XK$, not all subsets $H_{k-1,i} \subset H_{k,i}$ need to belong to 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 Figure 4.3.2, where the presence of the quadrilateral does not imply the presence of triangles or diagonal edges within the quadrlateral itself. For more details on the differences between simplicial and cell complexes, whereas for the role such differences induce in signal representation

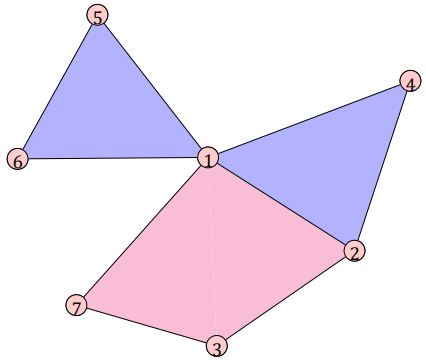

Figure 3.2: A regular geometric cell complex of order K = 2.

It is composed of 7 nodes, 9 edges, 2 filled triangles (shaded blue areas), and 1 filled polygon.

Algebraic Representation (3.4)[5] The structure of a simplicial complex XK is completely characterized by its set of incidence matrices B_k, k = 1,...,K. Extending the concept from graphs, the entries of the incidence matrix B_k specify which k-simplices are incident to which (k − 1)-simplices. We use the notation $H_{k-1,i} \sim H_{k,j}$ to indicate two simplices with the same orientation, and $H_{k-1,i} \sim H_{k,j}$ to indicate that they have opposite orientation. The entries of B_k are defined as:

$$[B_k]_{i,j} = \begin{cases} 0, & \text{if } H_{k-1,i} \not\subset H_{k,j} \\ 0, & \text{if } H_{k-1,i} \subset H_{k,j} \text{ and } H_{k-1,i} \sim H_{k,j} \\ -1, & \text{if } H_{k-1,i} \subset H_{k,j} \text{ and } H_{k-1,i} \sim H_{k,j} \end{cases} \quad (3.7)$$

As an example, considering a simplicial complex X_2 of order two, we have two incidence matrices: the node-to-edge incidence matrix, $B_1 \in R^{N\times E}$, and the edge-to-triangle incidence matrix $B_2 \in R^{E\times T.}$ The latter are illustrated in Figure 3.1. From the incidence information, we can build the higher-order Hodge Laplacian matrices, of orders k = 0,...,K, as:

$$L_0 = B_1 B_1^T , \quad (3.8)$$

$$L_k = \underbrace{B_1^T B_k}_{L_k^{(d)}} + \underbrace{B_{k+1}^T B_{k+1}}_{L_k^{(u)}} = \quad k = 1,..., K-1, \quad (3.9)$$

$$L_K = B_K^T B_K , \quad (3.10)$$

Here, L_0 is the combinatorial graph Laplacian defined and captures node-to-node proximities. All Laplacian matrices of intermediate orders k =1,..., K−1, contain two terms: The first term $L_k^{(d)}$ known as down Laplacian, encodes the lower adjacency of k-order simplices; the second term $L_k^{(u)}$, 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 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.

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).

Hodge Decomposition (3.5) [6] One useful property of the Hodge Laplacian is their link with the Hodge decomposition. In particular, given the k-th intermediate Hodge Laplacian $L_k = B^T_k B_k + B_{k+1}B^T_{k+1}$, 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:

$$\mathbb{R}^{N_k} = \underbrace{\operatorname{spn}\left(B_k^T\right)}_{\text{(a) gradient space}} \oplus \underbrace{\operatorname{spn}\left(B_{k+1}\right)}_{\text{(b) curlspace}} \oplus \underbrace{\operatorname{spn}\left(B_k\right)}_{\text{(c) harmonoct space}} . \quad (3.11)$$

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

$$x_k = \underbrace{B_k^T \tilde{x}_{k-1}}_{\text{(a) gradient space}} + \underbrace{B_{k+1}\tilde{x}_{k+1}}_{\text{(b) curlspace}} + \underbrace{\tilde{x}_k}_{\text{(c) harmonoct space}} . \quad (3.12)$$

That is, there exists three signals $\tilde{x}_{k-1}, \tilde{x}_k, \tilde{x}_{k+1}$ of respective orders k −1, k, k + 1 that can express the signal. In other words, the decomposition in (3.12) 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 (3.11) and (3.12) carry the following interpretation when discussing edge signals x_1 (i.e., k = 1):
i) Gradient space and gradient component: The space span (B^T_1) is called the gradient space. An edge flow signal $\tilde{x}_{1,g} B_1^T \tilde{x}_0$ living in this space is referred to as a gradient flow and it can be obtained by differentiating node signals $\tilde{x}_0$ along the edges connecting them. The component of an edge signal x_1 living in the gradient space is referred to as the gradient component (a.k.a., the irrotational component).
ii) Curl space and curl component: The space span(B_2) is called the curl space. An edge flow signal $x_{1,c} = B_2 \tilde{x}_2$ living in this space is referred to as a curl flow and it can be induced by some triangle signals $\tilde{x}_2$. The component of an edge signal x_1 living in the curl space is referred to as the curl component (a.k.a., the solenoidal component).
iii) Harmonic space and harmonic component: The space kernel (L_k) is called the harmonic space. An edge flow signal $\tilde{x}_k$ 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 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.

IV. Dirac Operator and Dirac Decomposition

The Hodge Laplacian and Hodge decomposition are conventionally used to represent, analyses, 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.

Dirac operator (4.1) [7] Focusing for simplicity on a simplicial complex X_2 of order K = 2, the Dirac operator reads as:

$$D_X := \begin{bmatrix} 0 & B_1 & 0 \\ B_1^T & 0 & B_2 \\ 0 & B_2^T & 0 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & B_1 & 0 \\ B_1^T & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{D_X^{(d)}} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & B_2 \\ 0 & B_2^T & 0 \end{bmatrix}}_{D_X^{(u)}} \quad (3.13)$$

where we refer to $D_X^{(d)}$ and $D_X^{(u)}$ 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., $D^2_X = \text{blading}(L_0,L_1,L_2)$.

Application to topological complex signals (4.2) [7] When applied to a SC signal, the Dirac operator yields the shifted signal:

$$x_X^{(1)} = \begin{bmatrix} x_0^{(1)} \\ x_1^{(1)} \\ x_2^{(1)} \end{bmatrix} = D_X x_X = \begin{bmatrix} B_1 x_1 \\ B_1^T x_0 + B_2 x_2 \\ B_2^T x_1 \end{bmatrix}. \quad (3.14)$$

The shifted node signal $x_0^{(1)} = B_1 x_1$ sums the edge flows flowing into a node and is referred to as the divergence of the edge flow; the shifted edge signal $x_1^{(1)} = B_1^T x_0 B_2 x_2$ 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 $x_2^{(1)} = B_2^T x_1$ is the curl signal of the edge flow. The operation in (3.14) 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, or design principled topological neural architectures.

Dirac decomposition (4.3) [8] Similar to the Hodge decomposition in (3.11), the Dirac decomposition states that the signal space associated with a topological signal x_X can also be decomposed as the direct sum of the following three orthogonal subspaces reminiscent of (3.13):

$$R^{\sum_{k=0}^{K} N_k} = \underbrace{\text{span}\left(D_X^{(d)}\right)}_{\text{(a) joint gradient space}} \oplus \underbrace{\text{span}\left(D_X^{(u)}\right)}_{\text{(b) joint curl space}} \oplus \underbrace{\text{spn}\left(D_X\right)}_{\text{(c) joint harmonoct space}}. \quad (3.15)$$

where by following the same terminology as for the Hodge decomposition, we refer to (a) span($D^{(d)}_{\mathcal{X}}$) as the joint gradient space; (b) span($D^{(u)}_{\mathcal{X}}$) as the joint curl space; and (c) kernel(D_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 $\mathbf{x}_X$ into the sum of a signal $\tilde{x}_X^{(d)} \in \text{span}(D_X^{(d)})$, a signal $\tilde{x}_X^{(u)} \in \text{span}(D_X^{(u)})$, and a harmonic signal $\tilde{x}_X^{(u)} \in \ker\text{nel}(D_X)$ e.,

$$x_k = \underbrace{x_X^{(d)}}_{\text{(a) joint gradient space}} + \underbrace{x_X^{(d)}}_{\text{(b) joint curl space}} + \underbrace{\tilde{x}_X}_{\text{(c) joint harmonoct space}}. \quad (3.16)$$

Spectral Processing(4.4)[8] 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 and then define the more general topological Fourier transform. We conclude by discussing the spectral duality for joint topological complex signals.

Graph Fourier Transform (4.5)[8] Given an undirected graph G = (V, E) with graph Laplacian L and graph signal x, the graph Fourier transform (GFT) of signal x is the signal projection onto the Laplacian eigenspace. More specifically, given the eigen decomposition $L = U\Lambda U^H$, the GFT of x is

$\hat{X} = U^H x$. The eigenvectors $U = [u_1,...,u_N]$ serve as the spectral basis expansion for the graph signal x, and the GFT coefficient $\hat{x}_i$ is the weight indicating how much eigenvector u_i contributes to represent the signal. Following the analogy with the classical Fourier transform, the eigenvalues in $\Lambda = diag(\lambda_1,...,\lambda_N)$ contain the so-called graph frequencies.

The Fourier notion of the projection $\hat{X} = U^H x$ comes from the fact that we can view each eigenvector $u_i = [u_{i1},...,u_{iN}]^T$ as a graph signal and analyze its variability w.r.t. the graph G. One way to do this is via the quadratic variation $QV(u_i) = u^H_i Lu_i = \lambda_i$, which indicates how smooth u_i is over the graph G. Thus, we can sort the eigenvectors based on their variability

$0 = QV(u_1) \leq QV(u_2) \leq ... \leq QV(u_N)$, which implies an ordering of the graph frequencies $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_N$. 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 u_i contributes to the variability of the graph signal x. We shall discuss next that a similar, yet slightly more involved, Fourier analysis can be derived also for topological signals of any order.

Topological Fourier Transform (4.6)[9] As for the graph Laplacian, any Hodge Laplacian of order k enjoys an eigen decomposition of the form

$$L_k = L_k^{(d)} + L_k^{(u)} = B_k^T B_k + B_{k+1} B_{k+1}^T = U_k \Lambda_k U_k^T \quad (3.17)$$

with orthogonal eigenvector matrix $U_k = [u_{k,1},...,u_{k,Nk}] \in R^{Nk \times Nk}$ and eigenvalue matrix $\Lambda_k = diag(\lambda_{k,1},...,\lambda_{k,Nk})$. Then, the topological Fourier transform (TFT) of a signal x_k is given by the projection onto the eigenvectors U_k, i.e., $\hat{x}_k = U_k^T x_k$. As for the GFT, the i-th entry of $\hat{x}_k$, i.e., $\hat{x}_{k,i}$ represents the weight of eigenvector $u_{k,i}$ in expressing signal x_k. The inverse TFT is given by $x_k = U_k \hat{x}_k$. Notice that the GFT 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 eigen decomposition has a correspondence with the Hodge decomposition in (3.11). More specifically, it is possible to rearrange the eigenvectors in U_k and eigenvalues in Λ_k respectively as:

$$U_k = [U_{k,g}, U_{k,c}, U_{k,h}] \text{ and } \Lambda_k = blkdiag(\Lambda_{k,g}, \Lambda_{k,c}, \Lambda_{k,h}). \quad (3.18)$$

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

i) Gradient space and gradient frequencies: The eigenvectors $U_{1,g} \in R^{N1 \times Ng}$ span the gradient space span(B^T_1) with dimension N_g. Hence, the component $\hat{x}_{1,g} = U_{1,g}^T x_1$ represents the gradient component of the TFT. The eigenvectors $u_g \in U_{1,g}$ also carry a notion of variability via the quadratic variation w.r.t. the Hodge Laplacian L_1, i.e.,

$$QV(u_g) = u_g^T L_1 u_g = \|B_1 u_g\|_2^2 + \|B_2 u_g\|_2^2 = \|B_1 u_g\|_2^2 = \lambda_g \quad (3.19)$$

where $B^T_2 u_g = 0$ since u_g is a gradient flow. Thus, the eigenvalue λ_g is the squared ℓ_2−norm of the divergence $B_1 u_g$ 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 $0 < QV(u_{g,i}) \leq QV(u_{g,j})$ implying an ordering of the eigenvalues $0 < \lambda_{g,i} < \lambda_{g,j}$ for $i, j \in \{1,...,N_g\}$. 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 $\mathbf{U}_{1,g}$ as the gradient frequencies.

ii) Curl space and curl frequencies: Analogously, the eigenvectors $U_{1,c} \in R^{N \times N}$ span the curl space span(B_2) with dimension N_c. The projection $\hat{x}_{1,g} = U_{1,g}^T x_1$ represents the curl component of the TFT, where the eigenvectors $u_c \in U_{1,c}$ do carry a notion of variability that is different from that seen above. Using again the quadratic variation w.r.t. the L_1 Hodge Laplacian we have

$$QV(u_c) = u_c^T L_1 u_c = \|B_1 u_c\|_2^2 + \|B_2 u_c\|_2^2 = \|B_1 u_c\|_2^2 = \lambda_c \quad (3.20)$$

where $B_1 u_c = 0$ since u_c is a curl flow. Now, the eigenvalue λ_c is the squared ℓ_2−norm of the total curl $B^T_2 u_c$ 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 $0 < QV(u_{c,i}) \leq QV(u_{c,j})$ implying an ordering of the curl eigenvalues $0 < \lambda_{c,i} < \lambda_{c,j}$ for $i,j \in \{1,...,N_c\}$. We refer to the eigenvalues λ_c associated to the curl eigenvectors $U_{1,c}$ as the curl frequencies.

iii) Harmonic space and harmonic frequencies: Finally, the eigenvectors

$U_{1,h} \in R^{N1 \times Nh}$ span the harmonic space kernel(L_k) with dimension N_h. The projection $\hat{x}_{1,h} = U^T_{1,h} x_1$ represents the harmonic component of the TFT, where the eigenvectors $u_h \in U_{1h}$ are all associated to the zero eigenvalue since

$$QV(u_h) = u_h^T L_1 u_h = \|B_1 u_h\|_2^2 + \|B_2 u_h\|_2^2 = \|B_1 u_h\|_2^2 = \lambda_h. \qquad (3.21)$$

Consequently, we will refer to the eigenvalues $\lambda_h = 0$ 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.

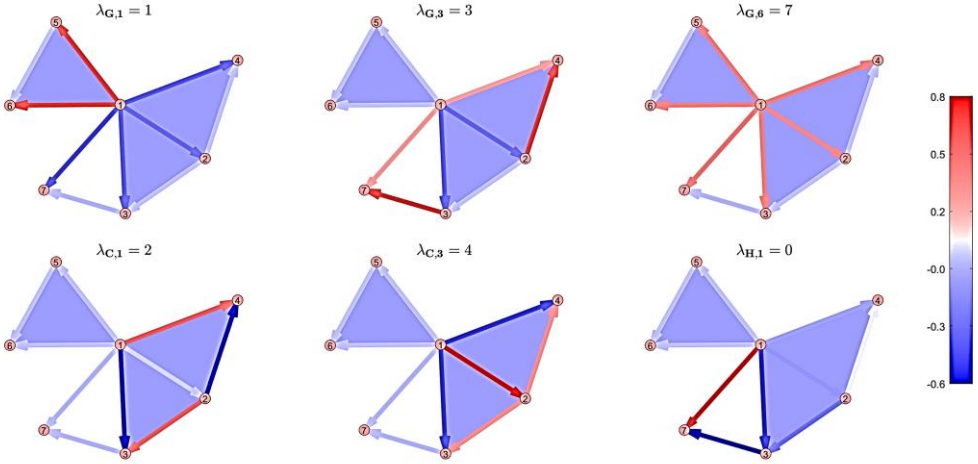

Figure 3.3: Eigenvectors of the L_1 Hodge Laplacian shown as edge flow signal along with their respective eigenvalues. (top-row) gradient eigenvectors with increasing total divergence. (bottom row) curl eigenvectors with increasing total curl variation, and the harmonic eigenvector. The latter is localized around the hole in the simplicial complex.

The above discussion shows that the notion of low and high frequency in a SC is only meaningful within a certain type. An illustration of the latter is shown in Figure 3.3. This behavior is a unique characteristic of topological spaces that is not conventionally seen in graph and discrete signal processing.

Remark (4.7)[9] Notice that the graph Laplacian can also be written as $\mathbf{L} = B_1 B^T_1$, which implies that the space of all graph signals, i.e., k = 0, enjoys also a Hodge decomposition of the form $R^{N0} = \text{span}(B_1) \oplus \text{kernel}(L)$. Since from a topological perspective, the case k = 0 implies that nodes are connected only via the upper Laplacian, the space $\text{span}(B_1)$ is the analogous of the curl space in (3.11), whereas kernel(L) is the harmonic space. Consequently, the GFT accounts for the projection of the graph signals onto these subspaces, where the eigenvectors U associated with a non-zero eigenvalue span the curl space $\text{span}(B_1)$ and those associated with a zero eigenvalue span the harmonic space kernel(L).

V. 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.

Processing network flow signals(5.1)[10] 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. One such case has been discussed 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 network has been used 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. Simplicial neural networks and Hodge

representation of networks has also been used for power outage detection for false data injection attacks in smart grids. When such anomalies can be localized on particular Hodge subspaces, a mathematically tractable matched subspace detector with optimality guarantees 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, 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 (5.2)[10] Historically, point cloud processing has been approached by graph-based techniques on the 3D mesh. Such a paradigm has in fact been the early roots of developing graph signal processing techniques till the latest progress with graph neural networks. 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.

Statistical ranking (5.3) [10] Hodge theory and decompositions have shown great potential in statistical ranking of lists. 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 a harmonic or curl flow would show local or global inconsistencies in rankings. This Hodge-based statistical ranking (Hodge Rank) has been successfully applied to top-N recommender system lists and currency exchange markets as well as in biomolecular data analysis. 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 can be used to learn deep statistical ranking models.

Biomolecular data (5.4)[10] A key challenge in graph-based learning is to classify molecules or even synthesize new one from a limited training set. In fact, many of the message-passing TNN advances have been developed to overcome the limited Weis Feiler-Lehman expressivity of GNNs in graph classification tasks. Molecules in particular can be seen as structurally rich graphs with (hidden) topological information that are combined their feature. By inducing higher-order simplicial or cell structures within a molecular graph –a technique known as lifting– TNNs can leverage these topological relations to learn more expressive representations

Brain networks(5.5)[10] Topological processing tools have been also largely applied to represent and extract information from brain network data. Specifically, the work 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 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 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.

Epidemic modeling(5.6)[10] Modeling and predicting epidemic spreading represent one of the cornerstone applications network sciences. 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.

VI. Conclusion:

Topological Signal Processing (TSP) is an emerging mathematical framework that analyzes complex, non-Euclidean data by integrating spectral graph theory with algebraic topology. It generalizes signal processing to higher-order structures like simplicial and cell complexes, utilizing key operators such as the Hodge Laplacian for spectral analysis and the Hodge Decomposition to separate signals into gradient, curl, and harmonic components. The Dirac operator further enables unified processing across different simplicial orders. With applications spanning infrastructure monitoring, brain network analysis, biomolecular science, epidemic modeling, and semantic communications, TSP provides a powerful and interpretable paradigm for uncovering the rich, global structure inherent in modern complex datasets.

References:

[1]. Carlsson, G. (2009). Topology and data. *Bulletin of the American Mathematical Society*, *46*(2), 255-308.
[2]. Singh, G., Mémoli, F., & Carlsson, G. E. (2007). Topological methods for the analysis of high dimensional data sets and 3D object recognition. In *Eurographics Symposium on Point-Based Graphics*.
[3]. Hatcher, A. (2002). *Algebraic Topology*. Cambridge University Press.
[4]. Chung, F. R. (1997). *Spectral Graph Theory*. American Mathematical Society.
[5]. Schaub, M. T., Benson, A. R., Horn, P., Lippner, G., & Jadbabaie, A. (2020). Random walks on simplicial complexes and the normalized Hodge 1-Laplacian. *SIAM Review*, *62*(2), 353-391.
[6]. Ebli, S., Defferrard, M., & Spreemann, G. (2020). Simplicial neural networks. In *Neural Information Processing Systems (NeurIPS) Workshop on Topological Data Analysis and Beyond*.
[7]. Roddenberry, T. M., Glaze, S., & Segarra, S. (2021). Principled simplicial neural networks for trajectory prediction. In *International Conference on Machine Learning (ICML)*.
[8]. Jiang, X., Bian, X., & Liang, Y. (2021). Topological signal processing and learning for brain networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
[9]. Jia, J., & Benson, A. R. (2019). Neural jump stochastic differential equations. In *Advances in Neural Information Processing Systems (NeurIPS)*.
[10]. Jiang, J., Gu, Y., & Ji, S. (2022). Rethinking graph neural networks for graph coloring.