Hodge-Aware Convolutional Learning on Simplicial Complexes

Anonymous authors
Paper under double-blind review

Abstract

Neural networks on simplicial complexes (SCs) can learn representations from data residing on simplices such as nodes, edges, triangles, etc. However, existing works often overlook the Hodge theorem that decomposes simplicial data into three orthogonal characteristic subspaces, such as the identifiable gradient, curl and harmonic components of edge flows. This provides a universal tool to understand the machine learning models on SCs, thus, allowing for better principled and effective learning. In this paper, we study the effect of this data inductive bias on learning on SCs via the principle of convolutions. Particularly, we present a general convolutional architecture that respects the three key principles of uncoupling the lower and upper simplicial adjacencies, accounting for the inter-simplicial couplings, and performing higher-order convolutions. To understand these principles, we first use Dirichlet energy minimizations on SCs to interpret their effects on mitigating simplicial oversmoothing. Then, we show the three principles promote the Hodge-aware learning of this architecture, through the lens of *spectral simplicial theory*, in the sense that the three Hodge subspaces are invariant under its learnable functions and the learning in two nontrivial subspaces is independent and expressive. Third, we investigate the learning ability of this architecture in optic of perturbation theory on simplicial topologies and prove that the convolutional architecture is stable to small perturbations. Finally, we corroborate the three principles by comparing with methods that either violate or do not respect them. Overall, this paper bridges learning on SCs with the Hodge theorem, highlighting its importance for rational and effective learning from simplicial data, and provides theoretical insights to convolutional learning on SCs.

1 Introduction

In the line of geometric deep learning (Bronstein et al., 2021), there is a growing interest in learning from data defined on simplicial complexes. The motivation behind this comes from two limitations of standard graph neural networks (GNNs). First, graphs are limited to model pairwise interactions between data entites on nodes, yet polyadic (multi-way) interactions often arise in real-world networks (Battiston et al., 2020; Benson et al., 2021; Torres et al., 2021), such as friendship networks (Newman et al., 2002), collaboration networks (Benson et al., 2018), gene regulatory networks (Masoomy et al., 2021). Second, graphs are often used to support signals on the nodes, and standard graph signal processing and GNN approaches often revolve around signals and features on nodes. Yet, signals involved with multiple entities are less researched compared to signals on nodes (with one entity). They arise as signal flows on edges, signals on triangles and so on. For example, in physical networks, we may encounter water flows in a water supply network (Money et al., 2022), traffic flows in a road network (Jia et al., 2019), trading flows in financial networks (Lim, 2020) and information flows in brain networks (Anand et al., 2022), as well as in human-generated networks, we have collaboration data, such as triadic collaborations in coauthorship networks (Benson et al., 2018).

Simplicial complexes are a popular higher-order network model and have been shown effective to address both limitations of graph-based models (Bick et al., 2021). They are composed of topological objects, namely, nodes, edges, triangles, etc., which are simplices of different orders. Simplicial complexes naturally describe more topological (higher-order) relationships in networks, thus, having more topological expressive power

than graphs. This has been the main motivation of recent neural networks developed on simplicial complexes (Roddenberry & Segarra, 2019; Bunch et al., 2020; Ebli et al., 2020; Roddenberry et al., 2021; Bodnar et al., 2021b; Chen et al., 2022b; Giusti et al., 2022). We also refer readers to the recent surveys (Papamarkou et al., 2024; Besta et al., 2024). In analogy to standard GNNs relying on the adjacency between nodes, the central idea behind these works is to rely on the relationships between simplices to enable learning. Such relations can be twofold: first, two simplices can be lower and upper adjacent to each other, e.g., an edge can be (lower) adjacent to another via a shared node, and can also be (upper) adjacent to another by locating in a common triangle; and second, there exist inter-simplicial couplings (or simplicial incidences) between simplices of different orders, as shown in Fig. 1a. The aforementioned works mainly vary in terms of either message-passing or convolutional flavor, or the type of simplicial relationships relying, either on only simplicial adjacencies or on both adjacencies and incidences.

Furthermore, signals can be defined on simplices to model the data related to multiple entites in networks. This has been the main focus of topological signal processing literature Barbarossa & Sardellitti (2020); Schaub et al. (2021); Yang et al. (2022a). The celebrated *combinatorial Hodge decomposition* arising from discrete calculus (Grady & Polimeni, 2010; Lim, 2020) provides a unique and characteristic decomposition of simplicial signals into three components. This is particularly intuitive for edge flows which allows their decomposition into *gradient flows, curl flows and harmonic flows*, that are, respectively, curl-free, divergence-free or both. These notions from discrete calculus interestingly allow us to capture some physical properties of the simplicial signals, such as the conservation laws (Grady & Polimeni, 2010). More importantly, this decomposition offers a tool to better analyze simplicial signals, as reported in statistical ranking problems, financial exchange markets (Jiang et al., 2011), traffic networks (Jia et al., 2019), brain networks (Anand et al., 2022) and game theory (Candogan et al., 2011). We hypothesize it will further promote better principled and effective learning methods on simplicial complexes.

Given this context, we reckon that the aforementioned works on simplicial neural networks mostly focus on the pure topological aspect of simplicial complexes. It lacks theoretical analyses of their learning capabilities from the Hodge spectral perspective. Also, since SCs are often built from data and are prone to estimation uncertainty, the learning on SCs benefits from a stability analysis to investigate their robustness against perturbations on the simplicial topologies. Thus, in this paper, after reviewing some background on simplicial complexes and simplicial signals in Section 2, we propose a more general and unified framework, namely, *simplicial complex convolutional neural network* (SCCNN), and we focus on the following three theoretical aspects.

Contributions

- In Section 3 we introduce SCCNN and emphasize its three principles, namely, uncoupling the lower and upper simplicial adjacencies, accounting for the inter-simplicial couplings, and performing higher-order convolutions. We then use the Dirichlet energy minimization on SCs to understand how uncoupling the lower and upper adjacencies in Hodge Laplacians, as well as the inter-simplicial couplings can mitigate simplicial oversmoothing.
- In Section 4, we characterize the spectral behavior of SCCNN and its expressive power under the help of spectral simplicial theory (Steenbergen, 2013; Barbarossa & Sardellitti, 2020; Yang et al., 2021). We show that an SCCNN performs independent and expressive learning in the three subspaces of the Hodge decomposition, which are invariant under its learning operators. This Hodge-awareness (or Hodge-aided bias) allows for effective and rational learning on SCs compared to MLPs or simplicial message-passing networks (Bodnar et al., 2021b).
- In Section 5, we obtain a theoretical stability bound on the SCCNN outputs against small perturbations on the simplicial connections. This allows us to see how the three principles and other network factors can affect the stability, as well as the limitations of SCCNNs. This analysis in turn guides the design of convolutional architectures.

In Section 6, we validate our theoretical findings and highlight the effect of the three principles, the need for the Hodge-aware learning, as well as the stability, based on different simplicial tasks including recovering

foreign currency exchange (forex) rates, predicting triadic and tetradic collaborations, and ocean current trajectories. Finally, we conclude the paper in Section 7 with a discussion on this work and its relations to existing works.

2 Background

We first review simplicial complexes and data supported on simplices, which are natural generalizations of the corresponding notions on graphs. Then, we introduce discrete calculus on simplicial complexes, which is linked to the incidence matrices. Finally, we discuss the Hodge decomposition, which uniquely characterizes the simplicial signals from three subspaces.

2.1 Simplicial complex and simplicial signals

Given a set $\mathcal{V} = \{1, \dots, n_0\}$ of vertices, a k-simplex s^k is a subset of $\mathcal{V}$ with cardinality $k+1$. Geometrically, a node is a 0-simplex, an edge connecting two vertices is a 1-simplex and a triangular face (we shorten it as a triangle) is a 2-simplex. A subset, with cardinality k, of s^k is a *face*. A *coface* of s^k is a $(k+1)$-simplex that has s^k as a face. Furthermore, one can collect k-simplices for $k = 0, \dots, K$ to form a simplicial complex (SC) $\mathcal{S}$ of order K with the *inclusion* restriction that if a simplex is in the SC, so are its subsets. A graph is an SC of order one and by including some triangles, we obtain an SC of order two, as shown in Fig. 1a. We denote the set of all k-simplices in $\mathcal{S}$ as $\mathcal{S}^k = \{s_i^k\}_{i=1,\dots,n_k}$ where $n_k = |\mathcal{S}^k|$, i.e., $\mathcal{S} = \cup_{k=0}^K \mathcal{S}^k$.

Simplicial adjacency For any two k-simplices, we say they are *lower (upper) adjacent* if they share a common face (coface), which natualy defines the notion of simplicial neighborhoods. For example, two nodes are (upper) adjacent in a graph if they are connected by an edge. In Fig. 1a, edges e_1 and e_3 are lower neighbors as they share node 1; while e_1 and e_2 are upper neighbors since they are located in the triangle t_1.

Orientation For computational purposes, we annotate each simplex with an *orientation*, as an ordering of the labels of its vertices (a node has a trivial orientation). Here we consider the increasing ordering as the reference orientation, that is, a triangle $s^2 = \{i, j, k\}$ is oriented as $[i, j, k]$ for $i < j < k$, and an edge $s^1 = \{i, j\}$ is oriented as $[i, j]$ for $i < j$.

Algebraic representation We use the incidence matrix $\boldsymbol{B}_k \in \mathbb{R}^{n_{k-1} \times n_k}$ to describe the relationships between $(k-1)$- and k-simplices. Thus, $\boldsymbol{B}_1$ encodes the node-to-edge incidence and $\boldsymbol{B}_2$ the edge-to-triangle incidence. In an oriented SC $\mathcal{S}$, the entries of $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ are given by

$$[\boldsymbol{B}_1]_{ie} = \begin{cases} -1, & \text{for } e = [i, \cdot] \\ 1, & \text{for } e = [\cdot, i] \\ 0, & \text{otherwise.} \end{cases} \quad [\boldsymbol{B}_2]_{et} = \begin{cases} 1, & \text{for } e = [i, j], t = [i, j, k] \\ -1, & \text{for } e = [i, k], t = [i, j, k] \\ 1, & \text{for } e = [j, k], t = [i, j, k] \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

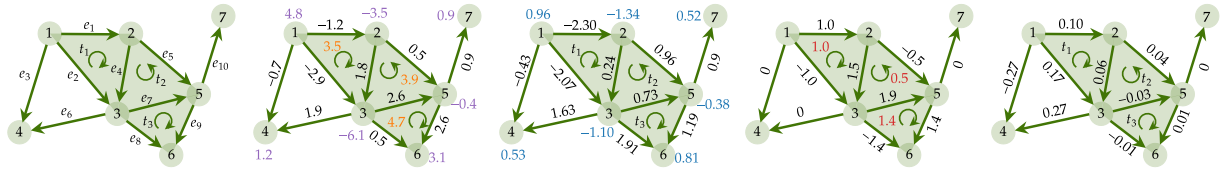
We further define the *k-Hodge Laplacian*

$$\boldsymbol{L}_k = \boldsymbol{B}_k^\top \boldsymbol{B}_k + \boldsymbol{B}_{k+1} \boldsymbol{B}_{k+1}^\top \tag{2}$$

with the *lower Laplacian* $\boldsymbol{L}_{k,\mathrm{d}} = \boldsymbol{B}_k^\top \boldsymbol{B}_k$ and the *upper Laplacian* $\boldsymbol{L}_{k,\mathrm{u}} = \boldsymbol{B}_{k+1} \boldsymbol{B}_{k+1}^\top$. We have a set of $\boldsymbol{L}_k, k = 1, \dots, K-1$ in an SC of order K with $\boldsymbol{L}_0 = \boldsymbol{B}_1 \boldsymbol{B}_1^\top$ the graph Laplacian, and $\boldsymbol{L}_K = \boldsymbol{B}_K^\top \boldsymbol{B}_K$. Topologically, $\boldsymbol{L}_{k,\mathrm{d}}$ and $\boldsymbol{L}_{k,\mathrm{u}}$ encode the lower and upper adjacencies of k-simplices, respectively. For example, $\boldsymbol{L}_{1,\mathrm{d}}$ encodes the edge-to-edge adjacencies through nodes while $\boldsymbol{L}_{1,\mathrm{u}}$ encodes the adjacencies through triangles.

2.2 Simplicial signals and Hodge decomposition

Simplicial signals A *k-simplicial signal* (or data) $\boldsymbol{x}_k \in \mathbb{R}^{n_k}$ supported on the simplicial set $\mathcal{S}^k$ is defined by an *alternating* map $f_k : \mathcal{S}^k \to \mathbb{R}^{n_k}$, which assigns a real value to a simplex, with the condition that if the orientation of a simplex is anti-aligned with the reference orientation, then the signal will change the sign



(a) SC (b) Edge flow (c) Gradient flow (d) Curl flow (e) Harmonic flow

Figure 1: (a) A simplicial 2-complex where green shaded triangles denote 2-simplices and the arrows denote the chosen reference orientations. (b) An edge flow where we denote its divergence and curl in purple and orange, respectively. (c)-(d) The Hodge decomposition of the edge flow in (b). The gradient flow is the gradient of some node signal (in blue) and is curl-free. The curl flow can be obtained from some triangle flow (in red), and is divergence-free. The harmonic flow has zero divergence and zero curl, and is circulating around the hole $\{1, 3, 4\}$. Note that in this figure, the flow numbers are rounded up to two decimal places. Thus, at some nodes or triangles with zero-divergence or zero-curl, the divergence or curl might not be exactly zero.

(Lim, 2020). A d-dimensional simplicial feature $\boldsymbol{X}_k \in \mathbb{R}^{n_k \times d}$ can be defined for a rich representation learning of simplices. For simplicity, we restrict our analysis to $d = 1$.

Incidence matrices as derivatives on SCs Given a simplicial signal $\boldsymbol{x}_k$, we can measure its variability with respect to the faces and cofaces of k-simplices by computing $\boldsymbol{B}_k\boldsymbol{x}_k$ and $\boldsymbol{B}_{k+1}^\top\boldsymbol{x}_k$ (Grady & Polimeni, 2010). Specifically, $\boldsymbol{B}_1^\top\boldsymbol{x}_0$ computes the *gradient* of a node signal $\boldsymbol{x}_0$ as the signal difference between the adjacent nodes, i.e., $[\boldsymbol{B}_1^\top\boldsymbol{x}_0]_{[i,j]} = [\boldsymbol{x}_0]_j - [\boldsymbol{x}_0]_i$, which is often used in the GNN literature. For an edge flow $\boldsymbol{x}_1$, $\boldsymbol{B}_1\boldsymbol{x}_1$ computes its *divergence*, which is the difference between the total in-flow and out-flow at node j, i.e., $[\boldsymbol{B}_1\boldsymbol{x}_1]_j = \sum_{i<j}[\boldsymbol{x}_1]_{[i,j]} - \sum_{j<k}[\boldsymbol{x}_1]_{[j,k]}$. Moreover, $\boldsymbol{B}_2^\top\boldsymbol{x}_1$ computes the *curl* of $\boldsymbol{x}_1$, i.e., $[\boldsymbol{B}_2^\top\boldsymbol{x}_1]_t = [\boldsymbol{x}_1]_{[i,j]} - [\boldsymbol{x}_1]_{[i,k]} + [\boldsymbol{x}_1]_{[j,k]}$, which is the net-flow circulation in triangle $t = [i, j, k]$. As illustrated in Fig. 1b, these two computations provide divergent and rotational variation measures of an edge flow, which are analogous to the notions of divergence and curl for vector fields in continuous domains. In the following, we introduce the Hodge decomposition (Hodge, 1989; Lim, 2020) which unfolds an edge flow into three unique characteristic components.

Lemma 1 ((Lim, 2020)). *We have $\boldsymbol{B}_2^\top\boldsymbol{B}_1^\top = \boldsymbol{0}$, i.e., the curl of the gradient is zero.*

Theorem 2 (Hodge decomposition). *The k-simplicial signal space $\mathbb{R}^{n_k}$ admits a direct sum decomposition*

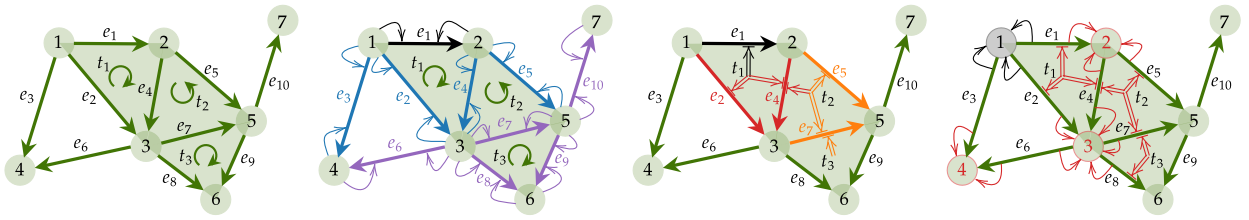
$$\mathbb{R}^{n_k} = \mathrm{im}(\boldsymbol{B}_k^\top) \oplus \ker(\boldsymbol{L}_k) \oplus \mathrm{im}(\boldsymbol{B}_{k+1}), \;\; \textit{thus, } \boldsymbol{x}_k = \boldsymbol{x}_{k,\mathrm{G}} + \boldsymbol{x}_{k,\mathrm{H}} + \boldsymbol{x}_{k,\mathrm{C}}, \tag{3}$$

where $\boldsymbol{x}_{k,\mathrm{G}} = \boldsymbol{B}_k^\top\boldsymbol{x}_{k-1}$ for some $\boldsymbol{x}_{k-1}$, and $\boldsymbol{x}_{k,\mathrm{C}} = \boldsymbol{B}_{k+1}\boldsymbol{x}_{k+1}$ for some $\boldsymbol{x}_{k+1}$. Moreover, we have $\ker(\boldsymbol{B}_{k+1}^\top) = \mathrm{im}(\boldsymbol{B}_k^\top) \oplus \ker(\boldsymbol{L}_k)$ and $\ker(\boldsymbol{B}_k) = \ker(\boldsymbol{L}_k) \oplus \mathrm{im}(\boldsymbol{B}_{k+1})$.

In the node space, this decomposition is trivial as $\mathbb{R}^{n_0} = \ker(\boldsymbol{L}_0) \oplus \mathrm{im}(\boldsymbol{B}_1)$ where the kernel of $\boldsymbol{L}_0$ contains constant node data and the image of $\boldsymbol{B}_1$ contains nonconstant data. In the edge case, the three subspaces carry more tangible meaning: the *gradient space* $\mathrm{im}(\boldsymbol{B}_1^\top)$ collects edge flows as the gradient of some node signal, which are *curl-free*; the *curl* space $\mathrm{im}(\boldsymbol{B}_2)$ consists of flows cycling around triangles, which are *div-free*; and flows in the *harmonic space* $\ker(\boldsymbol{L}_1)$ are both div- and curl-free. In this paper, we inherit the names of three edge subspaces to general k-simplices. The above theorem states that any simplicial signal $\boldsymbol{x}_k$ can be uniquely expressed as $\boldsymbol{x}_k = \boldsymbol{x}_{k,\mathrm{G}} + \boldsymbol{x}_{k,\mathrm{H}} + \boldsymbol{x}_{k,\mathrm{C}}$ with the gradient part $\boldsymbol{x}_{k,\mathrm{G}} = \boldsymbol{B}_k^\top\boldsymbol{x}_{k-1}$, the curl part $\boldsymbol{x}_{k,\mathrm{C}} = \boldsymbol{B}_{k+1}\boldsymbol{x}_{k+1}$, for some $\boldsymbol{x}_{k\pm1}$, and the harmonic part following $\boldsymbol{L}_k\boldsymbol{x}_{k,\mathrm{H}} = \boldsymbol{0}$. Figs. 1c to 1e provide the three Hodge components of the edge flow in Fig. 1b.

3 Simplicial Complex CNNs

We first introduce the general convolutional architecture on SCs, then discuss the propreties of SCCNN and study the effects of the three principles from an energy minimization perspective.



(a) SC example (b) Lower edge conv. (c) Upper edge conv. (d) Inter-simplicial locality

Figure 2: (a) An SC where arrows indicate the reference orientations of edges and triangles. 2-simplices are (filled) triangles shaded in green and open triangle $\{1,3,4\}$ is not in the SC. (b) Lower convolution via $\boldsymbol{H}_1$ and $\boldsymbol{H}_{1,\mathrm{d}}$ on edge e_1. (c) Upper convolution via $\boldsymbol{H}_1$ and $\boldsymbol{H}_{1,\mathrm{u}}$ on e_1. (d) Node 1 (in black) contains information from its neighbors $\{2,3,4\}$ (nodes in red), and projected information from edges which contribute to these neighbors (denoted by arrows in red from edges to nodes), and from triangles $\{t_1,t_2,t_3\}$ which contribute to those edges (denoted by double arrows in red from triangle centers to edges). This interaction is the coupling between the intra- and the extended inter-simplicial locality.

In an SC, taking $\boldsymbol{x}_{k-1}^{l-1}, \boldsymbol{x}_k^{l-1}$ and $\boldsymbol{x}_{k+1}^{l-1}$ as inputs, an SCCNN at layer $l=1,\ldots,L$ computes the k-simplicial output $\boldsymbol{x}_k^l$ via a map

$$\mathrm{SCCNN}_k^l : \{\boldsymbol{x}_{k-1}^{l-1}, \boldsymbol{x}_k^{l-1}, \boldsymbol{x}_{k+1}^{l-1}\} \to \boldsymbol{x}_k^l, \quad \boldsymbol{x}_k^l = \sigma(\boldsymbol{H}_{k,\mathrm{d}}^l \boldsymbol{x}_{k,\mathrm{d}}^{l-1} + \boldsymbol{H}_k^l \boldsymbol{x}_k^{l-1} + \boldsymbol{H}_{k,\mathrm{u}}^l \boldsymbol{x}_{k,\mathrm{u}}^{l-1}) \tag{4}$$

where

$$\boldsymbol{H}_k = \sum_{t=0}^{T_\mathrm{d}} w_{k,\mathrm{d},t} \boldsymbol{L}_{k,\mathrm{d}}^t + \sum_{t=0}^{T_\mathrm{u}} w_{k,\mathrm{u},t} \boldsymbol{L}_{k,\mathrm{u}}^t \tag{5}$$

is a *simplicial convolutional filter* (SCF, (Yang et al., 2022b)) with learnable coefficients $\{w_{k,\mathrm{d},t}\}, \{w_{k,\mathrm{u},t}\}$, and $\boldsymbol{H}_{k,\mathrm{d}} = \sum_{t=0}^{T_\mathrm{d}} w'_{k,\mathrm{d},t} \boldsymbol{L}_{k,\mathrm{d}}^t$ and $\boldsymbol{H}_{k,\mathrm{u}} = \sum_{t=0}^{T_\mathrm{u}} w'_{k,\mathrm{u},t} \boldsymbol{L}_{k,\mathrm{u}}^t$ are the lower and upper SCFs, respectively. Moreover, $\boldsymbol{x}_{k,\mathrm{d}}^{l-1} = \boldsymbol{B}_k^\top \boldsymbol{x}_{k-1}^{l-1}$ and $\boldsymbol{x}_{k,\mathrm{u}}^{l-1} = \boldsymbol{B}_{k+1} \boldsymbol{x}_{k+1}^{l-1}$ are the lower and upper projections from $(k\pm1)$-simplices via incidence relations to k-simplices, respectively, and $\sigma(\cdot)$ is an elementwise nonlinearity. The convolution operations in this SCCNN can be understood as follows: 1) The previous k-simplicial output $\boldsymbol{x}_k^{l-1}$ is passed to an SCF $\boldsymbol{H}_k^l$ of orders $T_\mathrm{d}, T_\mathrm{u}$, which performs a linear combination of the signals from the lower-adjacent (up to T_d-hop) and upper-adjacent (up to T_u-hop) simplices. 2) The previous $k\pm1$-simplicial outputs $\boldsymbol{x}_{k\pm1}^{l-1}$ are first projected to k-simplices, which are then convolved using a lower SCF and an upper SCF, respectively.

Example 3. In Fig. 2, we provide an example of SCCNN for the edge case $k=1$. We focus on edge e_1 and consider the cases $T_\mathrm{d}=T_\mathrm{u}=2$. On edge e_1, the SCF $\boldsymbol{H}_1$ linearly combines the signals from its direct lower neighbors (edges in blue) and two-hop lower neighbors (edges in purple), as shown in Fig. 2b. It also combines the signals from the direct upper neighbors (edges in red) and two-hop upper neighbors (edges in orange), as shown in Fig. 2c. At the same time, the signals on nodes are projected on the edges, denoted by arrows in blue and purple from nodes to edges in Fig. 2b, which are then combined to edge e_1 by the lower SCF $\boldsymbol{H}_{1,\mathrm{d}}$. The signals on triangles are projected on the edges as well, denoted by double arrows in red and orange in Fig. 2c, which are combined to edge e_1 by the upper SCF $\boldsymbol{H}_{1,\mathrm{u}}$.

This architecture subsumes the convolutional learning methods on SCs in Bunch et al. (2020); Ebli et al. (2020); Roddenberry et al. (2021); Yang et al. (2022a); Chen et al. (2022b); Yang et al. (2022c). We refer to Appendix C for a detailed discussion. Particularly, we here emphasize on the key three principles of an SCCNN layer:

(P1) It uncouples the lower and upper parts in the Hodge Laplacian. This leads to an independent treatment of the lower and upper adjacencies, achieved by using two sets of learnable weights. We shall see in Section 4 that how this relates to the independent and expressive learning in the Hodge subspaces given in Theorem 2.

(P2) It accounts for the inter-simplicial couplings via the incidence relations. The projections $\boldsymbol{x}_{k,\mathrm{d}}$ and $\boldsymbol{x}_{k,\mathrm{u}}$ carry nontrivial information contained in the faces and cofaces of simplices (by Theorem 2).

(P3) It performs higher-order convolutions. We consider $T_{\mathrm{d}}, T_{\mathrm{u}} \geq 1$ in SCFs which leads to a multi-hop receptive field on SCs.

In short, each SCCNN layer propagates information across SCs based on two simplicial adjacencies and two incidences in a multi-hop fashion.

3.1 Properties

Simplicial locality The simplicial convolutions admit an intra-simplicial locality where the output $\boldsymbol{H}_k \boldsymbol{x}_k$ is localized in T_{d}-hop lower and T_{u}-hop upper k-simplicial neighborhoods (Yang et al., 2022b). An SCCNN preserves such property as $\sigma(\cdot)$ does not alter the information locality. It also admits an inter-simplicial locality between k- and $(k \pm 1)$-simplices due to the inter-simplicial couplings. This further extends to simplices of orders $k \pm l$ if $L \geq l$ because $\boldsymbol{B}_k \sigma(\boldsymbol{B}_{k+1}) \neq \boldsymbol{0}$ (Schaub et al., 2021). Moreover, the intra- and inter-simplicial localities are coupled in a multi-hop way through higher-order convolutions such that, for example, a node not only interacts with its incident edges and the triangles including it, but also with those further hops away, as shown in Fig. 2d. We refer to Appendix B.1 for a more formal discussion.

Complexity An SCCNN layer has a parameter complexity of order $\mathcal{O}(T_{\mathrm{d}} + T_{\mathrm{u}})$ and a computational complexity $\mathcal{O}(k(n_k + n_{k+1}) + n_k m_k (T_{\mathrm{d}} + T_{\mathrm{u}}))$, which are linear in the simplex dimensions. Here, m_k is the maximum number of neighbors for k-simplices. We refer to Appendix B.2 for more details.

Equivariance SCCNNs are permutation-equivairant, which allows us to list simplices in any order. They are also orientation-equivariant if the activation function $\sigma(\cdot)$ is odd, which gives us the freedom to choose reference orientations. In Appendix B.3, we provide a more formal discussion on such equivariances and why permutations form a symmetry group of an SC and orientation changes are symmetries of data spaces but not SCs.

3.2 A simplicial Dirichlet energy perspective

Here we analyze the convolution architecture in Eq. (4) from an energy minimization perspective. First, we extend the notion of Dirichlet energy from graphs to SCs.

Definition 4. The *Dirichlet energy* of a k-simplicial signal $\boldsymbol{x}_k$ is

$$D(\boldsymbol{x}_k) = D_{\mathrm{d}}(\boldsymbol{x}_k) + D_{\mathrm{u}}(\boldsymbol{x}_k) := \|\boldsymbol{B}_k \boldsymbol{x}_k\|_2^2 + \|\boldsymbol{B}_{k+1}^\top \boldsymbol{x}_k\|_2^2. \tag{6}$$

This Dirichlet energy returns the graph Dirichlet energy when $k = 0$. In this case, $D(\boldsymbol{x}_0) = \|\boldsymbol{B}_1^\top \boldsymbol{x}_0\|_2^2 = \sum_i \sum_j \|x_{0,i} - x_{0,j}\|^2$ is the ℓ_2-norm of the *gradient* of the node signal $\boldsymbol{x}_0$. For edge flow $\boldsymbol{x}_1$, $D(\boldsymbol{x}_1)$ consists of two parts, $\|\boldsymbol{B}_1 \boldsymbol{x}_1\|_2^2$ and $\|\boldsymbol{B}_2^\top \boldsymbol{x}_1\|_2^2$, which measure the total divergence and curl of $\boldsymbol{x}_1$, respectively, i.e., the edge flow variations w.r.t. nodes and triangles. In the general case, $D_{\mathrm{d}}(\boldsymbol{x}_k)$ and $D_{\mathrm{u}}(\boldsymbol{x}_k)$ measure the lower and upper k-simplicial signal variations w.r.t. the faces and cofaces, respectively. A harmonic k-simplicial signal $\boldsymbol{x}_k$ has zero Dirichlet energy, e.g., a constant node signal and a div- and curl-free edge flow.

Simplicial shifting as Hodge Laplacian smoothing Bunch et al. (2020); Yang et al. (2022c) considered $\boldsymbol{H}_k$ to be a weighted variant of $\boldsymbol{I} - \boldsymbol{L}_k$, generalizing the graph convolutional network (GCN) (Kipf & Welling, 2017). This is necessarily a Hodge Laplacian smoothing as in Schaub et al. (2021)—given an initial signal $\boldsymbol{x}_k^0$, we consider the Dirichlet energy minimization:

$$\begin{aligned} &\min_{\boldsymbol{x}_k} \|\boldsymbol{B}_k \boldsymbol{x}_k\|_2^2 + \gamma \|\boldsymbol{B}_{k+1}^\top \boldsymbol{x}_k\|_2^2, \ \gamma > 0, \\ &\text{gradient descent: } \boldsymbol{x}_{k,\mathrm{gd}}^{l+1} = (\boldsymbol{I} - \eta \boldsymbol{L}_{k,\mathrm{d}} - \eta\gamma \boldsymbol{L}_{k,\mathrm{u}}) \boldsymbol{x}_k^l \end{aligned} \tag{7}$$

with step size $\eta > 0$. The simplicial shifting $\boldsymbol{x}_k^{l+1} = w_0(\boldsymbol{I} - \boldsymbol{L}_k)\boldsymbol{x}_k^l$ is a gradient descent step with $\eta = \gamma = 1$ and weighted by w_0. A minimizer of Eq. (7) with $\gamma = 1$ is in fact in the harmonic space $\ker(\boldsymbol{L}_k)$. Thus, a neural network composed of simplicial shifting layers may generate an output with an exponentially decreasing

Dirichlet energy as it deepens, formulated by the following proposition. We refer to this as *simplicial oversmoothing*, a notion that generalizes the oversmoothing of a GCN and its variants (Nt & Maehara, 2019; Cai & Wang, 2020; Rusch et al., 2023).

Proposition 5. *If* $w_0^2\|\boldsymbol{I}-\boldsymbol{L}_k\|_2^2<1$, $D(\boldsymbol{x}_k^{l+1})$ *in a neural network of simplicial shifting layers exponentially converges to zero.*

However, when uncoupling the lower and upper parts of $\boldsymbol{L}_k$ in this shifting, associated to the case $\gamma\neq 1$, the decrease of $D(\boldsymbol{x}_k)$ can slow down or cease because the objective function in Eq. (7) instead looks for a solution primarily in either $\ker(\boldsymbol{B}_k)$ (for $\gamma\ll 1$) or $\ker(\boldsymbol{B}_{k+1}^\top)$ (for $\gamma\gg 1$), not necessarily in $\ker(\boldsymbol{L}_k)$, as we shall corroborate in Section 6.

Inter-simplicial couplings as sources Given some nontrivial $\boldsymbol{x}_{k-1}$ and $\boldsymbol{x}_{k+1}$, we consider the optimization

$$\begin{gathered}\min_{\boldsymbol{x}_k}\|\boldsymbol{B}_k\boldsymbol{x}_k-\boldsymbol{x}_{k-1}\|_2^2+\|\boldsymbol{B}_{k+1}^\top\boldsymbol{x}_k-\boldsymbol{x}_{k+1}\|_2^2,\\ \text{gradient descent: } \boldsymbol{x}_{k,\mathrm{gd}}^{l+1}=(\boldsymbol{I}-\eta\boldsymbol{L}_k)\boldsymbol{x}_k^l+\eta(\boldsymbol{x}_{k,\mathrm{d}}+\boldsymbol{x}_{k,\mathrm{u}})\end{gathered}\tag{8}$$

with step size $\eta>0$. It resembles the convolutional layer, $\boldsymbol{x}_k^{l+1}=w_0(\boldsymbol{I}-\boldsymbol{L}_k)\boldsymbol{x}_k^l+w_1\boldsymbol{x}_{k,\mathrm{d}}+w_2\boldsymbol{x}_{k,\mathrm{u}}$ with some learnable weights, in Bunch et al. (2020); Yang et al. (2022c).

Proposition 6. *We have* $D(\boldsymbol{x}_k^{l+1})\leq w_0^2\|\boldsymbol{I}-\boldsymbol{L}_k\|_2^2D(\boldsymbol{x}_k^l)+w_1^2\lambda_{\max}(\boldsymbol{L}_{k,\mathrm{d}})\|\boldsymbol{x}_{k,\mathrm{d}}\|_2^2+w_2^2\lambda_{\max}(\boldsymbol{L}_{k,\mathrm{u}})\|\boldsymbol{x}_{k,\mathrm{u}}\|_2^2$.

The signal projections from the lower and upper simplices act as energy sources for $\boldsymbol{x}_k^l$, and also the objective function in Eq. (8) looks for an $\boldsymbol{x}_k$ in the image spaces of $\boldsymbol{B}_{k+1}$ and $\boldsymbol{B}_k^\top$, instead of $\ker(\boldsymbol{L}_k)$. Thus, inter-simplicial couplings pay a role in mitigating the oversmoothing.

Here we showed that simply generalzing GCNs to simplices will inherit its oversmoothing risks. However, by uncoupling the lower and upper Laplacians and accounting for the inter-simplicial couplings we could mitigate this issue. This can also be explained by means of a diffusion process on SCs (Ziegler et al., 2022), which we discuss in Appendix B.4.

4 From convolutional to Hodge-aware

In this section, we first introduce the *Hodge-invariant operator*, which is an operator such that the three Hodge subspaces are invariant under it. Then, we show that the SCF is such an operator and SCCNN, guided by the three principles (P1-P3), performs *Hodge-invariant learning*, allowing for rational and effective learning on SCs while remaining expressive. Throughout the exposition, we rely on the simplicial spectral theory (Barbarossa & Sardellitti, 2020; Yang et al., 2021; 2022b), which also allows us to characterize the expressive power of SCCNNs. We refer to the detailed derivations and proofs in Appendix E.

Definition 7 (Invariant subspace)**.** Let V be a finite-dimensional vector space over $\mathbb{R}$ with $\dim(V)\geq 1$, and let $T:V\to V$ be an operator in V. A subspace $U\subset V$ is an *invariant subspace* under T if $Tu\in U$ for all $u\in U$, i.e., the image of every vector in U under T remains within U. We denote this as $T|_U:U\to U$ where $T|_U$ is the restriction of T on U.

Given the notion of invariant subspace, we then define the Hodge-invariant operators.

Definition 8 (Hodge-invariant operator)**.** Let $\square\in\{\mathrm{im}(\boldsymbol{B}_k^\top),\mathrm{im}(\boldsymbol{B}_{k+1}),\ker(\boldsymbol{L}_k)\}$ be any Hodge subspace of $\mathbb{R}^{n_k}$. A linear transformation $F:\mathbb{R}^{n_k}\to\mathbb{R}^{n_k}$ is a *Hodge-invariant operator* if for all $\boldsymbol{x}_k\in\square$ it holds that $F(\boldsymbol{x}_k)\in\square$. That is, any simplicial signal in a certain Hodge subspace remains in that subspace under F.

Definition 9 ((Barbarossa & Sardellitti, 2020))**.** The *simplicial Fourier transform* (SFT) of $\boldsymbol{x}_k$ is $\tilde{\boldsymbol{x}}_k=\boldsymbol{U}_k^\top\boldsymbol{x}_k$ where the eigenbasis $\boldsymbol{U}_k$ of $\boldsymbol{L}_k$ acts as the simplicial Fourier basis and the eigenvalues in $\boldsymbol{\Lambda}_k=\mathrm{diag}(\boldsymbol{\lambda}_k)$ are *simplicial frequencies*.

Proposition 10 (Yang et al. (2022b))**.** *The SFT basis can be found as* $\boldsymbol{U}_k=[\boldsymbol{U}_{k,\mathrm{H}}\ \boldsymbol{U}_{k,\mathrm{G}}\ \boldsymbol{U}_{k,\mathrm{C}}]$ *where*

- $\boldsymbol{U}_{k,\mathrm{H}}$ *is the eigenvector matrix associated to the zero eigenvalues* $\boldsymbol{\Lambda}_{k,\mathrm{H}}=\mathrm{diag}(\boldsymbol{\lambda}_{k,\mathrm{H}})$, *named as* harmonic frequencies,

- $\boldsymbol{U}_{k,\mathrm{G}}$ *is associated to the nonzero eigenvalues in* $\boldsymbol{\Lambda}_{k,\mathrm{G}} = \mathrm{diag}(\boldsymbol{\lambda}_{k,\mathrm{G}})$ *of* $\boldsymbol{L}_{k,\mathrm{d}}$*, named as* gradient frequencies*, and*
- $\boldsymbol{U}_{k,\mathrm{C}}$ *is associated to the nonzero eigenvalues in* $\boldsymbol{\Lambda}_{k,\mathrm{C}} = \mathrm{diag}(\boldsymbol{\lambda}_{k,\mathrm{C}})$*, named as* curl frequencies*.*

Moreover, they span the Hodge subspaces:

$$\mathrm{span}(\boldsymbol{U}_{k,\mathrm{H}}) = \ker(\boldsymbol{L}_k),\ \mathrm{span}(\boldsymbol{U}_{k,\mathrm{G}}) = \mathrm{im}(\boldsymbol{B}_k^\top),\ \mathrm{span}(\boldsymbol{U}_{k,\mathrm{C}}) = \mathrm{im}(\boldsymbol{B}_{k+1}) \tag{9}$$

where $\mathrm{span}(\bullet)$ *denotes all possible linear combinations of columns of* $\bullet$*.*

Remark 11. The frequency notion in general carries the physical meaning of signal variations. In the simplicial case, gradient frequencies reflect the degree of lower variations $D_{\mathrm{d}}(\boldsymbol{u}_{k,\mathrm{G}})$ of the associated gradient Fourier basis, and curl frequencies reflect the degree of upper variations $D_{\mathrm{u}}(\boldsymbol{u}_{k,\mathrm{C}})$ of the associated curl basis. Harmonic frequencies (zeros) correspond to the basis having zero lower and upper variations. In the edge case, the gradient and curl frequencies, respectively, correspond to the total divergence and total curl, measuring how divergent and rotational the associated basis is (Yang et al., 2022b).

Proposition 12. *The SCF* $\boldsymbol{H}_k$ *is a Hodge-invariant operator. That is, for any* $\boldsymbol{x}_k \in \square$*, we have* $\boldsymbol{H}_k\boldsymbol{x}_k \in \square$*, for* $\square \in \{\mathrm{im}(\boldsymbol{B}_k^\top), \mathrm{im}(\boldsymbol{B}_{k+1}), \ker(\boldsymbol{L}_k)\}$*. Moreover, the SCF operation can be implicitly written as*

$$\boldsymbol{H}_k\boldsymbol{x}_k = \boldsymbol{H}_k|_{\mathrm{im}(\boldsymbol{B}_k^\top)}\boldsymbol{x}_{k,\mathrm{G}} + \boldsymbol{H}_k|_{\ker(\boldsymbol{L}_k)}\boldsymbol{x}_{k,\mathrm{H}} + \boldsymbol{H}_k|_{\mathrm{im}(\boldsymbol{B}_{k+1})}\boldsymbol{x}_{k,\mathrm{C}} \tag{10}$$

where $\boldsymbol{H}_k|_{\mathrm{im}(\boldsymbol{B}_k^\top)} = \sum_{t=1}^{T_{\mathrm{d}}} w_{k,\mathrm{d},t}\boldsymbol{L}_{k,\mathrm{d}}^t + (w_{k,\mathrm{d},0} + w_{k,\mathrm{u},0})\boldsymbol{I}$ *is the restriction of* $\boldsymbol{H}_k$ *on the gradient space* $im(\boldsymbol{B}_k^\top)$*,* $\boldsymbol{H}_k|_{\ker(\boldsymbol{L}_k)} = (w_{k,\mathrm{d},0} + w_{k,\mathrm{u},0})\boldsymbol{I}$ *is the restriction of* $\boldsymbol{H}_k$ *on the harmonic space, and* $\boldsymbol{H}_k|_{\mathrm{im}(\boldsymbol{B}_{k+1})} = \sum_{t=0}^{T_{\mathrm{u}}} w_{k,\mathrm{u},t}\boldsymbol{L}_{k,\mathrm{u}}^t + (w_{k,\mathrm{d},0} + w_{k,\mathrm{u},0})\boldsymbol{I}$ *is the restriction on the curl space.*

Provided with the Hodge-invariance of $\boldsymbol{H}_k$ and the SFT, we can perform a spectral analysis, which is of interest to further understand the SCCNN since simplicial frequencies reflect the variation characteristics of simplicial signals.

4.1 Spectral analysis

Consider the SFT $\tilde{\boldsymbol{x}}_k = [\tilde{\boldsymbol{x}}_{k,\mathrm{H}}^\top, \tilde{\boldsymbol{x}}_{k,\mathrm{G}}^\top, \tilde{\boldsymbol{x}}_{k,\mathrm{C}}^\top]^\top$ of $\boldsymbol{x}_k$ where each component is the intensity of $\boldsymbol{x}_k$ at a certain simplicial frequency. We can understand how an SCCNN convolutional layer $\boldsymbol{y}_k = \boldsymbol{H}_{k,\mathrm{d}}\boldsymbol{x}_{k,\mathrm{d}} + \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{H}_{k,\mathrm{u}}\boldsymbol{x}_{k,\mathrm{u}}$ regulates/learns from the simplicial signals at different frequencies by performing the SFT

$$\begin{cases} \tilde{\boldsymbol{y}}_{k,\mathrm{H}} = \tilde{\boldsymbol{h}}_{k,\mathrm{H}} \odot \tilde{\boldsymbol{x}}_{k,\mathrm{H}} \\ \tilde{\boldsymbol{y}}_{k,\mathrm{G}} = \tilde{\boldsymbol{h}}_{k,\mathrm{d}} \odot \tilde{\boldsymbol{x}}_{k,\mathrm{d}} + \tilde{\boldsymbol{h}}_{k,\mathrm{G}} \odot \tilde{\boldsymbol{x}}_{k,\mathrm{G}} \\ \tilde{\boldsymbol{y}}_{k,\mathrm{C}} = \tilde{\boldsymbol{h}}_{k,\mathrm{C}} \odot \tilde{\boldsymbol{x}}_{k,\mathrm{C}} + \tilde{\boldsymbol{h}}_{k,\mathrm{u}} \odot \tilde{\boldsymbol{x}}_{k,\mathrm{u}}, \end{cases} \tag{11}$$

with $\odot$ the elementwise multiplication. The n_k-dimensional vector $\tilde{\boldsymbol{h}}_k = \mathrm{diag}(\boldsymbol{U}_k^\top \boldsymbol{H}_k \boldsymbol{U}_k) = [\tilde{\boldsymbol{h}}_{k,\mathrm{H}}^\top\ \tilde{\boldsymbol{h}}_{k,\mathrm{G}}^\top\ \tilde{\boldsymbol{h}}_{k,\mathrm{C}}^\top]^\top$ is the frequency response vector of $\boldsymbol{H}_k$ with

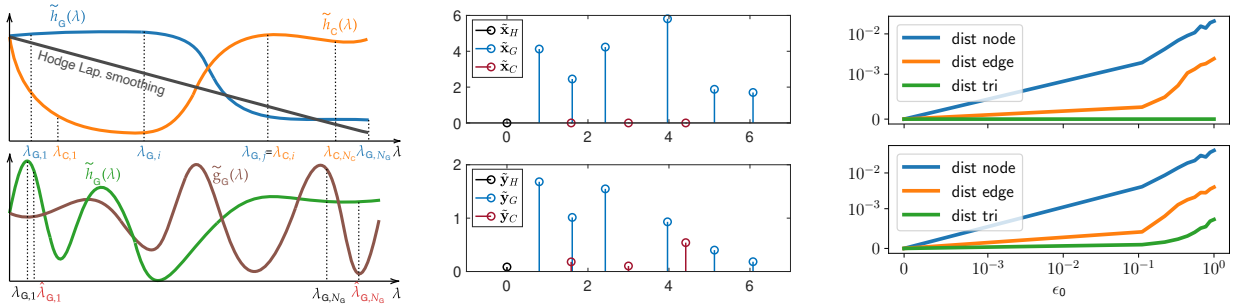
$$\tilde{\boldsymbol{h}}_{k,\mathrm{H}} = (w_{k,\mathrm{d},0} + w_{k,\mathrm{u},0})\boldsymbol{1}, \quad \tilde{\boldsymbol{h}}_{k,\mathrm{G}} = \textstyle\sum_{t=0}^{T_{\mathrm{d}}} w_{k,\mathrm{d},t}\boldsymbol{\lambda}_{k,\mathrm{G}}^{\odot t} + w_{k,\mathrm{u},0}\boldsymbol{1}, \quad \tilde{\boldsymbol{h}}_{k,\mathrm{C}} = \textstyle\sum_{t=0}^{T_{\mathrm{u}}} w_{k,\mathrm{u},t}\boldsymbol{\lambda}_{k,\mathrm{C}}^{\odot t} + w_{k,\mathrm{d},0}\boldsymbol{1}, \tag{12}$$

where $\cdot^{\odot t}$ is the elementwise t-th power of a vector. Likewise,

$$\tilde{\boldsymbol{h}}_{k,\mathrm{d}} = \textstyle\sum_{t=0}^{T_{\mathrm{d}}} w'_{k,\mathrm{d},t}\boldsymbol{\lambda}_{k,\mathrm{G}}^{\odot t} + w'_{k,\mathrm{u},0}\boldsymbol{1}, \text{ and } \tilde{\boldsymbol{h}}_{k,\mathrm{u}} = \textstyle\sum_{t=0}^{T_{\mathrm{u}}} w'_{k,\mathrm{u},t}\boldsymbol{\lambda}_{k,\mathrm{C}}^{\odot t} + w'_{k,\mathrm{d},0}\boldsymbol{1} \tag{13}$$

are the frequency response vectors of $\boldsymbol{H}_{k,\mathrm{d}}$ and $\boldsymbol{H}_{k,\mathrm{u}}$. The spectral relation in Eq. (11) shows that the gradient SFT $\tilde{\boldsymbol{x}}_{k,\mathrm{G}}$ is learned by a gradient response $\tilde{\boldsymbol{h}}_{k,\mathrm{G}}$, while the curl SFT $\tilde{\boldsymbol{x}}_{k,\mathrm{C}}$ is learned by a curl response $\tilde{\boldsymbol{h}}_{k,\mathrm{C}}$. The two learnable responses are independent and they only coincide at the trivial harmonic frequency, as shown by the two individual curves in Fig. 3a. Moreover, the lower and upper projections are independently learned by $\tilde{\boldsymbol{h}}_{k,\mathrm{d}}$ and $\tilde{\boldsymbol{h}}_{k,\mathrm{u}}$, respectively.

The elementwise nonlinearity induces an information spillage that one type of spectra could be spread over other types. That is, $\sigma(\tilde{\boldsymbol{y}}_{k,\mathrm{G}})$ could contain information in harmonic or curl subspaces, as illustrated in Fig. 3b. This is to increase the expressive power of SCCNN, which we characterize as follows.



(a) (b) (c)

Figure 3: (a) *(top)*: Independent gradient and curl learning responses. *(bottom)*: Stability-selectivity tradeoff of SCFs where $\tilde{h}_{\mathrm{G}}$ has better stability but smaller selectivity than $\tilde{g}_{\mathrm{G}}$. (b) Information spillage of nonlinearity. (c) The distance between the perturbed outputs and true when node adjacencies are perturbed. *(top)*: $L = 1$, triangle output remains clean. *(bottom)*: $L = 2$, triangle output is perturbed.

Proposition 13. *An SCCNN layer with inputs $\boldsymbol{x}_{k,\mathrm{d}}, \boldsymbol{x}_k, \boldsymbol{x}_{k,\mathrm{u}}$ is at most expressive as an MLP $\sigma(\boldsymbol{G}'_{k,\mathrm{d}}\boldsymbol{x}_{k,\mathrm{d}} + \boldsymbol{G}_k\boldsymbol{x}_k + \boldsymbol{G}'_{k,\mathrm{u}}\boldsymbol{x}_{k,\mathrm{u}})$ with $\boldsymbol{G}_k = \boldsymbol{G}_{k,\mathrm{d}} + \boldsymbol{G}_{k,\mathrm{u}}$ where $\boldsymbol{G}_{k,\mathrm{d}}$ and $\boldsymbol{G}'_{k,\mathrm{d}}$ are analytical matrix functions of $\boldsymbol{L}_{k,\mathrm{d}}$, while $\boldsymbol{G}_{k,\mathrm{u}}$ and $\boldsymbol{G}'_{k,\mathrm{u}}$ are analytical matrix functions of $\boldsymbol{L}_{k,\mathrm{u}}$. This expressivity can be achieved by setting $T_{\mathrm{d}} = T'_{\mathrm{d}} = n_{k,\mathrm{G}}$ and $T_{\mathrm{u}} = T'_{\mathrm{u}} = n_{k,\mathrm{C}}$ in Eq. (4) with $n_{k,\mathrm{G}}$ the number of distinct gradient frequencies and $n_{k,\mathrm{C}}$ the number of distinct curl frequencies.*

The proof follows from Cayley-Hamilton theorem (Horn & Johnson, 2012). This expressive power can be better understood from the spectral perspective — The gradient SFT $\tilde{\boldsymbol{x}}_{k,\mathrm{G}}$ can be learned as expressive as by an analytical vector-valued function $\tilde{\boldsymbol{g}}_{k,\mathrm{G}}$, which collects the eigenvalues of $\boldsymbol{G}_{k,\mathrm{d}}$ at gradient frequencies. The curl SFT $\tilde{\boldsymbol{x}}_{k,\mathrm{C}}$ can be learned as expressive as by another analytical vector-valued function $\tilde{\boldsymbol{g}}_{k,\mathrm{C}}$, which collects the eigenvalues of $\boldsymbol{G}_{k,\mathrm{u}}$ at curl frequencies. These two functions need only to coincide at the harmonic frequency. In addition, the SFTs of lower and upper projections can be learned as expressive as by two independent analytical vector-valued functions as well.

4.2 Hodge-aware learning

Given the expressive power in Proposition 13 and the spectral relation in Eq. (11), we show that SCCNN performs a Hodge-aware learning in the following sense, which comes with advantages over the existing approaches.

Theorem 14. *An SCCNN is Hodge-aware: 1) The SCF $\boldsymbol{H}_k$ is a Hodge-invariant learning operator. Specifically, three Hodge subspaces are invariant under $\boldsymbol{H}_k$; 2) The lower SCF $\boldsymbol{H}_{k,\mathrm{d}}$ and upper SCF $\boldsymbol{H}_{k,\mathrm{u}}$ are, respectively, gradient- and curl-invariant learning operators; 3) The learnings in the gradient and curl spaces are **independent**; And 4) the learnings in the gradient and curl spaces are **expressive** as in Proposition 13.*

This theorem shows that an SCCNN performs an expressive and independent learning in the gradient and curl subspaces from the three inputs while preserving the three subspaces to be invariant w.r.t its learnable SCFs. This allows for the *rational and effective learning* on SCs, as illustrated in Fig. 4, from the two aspects. These three-fold properties of an SCCNN, respectively, come from the convolutional architecture choice, the uncoupling of the lower and upper adjacencies, and the higher-order convolutions in the SCCNN.

On the one hand, Proposition 12 shows that the operation of $\boldsymbol{H}_k$ on the simplicial signal space is equivalent to a summation of its restrictions $\boldsymbol{H}_k|_{\square}$ on three smaller subspaces $\square$. This Hodge-invariant nature of the learnable SCFs substantially shrinks the learning space of an SCCNN and allows for an effective learning. On the other hand, simplicial signals often present implicit or explicit properties that different Hodge subspaces can capture. For example, water flows, traffic flows, electrical currents (Grady & Polimeni, 2010; Jia et al., 2019) follow flow conservation, i.e., being div-free in the gradient space $\ker(\boldsymbol{B}_1)$, while exchange rates can be modelled as curl-free edge flows (Jiang et al., 2011). Owing to the Hodge-invariance of $\boldsymbol{H}_1$ and its independent

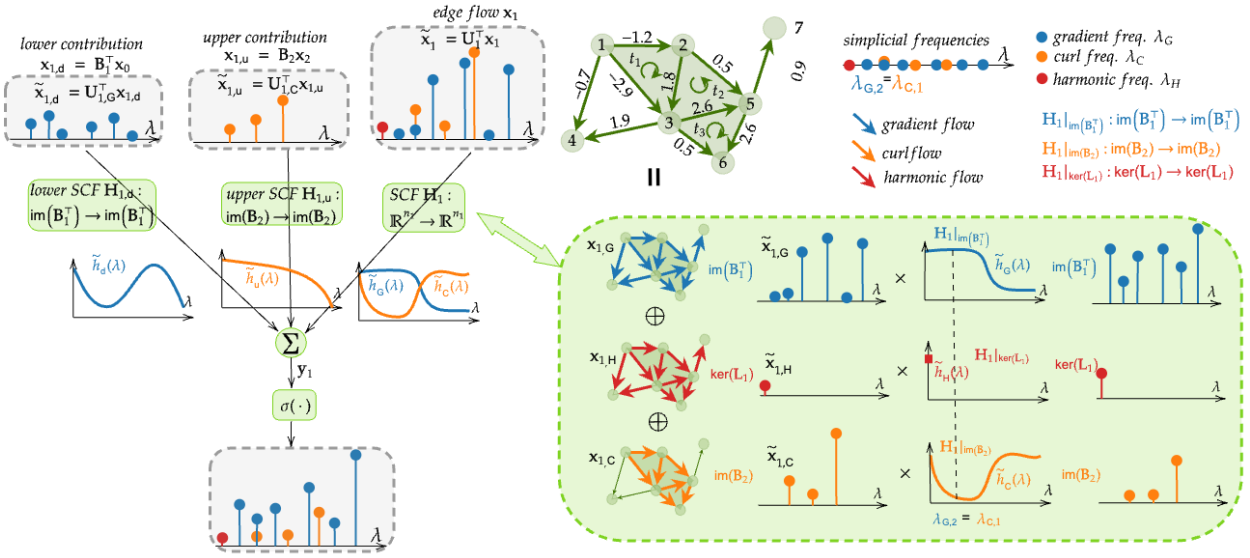


Figure 4: An illustration of the Hodge-aware learning of an SCCNN. We show that how an edge flow $\boldsymbol{x}_1$, together with the lower and upper projections $\boldsymbol{x}_{1,\mathrm{d}}$ and $\boldsymbol{x}_{1,\mathrm{u}}$, are transformed by an SCCNN in the spectral domain. The *implicit* operation $\boldsymbol{H}_1\boldsymbol{x}_1$ (in the dashed box on the right) reflects the Hodge-aware learning: 1) $\boldsymbol{H}_1$ is Hodge invariant: each component is learned within their own subspace, and $\boldsymbol{H}_1$ does not *mix up* the three subspaces; 2) The learning in the gradient and curl subspaces are independent where features at shared frequencies $\lambda_{G,2}$ and $\lambda_{C,1}$ can be separately learned; and 3) The learning operators are expressive in the sense that the spectral responses are as expressive as any analytical functions in the gradient and curl frequencies.

learning in the nontrivial subspaces, an SCCNN can capture such characteristics of real-world edge flows effectively. When it comes to regression tasks on SCs, an SCCNN can generate outputs respecting these physicial laws.

Remark 15 (Relation to message passing networks). Message-passing simplicial networks (MPSNs) (Bodnar et al., 2021b) using MLP to aggregate and update are non-Hodge-aware. Their learning functions pursue direct mappings between the much larger signal space $\mathbb{R}^{n_k}$, thus, requiring more training data for accurate learning, as well as a larger computational complexity. Moreover, MPSN does not preserve the Hodge subspaces, i.e., it is not Hodge-invariant. Thus, they might generate outputs with small losses (e.g., mean-squared-errors) in regression tasks, yet not respecting the physical laws being either div- or curl-free properties such as the above simplicial signals. We shall corroborate this in Appendix G.

Remark 16 (Relation to other convolutional methods). While most convolutional networks on SCs use Hodge-invariant learning operators, they are not strictly Hodge-aware, resulting in practical limits. For example, Ebli et al. (2020) considered $\boldsymbol{H}_k = \sum_i w_i \boldsymbol{L}_k^i$, which preserves the Hodge subspaces yet does not uncouple the lower and upper parts of $\boldsymbol{L}_k$. This makes it *strictly less expressive* and non-Hodge-aware. Consider two frequencies $\lambda_G = \lambda_C$ which share a common value but correspond to the gradient and curl subspaces, respectively. The simplicial signal components at these two frequencies are always learned in the same fashion, which induces contradicting issues when the underlying component in one subspace should be diminished while the one in the other subspace should be preserved. This underlines the importance of uncoupling the two adjacencies because the lower and upper Laplacians operate in different subspaces. Roddenberry et al. (2021) applied $\boldsymbol{H}_k$ with $T_\mathrm{d} = T_\mathrm{u} = 1$. Spatially, this limits the receptive field of each simplex to its direct neighbors. Spectrally, it leads to a linear learning frequency response. A similar treatment was considered in Bunch et al. (2020); Yang et al. (2022c) which simply generalized the GCN without uncoupling the two adjacencies, and gave a limited low-pass linear spectral response, as shown in Fig. 3a and discussed in Section 3.2.

5 How robust are SCCNNs to domain perturbations?

In practice, an SCCNN is often built on a weighted SC to capture the strengths of simplicial adjacencies and incidences. We defer the explicit formulations in Appendix F.1 since it has the same form as Eq. (4) in this case, except for that the Hodge Laplacians are weighted, as well as the incidence matrices. These matrices are often defined following Grady & Polimeni (2010); Horak & Jost (2013); Guglielmi et al. (2023). For example, Bunch et al. (2020); Yang et al. (2022c) considered a particular random walk formulation (Schaub et al., 2020). They can also be learned from data, e.g., via an attention method (Goh et al., 2022; Giusti et al., 2022). For the weighted incidence matrices $\boldsymbol{B}_k^\top, \boldsymbol{B}_{k+1}$, we use operators $\boldsymbol{R}_{k,\mathrm{d}}, \boldsymbol{R}_{k,\mathrm{u}}$ in this section.

To highlight the need for a stability analysis, note that, on the one hand, we may lack the true underlying topologies in SCs as they are often estimated from noisy data; and we may undergo adversarial attacks on the topologies. On the other hand, we want to characterize the stability-selectivity tradeoff of SCCNN, in analogy to the study for CNNs (Bruna & Mallat, 2013; Qiu et al., 2018; Bietti & Mairal, 2017) and GNNs (Gama et al., 2019b; 2020a; Kenlay et al., 2021; Parada-Mayorga et al., 2022).

This motivates us to investigate the stability of SCCNN: *how far are the outputs of an SCCNN before and after perturbations are applied to SCs?* We consider the following relative perturbation model, generalizing the graph perturbation model in Gama et al. (2019b)

Definition 17 (Relative perturbation)**.** Consider some perturbation matrix of an appropriate dimension. For the weighted Hodge Laplacian $\boldsymbol{L}_{k,\mathrm{d}}$, its relative perturbed version is $\widehat{\boldsymbol{L}}_{k,\mathrm{d}} = \boldsymbol{L}_{k,\mathrm{d}} + \boldsymbol{E}_{k,\mathrm{d}}\boldsymbol{L}_{k,\mathrm{d}} + \boldsymbol{L}_{k,\mathrm{d}}\boldsymbol{E}_{k,\mathrm{d}}$ with perturbation $\boldsymbol{E}_{k,\mathrm{d}}$; likewise for $\widehat{\boldsymbol{L}}_{k,\mathrm{u}}$ by $\boldsymbol{E}_{k,\mathrm{u}}$. For the weighted incidence matrix $\boldsymbol{R}_{k,\mathrm{d}}$, its relative perturbed version is $\widehat{\boldsymbol{R}}_{k,\mathrm{d}} = \boldsymbol{R}_{k,\mathrm{d}} + \boldsymbol{J}_{k,\mathrm{d}}\boldsymbol{R}_{k,\mathrm{d}}$ with perturbation $\boldsymbol{J}_{k,\mathrm{d}}$; likewise for $\widehat{\boldsymbol{R}}_{k,\mathrm{u}}$ by $\boldsymbol{J}_{k,\mathrm{u}}$.

This models the *domain perturbations* on the strengths of adjacent and incident relations, e.g., a large weight is applied when two edges are weakly or not adjacent, or data on a node is projected on an edge not incident to it. Moreover, this quantifies the relative perturbations with respect to the local simplicial topology in the sense that weaker connections in an SC are deviated by perturbations proportionally less than stronger connections. We further define the integral Lipschitz property of spectral filters to measure the variability of spectral response functions of $\boldsymbol{H}_k$.

Definition 18 (Intergral Lipschitz SCF)**.** An SCF $\boldsymbol{H}_k$ is *integral Lipschitz* with constants $c_{k,\mathrm{d}}$, $c_{k,\mathrm{u}} \geq 0$ if the derivatives of its spectral response functions $\tilde{h}_{k,\mathrm{G}}(\lambda)$ and $\tilde{h}_{k,\mathrm{C}}(\lambda)$ follow that $|\lambda\tilde{h}'_{k,\mathrm{G}}(\lambda)| \leq c_{k,\mathrm{d}}$ and $|\lambda\tilde{h}'_{k,\mathrm{C}}(\lambda)| \leq c_{k,\mathrm{u}}$.

This property provides a stability-selectivity tradeoff of SCFs independently in the gradient and curl frequencies. A spectral response can have both a good selectivity and stability in small frequencies (a large $|\tilde{h}'_{k,\cdot}|$ for $\lambda \to 0$), while it tends to be flat for having better stability at the cost of selectivity (a small variability for large λ) in large frequencies, as shown in Fig. 3a. As of the polynomial nature of responses, all SCFs of SCCNN are integral Lipschitz. We also denote the integral Lipschitz constant for the lower SCFs $\boldsymbol{H}_{k,\mathrm{d}}$ by $c_{k,\mathrm{d}}$ and for the upper SCFs $\boldsymbol{H}_{k,\mathrm{u}}$ by $c_{k,\mathrm{u}}$ without loss of generality.

Under the following assumptions, we now characterize the stability bound of SCCNN.

Assumption 19. *The perturbations are small such that* $\|\boldsymbol{E}_{k,\mathrm{d}}\|_2 \leq \epsilon_{k,\mathrm{d}}, \|\boldsymbol{J}_{k,\mathrm{d}}\|_2 \leq \varepsilon_{k,\mathrm{d}}, \|\boldsymbol{E}_{k,\mathrm{u}}\|_2 \leq \epsilon_{k,\mathrm{u}}$ *and* $\|\boldsymbol{J}_{k,\mathrm{u}}\|_2 \leq \varepsilon_{k,\mathrm{u}}$*, where* $\|\boldsymbol{A}\|_2 = \max_{|\boldsymbol{x}|_1=1}\|\boldsymbol{A}\boldsymbol{x}\|_2$ *is the operator norm (spectral radius) of a matrix* $\boldsymbol{A}$*.*

Assumption 20. *The SCFs* $\boldsymbol{H}_k$ *of an SCCNN have a normalized bounded frequency response (for simplicity, though unnecessary), likewise for* $\boldsymbol{H}_{k,\mathrm{d}}$ *and* $\boldsymbol{H}_{k,\mathrm{u}}$*.*

Assumption 21. *The lower and upper projections are finite such that* $\|\boldsymbol{R}_{k,\mathrm{d}}\|_2 \leq r_{k,\mathrm{d}}$ *and* $\|\boldsymbol{R}_{k,\mathrm{u}}\|_2 \leq r_{k,\mathrm{u}}$*.*

Assumption 22. *The nonlinearity* $\sigma(\cdot)$*, e.g.,* $\mathrm{relu}, \tanh, \mathrm{sigmoid}$*, is* c_σ*-Lipschitz with* $c_\sigma \geq 0$*.*

Assumption 23. *The initial inputs* $\boldsymbol{x}_k^0$*, for all* k*, are finite, such that* $\|\boldsymbol{x}_k^0\|_2 \leq [\boldsymbol{\beta}]_k$*. We collect them in* $\boldsymbol{\beta} = [\beta_0, \ldots, \beta_K]^\top$*.*

Theorem 24. *Let* $\boldsymbol{x}_k^L$ *be the* k*-simplicial signal output of an* L*-layer SCCNN on a weighted SC. Let* $\hat{\boldsymbol{x}}_k^L$ *be the output of the same SCCNN but on a relatively perturbed SC. Under Assumptions 19 to 23, the Euclidean*

distance between the two outputs is finite and upper-bounded

$$\|\hat{\boldsymbol{x}}_k^L - \boldsymbol{x}_k^L\|_2 \leq [\boldsymbol{d}]_k \text{ with } \boldsymbol{d} = c_\sigma^L \sum_{l=1}^{L} \widehat{\boldsymbol{Z}}^{l-1} \boldsymbol{T} \boldsymbol{Z}^{L-l} \boldsymbol{\beta}, \tag{14}$$

where for $K = 2$,

$$\boldsymbol{T} = \begin{bmatrix} t_0 & t_{0,\mathrm{u}} & \\ t_{1,\mathrm{d}} & t_1 & t_{1,\mathrm{u}} \\ & t_{2,\mathrm{d}} & t_2 \end{bmatrix}, \boldsymbol{Z} = \begin{bmatrix} 1 & r_{0,\mathrm{u}} & \\ r_{1,\mathrm{d}} & 1 & r_{1,\mathrm{u}} \\ & r_{2,\mathrm{d}} & 1 \end{bmatrix}, \widehat{\boldsymbol{Z}} = \begin{bmatrix} 1 & \hat{r}_{0,\mathrm{u}} & \\ \hat{r}_{1,\mathrm{d}} & 1 & \hat{r}_{1,\mathrm{u}} \\ & \hat{r}_{2,\mathrm{d}} & 1 \end{bmatrix}, \tag{15}$$

with $\hat{r}_{k,\mathrm{d}} = r_{k,\mathrm{d}}(1+\varepsilon_{k,\mathrm{d}})$ *and* $\hat{r}_{k,\mathrm{u}} = r_{k,\mathrm{u}}(1+\varepsilon_{k,\mathrm{u}})$. *Notice that* $\boldsymbol{T}, \boldsymbol{Z}$ *and* $\widehat{\boldsymbol{Z}}$ *are tridiagonal and follow a similar structure for a general* K. *The diagonal entries of* $\boldsymbol{T}$ *are* $t_k = c_{k,\mathrm{d}}\Delta_{k,\mathrm{d}}\epsilon_{k,\mathrm{d}} + c_{k,\mathrm{u}}\Delta_{k,\mathrm{u}}\epsilon_{k,\mathrm{u}}$. *The off-diagonal entries are* $t_{k,\mathrm{d}} = r_{k,\mathrm{d}}\varepsilon_{k,\mathrm{d}} + c_{k,\mathrm{d}}\Delta_{k,\mathrm{d}}\epsilon_{k,\mathrm{d}}r_{k,\mathrm{d}}$ *and* $t_{k,\mathrm{u}} = r_{k,\mathrm{u}}\varepsilon_{k,\mathrm{u}} + c_{k,\mathrm{u}}\Delta_{k,\mathrm{u}}\epsilon_{k,\mathrm{u}}r_{k,\mathrm{u}}$, *where* $\Delta_{k,\mathrm{d}}$ *captures the eigenvector misalignment between* $\boldsymbol{L}_{k,\mathrm{d}}$ *and perturbation* $\boldsymbol{E}_{k,\mathrm{d}}$ *with a factor* $n_k^{1/2}$, *and likewise for* $\Delta_{k,\mathrm{u}}$.

We refer to Appendix F.2 for a two-step proof. This result bounds the outputs of an SCCNN on all simplicial levels, showing that they are stable to small perturbations on the simplicial adjacencies and incidences. Specifically, we make two observations from the complicated expression. First, the stability bound depends on i) the degree of perturbations including their magnitudes $\epsilon_{k,\cdot}$ and $\varepsilon_{k,\cdot}$, and the eigenspace misalignment $\Delta_{k,\cdot}$; ii) the integral Lipschitz properties $c_{k,\cdot}$ of SCFs; and, iii) the degree of projections $r_{k,\cdot}$. Second, the stability of the k-output depends on the factors related to not only k-simplices, but also simplices of adjacent orders due to inter-simplicial couplings. For example, when $L = 1$, the node output bound d_0 is affected by factors in the node space, as well as the edge space factored by the projection degree. As the layer deepens, this mutual dependence expands further. When $L = 2$, the factors in the triangle space also affect the stability of the node output d_0, as we observe in Fig. 3c.

More importantly, this stability bound provides practical implications for convolutional learning on SCs. While accounting for inter-simplicial couplings may be beneficial, SCCNN becomes less stable as the number of layers increases due to the mutual dependence between outputs on different simplicial levels. Thus, to maintain the expressive power, higher-order SCFs can be used in exchange for shallow layers. This does not harm the stability.

- First, the high-frequency components can be spread over in the low frequency due to the nonlinearity (i.e., the information spillage in Fig. 3b) where the spectral responses are more selective without losing much the stability. For example, if the simplicial signal has large high gradient frequency components, we need the SCCNN to be integral Lipschitz in order to guarantee the stability. That is, the frequency response should be smooth in high gradient frequencies, as shown by $\tilde{h}_{\mathrm{G}}$ in Fig. 3a (bottom). However, the due to the nonlinearity, the information at high gradient frequencies could spill at lower frequencies, where the spectral responses are more selective and can help discriminate in the subsequent layers.
- Second, higher-order SCFs are easier to be learned with smaller integral Lipschitz constants than lower-order ones due to the increased degree of freedom, thus, leading to an increased stability. This can be easily seen by comparing one-order and two-order cases. We experimentally investigate this in Section 6.4. Moreover, we discuss how to guarantee better intergral Lipschitz properties in Appendix G.4.4 by means of regularizations.

6 Experiments

The goal of this section is to answer the following four research questions with experiments on various simplicial-level regression and classification tasks:

RQ 1 What are the effects of the three principles of SCCNN, i.e., uncoupling the lower and upper parts of Hodge Laplacians (P1), the inter-simplicial couplings (P2), and higher-order convolutions (P3)?

Table 1: Forex results (nmse|total arbitrage, ↓).

Methods	Random Noise	Curl Noise	Interpolation
Input	$0.119_{\pm 0.004}$|$29.19_{\pm 0.874}$	$0.552_{\pm 0.027}$|$122.4_{\pm 5.90}$	$0.717_{\pm .030}$|$106.4_{\pm 0.902}$
Baseline (ℓ_2 regularization)	$0.036_{\pm 0.005}$|$2.29_{\pm 0.079}$	$0.050_{\pm 0.002}$|$11.12_{\pm 0.537}$	$0.534_{\pm 0.043}$|$9.67_{\pm 0.082}$
SNN (Ebli et al., 2020)	$0.110_{\pm 0.005}$|$23.24_{\pm 1.03}$	$0.446_{\pm 0.017}$|$86.95_{\pm 2.20}$	$0.702_{\pm 0.033}$|$104.74_{\pm 1.04}$
PSNN (Roddenberry et al., 2021)	$0.008_{\pm 0.001}$|$0.984_{\pm 0.170}$	$0.000_{\pm 0.000}$|$0.000_{\pm 0.000}$	$0.009_{\pm 0.001}$|$1.13_{\pm 0.329}$
MPSN (Bodnar et al., 2021b)	$0.039_{\pm 0.004}$|$7.74_{\pm 0.88}$	$0.076_{\pm 0.012}$|$14.92_{\pm 2.49}$	$0.117_{\pm 0.063}$|$23.15_{\pm 11.7}$
SCCNN, id	$0.027_{\pm 0.005}$|$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$|$0.000_{\pm 0.000}$	$0.265_{\pm 0.036}$|$0.000_{\pm 0.000}$
SCCNN, tanh	$\mathbf{0.002_{\pm 0.000}}$|$\mathbf{0.325_{\pm 0.082}}$	$0.000_{\pm 0.000}$|$0.003_{\pm 0.003}$	$\mathbf{0.003_{\pm 0.002}}$|$\mathbf{0.279_{\pm 0.151}}$

RQ 2 How do the uncoupling of the lower and upper parts of Hodge Laplacians and the inter-simplicial couplings affect the simplicial oversmoothing?

RQ 3 How does the Hodge-aware property of SCCNN play a role in different tasks on SCs, compared to non-Hodge-aware methods?

RQ 4 How do different factors affect the stability of SCCNN, and how can we maintain the stability while keeping the expressive power?

For comparison, we consider the following learning methods on single-level simplices:

- simplicial neural network (SNN) (Ebli et al., 2020), which does not respect P1 and P2 and is non-Hodge-aware;
- principled simplicial neural network (PSNN) (Roddenberry et al., 2021), which does not respect P2 and P3 and is non-Hodge-aware;
- simplicial convolutional neural networks (SCNN)[1] (Yang et al., 2022a), which does not respect P2 but is Hodge-aware;

and the following learning methods on simplicial complexes:

- Bunch (Bunch et al., 2020), which does not respect P1 and P2 and is non-Hodge-aware;
- MPSN (Bodnar et al., 2021b), which is based on message-passing and not Hodge-aware.

We also considered the MLP and standard GNN (Defferrard et al., 2016) as baselines to highlight the effect of SC topology on simplicial-level tasks. We refer to Appendix C for the detailed comparisons between these methods and SCCNN, as well as to Appendix G for the full experimental details.

6.1 Foreign currency exchange (RQs 1, 3)

In forex problems, to build a fair market, the *arbitray-free* condition implies that for any currencies i, j, k, it follows that $r^{i/j} r^{j/k} = r^{i/k}$ where $r^{i/j}$ is the exchange rate between i and j. That is, the exchange path $i \to j \to k$ provides no profit or loss over a direct exchange $i \to k$. Following Jiang et al. (2011), we model exchange rates as edge flows in an SC of order two, specifically, via $[\boldsymbol{x}_1]_{[i,j]} = \log(r^{i/j})$. This conveniently translates the arbitrage-free condition into $\boldsymbol{x}_1$ being curl-free, i.e., $[\boldsymbol{x}_1]_{[i,j]} + [\boldsymbol{x}_1]_{[j,k]} - [\boldsymbol{x}_1]_{[i,k]} = 0$ in any triangle $[i, j, k]$. We consider a real-world forex market at three timestamps, which contains certain degree of arbitrage (Jia et al., 2019; Yang et al., 2024). We focus on recovering a fair market in two scenarios, first, from noisy exchange rates where random noise and noise only in the curl space modelling random arbitrage ("curl noise") are added, and second, when only 50% of the total rates are observed. To evaluate the performance, we measure the normalized mean squared error (nmse) and total arbitrage (total curl), both equally important for achieving a fair market.

From Table 1, we make the following observations on the impacts of P1 and P3, as well as the Hodge-awareness.

[1]Note that the difference between SCCNN and SCNN lies in that the latter does not include the inter-layer projections, as detailed in Appendix C thus, we refer to our method, simplicial complex CNN.

Table 2: Simplex prediction (AUC, ↑) .

Methods	2-simplex	3-simplex
Mean (Benson et al., 2018)	62.8±2.7	63.6±1.6
MLP	68.5±1.6	69.0±2.2
GNN (Defferrard et al., 2016)	93.9±1.0	96.6±0.5
SNN (Ebli et al., 2020)	92.0±1.8	95.1±1.2
PSNN (Roddenberry et al., 2021)	95.6±1.3	98.1±0.5
SCNN (Yang et al., 2022a)	96.5±1.5	98.3±0.4
Bunch (Bunch et al., 2020)	98.3±0.5	98.5±0.5
MPSN (Bodnar et al., 2021b)	98.1±0.5	99.2±0.3
SCCNN	**98.7±0.5**	**99.4±0.3**

Table 3: Ablation study on SCCNN and the hyperparameters for the best results.

Missing component	2-simplex	Hyper Params.
—	98.7±0.5	$L = 2, T = 2$
Edge-to-Node	93.9±1.0	$L = 5, T = 2$
Node-to-Node	98.7±0.4	$L = 4, T = 2$
Edge-to-Edge	98.5±1.0	$L = 3, T = 2$
Node-to-Edge	98.8±0.3	$L = 4, T = 2$
Node input	98.2±0.5	$L = 2, T = 4$
Edge input	98.1±0.4	$L = 2, T = 3$

1) MPSN performs poorly at this task: although it reduces the nmse, it outputs unfair rates with large arbitrage, against the forex principle, because it is not Hodge-aware and unable to capture the arbitrage-free property with small amount of data (cf. Remark 15).
2) SNN performs poorly as well: as discussed in Remark 16, it restricts the gradient and curl spaces to be always learned in the same fashion and makes it impossible to perform disjoint learning in two subspaces. However, since there are eigenvalues which share a common value but live in different subspaces in this SC, it requires preserving the gradient component while removing the curl one here.
3) PSNN can reconstruct relatively fair forex rates with small nmse. The reconstruction from curl noise is perfect, while in the other two cases, the nmse and arbitrage are three times larger than the proposed SCCNN due to the limited expressivity of linear learning responses.
4) SCCNN performs the best in both reducing the total error and the total arbitrage, ultimately, corroborating the impact of performing Hodge-aware learning.

We notice that with an identity activation function ($\sigma = \mathrm{id}$), the arbitrage-free rule is fully learned by an SCCNN. However, it has relatively large errors in the random noise and interpolation cases due to its limited linear expressive power. With a nonlinearity $\sigma = \tanh$, an SCCNN can tackle these more challenging cases, finding a good compromise between overall errors and data characteristics.

6.2 Simplicial oversmoothing analysis (RQ 2)

We use simplicial shifting layers (i.e., Eq. (7) composed with $\sigma = \tanh$) to illustrate the evolution of Dirichlet energies of the outputs on nodes, edges and triangles in an SC of order two with respect to the number of layers. The corresponding inputs are randomly sampled from a uniform distribution $\mathcal{U}([-5, 5])$. Fig. 5 (the dashed lines) shows that simply generalizing the GCN on SCs as in Bunch method could lead to oversmoothing on simplices of all orders. This aligns with our theoretical results in Section 3.2. However, uncoupling the lower and upper parts of $\boldsymbol{L}_1$ (e.g., by setting $\gamma = 2$ in Eq. (7)) could mitigate the oversmoothing on edges, as shown by the dotted line. Lastly, when we account for the inter-simplicial coupling, as shown by the solid lines (where we applied Eq. (8)), it could almost prevent the oversmoothing, since it provides energy sources. We refer to Appendix G.1 for other results.

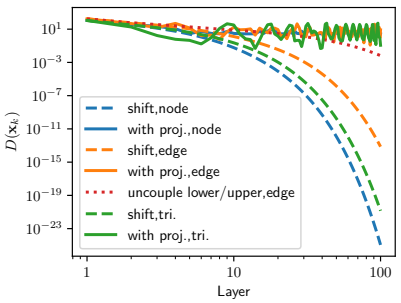


Figure 5: Simplicial oversmoothing.

6.3 Simplex prediction (RQs 1, 3-4)

We consider the prediction task of 2- and 3-simplices which extends the link (1-simplex) prediction in graphs. Our approach is to first learn the representations of lower-order simplices and then use an MLP with their concatenation as inputs to identify if a simplex is closed or open, which generalizes the link prediction method of Zhang & Chen (2018). Considering a coauthorship dataset (Ammar et al., 2018), we built an SC following

Ebli et al. (2020) where nodes represent authors and $(k-1)$-simplices thus represent the collaborations of k-authors. The input simplicial signals are the numbers of citations, e.g., $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are those of dyadic and triadic collaborations. Thus, 2-simplex (3-simplex) prediction amounts to predicting triadic (tetradic) collaborations. We evaluate the AUC (area under the curve) performance.

From Table 2, we make three observations on the effect of the three key principles. 1) SCCNN, MPSN and Bunch methods outperform the ones without inter-simplicial couplings. This highlights that accounting for contributions from faces and cofaces increases the representation power of the network. 2) SCNN performs better than an SNN, which shows that uncoupling the lower and upper parts in $\boldsymbol{L}_k$ improves the representation learning. 3) SCCNN performs better than Bunch (similarly, SCNN better than PSNN), showing that higher-order convolution further improves predictions. 4) While MPSN performs similar to SCCNN, it has three times more parameters than an SCCNN (Appendix G.3.6) under the settings of the best results.

Ablation study We then perform an ablation study to investigate the roles of different components in an SCCNN. As reported in Table 3, we remove certain simplicial relations in the SCCNN and evaluate the prediction performance. Without the edge-to-node incidence, when inputting the node features to the MLP predictor, it is equivalent to a GNN, which has a poor performance. When removing other adjacencies or incidences, the best performance remains similar but with an increased model complexity (more layers required). This however is not preferred, because the stability decreases as the architecture deepens and the model gets influenced by factors in other simplicial spaces, as discussed in Section 5 and shown in Fig. 3c. We also consider the case with limited input where the input on nodes or on edges is missing. The best performance of an SCCNN only slightly drops with an increase of the convolution order.

6.3.1 Stability analysis (RQ 4)

Stability bounds To investigate the stability bound in Eq. (14), we add perturbations to relatively shift the eigenvalues of the Hodge Laplacians and the singular values of the projection matrices by $\epsilon \in [0, 1]$ (cf. Assumption 19). We compare the bound in Eq. (14) to the experimental ℓ_2 distance on each simplex level. As shown in Fig. 6 where the dashed lines are the theoretical stability bounds whereas the solid ones are the experimental stability bounds, we see the bounds become tighter as perturbation increases.

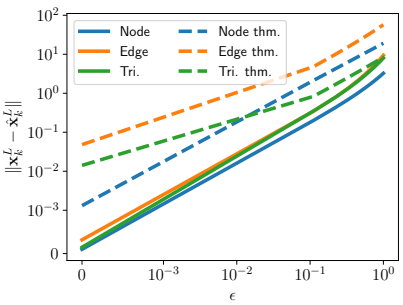


Figure 6: Stability bounds.

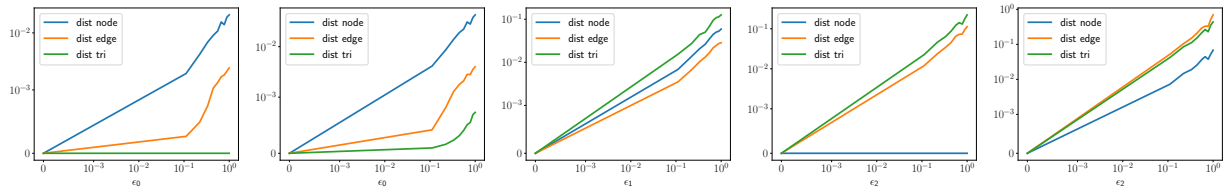
Stability dependence across simplices For 2-simplex prediction of $K = 2$, we measure the distance between the simplicial outputs of SCCNN with and without perturbations on nodes, edges, and triangles, i.e., $\|\boldsymbol{x}_k^L - \hat{\boldsymbol{x}}_k^L\|/\|\boldsymbol{x}_k^L\|$, for $k = 0, 1, 2$. Fig. 7 shows that overall the stabilities of different simplicial outputs are dependent on each other. Specifically, we see that the triangle output is not influenced by the perturbation on node weights until $L = 2$; likewise, the node output is not influenced by the perturbations on triangle weights when $L = 1$. Also, perturbations on the edge weights will perturbe the outputs on nodes, edges, triangles when $L = 1$. This corroborates our discussions in Section 5.

Effect of number of simplices We observe that the same degree of perturbations added to different simplices causes different degrees of instability, owing to the number n_k of k-simplices in Eq. (14). Since $n_0 < n_1 < n_2$, the perturbations on node weights cause less instability than those on edge and triangle weights.

Effect of number of layers As the number of layers increases, Fig. 7 also shows that the stability of SCCNN degrades, which corresponds to our analysis of using shallow layers.

6.4 Trajectory prediction (RQ 1, 4)

We consider the task of predicting trajectories in a synthetic SC and ocean drifters from Schaub et al. (2020), following Roddenberry et al. (2021). From Table 4, we first observe that the SCCNN and Bunch methods



(a) Node pert.$L = 1$ (b) Node pert.$L = 2$ (c) Edge pert.$L = 1$ (d) Tri. pert.$L = 1$ (e) Tri. pert.$L = 2$

Figure 7: The relative difference of SCCNN outputs on simplices of different orders when perturbations are applied to only nodes, edges and triangles and the number of layers varies.

Table 4: Trajectory prediction (accuracy, ↑).

Methods	Synthetic trajectories	Ocean drifters
SNN (Ebli et al., 2020)	65.5±2.4	52.5±6.0
PSNN (Roddenberry et al., 2021)	63.1±3.1	49.0±8.0
SCNN (Yang et al., 2022a)	**67.7±1.7**	53.0±7.8
Bunch (Bunch et al., 2020)	62.3±4.0	46.0±6.2
SCCNN	65.2±4.1	**54.5±7.9**

do not always perform better than those without inter-simplicial couplings. This is because zero inputs are applied on nodes and triangles following Roddenberry et al. (2021), which makes inter-simplicial couplings inconsequential. Secondly, an SCCNN performs better than Bunch on average, and SCNN better than PSNN, showing the advantages of higher-order convolutions. Note that the prediction here aims to find the best candidate from the neighborhood of the end node, which depends on the node degree. Since the average node degree of the synthetic SC is 5.24 and that in the ocean drifter data is 4.81, a random guess has around 20% accuracy. The high standard derivations may result from the limited ocean drifter data size.

6.4.1 Stability analysis (RQ 4)

Differently from Section 6.3.1, we further investigate the stability in terms of the integral Lipschitz properties and convolutional orders. We consider SCNNs (Yang et al., 2022a) with orders $T_{\mathrm{d}} = T_{\mathrm{u}} \in \{1, 3, 5\}$ and train them with regularizations on the integral Lipschitz constants. As shown in Fig. 8, the higher-order case has better stability (smaller ℓ_2 distance between the outputs without and with perturbations) and consistent better accuracy, compared to the lower-order case. This is because the additional flexibility in the higher-order case allows the filters to have better intergral Lipschitz properties and thus better stability, while maintaining the accuracy. We refer to Appendix G.4.4 for a detailed design of the regularizations, as well as more in-depth experimental analysis.

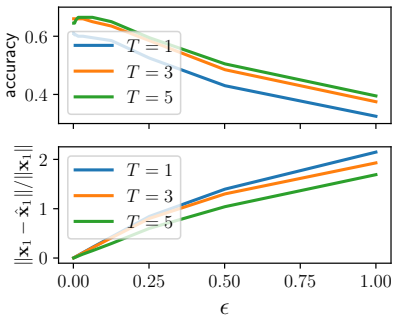


Figure 8: Stability and accuracy of SCCNN versus convolutional orders.

7 Related Works, Discussions and Conclusion

Related works Our work is mainly related to learning methods on SCs. Roddenberry & Segarra (2019) first used $\boldsymbol{L}_{1,\mathrm{d}}$ to build neural networks on edges in a graph setting without the upper edge adjacency. Ebli et al. (2020) then generalized convolutional GNNs (Kipf & Welling, 2017; Defferrard et al., 2016) to simplices by using the Hodge Laplacian. Roddenberry et al. (2021); Yang et al. (2022a) instead uncoupled the lower and upper Laplacians to perform one- and multi-order convolutions, to which Goh et al. (2022); Giusti et al. (2022); Lee et al. (2022) added attention schemes. Keros et al. (2022) considered a variant of Roddenberry et al. (2021) to identify topological "holes" and Chen et al. (2022b) combined shifting on nodes and edges for

link prediction. These works learned within a simplicial level and did not consider the incidence relations (inter-simplicial couplings) in SCs, which was included by Bunch et al. (2020); Yang et al. (2022c). These works considered convolutional-type methods, which can be subsumed by SCCNNs. Meanwhile, Bodnar et al. (2021b); Hajij et al. (2021) generalized the message passing on graphs (Xu et al., 2018a) to SCs, relying on both adjacencies and incidences. Most of these works focused on extending GNNs to SCs by varying the information propagation on SCs with limited theoretical insights into their components. Among them, Roddenberry et al. (2021) discussed the equivariance of PSNN to permutation and orientation, which SCCNNs admit as well. Bodnar et al. (2021b) studied the messgae-passing on SCs in terms of WL test of SCs built by completing cliques in a graph. The more closely related work is (Yang et al., 2022a), which gave only a spectral formulation based on SCFs but not SCCNNs. We refer to Papamarkou et al. (2024); Besta et al. (2024) for an overview of the current progress on learning on SCs.

Discussions In our opinion, the advantage of SCs is not only about them being able to model higher-order network structures, but also support simplicial data, which can be both human-generated data like coauthorship, and physical data like flow-type data. This is why we approached the analysis from the perspectives of both simplicial structures and the simplicial data, i.e., the Hodge theory and spectral simplicial theory (Hodge, 1989; Lim, 2020; Yang et al., 2021; Barbarossa & Sardellitti, 2020; Steenbergen, 2013; Yang et al., 2022b; Govek et al., 2018). We provided insights into why the three principles (P1-P3) are needed and how they can guide the effective and rational learning from simplicial data. As we have practically found, SCCNNs perform well in applications where data exhibits properties characterized by the Hodge decomposition due to the Hodge-awareness, while non-Hodge-aware learners fail at giving rational results. In cases where data does not possess such properties, SCCNNs have better or comparable performance than the ones which violate or do not respect the three principles.

Concurrently, there are works on more general cell complexes, e.g., (Hajij et al., 2020; 2022; Sardellitti et al., 2021; Roddenberry et al., 2022; Bodnar et al., 2021a), where 2-cells inlcude not only triangles, but also general polygon faces. We focus on SCs because a regular cell complex can be subdivided into an SC (Lundell et al., 1969; Grady & Polimeni, 2010) to which the analysis in this paper applies, or we can generalize our analysis by allowing $\boldsymbol{B}_2$ to include 2-cells. This is however informal and does not exploit the power of cell complexes, which relies on cellular sheaves, as studied in (Hansen & Ghrist, 2019; Bodnar et al., 2022).

Conclusion We proposed three principles (P1-P3) for convolutional learning on SCs, summarized in a general architecture, SCCNNs. Our analysis showed this architecture, guided by the three principles, demonstrates an awareness of the Hodge decomposition and performs rational, effective and expressive learning from simplicial data. Furthermore, our study reveals that SCCNNs exhibit stability and robustness against perturbations in the strengths of simplicial connections. Experimental results validate the benefits of respecting the three principles and the Hodge-awareness, as well as the stability results. Overall, our work establishes a solid theoretical fundation for convolutional learning on SCs, highlighting the importance of the Hodge theorem.

References

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, et al. Construction of the literature graph in semantic scholar. *arXiv preprint arXiv:1805.02262*, 2018. Cited on pages 14 and 39.

D Vijay Anand, Soumya Das, and Moo K Chung. Hodge-decomposition of brain networks. *arXiv preprint arXiv:2211.10542*, 2022. Cited on pages 1, 2, and 27.

Sergio Barbarossa and Stefania Sardellitti. Topological signal processing over simplicial complexes. *IEEE Transactions on Signal Processing*, 68:2992–3007, 2020. Cited on pages 2, 7, 17, and 24.

Claudio Battiloro, Lucia Testa, Lorenzo Giusti, Stefania Sardellitti, Paolo Di Lorenzo, and Sergio Barbarossa. Generalized simplicial attention neural networks. *arXiv preprint arXiv:2309.02138*, 2023. Cited on page 28.

Federico Battiston, Giulia Cencetti, Iacopo Iacopini, Vito Latora, Maxime Lucas, Alice Patania, Jean-Gabriel Young, and Giovanni Petri. Networks beyond pairwise interactions: structure and dynamics. *Physics Reports*, 874:1–92, 2020. Cited on page 1.

Austin R Benson, Rediet Abebe, Michael T Schaub, Ali Jadbabaie, and Jon Kleinberg. Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences*, 115(48):E11221–E11230, 2018. Cited on pages 1, 14, and 39.

Austin R Benson, David F Gleich, and Desmond J Higham. Higher-order network analysis takes off, fueled by classical ideas and new data. *arXiv preprint arXiv:2103.05031*, 2021. Cited on page 1.

Maciej Besta, Florian Scheidl, Lukas Gianinazzi, Shachar Klaiman, Jürgen Müller, and Torsten Hoefler. Demystifying higher-order graph neural networks. *arXiv preprint arXiv:2406.12841*, 2024. Cited on pages 2 and 17.

Christian Bick, Elizabeth Gross, Heather A Harrington, and Michael T Schaub. What are higher-order networks? *arXiv preprint arXiv:2104.11329*, 2021. Cited on page 1.

Alberto Bietti and Julien Mairal. Group invariance, stability to deformations, and complexity of deep convolutional representations. *arXiv preprint arXiv:1706.03078*, 2017. Cited on page 11.

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yu Guang Wang, Pietro Liò, Guido F Montufar, and Michael Bronstein. Weisfeiler and lehman go cellular: Cw networks. *Advances in Neural Information Processing Systems*, 34, 2021a. Cited on pages 17 and 29.

Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido F Montufar, Pietro Lio, and Michael Bronstein. Weisfeiler and lehman go topological: Message passing simplicial networks. In *International Conference on Machine Learning*, pp. 1026–1037. PMLR, 2021b. Cited on pages 2, 10, 13, 14, 17, 27, 28, 29, 38, 39, and 41.

Cristian Bodnar, Francesco Di Giovanni, Benjamin Chamberlain, Pietro Liò, and Michael Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns. *Advances in Neural Information Processing Systems*, 35:18527–18541, 2022. Cited on pages 17 and 39.

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021. Cited on page 1.

Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013. Cited on page 11.

Eric Bunch, Qian You, Glenn Fung, and Vikas Singh. Simplicial 2-complex convolutional neural networks. In *TDA & Beyond*, 2020. URL https://openreview.net/forum?id=TLbnsKrt6J-. Cited on pages 2, 5, 6, 7, 10, 11, 13, 14, 16, 17, 27, 28, 29, 30, 32, 38, 39, and 43.

Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. *arXiv preprint arXiv:2006.13318*, 2020. Cited on page 7.

Ozan Candogan, Ishai Menache, Asuman Ozdaglar, and Pablo A Parrilo. Flows and decompositions of games: Harmonic and potential games. *Mathematics of Operations Research*, 36(3):474–503, 2011. Cited on page 2.

Yuzhou Chen, Yulia Gel, and H Vincent Poor. Time-conditioned dances with simplicial complexes: Zigzag filtration curve based supra-hodge convolution networks for time-series forecasting. In *Advances in Neural Information Processing Systems*, 2022a. Cited on page 39.

Yuzhou Chen, Yulia R. Gel, and H. Vincent Poor. Bscnets: Block simplicial complex neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(6):6333–6341, 2022b. doi: 10.1609/aaai.v36i6.20583. URL https://ojs.aaai.org/index.php/AAAI/article/view/20583. Cited on pages 2, 5, 16, and 29.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/04df4d434d481c5bb723be1b6df1ee65-Paper.pdf. Cited on pages 13, 14, 16, 28, and 39.

Stefania Ebli, Michaël Defferrard, and Gard Spreemann. Simplicial neural networks. In *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*, 2020. Cited on pages 2, 5, 10, 13, 14, 15, 16, 28, 38, 39, and 43.

Fernando Gama, Antonio G Marques, Geert Leus, and Alejandro Ribeiro. Convolutional graph neural networks. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 452–456. IEEE, 2019a. Cited on page 28.

Fernando Gama, Alejandro Ribeiro, and Joan Bruna. Stability of graph scattering transforms. *Advances in Neural Information Processing Systems*, 32, 2019b. Cited on page 11.

Fernando Gama, Joan Bruna, and Alejandro Ribeiro. Stability properties of graph neural networks. *IEEE Transactions on Signal Processing*, 68:5680–5695, 2020a. Cited on pages 11 and 27.

Fernando Gama, Elvin Isufi, Geert Leus, and Alejandro Ribeiro. Graphs, convolutions, and neural networks: From graph filters to graph neural networks. *IEEE Signal Processing Magazine*, 37(6):128–138, 2020b. Cited on page 28.

Lorenzo Giusti, Claudio Battiloro, Paolo Di Lorenzo, Stefania Sardellitti, and Sergio Barbarossa. Simplicial attention neural networks. *arXiv preprint arXiv:2203.07485*, 2022. Cited on pages 2, 11, 16, and 28.

Christopher Wei Jin Goh, Cristian Bodnar, and Pietro Lio. Simplicial attention networks. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*, 2022. Cited on pages 11, 16, and 28.

Kiya W Govek, Venkata S Yamajala, and Pablo G Camara. Spectral simplicial theory for feature selection and applications to genomics. *arXiv preprint arXiv:1811.03377*, 2018. Cited on page 17.

Leo J Grady and Jonathan R Polimeni. *Discrete calculus: Applied analysis on graphs for computational science*, volume 3. Springer, 2010. Cited on pages 2, 4, 9, 11, 17, 27, 31, and 39.

Nicola Guglielmi, Anton Savostianov, and Francesco Tudisco. Quantifying the structural stability of simplicial homology. *arXiv preprint arXiv:2301.03627*, 2023. Cited on pages 11 and 31.

Mustafa Hajij, Kyle Istvan, and Ghada Zamzmi. Cell complex neural networks. In *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*, 2020. Cited on pages 17 and 29.

Mustafa Hajij, Ghada Zamzmi, Theodore Papamarkou, Vasileios Maroulas, and Xuanting Cai. Simplicial complex representation learning. *arXiv preprint arXiv:2103.04046*, 2021. Cited on pages 17 and 29.

Mustafa Hajij, Ghada Zamzmi, Theodore Papamarkou, Nina Miolane, Aldo Guzmán-Sáenz, and Karthikeyan Natesan Ramamurthy. Higher-order attention networks. *arXiv preprint arXiv:2206.00606*, 2022. Cited on pages 17 and 29.

Jakob Hansen and Robert Ghrist. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology*, 3(4):315–358, 2019. Cited on pages 17 and 29.

William Vallance Douglas Hodge. *The theory and applications of harmonic integrals*. CUP Archive, 1989. Cited on pages 4 and 17.

Danijela Horak and Jürgen Jost. Spectra of combinatorial laplace operators on simplicial complexes. *Advances in Mathematics*, 244:303–336, 2013. Cited on pages 11, 31, and 39.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012. Cited on pages 9 and 30.

Elvin Isufi and Maosheng Yang. Convolutional filtering in simplicial complexes. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5578–5582, 2022. doi: 10.1109/ICASSP43922.2022.9746349. Cited on page 25.

Junteng Jia, Michael T Schaub, Santiago Segarra, and Austin 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*, pp. 761–771, 2019. Cited on pages 1, 2, 9, and 13.

Xiaoye Jiang, Lek-Heng Lim, Yuan Yao, and Yinyu Ye. Statistical ranking and combinatorial hodge theory. *Mathematical Programming*, 127(1):203–244, 2011. Cited on pages 2, 9, and 13.

Henry Kenlay, Dorina Thano, and Xiaowen Dong. On the stability of graph convolutional neural networks under edge rewiring. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8513–8517. IEEE, 2021. Cited on page 11.

Alexandros D Keros, Vidit Nanda, and Kartic Subr. Dist2cycle: A simplicial neural network for homology localization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):7133–7142, 2022. doi: 10.1609/aaai.v36i7.20673. URL https://ojs.aaai.org/index.php/AAAI/article/view/20673. Cited on pages 16 and 28.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017. Cited on pages 6, 16, and 28.

See Hian Lee, Feng Ji, and Wee Peng Tay. Sgat: Simplicial graph attention network. *arXiv preprint arXiv:2207.11761*, 2022. Cited on page 16.

Lek-Heng Lim. Hodge laplacians on graphs. *SIAM Review*, 62(3):685–715, 2020. Cited on pages 1, 2, 4, 17, and 26.

Albert T Lundell, Stephen Weingram, Albert T Lundell, and Stephen Weingram. Regular and semisimplicial cw complexes. *The Topology of CW Complexes*, pp. 77–115, 1969. Cited on page 17.

Hosein Masoomy, Behrouz Askari, Samin Tajik, Abbas K Rizi, and G Reza Jafari. Topological analysis of interaction patterns in cancer-specific gene regulatory network: persistent homology approach. *Scientific Reports*, 11(1):1–11, 2021. Cited on page 1.

Rohan Money, Joshin Krishnan, Baltasar Beferull-Lozano, and Elvin Isufi. Online edge flow imputation on networks. *IEEE Signal Processing Letters*, 2022. Cited on page 1.

James R Munkres. *Elements of algebraic topology*. CRC press, 2018. Cited on page 26.

Mark EJ Newman, Duncan J Watts, and Steven H Strogatz. Random graph models of social networks. *Proceedings of the national academy of sciences*, 99(suppl_1):2566–2572, 2002. Cited on page 1.

Hoang Nt and Takanori Maehara. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019. Cited on page 7.

Theodore Papamarkou, Tolga Birdal, Michael M Bronstein, Gunnar E Carlsson, Justin Curry, Yue Gao, Mustafa Hajij, Roland Kwitt, Pietro Lio, Paolo Di Lorenzo, et al. Position: Topological deep learning is the new frontier for relational learning. In *Forty-first International Conference on Machine Learning*, 2024. Cited on pages 2 and 17.

Alejandro Parada-Mayorga, Zhiyang Wang, Fernando Gama, and Alejandro Ribeiro. Stability of aggregation graph neural networks. *arXiv preprint arXiv:2207.03678*, 2022. Cited on page 11.

Qiang Qiu, Xiuyuan Cheng, Guillermo Sapiro, et al. Dcfnet: Deep neural network with decomposed convolutional filters. In *International Conference on Machine Learning*, pp. 4198–4207. PMLR, 2018. Cited on page 11.

T Mitchell Roddenberry and Santiago Segarra. Hodgenet: Graph neural networks for edge data. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 220–224. IEEE, 2019. Cited on pages 2, 16, and 43.

T Mitchell Roddenberry, Nicholas Glaze, and Santiago Segarra. Principled simplicial neural networks for trajectory prediction. In *International Conference on Machine Learning*, pp. 9020–9029. PMLR, 2021. Cited on pages 2, 5, 10, 13, 14, 15, 16, 17, 27, 28, 38, 39, 42, and 43.

T Mitchell Roddenberry, Michael T Schaub, and Mustafa Hajij. Signal processing on cell complexes. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8852–8856. IEEE, 2022. Cited on pages 17 and 29.

Vincent Rouvreau. Alpha complex. In *GUDHI User and Reference Manual*. GUDHI Editorial Board, 2015. URL `http://gudhi.gforge.inria.fr/doc/latest/group__alpha__complex.html`. Cited on page 37.

T Konstantin Rusch, Michael M Bronstein, and Siddhartha Mishra. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*, 2023. Cited on page 7.

Aliaksei Sandryhaila and José MF Moura. Discrete signal processing on graphs. *IEEE Transactions on Signal Processing*, 61(7):1644–1656, 2013. Cited on page 28.

Aliaksei Sandryhaila and José MF Moura. Discrete signal processing on graphs: Frequency analysis. *IEEE Transactions on Signal Processing*, 62(12):3042–3054, 2014. Cited on page 28.

Stefania Sardellitti, Sergio Barbarossa, and Lucia Testa. Topological signal processing over cell complexes. In *2021 55th Asilomar Conference on Signals, Systems, and Computers*, pp. 1558–1562. IEEE, 2021. Cited on pages 17 and 29.

Michael T Schaub, Austin R Benson, Paul Horn, Gabor Lippner, and Ali Jadbabaie. Random walks on simplicial complexes and the normalized hodge 1-laplacian. *SIAM Review*, 62(2):353–391, 2020. Cited on pages 11, 15, 31, 39, and 43.

Michael T Schaub, Yu Zhu, Jean-Baptiste Seby, T Mitchell Roddenberry, and Santiago Segarra. Signal processing on higher-order networks: Livin'on the edge... and beyond. *Signal Processing*, 187:108149, 2021. Cited on pages 2, 6, and 25.

Gerard LG Sleijpen and Henk A Van der Vorst. A jacobi–davidson iteration method for linear eigenvalue problems. *SIAM review*, 42(2):267–293, 2000. Cited on page 44.

John Steenbergen. *Towards a spectral theory for simplicial complexes*. PhD thesis, Duke University, 2013. Cited on pages 2 and 17.

Leo Torres, Ann S Blevins, Danielle Bassett, and Tina Eliassi-Rad. The why, how, and when of representations for complex systems. *SIAM Review*, 63(3):435–485, 2021. Cited on page 1.

David S Watkins. *The matrix eigenvalue problem: GR and Krylov subspace methods*. SIAM, 2007. Cited on page 44.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pp. 6861–6871. PMLR, 2019. Cited on page 28.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018a. Cited on page 17.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pp. 5453–5462. PMLR, 2018b. Cited on page 25.

Maosheng Yang, Elvin Isufi, Michael T. Schaub, and Geert Leus. Finite Impulse Response Filters for Simplicial Complexes. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pp. 2005–2009, August 2021. doi: 10.23919/EUSIPCO54536.2021.9616185. ISSN: 2076-1465. Cited on pages 2, 7, 17, 24, and 39.

Maosheng Yang, Elvin Isufi, and Geert Leus. Simplicial convolutional neural networks. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8847–8851, 2022a. doi: 10.1109/ICASSP43922.2022.9746017. Cited on pages 2, 5, 13, 14, 16, 17, 28, 39, and 43.

Maosheng Yang, Elvin Isufi, Michael T. Schaub, and Geert Leus. Simplicial convolutional filters. *IEEE Transactions on Signal Processing*, 70:4633–4648, 2022b. doi: 10.1109/TSP.2022.3207045. Cited on pages 5, 6, 7, 8, 17, 24, 25, and 28.

Maosheng Yang, Viacheslav Borovitskiy, and Elvin Isufi. Hodge-compositional edge gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 3754–3762. PMLR, 2024. Cited on page 13.

Ruochen Yang, Frederic Sala, and Paul Bogdan. Efficient representation learning for higher-order data with simplicial complexes. In *The First Learning on Graphs Conference*, 2022c. URL https://openreview.net/forum?id=nGqJY4DODN. Cited on pages 5, 6, 7, 10, 11, 17, and 29.

Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31, 2018. Cited on pages 14 and 38.

Cameron Ziegler, Per Sebastian Skardal, Haimonti Dutta, and Dane Taylor. Balanced hodge laplacians optimize consensus dynamics over simplicial complexes. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 32(2):023128, 2022. Cited on pages 7 and 27.