Convolutional Filtering in Simplicial Complexes

Isufi, Elvin; Yang, Maosheng

DOI
10.1109/ICASSP43922.2022.9746349

Publication date
2022

Document Version
Final published version

Published in
Proceedings of the ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

Citation (APA)
Isufi, E., & Yang, M. (2022). Convolutional Filtering in Simplicial Complexes. In *Proceedings of the ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5578-5582). [9746349] IEEE . https://doi.org/10.1109/ICASSP43922.2022.9746349

ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9746349

CONVOLUTIONAL FILTERING IN SIMPLICIAL COMPLEXES

Elvin Isufi and Maosheng Yang

ABSTRACT

This paper proposes convolutional filtering for data whose structure can be modeled by a simplicial complex (SC). SCs are mathematical tools that not only capture pairwise relationships as graphs but account also for higher-order network structures. These filters are built by following the shift-and-sum principle of the convolution operation and rely on the Hodge-Laplacians to shift the signal within the simplex. But since in SCs we have also inter-simplex coupling, we use the incidence matrices to transfer the signal in adjacent simplices and build a filter bank to jointly filter signals from different levels. We prove some interesting properties for the proposed filter bank, including permutation and orientation equivariance, a computational complexity that is linear in the SC dimension, and a spectral interpretation using the simplicial Fourier transform. We illustrate the proposed approach with numerical experiments.

Index Terms— Hodge Laplacian, simplicial filter, topological signal processing.

1. INTRODUCTION

Processing data with an irregular structure has been the center of the research attention in the last decade, generalizing signal processing [1] and neural network techniques [2] to graphs. One common factor behind the success of these two directions is the concept of graph filtering, that extends the convolution operation from the Euclidean domain to the graph domain [3–5]. However, graph filters exploit only pair-wise relationships in the network and consider the data as signals over its vertices. But data often live on higher-order network structures such as edges and triangles [6–8]. Typical examples include water flows in water networks or traffic flows in transportation networks.

To deal with this type of data and account for their structure, the more recent research attention has shifted towards data processing with a simplicial structure [8–11], i.e., data living over edges, triangles, and so on. The work in [12] considers the problem of flow denoising via simplicial regularization, while [13] focused on flow interpolation. Authors in [7] introduced the concept of simplicial Fourier transform (SFT), which shows that the approaches in [12, 13] behave as low-pass filtering. To further increase the filter flexibility, the work in [14] used the shift-and-sum principle to develop convolutional simplicial filtering via polynomials in the Hodge Laplacian [6]. In parallel, [15] proposed the simplicial convolutional networks by extending the popular graph neural networks (GNNs) to the simplex. The aggregation function in [15] respects the convolution principle, and it is a particular case of the filter in [14]. Other simplicial neural networks include [16–18], which, although built on the principle of message passing [19] can be seen as an order one simplicial convolution nested into point-wise nonlinearities.

Despite the emerging success, simplicial convolutional filters operate only within a simplicial level and process only one type of simplicial signal. Consequently, they are agnostic to the full SC structure since simplicial signals have not only *intra*-simplex proximities but also *inter*-simplex proximities [6]. E.g., a simplicial convolutional filter operating over the edge space processes only edge signals but ignores the effect of the adjacent vertex and triangle signals. For instance, in a water network we may have missing edge flow measurements but not nodal pressures; hence, we could use the latter and the SC coupling to infer the edge signals. To account for such a coupling in a principled way, we extend [14] to a filter bank to process jointly all SC signals.

Our specific contribution is threefold: i) we propose a filter bank composed of simplicial convolutional filters to process jointly signals living on different simplicial levels; ii) we show the filter bank operates by using local information and it is equivariant to permutations and orientations in the simplex; iii) we characterize the spectral response of the filter bank and show how the different simplicial signals are filtered only in different simplicial frequencies.

2. SIMPLICIAL SIGNAL PROCESSING

In this section, we first lay down some basic concepts about simplicial complexes. Then, we discuss the simplicial signals.

2.1. Geometry of Simplicial Complexes

Given a set of vertices $\mathcal{V} = \{1, \ldots, N\}$, a $k-$simplex $\mathcal{S}^k$ is a subset of $\mathcal{V}$ containing $k+1$ distinct elements. Typical simplices are the 0-simplex including a node, the $1-$simplex including an edge, and the $2-$simplex including a triangle [6]. A simplicial complex of order K, $\mathcal{X}^K$, is a collection of simplices such that for any $k-$simplex $\mathcal{S}^k$, it includes any subset $\mathcal{S}^{k-1} \subset \mathcal{S}^k$ for all $k = 0, \ldots, K$. The number of $k-$simplices in $\mathcal{X}^K$ is N_k. A simplicial complex of order $K = 2$ formed by two disjoint triangles $N_2 = 2$, includes all their edges $N_1 = 6$ and nodes $N_0 = 6$. We also say two simplices $\mathcal{S}^k$ and $\mathcal{S}^{k+1}$ are adjacent if they belong to the same SC.

We represent the proximities between the different simplices via the incidence matrices $\mathbf{B}_k \in \mathbb{R}^{N_{k-1} \times N_k}$, which have as row index the $(k-1)-$simplices and as column index the $k-$simplices. For instance, $\mathbf{B}_1$ is the vertex-to-edge incidence matrix and $\mathbf{B}_2$ is the edge-to-triangle incidence matrix.

Property 1. *Two adjacent incidence matrices* $\mathbf{B}_k$ *and* $\mathbf{B}_{k+1}$ *satisfy the boundary condition* $\mathbf{B}_k\mathbf{B}_{k+1} = \mathbf{0}$ *for all* $k \geq 1$ *[6].*

Using the incidence matrices, we can fully represent the structure of the SC by the Hodge Laplacian matrices,

$$\begin{aligned}
&\mathbf{L}_0 = \mathbf{B}_1\mathbf{B}_1^\top \\
&\mathbf{L}_k = \mathbf{L}_{k\ell} + \mathbf{L}_{ku} = \mathbf{B}_k^\top\mathbf{B}_k + \mathbf{B}_{k+1}\mathbf{B}_{k+1}^\top \; k=1,\ldots,K-1 \quad (1)\\
&\mathbf{L}_K = \mathbf{B}_K^\top\mathbf{B}_K.
\end{aligned}$$

That is, the zero-Hodge Laplacian $\mathbf{L}_0 = \mathbf{B}_1\mathbf{B}_1^\top$ is the popular graph Laplacian and indicates vertex proximities based on their

Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands. This work is supported by the TU Delft AI Labs Programme. e-mail: {e.isufi-1}@tudelft.nl

upper-adjacency via an edge [4]. All the intermediate Laplacians $\mathbf{L}_k$ comprise two terms: the lower Laplacian $\mathbf{L}_{k\ell} := \mathbf{B}_k^\top \mathbf{B}_k$ that captures lower-adjacencies of $k-$simplices; and the upper Laplacian $\mathbf{L}_{ku} := \mathbf{B}_{k+1}\mathbf{B}_{k+1}$ that captures upper-adjacencies of $k-$simplices. E.g., in an SC of order $K = 2$ two edges are lower-adjacent if they have a common vertex and upper-adjacent if they belong to the same triangle. The $K-$th Hodge Laplacian $\mathbf{L}_K = \mathbf{B}_K^\top \mathbf{B}_K$ has only the lower-adjacencies since the SC is of order K.

2.2. Signals over Simplicial Complexes

We are interested in processing signals residing over the different simplicial levels by accounting for the overall structure of the simplicial complex. A $k-$simplicial signal, for short a $k-$signal, x^k is a mapping from the $k-$simplex to the real set $\mathbb{R}^{N_k}$, $x^k : \mathcal{S}^k \to \mathbb{R}^{N_k}$. We will collect the $k-$signal in vector $\mathbf{x}^k = [x_1^k, \dots, x_{N_k}^k]^\top$, where x_i^k is the signal on the ith simplex. The $0-$signal resides over the vertices and matches the graph signal [4]. Likewise, the $1-$signal resides over the edges, the $2-$signal over the triangles, and so on. We will also refer to the collection of all $k-$signals $\mathbf{x}^0, \dots, \mathbf{x}^K$ as a *simplicial complex signal*.

Given the simplicial adjacencies within an SC [cf. (1)] and the coupling between a simplicial complex and its signal, processing SC signals requires exploiting these interactions altogether. We do so by using the Hodge Laplacians in (1) and their Hodge decomposition [6]. The Hodge decomposition decomposes space $\mathbb{R}^{N_k}$ of a $k-$signal into three orthogonal subspaces

$$\mathbb{R}^{N_k} = \text{im}(\mathbf{B}_k^\top) \oplus \text{im}(\mathbf{B}_{k+1}) \oplus \ker(\mathbf{L}_k) \tag{2}$$

where $\text{im}(\cdot)$ and $\ker(\cdot)$ are the image and kernels spaces of a matrix and $\oplus$ is the direct sum of vector spaces. This implies that for any $k-$signal $\mathbf{x}^k$ we can find three signals $\mathbf{x}^{k-1}$, $\mathbf{x}_{\text{h}}^k$, and $\mathbf{x}^{k+1}$ of order $k-1$, k, and $k+1$, respectively such that we can write $\mathbf{x}^k$ as the sum of three orthogonal components

$$\mathbf{x}^k = \mathbf{B}_k^\top \mathbf{x}^{k-1} + \mathbf{B}_{k+1}\mathbf{x}^{k+1} + \mathbf{x}_{\text{h}}^k. \tag{3}$$

This decomposition shows how the inter-simplex coupling imposed by the Hodge Laplacians in (1) translates into an inter-signal coupling. Specifically:

- Operation $\mathbf{x}_{\text{g}}^k := \mathbf{B}_k^\top \mathbf{x}^{k-1} \in \text{im}(\mathbf{B}_k^\top)$ transforms a $k-1$ signal $\mathbf{x}^{k-1}$ into the upper-simplex. Likewise, we can consider the opposite operation $\mathbf{x}_{\text{div}}^{k-1} = \mathbf{B}_k \mathbf{x}^k$ which transforms a $k-$signal into a signal in the lower $k-1$ simplex. Particularizing to $k = 1$, we have that $\mathbf{x}_{\text{g}}^1 = \mathbf{B}_1^\top \mathbf{x}^0$ is an edge flow induced by differentiating the vertex signals $\mathbf{x}^0$; and we refer to as the *gradient flow*. Likewise, $\mathbf{x}_{\text{div}}^0 := \mathbf{B}_1 \mathbf{x}^1$ is a vertex signal obtained by computing the net flow of each node; and we refer to as the *divergence component*.
- Operation $\mathbf{x}_{\text{cur}}^k := \mathbf{B}_{k+1}\mathbf{x}^{k+1} \in \text{im}(\mathbf{B}_{k+1})$ transforms a $k+1$ signal $\mathbf{x}^{k+1}$ into the lower-simplex. Likewise, through the adjoint $\mathbf{B}_{k+1}^\top$ we can transform a $k-$signal into the upper-simplex as $\mathbf{x}_{\text{c}}^{k+1} := \mathbf{B}_{k+1}^\top \mathbf{x}^k$. Particularizing again to $k = 1$, signal $\mathbf{x}_{\text{cur}}^1 = \mathbf{B}_2 \mathbf{x}^2$ contains edge flows induced by triangle flows $\mathbf{x}^2$; and it is called a *curl flow*. Likewise, signal $\mathbf{x}_{\text{c}}^2 = \mathbf{B}_2^\top \mathbf{x}^1$ is a flow circulating along the triangles computed from edge flows.
- Signal $\mathbf{x}_{\text{h}}^k \in \ker(\mathbf{L}_k)$ is called the *harmonic component* and it is that part of a $k-$signals that cannot be induced from the adjacent simplex signals. We can get the harmonic component by solving $\mathbf{L}_k \mathbf{x}_{\text{h}}^k = \mathbf{0}$.

Given this *inter-simplex* coupling between different $k-$signals, we next leverage the shift-and-sum principle and the Hodge Laplacians (1) to induce an *intra-simplex* coupling and develop a principled convolutional filter for SC signals.

3. FILTERS ON SIMPLICIAL COMPLEXES

3.1. Simplicial Convolutional Filters

For a $k-$signal $\mathbf{x}^k$ over a simplex $\mathcal{S}^k$ with Hodge Laplacian $\mathbf{L}_k$, $k = 1, \dots, K-1$, a *simplicial convolution* is defined as

$$\mathbf{y}^k = \Big(\sum_{l_1=0}^{L_1} \alpha_{l_1} \mathbf{L}_{k\ell}^{l_1} + \sum_{l_2=0}^{L_2} \beta_{l_2} \mathbf{L}_{ku}^{l_2} \Big) \mathbf{x}^k \tag{4}$$

where $\alpha_0, \dots, \alpha_{L_1}$ and $\beta_0, \dots, \beta_{L_2}$ are parameters and L_1, L_2 are the convolutional orders in the lower-Laplacian $\mathbf{L}_{k\ell}$ and upper-Laplacian $\mathbf{L}_{ku}$, respectively. Defining then the *simplicial convolutional filtering* matrix

$$\mathbf{H}(\mathbf{L}_k) := \Big(\sum_{l_1=0}^{L_1} \alpha_{l_1} \mathbf{L}_{k\ell}^{l_1} + \sum_{l_2=0}^{L_2} \beta_{l_2} \mathbf{L}_{ku}^{l_2} \Big) \tag{5}$$

we can write (4) as $\mathbf{y}^k = \mathbf{H}(\mathbf{L}_k)\mathbf{x}^k$. The qualifier convolution comes from the fact that in (4) we are shifting the input $\mathbf{x}^k$ over simplex $\mathcal{S}^k$ using both its lower- and upper-adjacencies, weighting each shift, and summing all shifted versions. This is analogous to the convolutional operator in discrete-time signal processing [20] and in graph signal processing [3]. Filter $\mathbf{H}(\mathbf{L}_k)$ is the sum of two polynomials because of the form of the Hodge Laplacian $\mathbf{L}_k$ [cf. (1)] and of Property 1 (i.e., cross-terms are not present since $\mathbf{L}_{k\ell}^{l_1}\mathbf{L}_{ku}^{l_2} = \mathbf{0}$). For $k = 0$ the simplicial convolutional filter reduces to the graph filter $\mathbf{H}(\mathbf{L}_0) = \sum_{l=0}^{L} \alpha_l \mathbf{L}_{0\ell}^l$ [3] and for $k = K$ to $\mathbf{H}(\mathbf{L}_K) = \sum_{l=0}^{L} \beta_l \mathbf{L}_{K\ell}^l$ because $\mathbf{L}_K$ comprises only lower-adjacencies.

It follows from (4) that a simplicial convolutional filter acts on the $k-$signal and propagates neighboring information within simplex $\mathcal{S}^k$ by leveraging paths either via lower- or upper-adjacencies. Computing the output in (4) implies accounting for a local information from at most $L = \max\{L_1, L_2\}$ hops away following either of these paths [14]. Exploiting the recursions $\mathbf{L}_{k\ell}^{l_1}\mathbf{x}^k = \mathbf{L}_{k\ell}(\mathbf{L}_{k\ell}^{l_1-1}\mathbf{x}^k)$ and $\mathbf{L}_{ku}^{l_2}\mathbf{x}^k = \mathbf{L}_{ku}(\mathbf{L}_{ku}^{l_2-1}\mathbf{x}^k)$ the cost of running a simplicial convolution is of order $\mathcal{O}(LN_k)$.

But filter (5) ignores any influence of signals in the adjacent simplices. Since different signals $\mathbf{x}^0, \dots, \mathbf{x}^K$ influence each other via the SC localities, we extend filter (5) to account for the latter. For instance, in a water network we may want to process jointly measurements over the junctions (nodes), flows over pipes (edges), and catchments (triangles) to infer an output related to edge flows [21].

3.2. Simplicial Complex Filter Bank

Given an SC signal $\mathbf{x}^0, \dots, \mathbf{x}^K$, we define an SC filter bank as that generating the output SC signal

$$\begin{aligned} \mathbf{y}^0 &= \mathbf{H}_0(\mathbf{L}_0)\mathbf{x}^0 + \mathbf{H}_1(\mathbf{L}_0)\mathbf{B}_1\mathbf{x}^1 \\ \mathbf{y}^k &= \mathbf{H}_0(\mathbf{L}_k)\mathbf{B}_k^\top\mathbf{x}^{k-1} + \mathbf{H}_1(\mathbf{L}_k)\mathbf{x}^k + \mathbf{H}_2(\mathbf{L}_k)\mathbf{B}_{k+1}\mathbf{x}^{k+1} \\ \mathbf{y}^K &= \mathbf{H}_0(\mathbf{L}_K)\mathbf{B}_K^\top\mathbf{x}^{K-1} + \mathbf{H}_1(\mathbf{L}_K)\mathbf{x}^K \end{aligned} \tag{6}$$

where filters $\mathbf{H}_i(\mathbf{L}_k)$ are of the form in (5) and subscript i indicates they have different coefficients.

This SC filter bank is local in two directions. First, it is *intra*-simplex local since filters $\mathbf{H}_i(\mathbf{L}_k)$ capture information up to $L-$hops away within the $k-$simplex. Second, it is *inter*-simplex local since computing output $\mathbf{y}^k$ implies accounting also for the adjacent $(k-1)-$signal $\mathbf{x}^{k-1}$ and $(k+1)-$signal $\mathbf{x}^{k+1}$. The $0-$output signal is computed by (graph) filtering with $\mathbf{H}_0(\mathbf{L}_0)$ the vertex signal $\mathbf{x}^0$ and by (graph) filtering with $\mathbf{H}_1(\mathbf{L}_0)$ the divergence signal $\mathbf{x}^0_{\text{div}} = \mathbf{B}_1\mathbf{x}^1$. Likewise, we can write the input-output relation for the $1-$output signal as

$$\mathbf{y}^1 = \mathbf{H}_0(\mathbf{L}_1)\mathbf{x}^1_{\text{g}} + \mathbf{H}_1(\mathbf{L}_1)\mathbf{x}^1 + \mathbf{H}_2(\mathbf{L}_1)\mathbf{x}^1_{\text{cur}} \tag{7}$$

which is now simplicial filtering the gradient flow (induced by node signals) $\mathbf{x}^1_{\text{g}} = \mathbf{B}_1^\top\mathbf{x}^0$, the input edge flow signals $\mathbf{x}^1$, and the curl flow induced by triangle signals $\mathbf{x}^1_{\text{cur}} := \mathbf{B}_2\mathbf{x}^2$. The same discussion extends to any simplex $\mathcal{S}^k$. This joint locality explores the computational benefits of the simplicial convolutional filter [cf. (5)] and getting the output SC signal $\mathbf{y}^0, \dots, \mathbf{y}^K$ with a computational cost of order $\mathcal{O}(KLN)$, where $N = \max\{N_0, \dots, N_K\}$ is the maximum number simplices. The linear complexity in N makes the filter bank in (6) practical even for SCs of large dimension, which links well with the linear-complexity of graph filters [22, 23].

Invariances. As we process graph signals with an arbitrary node labeling, also in SCs we have simplicial signals with arbitrary labeling of the nodes and with an arbitrary orientation of the flows [8]. Thus, studying the invariances of the filer bank (6) shows how it exploits the symmetries in the simplicial complexes.

Proposition 1 (Permutation equivariance)**.** Let $\mathcal{X}^K$ be a simplicial complex and consider the permutation matrices $\mathbf{P}_k$ as those belonging to the set

$$\mathcal{P} = \{\mathbf{P}_k \in \{0,1\}^{N_k \times N_k} : \mathbf{P}_k\mathbf{1} = \mathbf{1}, \mathbf{P}_k^\top\mathbf{1} = \mathbf{1}, k \geq 0\}.$$

Permutation matrices $\mathbf{P}_k$ are such that products $\widetilde{\mathbf{x}}^k = \mathbf{P}_k^\top\mathbf{x}^k$ are reorderings of the entries of $\mathbf{x}^k$ and that the permuted Hodge Laplacian $\widetilde{\mathbf{L}}_k = \mathbf{P}_k^\top\mathbf{L}_k\mathbf{P}_k$ is a reordering of the rows and columns of $\mathbf{L}_k$. The simplicial complex filter bank in (6) is permutation equivariant.

Proof. (Sketch) Consider the kth input-output relation in (6). The permutation matrices $\mathbf{P}_k$ transform the incidence matrices as $\widetilde{\mathbf{B}}_k = \mathbf{P}_{k-1}^\top\mathbf{B}_k\mathbf{P}_k$ and $\widetilde{\mathbf{B}}_{k+1} = \mathbf{P}_k^\top\mathbf{B}_{k+1}\mathbf{P}_{k+1}$. Then, using the identify $\mathbf{P}_k^\top\mathbf{P}_k = \mathbf{P}_k\mathbf{P}_k^\top = \mathbf{I}$, the permuted lower- and upper-Laplacians are respectively $\widetilde{\mathbf{L}}_{k\ell} = \mathbf{P}_k^\top\mathbf{L}_{k\ell}\mathbf{P}_k$ and $\widetilde{\mathbf{L}}_{ku} = \mathbf{P}_{k+1}^\top\mathbf{L}_{ku}\mathbf{P}_{k+1}$. Consequently, we can write $\mathbf{H}(\widetilde{\mathbf{L}}_k) = \mathbf{P}_k^\top\mathbf{H}(\mathbf{L}_k)\mathbf{P}_k$. Using the above, we can apply the permuted filters to the permuted inputs $\widetilde{\mathbf{x}}^{k-1} = \mathbf{P}_{k-1}^\top\mathbf{x}^{k-1}$, $\widetilde{\mathbf{x}}^k = \mathbf{P}_k^\top\mathbf{x}^k$, and $\widetilde{\mathbf{x}}^{k+1} = \mathbf{P}_{k+1}^\top\mathbf{x}^{k+1}$ and prove that the output is permuted likewise, i.e., $\widetilde{\mathbf{y}}^k = \mathbf{P}_k^\top\mathbf{y}^k$. □

Proposition 2 (Orientation equivariance)**.** Let $\mathcal{X}^K$ be a simplicial complex of order K and consider the orientation matrices $\mathbf{D}_k$ as those belonging to the set

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

Orientation matrices $\mathbf{D}_k$ are such that products $\widetilde{\mathbf{x}}^k = \mathbf{D}_k^\top\mathbf{x}^k$ are reorientations of the flow directions in vectors $\mathbf{x}^k$ and that the reoriented Hodge Laplacian is given by $\widetilde{\mathbf{L}}_k = \mathbf{D}_k\mathbf{L}_k\mathbf{D}_k$. The simplicial complex filter bank in (6) is orientation equivariant.

Proof. (Sketch) Consider the kth input-output relation in (6). The orientation matrices $\mathbf{D}_k$ transform the incidence matrices as $\widetilde{\mathbf{B}}_k = \mathbf{D}_{k-1}\mathbf{B}_k\mathbf{D}_k$ and $\widetilde{\mathbf{B}}_{k+1} = \mathbf{D}_k\mathbf{B}_{k+1}\mathbf{D}_{k+1}$. Using then the identity $\mathbf{D}_k\mathbf{D}_k = \mathbf{I}$, the oriented lower- and upper-Laplacians are respectively $\widetilde{\mathbf{L}}_{k\ell} = \mathbf{D}_k\mathbf{L}_{k\ell}\mathbf{D}_k$ and $\widetilde{\mathbf{L}}_{ku} = \mathbf{D}_{k+1}\mathbf{L}_{ku}\mathbf{D}_{k+1}$, which in turn leads to the oriented filters $\mathbf{H}(\widetilde{\mathbf{L}}_k) = \mathbf{D}_k\mathbf{H}(\mathbf{L}_k)\mathbf{D}_k$. Using direct substitutions and simple algebra we can prove $\widetilde{\mathbf{y}}^k = \mathbf{D}_k\mathbf{y}^k$. □

Proposition 1 (resp. 2) implies that if relabel the SC (resp. reorient the flows) and apply the filter bank (6), the output is a relabeled (resp. reoriented) version of the output we would have gotten by applying the filter bank before relabelling (resp. reorientation). These equivariances also imply that we can learn the filter bank to process a given simplicial complex by seeing as examples only permuted and reoriented versions of it. I.e., if two parts of the SC are topologically identical and the simplices support identical signal flows, an SC filter bank yields identical outputs. These findings generalize the permutation equivariance seen for graph filters [5, 24].

Remark 1. When operating on signals transformed from lower-/upper-adjacent simplices, the simplicial filter bank uses only lower/upper- Laplacians. Consider the expression for $\mathbf{y}^k$ for $k = 1, \dots, K-1$ in (6). The first term reduces to $\mathbf{H}_0(\mathbf{L}_k)\mathbf{B}_k^\top\mathbf{x}^{k-1} = \mathbf{H}_0(\mathbf{L}_{k\ell})\mathbf{B}_k^\top\mathbf{x}^{k-1} + \beta_{00}\mathbf{B}_k^\top\mathbf{x}^{k-1}$ because of Property 1. Likewise, the third term reduces to $\mathbf{H}_2(\mathbf{L}_k)\mathbf{B}_{k+1}\mathbf{x}^{k+1} = \alpha_{20}\mathbf{B}_{k+1}\mathbf{x}^{k+1} + \mathbf{H}_2(\mathbf{L}_{ku})\mathbf{B}_{k+1}\mathbf{x}^{k+1}$. That is, a signal coming from lower simplices does not propagate into upper-adjacency paths and viceversa. □

4. FREQUENCY RESPONSE

We now analyze the properties of the SC filter bank in the spectral domain to give further insight into its filtering behavior.

4.1. Simplicial Fourier Transform

The $k-$th Hodge Laplacian can be eigendecomposed as $\mathbf{L}_k = \mathbf{U}_k\mathbf{\Lambda}_k\mathbf{U}_k^\top$, where matrix $\mathbf{U}_k = [\mathbf{u}_{k1}, \dots, \mathbf{u}_{kN_k}]$ collects the eigenvectors and $\mathbf{\Lambda}_k = \text{diag}(\lambda_{k1}, \dots, \lambda_{kN_k})$ the eigenvalues on the main diagonal. For a $k-$signal $\mathbf{x}^k$, the simplicial Fourier transform (SFT) is defined as $\hat{\mathbf{x}}^k = \mathbf{U}_k^\top\mathbf{x}^k$ [7]. The inverse SFT is $\mathbf{x}^k = \mathbf{U}\hat{\mathbf{x}}^k$. We refer to the eigenvalues λ_{ki} as the simplicial frequencies. The SFT generalizes the graph Fourier transform [7].

The eigenvectors of $\mathbf{L}_k$ span the three subspaces of the Hodge decomposition [cf. (2)]. That is, there exists some orthogonal eigenvectors $\mathbf{U}_{\text{g}k} \in \mathbb{R}^{N_k \times N_g}$ that span $\text{im}(\mathbf{B}_k^\top)$, $\mathbf{U}_{\text{c}k} \in \mathbb{R}^{N_k \times N_c}$ that span $\text{im}(\mathbf{B}_{k+1})$, and $\mathbf{U}_{\text{h}k} \in \mathbb{R}^{N_k \times N_h}$ that span $\ker(\mathbf{L}_k)$. We collect the corresponding eigenvalues (simplicial frequencies) in sets $\mathcal{Q}_{\text{g}} = \{\lambda_{\text{g}1}, \dots, \lambda_{\text{g}N_{\text{g}}}\}$, $\mathcal{Q}_{\text{c}} = \{\lambda_{\text{c}1}, \dots, \lambda_{\text{c}N_{\text{c}}}\}$, and $\mathcal{Q}_{\text{h}} = \{\lambda_{\text{h}1} = 0, \dots, \lambda_{\text{h}N_{\text{h}}} = 0\}$. Using these eigenvectors, we can project $k-$signals onto the respective spectral components as: $\hat{\mathbf{x}}^k_{\text{g}} = \mathbf{U}_{\text{g}k}^\top\mathbf{x}^k$, $\hat{\mathbf{x}}^k_{\text{c}} = \mathbf{U}_{\text{c}k}^\top\mathbf{x}^k$, and $\hat{\mathbf{x}}^k_{\text{h}} = \mathbf{U}_{\text{h}k}^\top\mathbf{x}^k$, which show how the simplicial Fourier coefficients are spread among the three types of simplicial frequencies. Next, we shall see how the simplicial filter bank acts on these projections and achieves the desired filtering.

4.2. Frequency Response of SC Filters

To understand the spectral behavior of filter bank (6), we need first to understand the spectral behavior of the simplicial convolutional filter (5). Using the above discussion and the eigendecomposition of the kth Hodge Laplacian, the frequency response of the simplicial convolutional filter (5) is

$$\widehat{H}(\lambda_{ki}) = \begin{cases} \alpha_0 + \beta_0 & \text{for } \lambda_{ki} \in \mathcal{Q}_{\text{h}} \\ \beta_0 + \sum_{l_1=0}^{L_1} \alpha_{l_1}\lambda_i^{l_1} & \text{for } \lambda_{ki} \in \mathcal{Q}_{\text{g}} \\ \alpha_0 + \sum_{l_2=0}^{L_2} \beta_{l_2}\lambda_i^{l_2} & \text{for } \lambda_{ki} \in \mathcal{Q}_{\text{c}} \end{cases}. \tag{8}$$

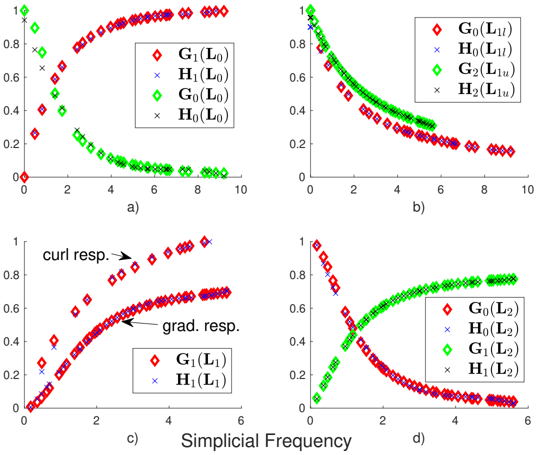

Fig. 1: Data-driven approximation of inverse filtering. The ◇ markers are the inverse filters, the × markers are their approximation with the filter bank (6). a) node filters $k = 1$; b) edge filters $k = 1$ for the gradient and curl influences; c) edge filters $k = 1$ for the curl and gradient frequencies; d) triangle filters $k = 2$.

That is, we have independent control on the simplicial frequencies $\mathcal{Q}_{\mathrm{g}}$ through $\alpha_0, \ldots, \alpha_{L_1}$ and on $\mathcal{Q}_{\mathrm{c}}$ through $\beta_0, \ldots, \beta_{L_2}$ but we have no independent control on the simplicial frequencies $\mathcal{Q}_{\mathrm{h}}$ [14]. The filter input-output relationship at the ith frequency is $\hat{y}_i^k = \widehat{H}(\lambda_i)\hat{x}_i^k$; i.e., it respects the convolution theorem by operating a point-wise multiplication in the SFT domain.

Using these insights, let us now analyze the frequency behavior of the kth input-output relation in (6). From Remark 1, we can write the kth branch as

$$\mathbf{y}^k{=}\mathbf{H}_0(\mathbf{L}_{k\ell})\mathbf{x}_{\mathrm{g}}^k{+}\beta_{00}\mathbf{x}_{\mathrm{g}}^k{+}\mathbf{H}_1(\mathbf{L}_k)\mathbf{x}^k{+}\alpha_{20}\mathbf{x}_{\mathrm{cur}}^k{+}\mathbf{H}_2(\mathbf{L}_{ku})\mathbf{x}_{\mathrm{cur}}^k \quad (9)$$

Using the SFT, its relation with the Hodge decomposition [cf. (2)], and the filter response in (8), the ith input-output spectral relation is

$$\hat{y}_i^k{=}\begin{cases}\alpha_{00}\hat{x}_{\mathrm{g}i}^k + \beta_{00}\hat{x}_{\mathrm{g}i}^k + \widehat{H}_1(\lambda_{ki})\hat{x}_i^k + \alpha_{20}\hat{x}_{\mathrm{cur}i}^k & \\ \qquad +\alpha_{20}\hat{x}_{\mathrm{cur}i}^k + \beta_{20}\hat{x}_{\mathrm{cur}i}^k & \text{for } \lambda_{ki} \in \mathcal{Q}_{\mathrm{h}} \\ \widehat{H}_0(\lambda_{ki})\hat{x}_{\mathrm{g}i}^k + \widehat{H}_1(\lambda_{ki})\hat{x}_i^k & \text{for } \lambda_{ki} \in \mathcal{Q}_{\mathrm{g}} \\ \widehat{H}_1(\lambda_{ki})\hat{x}_i^k + \widehat{H}_2(\lambda_{ki})\hat{x}_{\mathrm{cur}i}^k & \text{for } \lambda_{ki} \in \mathcal{Q}_{\mathrm{c}} \end{cases} \quad (10)$$

where $\hat{x}_{\mathrm{g}i}^k$ is the ith SFT coefficient of $\mathbf{x}_{\mathrm{g}}^k = \mathbf{B}_k^\top \mathbf{x}^{k-1}$ and $\hat{x}_{\mathrm{cur}i}^k$ is the ith SFT coefficient of $\mathbf{x}_{\mathrm{cur}}^k = \mathbf{B}_{k+1}\mathbf{x}^{k+1}$. This implies that filtering the signal from lower-adjacent simplices $\mathbf{B}_k^\top \mathbf{x}^{k-1}$ does not play a role in the filter response over $\mathcal{Q}_{\mathrm{c}}$ and likewise filtering the signal from upper-adjacent simplices $\mathbf{B}_k^\top \mathbf{x}^{k-1}$ does not play a role in the filter response over $\mathcal{Q}_{\mathrm{g}}$. This is intuitive because, for example, for $k = 1$, edge flows induced from a node signal have no curl components (being curl-free), and the edge flows induced from a triangle signal have no gradient components (being divergence-free).

5. NUMERICAL RESULTS

We test the proposed filter bank for two tasks. First, we use it to fit inverse SC filtering in a data-driven approach. Second, we implement a heat kernel diffusion on SC with a low complexity. For both experiments, we generated an alpha SC of 29 nodes, 71 edges and 43 triangles with Gudhi toolbox [25, 26].

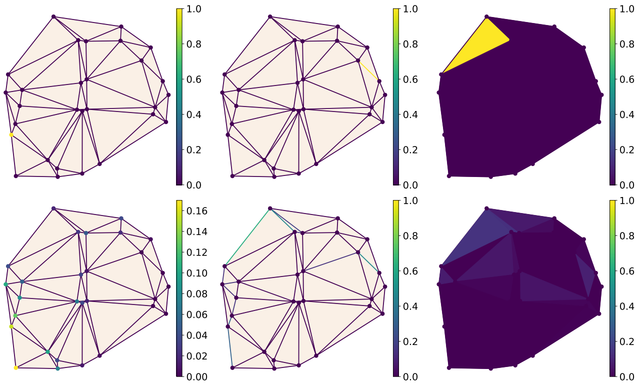

Fig. 2: GHD based on the kernel approximation by SC filter banks. Top three (from left to right): node, edge, and triangle indicator input signals, respectively. Bottom three (from left to right): the diffused results on nodes, edges and triangles.

Model fitting. We generated 10 training samples $(\mathbf{x}^k, \mathbf{y}^k)$ for $k = 0, 1, 2$ by inputing a random simplicial singal $\mathbf{x}^k$ drawn from a zero-mean normal distribution, and outputting $\mathbf{y}^k$ based on model (6). We considered rational filters $\mathbf{G}_i(\mathbf{L}_k) = \mathbf{H}_m^{-1}(\mathbf{L}_k)\mathbf{H}_n(\mathbf{L}_k)$ with m and n indicate different filter parameters. By stacking the shifted input and output training samples, we can solve least-squares problems to design an SC filter bank (6) to fit the desired model parametrized by $\mathbf{G}_i(\mathbf{L}_k)$. With filter orders between 4 and 7, we achieved NMSEs of $0.03, 0.01$ and 0.02 for $k = 0, 1, 2$, respectively, while with the simple simplicial filter [14] we get errors of at least one order higher (resp. 0.3, 0.68, and 0.9). Fig. 1 further shows how the different rational filter responses are well-approximated by the filter bank.

Generalized heat diffusion (GHD). The GHD is used to smooth meshes and identify key signatures in them [27]. The GHD behaves as an SC filter of the form $\mathbf{G}(\mathbf{L}_k) = \exp(-\gamma_k \mathbf{L}_k^2)$ which is computationaly heavy to compute because of the exponential matrix. Instead, we use its analytic frequency response $\widehat{G}(\lambda) = \exp(-\gamma_k \lambda_k^2)$ and universally approximate it with the convolutional filters [14], with an approximation error smaller than 0.1 but with an implementation cost of around two orders lower. Then, we used such filters for the filter bank in (6) with $\gamma_0 = 0.3, \gamma_1 = 0.05$, and $\gamma_2 = 0.5$ to see how indicator input signals $\mathbf{x}^k$ diffuse within their simplices and in the neighboring ones, illustrated in Fig. 2. We also observe that if γ_k is large, the diffusion on the k-simplices attenuates faster, which is expected from the frequency response $\hat{G}(\lambda)$.

6. CONCLUSION

We proposed a simplicial complex convolutional filter bank that can process signals defined on different levels of the SC by capturing both their intra- and inter-simplex proximities. The intra-simplex proximities are captured by leveraging the shift-and-sum principle of the convolutional operation via the Hodge Laplacian matrices of a simplicial complex. Instead, the inter-simplex proximities are captured by leveraging the incidence matrices to transform the signal onto adjacent simplices and then filter with a simplicial filter defined on the adjacent simplices. We show the proposed filter bank is local and equivariant to both permutations in the simplex labeling and flow orientation. We also analyze the filter bank in the simplicial spectral domain and show it acts as a point-wise multiplication between the filter's frequency response and the simplicial Fourier transform of the signal, respecting the convolution principle.

7. REFERENCES

[1] A.Ortega, P.Frossard, J.Kovačević, J. M.Moura, and P.Vandergheynst, “Graph signal processing: Overview, challenges, and applications,” *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[2] Z.Wu, S.Pan, F.Chen, G.Long, C.Zhang, and S. Y.Philip, “A comprehensive survey on graph neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[3] A.Sandryhaila and J. M. F.Moura, “Discrete signal processing on graphs,” *IEEE Trans. Sig. Proc.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

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

[5] F.Gama, E.Isufi, G.Leus, and A.Ribeiro, “Graphs, convolutions, and neural networks: From graph filters to graph neural networks,” *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 128–138, 2020.

[6] L.-H.Lim, “Hodge laplacians on graphs,” *Siam Review*, vol. 62, no. 3, pp. 685–715, 2020.

[7] S.Barbarossa and S.Sardellitti, “Topological signal processing over simplicial complexes,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 2992–3007, 2020.

[8] M. T.Schaub, Y.Zhu, J.-B.Seby, T. M.Roddenberry, and S.Segarra, “Signal processing on higher-order networks: Livin’on the edge... and beyond,” *Signal Processing*, vol. 187, pp. 108149, 2021.

[9] M.Robinson, *Topological signal processing*, vol. 81, Springer, 2014.

[10] O. T.Courtney and G.Bianconi, “Generalized network structures: The configuration model and the canonical ensemble of simplicial complexes,” *Physical Review E*, vol. 93, no. 6, pp. 062311, 2016.

[11] S.Barbarossa and S.Sardellitti, “Topological signal processing: Making sense of data building on multiway relations,” *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 174–183, 2020.

[12] M. T.Schaub and S.Segarra, “Flow smoothing and denoising: Graph signal processing in the edge-space,” in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2018, pp. 735–739.

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

[14] M.Yang, E.Isufi, M. T.Schaub, and G.Leus, “Finite impulse response filters for simplicial complexes,” *arXiv preprint arXiv:2103.12587*, 2021.

[15] S.Ebli, M.Defferrard, and G.Spreemann, “Simplicial neural networks,” *arXiv preprint arXiv:2010.03633*, 2020.

[16] T. M.Roddenberry and S.Segarra, “Hodgenet: Graph neural networks for edge data,” in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2019, pp. 220–224.

[17] T. M.Roddenberry, N.Glaze, and S.Segarra, “Principled simplicial neural networks for trajectory prediction,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 9020–9029.

[18] C.Bodnar, F.Frasca, Y. G.Wang, N.Otter, G.Montúfar, P.Lio, and M.Bronstein, “Weisfeiler and Lehman go topological: Message passing simplicial networks,” *arXiv preprint arXiv:2103.03212*, 2021.

[19] J.Gilmer, S. S.Schoenholz, P. F.Riley, O.Vinyals, and G. E.Dahl, “Neural message passing for quantum chemistry,” *arXiv preprint arXiv:1704.01212*, 2017.

[20] A. V.Oppenheim, J. R.Buck, and R. W.Schafer, *Discrete-time signal processing*, Upper Saddle River, NJ: Prentice Hall, 2001.

[21] M.Herrera, E.Abraham, and I.Stoianov, “A graph-theoretic framework for assessing the resilience of sectorised water distribution networks,” *Water Resources Management*, vol. 30, no. 5, pp. 1685–1699, 2016.

[22] S.Segarra, A.G. Marques, and A.Ribeiro, “Optimal graph-filter design and applications to distributed linear network operators,” *IEEE Trans. Sig. Proc.*, vol. 65, no. 15, pp. 4117–4131, Aug. 2017.

[23] M.Coutino, E.Isufi, and G.Leus, “Advances in distributed graph filtering,” *IEEE Transactions on Signal Processing*, vol. 67, no. 9, pp. 2320–2333, 2019.

[24] E.Isufi, F.Gama, and A.Ribeiro, “EdgeNets: Edge varying graph neural networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21 Jan. 2021.

[25] The GUDHI Project, *GUDHI User and Reference Manual*, GUDHI Editorial Board, 3.5.0 edition, 2022.

[26] V.Rouvreau, “Alpha complex,” in *GUDHI User and Reference Manual*. GUDHI Editorial Board, 3.5.0 edition, 2022.

[27] V.Zobel, J.Reininghaus, and I.Hotz, “Generalized heat kernel signatures,” 2011.