Stability of Simplicial Convolutional Neural Networks

Maosheng Yang*, Elvin Isufi* and Geert Leus†

***Abstract*—xxx**

***Index Terms*—xxx**

I. Introduction

- Network and networked data
- simplicial complex model compared to the graph model: higher-order structure; higher-order data
- simplicial filter from signal processing perspective
- simplicial neural networks: first introduce different simplicial neural network architecture (WL style, RNN style, mostly convolutional structure with different shift operator forms); second, introduce the simplicial complex neural network, which is applied to the case with different levels of data: block SNN, SC2NN
- a general framework by ICASSP paper, stability analysis missing, spectral analysis not complete

II. Background

Simplex, simplicial complex, Hodge Laplacian (weighted version), simplicial signal, simplicial filter

III. Simplicial signal processing

In this section, we recall some important simplicial signal processing concepts, including simplicial complexes and signals, the Hodge decomposition, and simplicial filters.

Simplicial complexes and signals. Given a finite set of vertices $\mathcal{V}$, a k-simplex $\mathcal{S}^k$ is a subset of $\mathcal{V}$ with cardinality $k+1$. A *face* of $\mathcal{S}^k$ is a subset with cardinality k and thus a k-simplex has $k+1$ faces. A *coface* of $\mathcal{S}^k$ is a $(k+1)$-simplex that includes $\mathcal{S}^k$ [**?**], [**?**]. A simplicial complex of order K, $\mathcal{X}^K$, is a collection of k-simplices $\mathcal{S}^k$, $k=0,\ldots,K$, with an inclusion property–for any $\mathcal{S}^k \in \mathcal{X}^K$, then $\mathcal{S}^{k-1} \in \mathcal{X}^K$ if $\mathcal{S}^{k-1} \subset \mathcal{S}^k$. We denote the number of k-simplices in $\mathcal{X}^K$ by N_k. If two simplices share a common face, then they are lower neighbours; if they share a common coface, they are upper neighbours [**?**]. A graph is a simplicial complex where nodes are 0-simplices, and edges are 1-simplices.

In a simplicial complex, we define a k-simplicial signal $\mathbf{x}^k = [x_1^k, \ldots, x_{N_k}^k]$ as a mapping from the k-simplices to the real space $\mathbb{R}^{N_k}$. For example, $\mathbb{R}^{N_0}$ is the graph signal space in GSP, and $\mathbb{R}^{N_1}$ is the space of edge flows. For an edge flow $\mathbf{x}^1 \in \mathbb{R}^{N_1}$, the sign of its entry denotes the direction of the flow relative to a chosen reference orientation [**?**], [**?**].

Hodge Laplacian and decomposition. We represent the relations between $(k-1)$- and k-simplices with the incidence matrix $\mathbf{B}_k$, $k=1,\ldots,K$. The rows of $\mathbf{B}_k$ are indexed by $(k-1)$-simplices and the columns by k-simplices. E.g., matrix $\mathbf{B}_1$ is the node-to-edge incidence matrix, and $\mathbf{B}_2$ is the edge-to-triangle incidence matrix [**?**], [**?**]. We can also use the Hodge Laplacian matrices, $\mathbf{L}_k = \mathbf{B}_k^\top \mathbf{B}_k + \mathbf{B}_{k+1}\mathbf{B}_{k+1}^\top$, where $\mathbf{L}_{k,\mathrm{l}} \triangleq \mathbf{B}_k^\top \mathbf{B}_k$ and $\mathbf{L}_{k,\mathrm{u}} \triangleq \mathbf{B}_{k+1}\mathbf{B}_{k+1}^\top$ are the lower and the upper Laplacians, which encode lower and upper neighbourhoods, respectively. When $k=0$, the Hodge Laplacian is the graph Laplacian $\mathbf{L}_0 = \mathbf{B}_1\mathbf{B}_1^\top$. Hodge Laplacians admit a *Hodge decomposition*, leading to three orthogonal subspaces which the simplicial signal space can be decomposed into, i.e., $\mathbb{R}^{N_k} = \mathrm{im}(\mathbf{B}_k^\top) \oplus \mathrm{im}(\mathbf{B}_{k+1}) \oplus \ker(\mathbf{L}_k)$, where $\oplus$ is the direct sum of vector spaces and $\mathrm{im}(\cdot)$ and $\ker(\cdot)$ are the *image* and *kernel* of a matrix. For $k=1$, these subspaces carry the following interpretations [**?**], [**?**].

Gradient space. By applying matrix $\mathbf{B}_1$ to an edge flow $\mathbf{x}^1$, we compute its net flow at each node, $\mathbf{B}_1\mathbf{x}^1$. The incidence matrix $\mathbf{B}_1$ is called a *divergence operator*. Its adjoint $\mathbf{B}_1^\top$ differentiates a node signal $\mathbf{x}^0$ along the edges to induce an edge flow $\mathbf{B}_1^\top\mathbf{x}^0$, i.e., it is the *gradient operator*. As a result, any flow within $\mathrm{im}(\mathbf{B}_1^\top)$ can be written as the gradient of a node signal $\mathbf{x}^0$, i.e., $\mathbf{x}^1 = \mathbf{B}_1^\top\mathbf{x}^0$. We call $\mathbf{x}^1 \in \mathrm{im}(\mathbf{B}_1^\top)$ a *gradient flow* and the space $\mathrm{im}(\mathbf{B}_1^\top)$ the *gradient space*.

Curl space. We can induce an edge flow from a triangle signal $\mathbf{x}^2$ as $\mathbf{x}^1 = \mathbf{B}_2\mathbf{x}^2$. The induced flow $\mathbf{x}^1 \in \mathrm{im}(\mathbf{B}_2)$ is called a *curl flow* and the space $\mathrm{im}(\mathbf{B}_2)$ is the *curl space*. The adjoint $\mathbf{B}_2^\top$ is the *curl operator*. We can use it to compute the net edge flow of $\mathbf{x}^1$ circulating along the triangles as $\mathbf{B}_2^\top\mathbf{x}^1$.

Harmonic space. The remaining space $\ker(\mathbf{L}_1)$ is the *harmonic space*. Any edge flow $\mathbf{x}^1 \in \ker(\mathbf{L}_1)$ has zero divergence and curl, i.e., it is *divergence-* and *curl-free*.

Due to the boundary condition $\mathbf{B}_1\mathbf{B}_2 = \mathbf{0}$, any gradient flow $\mathbf{x}^1 \in \mathrm{im}(\mathbf{B}_1^\top)$ is curl-free. The space orthogonal to the gradient space, i.e., $\ker(\mathbf{B}_1) = \mathrm{im}(\mathbf{B}_2) \oplus \ker(\mathbf{L}_1)$, is called the *cycle space*, which consists of both the curl space and harmonic space. Any flow in this space is divergence-free.

Simplicial filters. To process simplicial signals $\mathbf{x}^1$, we use a simplicial convolutional filter of the following form:

$$\mathbf{H} = w_0\mathbf{I} + \sum_{k=1}^{K_1} w_{1,k}\mathbf{L}_d^k + \sum_{k=1}^{K_2} w_{2,k}\mathbf{L}_u^k \tag{1}$$

Preliminary results were presented at ICASSP 2022 [**?**]. *Dept. of Intelligent Systems, Delft University of Technology (m.yang-2, e.isufi-1@tudelft.nl). †Dept. of Microelectronics, Delft University of Technology (g.j.t.leus@tudelft.nl). MY is supported by the TU Delft AI Labs Programme.

$$\mathbf{H} = \epsilon \mathbf{I} + \sum_{l_1=1}^{L_1} \alpha_{l_1} (\mathbf{B}_1^\top \mathbf{B}_1)^{l_1} + \sum_{l_2=1}^{L_2} \beta_{l_2} (\mathbf{B}_2 \mathbf{B}_2^\top)^{l_2} \quad (2)$$

where ϵ, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{L_1}]^\top$ and $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_{L_2}]^\top$ are the filter coefficients [**?**]. For ease of exposition, we only discuss the simplicial filter form (2) for the edge signal space $\mathbb{R}^{N_1}$, but similar discussions apply to general simplicial filters with $\mathbf{B}_k$ and $\mathbf{B}_{k+1}$.

Applying $\mathbf{H}$ to an input edge flow $\mathbf{x}^1$ consists of the *simplicial shifting* operations, $\mathbf{L}_{1,\text{l}}\mathbf{x}^1$ and $\mathbf{L}_{1,\text{u}}\mathbf{x}^1$. The ith entry of $\mathbf{L}_{1,\text{l}}\mathbf{x}^1$ is $[\mathbf{L}_{1,\text{l}}\mathbf{x}^1]_i = \sum_{j \in \{\mathcal{N}_{1,i} \cup i\}} [\mathbf{L}_{1,\text{l}}]_{ij} [\mathbf{x}^1]_j$, which is a local operation within the lower neighborhood of the ith edge, likewise for $\mathbf{L}_{1,\text{u}}\mathbf{x}^1$. Moreover, powers $\mathbf{L}_{1,\text{l}}^k \mathbf{x}^1 = \mathbf{L}_{1,\text{l}}(\mathbf{L}_{1,\text{l}}^{k-1}\mathbf{x}^1)$ can be recursively obtained by applying the local operation k times [**?**]. This leads to a distributed implementation of simplicial filtering with a complexity of order $\mathcal{O}(N_1 D)$ for each shifting with D being the maximal number of neighbours.

As we can observe from (2), in the simplicial domain, different sets of coefficients on $\mathbf{L}_{1,\text{l}}$ and $\mathbf{L}_{1,\text{u}}$ enable an independent and flexible filtering within the lower and upper simplicial neighbourhoods. When $L_1 = L_2$ and $\boldsymbol{\alpha} = \boldsymbol{\beta}$, filter $\mathbf{H}$ [cf. (2)] reduces to the basic form $\mathbf{H} = \sum_{l=0}^{L} h_l \mathbf{L}_1^l$ at the cost of losing expressive power and flexibility [**?**].

IV. SCNN

simplicial neural networks (ICASSP content)

- permutation equivariance
- orientation equivariance
- group theory

Consider an L-layer SCNN of a simplicial order p. At layer $l = 1, \ldots, L-1$, with the gth input feature $\mathbf{x}_g^{(l)}$ with $g = 1, \ldots, F_{l-1}$, the corresponding fth intermediate output feature $\mathbf{z}_{gf}^{(l)}$ with $f = 1, \ldots, F_l$ is given by

$$\mathbf{z}_{gf}^{(l)} = \mathbf{H}_{gf}^{(l)} \mathbf{x}_g^{(l-1)}, \quad (3)$$

with $F_0 = 1$ initial input $\mathbf{x}_g^{(0)} = \mathbf{x}$ and the convolutional filter $\mathbf{H}_{gf}^{(l)}$ defined as

$$\mathbf{H}_{gf}^{(l)} = \sum_{k=0}^{K_1} w_{1,kgf}^{(l)} \mathbf{L}_d^k + \sum_{k=0}^{K_2} w_{2,kgf}^{(l)} \mathbf{L}_u^k. \quad (4)$$

By summing over the input features and applying the nonlinearity, we obtain the fth output $\mathbf{x}_f^{(l+1)}$ at layer l as

$$\mathbf{x}_f^{(l)} = \sigma \left[\sum_{g=1}^{F_{l-1}} \mathbf{z}_{gf}^{(l)} \right]. \quad (5)$$

At the last layer $l = L$, to generate a single output feature ($F_L = 1$), we consider a single simplicial filter $\mathbf{H}_g^{(L)}$

$$\mathbf{x}^{(L)} = \sigma \left[\sum_{f=1}^{F_{L-1}} \mathbf{H}_g^{(L)} \mathbf{x}_g^{(L-1)} \right]. \quad (6)$$

Note the first and the last layers are different minorly.

Equivalently, we can write the SCNN layers in matrix compact form. $\mathbf{X}^{(l)} \in \mathbb{R}^{N_p \times F_l}$, $\mathbf{W}_{1,k}^{(l)}, \mathbf{W}_{2,k}^{(l)} \in \mathbb{R}^{F_{l-1} \times F_l}$

$$\mathbf{X}^{(l+1)} = \sigma \left[\sum_{k=0}^{K_1} \mathbf{L}_d^k \mathbf{X}^{(l)} \mathbf{W}_{1,k}^{(l)} + \sum_{k=0}^{K_2} \mathbf{L}_u^k \mathbf{X}^{(l)} \mathbf{W}_{2,k}^{(l)} \right] \quad (7)$$

We collect the weight matricex into tensors, i.e., $\mathcal{W}_1 \in \mathbb{R}^{K_1 \times F_{l-1} \times F_l}$ and $\mathcal{W}_2 \in \mathbb{R}^{K_2 \times F_{l-1} \times F_l}$.

Should we use ∞ instead of K_1, K_2?

V. SIMPLICIAL CONVOLUTIONAL NEURAL NETWORKS

Upon having a simplicial convolutional filter (2), we can build an SCNN by composing filter banks with elementwise nonlinearities. This SCNN applies to any k-simplicial signal but we again illustrate it for edge signals $\mathbf{x}^1$ for the ease of intuition, and omit the superscript to avoid overcrowded notation.

Consider a P-layer SCNN. In the first layer $p = 1$, we apply F filters $\mathbf{H}_1^f$ [cf. (2)] and an elementwise nonlinearity $\sigma(\cdot)$ to the input $\mathbf{x}_0$ to get a collection of F features $\mathbf{x}_1^f$ as

$$\mathbf{x}_1^f = \sigma[\mathbf{z}_1^f] = \sigma[\mathbf{H}_1^f \mathbf{x}_0], \quad f = 1, \ldots, F \quad (8)$$

which constitute the output feature matrix $\mathbf{X}_1 = [\mathbf{x}_1^1, \ldots, \mathbf{x}_1^F]$. In subsequent intermediate layers $p = 2, \ldots, P-1$, we have $\mathbf{X}_{p-1} = [\mathbf{x}_{p-1}^1, \ldots, \mathbf{x}_{p-1}^F] \in \mathbb{R}^{N_1 \times F}$ as input. Each input signal $\mathbf{x}_{p-1}^g, g = 1, \ldots, F$ is passed through a bank of filters $\mathbf{H}_p^{fg}$ to obtain F intermediate outputs $\mathbf{z}_p^{fg} = \mathbf{H}_p^{fg} \mathbf{x}_{p-1}^g, f = 1, \ldots, F$. To avoid exponential filter growth, the intermediate outputs of the different input signals $\mathbf{x}_{p-1}^g$ are summed, thus, the pth layer generates F features $\mathbf{x}_p^f$ as follows

$$\mathbf{x}_p^f = \sigma \left[\sum_{g=1}^{F} \mathbf{z}_p^{fg} \right] = \sigma \left[\sum_{g=1}^{F} \mathbf{H}_p^{fg} \mathbf{x}_{p-1}^g \right] \; f = 1, \ldots, F. \quad (9)$$

The processing in (9) is repeated until the last layer $p = P$, where we consider the output has a single feature, and hence each input is processed by a single filter $\mathbf{H}^g$. Thus, the final output of the SCNN is given by

$$\mathbf{x}_P = \sigma \left[\sum_{g=1}^{F} \mathbf{z}_P^g \right] = \sigma \left[\sum_{g=1}^{F} \mathbf{H}_P^g \mathbf{x}_{P-1}^g \right]. \quad (10)$$

Equations (8), (9) and (10) constitute the SCNN architecture based on the simplicial filter form defined in (2). The lower and upper Hodge Laplacians encode the lower and upper simplicial neighbourhoods, respectively. Simplicial convolutions through filter (2) are performed independently within the lower and upper neighbourhoods and controlled by different sets of coefficients. As we shall show later on, this means that the gradient and curl components of the input features are convolved independently, leading to more expressive power. Next, we discuss the connections of the SCNN with current alternatives and analyze its properties.

Links with related works. In [**?**], a similar convolutional layer was proposed but based on filter $\mathbf{H} = \sum_{l=0}^{L} h_l \mathbf{L}_1^l$. This SNN architecture is a particular case of the proposed SCNN with less expressive power but also with less parameters. The message passing neural network (MPNN) for simplicial

complexes [**?**] aggregates and updates features from direct simplicial neighbours and simplices of different orders, e.g., nodes and triangles. By considering an order-one simplicial convolution as the message aggregation step, we then obtain the architectures in [**?**, Eq. 4] and [**?**, Eq. 7]. When only edge features are available, such approaches are a particular case of the SCNN with order $L_1 = 1$ and $L_2 = 1$. Recurrent architectures are considered for flow interpolation and graph classification in [**?**]. Compared to these works, the SCNN treats features from the lower and upper neighbours differently and considers features from not only direct neighbours but also for multihop neighbors.

Locality and complexity. The intermediate output $\mathbf{z}_p^{fg}$ at the pth layer collects for each edge information from lower neighbours up to L_1 hops away and upper neighbours L_2 hops away through filter (2).This locality comes from the structure of the Hodge Laplacian, likewise that of GNNs [**?**].

When only a single feature is available, we have $1+L_1+L_2$ parameters in such layers. For layers with multiple input and output features, the number of parameters grows F^2 times. The major complexity comes from the convolutional filtering step, which as seen before it is a weighted linear combination of different shifts of a simplicial signal; a local operation within the simplicial neighbourhoods that can be computed recursively. Hence, an SCNN layer performs the simplicial filtering for each edge with a cost of order $\mathcal{O}((L_1 + L_2)D)$. Again, this complexity grows F^2 times when multiple features are used and P times if P layers are considered.

Equivariance and invariance. Here we show that our SCNN is equivalent with respect to a different simplex labeling and different reference flow orientations, introduced in [**?**], [**?**]. Consider the set of simplicial permutation matrices

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

where products $\mathbf{P}_k\mathbf{x}^k$ permute the k-simplicial signal $\mathbf{x}^k$. Let $\mathbf{P} = (\mathbf{P}_0, \mathbf{P}_1, \dots)$ denote a sequence of permutations. Then, the following holds.

Proposition 1: The SCNN is a permutation equivariant architecture. For an input edge flow $\mathbf{x}$, the output of an edge space SCNN layer with a simplicial filter $\mathbf{H}$, $\mathbf{y} = \sigma[\mathbf{H}\mathbf{x}]$, becomes $\mathbf{y}' = \mathbf{P}_1\mathbf{y}$ after a permutation sequence $\mathbf{P}$.

Proof: An SCNN layer with filter $\mathbf{H}$ gives the output $\mathbf{z} = \mathbf{H}\mathbf{x}$. After a sequence of permutations $\mathbf{P}$, the input edge flow $\mathbf{x}$ becomes $\mathbf{P}_1\mathbf{x}$ and the boundary operators become $\mathbf{P}_0\mathbf{B}_1\mathbf{P}_1^\top$ and $\mathbf{P}_1\mathbf{B}_2\mathbf{P}_2^\top$. Thus, the Hodge Laplacians become $\mathbf{P}_1\mathbf{L}_{1,\mathrm{l}}\mathbf{P}_1^\top$ and $\mathbf{P}_1\mathbf{L}_{1,\mathrm{u}}\mathbf{P}_1^\top$ due to $\mathbf{P}_k^\top\mathbf{P}_k = \mathbf{I}$. Then we can express the permuted intermediate output as

$$\begin{aligned}\mathbf{z}' &= \left(\epsilon\mathbf{I} + \sum_{l_1=1}^{L_1}\alpha_{l_1}(\mathbf{P}_1\mathbf{L}_{1,\mathrm{l}}\mathbf{P}_1^\top)^{l_1} + \sum_{l_2=1}^{L_2}\beta_{l_2}(\mathbf{P}_1\mathbf{L}_{1,\mathrm{u}}\mathbf{P}_1^\top)^{l_2}\right)\mathbf{P}_1\mathbf{x} \\ &= \mathbf{P}_1\left(\epsilon\mathbf{I} + \sum_{l_1=1}^{L_1}\alpha_{l_1}\mathbf{L}_{1,\mathrm{l}}^{l_1} + \sum_{l_2=1}^{L_2}\beta_{l_2}\mathbf{L}_{1,\mathrm{u}}^{l_2}\right)\mathbf{x} = \mathbf{P}_1\mathbf{z}.\end{aligned} \tag{11}$$

Thus, the simplicial filter $\mathbf{H}$ is permutation equivariant. Furthermore, since the nonlinearity $\sigma(\cdot)$ is elementwise, SCNNs are permutation equivariant. ∎

In addition, in a simplicial complex, we have also set an arbitrary reference orientation for a simplex. A new reference orientation can be modelled by multiplying the rows and columns of the boundary matrices $\mathbf{B}_k$ and $\mathbf{B}_{k+1}$ where that k-simplex appears and the corresponding simplicial signal value by -1. Let then $\mathbf{D}_k$ be diagonal matrices from the set

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

where $\mathbf{D}_k\mathbf{x}^k$ is the updated k-simplicial signal $\mathbf{x}^k$. Let $\mathbf{D} = (\mathbf{D}_0, \mathbf{D}_1, \dots)$ denote a sequence of orientation changes. Then, the following holds.

Proposition 2: The SCNN is orientation equivariant if the nonlinearity $\sigma(\cdot)$ is odd. Without loss of generality, for an input flow $\mathbf{x}$, the output of an edge space SCNN layer with a simplicial filter $\mathbf{H}$, $\mathbf{y} = \sigma[\mathbf{H}\mathbf{x}]$ becomes $\mathbf{y}' = \mathbf{D}_1\mathbf{y}$ after a sequence of orientation changes $\mathbf{D}$.

Proof: After an orientation change, the edge flow $\mathbf{x}$ becomes $\mathbf{D}_1\mathbf{x}$ and the boundary operators $\mathbf{B}_1$ and $\mathbf{B}_2$ are updated as $\mathbf{D}_0\mathbf{B}_1\mathbf{D}_1$ and $\mathbf{D}_1\mathbf{B}_2\mathbf{D}_2$. Then, the Hodge Laplacians become $\mathbf{D}_1\mathbf{L}_{1,\mathrm{l}}\mathbf{D}_1$ and $\mathbf{D}_1\mathbf{L}_{1,\mathrm{u}}\mathbf{D}_1$. We can then express the new filter output as $\mathbf{z}' = \mathbf{D}_1\mathbf{H}\mathbf{x} = \mathbf{D}_1\mathbf{z}$, following similar steps as in (11). Thus, simplicial filter $\mathbf{H}$ is orientation equivariant. When the nonlinearity $\sigma(\cdot)$ is an odd function, we have $\mathbf{y}' = \sigma(\mathbf{z}') = \mathbf{D}_1\sigma(\mathbf{z}) = \mathbf{D}_1\mathbf{y}$ that completes the proof. ∎

Both propositions can be extended to $k > 1$ cases. Permutation and orientation equivariances preserve the output of an SCNN regardless of the choices of the labeling and reference orientation of the edges. In turn, they allow the SCNN to exploit the internal symmetries in the complex.

VI. Spectral analysis

Hodge decomposition, filter frequency response, Integral Lipschitz filter

Spectral analysis. For the spectral analysis, consider first that the eigenvectors of the Hodge Laplacian $\mathbf{L}_1$ span the three spaces given by the Hodge decomposition: (i) the gradient space $\mathrm{im}(\mathbf{B}_1^\top)$ is spanned by a set of eigenvectors $\mathbf{U}_\mathrm{G}$ of $\mathbf{L}_{1,\mathrm{l}}$ with positive eigenvalues; (ii) the curl space $\mathrm{im}(\mathbf{B}_2)$ is spanned by a set of eigenvectors $\mathbf{U}_\mathrm{C}$ of $\mathbf{L}_{1,\mathrm{u}}$ with positive eigenvalues; and (iii) the harmonic space $\ker(\mathbf{L}_1)$ is spanned by the eigenvectors $\mathbf{U}_\mathrm{H}$ of $\mathbf{L}_1$ with zero eigenvalue. Moreover, we have $\mathrm{im}(\mathbf{L}_1) = \mathrm{im}(\mathbf{U}_\mathrm{G}) \oplus \mathrm{im}(\mathbf{U}_\mathrm{C})$, i.e., gradient and curl spaces make up the image of $\mathbf{L}_1$ [**?**]. Then, we can eigendecompose $\mathbf{L}_1$ as $\mathbf{L}_1 = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ where $\mathbf{U} = [\mathbf{U}_\mathrm{H}\ \mathbf{U}_\mathrm{G}\ \mathbf{U}_\mathrm{C}]$ provides a simplicial Fourier basis, and $\mathbf{\Lambda} = \mathrm{diag}(\mathbf{\Lambda}_\mathrm{H}, \mathbf{\Lambda}_\mathrm{G}, \mathbf{\Lambda}_\mathrm{C})$ with $\mathbf{\Lambda}_\mathrm{H} = \mathrm{diag}(\mathbf{0}_{N_H})$, $\mathbf{\Lambda}_\mathrm{G} = \mathrm{diag}(\lambda_{\mathrm{G},1}, \dots, \lambda_{\mathrm{G},N_\mathrm{G}})$, and $\mathbf{\Lambda}_\mathrm{C} = \mathrm{diag}(\lambda_{\mathrm{C},1}, \dots, \lambda_{\mathrm{C},N_\mathrm{C}})$ collecting the harmonic, gradient, and curl frequencies, respectively; i.e., the simplicial frequencies [**?**].

For an edge flow $\mathbf{x}$, we can find its simplicial Fourier transform (SFT) as $\tilde{\mathbf{x}} = \mathbf{U}^\top\mathbf{x}$. This further defines three embeddings $\tilde{\mathbf{x}} = [\tilde{\mathbf{x}}_\mathrm{H}^\top\ \tilde{\mathbf{x}}_\mathrm{G}^\top\ \tilde{\mathbf{x}}_\mathrm{C}^\top]$: the harmonic embedding $\tilde{\mathbf{x}}_\mathrm{H} = \mathbf{U}_\mathrm{H}^\top\mathbf{x}$, the gradient embedding $\tilde{\mathbf{x}}_\mathrm{G} = \mathbf{U}_\mathrm{G}^\top\mathbf{x}$, and the curl embedding $\tilde{\mathbf{x}}_\mathrm{C} = \mathbf{U}_\mathrm{C}^\top\mathbf{x}$, which contain the weights of $\mathbf{x}$ at harmonic, gradient, and curl frequencies, respectively.

Via the eigendecomposition of $\mathbf{L}_1$, we can analyze the proposed SCNNs in the spectral domain. First, the frequency response of a simplicial convolutional filter $\mathbf{H}$ is given by

$$\tilde{H}(\lambda_i) = \begin{cases} \epsilon, & \text{for } \lambda_i = 0, \\ \epsilon + \sum_{l_1=1}^{L_1} \alpha_{l_1} \lambda_i^{l_1}, & \text{for } \lambda_i \in \mathcal{Q}_{\mathrm{G}}, \\ \epsilon + \sum_{l_2=1}^{L_2} \beta_{l_2} \lambda_i^{l_2}, & \text{for } \lambda_i \in \mathcal{Q}_{\mathrm{C}}, \end{cases} \tag{12}$$

where $\mathcal{Q}_{\mathrm{G}}$ and $\mathcal{Q}_{\mathrm{C}}$ respectively collect distinct gradient and curl frequencies. Thus, the ith entry of the SFT of the intermediate output $\mathbf{z}$ of an SCNN layer can be expressed as $\tilde{z}_i = \tilde{H}(\lambda_i)\tilde{x}_i$. This spectral analysis shows that the SCNN layers compute the high-level simplicial components as the pointwise multiplication of the input embedding and the simplicial filter frequency response, ultimately respecting the convolution theorem. Furthermore, we have here different frequency responses for the gradient and curl components, which corresponds to the different weights on lower and upper Laplacians in filter $\mathbf{H}$. However, for a filter with $\boldsymbol{\alpha} = \boldsymbol{\beta}$ as in [**?**], this would lead to coupling between independent frequencies, and in turn to a limited learning expressiveness.

VII. Stability

Perturbation on the Hodge Laplacian, on the lower and upper parts

Simplicial filter

$$\mathbf{H}(\mathbf{L}_d, \mathbf{L}_u) = \sum_{k=0}^{\infty} h_{1k} \mathbf{L}_d^k + \sum_{k=0}^{\infty} h_{2k} \mathbf{L}_u^k \tag{13}$$

$$h(\lambda) = \begin{cases} h_{1k} + h_{2k}, & \text{for } \lambda = 0, \\ \sum_{k=0}^{\infty} h_{1k} \lambda^k + h_{2k}, & \text{for } \lambda \in \mathcal{Q}_{\mathrm{G}}, \\ \sum_{k=0}^{\infty} h_{2k} \lambda^k + h_{1k}, & \text{for } \lambda \in \mathcal{Q}_{\mathrm{C}}, \end{cases} \tag{14}$$

A. Stability of Simplicial Filters

Definition 1: (Lipschitz simplicial filter). Given a simplicial filter with coefficients $\{h_{1k}, h_{2k}\}_{k=0}^{\infty}$, its frequency response $h(\lambda)$ is given by (14) and sastifies $|h(\lambda)| \leq 1$. We sat that the filter is Lipschitz if there exists constants $C_1, C_2 > 0$ such that the following holds

$$\begin{aligned} &|h(\lambda_2) - h(\lambda_1)| \leq C_1 |\lambda_2 - \lambda_1|, \forall \lambda_1, \lambda_2 \in \mathcal{Q}_{\mathrm{G}}, \text{ and} \\ &|h(\lambda_2) - h(\lambda_1)| \leq C_2 |\lambda_2 - \lambda_1|, \forall \lambda_1, \lambda_2 \in \mathcal{Q}_{\mathrm{C}}. \end{aligned} \tag{15}$$

Absolute perturbation model: Consider a kth order Hodge Laplacian $\mathbf{L} = \mathbf{L}_d + \mathbf{L}_u$, and perturbation matrices $\mathbf{E}_d = \mathbf{V}_1 \mathbf{M}_1 \mathbf{V}_1^{\top}$ and $\mathbf{E}_u = \mathbf{V}_2 \mathbf{M}_2 \mathbf{V}_2^{\top}$, respectively, on the lower and upper counterparts of $\mathbf{L}$. The absolute perturbated Laplacian then can be modeled as follows

$$\hat{\mathbf{L}}_d = \mathbf{L}_d + \mathbf{E}_d, \hat{\mathbf{L}}_u = \mathbf{L}_u + \mathbf{E}_u \tag{16}$$

Given a linear operator $\mathbf{A}$, we use the following operator norm to measure its distance to the perturbated version $\hat{\mathbf{A}}$

$$d(\mathbf{A}, \hat{\mathbf{A}}) = \max_{\mathbf{x}: \|\mathbf{x}\|=1} \|(\mathbf{A} - \hat{\mathbf{A}})\mathbf{x}\| \tag{17}$$

By considering the operator distance measure $d(\mathbf{H}, \hat{\mathbf{H}})$ with $\mathbf{H} := \mathbf{H}(\mathbf{L}_d, \mathbf{L}_u)$ and $\hat{\mathbf{H}} := \mathbf{H}(\hat{\mathbf{L}}_d, \hat{\mathbf{L}}_u)$, we can study the stability of the simplicial filters under small perturbations; likewise, using $d(\Phi, \hat{\Phi})$ with $\Phi := \Phi(\mathbf{L}_d, \mathbf{L}_u)$ and $\hat{\Phi} := \Phi(\hat{\mathbf{L}}_d, \hat{\mathbf{L}}_u)$, we can study the stability of the SCNNs.

Lemma 1: Assume that the small perturbation matrices follow that

$$\|\mathbf{E}_d\| \leq \epsilon_d, \|\mathbf{E}_u\| \leq \epsilon_u. \tag{18}$$

Then, for the ith eigenvector $\mathbf{u}_i$ of $\mathbf{L}$, we have that

$$\begin{aligned} \mathbf{E}_d \mathbf{u}_i &= m_{1i} \mathbf{u}_i + \mathbf{E}_1 \mathbf{u}_i \\ \mathbf{E}_u \mathbf{u}_i &= m_{2i} \mathbf{u}_i + \mathbf{E}_2 \mathbf{u}_i \end{aligned} \tag{19}$$

with $\mathbf{E}_1 = \mathbf{U}\mathbf{M}_1\mathbf{U}^{\top}$ and $\mathbf{E}_2 = \mathbf{U}\mathbf{M}_2\mathbf{U}^{\top}$. Furthermore, we have that $\|\mathbf{E}_1\| \leq \epsilon_d \delta_d$ and $\|\mathbf{E}_2\| \leq \epsilon_u \delta_u$, with $\delta_d = (\|\mathbf{V}_1 - \mathbf{U}\|^2 + 1)^2 - 1$ and $\delta_u = (\|\mathbf{V}_2 - \mathbf{U}\|^2 + 1)^2 - 1$ measuring the eigenvector misalignments.

Proof: We first prove that $\mathbf{E}_d \mathbf{u}_i = m_{1i} \mathbf{u}_i + \mathbf{E}_1 \mathbf{u}_i$. The perturbation matrix on the lower Laplacian can be written as $\mathbf{E}_d = \mathbf{E}_d^{'} + \mathbf{E}_1$ with $\mathbf{E}_d^{'} = \mathbf{U}\mathbf{M}_1\mathbf{U}^{\top}$ and $\mathbf{E}_1 = (\mathbf{V}_1 - \mathbf{U})\mathbf{M}_1(\mathbf{V}_1 - \mathbf{U})^{\top} + \mathbf{U}\mathbf{M}_1(\mathbf{V}_1 - \mathbf{U})^{\top} + (\mathbf{V}_1 - \mathbf{U})\mathbf{M}_1\mathbf{U}$. For the ith eigenvector $\mathbf{u}_i$, we have that

$$\mathbf{E}_d \mathbf{u}_i = \mathbf{E}_d^{'} \mathbf{u}_i + \mathbf{E}_1 \mathbf{u}_i = m_{1i} \mathbf{u}_i + \mathbf{E}_1 \mathbf{u}_i \tag{20}$$

where the second equality follows from the orthogonality of $\mathbf{U}$. Since we assume that $\|\mathbf{E}_d\| \leq \epsilon_d$, it follows that $\|\mathbf{M}_1\| \leq \epsilon_d$. Then, due to the triangle inequality, we have that

$$\begin{aligned} \|\mathbf{E}_1\| \leq & \|(\mathbf{V}_1 - \mathbf{U})\mathbf{M}_1(\mathbf{V}_1 - \mathbf{U})^{\top}\| \\ & + \|\mathbf{U}\mathbf{M}_1(\mathbf{V}_1 - \mathbf{U})^{\top}\| + \|(\mathbf{V}_1 - \mathbf{U})\mathbf{M}_1\mathbf{U}\| \\ \leq & \|\mathbf{V}_1 - \mathbf{U}\|^2 \|\mathbf{M}_1\| + 2\|\mathbf{V}_1 - \mathbf{U}\| \|\mathbf{M}_1\| \|\mathbf{U}\| \\ \leq & \epsilon_d \|\mathbf{V}_1 - \mathbf{U}\|^2 + 2\epsilon \|\mathbf{V}_1 - \mathbf{U}\| \\ = & \epsilon_d ((\|\mathbf{V}_1 - \mathbf{U}\|^2 + 1)^2 - 1) = \epsilon_d \delta_d, \end{aligned} \tag{21}$$

which completes the proof for the lower perturbation matrix. Likewise, we can prove the perturbation matrix of the upper Laplacian operating on the eigenvector $\mathbf{u}_i$. ∎

Theorem 1: (Stability of simplicial filters) Consider the pth order simplices with cardinality N_p. Let $\mathbf{L}_d$ and $\mathbf{L}_u$ be the corresponding lower and upper Hodge Laplacians. Let $\hat{\mathbf{L}}_d$ and $\hat{\mathbf{L}}_u$ be the perturbated versions based on the model (16) with $\mathbf{E}_d = \mathbf{V}_1 \mathbf{M}_1 \mathbf{V}_1^{\top}$ and $\mathbf{E}_u = \mathbf{V}_2 \mathbf{M}_2 \mathbf{V}_2^{\top}$ being the absolute perturbation matrices. Assume the perturbations satisfy that

$$\begin{aligned} d(\mathbf{L}_d, \hat{\mathbf{L}}_d) &= \|\mathbf{E}_d\| \leq \epsilon_d \\ d(\mathbf{L}_u, \hat{\mathbf{L}}_u) &= \|\mathbf{E}_u\| \leq \epsilon_u. \end{aligned} \tag{22}$$

For a Lipschitz simplicial filter with Lipschitz constants C_1 and C_2, the operator distance between filters $\mathbf{H}(\mathbf{L}_d, \mathbf{L}_u)$ and $\mathbf{H}(\hat{\mathbf{L}}_d, \hat{\mathbf{L}}_u)$ follows that

$$d(\mathbf{H}, \hat{\mathbf{H}}) \leq C_1 (1 + \delta_d \sqrt{N_p}) \epsilon_d + C_2 (1 + \delta_u \sqrt{N_p}) \epsilon_u + \mathcal{O}(\epsilon_d^2 + \epsilon_u^2) \tag{23}$$

with $\delta_d = (\|\mathbf{V}_1 - \mathbf{U}\|^2 + 1)^2 - 1$ and $\delta_u = (\|\mathbf{V}_2 - \mathbf{U}\|^2 + 1)^2 - 1$ measuring the eigenvector misalignments between the Hodge Lapalcians and the perturbations.

B. Stability under relative perturbations

Relative perturbation model. Consider a kth order Hodge Laplacian $\mathbf{L} = \mathbf{L}_d + \mathbf{L}_u$. The relative perturbated Laplacian

can be modeled as

$$\begin{aligned}\hat{\mathbf{L}}_d &= \mathbf{L}_d + \mathbf{E}_d\mathbf{L}_d + \mathbf{L}_d\mathbf{E}_d,\\ \hat{\mathbf{L}}_u &= \mathbf{L}_u + \mathbf{E}_u\mathbf{L}_u + \mathbf{L}_u\mathbf{E}_u\end{aligned} \tag{24}$$

Definition 2: (Integral Lipschitz simplicial filter). Given a simplicial filter with coefficients $\{h_{1k}, h_{2k}\}_{k=0}^{\infty}$, its frequency response $h(\lambda)$ is given by (14) and sastifies $|h(\lambda)| \leq 1$. We sat that the filter is Lipschitz if there exists constants $C_1, C_2 > 0$ such that the following holds

$$\begin{aligned}|h(\lambda_2) - h(\lambda_1)| &\leq C_1 \frac{|\lambda_2 - \lambda_1|}{|\lambda_2 + \lambda_1|/2}, \forall \lambda_1, \lambda_2 \in \mathcal{Q}_{\mathrm{G}},\\ |h(\lambda_2) - h(\lambda_1)| &\leq C_2 \frac{|\lambda_2 - \lambda_1|}{|\lambda_2 + \lambda_1|/2}, \forall \lambda_1, \lambda_2 \in \mathcal{Q}_{\mathrm{C}},\end{aligned} \tag{25}$$

which leads to the derivative of the frequency response satisfying

$$|\lambda h'(\lambda)| \leq C_1, \forall \lambda \in \mathcal{Q}_{\mathrm{G}} \text{ and } |\lambda h'(\lambda)| \leq C_2, \forall \lambda \in \mathcal{Q}_{\mathrm{C}} \tag{26}$$

Theorem 2: (Stability of simplicial filters) Consider the set of the pth order simplices with cardinality N_p. Let $\mathbf{L}_d$ and $\mathbf{L}_u$ be the corresponding lower and upper Hodge Laplacians. Let $\hat{\mathbf{L}}_d$ and $\hat{\mathbf{L}}_u$ be the perturbated versions based on the relative model (24) with $\mathbf{E}_d = \mathbf{V}_1\mathbf{M}_1\mathbf{V}_1^{\top}$ and $\mathbf{E}_u = \mathbf{V}_2\mathbf{M}_2\mathbf{V}_2^{\top}$ being the relative perturbation matrices. Assume the perturbations satisfy that

$$\|\mathbf{E}_d\| \leq \epsilon_d \text{ and } \|\mathbf{E}_u\| \leq \epsilon_u. \tag{27}$$

For a Lipschitz simplicial filter with integral Lipschitz constants C_1 and C_2, the operator distance between filters $\mathbf{H}(\mathbf{L}_d, \mathbf{L}_u)$ and $\mathbf{H}(\hat{\mathbf{L}}_d, \hat{\mathbf{L}}_u)$ follows that

$$d(\mathbf{H}, \hat{\mathbf{H}}) \leq 2C_1(1+\delta_d\sqrt{N_p})\epsilon_d + 2C_2(1+\delta_u\sqrt{N_p})\epsilon_u + \mathcal{O}(\epsilon_d^2+\epsilon_u^2) \tag{28}$$

with $\delta_d = (\|\mathbf{V}_1 - \mathbf{U}\|^2+1)^2 - 1$ and $\delta_u = (\|\mathbf{V}_2 - \mathbf{U}\|^2+1)^2 - 1$ measuring the eigenvector misalignments between the Hodge Lapalcians and the perturbations.

C. SCNN stability

Definition 3: (Lipschitz continuous nonlinearity) A pointwise nonlinearity $\sigma(\cdot)$ is Lipschitz continuous with a constant C_σ if it follows that $|\sigma(b) - \sigma(a)| \leq C_\sigma|b - a|, \forall a, b \in \mathbb{R}$.

Theorem 3: (SCNN stability). Consider the pth order simplex domain with cardinality N_p. Let $\mathbf{L}_d, \mathbf{L}_u$ and $\hat{\mathbf{L}}_d, \hat{\mathbf{L}}_u$ be the lower and upper Hodge Laplacians before and after the perturbations by $\mathbf{E}_d$ and $\mathbf{E}_u$, respectively, following the absolute or relative model [cf. (16) or (24)]. Assume that $\|\mathbf{E}_d\| \leq \epsilon_d$ and $\|\mathbf{E}_u\| \leq \epsilon_u$. Define an L-layer SCNN $\mathbf{\Phi}(\cdot)$ with a bank of filter weigths $\{w_{1,kgf}^{(l)}\}$ and $\{w_{2,kgf}^{(l)}\}$ on the lower and upper Laplacians, respectively, for $l = 1, \ldots, L$, $g = 1, \ldots, F_{l-1}$ and $f = 1, \ldots, F_l$, and a C_σ-Lipschitz continuous pointwise nonlinearity $\sigma(\cdot)$. Then, for two SCNNs $\mathbf{\Phi} := \mathbf{\Phi}(\mathbf{L}_d, \mathbf{L}_u)$ and $\hat{\mathbf{\Phi}} := \mathbf{\Phi}(\hat{\mathbf{L}}_d, \hat{\mathbf{L}}_u)$, it holds that

$$d(\mathbf{\Phi} - \hat{\mathbf{\Phi}}) \leq \Delta(C_\sigma B)^{L-1} L C_\sigma \prod_{l=1}^{L-1} F_l + \mathcal{O}(\epsilon_d^2 + \epsilon_u^2). \tag{29}$$

where we have $\Delta = C_1(1 + \delta_d\sqrt{N_p})\epsilon_d + C_2(1 + \delta_u\sqrt{N_p})\epsilon_u$ for the absolute perturbation model (16) with C_1, C_2-Lipschitz SCNN filter banks and $\|\hat{\mathbf{H}}_{gf}^{(l)}\| \leq B$, or we have $\Delta = 2C_1(1 + \delta_d\sqrt{N_p})\epsilon_d + 2C_2(1 + \delta_u\sqrt{N_p})\epsilon_u$ for the relative perturbation model (24) with C_1, C_2-integral Lipschitz filter banks and $\|\hat{\mathbf{H}}_{gf}^{(l)}\| \leq B$.

VIII. Experiments

- Dataset primary school dataset (also check the reviewed EUSIPCO paper for how to use this dataset)
-

IX. Conclusion

The conclusion goes here.

HOW TO SOLVE THE PROBLEMS IN OUR PROJECT?

Tikhonov regularization: consider the item-item graph Tikhonov regularization

$$\min_{\mathbf{X}\in\mathbb{R}^{U\times I}} \|\mathbf{Y}-\mathbf{X}\|_F^2+\mu\, tr\left(\mathbf{X}^\top\mathbf{L}\mathbf{X}\right) \tag{30}$$

where $\|\cdot\|_F$ is the Frobenius norm.

Equivalent, for the ith item, the regularization problem has the form to solve the ratings $\mathbf{x}_i$ for all users

$$\min_{\mathbf{x}_i\in\mathbb{R}^U} \|\mathbf{y}_i-\mathbf{x}_i\|_2^2+\mu\,\mathbf{x}_i^\top\mathbf{L}\mathbf{x}_i. \tag{31}$$

Let us write the objective out as the following form (subscript i omitted)

$$\begin{aligned}&(\mathbf{y}-\mathbf{x})^\top(\mathbf{y}-\mathbf{x})+\mu\,\mathbf{x}^\top\mathbf{L}\mathbf{x}\\ =&(\mathbf{x}^\top\mathbf{x}-2\mathbf{y}^\top\mathbf{x}+\mathbf{y}^\top\mathbf{y})+\mu\,\mathbf{x}^\top\mathbf{L}\mathbf{x}\end{aligned} \tag{32}$$

which is a convex function w.r.t., $\mathbf{x}$. We can find its gradient as follows

$$\nabla_{\mathbf{x}} = 2\mathbf{x}-2\mathbf{y}+2\mu\mathbf{L}\mathbf{x}. \tag{33}$$

Since above gradient has a zeros, i.e., $\nabla_{\mathbf{x}}=\mathbf{0}\in\mathbb{R}^U$ has a closed-form solution, we can find the optimum $\mathbf{x}^*$ as

$$\begin{aligned}\nabla_{\mathbf{x}} &= \mathbf{0}\\ (\mathbf{I}+\mu\mathbf{L})\mathbf{x} &= \mathbf{y}\\ \mathbf{x}^* &= (\mathbf{I}+\mu\mathbf{L})^{-1}\mathbf{y}.\end{aligned} \tag{34}$$

matrix form solution

Intuitively, the solution in the matrix form should be

$$\mathbf{X}^* = (\mathbf{I}+\mu\mathbf{L})^{-1}\mathbf{Y}. \tag{35}$$

But can you find this solution starting from the objective function of the matrix form? Try it with matrix cookbook.

other regularizers

Depends on if we have a closed-form gradient? closed-form solution from setting the gradient as zero? then we choose the way of solving the problem

- adjacency based 2-norm variation measure $\|\mathbf{x}-\mathbf{A}_n\mathbf{x}\|_2^2$:
- total variation 1-norm $\|\mathbf{x}-\mathbf{A}_n\mathbf{x}\|_1$: check the ISTA algorithm based on soft-thresholding
- SLIM method has two regularizers: $\|\mathbf{w}\|_2^2+\|\mathbf{w}\|_1$

ℓ_1 norm

Let us have a closer look at the ℓ_1 norm, the uth element

$$(\|\mathbf{x}\|_1)_u = \begin{cases} x_u, & \text{if } x_u>0\\ -x_u, & \text{if } x_u<0\\ 0, & \text{if } x_u=0.\end{cases} \tag{36}$$

What about its gradient? Technically, we cannot find the gradient, why? Check the plot of $|x|$, at $x=0$ there are indefinite gradients. Thus, so-called subgradient is introduced.

$$(\nabla_{\mathbf{x}}\|\mathbf{x}\|_1)_u = \begin{cases} 1, & \text{if } x_u>0\\ -1, & \text{if } x_u<0\\ [-1,1], & \text{if } x_u=0.\end{cases} \tag{37}$$

Puting this subgradient together with the gradient of the data fitting term, the uth gradient has the form

$$0 = \begin{cases} x_u-y_u+\mu, & \text{if } x_u>0\\ x_u-y_u-\mu, & \text{if } x_u<0\\ [-y_u-\mu, y_u+\mu], & \text{if } x_u=0\end{cases} \tag{38}$$

Then, we have the optimum $\mathbf{x}^*$ with the uth entry

$$x_u = \begin{cases} y_u-\mu, & \text{if } x_u>0 \text{ or } y_u\geq\mu\\ y_u+\mu, & \text{if } x_u<0 \text{ or } y_u\leq-\mu\\ 0, & \text{if } -\mu\leq y_u\geq\mu.\end{cases} \tag{39}$$

We also write above operator as, known as soft-thresholding operator

$$\mathbf{x} = S_\mu(\mathbf{y}) := (|\mathbf{y}|-\mu)_+\text{sign}(\mathbf{y}). \tag{40}$$

The famous ISTA algorithm is a method with soft-thresholding the gradient-descent step.

final notes

We introduce these for the sake of understanding what optimization is doing, why ℓ_1 norm is more involved than ℓ_2 norm. But for this research project, we still also suggest to use the well-known iterative algorithms, e.g., sgd, adam, coordinate descent. These algorithms are all well embedded in python packages or in other languages.