
Sheaves Reloaded: A Directional Awakening

Stefano Fiorini*

Pattern Analysis & Computer Vision
Istituto Italiano di Tecnologia, Genoa, Italy
s.fiorini1994@gmail.com

Hakan Aktas*

University of Cambridge, Cambridge, UK
hea39@cam.ac.uk

Iulia Dută

University of Cambridge, Cambridge, UK
id366@cam.ac.uk

Stefano Coniglio

Department of Economics
University of Bergamo, Bergamo, Italy

Pietro Morerio

Pattern Analysis & Computer Vision
Istituto Italiano di Tecnologia, Genoa, Italy

Alessio Del Bue

Pattern Analysis & Computer Vision
Istituto Italiano di Tecnologia, Genoa, Italy

Pietro Liò

University of Cambridge, Cambridge, UK

Abstract

Sheaf Neural Networks (SNNs) represent a powerful generalization of a Graph Neural Networks (GNNs) that have significantly improved our ability to model complex relational data. While directionality has been shown to substantially boost the performance of graph-learning tasks and is key to many real-world applications, existing SNNs fall short in representing it. To address this limitation, we introduce the Directed Cellular Sheaf, a special type of cellular sheaf designed to explicitly account for edge orientation. Building on it, we define a new sheaf Laplacian, the Directed Sheaf Laplacian $L^{\tilde{F}}$, which captures both the graph’s topology and its directional information. This operator serves as the backbone of the Directed Sheaf Neural Network (DSNN), the first SNN model to embed a directional bias into its architecture. Extensive experiments on nine real-world benchmarks show that DSNN consistently outperforms baseline methods.

1 Introduction

The fast-paced progress in neural networks and deep learning has provided researchers and practitioners with ever more powerful tools for capturing the relationships underlying complex data. Sheaf Neural Networks (SNNs) have recently emerged as a powerful extension of traditional Graph Neural Networks (GNNs) [1, 2]. Such networks rely on the algebraic notion of a *cellular sheaf*, which equips a graph with a geometrical structure that assigns vector spaces to nodes/edges and *restriction maps* which relate vertex features to edge features, shaping the edge-specific communication. SNNs not only allow working in a higher-dimensional feature space, but also naturally mitigate over-smoothing and improve performance in heterophilic graphs in which neighboring nodes may have dissimilar features [2].

A key limitation of current SNNs is their inability to fully capture the underlying topology of a graph, as they ignore edge orientations and only model undirected connections. For this reason, in this paper,

*Equal contribution.

we explore how to incorporate edge directions within the SNN framework in a principled way. Such a research endeavor is crucial not only to replicate the gains that SNNs have shown over traditional GNNs in the undirected setting, but also to harness the benefits that edge directionality has brought to classical GNNs [3] in tasks where data is inherently structured as a directed graph. Moreover, our work aligns with evidence that edge orientation plays a critical role in complex networks [4], as directionality underpins key topological and dynamical phenomena that can profoundly influence a system’s behavior [5, 6].

To enhance the representational capacity of SNNs in scenarios where directional information is critical, we introduce the concept of *Directed Cellular Sheaves*. Unlike traditional cellular sheaves used within state-of-the-art SNNs, which assign vector spaces (or more general algebraic structures) to the cells of a complex without an intrinsic notion of orientation, our framework incorporates directionality directly into the sheaf structure. Building on this framework, we first define the *Directed Coboundary Operator* $\tilde{\delta}$ associated with the Directed Cellular Sheaf. Subsequently, we rely on $\tilde{\delta}$ to define the *Directed Sheaf Laplacian* (DSL) operator $L^{\tilde{\mathcal{F}}}$. This operator not only captures the topological structure of the underlying graph but also faithfully integrates the sign and directionality of its edges.

Our main contributions are summarized as follows:

- We introduce the *Directed Cellular Sheaf*, a mathematical construct that enriches directed graphs by enabling a principled representation of directional interactions between nodes. This structure assigns linear maps between the data spaces associated with the edges and vertices of a graph, which ensures that information flow between nodes depends on the orientation of the edges.
- We propose the *Directed Sheaf Neural Network* (DSNN)—an SNN architecture explicitly designed to include an inductive bias that reflects the directional structure of the graph.
- We conduct extensive experiments on nine real-world datasets and one synthetic dataset, demonstrating the advantages of our proposed way to incorporate directionality in an SNNs via the Directed Cellular Sheaf and its Laplacian operator $L^{\tilde{\mathcal{F}}}$.

2 Background & Related Work

2.1 Cellular Sheaves

In the classical setting, a *sheaf* assigns data (such as sets, groups, or vector spaces) to open sets of a topological space (such as points, open segments, and open disks), together with restriction maps that propagate this data to open subsets within them. A *cellular sheaf* [7, 8] modifies this perspective by replacing open sets with cells of a cell complex (where 0-cells are points, 1-cells edges, 2-cells faces, etc.). It assigns a vector space to each cell and a linear restriction map from each higher-dimensional cell to each of its faces, reflecting the hierarchical structure of the complex. In line with recent works on SNNs [1, 2], we focus on cell complexes consisting only of 0-cells and 1-cells, which coincide with the nodes and edges of a graph, and on lower-to-higher dimensional mappings from nodes to edges. In such models, the sheaf structure enables a principled generalization of message-passing architectures by allowing node features to propagate through edge-level transformations governed by linear restriction maps.

Following [1], we define the *cellular sheaf* of an undirected graph $G = (V, E)$ with $n = |V|$ and $m = |E|$ as the triple $(\{\mathcal{F}(u)\}_{u \in V}, \{\mathcal{F}(e)\}_{e \in E}, \{\mathcal{F}_{u \leq e}\}_{e \in \Gamma(u)})$, containing a vector space $\mathcal{F}(u)$ associated with each vertex $u \in V$, a vector space $\mathcal{F}(e)$ associated with each edge $e \in E$, and a linear map $\mathcal{F}_{u \leq e} : \mathcal{F}(u) \rightarrow \mathcal{F}(e)$ for each edge $e \in \Gamma(u)$, where $\Gamma(u)$ is the subset of edges incident on u . In line with the SNN literature, all vector spaces are assumed to be real. In the cellular sheaf, the vector spaces are referred to as *stalks*, while the linear maps are called *restriction maps*. In this framework, the vertex stalks $\mathcal{F}(u)$ represent the node feature vectors (traditionally denoted as x_v in the graph-learning literature). The space formed by all the spaces associated with the nodes (resp., edges) of the graph is called the space of 0-cochains $C^0(G; \mathcal{F}) = \bigoplus_{u \in V} \mathcal{F}(u)$ (resp., the space of 1-cochains $C^1(G; \mathcal{F}) = \bigoplus_{e \in E} \mathcal{F}(e)$). The inter-vertex constraints are captured by the *coboundary operator* $\delta : C^0(G; \mathcal{F}) \rightarrow C^1(G; \mathcal{F})$, which, given an arbitrary orientation on the edges (where, for each $e = \{u, v\} \in E$, either $\mathcal{F}_{u \leq e}$ or $\mathcal{F}_{v \leq e}$ is multiplied by -1), is defined as

$\delta(x)_e = \mathcal{F}_{u \leq e} x_u - \mathcal{F}_{v \leq e} x_v$. From the coboundary operator, one can define the *sheaf Laplacian* as $L^{\mathcal{F}} = \delta^T \delta$ which, for a given $x \in C^0(G; \mathcal{F})$, reads:

$$L^{\mathcal{F}}(x)_u = \sum_{e=\{u,v\}} \mathcal{F}_{u \leq e}^T (\mathcal{F}_{u \leq e} x_u - \mathcal{F}_{v \leq e} x_v) \quad \forall u \in V.$$

Both $L^{\mathcal{F}}$ and its normalized version $L_N^{\mathcal{F}}$ are positive semidefinite operators on the space of 0-cochains $C^0(G; \mathcal{F})$, and are independent of the chosen edge orientation, mirroring a similar property that holds for the standard graph Laplacian L [9].

Several approaches have explored the use of sheaves in the context of graph-based learning. The first SNN was introduced by [1], and later extended by [2], who proposed the Neural Sheaf Diffusion (NSD) model. More recent SNN models build upon the NSD framework, incorporating attention mechanisms [10], extending the architecture to hypergraph data [11], and introducing nonlinearities [12].

The SNN literature assumes that all node and edge stalks are finite-dimensional vector spaces of dimension d , all of which are isomorphic to \mathbb{R}^d . In this way, every restriction map coincides with a $d \times d$ matrix. As a result, the sheaf Laplacian is a block-matrix of size $nd \times nd$ with blocks of size $d \times d$ which operates on an nd -dimensional vector-valued signal obtained by stacking the d -dimensional node signals $x_u \in \mathcal{F}(u)$ for all $u \in V$ associated with the graph's vertices (the 0-cochain). When considering multi-feature vertex signals with $f > 1$ features (or channels), a SNN operates on a matrix-valued graph signal of size $nd \times f$. For any $u, v \in V$, the block of indices u, v of $L^{\mathcal{F}}$ is equal to the $d \times d$ matrix $-\mathcal{F}_{u \leq e}^T \mathcal{F}_{v \leq e}$. The *sheaf Laplacian* generalizes the classical graph Laplacian on an undirected and unweighted graph G . This is because, in the special case of a *trivial sheaf*—a sheaf where each stalk is isomorphic to \mathbb{R} and each restriction map is the identity map over \mathbb{R} —we recover the standard $n \times n$ graph Laplacian $L = D - A$, where $A \in \{0, 1\}^{n \times n}$ is the adjacency matrix, and $D := \text{diag}(\mathbf{1}_n^\top A)$ where $\mathbf{1}_n$ is the all-one vector.

To the best of our knowledge, no SNNs, including those introduced in the above-mentioned papers, have been proposed to incorporate the edge directions directly. We set out to do so in this paper.

2.2 Discrete Laplacian matrices for undirected and directed graphs

In the literature, GNNs are typically classified into two categories: spectral-based and spatial-based [13]. Spatial-based GNNs define the convolution as a localized-aggregation/message-passing operator [14]. For example, GatedGCN [15] handles directed graphs by aggregating information from out-neighbors (ignoring, though, potentially valuable signals from in-neighbors) and, more recently, Dir-GNN [16] employs separate aggregation schemes with distinct weights for in-neighbors and out-neighbors. In contrast, spectral-based GNNs define the convolution operator rigorously as a function of the eigenvalue decomposition of the graph Laplacian [17]. Over the past few years, several approaches have been proposed to generalize spectral convolutions to directed graphs. In particular, DGCN [18] introduces a first-order proximity matrix along with two second-order proximity matrices to describe both the neighborhood of each vertex and the vertices that are reachable from a given vertex in one hop. DiGCN [19] adopts the Personalized PageRank matrix and incorporates k -hop diffusion matrices. Finally, several methods generalized the classical Laplacian matrix L to suitably defined complex-valued, Hermitian matrices such as the Magnetic Laplacian [20] and the Sign-Magnetic Laplacian [21].

The *Magnetic Laplacian* $L^{(q)}$, originally introduced by [20] in the study of electromagnetic fields and later employed in spectral GNNs by [3, 22], is a complex-valued Hermitian matrix that captures directional information in graphs while admitting an eigenvalue decomposition with a real, nonnegative spectrum. Letting $A_s := \frac{1}{2} (A + A^\top)$ be the symmetrized version of A and letting $D_s := \text{diag}(\mathbf{1}_n^\top A_s)$, the *Magnetic Laplacian* and its normalized version are defined as follows:

$$L^{(q)} := D_s - H^{(q)} \text{ and } L_N^{(q)} := I - D_s^{-\frac{1}{2}} H^{(q)} D_s^{-\frac{1}{2}}, \text{ with } H^{(q)} := A_s \odot \exp(\mathbf{i} 2\pi q (A - A^\top)),$$

where \mathbf{i} is the imaginary unit and $q \in [0, 1]$.

The *Sign-Magnetic Laplacian* L^σ , introduced by [21], is a Hermitian matrix that is well-defined even for graphs with negative edge weights and possesses several additional desirable properties. When $q = \frac{1}{4}$, L^σ and $L^{(q)}$ coincide if the latter is first computed on the unweighted version of the graph and

then element-wise multiplied by A_s . Thus, L^σ is invariant to a positive weight scaling which could otherwise alter the sign pattern of $L^{(q)}$ and, thus, the edge direction. Letting $\bar{D}_s := \text{diag}(\mathbf{1}_n^\top |A_s|)$ and $\text{sgn} : \mathbb{R}^{n \times n} \rightarrow \{-1, 0, 1\}^{n \times n}$ be the component-wise *signum* function, L^σ and its normalized version are defined as follows:

$$L^\sigma := \bar{D}_s - H^\sigma \text{ and } L_N^\sigma := I - \bar{D}_s^{-\frac{1}{2}} H^\sigma \bar{D}_s^{-\frac{1}{2}}, \text{ with } H^\sigma := A_s \odot (e^\top - \text{sgn}(|A - A^\top|) + i \text{sgn}(|A| - |A^\top|)).$$

3 Directed Cellular Sheaf, Directed Sheaf Laplacian and Its Properties

In this paper, we introduce the notion of a *Directed Cellular Sheaf*, a special type of cellular sheaf where the node and edge stalks are vector spaces defined over the complex field and in which, assuming finite-dimensional vector spaces, the restriction maps are either real-valued or complex-valued matrices where the latter encode the graph's direction.

3.1 Directed Cellular Sheaf

We introduce the Directed Cellular Sheaf for the case of finite-dimensional stalks. This is done solely for the ease of notation, as that definition can be directly extended to the infinite-dimensional case.

Definition 1. The *Directed Cellular Sheaf* of a directed graph $G = (V, E)$ with adjacency matrix $A \in \{0, 1\}^{n \times n}$ is the tuple $(T^{(q)}, \{\tilde{\mathcal{F}}(u)\}_{u \in V}, \{\tilde{\mathcal{F}}(e)\}_{e \in E}, \{\tilde{\mathcal{F}}_{u \leq e}\}_{e \in \Gamma(u)})$ consisting of:

1. A directional and topological Hermitian matrix $T^{(q)} := \exp(i 2\pi q (A - A^\top))$, with $q \in \mathbb{R}$.
2. A vector space $\tilde{\mathcal{F}}(u) \in \mathbb{C}^d$ associated with each vertex $u \in V$;
3. A vector space $\tilde{\mathcal{F}}(e) \in \mathbb{C}^d$ associated with each edge $e \in E$;
4. Two linear maps $\tilde{\mathcal{F}}_{u \leq e}, \tilde{\mathcal{F}}_{v \leq e}$ that map $\tilde{\mathcal{F}}(u), \tilde{\mathcal{F}}(v)$ to $\tilde{\mathcal{F}}(e)$ for each edge $e \in E$ with $u \sim_e v$ where $\tilde{\mathcal{F}}_{u \leq e} \in \mathbb{R}^{d \times d}$ and $\tilde{\mathcal{F}}_{v \leq e} = \tilde{\mathcal{F}}_{u \leq e}^0 T_{uv}^{(q)} \in \mathbb{C}^{d \times d}$, with $\tilde{\mathcal{F}}_{u \leq e}^0 \in \mathbb{R}^{d \times d}$,

where $u \sim_e v$ indicates that e is incident to both u and v regardless of whether it is directed or not. An illustration is provided in Figure 1.

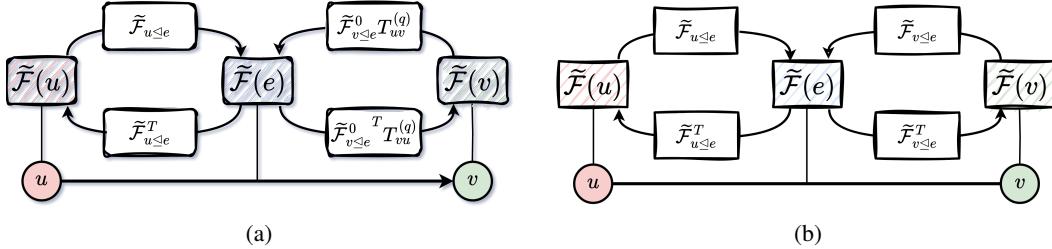


Figure 1: An illustration of the complex-valued restriction maps of the *Directed Cellular Sheaf* showing how they encode the graph's directionality for (a) a directed edge and (b) an undirected edge.

The rationale of such a definition is to encode the direction of each edge in the imaginary part of the restriction map of the tail node in the following sense. *i*) In the undirected case where $e = \{u, v\}$, $A_{uv} = A_{vu} = 1$, $T_{uv}^{(q)} = \cos(0) + i \sin(0) = 1$ and, thus, $\tilde{\mathcal{F}}_{v \leq e} = \tilde{\mathcal{F}}_{u \leq e}^0$. *ii*) Assume $q = \frac{1}{4}$; in the directed case where $e = (u, v)$, $A_{uv} = 1$ and $A_{vu} = 0$; we have $T_{uv}^{(q)} = \cos(-\pi \frac{1}{2}) + i \sin(-\pi \frac{1}{2}) = -i$, and thus $\tilde{\mathcal{F}}_{v \leq e} = -\tilde{\mathcal{F}}_{u \leq e}^0 i$. Notice that our proposed Directed Cellular Sheaf generalizes the classical Cellular Sheaf since, if G is an undirected graph, $T_{vu}^{(q)} = 1$ for all $\{u, v\} \in E$ for any choice of q and, thus, the two coincide. If G is directed, but we set $q = 0$, we obtain the classical Cellular Sheaf associated with the undirected version of G .

Let $E^0 \cup E^1 = E$ be a partition of the edge set E into undirected edges (E^0) and directed edges (E^1). We define the *Directed Coboundary Operator* $\tilde{\delta}$ associated with the Directed Cellular Sheaf

as $\tilde{\delta}(x)_e = \tilde{\mathcal{F}}_{u \leq e} x_u - \tilde{\mathcal{F}}_{v \leq e} x_v$ for all $e \in E$, where x is a co-chain of the Directed Cellular Sheaf. Thanks to our definition of $\tilde{\mathcal{F}}_{u \leq e}, \tilde{\mathcal{F}}_{v \leq e}$, we have:

$$\tilde{\delta}(x)_e = \begin{cases} \tilde{\mathcal{F}}_{u \leq e} x_u - \tilde{\mathcal{F}}_{v \leq e} x_v & \text{if } e \in E^0 \\ \tilde{\mathcal{F}}_{u \leq e} x_u - \tilde{\mathcal{F}}_{v \leq e}^0 T_{uv}^{(q)} x_v & \text{if } e \in E^1. \end{cases} \quad (1)$$

We define the *Directed Sheaf Laplacian* (DSL) $L^{\tilde{\mathcal{F}}}$ associated with a Directed Cellular Sheaf as $L^{\tilde{\mathcal{F}}} := \tilde{\delta}^* \tilde{\delta}$, where $*$ is the conjugate transpose operator. Each $d \times d$ block of $L^{\tilde{\mathcal{F}}}$ of index $u, v \in V$ is:

$$L_{uv}^{\tilde{\mathcal{F}}} = \begin{cases} -\tilde{\mathcal{F}}_{u \leq e}^* \tilde{\mathcal{F}}_{v \leq e} = -\tilde{\mathcal{F}}_{u \leq e}^T \tilde{\mathcal{F}}_{v \leq e}^0 T_{uv}^{(q)} & \text{if } e = (u, v) \\ -\tilde{\mathcal{F}}_{u \leq e}^* \tilde{\mathcal{F}}_{v \leq e} = -(\tilde{\mathcal{F}}_{u \leq e}^0 T_{vu}^{(q)})^* \tilde{\mathcal{F}}_{v \leq e} & \text{if } e = (v, u) \\ -\tilde{\mathcal{F}}_{u \leq e}^* \tilde{\mathcal{F}}_{v \leq e} = -\tilde{\mathcal{F}}_{u \leq e}^T \tilde{\mathcal{F}}_{v \leq e} & \text{if } e = \{u, v\} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$L_{uu}^{\tilde{\mathcal{F}}} = \sum_{e \in \Gamma(u)} \tilde{\mathcal{F}}_{u \leq e}^* \tilde{\mathcal{F}}_{u \leq e}, \quad (3)$$

where $\Gamma(u)$ is the set of edges incident to u regardless of their direction. Notice that, since $(T_{vu}^{(q)})^* = -T_{uv}^{(q)}$, for a directed edge $e = (u, v)$ or $e = (v, u)$, $L_{uv}^{\tilde{\mathcal{F}}}$ and $L_{vu}^{\tilde{\mathcal{F}}}$ only differ by the sign of their imaginary part.

As one can see (the full derivation is reported in the appendix), when applied to a 0-cochain x , the Directed Sheaf Laplacian operator reads as follows for each $u \in V$:

$$L^{\tilde{\mathcal{F}}}(x)_u = \underbrace{\sum_{e=(v,u) \in E} (\tilde{\mathcal{F}}_{u \leq e}^0 T_{vu}^{(q)})^* (\tilde{\mathcal{F}}_{u \leq e} x_u - \tilde{\mathcal{F}}_{v \leq e} x_v)}_{\text{inflow}} + \underbrace{\sum_{e=(u,v) \in E} \tilde{\mathcal{F}}_{u \leq e}^T (\tilde{\mathcal{F}}_{u \leq e} x_u - \tilde{\mathcal{F}}_{v \leq e}^0 T_{uv}^{(q)} x_v)}_{\text{outflow}} + \underbrace{\sum_{e=\{u,v\} \in E} \tilde{\mathcal{F}}_{u \leq e}^T (\tilde{\mathcal{F}}_{u \leq e} x_u - \tilde{\mathcal{F}}_{v \leq e} x_v)}_{\text{undirected}}. \quad (4)$$

We define the *normalized Directed Sheaf Laplacian* as:

$$L_N^{\tilde{\mathcal{F}}} := \tilde{D}^{-\frac{1}{2}} L^{\tilde{\mathcal{F}}} \tilde{D}^{-\frac{1}{2}}, \quad (5)$$

where $\tilde{D} := \text{diag}(\tilde{D}_1, \tilde{D}_2, \dots, \tilde{D}_n)$ and, for all $u \in V$, $\tilde{D}_u := \sum_{e \in \Gamma(u)} \tilde{\mathcal{F}}_{u \leq e}^* \tilde{\mathcal{F}}_{u \leq e}$.

3.2 Spectral properties of the Directed Sheaf Laplacian

The Directed Sheaf Laplacian enjoys several key spectral properties, which we now illustrate. The proofs of the theorems of this section and the next can be found in Appendix C. First, we show that both $L^{\tilde{\mathcal{F}}}$ and $L_N^{\tilde{\mathcal{F}}}$ are diagonalizable with a real spectrum and that their spectra are nonnegative:

Theorem 1. $L^{\tilde{\mathcal{F}}}$ is Hermitian and $L^{\tilde{\mathcal{F}}} \succeq 0$, and the same holds for $L_N^{\tilde{\mathcal{F}}}$.

Next, we show that the spectrum of the Normalized Sheaf Laplacian is upper-bounded by 2:

Theorem 2. $L_N^{\tilde{\mathcal{F}}} \preceq 2I$.

These theorems show that $L^{\tilde{\mathcal{F}}}$ and $L_N^{\tilde{\mathcal{F}}}$ enjoy the same spectral properties as the classical Laplacian matrix L defined for undirected graphs. These are the essential to define a principled convolutional operator which coincides with approximating the graph-Fourier transform of a graph signal with Chebyshev polynomials of the first kind of degree 1, as proposed by [17] for the undirected case.

3.3 Generalization properties of the Directed Sheaf Laplacian

First, we show that the Directed Sheaf Laplacian generalizes both the Sheaf Laplacian and the classical graph Laplacian:

Theorem 3. *If G is undirected, $L^{\tilde{\mathcal{F}}}$ coincides with the classical sheaf Laplacian $L^{\mathcal{F}}$ for any choice of $q \in \mathbb{R}$. Also, if the sheaf is trivial and G is undirected and unweighted, $L^{\tilde{\mathcal{F}}}$ coincides with the classical graph Laplacian L . If G is directed and we set $q = 0$, $L^{\tilde{\mathcal{F}}}$ coincides with the classical sheaf Laplacian associated with the undirected version of G .*

Let a *Trivial Directed Cellular Sheaf* be any Directed Cellular Sheaf with $d = 1$ where, for all directed edges $e = (u, v)$, $\tilde{\mathcal{F}}_{u \leq e} = 1$ and $\tilde{\mathcal{F}}_{v \leq e} = T_{uv}^{(q)}$. With the next theorem, we show that, for a given directed graph without weights, $L^{\tilde{\mathcal{F}}}$ generalized the Magnetic Laplacian and, when choosing $q = \frac{1}{4}$, also the Sign-Magnetic Laplacian. The following holds:

Theorem 4. *Letting G be a directed graph with unit weights, the Directed Sheaf Laplacian $L^{\tilde{\mathcal{F}}}$ associated with a Trivial Directed Cellular Sheaf coincides with the Magnetic Laplacian $L^{(q)}$. In the special case where $q = \frac{1}{4}$, $L^{\tilde{\mathcal{F}}}$ also coincides with the Sign-Magnetic Laplacian L^σ .*

It is well-known that the classical Laplacian matrix L defined for an undirected graph can be equivalently defined as $L = D - A$ or $L = BB^T$, where $B \in \{-1, 0, 1\}^{n \times m}$ is the node-to-edge incidence matrix of the graph in which either of the two entries of each column has been arbitrarily multiplied by -1 . While, to the best of our knowledge, no similar construction is known for the Magnetic Laplacian and the Sign-Magnetic Laplacian, with the following theorem, we show that one such decomposition exists and can be obtained via the lens of our Directed Sheaf Laplacian, thanks to its generality. Indeed, we have the following:

Theorem 5. *Let G be a directed graph with unit weights. Assuming a Trivial Directed Cellular Sheaf, the conjugate transpose $\tilde{\delta}^*$ of the Directed Coboundary Operator $\tilde{\delta}$ boils down to the complex-valued node-to-edge incidence matrix $\hat{B} \in \mathbb{C}^{n \times m}$ defined for an edge $e \in E$ incident to a vertex u :*

$$\hat{B}_{ue} = \begin{cases} 1 & \text{if } e = (u, v) \text{ or } e = \{u, v\} \text{ with } u < v \\ -1 & \text{if } e = \{u, v\} \text{ with } u > v \\ -T_{uv}^{(q)} & \text{if } e = (v, u). \end{cases}$$

It follows that $L^{(q)} = \hat{B}\hat{B}^*$. With $q = \frac{1}{4}$, $L^{(\frac{1}{4})} = L^\sigma = \hat{B}\hat{B}^*$.

Incidentally, this result also allows to obtain substantially simpler proofs of the positive semidefiniteness of both Laplacian matrices than those reported in their original papers.

4 The Directed Sheaf Neural Network (DSNN)

The *sheaf diffusion* process on a graph G is introduced in [23] as a generalization of the classical heat diffusion process that governs classical spectral-based GNNs [17]. It follows the differential equation

$$\dot{X}(t) = -L_N^{\mathcal{F}} X(t),$$

where $X(t)$ is a time-dependent graph signal X . More precisely, X_u is the stalk of each node $u \in V$, and it coincides with a matrix in $\mathbb{R}^{d \times f}$, where d denotes the dimensionality of the vertex stalk and f is the number of feature channels. X is typically obtained starting from a matrix of node features of size $n \times f$ to which one applies a linear projection to obtain an $n \times (df)$ matrix, which is then reshaped to $(nd) \times f$.

By relying on our proposed Directed Sheaf Laplacian $L^{\tilde{\mathcal{F}}}$, we introduce the *Directed Neural Sheaf Diffusion* process as the following generalization of the Neural Sheaf Diffusion process proposed by [2]:

$$\dot{X}(t) = -\sigma \left(L_N^{\tilde{\mathcal{F}}}(t) \left(I_n \otimes W_1(t) \right) X(t) W_2(t) \right), \quad (6)$$

where $W_1 \in \mathbb{R}^{d \times d}$, $W_2 \in \mathbb{R}^{f \times f}$ are two time-dependent weight matrices and σ is a nonlinear activation function.

We then define the Directed Sheaf Neural Network (DSNN) as the convolutional neural network whose convolution operator is obtained from the discretized version of Equation 6, which leads to:

$$X^{(t+1)} = X^{(t)} - \sigma \left(L_N^{\tilde{\mathcal{F}}(t)} \left(I_n \otimes W_1^{(t)} \right) X^{(t)} W_2^{(t)} \right), \quad (7)$$

where $X^{(t)}, X^{(t+1)} \in \mathbb{C}^{nd \times f}$.

The expressiveness of Equation (7) is further enhanced by learning a parameter $\epsilon \in [-1, 1]^d$ that allows the model to adjust the relative magnitude of the features in each stalk dimension. This gives the update rule as:

$$X^{(t+1)} = \text{diag}(1 + \varepsilon)X^{(t)} - \sigma \left(L_N^{\tilde{\mathcal{F}}(t)} \left(I_n \otimes W_1^{(t)} \right) X^{(t)} W_2^{(t)} \right), \quad (8)$$

where $\varepsilon \in [-1, 1]^{nd}$ is obtained by concatenating ϵ n times. As activation function σ , we adopt a complex extension of the *ReLU* function, defined for a given $z \in \mathbb{C}$, as

$$\sigma(z) = \begin{cases} z & \text{if } \Re(z) \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

This choice is consistent with previous work on complex-valued GNNs and HNNs, such as [3, 24].

Finally, since our model operates in the complex domain, we project the output of the final layer to the real domain using an *unwind* operation. Given $X^{(\tau)} \in \mathbb{C}^{n \times c}$, the projection is defined as:

$$\text{unwind}(X(\tau)) = (\Re(X(\tau)) \parallel \Im(X(\tau))) \in \mathbb{R}^{n \times 2c},$$

where τ is the last convolutional layer of the network, \parallel denotes concatenation along the feature dimension, and c is the output dimension.

Learnable Sheaf Laplacian. A key strength of SSNs is their ability to operate over richer structures—sheaves—rather than just the underlying graph. Since multiple sheaf structures can be associated with the same graph, effectively modeling the most suitable one is critical for meaningful representation learning. In our proposed models, the restriction maps are learned end-to-end as a function of the input vertex features. Specifically, for each edge $e \in E$ with endpoints $u, v \in V$, each $d \times d$ matrix $\mathcal{F}_{u \leq e}$ is parameterized as $\mathcal{F}_{u \leq e} = \Phi(x_u \parallel x_v)$, where x_v and x_u denote the feature vectors of the nodes incident to e . The resulting vector is reshaped into a $d \times d$ matrix, thus obtaining the linear restriction map $\mathcal{F}_{u \leq e}$.

Connection with Neural Sheaf Diffusion. The Neural Sheaf Diffusion process proposed by [2] relies on the *Normalized Sheaf Laplacian* $L_N^{\mathcal{F}}$ instead of our proposed *Directed Sheaf Laplacian* $L_N^{\tilde{\mathcal{F}}}$ in Equation (7). Since, as shown in Theorem 3, $L_N^{\mathcal{F}} = L_N^{\tilde{\mathcal{F}}}$ when the graph is undirected, NSD is a special case of SNN when the graph is undirected.

Computational Complexity. Let f be the number of channels, assumed constant throughout the layers, and let's focus on a single convolutional layer. In the case of an undirected graph, where all restriction maps are real-valued, the complexity of DSNN is identical to the complexity of NSD, and reads $O(n(c^2 + d^3) + m(cd^2 + d^3))$, which, with $d = 1$, coincides with the complexity of a classical spectral-based GNNs [17], which is $\mathcal{O}(nc^2 + mc)$. In the experiments, we use $d \in \{2, 5\}$, which only introduced a small, constant overhead with no asymptotic impact. For a directed graph, the restriction maps are complex-valued, and thus, the stalks are complex-valued from layer 2 onward. This, though, only leads to an extra multiplicative cost of about 4, which is independent of the graph and size of the network and plays no role in the complexity of DSNN.

For the proof of the theorems in this section, and for additional details on DSNN's inference complexity, please refer to Appendix C and Appendix D, respectively.

5 Experiments

We evaluate DSNN against state-of-the-art methods on a diverse set of benchmark datasets spanning from real-world (Section 5.1) to synthetic dataset (Section 5.2). Following [2], we experiment with three types of $d \times d$ blocks in the Directed Sheaf Laplacian $L_N^{\tilde{\mathcal{F}}}$, *diagonal*, *orthogonal*, and *general* matrices, which lead to three variants of DSNN: Diag-DSNN, O(d)-DSNN, Gen-DSNN. The tables in this section report the best results in boldface, and the second-best results are underlined. The datasets and code we used are available on GitHub (see Appendix A). Further details on our experiments are reported in Appendix E, F.

Table 1: Mean and standard deviation of the accuracy on a collection of real-world graph benchmarks on the node classification task.

Model	Texas	Wisconsin	Film	Squirrel	Chameleon	Cornell	Citeseer	Pubmed	Cora
Homophily Level	0.11	0.21	0.22	0.22	0.23	0.30	0.74	0.80	0.81
# Nodes	183	251	7,600	2,223	890	183	3,327	18,717	2,708
# Edges	295	466	26,752	46,998	8,854	280	4,676	44,327	5,278
# Classes	5	5	5	5	5	5	6	3	7
Diag-DSNN	88.65±4.95	90.20±4.02	38.34±1.01	45.37±2.21	46.84±4.03	87.84±5.70	79.80±1.49	90.23±0.44	87.36±1.41
O(d)-DSNN	87.57±4.04	89.80±3.82	37.37±0.98	44.54±2.26	45.36±3.29	87.30±7.26	77.28±1.63	90.05±0.55	87.30±1.62
Gen-DSNN	87.57±5.43	89.22±3.31	38.40±0.75	45.34±1.69	47.16±3.54	87.84±6.86	79.88±1.21	90.17±0.44	87.58±0.72
Diag-NSD-Comp	86.49±5.35	89.01±4.81	37.84±1.04	45.61±1.62	46.47±2.83	83.51±6.56	77.20±1.43	89.74±0.46	87.65±1.04
O(d)-NSD-Comp	87.29±5.54	89.21±4.57	37.06±0.96	41.51±2.01	45.56±4.18	84.86±0.53	76.95±1.67	88.65±0.49	87.23±1.85
Gen-NSD-Comp	87.56±3.86	89.04±2.93	37.72±1.17	45.13±2.19	47.01±2.55	86.49±6.84	77.02±1.75	89.68±0.31	87.82±0.84
Diag-NSD	85.67±6.95	88.63±2.75	37.79±1.01	45.52±2.22	46.55±3.03	86.49±7.35	77.14±1.85	89.42±0.43	87.14±1.06
O(d)-NSD	85.95±5.51	89.41±4.74	37.81±1.15	45.59±2.23	46.26±3.11	84.86±4.71	76.70±1.57	89.49±0.40	86.90±1.13
Gen-NSD	82.97±5.13	89.21±3.84	37.80±1.22	45.31±2.05	45.60±3.36	85.68±6.51	76.32±1.65	89.33±0.35	87.30±1.15
DirGnn	74.22±3.97	71.37±6.57	29.30±1.22	44.48±1.94	45.56±3.36	61.46±3.63	76.09±1.53	85.14±0.44	86.20±1.18
SigMaNet	78.92±4.49	80.21±5.07	36.59±0.55	40.89±1.97	40.98±3.88	73.53±5.91	74.35±0.96	88.35±0.64	85.51±1.14
MagNet	79.46±8.13	81.18±2.80	36.51±0.96	41.04±1.84	43.82±4.56	75.99±5.59	77.21±1.69	88.47±0.54	86.32±1.39
GGCN	84.86±4.55	86.86±3.29	37.54±1.56	40.75±2.44	39.71±3.25	85.68±6.63	77.14±1.45	89.15±0.37	87.95±1.05
H2GCN	84.86±7.23	87.65±4.98	35.70±1.00	37.77±1.92	42.07±4.13	82.70±5.28	77.11±1.57	89.49±0.38	87.87±1.20
GPRGNN	78.38±4.36	82.94±4.21	34.63±1.22	36.62±2.28	40.67±2.89	80.27±8.11	77.13±1.67	87.54±0.38	87.95±1.18
FAGCN	82.43±6.89	82.94±7.95	34.87±1.25	41.08±2.27	41.90±2.72	79.19±9.79	77.10±1.81	90.21±0.36	88.17±1.24
MixHop	77.84±7.73	75.88±4.90	32.22±2.34	43.46±2.52	45.09±3.02	73.51±6.34	76.26±1.33	85.31±0.61	87.61±0.85
GCNII	77.57±3.83	80.39±3.40	37.44±1.30	42.22±2.13	43.76±2.49	77.86±3.79	77.33±1.48	90.15±0.43	88.37±1.25
GraphSAGE	82.43±6.14	81.18±5.56	34.23±0.99	39.22±1.20	41.67±2.52	75.95±5.01	76.04±1.30	88.45±0.50	86.90±1.04
GCN	55.14±5.16	51.76±3.06	27.32±1.10	39.47±1.47	40.89±4.12	60.54±5.30	76.50±1.36	88.42±0.50	86.98±1.27
GAT	52.16±6.63	49.41±4.09	27.44±0.89	35.62±2.06	39.21±3.08	61.89±5.05	76.55±1.23	87.30±1.10	86.33±0.48
MLP	80.81±4.75	85.29±3.31	36.53±0.70	40.45±1.41	42.79±3.80	81.89±6.40	74.02±1.90	87.16±0.37	75.69±2.00

5.1 Real World Experiments

We measure the performance of our proposed DSNN on the node classification task of predicting the class label of each node. The *Texas*, *Wisconsin*, *Cornell*, and *Film* datasets are taken from [25], while *Citeseer*, *PubMed*, and *Cora* are sourced from [26]. The *Squirrel* and *Chameleon* datasets come from [27]. Since it is known that GNN-type methods often suffer from poor performance on heterophilic datasets (datasets where neighboring nodes have, on average, different labels), it is crucial to assess the performance of one’s method also on graphs of this type. For this reason, the datasets we consider span a wide range of edge homophily coefficients, from as low as 0.11 (highly heterophilic) to as high as 0.81 (highly homophilic), allowing us to assess model performance across varying levels of structural homophily. Following [2], we split the data 10 times and report the mean accuracy and standard deviation. Each split allocates 48%/32%/20% of the nodes per class for training, validation, and testing, respectively.

As baselines, we use a large set of GNN and SNN models from five categories: *i*) classical GNN models: GCN [17], GAT [28], GraphSAGE [29]; *ii*) GNN models designed for heterophilic graphs: GGCN [26], Geom-GCN [25], H2GCN [30], GPRGNN [31], FAGCN [32], MixHop [33]; *iii*) GNN models that address the oversmoothing problem: GCNII [34]; *iv*) GNN models that incorporate edge directionality: MagNet [3], SigMaNet [21], DirGNN [16]; *v*) SNN models: NSD [2], and NSD-Compl (a variant of NSD employing complex restriction maps which we introduce in this paper for comparison purposes), with the same Diag-, O(d)-, Gen- variants as DSNN.

The results are reported in Table 1. DSNN consistently outperforms the second-best baseline on 7 out of 9 datasets (also in highly heterophilic cases), obtaining an overall improvement of 1.29% with the largest improvement of 3.46% on *Citeseer*. When compared to SNN methods (NSD and NSD-Compl, which do not incorporate a notion of direction), the directional component of DSNN leads to a strictly stronger performance on 7 datasets out of 9. Compared with GNN methods which incorporate a notion of direction (DirGNN, SigMaNet and MagNet), the directional component of DSNN leads to a better performance across all the 9 dataset, with an average improvement of approximately 6% and a largest gain, obtained on *Cornell*, of 15.59%. These results show that, by relying on both the expressivity of a cellular sheaf as well as on a notion of directionality, DSNN achieves a more expressive propagation of directional signals. Thanks to this, it manages to outperform both SNN and GNN methods.

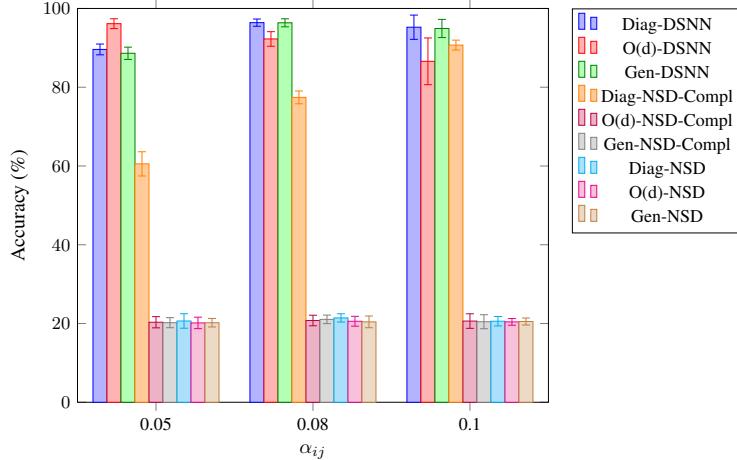


Figure 2: Mean and standard deviation of the accuracy on a collection of synthetic benchmarks on the node classification task as a function of the parameter α_{ij} used during graph generation.

5.2 Synthetic Datasets

To further investigate the role of directionality in the context of Sheaf Neural Networks, we compare DSNN against NSD and NSD-Compl on a set of synthetic graphs generated using the Direct Stochastic Block Model (DSBM). These datasets are generated by varying: *i*) the number of nodes n ; *ii*) the number of clusters C ; *iii*) the probability α_{ij} to create an undirected edge between nodes i and j belonging to different clusters; *iv*) the probability α_{ii} to create an undirected edge between two nodes in the same cluster, and *v*) the probability β_{ij} of an edge taking a certain direction. For this experiment, the DBSM datasets are generated with $n = 2500$, $C = 5$, $\alpha_{ii} = 0.1$, $\beta_{ij} = 0.2$, with an increasing inter-cluster density $\alpha_{ij} \in \{0.05, 0.08, 0.1\}$. We employ unit-dimensional feature vectors defined as each node’s sum of in- and out-degrees. We run the experiments 10 times per dataset with a 80%/5%/15% training/validation/testing split, and report the mean accuracy and standard deviation.

As shown in Figure 2, the three variants of DSNN—diagonal, O(d), and general—which rely on the Directed Sheaf Laplacian substantially outperform both NSD and NSD-Compl. The three variants of DSNN consistently achieve an average performance between 86% and 96% across all three settings, one per value of α_{ij} , whereas all the other baselines except for one (Diag-NSD-Compl) achieve an average accuracy of 20% which, due to having $C = 5$ classes, is as poor as the accuracy of trivial uniform predictor. The better performance achieved by Diag-NSD-Compl over the other baselines (but not DSNN) can be attributed to its more limited number of trainable parameters compared to the other NSD-Compl variants, which facilitates convergence and enables the model to partially capture some directional interactions in its complex-valued (albeit classical) Sheaf Laplacian. Nonetheless, its performance remains substantially lower than DSNN’s by 29% on average. These results underscore DSNN’s ability to leverage edge directionality. They further substantiate that the way we defined our complex-valued Laplacian matrix by embedding topological information via the imaginary part of the restriction maps leads to a substantially better performance than what one would obtain by simply learning complex-valued restriction maps end-to-end without relying on an inductive bias coming from the graph’s topology (which is what is done in NSD-Compl).

6 Conclusion

Summary and findings. We introduced the Directed Cellular Sheaf, from which we derived the Directed Sheaf Laplacian $L^{\bar{\mathcal{F}}}$. By encoding the edge direction in its imaginary components, $L^{\bar{\mathcal{F}}}$ carries a directional inductive bias thanks to which we obtain a convolution operator implementing a message-passing scheme capable of handling asymmetric interactions. We embedded such an operator in the Directed Sheaf Neural Network (DSNN). Our theoretical results have shown that DSNN generalizes several well-established graph-learning models, including NSD, MagNet, and SigMaNet. Empirically, DSNN exhibits strong performance across both real-world and synthetic datasets,

consistently outperforming both traditional GNNs and SNNs. This demonstrates that DSNN’s explicit treatment of directionality leads to superior generalization, particularly in heterophilic graph settings.

Limitations and future works. Further work may include considering node-varying stalk sizes as well as an extended assessment of DSNN on datasets and tasks arising from various real-world applications, e.g., life science and healthcare. Further work may also address low-level optimization aspects of our pipeline to potentially reduce its computational footprint, allowing for its adoption in low-power and low-compute settings.

References

- [1] Jakob Hansen and Robert Ghrist. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology*, 3(4):315–358, 2019.
- [2] Cristian Bodnar, Francesco Di Giovanni, Benjamin Chamberlain, Pietro Lio, 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.
- [3] Xitong Zhang, Yixuan He, Nathan Brugnone, Michael Perlmutter, and Matthew Hirn. Magnet: A neural network for directed graphs, 2021.
- [4] Ginestra Bianconi, Natali Gulbahce, and Adilson E Motter. Local structure of directed networks. *Physical review letters*, 100(11):118701, 2008.
- [5] Uzi Harush and Baruch Barzel. Dynamic patterns of information flow in complex networks. *Nature communications*, 8(1):2181, 2017.
- [6] Malbor Asllani, Renaud Lambiotte, and Timoteo Carletti. Structure and dynamical behavior of non-normal networks. *Science advances*, 4(12):eaau9403, 2018.
- [7] Allen Dudley Shepard. *A Cellular Description of the Derived Category of a Stratified Space*. PhD thesis, Brown University, 1985.
- [8] Justin Michael Curry. *Sheaves, Cosheaves and Applications*. PhD thesis, University of Pennsylvania, 2014.
- [9] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.
- [10] Federico Barbero, Cristian Bodnar, Haitz Sáez de Ocáriz Borde, and Pietro Lio. Sheaf attention networks. In *NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations*, 2022.
- [11] Iulia Duta, Giulia Cassarà, Fabrizio Silvestri, and Pietro Liò. Sheaf hypergraph networks. *Advances in Neural Information Processing Systems*, 36:12087–12099, 2023.
- [12] Olga Zagheni, Antonio Longa, Steve Azzolini, Lev Telyatnikov, Andrea Passerini, et al. Sheaf diffusion goes nonlinear: Enhancing gnns with adaptive sheaf laplacians. *PROCEEDINGS OF MACHINE LEARNING RESEARCH*, 251, 2024.
- [13] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2020.
- [14] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.
- [15] Yujia Li, Richard Zemel, Marc Brockschmidt, and Daniel Tarlow. Gated graph sequence neural networks. In *Proceedings of ICLR’16*, April 2016.
- [16] Emanuele Rossi, Bertrand Charpentier, Francesco Di Giovanni, Fabrizio Frasca, Stephan Günnemann, and Michael M Bronstein. Edge directionality improves learning on heterophilic graphs. In *Learning on graphs conference*, pages 25–1. PMLR, 2024.

- [17] Thomas. N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2017.
- [18] Zekun Tong, Yuxuan Liang, Changsheng Sun, David S. Rosenblum, and Andrew Lim. Directed graph convolutional network, 2020.
- [19] Zekun Tong, Yuxuan Liang, Changsheng Sun, Xinkie Li, David S. Rosenblum, and Andrew Lim. Digraph inception convolutional networks. *Advances in Neural Information Processing Systems*, 2020-December(NeurIPS):1–12, 2020.
- [20] Elliott H Lieb and Michael Loss. Fluxes, Laplacians, and Kasteleyn’s theorem. In *Statistical Mechanics*, pages 457–483. Springer, 1993.
- [21] Stefano Fiorini, Stefano Coniglio, Michele Ciavotta, and Enza Messina. Sigmanet: One laplacian to rule them all. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7568–7576, 2023.
- [22] Jie Zhang, Bo Hui, Po-Wei Harn, Min-Te Sun, and Wei-Shinn Ku. smgc: A complex-valued graph convolutional network via magnetic laplacian for directed graphs, 2021.
- [23] Jakob Hansen and Thomas Gebhart. Sheaf neural networks. *arXiv preprint arXiv:2012.06333*, 2020.
- [24] Stefano Fiorini, Stefano Coniglio, Michele Ciavotta, and Alessio Del Bue. Let there be direction in hypergraph neural networks. *Transactions on Machine Learning Research*, 2024.
- [25] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*, 2020.
- [26] Yujun Yan, Milad Hashemi, Kevin Swersky, Yaoqing Yang, and Danai Koutra. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 1287–1292. IEEE, 2022.
- [27] Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress? In *The Eleventh International Conference on Learning Representations*, 2023.
- [28] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.
- [29] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30:1–11, 2017.
- [30] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in neural information processing systems*, 33:7793–7804, 2020.
- [31] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *International Conference on Learning Representations*, 2021.
- [32] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 3950–3957, 2021.
- [33] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *international conference on machine learning*, pages 21–29. PMLR, 2019.
- [34] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *International conference on machine learning*, pages 1725–1735. PMLR, 2020.

A Licensing

The Cora, Citeseer, and PubMed datasets are available at <https://linqs.org/datasets/>. Citeseer, PubMed, and Cora are sourced from [26]. The Squirrel and Chameleon datasets come from [27].

Regarding the models used in this paper, we rely on publicly available implementations from the following sources:

- **MLP, GCN, GAT, GGCN, GCNII, Geom-GCN, GPRGNN:** https://github.com/Yujun-Yan/Heterophily_and_oversmoothing with MIT license.
- **GraphSAGE:** https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.SAGEConv.html with MIT license.
- **H2GCN:** <https://github.com/Godofnothing/HeterophilySpecificModels/tree/main/H2GCN>.
- **FAGCN:** <https://github.com/Godofnothing/HeterophilySpecificModels/tree/main/FAGCN>.
- **MixHop:** <https://github.com/benedekrozemberczki/MixHop-and-N-GCN> with GNU General Public License v3.0 (GPL-3.0) license.
- **MagNet:** <https://github.com/matthew-hirn/magnet> with Apache License 2.0.
- **SigMaNet:** <https://github.com/Stefa1994/SigMaNet> with Apache License 2.0.
- **DirGNN:** <https://github.com/emalgorith/directed-graph-neural-network> with Apache License 2.0.
- **NSD:** <https://github.com/twitter-research/neural-sheaf-diffusion> with Apache License 2.0.

B Derivation of the equation of $L^{\tilde{\mathcal{F}}}$

Since, by construction, $L^{\tilde{\mathcal{F}}} = \tilde{\delta}^* \tilde{\delta}$, the following equation holds:

$$\begin{aligned} L^{\tilde{\mathcal{F}}}(x)_u &= \underbrace{\sum_{e \in \Gamma(u)} \tilde{\mathcal{F}}_{u \leq e}^* \tilde{\mathcal{F}}_{u \leq e} x_u}_{\text{self-loop}} - \underbrace{\sum_{e=(v,u) \in E} (\mathcal{F}_{u \leq e}^0 T_{vu}^{(q)})^* \mathcal{F}_{v \leq e} x_v}_{\text{inflow}} \\ &\quad - \underbrace{\sum_{e=(u,v) \in E} \mathcal{F}_{u \leq e}^T \mathcal{F}_{v \leq e}^0 T_{uv}^{(q)} x_v}_{\text{outflow}} - \underbrace{\sum_{e=\{u,v\} \in E} \mathcal{F}_{u \leq e}^T \mathcal{F}_{v \leq e} x_v}_{\text{undirected}}. \end{aligned}$$

For a directed graph G , the uu component of $L^{\tilde{\mathcal{F}}}$ can be rewritten as follows:

$$\begin{aligned} L_{uu}^{\tilde{\mathcal{F}}} &= \sum_{e \in \Gamma(u)} \tilde{\mathcal{F}}_{u \leq e}^* \tilde{\mathcal{F}}_{u \leq e} = \sum_{e=(u,v) \in E} \tilde{\mathcal{F}}_{u \leq e}^T \tilde{\mathcal{F}}_{u \leq e} \\ &\quad + \sum_{e=(v,u) \in E} (\tilde{\mathcal{F}}_{u \leq e}^0)^T (T_{vu}^{(q)})^* \tilde{\mathcal{F}}_{u \leq e}^0 T_{vu}^{(q)} + \sum_{e=\{u,v\} \in E} \tilde{\mathcal{F}}_{u \leq e}^T \tilde{\mathcal{F}}_{u \leq e}. \end{aligned}$$

Equation (4) is obtained by combining this equation with the previous one and factoring each summation by $(\tilde{\mathcal{F}}_{u \leq e})^T$.

C Proofs of our theorems

Theorem 1. $L^{\tilde{\mathcal{F}}}$ is Hermitian and $L^{\tilde{\mathcal{F}}} \succeq 0$, and the same holds for $L_N^{\tilde{\mathcal{F}}}$.

Proof. By definition, we have $L^{\tilde{\mathcal{F}}} := \tilde{\delta}^* \tilde{\delta}$. Therefore, for any pair of indices $u, v \in V$, $L_{uv}^{\tilde{\mathcal{F}}} = \tilde{\delta}_{\bullet u}^* \tilde{\delta}_{\bullet v}$ and $L_{vu}^{\tilde{\mathcal{F}}} = \tilde{\delta}_{\bullet v}^* \tilde{\delta}_{\bullet u}$. Since this implies $L_{uv}^{\tilde{\mathcal{F}}} = (L_{vu}^{\tilde{\mathcal{F}}})^*$ for all of its entries u, v , we conclude that the matrix is Hermitian. It follows that the spectrum of $L^{\tilde{\mathcal{F}}}$ is real. By following essentially the same arguments using $\tilde{\delta}' := \tilde{\delta} D^{-\frac{1}{2}}$, one can show that the spectrum of $L_N^{\tilde{\mathcal{F}}}$ is real as well.

Since, again by definition, $L^{\tilde{\mathcal{F}}} = \tilde{\delta}^* \tilde{\delta}$, its associated quadratic form $x^* \tilde{\delta}^* \tilde{\delta} x$ (with $x \in \mathbb{C}$) can be rewritten as $x^* \tilde{\delta}^* \tilde{\delta} x = (\tilde{\delta}x)^* (\tilde{\delta}x) = \|\tilde{\delta}x\|_2^2$. Since $\|\tilde{\delta}x\|_2^2$ is a norm, $\|\tilde{\delta}x\|_2^2 \geq 0$ holds for all $x \in \mathbb{C}$, thus implying $L^{\tilde{\mathcal{F}}} \succeq 0$ for all $x^* \in \mathbb{C}$. Thus, $L^{\tilde{\mathcal{F}}} \succeq 0$. By following the same arguments using $\tilde{\delta}' := \tilde{\delta} D^{-\frac{1}{2}}$, one can show that $L_N^{\tilde{\mathcal{F}}} \succeq 0$ as well. \square

Theorem 2. $L_N^{\tilde{\mathcal{F}}} \preceq 2I$.

Proof. Let $Q_N^{\tilde{\mathcal{F}}} := D^{-\frac{1}{2}} \tilde{\delta}^* \tilde{\delta} D^{-\frac{1}{2}}$ for the case where $\tilde{\delta}$ has *not* been given an arbitrary orientation (this is in line with the classical construction of the Signless Laplacian Q for undirected unweighted graphs). Since $Q_N^{\tilde{\mathcal{F}}}$ is the product of a matrix and its conjugate, we have $Q_N^{\tilde{\mathcal{F}}} \succeq 0$. It is easy to show that $Q_N^{\tilde{\mathcal{F}}} = 2I - \hat{L}_N^{\mathcal{F}}$. From this, we deduce:

$$Q_N^{\tilde{\mathcal{F}}} = 2I - L_N^{\tilde{\mathcal{F}}} \succeq 0 \Leftrightarrow -L_N^{\tilde{\mathcal{F}}} \succeq -2I \Leftrightarrow L_N^{\tilde{\mathcal{F}}} \preceq 2I.$$

This shows that not only $L_N^{\tilde{\mathcal{F}}}$ has a nonnegative spectrum, but also that its spectrum is upper-bounded by 2. \square

Theorem 3. *If G is undirected, $L^{\tilde{\mathcal{F}}}$ coincides with the classical sheaf Laplacian $L^{\mathcal{F}}$ for any choice of $q \in \mathbb{R}$. Also, if the sheaf is trivial and G is undirected and unweighted, $L^{\tilde{\mathcal{F}}}$ coincides with the classical graph Laplacian L . If G is directed and we set $q = 0$, $L^{\tilde{\mathcal{F}}}$ coincides with the classical sheaf Laplacian associated with the undirected version of G .*

Proof. **Part 1.** If G is undirected, all restriction maps of the Directed Cellular Sheaf are real for every choice of $q \in \mathbb{R}$ —this is because, for all $u, v \in V$, $A = A^T$ implies $\Re(T_{uv}^{(q)}) = 1$ and $\Im(T_{uv}^{(q)}) = 0$ for any choice of q . This implies $\Im(\tilde{\mathcal{F}}_{u \leq e}) = 0$ for all $e \in E$ where u is one of its endpoints; therefore, $L^{\tilde{\mathcal{F}}}$ is real valued and $L^{\tilde{\mathcal{F}}} = L^{\mathcal{F}}$.

Part 2. Under the same assumptions on G , if the Directed Cellular Sheaf is trivial, $d = 1$ and $\tilde{\mathcal{F}}_{u \leq e} = 1$ for all edges $e \in E$ with u being one of its endpoints. Thus, $L_{uv}^{\tilde{\mathcal{F}}} = -1$ if $\{u, v\} \in E$ and 0 otherwise, while $L_{uu}^{\tilde{\mathcal{F}}} = |\{e \in E : e = \{u, v\}\}|$; by definition, it follows that $L^{\tilde{\mathcal{F}}}$ coincides with the classical Laplacian matrix $L = D - A$ with $A \in \{0, 1\}^{n \times n}$.

Part 3. Setting $q = 0$ leads to, for all $u, v \in V$, $T_{uv}^{(q)} = \cos(0) + i \sin(0) = 1$. Thus, $L^{\tilde{\mathcal{F}}}$ coincides with the Directed Sheaf Laplacian $L^{\mathcal{F}}$ associated with the undirected version of G which is obtained from it by preserving each of its edges and making all of them undirected—this coincides with discarding $\Im(\tilde{\mathcal{F}}_{u \leq e}) = 0$ for all $e \in E$ where u is one of its endpoints. \square

Theorem 4. *Letting G be a directed graph with unit weights, the Directed Sheaf Laplacian $L^{\tilde{\mathcal{F}}}$ associated with a Trivial Directed Cellular Sheaf coincides with the Magnetic Laplacian $L^{(q)}$. In the special case where $q = \frac{1}{4}$, $L^{\tilde{\mathcal{F}}}$ also coincides with the Sign-Magnetic Laplacian L^σ .*

Proof. **Part 1.** First, we show that, when adopting a Trivial Directed Cellular Sheaf for a directed graph G with unit weights, we have:

$$\begin{aligned} L_{uv}^{\tilde{\mathcal{F}}} &= -T_{uv}^{(q)} & u, v \in V : u \neq v \\ L_{uu}^{\tilde{\mathcal{F}}} &= |\Gamma(u)| & u \in V. \end{aligned}$$

Eq. 2 and 3 read:

$$L_{uv}^{\tilde{\mathcal{F}}} = \begin{cases} -\tilde{\mathcal{F}}_{u \leq e}^* \tilde{\mathcal{F}}_{v \leq e} = -\tilde{\mathcal{F}}_{u \leq e}^T \tilde{\mathcal{F}}_{v \leq e}^0 T_{uv}^{(q)} & \text{if } e = (u, v) \\ -\tilde{\mathcal{F}}_{u \leq e}^* \tilde{\mathcal{F}}_{v \leq e} = -(\tilde{\mathcal{F}}_{u \leq e}^0 T_{vu}^{(q)})^* \tilde{\mathcal{F}}_{v \leq e} & \text{if } e = (v, u) \\ -\tilde{\mathcal{F}}_{u \leq e}^* \tilde{\mathcal{F}}_{v \leq e} = -\tilde{\mathcal{F}}_{u \leq e}^T \tilde{\mathcal{F}}_{v \leq e} & \text{if } e = \{u, v\} \\ 0 & \text{otherwise} \end{cases}$$

$$L_{uu}^{\tilde{\mathcal{F}}} = \sum_{e \in \Gamma(u)} \tilde{\mathcal{F}}_{u \leq e}^* \tilde{\mathcal{F}}_{u \leq e}.$$

When considering a Trivial Directed Cellular Sheaf, we have

- $\tilde{\mathcal{F}}_{u \leq e} = \tilde{\mathcal{F}}_{v \leq e} = 1$ if $e = \{u, v\} \in E$ and, thus, $L_{uv}^{\tilde{\mathcal{F}}} = -1 = -T_{uv}^{(q)}$ (the latter is because $A_{uv} = A_{vu}$ implies $T_{uv}^{(q)} = \cos(0) + i \sin(0) = 1$).
- $\tilde{\mathcal{F}}_{u \leq e} = 1$ and $\tilde{\mathcal{F}}_{v \leq e} = T_{uv}^{(q)}$ if $e = (u, v) \in E$ and, thus, $L_{uv}^{\tilde{\mathcal{F}}} = -T_{uv}^{(q)}$;
- $\tilde{\mathcal{F}}_{u \leq e} = T_{vu}^{(q)}$ and $\tilde{\mathcal{F}}_{v \leq e} = 1$ if $e = (v, u) \in E$ and, thus, $L_{uv}^{\tilde{\mathcal{F}}} = -(T_{vu}^{(q)})^* = -T_{uv}^{(q)}$.

Each diagonal term $L_{uu}^{\tilde{\mathcal{F}}}$ of $L^{\tilde{\mathcal{F}}}$ reads

$$\begin{aligned} L_{uu}^{\tilde{\mathcal{F}}} &= \sum_{e \in \Gamma(u)} \tilde{\mathcal{F}}_{u \leq e}^* \tilde{\mathcal{F}}_{u \leq e} = \sum_{e=(u,v) \in E} \underbrace{\tilde{\mathcal{F}}_{u \leq e}^T \tilde{\mathcal{F}}_{u \leq e}}_{=1} \\ &\quad + \sum_{e=(v,u) \in E} \underbrace{(\tilde{\mathcal{F}}_{u \leq e}^0)^T (T_{vu}^{(q)})^* \tilde{\mathcal{F}}_{u \leq e}^0 T_{vu}^{(q)}}_{=(T_{vu}^{(q)})^* (T_{vu}^{(q)})=1} + \sum_{e=\{u,v\} \in E} \underbrace{\tilde{\mathcal{F}}_{u \leq e}^T \tilde{\mathcal{F}}_{u \leq e}}_{=1} \\ &= |\Gamma(u)|, \end{aligned}$$

where $(T_{vu}^{(q)})^* (T_{vu}^{(q)}) = 1$ holds since $T_{vu}^{(q)} = i$. With this, Part 1 is shown.

Part 2.

The Magnetic Laplacian reads

$$L^{(q)} := D_s - H^{(q)}, \text{ with } H^{(q)} := A_s \odot \exp(i 2\pi q (A - A^\top)),$$

with $A_s := \frac{A+A^\top}{2}$ and $D_s = \text{diag}(\mathbf{1}_n A_s)$.

By definition we gave of $T_{uv}^{(q)}$, for a component u, v with $u, v \in V$, we have:

$$L_{uv}^{(q)} := D_{s_{uv}} - H_{uv}^{(q)} = D_{s_{uv}} - A_{s_{uv}} T_{uv}^{(q)}.$$

Part 2a. Let's assume G undirected. In such a case, we have we have $A_{s_{uv}} = 1$ whenever $\{u, v\} \in E$ and $A_{s_{uv}} = 0$ otherwise. This implies $D_{s_{uu}} = |\Gamma(u)|$. Thus, we have:

$$L_{uv}^{(q)} = -A_{s_{uv}} T_{uv}^{(q)} = -T_{uv}^{(q)} = L_{uv}^{\tilde{\mathcal{F}}} \quad u, v \in V : u \neq v$$

$$L_{uu}^{(q)} = D_{s_{uu}} - A_{s_{uu}} T_{uu}^{(q)} = D_{s_{uu}} = |\Gamma(u)| = L_{uu}^{\tilde{\mathcal{F}}} \quad u \in V,$$

where the last equation holds since $T_{uu}^{(q)} = 0$ for any q . Thus, $L^{\tilde{\mathcal{F}}} = L^{(q)}$.

Part 2b. Let's assume G directed without digons. In such a case, we have $A_{s_{uv}} = \frac{1}{2}$ whenever either $(u, v) \in E$ or $(v, u) \in E$ and $A_{s_{uv}} = 0$ otherwise. This implies $D_{s_{uu}} = \frac{1}{2}|\Gamma(u)|$. Thus, we have:

$$L_{uv}^{(q)} = -A_{s_{uv}} T_{uv}^{(q)} = -\frac{1}{2} T_{uv}^{(q)} = \frac{1}{2} L_{uv}^{\tilde{\mathcal{F}}} \quad u, v \in V : u \neq v$$

$$L_{uu}^{(q)} = D_{s_{uu}} - A_{s_{uu}} T_{uu}^{(q)} = D_{s_{uu}} = \frac{1}{2} |\Gamma(u)| = \frac{1}{2} L_{uu}^{\tilde{\mathcal{F}}} \quad u \in V,$$

where the last equation holds since $T_{uu}^{(q)} = 0$ for any q . Thus, $L^{\tilde{\mathcal{F}}} = 2L^{(q)}$. Notice that the scaling factor is immaterial when the Laplacian matrix is embedded in a GCN/SNN, as it is directly subsumed by either W_1 or W_2 in Eq (8) (only by the latter in a GCN, where W_1 is not present).

Part 3. Since, as shown in [21], $L^{(q)}$ and L^σ coincide with $q = \frac{1}{4}$, the last part of the claim follows directly from Parts 2a and 2b. \square

Theorem 5. Let G be a directed graph with unit weights. Assuming a Trivial Directed Cellular Sheaf, the conjugate transpose $\tilde{\delta}^*$ of the Directed Coboundary Operator $\tilde{\delta}$ boils down to the complex-valued node-to-edge incidence matrix $\hat{B} \in \mathbb{C}^{n \times m}$ defined for an edge $e \in E$ incident to a vertex u :

$$\hat{B}_{ue} = \begin{cases} 1 & \text{if } e = \{u, v\} \text{ or } e = \{u, v\} \text{ with } u < v \\ -1 & \text{if } e = \{u, v\} \text{ with } u > v \\ -T_{uv}^{(q)} & \text{if } e = (v, u). \end{cases}$$

It follows that $L^{(q)} = \hat{B}\hat{B}^*$. With $q = \frac{1}{4}$, $L^{(\frac{1}{4})} = L^\sigma = \hat{B}\hat{B}^*$.

Proof. (First, notice the arbitrary orientation that was given to the undirected edges).

From the proof of the previous theorem, we know that, if G has unit weights and the Directed Cellular Sheaf is trivial, we have:

$$\begin{aligned} L_{uv}^{\tilde{\mathcal{F}}} &= -T_{uv}^{(q)} & u, v \in V : u \neq v \\ L_{uu}^{\tilde{\mathcal{F}}} &= |\Gamma(u)| & u \in V. \end{aligned}$$

Let's consider $(\hat{B}\hat{B}^*)_{uv} = \sum_{e' \in E} \hat{B}_{ue'} (\hat{B}_{ve'})^*$. Since we are considering a graph, u, v can only share a single edge. Calling it e , we have $(\hat{B}\hat{B}^*)_{uv} = \hat{B}_{ue} (\hat{B}_{ve})^*$ if $e \in E$ or 0 if they share no edge at all. Let's assume they do, and considering three cases:

- If $e = \{u, v\}$ with $u < v$, $\hat{B}_{ue} = 1$ and $(\hat{B}_{ve})^* = -1$ with an arbitrary orientation and, thus, $\hat{B}_{ue} (\hat{B}_{ve})^* = -1 = -T_{uv}^{(q)}$ (this is correct since, as shown before, $T_{uv}^{(q)}$ is always equal to 1 if $A_{uv} = A_{vu}$).
- If $e = \{u, v\}$ with $u > v$, $\hat{B}_{ue} = -1$ and $(\hat{B}_{ve})^* = 1$ with an arbitrary orientation and, thus, $\hat{B}_{ue} (\hat{B}_{ve})^* = -1 = -T_{uv}^{(q)}$ (as shown before, the latter is always equal to 1 if $A_{uv} = A_{vu}$).
- If $e = (u, v)$, $\hat{B}_{ue} = 1$ and $(\hat{B}_{ve})^* = (-T_{vu}^{(q)})^*$ and, thus, $\hat{B}_{ue} (\hat{B}_{ve})^* = (-T_{vu}^{(q)})^* = -T_{uv}^{(q)}$ since $T^{(q)}$ is Hermitian by construction.
- If $e = (v, u)$, $\hat{B}_{ue} = -T_{uv}^{(q)}$ and $(\hat{B}_{ve})^* = 1$ and, thus, $\hat{B}_{ue} (\hat{B}_{ve})^* = -T_{uv}^{(q)}$.

This shows that, if G has unit weights and assuming a Trivial Directed Cellular Sheaf, we have $L^{\tilde{\mathcal{F}}} = \hat{B}\hat{B}^*$. The fact that (with a scaling factor of 2, when needed) $L^{(q)} = \hat{B}\hat{B}^*$ and $L^\sigma = \hat{B}\hat{B}^*$ when $q = \frac{1}{4}$ follow from the previous theorem. \square

D Complexity of DSNN

As mentioned in the paper, the complexity of DSNN coincides, asymptotically, with that of NSD. This is because the adoption of complex-valued restriction maps—which are specific to DSNN and not present in NSD—does not affect the asymptotic inference complexity of DSNN. This is because complex-valued synaptic weights, pre-activations, and activations only incur a constant multiplicative overhead (approximately a factor of 4) in the forward pass and, thus, do not alter the asymptotic complexity from the real-valued case analysis. To better see this, consider three complex-valued matrices:

$$A = A_R + iA_I, \quad X = X_R + iX_I, \quad Y = Y_R + iY_I,$$

with

$$A_R, A_I \in \mathbb{R}^{m \times n}, \quad X_R, X_I \in \mathbb{R}^{n \times p}, \quad Y_R, Y_I \in \mathbb{R}^{m \times p},$$

satisfying the complex linear equation $Y = AX$. This equation can be rewritten purely in the real domain using the *lifting* transformation:

$$X_{\mathbb{R}} = \begin{bmatrix} X_R \\ X_I \end{bmatrix} \in \mathbb{R}^{2n \times p}, \quad Y_{\mathbb{R}} = \begin{bmatrix} Y_R \\ Y_I \end{bmatrix} \in \mathbb{R}^{2m \times p}, \quad A_{\mathbb{R}} = \begin{bmatrix} A_R & -A_I \\ A_I & A_R \end{bmatrix} \in \mathbb{R}^{2m \times 2n},$$

so that $Y_{\mathbb{R}} = A_{\mathbb{R}}X_{\mathbb{R}}$ holds. Hence, complex-valued operations can be reduced to real-valued operations with a constant factor overhead, which is immaterial in the asymptotic complexity.

E Further Details on the Datasets

Real-world dataset. The Texas, Wisconsin, and Cornell datasets are part of the WebKB collection, modeling links between websites from different universities. In these datasets, nodes are labeled as student, project, course, staff, or faculty.

The Film dataset is derived from a film–director–actor–writer network. Each node represents an actor, and edges indicate co-occurrence on the same Wikipedia page. Node features correspond to keywords extracted from these Wikipedia pages. The nodes are classified into five categories based on the content of the actors’ Wikipedia entries.

The CiteSeer dataset contains 3,312 scientific publications classified into six categories. The citation network includes 4,732 links. Each publication is represented by a binary word vector indicating the presence or absence of words from a dictionary of 3,703 unique terms.

The PubMed dataset consists of 19,717 scientific publications related to diabetes, categorized into three classes. The citation network contains 44,338 links. Each publication is described by a TF-IDF weighted word vector derived from a dictionary of 500 unique words.

The Cora dataset includes 2,708 scientific publications classified into seven classes, with a citation network comprising 5,429 links. Each publication is represented by a binary word vector indicating the presence or absence of words from a dictionary of 1,433 unique terms.

The Squirrel and Chameleon datasets consist of articles from the English Wikipedia (December 2018). Nodes represent articles, and edges represent mutual links between them. Node features indicate the presence of specific nouns in the articles. Nodes are grouped into five categories based on the original regression targets.

Synthetic dataset. Following [3], we generate synthetic graphs using the directed stochastic block model (DSBM) as follows. Let n be the number of nodes and C the number of equal-sized communities $\{C_1, \dots, C_C\}$. First, we sample an undirected graph by connecting each pair of nodes $u \in C_i$ and $v \in C_j$ independently with probability $\alpha_{ij} \in [0, 1]$, $\alpha_{ij} = \alpha_{ji}$, where α_{ii} controls intra-community edge density and α_{ij} for $i \neq j$ controls inter-community connectivity. To obtain a directed graph, we introduce a rule to transform the graph from undirected to directed: we define a collection of probabilities $\{\beta_{ij}\}_{1 \leq i,j \leq C}$, where $\beta_{ij} \in [0, 1]$, such that $\beta_{ij} + \beta_{ji} = 1$. If $u \in C_i$ and $v \in C_j$, we orient the edge $u \rightarrow v$ with probability β_{ij} , and $v \rightarrow u$ with probability β_{ji} .

F Further Details on the Experiments

Hardware. The experiments were conducted on 2 different machines:

1. An Intel(R) Xeon(R) Gold 6326 CPU @ 2.90GHz with 380 GB RAM, equipped with an NVIDIA Ampere A100 40GB.
2. A 12th Gen Intel(R) Core(TM) i9-12900KF CPU @ 3.20GHz CPU with 64 GB RAM, equipped with an NVIDIA RTX 4090 GPU.

Model Settings. We trained every learning model considered in this paper for up to 1000 epochs with early stops of 200. We adopted a learning rate of $\{1 \cdot 10^{-2}, 2 \cdot 10^{-2}, 5 \cdot 10^{-3}\}$ and employed the optimization algorithm Adam.

We adopted a hyperparameter optimization procedure to identify the best set of parameters for each model. For every model, we searched for the optimal combination of the following hyperparameters:

- **Dropout:** $\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$
- **Number of layers:** $\{2, 3, 4, 5, 6\}$
- **Hidden channels:** $\{8, 16, 32, 64, 128\}$.

For some specific models, we also included additional hyperparameters in the search space:

- **NSD-comp and NSD:** $\text{sheaf_act} \in \{\text{elu}, \text{tanh}, \text{relu}\}$; $d \in \{2, 3, 4, 5\}$; $\text{add_lp} \in \{\text{True}, \text{False}\}$; $\text{add_hp} \in \{\text{True}, \text{False}\}$

- **DirGNN:** $\alpha_{\text{DirGNN}} \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$; $\text{jk} \in \{\text{cat}, \text{max}\}$
- **MagNet:** $q \in \{0.0, 0.05, 0.1, 0.15, 0.2, 0.25\}$
- **GCNII:** $\alpha_{\text{GCNII}} \in \{0.0, 0.1, 0.2\}$; $\lambda \in \{0.0, 1.0, 1.5\}$
- **FAGCN:** $\varepsilon \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$
- **GGCN:** $\text{decay_rate} \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2\}$
- **GPRGNN:** $\alpha_{\text{GPRGNN}} \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$; $\text{dprate_GPRGNN} \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$
- **DSNN:** $q \in [0, 0.1, 0.15, 0.20, 0.25, 0.5, 0.75, 1]$, $\text{sheaf_act} \in \{\text{elu}, \text{tanh}, \text{relu}\}$, $d \in \{2, 3, 4, 5\}$; $\text{add_lp} \in \{\text{True}, \text{False}\}$; $\text{add_hp} \in \{\text{True}, \text{False}\}$