Neural Sheaf Diffusion: A Topological Perspective on Heterophily and Oversmoothing in GNNs

Anonymous Author(s)
Affiliation
Address
email

Abstract

Cellular sheaves equip graphs with a "geometrical" structure by assigning vector spaces and linear maps to nodes and edges. Graph Neural Networks (GNNs) implicitly assume a graph with a trivial underlying sheaf. This choice is reflected in the structure of the graph Laplacian operator, the properties of the associated diffusion equation, and the characteristics of the convolutional models that discretise this equation. In this paper, we use cellular sheaf theory to show that the underlying geometry of the graph is deeply linked with the performance of GNNs in heterophilic settings and their oversmoothing behaviour. By considering a hierarchy of increasingly general sheaves, we study how the ability of the sheaf diffusion process to achieve linear separation of the classes in the infinite time limit expands. At the same time, we prove that when the sheaf is non-trivial, discretised parametric diffusion processes have greater control than GNNs over their asymptotic behaviour. On the practical side, we study how sheaves can be learned from data. The resulting sheaf diffusion models have many desirable properties that address the limitations of classical graph diffusion equations (and corresponding GNN models) and obtain state-of-the-art results in heterophilic settings. Overall, our work provides new connections between GNNs and algebraic topology and would be of interest to both fields.

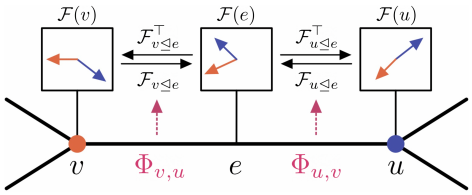

Figure 1: A sheaf $(G, \mathcal{F})$ shown for a single edge of the graph. The *stalks* are isomorphic to $\mathbb{R}^2$. The *restriction maps* $\mathcal{F}_{v \trianglelefteq e}$, $\mathcal{F}_{u \trianglelefteq e}$ and their adjoints move the vector features between these spaces. In practice, we learn the sheaf from data via a parametric function Φ.

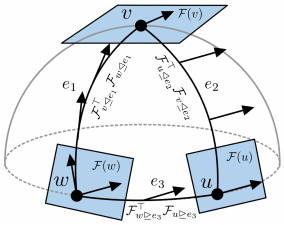

Figure 2: Analogy between parallel transport on a sphere and transport on a discrete vector bundle. A tangent vector is moved from $\mathcal{F}(w) \to \mathcal{F}(v) \to \mathcal{F}(u)$ and back.

1 Introduction

Graph Neural Networks (GNNs) [7, 14, 18–20, 29, 43, 48] have recently become very popular in the ML community as a model of choice to deal with relational and interaction data due to their multiple successful applications in domains ranging from social science and particle physics to structural biology and drug design. In this work, we focus on two main problems often observed in GNNs: their poor performance in heterophilic graphs [57] and their oversmoothing behaviour [36, 38]. The

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

former arises from the fact that GNNs are usually built on the strong assumption of *homophily*, i.e., that nodes tend to connect to other similar nodes. The latter refers to a phenomenon of deeper GNNs producing features that are too smooth to be useful.

Contributions. We show that these two fundamental problems are linked by a common cause: the underlying "geometry" of the graph (used here in a very loose sense). When this geometry is trivial, as is typically the case, the two phenomena described above emerge. We make these statements precise through the lens of cellular sheaf theory [13, 46], a subfield of algebraic topology [25]. Intuitively, a cellular sheaf associates a vector space to each node and edge of a graph, and a linear map between these spaces for each incident node-edge pair (Figure 1).

In Section 4 we analyse how by considering a hierarchy of increasingly general sheaves, starting from a trivial one, a diffusion equation based on the sheaf Laplacian [23] can solve increasingly more complicated node-classification tasks in the infinite time limit. In this regime, we show that oversmoothing and problems due to heterophily can be avoided by equipping the graph with the right sheaf structure for the task. In Section 5, we study the behaviour of a non-linear, parametric, and discrete version of this process. This results in a *Sheaf Convolutional Network* [22] that generalises GCN of Kipf and Welling [29]. We prove that this discrete diffusion process is more flexible and has greater control over its asymptotic behaviour than GCNs [8, 39].

All these results are based on the properties of the harmonic space of the sheaf Laplacian, which we study from a spectral perspective in Section 3. We provide a new Cheeger-type inequality for the spectral gap of the sheaf Laplacian and note that these results might be of independent interest for spectral sheaf theory. Finally, in Section 6, we apply our theory for designing simple and practical GNN models. We describe how to construct Sheaf Neural Networks by learning sheaves from data, thus making these types of model applicable beyond the toy experimental setting where they were originally introduced [22]. The resulting models obtain state-of-the-art results in heterophilic graphs and show strong performance in homophilic ones. Code is attached in the supplementary.

2 Background

A *cellular sheaf* [13, 46] over a graph (Figure 1) is a mathematical object associating a vector space to each node and edge in the graph and a map between these spaces for each incident node-edge pair. We define this formally below:

Definition 1. *A cellular sheaf $(G, \mathcal{F})$ on an undirected graph $G = (V, E)$ consists of:*

- *A vector space $\mathcal{F}(v)$ for each $v \in V$.*
- *A vector space $\mathcal{F}(e)$ for each $e \in E$.*
- *A linear map $\mathcal{F}_{v \trianglelefteq e} : \mathcal{F}(v) \to \mathcal{F}(e)$ for each incident $v \trianglelefteq e$ node-edge pair.*

The vector spaces of the nodes and edges are called *stalks*, while the linear maps are referred to as *restriction maps*. The space formed by all the spaces associated to the nodes of the graph is called the space of 0-*cochains* $C^0(G; \mathcal{F}) := \bigoplus_{v \in V} \mathcal{F}(v)$, where $\bigoplus$ denotes the direct sum of vector spaces. For a 0-cochain $\mathbf{x} \in C^0(G; \mathcal{F})$, we use $\mathbf{x}_v$ to refer to the vector in $\mathcal{F}(v)$ of node v. Hansen and Ghrist [24] have constructed a convenient mental model for these objects based on opinion dynamics. In this context, $\mathbf{x}_v$ is the 'private opinion' of node v, while $\mathcal{F}_{v \trianglelefteq e}\mathbf{x}_v$ expresses how that opinion manifests publicly in a 'discourse space' formed by $\mathcal{F}(e)$. A particularly important subspace of $C^0(G; \mathcal{F})$ is the space of *global sections* $H^0(G; \mathcal{F}) := \{\mathbf{x} \in C^0(G; \mathcal{F}) : \mathcal{F}_{v \trianglelefteq e}\mathbf{x}_v = \mathcal{F}_{u \trianglelefteq e}\mathbf{x}_u\}$ containing those private opinions $\mathbf{x}$ for which all neighbours (v, u) agree with each other in the discourse space.

Given a cellular sheaf $(G, \mathcal{F})$, we can define a *sheaf Laplacian* operator [23] measuring the aggregated 'disagreement of opinions' at each node:

Definition 2. *The sheaf Laplacian of a sheaf $(G, \mathcal{F})$ is a linear map $L_{\mathcal{F}} : C^0(G, \mathcal{F}) \to C^0(G, \mathcal{F})$ defined node-wise as $L_{\mathcal{F}}(\mathbf{x})_v := \sum_{v, u \trianglelefteq e} \mathcal{F}^{\top}_{v \trianglelefteq e}(\mathcal{F}_{v \trianglelefteq e}\mathbf{x}_v - \mathcal{F}_{u \trianglelefteq e}\mathbf{x}_u)$.*

The sheaf Laplacian is a positive semi-definite block matrix. The diagonal blocks are $L_{\mathcal{F}vv} = \sum_{v \trianglelefteq e} \mathcal{F}^{\top}_{v \trianglelefteq e}\mathcal{F}_{v \trianglelefteq e}$, while the non-diagonal blocks $L_{\mathcal{F}vu} = -\mathcal{F}^{\top}_{v \trianglelefteq e}\mathcal{F}_{u \trianglelefteq e}$. Denoting by D the block-diagonal of $L_{\mathcal{F}}$, the normalised sheaf Laplacian is given by $\Delta_{\mathcal{F}} := D^{-1/2} L_{\mathcal{F}} D^{-1/2}$. For simplicity, we assume that all the stalks have a fixed dimension d. In that case, the sheaf Laplacian is a $nd \times nd$ real matrix, where n is the number of nodes of G. When the vector spaces are set to $\mathbb{R}$ (i.e., $d = 1$)

and the linear maps to the identity map over $\mathbb{R}$, the underlying sheaf is trivial and one recovers the well-known $n \times n$ graph Laplacian matrix and its normalised version.

Harmonic signals have perfect 'agreement of opinions' w.r.t. the sheaf structure, i.e., $L_{\mathcal{F}}\mathbf{x} = 0$. The central theorem of discrete Hodge theory [24] provides a nice formalisation of this intuition and shows harmonic signals coincide with the global sections of the sheaf:

Theorem 3. *$H^0(G;\mathcal{F})$ and $\ker(L_{\mathcal{F}})$ are isomorphic as vector spaces.*

The sheaves with orthogonal maps (i.e. $\mathcal{F}_{v \trianglelefteq e} \in O(d)$ the Lie group of $d \times d$ orthogonal matrices) provide a more geometric interpretation of sheaves and play an important role in our analysis. Such sheaves are called *discrete $O(d)$-bundles* and can be seen as a discrete analogy of vector bundles from differential geometry [50]. Intuitively, these are objects describing how vector spaces are attached to the points of a manifold. In our discrete case, the role of the manifold is played by the graph and the sheaf Laplacian describes how the elements of a vector space are transported via rotations in another neighbouring vector space similarly to how tangent vectors are moved across a manifold via parallel transport (connection; see Figure 2). Due to this analogy, the sheaf Laplacian on $O(d)$-*bundles* is also referred to as *connection Laplacian* [47].

3 Harmonic Space of Sheaf Laplacians

Preliminaries. Since the harmonic space of the sheaf Laplacian will be related to the linear separation power of sheaf diffusion, we investigate in this section when this space is non-empty and derive to what extent the graph structure affects that. Because a major role in the analysis below is played by discrete vector bundles, we concentrate on this case. We note though, that our results below generalise to the general linear group $\mathcal{F}_{v \trianglelefteq e} \in GL(d)$, the Lie group of $d \times d$ invertible matrices, provided we can also control the norm of the restriction maps from below. Finally, we note that the normalised Laplacian $\Delta_{\mathcal{F}}$ is preferred to the unnormalised $L_{\mathcal{F}}$ for most practical purposes due to its bounded spectrum and, therefore, our paper is focused on the former.

Given a discrete $O(d)$-bundle, the block diagonal of $L_{\mathcal{F}}$ has a diagonal structure since $L_{\mathcal{F}vv} = d_v \mathbf{I}_d$, where d_v is the degree of node v. Accordingly, if a signal $\tilde{\mathbf{x}} \in \ker(L_{\mathcal{F}})$, then the signal $\mathbf{x} : v \mapsto \sqrt{d_v}\tilde{\mathbf{x}}_v \in \ker(\Delta_{\mathcal{F}})$ and similarly for the inverse transformation. In general, the harmonic space may be trivial (i.e. $\ker(\Delta_{\mathcal{F}}) = 0$). This happens when the constraints $\mathcal{F}_{v \trianglelefteq e}\mathbf{x}_v = \mathcal{F}_{u \trianglelefteq e}\mathbf{x}_u$ cannot be satisfied along all edges.

Key to our analysis is studying *transport* operators induced by the restriction maps of the sheaf. Given nodes $v, u \in V$ and a path $\gamma_{v \to u} = (v, v_1, \ldots, v_\ell, u)$ from v to u, we consider a notion of *transport* from the stalk $\mathcal{F}(v)$ to the stalk $\mathcal{F}(u)$, constructed by composing restriction maps (and their transposes) along the edges:

$$\mathbf{P}^{\gamma}_{v \to u} := (\mathcal{F}^{\top}_{u \trianglelefteq e}\mathcal{F}_{v_\ell \trianglelefteq e}) \ldots (\mathcal{F}^{\top}_{v_1 \trianglelefteq e}\mathcal{F}_{v \trianglelefteq e}) : \mathcal{F}(v) \to \mathcal{F}(u).$$

For general sheaf structures, the graph transport is *path dependent*, meaning that how the vectors are transported across two nodes depends on the path between them (see Figure 2). In fact, we show that this property characterises the *spectral gap* of a sheaf Laplacian, i.e. the smallest eigenvalue of $\Delta_{\mathcal{F}}$.

Proposition 4. *If $\mathcal{F}$ is a discrete $O(d)$ bundle over a connected graph and $r := \max_{\gamma_{v \to u}, \gamma'_{v \to u}} ||\mathbf{P}^{\gamma}_{v \to u} - \mathbf{P}^{\gamma'}_{v \to u}||$, then we have $\lambda^{\mathcal{F}}_0 \leq \frac{r^2}{2}$.*

A consequence of this result is that there is always a non-trivial harmonic space (i.e. $\lambda^{\mathcal{F}}_0 = 0$) if the transport maps generated by an orthogonal sheaf are *path-independent*. Next, we address the opposite direction.

Proposition 5. *If $\mathcal{F}$ is a discrete $O(d)$ bundle over a connected graph and $\mathbf{x} \in H^0(G, \mathcal{F})$, then for any cycle γ based at $v \in V$ we have $\mathbf{x}_v \in \ker(\mathbf{P}^{\gamma}_{v \to v} - \mathbf{I})$.*

This proposition highlights the interplay between the graph and the sheaf structure. A simple consequence of this result is that for any cycle-free subset $S \subset V$, we have that any sheaf (or connection-) Laplacian restricted to S always admits a non-trivial harmonic space. A natural question connected to the previous result is whether a Cheeger-like inequality holds in the other direction. This turns out to be the case:

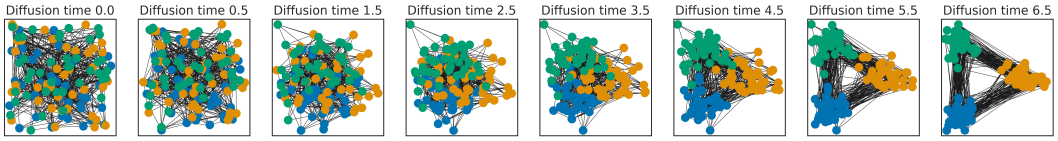

Figure 3: Diffusion process on $O(2)$-bundles progressively separates the classes of the graph.

Proposition 6. *Let $\mathcal{F}$ be a discrete $O(d)$ bundle over a connected graph G with n nodes and let $||(\mathbf{P}^{\gamma}_{v\to v} - \mathbf{I})\mathbf{x}|| \geq \epsilon||\mathbf{x}||$ for all cycles $\gamma_{v\to v}$. Then $\lambda_0^{\mathcal{F}} \geq \epsilon^2(2\mathrm{diam}(G)n\, d_{max})^{-1}$.*

While the bound above is of little use in practice, it shows how the spectral gap of a sheaf Laplacian is indeed related to the deviation of the transport maps from being path-independent. We note that the Cheeger-like inequality presented here is not unique and other types of bounds on $\lambda_0^{\mathcal{F}}$ have been derived [2]. We conclude this section by further analysing the dimensionality of the harmonic space of discrete $O(d)$-bundles:

Lemma 7. *Let $\mathcal{F}$ be a discrete $O(d)$ bundle over a connected graph G. Then $\dim(H^0) \leq d$ and $\dim(H^0) = d$ iff the transport is path-independent.*

4 The Separation Power of Sheaf Diffusion

Preliminaries. Let us now assume G to be a graph with d-dimensional node feature vectors $\mathbf{x}_v \in \mathcal{F}(v)$. The features of all nodes are represented as a single vector $\mathbf{x} \in C^0(G;\mathcal{F})$ stacking all the individual d-dimensional vectors. Additionally, if we allow for f feature channels, everything can be represented as a matrix $\mathbf{X} \in \mathbb{R}^{(nd)\times f}$, whose columns are vectors in $C^0(G;\mathcal{F})$. We are interested in the spatially discretised *sheaf diffusion* process governed by the following PDE:

$$\mathbf{X}(0) = \mathbf{X}, \quad \dot{\mathbf{X}}(t) = -\Delta_{\mathcal{F}}\mathbf{X}(t). \tag{1}$$

It can be shown that in the time limit, each feature channel is projected into $\ker(\Delta_{\mathcal{F}})$ [23]. As described above (up to a $D^{-1/2}$ normalisation), this space contains the signals that agree with the restriction maps of the sheaf along all the edges. Thus, sheaf diffusion can be seen as a global 'synchronisation' process. In what follows, we analyse the ability of certain classes of sheaves to linearly separate the features in the limit of the diffusion processes they induce. We utilise this as a proxy for the capacity of certain diffusion processes to avoid oversmoothing.

Definition 8. *A hypothesis class of sheaves with d-dimensional stalks $\mathcal{H}^d$ has* linear separation power *over a family of graphs $\mathcal{G}$ if for any labelled graph $G = (V, E) \in \mathcal{G}$, there is a sheaf $(\mathcal{F}, G) \in \mathcal{H}^d$ that can linearly separate the classes of G in the time limit of Equation 1 for a dense subset $\mathcal{X}_{\mathcal{F}} \subset \mathbb{R}^{nd\times f}$ of initial conditions.*

Note that the restriction to a set of initial conditions that is dense in the ambient space is necessary because, in the limit, diffusion behaves like a projection in the harmonic space and there will always be degenerate initial conditions that will yield a zero projection.

We will now show how the choice of the sheaf impacts the behaviour of the diffusion process. For this purpose, we will consider a hierarchy of increasingly general classes of sheaves.

Symmetric invertible. $\mathcal{H}^d_{\mathrm{sym}} := \{(\mathcal{F}, G) : \mathcal{F}_{v\trianglelefteq e} = \mathcal{F}_{u\trianglelefteq e},\ \det(\mathcal{F}_{v\trianglelefteq e}) \neq 0\}$. We note that for $d = 1$, the sheaf Laplacians induced by this class of sheaves coincides with the set of the well-known weighted graph Laplacians with strictly positive weights, which also includes the usual graph Laplacian (see proof in Appendix B). Therefore, this hypothesis class is of particular interest since it includes those graph Laplacians typically used by graph convolutional models such as GCN [29] and ChebNet [14]. We first show that this class of sheaf Laplacians can linearly separate the classes in binary classification settings under certain homophily assumptions:

Proposition 9. *Let $\mathcal{G}$ be the set of connected graphs $G = (V, E)$ with two classes $A, B \subset V$ such that for each $v \in A$, there exists $u \in A$ and an edge $(v, u) \in E$. Then $\mathcal{H}^1_{\mathrm{sym}}$ has linear separation power over $\mathcal{G}$.*

In contrast, under certain heterophilic conditions, this hypothesis class is not powerful enough to linearly separate the two classes no matter what the initial conditions are:

Proposition 10. *Let $\mathcal{G}$ be the set of connected bipartite graphs $G = (A, B, E)$, with partitions A, B forming two classes and $|A| = |B|$. Then $\mathcal{H}^1_{\text{sym}}$ cannot linearly separate any graph in $\mathcal{G}$ for any initial conditions $\mathbf{X}(0) \in \mathbb{R}^{n \times f}$.*

Non-symmetric invertible. $\mathcal{H}^d := \{(\mathcal{F}, G) : \det(\mathcal{F}_{v \trianglelefteq e}) \neq 0\}$. This larger hypothesis class addresses the above limitation by allowing non-symmetric relations:

Proposition 11. *Let $\mathcal{G}$ contain all the connected graphs $G = (V, E)$ with two classes $A, B \subseteq V$. Consider a sheaf $(\mathcal{F}; G) \in \mathcal{H}^1$ with $\mathcal{F}_{v \trianglelefteq e} = -\alpha_e$ if $v \in A$ and $\mathcal{F}_{u \trianglelefteq e} = \alpha_e$ if $u \in B$ with $\alpha_e > 0$ for all $e \in E$. Then the diffusion induced by $(\mathcal{F}; G)$ can linearly separate the classes of G for a dense subset of initial conditions and $\mathcal{H}^1$ has linear separation power over $\mathcal{G}$.*

The type of sheaf above can also be interpreted as a discrete $O(1)$-bundle over a weighted graph with edge weights α_e^2 and transport maps $\mathcal{F}^\top_{v \trianglelefteq e} \mathcal{F}_{u \trianglelefteq e} = -1$ for the inter-class edges and $+1$ for the intra-class edges. In agreement with the results in Section 3, it can be checked that this transport is path-independent. Importantly, this result provides a sheaf-theoretic explanation for why a recent body of work [4, 12, 55] has found negatively-weighted edges to help in heterophilic settings.

So far we have only studied the effects of changing the type of sheaves in dimension one. We now consider the effects of adjusting the dimension of the stalks and begin by stating a fundamental limitation of (sheaf) diffusion when $d = 1$.

Proposition 12. *Let G be a connected graph with $C \geq 3$ classes. $\mathcal{H}^1$ cannot linearly separate any $\mathbf{X} \in \mathbb{R}^{n \times f}$.*

This is essentially a consequence of $\dim\big(\ker(\Delta_{\mathcal{F}})\big) \leq 1$ in this case, by virtue of Lemma 7. From a GNN perspective, this means that in the infinite depth setting, sufficient *stalk width* (i.e., dimension d) is needed in order to solve tasks involving more than two classes. Note that d is different from the classical notion of feature channels f. As the result above shows, the latter has no effect on the linear separability of the classes in $d = 1$. Next, we will see that the former does.

Diagonal invertible. $\mathcal{H}^d := \{(\mathcal{F}, G) : \text{diagonal } \mathcal{F}_{v \trianglelefteq e}, \det(\mathcal{F}_{v \trianglelefteq e}) \neq 0\}$. The sheaves in this class can be seen as d independent sheaves from $\mathcal{H}^1$ encoded in the d-dimensional diagonals of their restriction maps. This perspective allows us to generalise Proposition 11 to a multi-class setting:

Proposition 13. *Let $\mathcal{G}$ be the set of connected graphs with nodes belonging to $C \geq 3$ classes. Then for $d \geq C$, $\mathcal{H}^d_{\text{diag}}$ has linear separation power over $\mathcal{G}$.*

This result illustrates the benefits of using higher-dimensional stalks, while maintaining a simple and computationally convenient class of diagonal restriction maps. Next, with more complex restriction maps, we can show that lower-dimensional stalks can be used to achieve linear separation in the presence of even more classes.

Orthogonal. $\mathcal{H}^d_{\text{orth}} := \{(\mathcal{F}, G) : \mathcal{F}_{v \trianglelefteq e} \in O(d)\}$ is the class of $O(d)$-bundles. Orthogonal maps are able to make more efficient use of the space available to them than diagonal restriction maps:

Theorem 14. *Let $\mathcal{G}$ be the class of connected graphs with $C \leq 2d$ classes. Then, for all $d \in \{2, 4\}$, $\mathcal{H}^d_{\text{orth}}$ has linear separation power over $\mathcal{G}$.*

Figure 3 includes an example diffusion process over an $O(2)$-bundle.

Summary: Different sheaf classes give rise to different behaviours of the diffusion process and consequently, to different separation capabilities. Taken together, these results show that solving any node classification task can be reduced to performing diffusion with the right sheaf.

5 Energy Flows and Sheaf Convolutions

Preliminaries The continous diffusion process from Equation 1 has the Euler discretisation with unit step-size $\mathbf{X}(t+1) = \mathbf{X}(t) - \Delta_{\mathcal{F}}\mathbf{X}(t) = (\mathbf{I}_{nd} - \Delta_{\mathcal{F}})\mathbf{X}(t)$. Assuming $\mathbf{X} \in \mathbb{R}^{nd \times f_1}$, we can equip the right side with weight matrices $\mathbf{W}_1 \in \mathbb{R}^{d \times d}, \mathbf{W}_2 \in \mathbb{R}^{f_1 \times f_2}$ and a non-linearity σ to arrive

at the following model originally proposed by Hansen and Gebhart [22]:

$$\mathbf{Y} = \sigma\Big(\big(\mathbf{I}_{nd} - \Delta_{\mathcal{F}}\big)(\mathbf{I}_n \otimes \mathbf{W}_1)\mathbf{X}\mathbf{W}_2\Big) \in \mathbb{R}^{nd \times f_2}, \tag{2}$$

where f_1, f_2 are the number of input and output feature channels, and $\otimes$ denotes the Kronecker product. Here, $\mathbf{W}_1$ multiplies from the left the vector feature of all the nodes in all channels (i.e. $\mathbf{W}_1\mathbf{x}_v^i$ for all v and channels i), while $\mathbf{W}_2$ multiplies the features from the right and can adjust the number of feature channels, just like in GCNs.

It is natural to call this model a **Sheaf Convolutional Network** (SCN) since when $\Delta_{\mathcal{F}}$ is the usual normalised graph Laplacian, $\mathbf{W}_1$ becomes a scalar and one recovers the GCN of Kipf and Welling [29]. This shows that GCNs and, more generally, SCNs are a non-linear, parametric and discrete version of (sheaf) diffusion. In what follows, we analyse how expressive these non-linear models are compared to their base diffusion process. To do so, we investigate how SCNs affect the *sheaf Dirichlet energy* $E_{\mathcal{F}}(\mathbf{x})$, which sheaf diffusion is known to minimise over time.

Definition 15. $E_{\mathcal{F}}(\mathbf{x}) := \mathbf{x}^\top \Delta_{\mathcal{F}} \mathbf{x} = \frac{1}{2}\sum_{e:=(v,u)} \|\mathcal{F}_{v \trianglelefteq e} D_v^{-1/2}\mathbf{x}_v - \mathcal{F}_{u \trianglelefteq e} D_u^{-1/2}\mathbf{x}_u\|_2^2$

Similarly, for multiple channels the energy is $E_{\mathcal{F}}(\mathbf{X}) := \mathrm{trace}(\mathbf{X}^\top \Delta_{\mathcal{F}} \mathbf{X})$. This is a measure of how close a signal $\mathbf{x}$ is to $\ker(\Delta_{\mathcal{F}})$ and it is easy to see that $\mathbf{x} \in \ker(\Delta_{\mathcal{F}}) \Leftrightarrow E_{\mathcal{F}}(\mathbf{x}) = 0$. We begin by studying the sheaves for which the energy decreases and features end up asymptotically in $\ker(\Delta_{\mathcal{F}})$. Let $\lambda_* := \max_{i>0}(\lambda_i^{\mathcal{F}} - 1)^2$ and denote by $\mathcal{H}^d_{\mathrm{orth,sym}} := \mathcal{H}^d_{\mathrm{orth}} \cap \mathcal{H}^d_{\mathrm{sym}}$.

Theorem 16. *If $(\mathcal{F}, G) \in \mathcal{H}^d_{\mathrm{orth,sym}}$ and σ = (Leaky)ReLU, $E_{\mathcal{F}}(\mathbf{Y}) \leq \lambda_* \|\mathbf{W}_1\|_2^2 \|\mathbf{W}_2^\top\|_2^2 E_{\mathcal{F}}(\mathbf{X})$.*

In particular, this means that if $\lambda_* \|\mathbf{W}_1\|_2^2 \|\mathbf{W}_2^\top\|_2^2 < 1$, the signal converges exponentially fast to $\ker(\Delta_{\mathcal{F}})$. In some sense, this is not surprising because when we have symmetric relations along all edges (i.e. $\mathcal{F}_{v \trianglelefteq e} = \mathcal{F}_{u \trianglelefteq e}$) then the harmonic space $\ker(\Delta_{\mathcal{F}})$ contain the same information as the kernel of the classical normalised graph Laplacian (see Proposition 30 in Appendix C). Importantly, the symmetry of the relations along all edges is a necessary condition in Theorem 16. As soon as we have an asymmetric transport map, we can find an arbitrarily small linear transformation $\mathbf{W}$ that increases the energy.

Proposition 17. *For any connected graph G and $\varepsilon > 0$, there exist a sheaf $(G, \mathcal{F}) \notin \mathcal{H}^d_{\mathrm{sym}}$, $\mathbf{W}$ with $\|\mathbf{W}\|_2 < \varepsilon$ and feature vector $\mathbf{x}$ such that $E_{\mathcal{F}}((\mathbf{I} \otimes \mathbf{W})\mathbf{x}) > E_{\mathcal{F}}(\mathbf{x})$.*

Beyond $O(d)$-bundles, we also have the following result for sheaves with stalks of dimension $d = 1$ which includes GCNs as a particular case [8, 39]. Denote by $\mathcal{H}^1_+ := \{(\mathcal{F}, G) \mid \mathcal{F}_{v \trianglelefteq e}\mathcal{F}_{u \trianglelefteq e} > 0\}$.

Theorem 18. *For $(\mathcal{F}, G) \in \mathcal{H}^1_+$ and σ being (Leaky)ReLU, $E_{\mathcal{F}}(\mathbf{Y}) \leq \lambda_* \|\mathbf{W}_1\|_2^2 \|\mathbf{W}_2^\top\|_2^2 E_{\mathcal{F}}(\mathbf{X})$.*

This proves that GCN/SCNs using Laplacians in this class also exponentially converge to $\ker(\Delta_{\mathcal{F}})$. Due to this result, we note that Propositons 10 and 12 also generalise to GCNs with infinite layers and Laplacians from $\mathcal{H}^d_{\mathrm{sym}} \subset \mathcal{H}^1_+$. Finally, we note the proof also holds for negatively-signed relations when σ is the identity.

Summary: Not only that sheaf diffusion is more expressive than heat diffusion as shown in Section 4, but SCNs are also more expressive than GCNs in the sense that they are generally not constrained to decrease the Dirichlet energy. This gives them greater control over their asymptotic behaviour than GCNs.

6 Neural Sheaf Diffusion and Sheaf Learning

In the previous sections we discussed the various advantages provided by sheaf diffusion and sheaf convolutions. However, in general, the ground truth sheaf is unknown or unspecified. Therefore, we aim to learn the underlying sheaf from data end-to-end, thus allowing the model to pick the right geometry for solving the task.

Neural Sheaf Diffusion. We propose the diffusion-type model from Equation 3. We note that by setting $\mathbf{W}_1, \mathbf{W}_2$ to identity and $\sigma(\mathbf{x}) = \mathrm{ELU}(\epsilon\mathbf{x})$ with $\epsilon > 0$ small enough or simply $\sigma = \mathrm{id}$, we

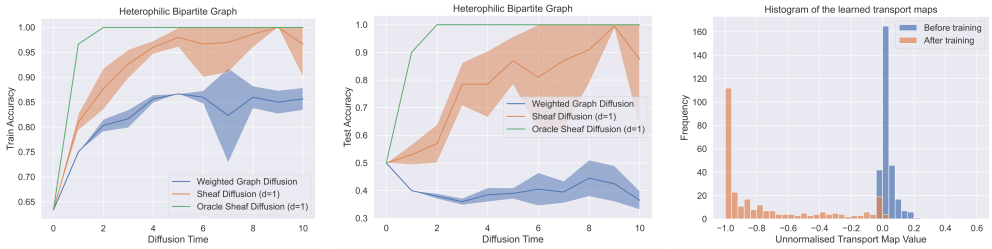

Figure 4: (*Left*) Train and (*Middle*) test accuracy as a function of diffusion time. (*Right*) Histogram of the learned scalar transport maps. The performance of the sheaf diffusion model is superior to that of weighted-graph diffusion and correctly learns to invert the features of the two classes.

recover (up to a scaling) the sheaf diffusion equation. Therefore, the model is at least as expressive as sheaf diffusion and benefits from all the positive properties outlined in Section 4.

$$\dot{\mathbf{X}}(t) = -\sigma\Big(\Delta_{\mathcal{F}(t)}(\mathbf{I}_n \otimes \mathbf{W}_1)\mathbf{X}(t)\mathbf{W}_2\Big), \tag{3}$$

Crucially, the sheaf Laplacian $\Delta_{\mathcal{F}(t)}$ is that of a sheaf $(G, \mathcal{F}(t))$ that *evolves over time*. More specifically, the evolution of the sheaf structure is described by a learnable function of the data $(G, \mathcal{F}(t)) = g(G, \mathbf{X}(t); \theta)$. This allows the model to make use of the latest available features to manipulate the underlying geometry of the graph and implicitly the behaviour of the diffusion process.

In our experiments, we focus on the time-discretised version of this model from Equation 4, which allows us to use a new set of weights at each layer t, while maintaining the nice theoretical properties of the model above. Additionally, we use an MLP to compute $\mathbf{X}(0)$ from the raw features and a final linear layer to perform the node classification.

$$\mathbf{X}_{t+1} = \mathbf{X}_t - \sigma\Big(\Delta_{\mathcal{F}(t)}(\mathbf{I} \otimes \mathbf{W}_1^t)\mathbf{X}_t\mathbf{W}_2^t\Big) \tag{4}$$

We note that this model is different from the SCN model from Equation 2 in two major ways. First, Hansen and Gebhart [22] used a *hand-crafted* sheaf with $d = 1$, constructed in a synthetic setting with full knowledge of the data generating process. In contrast, we *learn* a sheaf, which makes our model applicable to any real-world graph dataset, even in the absence of a sheaf structure. Additionally, motivated by our theoretical results, we make use of the full generality of sheaves by using stalks with $d \geq 1$ and higher-dimensional maps. Second, our model uses a more residual parametrisation of the discretised diffusion process, which empirically improves its performance.

Sheaf Learning. The advantage of learning a sheaf is that it does not require any sort of embedding of the nodes in an ambient space (as e.g. done in [9]). Instead, everything can be learned *locally*. Each $d \times d$ matrix $\mathcal{F}_{v \trianglelefteq e}$ is learned via a parametric matrix-valued function $\Phi : \mathbb{R}^{d \times 2} \to \mathbb{R}^{d \times d}, \mathcal{F}_{v \trianglelefteq e := (v,u)} = \Phi(\mathbf{x}_v, \mathbf{x}_u)$. In practice, we set $\Phi(\mathbf{x}_v, \mathbf{x}_u) = \sigma(\mathbf{W}[\mathbf{x}_v || \mathbf{x}_u])$ followed by a reshaping of the output. For simplicity, the equations above uses a single feature channel, but in practice, all channels are supplied as input. This function retains the inductive bias of locality specific to GNNs since it only utilises the features of the nodes forming the edge. At the same time, it is important that this function is non-symmetric in order to be able learn asymmetric transport maps along each edge. More generally, we can show that if the function Φ has enough capacity and the features are diverse enough, we can learn any sheaf over a graph.

Proposition 19. *Let $G = (V, E)$ be a finite graph with features $\mathbf{X}$. Then, if $(\mathbf{x}_v, \mathbf{x}_u) \neq (\mathbf{x}_w, \mathbf{x}_z)$ for any $(v, u) \neq (w, z) \in E$ and Φ is an MLP with sufficient capacity, Φ can learn any sheaf $(\mathcal{F}, G)$.*

First, this result formally motivates learning a sheaf at each layer since the model can learn to distinguish more nodes after each aggregation step. Second, this suggests that more expressive models (in the Weisfeiler-Lehman sense [5, 6, 34, 54]) could learn a more general family of sheaves. We leave the deeper investigation of these aspects for future work. In what follows, we distinguish between several types of functions Φ depending on the type of matrix they learn.

Diagonal. The main advantage of this parametrization is that fewer parameters need to be learned per edge and the sheaf Laplacian ends up being a matrix with diagonal blocks, which also results in fewer operations in sparse matrix multiplications. The main disadvantage is that the d dimensions of the stalks interact only via the left $\mathbf{W}_1$ multiplication.

Table 1: Results on node classification datasets sorted by their homophily level. Top three models are coloured by **First**, **Second**, **Third**. Our models are marked **NSD**.

	Texas	Wisconsin	Film	Squirrel	Chameleon	Cornell	Citeseer	Pubmed	Cora
Hom level	**0.11**	**0.21**	**0.22**	**0.22**	**0.23**	**0.30**	**0.74**	**0.80**	**0.81**
#Nodes	183	251	7,600	5,201	2,277	183	3,327	18,717	2,708
#Edges	295	466	26,752	198,493	31,421	280	4,676	44,327	5,278
#Classes	5	5	5	5	5	5	7	3	6
Diag-NSD	**85.67**±6.95	**88.63**±2.75	**37.79**±1.01	**54.78**±1.81	**68.68**±1.73	**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	**56.34**±1.32	**68.04**±1.58	**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	53.17±1.31	67.93±1.58	**85.68**±6.51	76.32±1.65	89.33±0.35	87.30±1.15
GGCN	**84.86**±4.55	86.86±3.29	37.54±1.56	**55.17**±1.58	**71.14**±1.84	**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	36.48±1.86	60.11±2.15	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	31.61±1.24	46.58±1.71	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	42.59±0.79	55.22±3.19	79.19±9.79	N/A	N/A	N/A
MixHop	77.84±7.73	75.88±4.90	32.22±2.34	43.80±1.48	60.50±2.53	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	38.47±1.58	63.86±3.04	77.86±3.79	**77.33**±1.48	**90.15**±0.43	**88.37**±1.25
Geom-GCN	66.76±2.72	64.51±3.66	31.59±1.15	38.15±0.92	60.00±2.81	60.54±3.67	**78.02**±1.15	**89.95**±0.47	85.35±1.57
PairNorm	60.27±4.34	48.43±6.14	27.40±1.24	50.44±2.04	62.74±2.82	58.92±3.15	73.59±1.47	87.53±0.44	85.79±1.01
GraphSAGE	82.43±6.14	81.18±5.56	34.23±0.99	41.61±0.74	58.73±1.68	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	53.43±2.01	64.82±2.24	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	40.72±1.55	60.26±2.50	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	28.77±1.56	46.21±2.99	81.89±6.40	74.02±1.90	75.69±2.00	87.16±0.37

Orthogonal. In this case, the model effectively learns a discrete vector bundle. Orthogonal matrices provide several advantages: (1) they are able to mix the various dimension of the stalks, (2) the orthogonality constraint prevents overfitting while reducing the number of parameters, (3) they have better understood theoretical properties, and (4) the resulting Laplacians are easier to normalise numerically since the diagonal entries correspond to the degrees of the nodes. In our model, we build orthogonal matrices from a composition of Householder reflections [33].

General. Finally, we consider the most general option of learning arbitrary matrices. The maximal flexibility provided by these maps can be useful, but it also comes with the danger of overfitting. At the same time, the sheaf Laplacian is more challenging to normalise numerically since one has to compute $D^{-1/2}$ for a positive semi-definite matrix D. To perform this at scale, one has to rely on SVD, whose gradients can be infinite if D has repeated eigenvalues. Therefore, this model is more challenging to train.

Computational Complexity. Treating the number of channels and layers as constant, a typical message passing GNN scales with $\mathcal{O}(n+m)$, where n is the number of nodes and m is the number of edges. A sheaf diffusion model with diagonal maps has complexity $\mathcal{O}(d(n+m))$ because the matrix multiplication required to compute the transport maps reduces to a scalar multiplication. When learning orthogonal or general $d \times d$ matrices, the complexity becomes $\mathcal{O}(d^3(n+m))$ because matrix multiplication is $\mathcal{O}(d^3)$. In practice, we use $1 \leq d \leq 5$ and all the operations above benefit from batched GPU computations in PyTorch [40] which effectively results in a constant overhead. For learning orthogonal matrices, we rely on the library Torch Householder [37] which provides support for fast transformations with very large batch sizes.

7 Experiments

Synthetic experiments. We consider a simple setup given by a connected bipartite graph with equally sized partitions. We sample the features from two overlapping isotropic Gaussian distributions in order to make the classes linearly non-separable at initialisation time. From Proposition 10 we know that diffusion models using symmetric restriction maps cannot separate the classes in the limit, while a diffusion process using negative transport maps can. Therefore, we consider two simple versions of the model from Equation 3 with $d = 1$, where we set $\mathbf{W}_1 = \mathbf{I}_d$, $\mathbf{W}_2 = \mathbf{I}_f$ and $\sigma = \mathrm{id}$. In the first, we learn general maps $\mathcal{F}_{v \trianglelefteq e} \in \mathbb{R}$. For the second model, we use a similar layer but constraint $\mathcal{F}_{v \trianglelefteq e} = \mathcal{F}_{u \trianglelefteq e}$, obtaining a weighted graph Laplacian.

Figure 4 presents the results across five seeds. As expected, for diffusion time zero (i.e. no diffusion), we see that a linear classifier cannot separate the classes. Also as expected, the diffusion process using symmetric maps cannot perfectly fit the data. In contrast, with the more general sheaf diffusion, as time increases and the signal approaches the harmonic space, the model gets better and the features become linearly separable. In the last subfigure we take a closer look at the sheaf that the model learns in the time limit by plotting a histogram of all the transport (scalar) maps $\mathcal{F}^{\top}_{v \trianglelefteq e}\mathcal{F}_{u \trianglelefteq e}$. In

accordance with Proposition 11, the model learns a negative transport map for all edges. This shows that the model manages to avoid oversmoothing (see Appendix F for more synthetic experiments).

Real-world experiments. We test our models on multiple real-world datasets [35, 41, 42, 45, 49] with an edge homophily coefficient h ranging from $h = 0.11$ (very heterophilic) to $h = 0.81$ (very homophilic). Therefore, they offer a view of how a model performs over this entire spectrum. We evaluate our models on the 10 fixed splits provided by Pei et al. [41] and report the mean accuracy and standard deviation. Each split contains $48\%/32\%/20\%$ of nodes per class for training, validation and testing, respectively. As baselines, we use an ample set of GNN models that can be placed in three categories: (1) classical: GCN [29], GAT [51], GraphSAGE [21]; (2) models specifically designed for heterophilic settings: GGCN [55], Geom-GCN [41], H2GCN [57], GPRGNN [12], FAGCN [4], MixHop [1]; (3) models addressing oversmoothing: GCNII [11], PairNorm [56]. All the results are taken from Yan et al. [55], except for FAGCN and MixHop, which come from Lingam et al. [31] and Zhu et al. [57], respectively. All of these were evaluated on the same set of splits as ours. In Appendix F we also include experiments with continous GNN models.

Results. From Table 1 we see that our models are first in $5/6$ benchmarks with high heterophily ($h < 0.3$) and second-ranked on the remaining one (i.e. Chameleon). At the same time, NSD also shows strong performance on the homophilic graphs by being within approximately 1% of the top model. Overall, NSD models are among the top three models on $8/9$ datasets. The $O(d)$-bundle diffusion model performs best overall confirming the intuition that it can better avoid overfitting, while also transforming the vectors in sufficiently complex ways. We also remark the strong performance of the model learning diagonals maps, despite the simpler functional form of the Laplacian.

8 Related Work, Discussion and Conclusion

Sheaf Neural Networks. Sheaf Neural Networks [22] with a *hand-crafted* sheaf Laplacian were originally introduced in a toy experimental setting. Since then, they have remained completely unexplored and we hope that this paper will fill this lacuna. In contrast to [22], we provide an ample theoretical analysis justifying the use of sheaves in Graph ML and study for the first time how sheaves can be *learned from data* using neural networks. Finally, we show the first successful application of sheaf neural networks on real-world datasets.

Heterophily and Oversmoothing. While good empirical designs jointly adressing these two problems have been proposed before [12, 55], Yan et al. [55] is the only other work connecting the two theoretically. Their analysis [55] is very different in terms of methods and assumptions and, therefore, their results are completely orthogonal. Concretely, the authors analyse the performance of linear SGCs [52] (i.e. GCN without nonlinearities) on random attributed graphs. In contrast, our analysis is not probabilistic, focuses on diffusion PDEs and also extends to GCNs in the non-linear regime. Furthermore, we employ a new set of mathematical tools from cellular sheaf theory, which brings a new language and new tools to analyse these problems. Perhaps the only commonality is that both works find evidence for the benefits of negatively signed edges in GNNs, although with different mathematical motivations. At the same time, other recent works [15, 32] have shown that GCNs with finite layers (typically one) can perform well in heterophilic graphs (including bipartite). This is in no contradiction with our results, which consider an *infinite time/layer* regime (i.e. not finite) and *perfect* linear separation (i.e. a model that cannot fit the data can still achieve high accuracy).

Limitations and societal impact. One of the main limitations of our theoretical analysis is that it does not address the generalisation properties of sheaves, but this remains a major impediment for the entire field of deep learning. Nonetheless, our setting was sufficient to produce many valuable insights about heterophily and oversmoothing and a basic understanding of what various types of sheaves can do. Much more theoretical work remains to be done in this direction, and we expect to see further cross-fertilization between ML and algebraic topology in the future. Finally, due to the theoretical nature of this work, we do not foresee any immediate negative societal impacts.

Conclusion. In this work, we used cellular sheaf theory to provide a novel topological perspective on heterophily and oversmoothing in GNNs. We showed that the underlying sheaf structure of the graph is intimately connected with both of these important factors affecting the performance of GNNs. To mitigate this, we proposed a new paradigm for graph representation learning where models do not only evolve the features at each layer but also the underlying geometry of the graph. In practice, we demonstrated that this framework can achieve state-of-the-art results in heterophilic settings.

References

[1] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Hrayr Harutyunyan, Nazanin Alipourfard, Kristina Lerman, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *The Thirty-sixth International Conference on Machine Learning (ICML)*, 2019. URL `http://proceedings.mlr.press/v97/abu-el-haija19a/abu-el-haija19a.pdf`.

[2] Afonso S Bandeira, Amit Singer, and Daniel A Spielman. A Cheeger inequality for the graph connection laplacian. *SIAM Journal on Matrix Analysis and Applications*, 34(4):1611–1630, 2013.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

[4] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *AAAI*. AAAI Press, 2021.

[5] Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yu Guang Wang, Pietro Liò, Guido Montúfar, and Michael Bronstein. Weisfeiler and Lehman Go Cellular: CW Networks. In *NeurIPS*, 2021.

[6] Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido F Montufar, Pietro Lió, and Michael Bronstein. Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks. In *ICML*, 2021.

[7] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *ICLR*, 2014.

[8] Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. *arXiv:2006.13318*, 2020.

[9] Benjamin Paul Chamberlain, James Rowbottom, Davide Eynard, Francesco Di Giovanni, Dong Xiaowen, and Michael M Bronstein. Beltrami flow and neural diffusion on graphs. In *NeurIPS*, 2021.

[10] Benjamin Paul Chamberlain, James Rowbottom, Maria Goronova, Stefan Webb, Emanuele Rossi, and Michael M Bronstein. Grand: Graph neural diffusion. In *ICML*, 2021.

[11] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1725–1735. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/chen20v.html`.

[12] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=n6jl7fLxrP`.

[13] Justin Michael Curry. *Sheaves, cosheaves and applications*. University of Pennsylvania, 2014.

[14] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, 2016.

[15] Lun Du, Xiaozhou Shi, Qiang Fu, Xiaojun Ma, Hengyu Liu, Shi Han, and Dongmei Zhang. Gbk-gnn: Gated bi-kernel graph neural networks for modeling both homophily and heterophily. In *Proceedings of the ACM Web Conference 2022*, pages 1550–1558, 2022.

[16] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *arXiv:2003.00982*, 2020.

[17] Robert Ghrist and Hans Riess. Cellular sheaves of lattices and the tarski laplacian. *arXiv:2007.04099*, 2020.

[18] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, 2017.

[19] Christoph Goller and Andreas Kuchler. Learning task-dependent distributed representations by backpropagation through structure. In *ICNN*, 1996.

[20] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *IJCNN*, 2005.

[21] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin*, 2017.

[22] Jakob Hansen and Thomas Gebhart. Sheaf neural networks. In *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*, 2020.

[23] Jakob Hansen and Robert Ghrist. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology*, 3(4):315–358, 2019.

[24] Jakob Hansen and Robert Ghrist. Opinion dynamics on discourse sheaves. *SIAM Journal on Applied Mathematics*, 81(5):2033–2060, 2021.

[25] Allen Hatcher. *Algebraic topology*. Cambridge Univ. Press, Cambridge, 2000.

[26] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[27] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[29] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[30] Hyun Deok Lee. On some matrix inequalities. *Korean Journal of Mathematics*, 16(4):565–571, 2008.

[31] Vijay Lingam, Rahul Ragesh, Arun Iyer, and Sundararajan Sellamanickam. Simple truncated svd based model for node classification on heterophilic graphs. *arXiv preprint arXiv:2106.12807*, 2021.

[32] Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Is heterophily a real nightmare for graph neural networks to do node classification? *arXiv preprint arXiv:2109.05641*, 2021.

[33] Zakaria Mhammedi, Andrew Hellicar, Ashfaqur Rahman, and James Bailey. Efficient orthogonal parametrisation of recurrent neural networks using householder reflections. In *International Conference on Machine Learning*, pages 2401–2409. PMLR, 2017.

[34] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and Leman go neural: Higher-order graph neural networks. In *AAAI*, 2019.

[35] Galileo Namata, Ben London, Lise Getoor, Bert Huang, and U Edu. Query-driven active surveying for collective classification. In *10th International Workshop on Mining and Learning with Graphs*, volume 8, page 1, 2012.

[36] H Nt and T Maehara. Revisiting graph neural networks: all we have is low pass filters. *arXiv preprint arXiv:1812.08434v4*, 2019.

[37] Anton Obukhov. Efficient householder transformation in pytorch, 2021. URL `www.github.com/toshas/torch-householder`. Version: 1.0.1, DOI: 10.5281/zenodo.5068733.

[38] K Oono and T Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2020.

[39] Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. *arXiv:1905.10947*, 2019.

[40] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[41] 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.

[42] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021.

[43] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Trans. Neural Networks*, 20(1):61–80, 2008.

[44] Richard D Schafer. *An introduction to nonassociative algebras*. Courier Dover Publications, 2017.

[45] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

[46] Allen Dudley Shepard. *A cellular description of the derived category of a stratified space*. PhD thesis, Brown University, 1985.

[47] Amit Singer and H-T Wu. Vector diffusion maps and the connection laplacian. *Communications on pure and applied mathematics*, 65(8):1067–1144, 2012.

[48] Alessandro Sperduti. Encoding labeled graphs by labeling RAAM. In *NIPS*, 1994.

[49] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 807–816, 2009.

[50] Loring W Tu. Manifolds. In *An Introduction to Manifolds*, pages 47–83. Springer, 2011.

[51] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[52] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr, Christopher Fifty, Tao Yu, and Kilian Q Weinberger. Simplifying graph convolutional networks. In *ICML*, 2019.

[53] Louis-Pascal Xhonneux, Meng Qu, and Jian Tang. Continuous graph neural networks. In *ICML*, 2020.

[54] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

[55] 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. *arXiv:2102.06462*, 2021.

[56] Lingxiao Zhao and Leman Akoglu. Pairnorm: Tackling oversmoothing in gnns. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=rkecl1rtwB`.

[57] 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, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] Proofs are included in the appendix
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code is provided with the supplementary
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Section 7 and Appendix E
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Appendix E
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] Appendix F
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The code of our submission.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]