
Dirichlet Energy Enhancement of Graph Neural Networks by Framelet Augmentation

Jialin Chen
Yale University
jialin.chen@yale.edu

Yuelin Wang
Shanghai Jiao Tong University
sjtu_wyl@sjtu.edu.cn

Cristian Bodnar
University of Cambridge
cb2015@cam.ac.uk

Rex Ying
Yale University
rex.ying@yale.edu

Pietro Liò
University of Cambridge
p1219@cam.ac.uk

Yu Guang Wang
Shanghai Jiao Tong University
yuguang.wang@sjtu.edu.cn

Abstract

Graph convolutions have been a pivotal element in learning graph representations. However, recursively aggregating neighboring information with graph convolutions leads to indistinguishable node features in deep layers, which is known as the over-smoothing issue. The performance of graph neural networks decays fast as the number of stacked layers increases, and the Dirichlet energy associated with the graph decreases to zero as well. In this work, we introduce a framelet system into the analysis of Dirichlet energy and take a multi-scale perspective to leverage the Dirichlet energy and alleviate the over-smoothing issue. Specifically, we develop a **Framelet Augmentation** strategy by adjusting the update rules with positive and negative increments for low-pass and high-passes respectively. Based on that, we design the **Energy Enhanced Convolution (EEConv)**, which is an effective and practical operation that is proved to strictly enhance Dirichlet energy. From a message-passing perspective, EEConv inherits multi-hop aggregation property from the framelet transform and takes into account all hops in the multi-scale representation, which benefits the node classification tasks over heterophilous graphs. Experiments show that deep GNNs with EEConv achieve state-of-the-art performance over various node classification datasets, especially for heterophilous graphs, while also lifting the Dirichlet energy as the network goes deeper.

1 Introduction

Many types of real-world data, such as social networks, recommendation systems, chemical molecules, contain indispensable relational information, and thus can be naturally represented as a graph. Recently, Graph Neural Networks (GNNs) [1–3] have achieved a myriad of eye-catching performances in multiple applications on graph-structured data. However, for traditional GCNs or other extensions of GNNs, there is a key limitation: the over-smoothing phenomenon, which means that the increase of the model’s depth gives rise to the decay of predictive performance.

There are mainly two types of approaches to enable deep graph neural networks. One is from empirical techniques in graph convolutional layers, like residual connections [4, 5], weight normalization [6], edge dropout [7], etc. The other controls Dirichlet energy to alleviate the over-smoothing phenomenon [8]. Dirichlet energy is a metric to measure the average distance between connected nodes in the feature space, however, rapidly converges to zero [9, 10] as the number of stacked layers increases. From a spectral perspective, recent works [11–13] discover that graph convolution works well for the case where the low-frequency components are sufficient for prediction, but fails in the scenarios where the high-frequency information is also necessary, which often happens in real-world heterophilous graphs. The failure is due to the denoising effect of graph convolution layers. The unsatisfying performance of GNNs usually stems from insufficient attention to high-frequency components, especially for heterophilous graphs. Therefore, the impact and the potential advantage of multi-scale representation on the over-smoothing issue deserve further exploration.

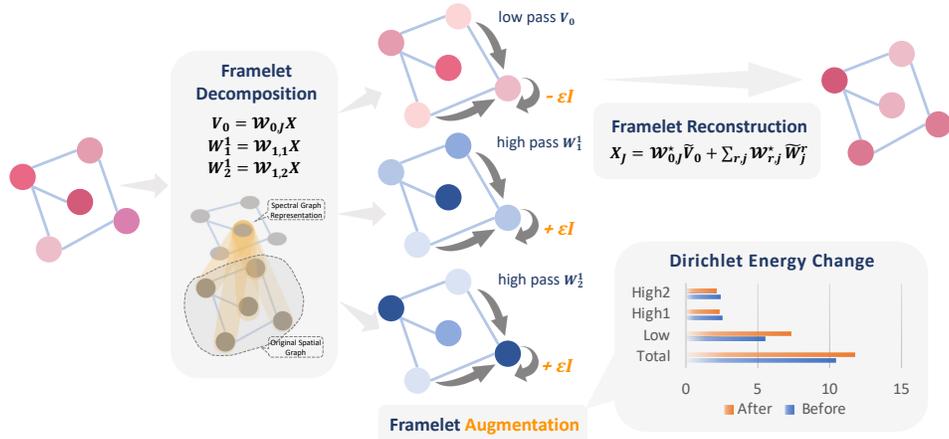


Figure 1: An illustration of the proposed Energy Enhanced Convolution. We first conduct framelet decomposition on the original graph signal (Eq. 1) and obtain one low-pass and two high-passes. The Framelet Augmentation is applied by adding or subtracting an increment for low and high-passes (Eq. 8). The total Dirichlet energy will be lifted in this process. A framelet reconstruction operator follows to resize the framelet coefficients to the original size.

Most existing methods that target over-smoothing only consider graph information in the spatial domain and do not characterize the asymptotic behaviors of different frequency components and their different contributions to over-smoothing problem. Besides, empirical techniques lack a theoretical guarantee of the stability or enhancement of the Dirichlet energy. Compared with the previous method that alleviates over-smoothing by controlling the Dirichlet energy, we are the first to theoretically guarantee the enhancement of Dirichlet energy. Furthermore, we emphasize a multi-scale representation for graph-structured data [14–16], to study asymptotic behaviors of different frequency components.

Present Work. We materialize this idea in a novel **Energy Enhanced Convolution (EEConv)** that can be repeatedly stacked to construct a more robust and deeper GNN architecture by lifting the Dirichlet energy to a higher and steady value. Figure 1 illustrates the computational flow of an EEConv layer. We first decompose the graph signal into framelet coefficients (Section 2), where the global graph structure and all-hop information are embedded by the framelet transform. Then, **Framelet Augmentation** is applied by modifying the corresponding diagonals of the adjacency matrices for low-pass and high-passes respectively (Section 3). Meanwhile, the Dirichlet energy associated with the graph is enhanced in this operation. Finally, the framelet coefficients will be reconstructed back to the original size and fed to the non-linear activation. Our proposed framelet augmentation strategy can be easily extended to other message-passing models with Laplacian-based propagation rules, such as heat diffusion on manifolds. We discuss possible extensions in Section 5.

We utilize the different roles and contributions of frequency components in graph prediction tasks to control Dirichlet energy. Low-frequency signals can make the representations of adjacent nodes similar and closer, while high-frequency signals make them more distant and distinguishable. Intuitively, we let the model reduce the focus to the low-pass information of the node itself, while increasing the focus to the high-pass components of the neighboring information. Moreover, the decomposability of Dirichlet energy provides us with the feasibility of regulating the energy ratio of each pass. It can be proved that Dirichlet energy is strictly enhanced with the framelet augmentation strategy.

To this end, the contributions of this work are threefold: (1) We perform a systematic analysis of the Dirichlet energy based on the framelet system and propose a novel Framelet Augmentation strategy to enhance the Dirichlet energy. (2) We theoretically prove the different asymptotic behaviors and Dirichlet energy of low-pass and high-passes during the feature propagation, and validate them through sufficient experiments. (3) We carry out experiments to verify the effectiveness of Framelet Augmentation and demonstrate that the proposed approach achieves outstanding performance on real-world node classification tasks, especially for heterophilous graphs.

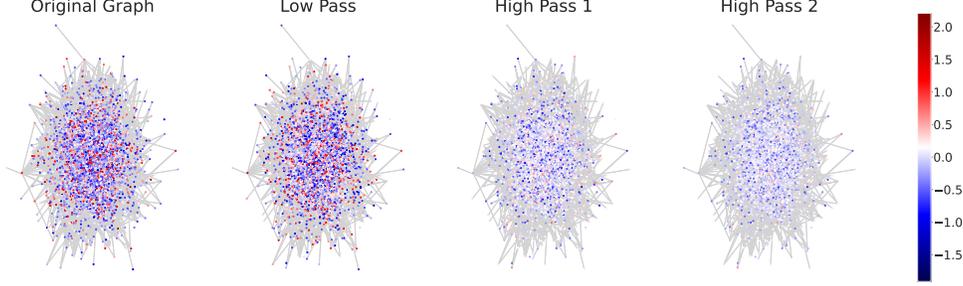


Figure 2: Visualization of framelet coefficients for node classification task on Cora. From left to right we show the original graph, low-pass, high-pass 1 and high-pass 2 respectively. The projected value is the first principal component of the high-dimensional features.

2 Background and Preliminaries

2.1 Framelet Analysis on Graph

Wavelet analysis on manifolds provides a powerful multi-scale representation tool for geometric deep learning. In this paper, we mainly focus on tight framelets on a graph [15, 17, 18], which is a multi-scale affine system. For a graph \mathcal{G} with N nodes and graph Laplacian Δ , let $U = [\mathbf{u}_1, \dots, \mathbf{u}_N]$ be the matrix of eigenvectors of Δ , whose transpose is used for the Graph Fourier Transform, and $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N)$ be the diagonal matrix of the eigenvalues. Framelets over the graph is generated by a set of *scaling functions* $\Phi = \{\alpha; \beta^{(1)}, \dots, \beta^{(n)}\} \subset L_1(\mathbb{R})$ associated with a *filter bank* $\eta = \{a; b^{(1)}, \dots, b^{(n)}\}$, satisfying $\widehat{\alpha}(2\xi) = \widehat{\alpha}(\xi)\widehat{\alpha}(\xi)$ and $\widehat{\beta}^{(r)}(2\xi) = \widehat{b}^{(r)}(\xi)\widehat{\alpha}(\xi)$, for any $\xi \in \mathbb{R}$, where $\widehat{h}(\xi)$ denotes the *Fourier transform* of h , which is defined by $\widehat{h}(\xi) := \sum_{k \in \mathbb{Z}} h(k)e^{-2\pi i k \xi}$. n denotes the number of high-pass filters. $\varphi_{j,p}(v)$ and $\psi_{j,p}^r(v)$ are the *low-pass* and *high-pass* framelets at node v associated to node p for *scale level* $j \in \{1, \dots, J\}$ respectively, which is defined by

$$\varphi_{j,p}(v) = \sum_{l=1}^N \widehat{\alpha} \left(\frac{\lambda_l}{2^j} \right) u_l(p)u_l(v); \quad \psi_{j,p}^r(v) = \sum_{l=1}^N \widehat{\beta}^{(r)} \left(\frac{\lambda_l}{2^j} \right) u_l(p)u_l(v), \quad r = 1, \dots, n.$$

Therefore, the framelet transforms actually take into account the global information and all the hops of the graph into its multi-scale representations. The low-pass and high-pass framelets distill the coarse-grained and fine-grained information of graph signals.

The *framelet coefficients* $V_0, W_j^r \in \mathbb{R}^{N \times d}$ are defined as the inner-product of the framelet and the graph signal $X \in \mathbb{R}^{N \times d}$, where d denotes the feature dimension. The size of V_0, W_j^r is the same as the graph signal (node features) X .

$$V_0 = \langle \varphi_{0,\cdot}, X \rangle = U^\top \widehat{\alpha} \left(\frac{\Lambda}{2} \right) U X \quad \text{and} \quad W_j^r = \langle \psi_{j,\cdot}^r, X \rangle = U^\top \widehat{\beta}^{(r)} \left(\frac{\Lambda}{2^{j+1}} \right) U X, \quad (1)$$

Let $\mathcal{W}_{r,j}$ denote the decomposition operators given by $V_0 = \mathcal{W}_{0,J} X$ and $W_j^r = \mathcal{W}_{r,j} X$. Then according to Eq. 1, we obtain the framelet transform matrices for decomposition:

$$\begin{aligned} \mathcal{W}_{0,J} &= U^\top \widehat{\alpha}(2^{-K+J-1}\Lambda) \dots \widehat{\alpha}(2^{-K}\Lambda) U := U^\top \Lambda_{0,J} U, \\ \mathcal{W}_{r,1} &= U^\top \widehat{b}^{(r)}(2^{-K}\Lambda) U := U^\top \Lambda_{r,1} U, \\ \mathcal{W}_{r,j} &= U^\top \widehat{b}^{(r)}(2^{-K+j-1}\Lambda) \widehat{\alpha}(2^{-K+j-2}\Lambda) \dots \widehat{\alpha}(2^{-K}\Lambda) U := U^\top \Lambda_{r,j} U. \end{aligned} \quad (2)$$

Here, K is a sufficiently large value such that the Laplacian's biggest eigenvalue $\lambda_{max} \leq 2^K \pi$. We use Haar-type filters, a classic multi-scale system with acceptable computational cost in our implementation. With Haar-type filters, we have $\widehat{\alpha}(\frac{\Lambda}{2}) = \cos(\frac{\Lambda}{8})\cos(\frac{\Lambda}{16})$, $\widehat{\beta}(\frac{\Lambda}{2}) = \sin(\frac{\Lambda}{8})\cos(\frac{\Lambda}{16})$ and $\widehat{\beta}(\frac{\Lambda}{4}) = \sin(\frac{\Lambda}{16})$ to construct a framelet system of 2 scale level ($j = 1, 2$) and 1 high-pass filter ($r = 1$). Thus, we obtain one low-pass (V_0) and two high-passes (W_1^1, W_2^1). Figure 2 shows the scattering plots of the principal component of framelet coefficients on the Cora dataset. We can observe that the low-pass provides an approximation of the original graph signal while the high-passes distill the detail information.

Energy Gap. The magnitude of the high-passes coefficients is usually much smaller than the low-pass. With L_2 norm as the energy of signals, we can prove that the sum of high-pass energies is less than that of the low-pass, or precisely, $\|W_1^1\|^2 + \|W_2^1\|^2 \leq \|V_0\|^2$. See the proof and L_2 norm statistical results in Figure 5 in Appendix A.2. This motivates us to consider the energy imbalance between low and high-passes.

2.2 Dirichlet Energy

Dirichlet Energy measures the degree of over-smoothing phenomenon, by calculating the average representation distance between connected nodes. Over-smoothing representations will produce a small value of Dirichlet Energy and cause the decay of the model’s prediction performance. Let $\tilde{A} = A + I_N$ be the adjacency matrix of the original graph with self-loops. \tilde{D} is the diagonal degree matrix associated with \tilde{A} . With the augmented adjacency matrix $\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$, augmented normalized Laplacian [19] of the input graph is defined as $\tilde{\Delta} = I_N - \hat{A} = I_N - \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$.

Definition 1 *Dirichlet energy $E(X)$ of the signal $X \in \mathbb{R}^{N \times 1}$ on the graph $\mathcal{G}(V, E)$ is defined as*

$$E(X) = X^\top \tilde{\Delta} X = \frac{1}{2} \sum_{(i,j) \in E} w_{ij} \left(\frac{X_i}{\sqrt{1+d_i}} - \frac{X_j}{\sqrt{1+d_j}} \right)^2,$$

where $\tilde{\Delta}$ is the augmented normalized Laplacian. Similarly, for multiple channels the Dirichlet energy is defined as $\text{trace}(X^\top \tilde{\Delta} X)$.

For the propagation rule of GCN [1]: $H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)})$, where $H^{(l)}$ is the feature representations of the l -th layer, $W^{(l)}$ is the l -th layer weight matrix, the following Theorem 1 [10] implies the convergent behavior of node features. The subspace that node features converge to is formulated with the bases of the eigenspace of graph Laplacian [9, 10]. We first clarify the notations for Theorem 1. Let K be the null space of the graph laplacian $\tilde{\Delta}$. The subspace \mathcal{M} is defined by $\mathcal{M} = K \otimes \mathbb{R}^d = \{\sum_{m=1}^M e_m \otimes w_m | w_m \in \mathbb{R}^d, e_m \in K\} \subseteq \mathbb{R}^{N \times d}$. d is the feature dimension of the graph signal. The distance between graph signal $X \in \mathbb{R}^{N \times d}$ and subspace \mathcal{M} is defined as $d_{\mathcal{M}}(X) = \inf_{m \in \mathcal{M}} \{\|X - m\|_F\}$, where F denotes the Frobenius norm.

Theorem 1 [10] *For GCN models, we have that $d_{\mathcal{M}}(H^{(l+1)}) \leq s_l \lambda d_{\mathcal{M}}(H^{(l)})$, where λ is the second largest eigenvalue of the augmented adjacency matrix \hat{A} and s_l is the supremum of all singular values of the l -th layer weight matrix $W^{(l)}$.*

The convergence rate of the distance between node features and the subspace is positively related to the eigenvalues of the \hat{A} [20], generating the consistent feature representations of nodes. over-smoothing is especially detrimental in heterophilous graph tasks, where adjacent nodes are more likely to have different labels. Thus, too similar feature representations between connected nodes (but most likely with different labels) lead to the failure of GNNs in these tasks.

3 Framelet Augmentation Strategy

3.1 Framelet Convolution

With the above Laplacian-based framelet transforms, we develop the framelet (graph) convolution similar to the graph convolution (GCNConv [1]) as follows:

$$H_{r,j}^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \mathcal{W}_{r,j} H^{(l)} W_{r,j}^{(l)}) \quad H^{(l+1)} = \mathcal{V}(H_{0,J}^{(l+1)}; H_{1,1}^{(l+1)}, \dots, H_{n,J}^{(l+1)}), \quad (3)$$

where $(r, j) \in \{(r, j) | r = 1, \dots, n; j = 1, \dots, J\} \cup \{(0, J)\}$, $W_{r,j}^{(l)}$ is the trainable weight matrix corresponding to the l -th layer and the (r, j) -th pass, \mathcal{V} is the framelet reconstruction operator given by $X_J = \mathcal{V}(V_0, W_{1,1}, \dots, W_{n,J}) = \mathcal{W}_{0,J}^* V_0 + \sum_{r=1}^n \sum_{j=1}^J \mathcal{W}_{r,j}^* W_{r,j}$, where the superscript \star indicates the conjugate transpose of the matrix. We can observe that \mathcal{V} reconstructs the low-pass and high-pass coefficients back to the original size.

Compared with the existing framelet graph model, UFG [15], which is a filter learning method in the frequency domain, our framelet graph convolution inherits the message-passing pattern and generalizes that to multi-scales representation systems.

3.2 Framelet Dirichlet Energy

Based on Section 2, we define framelet Dirichlet energy for low-pass and high-passes signals.

$$E_{0,J}(X) = (\mathcal{W}_{0,J}X)^\top \tilde{\Delta}(\mathcal{W}_{0,J}X); \quad E_{r,j}(X) = (\mathcal{W}_{r,j}X)^\top \tilde{\Delta}(\mathcal{W}_{r,j}X). \quad (4)$$

The total framelet Dirichlet energy is then defined as the sum of Dirichlet energy in each pass:

$$E_{total}(X) = \sum_{r=1}^n \sum_{j=1}^J E_{r,j}(X) + E_{0,J}(X).$$

Proposition 1 *The Dirichlet energy is conserved under framelet decomposition:*

$$E_{total}(X) = \sum_{r=1}^n \sum_{j=1}^J E_{r,j}(X) + E_{0,J}(X) = E(X). \quad (5)$$

Remark 1 *The Dirichlet energy components $E_{r,j}(X) := X^\top \mathcal{W}_{r,j}^\top \tilde{\Delta} \mathcal{W}_{r,j} X$ are controlled by $\Lambda_{r,j}^2$, the diagonal matrix given in Eq. 2, where $(r,j) \in \{(r,j) | r = 1, \dots, n; j = 1, \dots, J\} \cup \{(0,J)\}$.*

Proposition 1 and Remark 1 guarantee that we can decompose the signal into low-pass and high-passes and precisely control their Dirichlet energy proportions, without changing the overall Dirichlet energy. See the proofs in Appendix A.1.

3.3 Dirichlet Energy Enhancement Architecture

Energy Enhanced Convolution. The key idea to tackle the over-smoothing issue is to preserve the Dirichlet Energy and avoid its exponential decay to zero with respect to the number of layers. Motivated by this, we propose a **Framelet Augmentation** strategy, using the properties of multi-scale framelets to enhance the overall Dirichlet energy. To take advantage of the energy gap between low-pass and high-passes, we decouple the low-pass and high-passes propagation and modify the low-pass adjacency matrix \hat{A}^L and high-pass adjacency matrix \hat{A}^H separately. The augmented normalized Laplacian $\tilde{\Delta}$ is changed correspondingly, since $\tilde{\Delta} = I_N - \hat{A}$. ϵ controls the level of self-enhancement and impairment, which is a hyper-parameter in the implementation.

$$\hat{A}^L = \hat{D}^{-\frac{1}{2}}(\tilde{A} - \epsilon I)\hat{D}^{-\frac{1}{2}} = \tilde{A} - \epsilon \hat{D}^{-1}, \quad \hat{A}^H = \hat{D}^{-\frac{1}{2}}(\tilde{A} + \epsilon I)\hat{D}^{-\frac{1}{2}} = \tilde{A} + \epsilon \hat{D}^{-1}. \quad (6)$$

$$\tilde{\Delta}^L = I_N - \hat{A}^L = \tilde{\Delta} + \epsilon \hat{D}^{-1}, \quad \tilde{\Delta}^H = I_N - \hat{A}^H = \tilde{\Delta} - \epsilon \hat{D}^{-1}. \quad (7)$$

Next, with the modified adjacency matrices in the low-pass and high-passes, we have the following layer-wise propagation rule of **Energy Enhanced Convolution**:

$$\begin{aligned} H_{0,J}^{(l+1)} &= \sigma(\hat{A}^L \mathcal{W}_{0,J} H^{(l)} \mathcal{W}_{0,J}^{(l)}) \\ H_{r,j}^{(l+1)} &= \sigma(\hat{A}^H \mathcal{W}_{r,j} H^{(l)} \mathcal{W}_{r,j}^{(l)}), \quad \text{for } (r,j) \in \{(r,j) | r = 1, \dots, n; j = 1, \dots, J\} \\ H^{(l+1)} &= \mathcal{V}(H_{0,J}^{(l+1)}; H_{1,1}^{(l+1)}, \dots, H_{n,J}^{(l+1)}) \end{aligned} \quad (8)$$

Dirichlet Energy Enhancement. The low-pass component $E_{0,J}^\epsilon$ and high-pass components $E_{r,j}^\epsilon$ of Dirichlet energy with modified Laplacian are defined correspondingly as Eq. 9.

$$\begin{aligned} E_{0,J}^\epsilon(X) &= (\mathcal{W}_{0,J}X)^\top \tilde{\Delta}^L(\mathcal{W}_{0,J}X) = (\mathcal{W}_{0,J}X)^\top (\tilde{\Delta} + \epsilon \hat{D}^{-1})(\mathcal{W}_{0,J}X) \\ E_{r,j}^\epsilon(X) &= (\mathcal{W}_{r,j}X)^\top \tilde{\Delta}^H(\mathcal{W}_{r,j}X) = (\mathcal{W}_{r,j}X)^\top (\tilde{\Delta} - \epsilon \hat{D}^{-1})(\mathcal{W}_{r,j}X) \end{aligned} \quad (9)$$

The following theorem guarantees that we can obtain a strict enhancement of Dirichlet energy during the feature propagation by Framelet Augmentation.

Theorem 2 *The total framelet Dirichlet energy is increased with low-pass adjacency matrix \hat{A}^L and high-pass adjacency matrix \hat{A}^H when $\epsilon > 0$, i.e., $E_{total}^\epsilon(X) = \sum_{r=1}^n \sum_{j=1}^J E_{r,j}^\epsilon(X) + E_{0,J}^\epsilon(X) > E_{total}(X) = E(X)$.*

The proof of Theorem 2 is given in Appendix A.3. $\epsilon > 0$ indicates strengthening self-connection to the high-passes and weakening that to the low-pass.

3.4 Computational Complexity

To reduce the computational complexity caused by eigendecomposition for graph Laplacians, we use Chebyshev polynomials to approximate the framelet decomposition in our implementation. The framelet transform is then equivalent to left-multiplying a specific transformation matrix. We stack the transformation matrices to obtain a tensor-based framelet transform with the computational complexity of $\mathcal{O}(N^2(nJ + 1)d)$. N is the number of nodes, d is the feature dimension, n is the number of high-pass filters and J is the scale level of the low-pass. See Appendix C.4 for an empirical study of the complexity.

3.5 Asymptotic Behavior of Framelet Components

We can understand the effect of Framelet Augmentation in terms of the asymptotic behavior of framelet components. Framelet Augmentation helps to increase the weight of the high-frequency information of the node itself during the message passing process. It also reduces the proportion of high-pass component $E_{r,j}^\epsilon$ in the total Dirichlet energy, giving rise to the closer distances between the high-frequency components of node representations. The following proposition implies the asymptotic behaviors of low-pass and high-pass signals during the learning process.

Proposition 2 *Let A be an $n \times n$ augmented adjacency matrix, which is (symmetric) positive definite. $\lambda_k(A)$ is the k -th largest eigenvalue of A ($k = 1, 2, \dots, n$). Let $A(\epsilon) = A + \epsilon D$, where D is a positive diagonal matrix. Then $\lambda_k(A(\epsilon))$ increases monotonically with ϵ and the following relation holds:*

$$\lambda_k(A^L) \leq \lambda_k(A) \leq \lambda_k(A^H) \quad (\epsilon \geq 0),$$

where A^L and A^H are low-pass and high-passes adjacency matrices as defined in Eq. 6.

See proof of Proposition 2 and empirical study of asymptotic behavior of each pass (Figure 4) in Appendix A.4. According to Theorem 1, adding ϵI as a self-connectivity term in the high-pass increases its second largest eigenvalue of the adjacency matrix, leading to the slower convergence to the subspace and impeding the over-smoothing with an overall enhanced Dirichlet energy.

3.6 Equivariance of Framelet Convolution

Equivariance and Invariance are important properties for graph neural networks and we have the following Proposition.

Proposition 3 *An EEConv layer is permutation equivariant.*

See the proof in Appendix A.5. The framelet transforms are naturally generalized from the graph Fourier transform, therefore, framelet decomposition does not destroy the permutation invariance of graph neural networks. In hence, we can stack multiple EEConv layers, followed by a final invariant read-out function to obtain an equivariant deep graph neural network.

4 Experiments

To verify the effectiveness of Framelet Augmentation strategy, we evaluate: (A) Dirichlet energy behavior with respect to the number of layers and homophily level of the graphs and (B) the model’s performance of node classification over real-world datasets with different homophily levels and the change of performance with respect to the number of layers.

4.1 Dirichlet Energy Behavior

We select two real-world datasets to verify the effect of framelet augmentation for alleviating the exponentially decay of Dirichlet energy: Cora which is a relatively homophilous graph dataset with a homophily level of 0.81¹, and Chameleon with lower homophily level of 0.23. We show the layer-wise (logarithm of) Dirichlet energy during the feature propagation through GCN [1], GAT [3], FeaStNet [22], EGNN [8], FAGCN [11] and our Energy-enhanced UFG (EE-UFG) in Figure 3(a) and Figure 3(d). When ϵ is selected as 0, our EE-UFG is equivalent to the spatial version of UFG [15]. We can observe that the Dirichlet energy usually decays fast to zero with respect to the number

¹We use the homophily level defined in [21].

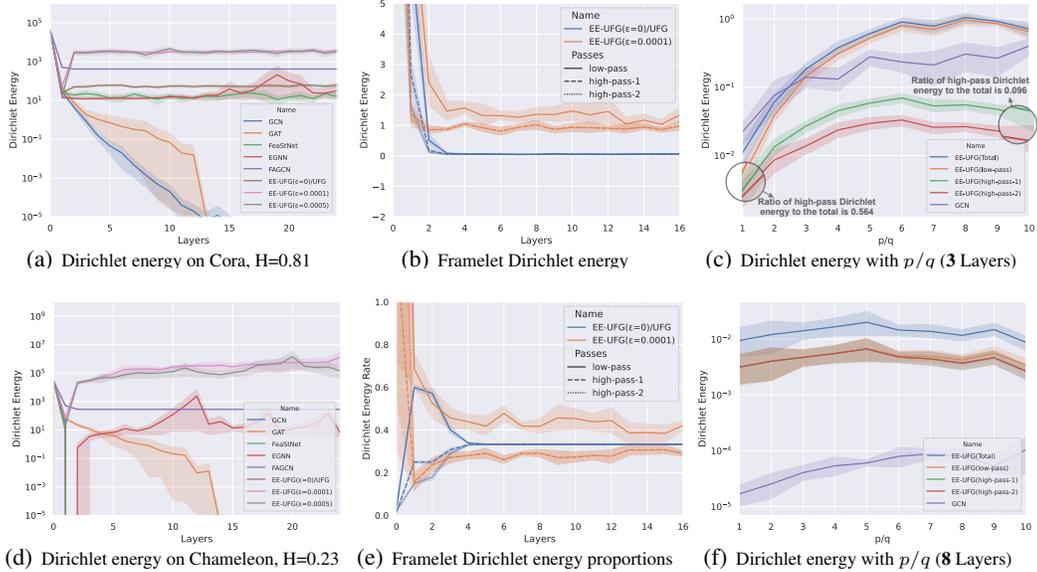


Figure 3: (a) Layer-wise Dirichlet energy with different models on Cora dataset. (b) Layer-wise framelet Dirichlet energy components. (c) Dirichlet energy over graphs with different p/q ratios by 3-layers model. (d) Layer-wise Dirichlet energy with different models on Chameleon dataset. (e) Layer-wise framelet Dirichlet energy ratios to the total Dirichlet energy. (f) Dirichlet energy over graphs with different p/q ratios by 8-layers model.

of layers in the regular graph convolutional models. In heterophilous graphs, some existing models suffer from Dirichlet energy instability (e.g., EGNN). With framelet augmentation, EE-UFG can lift the Dirichlet energy to a higher and steady state compared with other baseline models that target over-smoothing issues, thus making the output features distinguishable.

In Figure 3(b) and Figure 3(e), we plot the absolute value of framelet Dirichlet energy components and the ratios to the total Dirichlet energy with respect to the number of layers. For the case where $\epsilon = 0$, the low-pass component and high-passes components quickly decay to zero. One effect of the message-passing convolutions is that the proportions of high-passes and low-pass in the total Dirichlet energy tend to be the same, which means the message-passing mechanism automatically eliminates the energy gap between high-frequency and low-frequency components. However, the EE-UFG with framelet augmentation not only enhances the overall Dirichlet energy, but also decouples the low-pass and high-passes signals and preserves the energy gap during the feature propagation.

In Figure 3(c) and Figure 3(f), we use the Stochastic Block Model (SBM) to randomly generate undirected graphs with 100 nodes that are divided into 2 classes. The node features are sampled from Gaussian distribution $\mathcal{N}(0.5, 1)$ and $\mathcal{N}(-0.5, 1)$ for two classes. Edges are generated as Bernoulli random variables following intra-class connection probability p and inter-class connection probability q . The ratio of p/q depicts the homophily of the graph. The higher the ratio, the more homophilous the graph is. Figure 3(c) shows the Dirichlet energy with different p/q ratios. When using 3-layer models, which is empirically the optimal number of layers for classic GCN, Dirichlet energy decreases as the homophily decreases. This implies that heterophily and over-smoothing are correlated. Moreover, the ratio of high-passes Dirichlet energy to the total is 0.564 when $p/q = 1$ and decreases to 0.096 when $p/q = 10$. The high passes account for a larger proportion of the total Dirichlet energy in heterophilous graphs compared with homophilous cases, which indicates that high-pass information needs more attention in heterophilous graphs. A deeper network is essential for such graphs, because it creates a larger receptive field during feature propagation and can thus accept more information from the nodes with the same label. From Figure 3(f), we can observe that in a deeper GNN model, EE-UFG maintains a consistently higher Dirichlet energy with different p/q ratios than GCN.

4.2 Node Classification Performance

Real-world Datasets. We evaluate our proposed models for node classification tasks on nine real-world datasets: Texas, Wisconsin, Cornell introduced by [21], Squirrel, Chameleon introduced by [23] and Cora, PubMeb, CiteSeer introduced by [24] and ogb-arxiv introduced by [25]. The

homophily level ranges from 0.11 to 0.81, which measures the probability of connectivity between nodes with the same label in the graph. We test our model’s performance on the public split [24] and calculate the average test accuracy and standard deviation. For each dataset, all models are fine-tuned and tested on the same train/validation/test split.

Baselines. We select classic GNNs and state-of-the-art methods for heterophilous graphs and over-smoothing issue as our baselines: (1) classic GNN models: GCN [1], GAT [3], GraphSAGE [2], UFG [15]; (2) GNNs that can circumvent over-smoothing: GRAND [26], PairNorm [6], GCNII [27], EGNN [8]; (3) models for heterophilous graphs: FAGCN [11], MixHop [28]. We use the official codes provided by the authors for all baselines. The hyper-parameter search space for EE-UFG is given in Appendix C.5. ϵ is a hyper-parameter in our architecture and we search the parameter space to get the optimal value, which might be different for different tasks. It is also demonstrated in Figure 3(a) that Dirichlet energy is not sensitive to ϵ .

	Texas	Wisconsin	Squirrel	Chameleon	Cornell	Ogb-arxiv	CiteSeer	PubMed	Cora	Rank
Homophily level	0.11	0.21	0.22	0.23	0.30	0.63	0.74	0.80	0.81	
#Nodes	183	251	5201	2277	183	169343	3327	18717	2708	
#Edges	295	466	198493	31421	280	1166243	4676	44327	5278	
#Classes	5	5	5	5	5	40	7	3	6	
GCN	55.1±5.2	51.8±3.1	53.2±2.1	64.8±2.4	60.5±5.3	71.7±0.3	71.9±1.8	78.7±2.9	81.5±1.3	6.9
GAT	52.2±6.6	49.4±4.1	40.7±1.5	60.3±2.5	61.9±3.1	72.3±0.9	71.4±1.9	78.7±2.3	81.8±1.3	7.8
GraphSAGE	<u>82.4±6.1</u>	81.2±5.6	41.6±0.7	58.7±1.7	76.0±5.0	71.5±0.3	71.6±1.9	77.4±2.2	79.2±7.7	6.7
GRAND	75.7±3.3	79.4±3.6	40.1±1.5	54.7±2.5	<u>82.2±7.1</u>	72.2±0.2	<u>74.1±1.7</u>	78.8±1.7	83.6±1.0	5.7
PairNorm	60.3±4.3	48.4±6.1	50.4±2.0	62.7±2.8	58.9±3.2	70.4±1.3	73.6±1.5	78.3±0.4	82.3±1.0	7.2
GCNII	77.5±3.8	80.4±3.4	38.5±1.6	63.9±3.0	77.9±3.8	72.5±0.3	73.4±0.6	<u>80.3±0.4</u>	<u>85.5±0.5</u>	4.5
EGNN	81.0±0.8	88.6±3.2	48.3±2.3	62.7±2.6	83.8±4.6	<u>72.7±1.2</u>	70.4±2.8	80.1±3.6	85.7±3.7	<u>3.3</u>
FAGCN	82.4±6.9	82.9±7.9	42.6±0.8	55.2±3.2	79.2±3.2	70.6±0.8	72.7±0.8	79.4±0.3	84.1±0.5	5.0
MixHop	77.8±2.5	75.4±4.9	43.8±3.4	60.5±3.5	73.5±6.3	-	71.4±0.6	80.8±0.3	81.9±1.2	6.0
UFG	79.3±2.8	78.8±3.2	<u>53.3±1.5</u>	<u>66.9±1.1</u>	75.3±1.1	71.9±0.1	72.7±0.6	79.7±0.1	83.6±0.6	4.4
EE-UFG (ours)	<u>82.3±3.2</u>	<u>85.3±3.3</u>	55.3±1.3	68.0±0.9	<u>82.2±2.8</u>	73.2±3.8	74.2±1.3	79.4±0.9	83.5±0.2	2.2

Table 1: Node classification performance comparison. Best result in **bold** and second best underlined. "-" denotes out of memory or inapplicable.

	Chameleon (H=0.23)				Cornell (H=0.30)				CiteSeer (H=0.74)				Cora (H=0.81)			
#Layer	2	8	16	32	2	8	16	32	2	8	16	32	2	8	16	32
GCN	63.2	58.9	50.2	32.4	60.5	56.4	44.3	28.9	68.7	33.6	28.7	23.1	81.5	35.8	28.5	22.0
UFG	66.2	58.8	53.4	47.7	74.3	65.2	58.4	53.5	71.3	51.2	46.8	40.4	75.1	79.4	57.1	39.1
PairNorm	62.4	54.1	46.4	33.7	50.3	58.4	57.2	57.9	73.6	70.3	58.4	35.8	74.5	81.6	82.3	60.3
GCNII	60.7	62.5	58.7	42.8	67.6	63.2	77.8	76.4	68.2	70.6	72.9	73.4	82.2	84.2	84.6	85.4
EE-UFG	66.2	68.0	63.5	63.5	75.0	82.2	81.3	79.2	64.8	73.6	73.8	72.4	83.5	82.4	83.5	81.4

Table 2: Performance comparison for GCN, UFG and EE-UFG with fix number of layers on three citation network datasets. The best result of each model is highlighted in **Bold**.

Results. Table 1 shows the performance comparison on nine node classification tasks. We can observe that for heterophilous tasks, EE-UFG obtains a great boost compared with baselines, by better extracting the high-frequency information of the node itself. Our model ranks top 2 over seven real-world datasets with $H < 0.8$ that are moderately or highly heterophilous. Over-smoothing issue is especially detrimental in heterophilous graph tasks, where multi-hop and deeper GNNs are necessary. In heterophilous graphs, the aggregated information from adjacent nodes contains more high-frequency information, thus, the high-pass components of the node itself should be better focused. Besides, EE-UFG inherits multi-hop aggregation properties from the framelet transform, taking into account all hops in the multi-scale framelet representation, which is essential for heterophilous graphs. The experimental results emphasize our model’s advantage over heterophilous graphs.

It is known that the performance of GNNs will rapidly decay as the layers are stacked too much. The GCN-row in Table 2 verifies this phenomenon. We can observe from Table 2 that the UFG suffers less over-smoothing than classic GCN, partly due to its adaptive filter learning in the frequency domain. Other baselines, such as PairNorm, GCNII, alleviate the over-smoothing issue to some extent, which however sacrifices performance, especially for heterophilous graphs. Without Framelet Augmentation, the performance of GNNs may begin to drop before it reaches optimal performance. The effect of Framelet Augmentation here is to delay the performance decay so that it can achieve the

best performance with an appropriate number of stacked layers. Our proposed EE-UFG can basically circumvent over-smoothing and achieve better and more stable performance as the number of layers increases. From Table 2, we can observe that our model with 32 layers can still perform better than the best performance of other baselines on heterophilous datasets (e.g., Chameleon and Cornell). Besides, we can see that the best performance of EE-UFG occurs at a deeper layer.

5 Discussion and Extension

Framelet Systems on Manifold. Framelet systems can be well applied to manifold signals, $f \in L^2(\mathcal{M})$. Akin to the graph Laplacian, for a given manifold \mathcal{M} , we consider its Laplace-Beltrami operator \mathcal{L}_B which is defined as $\mathcal{L}_B f = -\text{div}(\nabla f)$. \mathcal{L}_B^L and \mathcal{L}_B^H can then be defined similarly as Eq. 7 respectively. In general, our proposed framelet augmentation method can be naturally extended to any other (symmetric) Laplacian-based propagation rules, using the framelet theory on manifolds. More details about Framelet extension on manifolds are given in Appendix B.

Limitations. Framelet augmentation is based on a symmetric Laplacian and a symmetric adjacency matrix, which is the general case. However, for some specific cases in geometric deep learning, such as simplicial complexes, non-square boundary matrices are involved to relate the signals between simplices of different dimensions. In such cases, our framelet augmentation can not be implemented. We will consider framelet augmentation strategy for these cases in future work. Besides, the computational complexity of framelet transform is $\mathcal{O}(N^2(nJ + 1))$ which is a bit high.

6 Related Work

Over-smoothing and Dirichlet Energy. One of the widely known plights of GNNs is over-smoothing, which has been studied by [4, 5, 10, 12, 29]. The Dirichlet energy was commonly used in these studies. Explanation paying attention to the structure of Laplacian has been undergone by [4, 9, 10, 30]. A large part of the methods come from empirical techniques in graph convolutional layers, like relieving the adjacent matrix by sparsification [7], scaling node representations to avoid features caught into the invariant regime [8], adding residual connections [4, 5, 13]. Several other empirical methods have been studied recently, like weight normalization [6], edge dropout [7], etc. Many other attempts beyond the graph matrix analysis also emerged like GCON [31] using ODE dynamics, GRAND [26] and PDE-GCN [32] regarding GNNs as continuous diffusion processes. Besides, in the field of spectral analysis, GNNs’ updating process can be viewed as tackling low-frequency information [11, 11, 19]. However, these empirical techniques lack a necessary theoretical guarantee. Another type of method is controlling Dirichlet energy to alleviate the over-smoothing issue, e.g., EGNN. However, Figure 3 (d) shows that EGNN suffers from Dirichlet energy instability in some heterophilous cases. In contrast, to our best knowledge, we are the first to theoretically prove the enhancement of Dirichlet energy, taking advantage of multi-scale graph representation.

Wavelet Analysis on Graphs. [33] firstly proposed a formal approach to spatial traffic analysis on the wavelet transform. Polynomials of a differential operator were used to build a multi-scale tight frame by [16]. [34] gave the tight framelets framework on manifolds, which was then extended to graphs by [17, 35] with the fast decomposition and reconstruction algorithms on undecimated and decimated frames for graph signals. In the regime of signal processing, [36] established a tree-based wavelet system with localization properties, which is a milestone in the multi-resolution analysis. [37] apply harmonic analysis to semi-supervised learning and construct Haar-like bases for it. [38–40] used the Haar-like wavelets system [41] to cope with deep learning tasks.

7 Conclusion

In this work, we develop a framelet analysis on graphs and generalize the generic graph convolution to a framelet version. Due to the energy difference between the low-pass and high-passes, we originally propose framelet augmentation which is surprisingly discovered to increase the Dirichlet Energy associated with the graph and keep it at a high and steady value. In practice, we demonstrate the behavior of framelet features during the training and the effectiveness of framelet augmentation to relieve the over-smoothing problem. Experimental Results also show that the proposed EE-UFG achieve excellent performance on node classification tasks.

References

- [1] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ICLR*, 2016. 1, 4, 6, 8, 18
- [2] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *NeurIPS*, 30, 2017. 8, 18
- [3] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lió, and Yoshua Bengio. Graph attention networks. *ICLR*, 2017. 1, 6, 8, 18
- [4] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. DeepGCNs: Can GCNs go as deep as CNNs? In *CVPR*, pages 9267–9276, 2019. 1, 9
- [5] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *ICML*, pages 5453–5462, 2018. 1, 9
- [6] Lingxiao Zhao and Leman Akoglu. PairNorm: tackling oversmoothing in GNNs. In *ICLR*, 2020. URL <https://openreview.net/forum?id=rkecl1rtwB>. 1, 8, 9, 18
- [7] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. DropEdge: towards deep graph convolutional networks on node classification. In *ICLR*, 2020. URL <https://openreview.net/forum?id=Hkx1qkrKPr>. 1, 9
- [8] Kaixiong Zhou, Xiao Huang, Daochen Zha, Rui Chen, Li Li, Soo-Hyun Choi, and Xia Hu. Dirichlet energy constrained learning for deep graph neural networks. *NeurIPS*, 34, 2021. 1, 6, 8, 9, 18
- [9] Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. *ICML Workshop on Graph Representation Learning*, 2020. 1, 4, 9
- [10] Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *ICLR*, 2020. 1, 4, 9, 15
- [11] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *AAAI*, 2021. 1, 6, 8, 9, 18
- [12] Hoang Nt and Takanori Maehara. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019. 9
- [13] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *AAAI*, volume 34, pages 3438–3445, 2020. 1, 9
- [14] Bingbing Xu, Huawei Shen, Qi Cao, Yunqi Qiu, and Xueqi Cheng. Graph wavelet neural network. In *ICLR*, 2019. 2
- [15] Xuebin Zheng, Bingxin Zhou, Junbin Gao, Yu Guang Wang, Pietro Lió, Ming Li, and Guido Montúfar. How framelets enhance graph neural networks. In *ICML*, 2021. 3, 4, 6, 8, 18
- [16] Mauro Maggioni and Hrushikesh N Mhaskar. Diffusion polynomial frames on metric measure spaces. *Applied and Computational Harmonic Analysis*, 24(3):329–353, 2008. 2, 9
- [17] Bin Dong. Sparse representation on graphs by tight wavelet frames and applications. *Applied and Computational Harmonic Analysis*, 42(3):452–479, 2017. 3, 9
- [18] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *AAAI*, 2018. 3
- [19] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *ICML*, pages 6861–6871, 2019. 4, 9
- [20] Wenbing Huang, Yu Rong, Tingyang Xu, Fuchun Sun, and Junzhou Huang. Tackling over-smoothing for general graph convolutional networks. *arXiv preprint arXiv:2008.09864*, 2020. 4
- [21] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-GCN: geometric graph convolutional networks. In *ICLR*, 2020. 6, 7, 18
- [22] Nitika Verma, Edmond Boyer, and Jakob Verbeek. Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *CVPR*, pages 2598–2606, 2018. 6

- [23] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021. 7, 18
- [24] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pages 40–48. PMLR, 2016. 7, 8, 18
- [25] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies*, 1(1):396–413, 2020. 7, 18
- [26] Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. Grand: Graph neural diffusion. In *International Conference on Machine Learning*, pages 1407–1418. PMLR, 2021. 8, 9, 18
- [27] 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. 8, 18
- [28] 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. 8, 18
- [29] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *ICLR*, 2019. 9
- [30] Sitao Luan, Mingde Zhao, Xiao-Wen Chang, and Doina Precup. Break the ceiling: Stronger multi-scale deep graph convolutional networks. In *NeurIPS*, volume 32, 2019. 9
- [31] T Konstantin Rusch, Benjamin P Chamberlain, James Rowbottom, Siddhartha Mishra, and Michael M Bronstein. Graph-coupled oscillator networks. *arXiv preprint arXiv:2202.02296*, 2022. 9
- [32] Moshe Eliasof, Eldad Haber, and Eran Treister. Pde-gcn: Novel architectures for graph neural networks motivated by partial differential equations. *Advances in Neural Information Processing Systems*, 34:3836–3849, 2021. 9
- [33] Mark Crovella and Eric Kolaczyk. Graph wavelets for spatial traffic analysis. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, volume 3, pages 1848–1857. IEEE, 2003. 9
- [34] Yu Guang Wang and Xiaosheng Zhuang. Tight framelets and fast framelet filter bank transforms on manifolds. *Applied and Computational Harmonic Analysis*, 48(1):64–95, 2020. 9
- [35] Xuebin Zheng, Bingxin Zhou, Yu Guang Wang, and Xiaosheng Zhuang. Decimated framelet system on graphs and fast G-framelet transforms. *Journal of Machine Learning Research*, 23(18):1–68, 2022. 9
- [36] Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989. 9
- [37] Matan Gavish, Boaz Nadler, and Ronald R Coifman. Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning. In *ICML*, 2010. 9
- [38] Yu Guang Wang, Ming Li, Zheng Ma, Guido Montufar, Xiaosheng Zhuang, and Yanan Fan. Haar graph pooling. In *ICML*, pages 9952–9962, 2020. 9
- [39] Ming Li, Zheng Ma, Yu Guang Wang, and Xiaosheng Zhuang. Fast haar transforms for graph neural networks. *Neural Networks*, 128:188–198, 2020.
- [40] Xuebin Zheng, Bingxin Zhou, Ming Li, Yu Guang Wang, and Junbin Gao. Mathnet: Haar-like wavelet multiresolution-analysis for graph representation and learning. *arXiv preprint arXiv:2007.11202*, 2020. 9
- [41] Charles K Chui, F Filbir, and Hrushikesh N Mhaskar. Representation of functions on big data: graphs and trees. *Applied and Computational Harmonic Analysis*, 38(3):489–509, 2015. 9

- [42] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 16
- [43] Cristian Bodnar, Francesco Di Giovanni, Benjamin Paul Chamberlain, Pietro Liò, and Michael M Bronstein. Neural Sheaf Diffusion: A topological perspective on heterophily and oversmoothing in GNNs. *arXiv preprint arXiv:2202.04579*, 2022. 17
- [44] Jakob Hansen and Thomas Gebhart. Sheaf neural networks. *arXiv preprint arXiv:2012.06333*, 2020. 17
- [45] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 18
- [46] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019. 18
- [47] 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. 18
- [48] 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 preprint arXiv:2102.06462*, 2021. 18
- [49] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 20
- [50] Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Moritz Hardt, Ben Recht, and Ameet Talwalkar. Massively parallel hyperparameter tuning. 2018. 20
- [51] Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E Gonzalez, and Ion Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018. 20

A Theoretical Support

A.1 Framelet Dirichlet Energy Conservation

Here, we give the proof of proposition and remark mentioned in Section 3.

Proposition 4 *The Dirichlet energy is conserved under framelet decomposition:*

$$E_{total}(X) = E(X). \quad (10)$$

Proof. Let $(r, j) \in \{(r, j) | r = 1, \dots, n; j = 1, \dots, J\} \cup \{(0, J)\}$,

$$\begin{aligned} E_{total}(X) &= \sum_{r,j} X^\top \mathcal{W}_{r,j}^\top \tilde{\Delta} \mathcal{W}_{r,j} X \\ &= \sum_{r,j} X^\top U^\top \Lambda_{r,j} U U^\top \Lambda U U^\top \Lambda_{r,j} U X \\ &= \sum_{r,j} X^\top U^\top \Lambda_{r,j}^2 \Lambda U X \\ &= \sum_{r,j} \sum_i (UX)_i^2 (\lambda_{r,j}^i)^2 \lambda_i \\ &= \sum_i (UX)_i^2 \lambda_i \\ &= X^\top U^\top \Lambda U X \\ &= E(X), \end{aligned}$$

where $(UX)_i$ is the i th component of UX , and $\lambda_i, \lambda_{r,j}^i$ are the eigen-values of $\Lambda, \Lambda_{r,j}$. The fifth equality is because of the relation $\sum_{r,j} (\lambda_{r,j}^i)^2 = 1, \Lambda_{r,j}$ as follows,

$$\begin{aligned} \Lambda_{0,J} &= \widehat{a}(2^{-K+J-1}\Lambda) \cdots \widehat{a}(2^{-K}\Lambda), \\ \Lambda_{r,1} &= \widehat{b}^{(r)}(2^{-K+j-1}\Lambda), \\ \Lambda_{r,j} &= \widehat{b}^{(r)}(2^{-K+j-1}\Lambda) \widehat{a}(2^{-K+j-2}\Lambda) \cdots \widehat{a}(2^{-K}\Lambda). \end{aligned}$$

Therefore, the Dirichlet energy is preserved after the framelet decomposition. \blacksquare

Remark 2 *The dirichlet energy components $E_{r,j}(X) := X^\top \mathcal{W}_{r,j}^\top \tilde{\Delta} \mathcal{W}_{r,j} X$ are controlled by $\Lambda_{r,j}^2$, the diagonal matrix given in Eq. 2, where $(r, j) \in \{(r, j) | r = 1, \dots, n; j = 1, \dots, J\} \cup \{(0, J)\}$.*

Proof. Using $E_{r,j}(X) := X^\top \mathcal{W}_{r,j}^\top \tilde{\Delta} \mathcal{W}_{r,j} X$, we obtain

$$\min_i \{(\lambda_{r,j}^i)^2\} E(X) \leq E_{r,j}(X) \leq \max_i \{(\lambda_{r,j}^i)^2\} E(X) \leq 4E,$$

where we use that the eigenvalues of the normalized Laplacian are in the range of $[0, 2]$, $\lambda_{r,j}^i$ is the i th eigenvalue of $\Lambda_{r,j}$. \blacksquare

A.2 Framelet Energy Gap

In this part, we prove the energy gap between low-pass and high-pass coefficients, with the specific Haar-type filters. We consider L_2 norm of feature X as its energy.

Proposition 5 *In the framelet system of 2 scales ($j = 1, 2$) and 1 high-pass ($r = 1$) with Haar-type filters, the energy of the low-pass is larger than the sum of the energy of the high-passes, i.e. $\|W_1^1\|^2 + \|W_2^1\|^2 \leq \|V_0\|^2$.*

Proof. With the relations that $\begin{cases} \widehat{\alpha}(2\xi) = \widehat{a}(\xi)\widehat{\alpha}(\xi) \\ \widehat{\beta}(2\xi) = \widehat{b}(\xi)\widehat{\alpha}(\xi) \end{cases}$ and $\begin{cases} \widehat{a}(\xi) = \cos(\xi/2) \\ \widehat{b}(\xi) = \sin(\xi/2) \end{cases}$, we obtain

$$\frac{\widehat{\beta}(2\xi)}{\widehat{\alpha}(2\xi)} = \frac{\widehat{b}(\xi)}{\widehat{a}(\xi)} = \tan\left(\frac{\xi}{2}\right).$$

As the framelets constitute a tight frame, we have the Parseval identity $\|\widehat{W}_1^1\|^2 + \|\widehat{V}_0\|^2 = \|\widehat{X}\|^2$. Thus, we can obtain the explicit expression of $\widehat{\alpha}$ and $\widehat{\beta}$ as follows,

$$\widehat{\alpha}(\Lambda/2) = \cos(\Lambda/8) \cos(\Lambda/16), \quad \widehat{\beta}(\Lambda/2) = \sin(\Lambda/8) \cos(\Lambda/16), \quad \widehat{\beta}(\Lambda/4) = \sin(\Lambda/16).$$

This implies the energy difference between low-pass and high-passes reads

$$\|\widehat{V}_0\|^2 - \|\widehat{W}_1^1\|^2 - \|\widehat{W}_2^1\|^2 = \|\widehat{X}\|^2 \left(\|\widehat{\alpha}(\Lambda/2)\|^2 - \|\widehat{\beta}(\Lambda/2)\|^2 - \|\widehat{\beta}(\Lambda/4)\|^2 \right) \quad (11)$$

The RHS of (11) equals to $\cos^2(\frac{\Lambda}{8})\cos^2(\frac{\Lambda}{16}) - \sin^2(\frac{\Lambda}{8})\cos^2(\frac{\Lambda}{16}) - \sin^2(\frac{\Lambda}{16})$. Since the eigenvalues of the normalized Laplacian are in the range of $[0, 2]$, it can be easily verified that the above trigonometric function is always larger than zero. This then gives $\|\widehat{W}_1^1\|^2 + \|\widehat{W}_2^1\|^2 \leq \|\widehat{V}_0\|^2$. ■

Figure 5 shows the L_2 norms of low-pass, high-passes, and the sum of high-passes of datasets with different homophily levels. It empirically verified that there exists an energy imbalance between low and high-passes, which inspires our energy enhancement strategy.

A.3 Dirichlet Energy Enhancement

Next, we show how the Dirichlet energy is enhanced with framelet augmentation.

Proposition 6 For $\epsilon > 0$, the total framelet Dirichlet energy is increased with low-pass adjacency matrix \widehat{A}^L and high-pass adjacency matrix \widehat{A}^H , i.e., $E_{total}^\epsilon(X) > E_{total}(X) = E(X)$.

Proof.

$$\begin{aligned} E_{total}^\epsilon(X) &= \sum_{r,j} E_{r,j}^\epsilon(X) + E_{0,J}^\epsilon(X) \\ &= \sum_{r,j} (\mathcal{W}_{r,j}X)^\top (\tilde{\Delta} - \epsilon\widehat{D}^{-1})(\mathcal{W}_{r,j}X) + (\mathcal{W}_{0,J}X)^\top (\tilde{\Delta} + \epsilon\widehat{D}^{-1})(\mathcal{W}_{0,J}X) \\ &= \sum_{r,j} X^\top U^\top \Lambda_{r,j} U U^\top (\Lambda - \epsilon\widehat{D}^{-1}) U U^\top \Lambda_{r,j} U X + X^\top U^\top \Lambda_{0,J} U U^\top (\Lambda + \epsilon\widehat{D}^{-1}) U U^\top \Lambda_{0,J} U X \\ &= \left(\epsilon X^\top U^\top \widehat{D}^{-1} \Lambda_{0,J}^2 U X - \sum_{r,j} \epsilon X^\top U^\top \widehat{D}^{-1} \Lambda_{r,j}^2 U X \right) \\ &\quad + \left(X^\top U^\top \Lambda \Lambda_{0,J}^2 U X + \sum_{r,j} X^\top U^\top \Lambda \Lambda_{r,j}^2 U X \right) \\ &= \epsilon X^\top U^\top \widehat{D}^{-1} \left(\Lambda_{0,J}^2 - \sum_{r,j} \Lambda_{r,j}^2 \right) U X + E(X). \end{aligned}$$

By Proposition 5 and its specific framelet system, $\Lambda_{0,J}^2 - \sum_{r,j} \Lambda_{r,j}^2 \geq 0$, thus, $E_{total}^\epsilon(X) \geq E(X)$. ■

A.4 Asymptotic Behavior of EE-UFG

Proposition 7 Let A be an $n \times n$ augmented adjacency matrix, which is (symmetric) positive definite. Let $\lambda_k(A)$ be the k -th largest eigenvalue of A ($k = 1, 2, \dots, n$), and $A(\epsilon)$ denote $A + \epsilon D$, where D is a positive diagonal matrix. Then, $\lambda_k(A(\epsilon))$ increases monotonically with ϵ and the following relation holds:

$$\lambda_k(A^L) \leq \lambda_k(A) \leq \lambda_k(A^H) \quad \text{for } \epsilon \geq 0,$$

where A^L and A^H are low-pass and high-passes adjacency matrices as defined in Eq. 6.

Proof. By Eq. 6, we know that $A^L = \widehat{A} - \epsilon\widehat{D}^{-1}$ and $A^H = \widehat{A} + \epsilon\widehat{D}^{-1}$, where \widehat{D}^{-1} is a positive diagonal matrix. For symmetric matrices, we have the Courant-Fischer min-max theorem:

$$\lambda_k(A) = \min\{\max\{R_A(x) | x \in U \text{ and } x \neq 0\} | \dim(U) = k\}$$

with $R_A(x) = \frac{\langle Ax, x \rangle}{\langle x, x \rangle}$. We have $R_{A+B}(x) = \frac{\langle (A+B)x, x \rangle}{\langle x, x \rangle} = \frac{\langle Ax, x \rangle}{\langle x, x \rangle} + \frac{\langle Bx, x \rangle}{\langle x, x \rangle} > \max\{R_A(x)\}$, if B is positively definite. Thus, we have $\lambda_k(A + \epsilon\widehat{D}^{-1}) \geq \lambda_k(A)$. Similarly, $\lambda_k(A - \epsilon\widehat{D}^{-1}) \leq \lambda_k(A)$. Therefore, $\lambda_k(A^L) \leq \lambda_k(A) \leq \lambda_k(A^H)$ holds when $\epsilon \geq 0$. ■

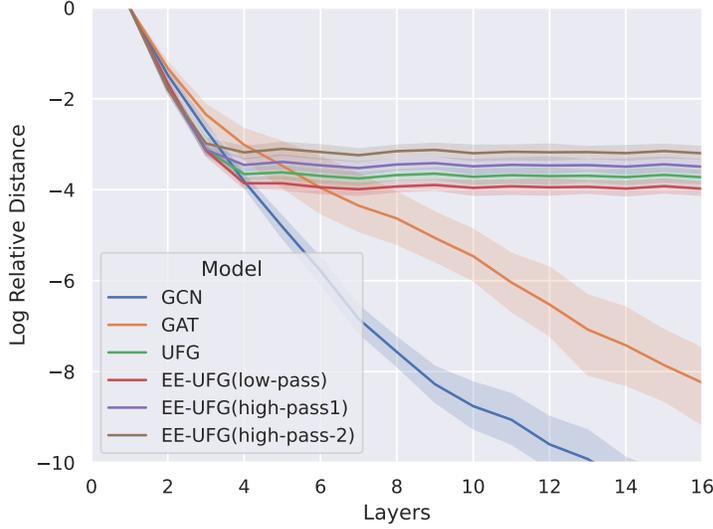


Figure 4: Layer-wise distances from the feature to the subspace \mathcal{M} . The result is an average of 100 runs. The Y-axis is the log relative distance, defined by $y^{(l)} = \log(d_{\mathcal{M}}(X^{(l)})/d_{\mathcal{M}}(X^{(0)}))$. $X^{(0)}$ is the initial feature representation, and $X^{(l)}$ is the output of the l -th layer.

Figure 4 plots the logarithm of the relative distance from l -th layer’s output to the subspace \mathcal{M} , i.e., $y^{(l)} = \log(d_{\mathcal{M}}(X^{(l)})/d_{\mathcal{M}}(X^{(0)}))$. The subspace \mathcal{M} is defined by $\mathcal{M} = U \otimes \mathbb{R}^C = \{\sum_{m=1}^M e_m \otimes w_m | w_m \in \mathbb{R}^C, e_m \in U\} \subseteq \mathbb{R}^{N \times C}$, where U is the eigenspace associated with the smallest eigenvalue (that is, zero) of a (normalized) graph Laplacian Δ . We can observe that the low-pass of EE-UFG converges to a relatively closer distance to the subspace than the high-passes. It can also be predicted by Proposition 2. The layer-wise outputs of GCN and GAT exponentially approach the subspace \mathcal{M} . This subspace is invariant under any polynomial of the Laplacian Matrix, i.e., $\forall x \in \mathcal{M}, g(\Delta)x \in \mathcal{M}$. It corresponds to the low-frequency part of graph spectra and only carries the information of the connected components of the graph [10].

A.5 Proof of Equivariance

Proposition 8 *An EEConv layer is permutation-equivariant.*

Proof Let $X \in \mathbb{R}^{n \times d}$ be the node features, the decomposition $\mathcal{W} = [\mathcal{W}_{0,2}; \mathcal{W}_{1,1}; \mathcal{W}_{1,2}]$, the augmented adjacency matrix $\mathbf{A} = [A^L; A^H; A^H]$, P is the permutation matrix that is applied to the node features, and \mathcal{W}^P the framelet transform of the permuted graph. Then, the following holds for any permutation matrix $P \in \mathbb{R}^{n \times n}$.

$$\mathbf{P} \mathbf{A} \mathbf{P}^\top \mathcal{W}^P P X = \begin{bmatrix} P A^L P^\top \mathcal{W}_{0,2}^P P X \\ P A^H P^\top \mathcal{W}_{1,1}^P P X \\ P A^H P^\top \mathcal{W}_{1,2}^P P X \end{bmatrix} = \begin{bmatrix} P A^L P^\top P U^\top \Lambda_{0,2} U P^\top P X \\ P A^H P^\top P U^\top \Lambda_{1,1} U P^\top P X \\ P A^H P^\top P U^\top \Lambda_{1,2} U P^\top P X \end{bmatrix} = \mathbf{P} \mathbf{A} \mathcal{W} X,$$

where U is the graph Fourier transform, $\Lambda_{r,j}$ is defined in Eq. 2. Similarly, framelet reconstruction is permutation equivariant. Let f be the function that represents EEConv in Eq. 8, thus, for any permutation matrix P , we have $f(PH^{(l)}) = Pf(H^{(l)})$, i.e., f is permutation equivariant.

B An Example: Enhancing Energy for Sheaf Convolution

Framelet systems can be well applied to manifold signals, $f \in L^2(\mathcal{M})$. Akin to the graph Laplacian, for a given manifold \mathcal{M} , we consider its Laplace-Beltrami operator \mathcal{L}_B which is defined as $\mathcal{L}_B f = -\text{div}(\nabla f)$. \mathcal{L}_B^L and \mathcal{L}_B^H can then be defined similarly as Eq. 7 respectively. In general, our proposed framelet augmentation method can be naturally extended to any other (symmetric) Laplacian-based propagation rules, using the framelet theory on manifolds.

The gradient operator ∇ maps $x \in L^2(\mathcal{M})$ to its associated tangent plane $T_x(X) \in L^2(T\mathcal{M})$.

To obtain the discrete version, we can sample the manifold \mathcal{M} at N points, using polynomial-exact quadrature rules, like Gauss–Legendre quadrature sampling method. With these points, we can construct a set of triangular meshes (V, E, F) . The edge connection between i and j indicates that $(i, j) \in E$ is shared by two triangular meshes, as originally proposed by [42]. Using the same formula for the graph, but replacing the graph Laplacian by the Laplace-Beltrami operator \mathcal{L}_B , we can define *Manifold framelet transforms* as

$$\begin{aligned} \mathcal{W}_{0,J} &\approx \mathcal{T}_0(2^{-K+J-2}\mathcal{L}_B) \cdots \mathcal{T}_0(2^{-K}\mathcal{L}_B), \\ \mathcal{W}_{r,1} &\approx \mathcal{T}_r(2^{-K}\mathcal{L}_B), \text{ and } \mathcal{W}_{r,j} \approx \mathcal{T}_r(2^{-K+j-1}\mathcal{L})\mathcal{T}_0(2^{-K+j-2}\mathcal{L}_B) \cdots \mathcal{T}_0(2^{-K}\mathcal{L}_B). \end{aligned} \quad (12)$$

B.1 Diffusion Problems on Manifold

The Laplace-Beltrami operator is closely related to the diffusion process over the manifold, which is governed by the PDE

$$\dot{f}(x, t) = -\mathcal{L}_B f(x, t), \quad f(x, 0) = f_0(x),$$

where $f(x, t)$ is the signal at point x at time t , $f_0(x)$ is the initial condition at point x . In the discrete setting, let $X^{(t)} \in \mathbb{R}^{N \times d}$ denote the feature representation at time point t . The following propagation rule can be used to approximate the continuous diffusion process on the manifold:

$$X^{(t+1)} = X^{(t)} - \mathcal{L}_B X^{(t)} = (I - \mathcal{L}_B)X^{(t)}.$$

Similar to (8), a framelet manifold convolution for the diffusion process can be derived as follows. For $r = 1, \dots, n$ and $j = 1, \dots, J$,

$$\begin{aligned} H_{0,J}^{(l+1)} &= \sigma((I - \mathcal{L}_B^L)\mathcal{W}_{0,J}H^{(l)}W_{0,J}^{(l)}), \quad H_{r,j}^{(l+1)} = \sigma((I - \mathcal{L}_B^H)\mathcal{W}_{r,j}H^{(l)}W_{r,j}^{(l)}); \\ H^{(l+1)} &= \mathcal{V}(H_{0,J}^{(l+1)}; H_{1,1}^{(l+1)}, \dots, H_{n,J}^{(l+1)}), \end{aligned} \quad (13)$$

B.2 Sheaf Laplacian

Here we discuss Sheaf Laplacians as an example of the generalization of our method to general manifolds. The definition of the framelet system on the sheaf, which we name as *sheaflets*, is very similar to that on the graph but with a sheaf Laplacian which contains tunable parameters. Sheaflets can then be used to define *sheaflet convolution* like (3) and then the enhanced sheaflet convolution as (8). The latter can be proved to follow the same energy enhancement as the graph framelet convolution.

A cellular sheaf defined over a graph assigns each node and each edge a vector space and introduces a linear map between the associated spaces of each node-edge pair. In the mathematical language, a cellular sheaf \mathcal{F} on an undirected graph \mathcal{G} is given by

1. a vector space $\mathcal{F}(v)$ for each vertex v of \mathcal{G} ,
2. a vector space $\mathcal{F}(e)$ for each edge e of \mathcal{G} ,
3. a linear map $\mathcal{F}_{v \triangleleft e}: \mathcal{F}(v) \rightarrow \mathcal{F}(e)$ for each incident vertex-edge pair $v \triangleleft e$ of \mathcal{G} .

Construct the *Sheaf Laplacian* $L_{\mathcal{F}}: C^0(\mathcal{G}, \mathcal{F}) \rightarrow C^0(\mathcal{G}, \mathcal{F})$, where the diagonal blocks are $L_{\mathcal{F}_{vv}} = \sum_{v \triangleleft e} \mathcal{F}_{v \triangleleft e}^\top \mathcal{F}_{v \triangleleft e}$ and the non-diagonal blocks $L_{\mathcal{F}_{vu}} = -\mathcal{F}_{v \triangleleft e}^\top \mathcal{F}_{u \triangleleft e}$. Compared with the graph Laplacian, sheaf Laplacian is a $Nd \times Nd$ matrix, consisting of a class of linear operators over the graph, thus allowing the more underlying geometric and algebraic structure of the graph. N is the number of nodes of \mathcal{G} , d is the dimension of the stalks that associated to each node.

B.3 Sheaflets

Let $\{(u_l, \lambda_l)\}_{l=1}^{Nd}$ the eigen-pair for the sheaf Laplacian $L_{\mathcal{F}}$ on $l_2(\mathcal{G})$. For $j \in \mathbb{Z}$ and $p \in V$, the *undecimated sheaflets* $\phi_{j,p}(v)$ and $\psi_{j,p}^r(v)$, $v \in V$ at scale j are *filtered Bessel kernels*

$$\begin{aligned} \phi_{j,p}(v) &:= \sum_{l=1}^{Nd} \hat{\alpha} \left(\frac{\lambda_l}{2^j} \right) \overline{u_l(p)} u_l(v), \\ \psi_{j,p}^r(v) &:= \sum_{l=1}^{Nd} \hat{\beta} \left(\frac{\lambda_l}{2^j} \right) \overline{u_l(p)} u_l(v), \quad r = 1, \dots, n. \end{aligned} \quad (14)$$

Here, j and p in $\phi_{j,p}(v)$ and $\psi_{j,p}^r(v)$ indicate the ‘‘dilation’’ at scale j and the ‘‘translation’’ at a vertex $p \in V$. $\alpha(\cdot), \beta(\cdot)$ are the scaling functions as defined in Section 2. Let $J, J_1, J > J_1$ be two integers. An *undecimated sheaflet system* $\text{UFS}(\Psi, \eta; \mathcal{G})$ (starting from a scale J_1) as a non-homogeneous, stationary affine system:

$$\begin{aligned} \text{UFS}_{J_1}^J(\Psi, \eta) &= \text{UFS}_{J_1}^J(\Psi, \eta; \mathcal{G}) \\ &:= \{\phi_{J_1,p} : p \in V\} \cup \{\psi_{j,p}^r : p \in V, j = J_1, \dots, J\}_{r=1}^n. \end{aligned} \quad (15)$$

The system $\text{UFS}_{J_1}^J(\Psi, \eta)$ is then called an *undecimated tight frame* for $l_2(\mathcal{G})$ and the elements in $\text{UFS}_{J_1}^J(\Psi, \eta)$ are called *undecimated tight sheaflets* on \mathcal{G} .

The *sheaflet coefficients* $V_0, W_j^r \in \mathbb{R}^{Nd \times f}$ are defined as the inner-product of the sheaflet and the sheaf signal $X \in \mathbb{R}^{Nd \times f}$, where f denotes the feature dimension. The size of V_0 and W_j^r is the same as the sheaf signal X . Then,

$$V_0 = \langle \phi_{0,\cdot}, X \rangle = U^\top \widehat{\alpha}\left(\frac{\Lambda}{2}\right)UX \quad \text{and} \quad W_j^r = \langle \psi_{j,\cdot}^r, X \rangle = U^\top \widehat{\beta}^{(r)}\left(\frac{\Lambda}{2^{j+1}}\right)UX, \quad (16)$$

where the scaling functions on \mathcal{G} are as follows,

$$\widehat{\alpha}\left(\frac{\Lambda}{2^{j+1}}\right) = \text{diag}\left(\widehat{\alpha}\left(\frac{\lambda_1}{2^{j+1}}\right), \dots, \widehat{\alpha}\left(\frac{\lambda_{Nd}}{2^{j+1}}\right)\right), \quad \widehat{\beta}^{(r)}\left(\frac{\Lambda}{2^{j+1}}\right) = \text{diag}\left(\widehat{\beta}^{(r)}\left(\frac{\lambda_1}{2^{j+1}}\right), \dots, \widehat{\beta}^{(r)}\left(\frac{\lambda_{Nd}}{2^{j+1}}\right)\right).$$

B.4 Implementation Format

To reduce the computational complexity caused by eigendecomposition for Sheaf Laplacians, we use Chebyshev polynomials to approximate. Consider Chebyshev polynomials $\mathcal{T}_0, \dots, \mathcal{T}_n$ of fixed degree t , and filter $a \approx \mathcal{T}_0$ and $b^{(r)} \approx \mathcal{T}_r$, then the above 2 can be approximated

$$\begin{aligned} \mathcal{W}_{0,J} &\approx U^\top \mathcal{T}_0(2^{-K+J-1}\Lambda) \dots \mathcal{T}_0(2^{-K}\Lambda)U = \mathcal{T}_0(2^{K+J-2}L_{\mathcal{F}}) \dots \mathcal{T}_0(2^{-K}L_{\mathcal{F}}), \\ \mathcal{W}_{r,1} &\approx U^\top \mathcal{T}_r(2^{-K}\Lambda)U = \mathcal{T}_r(2^{-K}L_{\mathcal{F}}), \\ \mathcal{W}_{r,j} &\approx U^\top \mathcal{T}_r(2^{-K+j-1}\Lambda)\mathcal{T}_0(2^{-K+j-2}\Lambda) \dots \mathcal{T}_0(2^{-K}\Lambda)U \\ &= \mathcal{T}_r(2^{K+j-1}L_{\mathcal{F}})\mathcal{T}_0(2^{K+j-2}L_{\mathcal{F}}) \dots \mathcal{T}_0(2^{-K}L_{\mathcal{F}}). \end{aligned}$$

$\mathcal{L}_{\mathcal{F}}$ is the sheaf Laplacian.

B.5 From Sheaf Convolution to Sheaflet Convolution

Sheaf convolution [43, 44] is defined as follows,

$$Y = \sigma((I_{Nd} - L_{\mathcal{F}})(I_N \otimes W_1)XW_2) \in \mathbb{R}^{Nd \times f_2}, \quad (17)$$

where $(I_N \otimes W_1)XW_2 = \tilde{X} \in \mathbb{R}^{Nd \times f_2}$. Inheriting the characteristics of sheaf convolution in Eq. (17), we define *Sheaflet Convolution* based on the sheaf framelet system as

$$\begin{aligned} Y_{0,J} &= \sigma((I_{Nd} - \mathcal{L}_{\mathcal{F}})(I_N \otimes W_1)\mathcal{W}_{0,J}XW_{0,J}), \\ Y_{r,j} &= \sigma((I_{Nd} - \mathcal{L}_{\mathcal{F}})(I_N \otimes W_1)\mathcal{W}_{r,j}XW_{r,j}), \\ Y &= \mathcal{V}(Y_{0,J}; Y_{1,1}, \dots, Y_{n,J}). \end{aligned} \quad (18)$$

B.6 Dirichlet Energy for Sheaflets

Applying our Dirichlet energy enhancement strategy to Sheaflet Convolution (Eq. 18), we obtain the following *Energy Enhanced Sheaflet Convolution*,

$$\begin{aligned} Y_{0,J} &= \sigma((I_{Nd} - \mathcal{L}_{\mathcal{F}}^L)(I_N \otimes W_1)\mathcal{W}_{0,J}XW_{0,J}), \\ Y_{r,j} &= \sigma((I_{Nd} - \mathcal{L}_{\mathcal{F}}^H)(I_N \otimes W_1)\mathcal{W}_{r,j}XW_{r,j}), \\ Y &= \mathcal{V}(Y_{0,J}; Y_{1,1}, \dots, Y_{n,J}), \end{aligned} \quad (19)$$

where $\mathcal{L}_{\mathcal{F}}^L$ and $\mathcal{L}_{\mathcal{F}}^H$ are defined similarly as Eq. 7.

The sheaflet Dirichlet energy with modified sheaf Laplacian ($\epsilon > 0$) is defined as

$$\begin{aligned} E_{0,J}^\epsilon(X) &= ((I_N \otimes W_1)W_{0,J}X)^\top (\mathcal{L}_F + \epsilon D^{-1})((I_N \otimes W_1)W_{0,J}X) \\ E_{r,j}^\epsilon(X) &= ((I_N \otimes W_1)W_{r,j}X)^\top (\mathcal{L}_F - \epsilon D^{-1})((I_N \otimes W_1)W_{r,j}X), \end{aligned} \quad (20)$$

where D is the degree matrix of the sheaf Laplacian. The total sheaflet Dirichlet energy $E_{\mathcal{F}}^\epsilon(X) = E_{0,J}^\epsilon(X) + \sum_{r,j} E_{r,j}^\epsilon(X)$. The original sheaf Dirichlet energy is defined as

$$E_{\mathcal{F}}(X) = ((I_N \otimes W_1)X)^\top \mathcal{L}_F((I_N \otimes W_1)X).$$

It can be similarly proved that $E_{\mathcal{F}}^\epsilon(X) > E_{\mathcal{F}}(X)$ when $\epsilon > 0$.

C Experimental Details

C.1 Experimental Setting

The implementation of our model and training is based on PyTorch [45] on NVIDIA Tesla A100 GPU with 6,912 CUDA cores and 80GB HBM2 mounted on an HPC cluster. PyTorch Geometric Library [46] is employed for all the benchmark datasets and baseline models. For each model, we run 2000 epochs for ogb-arxiv and 300 epochs for other datasets and select the configuration with the highest validation accuracy. The results in Table 1 and Table 2 are the average performance of each model over 10 fixed public splits.

C.2 Datasets

We conduct experiments over 8 node classification datasets in 3 types:

1. **Citation Network:** The Cora, CiteSeer, PubMed are citation network datasets introduced by [24], where nodes represent documents in the computer science fields and edges represent citation links.
2. **Webpage Network:** The Texas, Wisconsin, and Cornell are webpage network datasets introduced by [21]. Nodes are the web pages and edges are the hyperlinks between them. Node features are bag-of-words representations of web pages. Nodes are classified into one of five categories: Students, Projects, Courses, Faculty and Staff.
3. **Wikipedia Network:** The Chameleon and Squirrel are Wikipedia network datasets, introduced by [23]. Nodes are the web pages and edges are the hyperlinks between them. Node features represent several informative nouns on Wikipedia pages.
4. **Ogb-arxiv:** The ogb-arxiv dataset is a citation network of Computer Science arxiv papers introduced by [25]. Each node represents a paper and each directed edge indicates that one paper cites another one. Each node has a 128-dimensional feature that is averaged from the embeddings of words in its title and abstract. The task is to predict the 40 categories of the arXiv CS papers, which used to be labeled manually. However, with the increasing volume of CS papers, it is necessary to develop an automatic classification model.

Citation Network, Webpage Network and Wikipedia Network datasets are available at <https://pytorch-geometric.readthedocs.io/en/latest/modules/datasets.html>. Ogb-arxiv dataset is available at <https://ogb.stanford.edu/docs/nodeprop/#ogbn-arxiv>.

C.3 Baselines Selection

We select classic GNNs and state-of-the-art methods for heterophilous graphs and over-smoothing issues as our baselines: (1) classic GNN models: GCN [1], GAT [3], GraphSAGE [2], UFG [15]; (2) GNNs that can circumvent over-smoothing: GRAND [26], PairNorm [6], GCNII [27], EGNN [8]; (3) models for heterophilous graphs: FAGCN [11], MixHop [28]. For each model, we use the official codes provided by the authors. All models use the same train/validate/test split for a fair comparison. We notice that there were some impressive performance over Cora/CiteSeer/PubMed reported in the previous literature, like H2GCN [47], Geom-GCN [21], GGCN [48]. We do not adopt them as baseline models here because they randomly generate the train/validate/test split, which is different from our experimental setting.

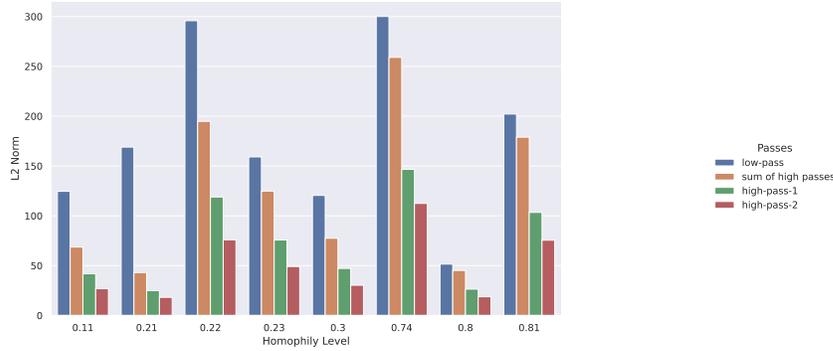


Figure 5: Energy (L_2 norm) of framelet coefficients for the 8 datasets.

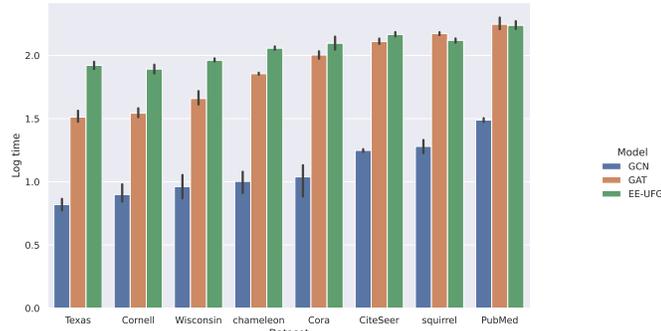


Figure 6: Running time comparison

C.4 Computational Complexity

The time complexity of the algorithm is important for real-world deployment, especially for extremely large graph data. The framelet transform is equivalent to left-multiplying a specific transformation matrix. We stack the transformation matrices to obtain a tensor-based framelet transform with the computational complexity of $\mathcal{O}(N^2(nJ+1)d)$. N is the number of nodes, d is the feature dimension, n is the number of high-pass filters and J is the scale level of the low-pass. In our implementation, we fix $n = 1$, $J = 2$. Benefiting from an efficient message passing operator in PyTorch Geometry, we construct a large sparse adjacency matrix and stack all the passes, thus, the message passing in all passes can be executed in parallel.

Figure 6 plots the running time on eight datasets we used. The number of nodes increases sequentially from left to right on the X-axis. The Y-axis is the logarithm of running time (in seconds). Each model has the same configuration, including hidden units, number of layers, etc., and run 300 epochs. We can observe from the figure summary that EE-UFG has a computational complexity close to GAT, especially when the number of nodes is large.

Parameter	Search Space
Learning rate	$[1 \times 10^{-5}, 1 \times 10^{-1}]$
Hidden units	$\{16, 32, 64, 128\}$
Number of layers	$[1, 10]$
Epsilon	$[1 \times 10^{-5}, 1 \times 10^{-1}]$
Dropout rate	$\{0.2, 0.4, 0.6, 0.8\}$
Weight decay	$[5 \times 10^{-3}, 1 \times 10^{-2}]$

Table 3: Hyper-parameter Search Spaces of EE-UFG

C.5 Hyper-parameter and Model Implementation

We employ Adam [49] as our optimizer, and ASHAScheduler [50] as our scheduler. Each model is fine-tuned with Ray [51]. Table 3 provides the hyper-parameter search space for reproduction. All baseline models are implemented with the official codes released by the authors.