Dirichlet Energy Enhancement of Graph Neural Networks by Framelet Augmentation

Jialin Chen
Shanghai Jiao Tong University
sjtuchenjl@sjtu.edu.cn

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

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

Pietro Liò
University of Cambridge
pl219@cam.ac.uk

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

Abstract

Graph convolutions have been a key element in learning graph representations. However, many existing graph convolutions suffer from oversmoothing – as the network gets deeper, the feature becomes increasingly indistinguishable and the Dirichlet energy decays to zero. In this work, we consider framelet convolutions, which are spectral convolutions on graphs based on framelet transforms that can represent multi-scale graph features through low and high-frequency passes. We develop a framelet augmentation strategy by adjusting the low and high pass filters with positive and negative increments, respectively. We prove that the resulting approach can increase the Dirichlet energy and thus leads to an Energy Enhanced Convolution (EEConv) for GNNs. From a message-passing perspective, EEConv inherits multi-hop aggregation properties from the framelet transform, which takes into account all hops in the multiscale framelet representation. Experiments show that deep GNNs with EEConv can achieve state of the art performance on various node classification datasets with different homophily levels while also managing to lift 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, which can naturally be represented as a graph. Recently, Graph Neural Networks (GNNs) [1, 2, 3] have achieved a myriad of eye-catching performances in multiple applications on graph-structured data. However, for traditional GCN or other extensions of GNNs, there is a key limitation: the oversmoothing phenomenon, which means that the increase of the model's depth gives rise to the decay of predictive performance. One metric to quantify the oversmoothing phenomenon is the Dirichlet energy. As the number of stacked layers increases, the Dirichlet energy rapidly converges to zero [4, 5], leading to identical node features. Some algorithms have been proposed to leverage the Dirichlet energy to control the average node pair distances, thus alleviating the oversmoothing issue [6, 7].

Wavelet analysis on graphs provides a multi-scale representation for graph-structured data, which provides an effective tool for complex graph signal processing [8, 9, 10]. Framelets are a kind of tight wavelet frame on compact Riemannian manifolds that can be naturally applied to graphs. As shown by [9], graph framelet convolution forms an efficient framework for learning effective graph representations. The low-pass filter distills the approximation component of the input graph and the high-pass filters extract more detailed information. The energy difference between the low and high passes inspires a potential strategy to improve the Dirichlet energy.

Preprint. Under review.

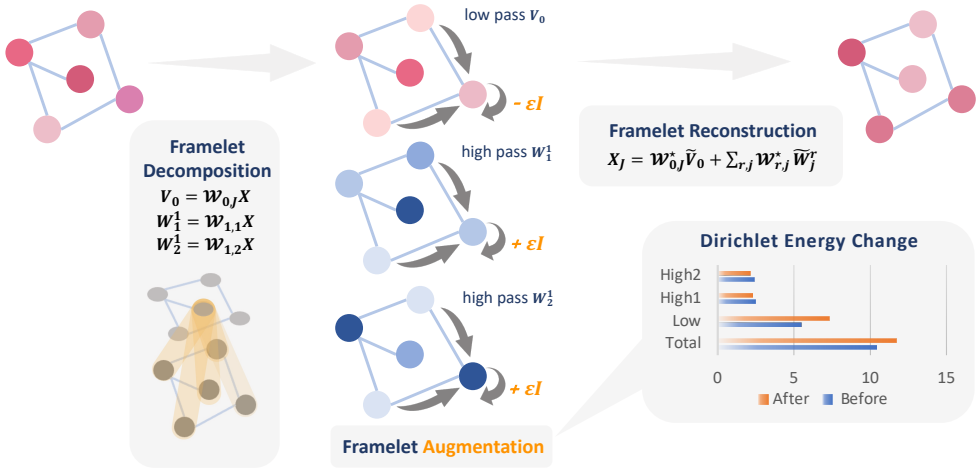


Figure 1: An illustration of the proposed Energy Enhanced Convolution. We first conduct framelet decomposition on the original graph signal (Eq. 18) and obtain one low pass and two high passes. The Framelet Augmentation is applied by adding or subtracting an increment for low and high passes in the original framelet convolution (Eq. 10). The total Dirichlet energy will be lifted in this process. A framelet reconstruction operator follows to resize the framelet coefficients to the original size.

In this paper, we materialise this idea in a novel **E**nergy **E**nhanced **Conv**olution (**EEConv**) that can be repeatedly stacked to construct a more robust and deeper GNN architecture without sacrificing predictive performance 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-hops information are considered. Then framelet augmentation is applied to different passes, by modifying the corresponding diagonals of the adjacency matrices (Section 3). Meanwhile, the Dirichlet energy associated with the graph will be enhanced. 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 convolutional models with Laplacian-based propagation rules, such as heat diffusion on manifolds. We discuss possible extension of the framelets and framelet convolution in Section 5.

There are two motivations for **EEconv** to ease oversmoothing. **(1)** One can intuitively understand the oversmoothing phenomenon by being aware of the existence of the receptive filed. In GNNs, the node embeddings are determined by the receptive field which typically includes the node itself and its neighbors. As the node features evolve as the number of layers increases, the receptive field of the node dilates exponentially since all its neighbors' domains are merged in each layer. Hence, intuitively we want to retard the expansion of the receptive field. Spectral graph theory provides inspiration for how this can be achieved. Some studies show that information from neighbors can be believed strongly associated with low-frequency components [11, 12]. This encourages us to enhance the Dirichlet energy of the low-pass in the propagation of the network and for balancing weaken high-passes. **(2)** In the analysis of Dirichlet energy, the whole Dirichlet energy can be decomposed into the energy of information in all passes. In the decomposition of framelet Dirichlet energy, one important observation is that the diagonal matrix is translatable, which implies that the self-enhancement/impairment are allowed to some degree. For a simple trick of posing a small perturbation, the energy can be separated into two parts: one remains the original part and another depends on the perturbation. For a remedy, we carry out an augmentation for the framelet convolution where the low pass and high passes in the framelet transforms are added/subtracted by a small perturbation. As a result, the Dirichlet energy does not decay.

To this end, the contributions of this work are threefolds: **(1)** We conduct a systematic study of the Dirichlet energy based on the framelet system and propose a novel framelet augmentation strategy to upgrade the Dirichlet energy. **(2)** We theoretically prove some important observations about the effect of framelet augmentation and study the different behaviors of low pass and high passes during the feature propagation. **(3)** We use the proposed Energy Enhanced Convolution to tackle the

oversmoothing problem and conduct experimental investigation to demonstrate the effectiveness of framelet augmentation for real-world node classification tasks.

2 Graph Framelet Convolution

Framelet and wavelet analysis on the manifold provides a powerful tool for defining neural network operation modules of geometric deep learning. In this paper, we mainly focus on tight framelets on a graph [9, 13, 14], which is a multiscale affine system.

Framelets on Graph For a graph $\mathcal{G}$ with N nodes and graph Laplacian Δ, let $U = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_N]$ be the matrix of eigenvectors of Δ, which is also the graph Fourier transform, and $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_N)$ be the diagonal matrix of the eigenvalues. Framelets over the graph is defined by a set of *scaling functions* $\Phi = \{\alpha; \beta^{(1)}, \cdots, \beta^{(n)}\} \subset L_1(\mathbb{R})$ and an associated *filter bank* $\eta = \{a; b^{(1)}, \cdots, b^{(n)}\} \subset \ell_1(\mathbb{Z})$, whose elements are compactly supported, ℓ_1 denotes ℓ_1 norm (i.e., $\|x\|_1 = (\sum_{i=1}^{\infty} |x_i|)$), n denotes the number of high-pass filters. For $\forall \xi \in \mathbb{R}$, the Fourier transform of a, b, α, β satisfy that $\widehat{\alpha}(2\xi) = \widehat{a}(\xi)\widehat{\alpha}(\xi)$ and $\widehat{\beta^{(r)}}(2\xi) = \widehat{b^{(r)}}(\xi)\widehat{\alpha}(\xi)$. The *framelets* for graph $\mathcal{G}$ are defined by

$$\varphi_{j,p}(v) = \sum_{l=0}^{N} \widehat{\alpha}\left(\frac{\lambda_l}{2^j}\right) u_l(p)u_l(v); \quad \psi^r_{j,p}(v) = \sum_{l=0}^{N} \widehat{\beta}^{(n)}\left(\frac{\lambda_l}{2^j}\right) u_l(p)u_l(v),\ r = 1, \ldots, n, \quad (1)$$

for *scale level* $j = 1, \cdots, J$. $\varphi_{j,p}(v)$ and $\psi^r_{j,p}(v)$ are the *low-pass* and *high-pass* framelets at node v associated to node p respectively. Therefore, the framelet transforms actually take into account the global information and all hops of the graph into multi-scale graph representations. The low-pass and high-pass framlets can distill approximation and detail information of graph signals.

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

$$V_0 = \langle \varphi_{0,\cdot}, X\rangle = U^\top \widehat{\alpha}\big(\frac{\Lambda}{2}\big)UX \ \text{ and } \ W^r_j = \big\langle \psi^r_{j,\cdot}, X\big\rangle = U^\top \widehat{\beta^{(r)}}\big(\frac{\Lambda}{2^{j+1}}\big)UX, \quad (2)$$

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

$$\widehat{\alpha}\big(\frac{\Lambda}{2^{j+1}}\big) = \mathrm{diag}\left(\widehat{\alpha}\big(\frac{\lambda_1}{2^{j+1}}\big), \cdots, \widehat{\alpha}\big(\frac{\lambda_N}{2^{j+1}}\big)\right)$$
$$\widehat{\beta^{(r)}}\big(\frac{\Lambda}{2^{j+1}}\big) = \mathrm{diag}\left(\widehat{\beta^{(r)}}\big(\frac{\lambda_1}{2^{j+1}}\big), \cdots, \widehat{\beta^{(r)}}\big(\frac{\lambda_N}{2^{j+1}}\big)\right).$$

Let $\mathcal{W}_{r,j}$ denote the decomposition operators given by $V_0 = \mathcal{W}_{0,J}X$ and $W^r_j = \mathcal{W}_{r,j}X$. Then according to Eq. 18, we obtain the framelet transform matrices for decomposition:

$$\begin{aligned}
\mathcal{W}_{0,J} &= U^\top \widehat{a}(2^{-K+J-1}\Lambda)\cdots\widehat{a}(2^{-K}\Lambda)U := U^\top \Lambda_{0,J}U,\\
\mathcal{W}_{r,1} &= U^\top \widehat{b^{(r)}}(2^{-K}\Lambda)U := U^\top \Lambda_{r,1}U,\\
\mathcal{W}_{r,j} &= U^\top \widehat{b^{(r)}}(2^{-K+j-1}\Lambda)\widehat{a}(2^{-K+j-2}\Lambda)\cdots\widehat{a}(2^{-K}\Lambda)U := U^\top \Lambda_{r,j}U.
\end{aligned} \quad (3)$$

K is a sufficiently large value such that the Laplacian's biggest eigenvalue $\lambda_{max} \le 2^K\pi$. For graph signals, we use Haar-type filters to construct a framelet system of 2 scale level ($j = 1, 2$) and 1 high-pass filter ($r = 1$), where the low and high pass filters are $\widehat{a}(x) = \cos(x/2)$ and $\widehat{b^{(1)}}(x) = \sin(x/2)$ respectively. Intuitively, the magnitude of the high passes coefficients is usually much smaller than the low pass. This motivates us to consider the energy imbalance between low and high passes. With L_2 norm as the energy of signals, we can prove that the sum of high-passes energy is less than that of the low-pass, or precisely, $\|W^1_1\|^2 + \|W^1_2\|^2 \le \|V_0\|^2$. See the proof in Appendix A.2. 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 the approximation information of the original graph signal, and the high passes distill the detail part in the signal.

To reduce the computational complexity caused by eigendecomposition for graph Laplacians, we use Chebyshev polynomials to approximate the framelet decomposition in our implementation. Let

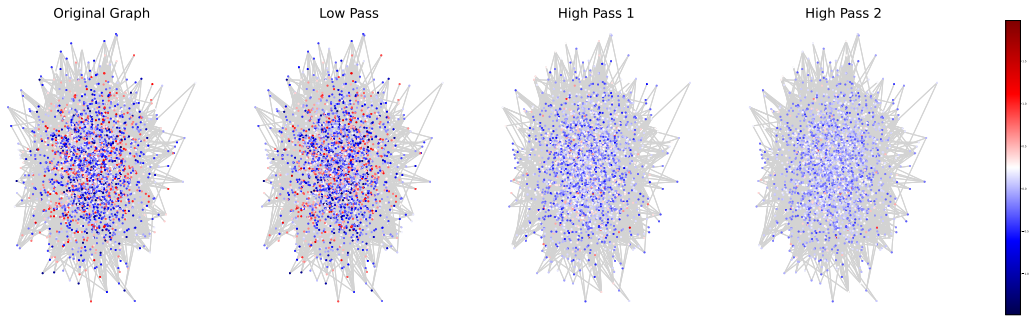


Figure 2: Visualization of framelet coefficients on the Cora dataset. From left to right we show the original graph, low-pass, high-pass 1 and high-pass 2 respectively. The projected value is the first principal component of the high-dimensional features.

$\mathcal{T}_0, \cdots, \mathcal{T}_n$ be Chebyshev polynomials of fixed degree t, and filter $a \approx \mathcal{T}_0$ and $b^{(r)} \approx \mathcal{T}_r$, then the above Eq. 3 can be approximated by $\mathcal{W}_{0,J} \approx \mathcal{T}_0(2^{-K+J-2}\Delta)\cdots\mathcal{T}_0(2^{-K}\Delta); \mathcal{W}_{r,1} \approx \mathcal{T}_r(2^{-K}\Delta)$ and $\mathcal{W}_{r,j} \approx \mathcal{T}_r(2^{-K+j-1}\Delta)\mathcal{T}_0(2^{-K+j-2}\Delta)\cdots\mathcal{T}_0(2^{-K}\Delta)$. Here, the Δ can be taken as the well-known normalized graph Laplacian matrix Δ or other Laplacian matrix over manifolds, such as Laplacian-Beltrami matrix, sheaf Laplacian [7]. See Section 5.

Framelet Convolution The layer-wise propagation rule of generic GCN [1] is

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \tag{4}$$

where $\tilde{A} = A + I_N$ is the adjacency matrix of the original graph with self-connections, $\tilde{D}$ is the diagonal degree matrix associated with $\tilde{A}$, $H^{(l)}$ is the feature representations of the l-th layer, $W^{(l)}$ is the l-th layer weight matrix. Let the augmented adjacency matrix $\widehat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$, then augmented normalized Laplacian [11] of the input graph is defined as $\tilde{\Delta} = I_N - \widehat{A} = I_N - \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$.

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

$$\begin{aligned} H_{r,j}^{(l+1)} &= \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}\mathcal{W}_{r,j}H^{(l)}W_{r,j}^{(l)}) \\ H^{(l+1)} &= \mathcal{V}(H_{0,J}^{(l+1)}; H_{1,1}^{(l+1)}, \cdots, H_{n,J}^{(l+1)}) \end{aligned} \tag{5}$$

where $(r,j) \in \{(r,j)|r = 1,\cdots,n; j = 1,\cdots,J\} \cup \{(0,J)\}$, $W_{r,j}^{(l)}$ 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 $X_J = \mathcal{V}(V_0, W_{1,1}, \cdots, W_{n,J}) = \mathcal{W}_{0,J}^{\star}V_0 + \sum_{r=1}^{n}\sum_{j=1}^{J}\mathcal{W}_{r,j}^{\star}W_{r,j}$, 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 [9], which is more like the filter learning in the frequency domain and its feature propagation rule is $y = \mathrm{ReLU}(\mathcal{V}(\theta'(\mathcal{W}(X))))$, our framelet graph convolution is defined in the spatial domain and generalize the message passing mechanism to multi-scales representation.

3 Dirichlet Energy Enhancement

3.1 Dirichlet Energy

One bottleneck of the graph-based models like GCN [1], GAT [3] is the oversmoothing issue, which means that the node representations tend to be similar and indistinguishable as the graph neural network deepens. Consequently, it destroys the performance of GNNs when the network goes deeper. Dirichlet Energy is used to analyse this phenomenon. It is proved that the Dirichlet energy exponentially converges to zero with respect to the number of the stacked layers. The representations then converge to a subspace formulated with the bases of eigenspace of graph Laplacian Δ, see [4, 5].

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 $\mathrm{trace}(X^\top \tilde{\Delta} X)$.

With the framelets and their transforms defined in Section 2, we define framelet Dirichlet energy for low-pass and high-passes signals respectively.

$$E_{0,J}(X) = (\mathcal{W}_{0,J}X)^\top \tilde{\Delta}(\mathcal{W}_{0,J}X); \qquad E_{r,j}(X) = (\mathcal{W}_{r,j}X)^\top \tilde{\Delta}(\mathcal{W}_{r,j}X). \tag{6}$$

The total framelet Dirichlet energy is then defined as $E_{total} = \sum_{r,j} E_{r,j} + E_{0,J}$.

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

$$E_{total}(X) = E(X). \tag{7}$$

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. 3, where $(r,j) \in \{(r,j)|r = 1,\cdots,n; j = 1,\cdots,J\} \cup \{(0,J)\}$.*

The proofs of Proposition 1 and Remark 1 are given in Appendix A.1.

3.2 Enhanced Dirichlet Energy by Framelet Augmentation

Energy Enhanced Convolution The key idea to tackle the issue of oversmoothing is to preserve the Dirichlet Energy and avoid its exponential decay with respect to the number of layers. To take the advantage of the energy gap between low-pass and high-passes, we decouple the low-pass and high-passes propagation and define the low-pass adjacency matrix $\widehat{A}^L$ and high-pass adjacency matrix $\widehat{A}^H$ respectively. The augmented normalized Laplacian $\tilde{\Delta}$ is also changed correspondingly, since $\tilde{\Delta} = I_N - \widehat{A}$.

$$\widehat{A}^L = \widehat{D}^{-\frac{1}{2}}(\tilde{A} - \epsilon I)\widehat{D}^{-\frac{1}{2}} = \tilde{A} - \epsilon\widehat{D}^{-1}, \quad \widehat{A}^H = \widehat{D}^{-\frac{1}{2}}(\tilde{A} + \epsilon I)\widehat{D}^{-\frac{1}{2}} = \tilde{A} + \epsilon\widehat{D}^{-1}. \tag{8}$$

$$\tilde{\Delta}^L = I_N - \widehat{A}^L = \tilde{\Delta} + \epsilon\widehat{D}^{-1}, \qquad \tilde{\Delta}^H = I_N - \widehat{A}^H = \tilde{\Delta} - \epsilon\widehat{D}^{-1}. \tag{9}$$

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.

$$\begin{aligned} H_{0,J}^{(l+1)} &= \sigma(\widehat{A}^L\mathcal{W}_{0,J}H^{(l)}W_{0,J}^{(l)}) \\ H_{r,j}^{(l+1)} &= \sigma(\widehat{A}^H\mathcal{W}_{r,j}H^{(l)}W_{r,j}^{(l)}), \quad \text{for}(r,j) \in \{(r,j)|r = 1,\cdots,n; j = 1,\cdots,J\} \\ H^{(l+1)} &= \mathcal{V}(H_{0,J}^{(l+1)}; H_{1,1}^{(l+1)},\cdots,H_{n,J}^{(l+1)}) \end{aligned} \tag{10}$$

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 also changed as Eq. 11. The total framelet Dirichlet energy $E_{total}^\epsilon = \sum_{r,j} E_{r,j}^\epsilon + E_{0,J}^\epsilon$.

$$\begin{aligned} E_{0,J}^\epsilon(X,) &= (\mathcal{W}_{0,J}X)^\top \tilde{\Delta}^L(\mathcal{W}_{0,J}X) = (\mathcal{W}_{0,J}X)^\top(\tilde{\Delta} + \epsilon\widehat{D}^{-1})(\mathcal{W}_{0,J}X) \\ E_{r,j}^\epsilon(X) &= (\mathcal{W}_{r,j}X)^\top \tilde{\Delta}^H(\mathcal{W}_{r,j}X) = (\mathcal{W}_{r,j}X)^\top(\tilde{\Delta} - \epsilon\widehat{D}^{-1})(\mathcal{W}_{r,j}X) \end{aligned} \tag{11}$$

Proposition 2 *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., $E_{total}^\epsilon(X) > E_{total}(X) = E(X)$ ($\epsilon > 0$).*

The proof of Proposition 2 is given in Appendix A.3. Therefore, using ϵI to weaken or strengthen the self-connectivity of low-pass and high-passes, we can obtain the Dirichlet energy enhancement during the feature propagation.

3.3 Topological Understanding of Framelet Augmentation

We can understand the role of Framelet Augmentation from a topological perspective. Let U be the eigenspace that is associated with the smallest eigenvalue of the graph laplacian $\tilde{\Delta}$. We define the subspace $\mathcal{M}$ by $\mathcal{M} = U \otimes \mathbb{R}^C = \{\sum_{m=1}^{M} e_m \otimes w_m | w_m \in \mathbb{R}^C; e_m \in U\} \subseteq \mathbb{R}^{N \times C}$. The distance between node features x 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. We exploit the theorem in [5] to explain the topological role of adding self-connectivity.

Theorem 1 *For GCN models defined in Eq. 4, 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 $\widehat{A}$ and s_l is the supremum of all singular values of the l-th layer weight matrix $W^{(l)}$.*

For other GCN-like models, we have similar results that the convergence rate of the distance between features and the subspace is also positively related to the eigenvalues of the $\widehat{A}$ [15], generating the consistent feature representations of nodes, which is beneficial for downstream node classification tasks for shallow GNNs, especially for homogeneous graphs. However, when the layers are stacked, the oversmoothing phenomenon will destroy the model's performance. It is much more severe for heterogeneous graph tasks, where adjacent nodes are more likely to have different labels. Therefore, similar feature representations of neighboring nodes from different classes make shallow GNNs to fail in such kind of tasks. Recent works [16, 17, 18] discover that graph convolution works for the case where only the low-frequency components are important and fails when the detail information is necessary due to its automatically denoising mechanism. The authors in [16] explain that the unsatisfying performance of GNNs usually stems from insufficient attention to high-frequency components. Therefore, adding self-attention and self-connection to high-pass signals helps to increase the weight of high-frequency components in the overall feature learning. It also reduces the proportion of high-pass component $E^{\epsilon}_{r,j}$ in the total Dirichlet energy, giving rise to the closer distances between the high-frequency components of node representations. Besides, the following proposition implies the topological behavior of high-pass signals during the learning.

Proposition 3 *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 ($k = 1, 2, \cdots, n$). $A(\epsilon)$ denotes $A + \epsilon D$, where D is a positive diagonal matrix. Then $\lambda_k(A(\epsilon))$ increases monotonically with ϵ and the following relation holds:*

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

where A^L and A^H are low-pass and high-passes adjacency matrices as defined in Eq. 8.

See proof of Proposition 3 in Appendix A.4. According to Theorem 1, adding self-connectivity in high-pass signals increases the second largest eigenvalue of adjacency matrix, leading to the slower convergence to the subspace and impeding the oversmoothing of the whole graph, with an enhanced Dirichlet energy.

3.4 Equivariance of Framelet Convolution

Equivariance and Invariance are important properties for graph neural networks. Here we discuss the permutation equivariance of EEConv with framelet augmentation.

Proposition 4 *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 $\mathbf{A} = [A^L; A^H; A^H]$, and P is the permutation matrix that is applied to the node features. Then, the following holds for any permutation matrix $P \in \mathbb{R}^{n \times n}$.*

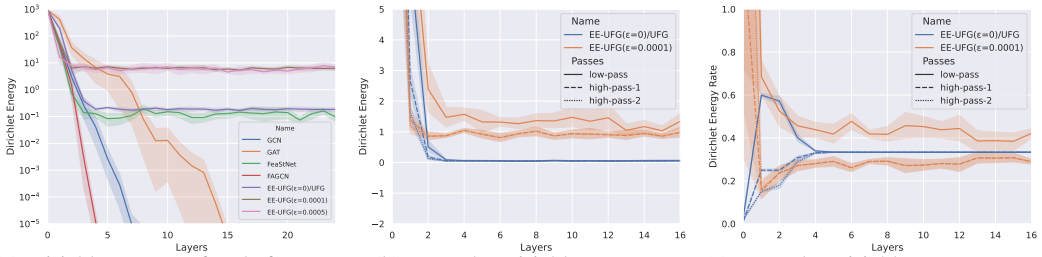
$$\mathbf{PAP}^\top \mathcal{W} PX = \begin{bmatrix} PA^L P^\top \mathcal{W}_{0,2} PX \\ PA^H P^\top \mathcal{W}_{1,1} PX \\ PA^H P^\top \mathcal{W}_{1,2} PX \end{bmatrix} = \begin{bmatrix} PA^L P^\top U^\top \Lambda_{0,2} UPX \\ PA^H P^\top U^\top \Lambda_{1,1} UPX \\ PA^H P^\top U^\top \Lambda_{1,2}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

where U is the graph Fourier transform, $\Lambda_{r,j}$ is defined in Eq. 3. The third equality uses $P^\top U^\top \Lambda_{r,j} UP = U^\top \Lambda_{r,j} U = \mathcal{W}_{r,j}$. Let f be the function that represents EEConv in Eq. 10, thus, for any permutation matrix P, we have $f(PH^{(l)}) = Pf(H^{(l)})$, i.e., f is permutation equivariant.

Since the framelet transforms are naturally generalized from the graph Fourier transform, 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 get an equivariant deep graph neural network.

4 Experiments

Experimental Investigation We use random graphs to verify the effect of framelet augmentation for alleviating the exponentially decay of Dirichlet energy. We randomly generate an undirected graph with 100 nodes that are divided into 2 classes and with connectivity probability of $p = 0.1$ for two nodes. The node features are sampled from Gaussian distribution $\mathcal{N}(0.5, 1)$ and $\mathcal{N}(-0.5, 1)$ for two classes. We show the layer-wise (logarithm of) Dirichlet energy during the feature propagation through GCN [1], GAT [3], FeaStNet [19], FAGCN [16] and our Energy-enhanced UFG (EE-UFG) in Figure 3(a). When ϵ is selected as 0, our EE-UFG is equivalent to the spatial version of UFG [9]. We can observe that the Dirichlet energy usually decays fast to zero with respect to the number of layers in the normal convolutional models. With framelet augmentation, EE-UFG lifts the Dirichlet energy to a steady state and makes the output features to be devoid of being indistinguishable. In Figure 3(b), we plot the change of framelet Dirichlet energy components to the number of layers. For the case where $\epsilon = 0$, the low-pass component and high-passes components quickly decay to zero. However, the EE-UFG with framelet augmentation not only enhances the Dirichlet energy, but also decouples the low-pass and high-passes signals and preserves the energy gap during the feature propagation. It is also verified by Figure 3(c), which shows how Dirichlet energy proportions of low-pass and high-passes are adjusted. One behavior of the output 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. This is due to that EE-UFG increases the proportion of low pass in the total Dirichlet energy and reduces that of high passes with framelet augmentation, and thus achieves an overall energy enhancement. Finally, we can observe that the framelet augmentation helps to preserve the energy gap between high-frequency and low-frequency components.



(a) Dirichlet energy of node features (b) Framelet Dirichlet energy (c) Framelet Dirichlet energy rate

Figure 3: (a) Layer-wise Dirichlet energy with different models. EE-UFG with framelet augmentation lifts the Dirichlet energy to a higher steady state compared with other graph convolutional models. (b) Layer-wise framelet Dirichlet energy components. (c) Changes of framelet Dirichlet energy rate.

Real-world Datasets We evaluate our proposed models for node classification tasks on eight real-world datasets: Texas, Wisconsin, Cornell introduced by [20], Squirrel, Chalmeleon introduced by [21] and Cora, PubMeb, CiteSeer introduced by [17]. 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 10 public split introduced by [17] and calculate the average test accuracy and standard deviation.

Baselines and Experimental Setup We select several standard graph neural networks as our baselines: GCN [1], GAT [3], GraphSAGE [2], FAGCN [16], PairNorm[22], and UFG [9]. Among them, FAGCN chooses to shorten or enlarge the distance between node representations flexibly. PairNorm is specifically designed for oversmoothing problems. UFG is the frequency version of our

	Texas	Wisconsin	Squirrel	Chameleon	Cornell	CiteSeer	PubMed	Cora
Homophily level	0.11	0.21	0.22	0.23	0.30	0.74	0.80	0.81
#Nodes	183	251	5201	2277	183	3327	18717	2708
#Edges	295	466	198493	31421	280	4676	44327	5278
#Classes	5	5	5	5	5	7	3	6
GCN	55.1±5.2	51.8±3.1	53.2±2.1	64.8±2.4	60.5±5.3	71.9±1.8	78.7±2.9	81.5±1.3
GAT	52.2±6.6	49.4±4.1	40.7±1.5	60.3±2.5	61.9±3.1	71.4±1.9	78.7±2.3	81.8 ±1.3
GraphSAGE	**82.4**±6.1	81.2±5.6	41.6±0.7	58.7±1.7	76.0±5.0	71.6±1.9	77.4±2.2	79.2 ±7.7
FAGCN	**82.4**±6.9	**82.9**±7.9	42.6±0.8	55.2±3.2	**79.2**±3.2	72.3±0.8	79.4±0.3	**84.1**±0.5
PairNorm	60.3±4.3	48.4±6.1	50.4±2.0	62.7±2.8	58.9±3.2	**73.6**±1.5	**87.5**±0.4	**85.8**±1.0
UFG	79.3±2.8	78.8±3.2	**53.3**±1.5	**66.9**±1.1	75.3±1.1	72.7±0.6	79.7±0.1	83.6±0.6
EE-UFG (ours)	**81.8**±3.2	**83.3**±3.3	**55.3**±1.3	**68.0**±0.9	**77.8**±2.8	**73.7**±1.3	**81.5**±0.9	83.5±0.2

Table 1: Performance comparison on eight node classification tasks. The first and the second are highlighted by **First**, **Second.**

	Cora			CiteSeer			PubMed		
#Layer	**GCN**	**UFG**	**EE-UFG**	**GCN**	**UFG**	**EE-UFG**	**GCN**	**UFG**	**EE-UFG**
1	54.6±1.2	58.3±3.1	65.3±0.2	39.7±3.4	44.8±2.6	43.7±3.7	52.8±3.3	55.2±1.1	54.3±1.4
2	**81.5**±0.2	80.3±6.2	77.1±1.1	**68.7**±0.9	71.3±5.4	64.8±3.3	**76.6**±0.6	77.9±3.4	75.3±1.2
3	79.0±1.2	**83.6**±0.6	83.2±1.8	65.9±1.6	**72.7**±0.6	72.3±1.9	76.2±0.6	**79.7**±0.1	77.9±3.3
4	70.8±5.6	75.1±3.3	**83.5**±0.2	55.6±6.0	70.2±3.7	**73.8**±1.3	67.3±5.6	76.2±3.8	80.5±2.2
5	46.9±9.7	69.9±1.3	81.9±1.4	35.1±8.5	62.3±5.6	70.9±6.8	55.1±9.4	73.2±4.8	**81.5**±0.9
6	36.2±9.1	57.8±5.9	79.7±5.4	28.9±6.0	53.7±3.4	68.9±7.7	49.5±7.1	67.2±3.3	80.2±4.6

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**.

EE-UFG without framelet augmentation. We use NVIDIA A100 GPU for training, run 300 epochs and select the model with the highest validation accuracy. The hyper-parameter search space for EE-UFG is given in Appendix C.

Results Table 1 shows the performance comparison on eight node classification tasks. We can observe that for heterogeneous tasks, EE-UFG obtains a great boost compared with GCN and GAT, by better extracting the high-frequency information of the node itself. It is consistent with the idea of the previous literature [16, 17, 18], that for heterogeneous graphs, the aggregated information from adjacent nodes is usually the noisy information, and the detail information of the node itself should be better distilled. On the other hand, EE-UFG inherits multi-hop aggregation properties from the framelet transform, which takes into account all hops in the multiscale framelet representation, which is essential for heterogenous graphs.

It is known that the performance of GNNs will rapidly decay as the layers are stacked too much. The GCN-columns in Table 2 verifies this phenomenon. We can observe from the table that the UFG suffers less oversmoothing, partly because its adaptive filter learning in the frequency domain. Our proposed EE-UFG can basically circumvent the oversmoothing and keep a relative good and stable performance as the number of layers increases. On the other hand, we can see that the best performance of EE-UFG occurs at a deeper layers than GCN and UFG.

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 Laplacian-Beltrami operator $\mathcal{L}_B$ which is defined as $\mathcal{L}_B f = -\text{div}(\nabla f)$. 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 [23]. Using the same formula for the graph, but replacing the graph Laplacian by the

Laplacian-Beltrami operator $\mathcal{L}_B$, we can define *Manifold framelet transforms* as

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

Example: Diffusion Problems on Manifold The Laplacian-Beltrami operator is closely related to the diffusion process over the manifold, which is governed by the PDE:

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

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 Eq. 10, a framelet manifold convolution for diffusion process can be derived as follows, for $(r,j) \in \{(r,j)|r = 1,\cdots,n; j = 1,\cdots,J\}$,

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

where $\mathcal{L}_B^L$ and $\mathcal{L}_B^H$ are defined similarly in Eq. 9. In general, our proposed framelet augmentation method can be naturally extended to those (symmetric) Laplacian-based propagation rules, due to the complete framelet theory on manifold.

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 simplical complex, where the simplicial Hodge Laplacian is used to provide the connection information. The adjacency matrices on these higher-dimensional domains are non-symmetric, due to the different numbers of simplex of different orders. In such cases, our framelet augmentation can not be implemented. We will consider framelet augmentation strategy for these cases in future works.

6 Related Work

Oversmoothing and Dirichlet Energy One of the widely known plights of GNNs is oversmoothing, which was first introduced by [24]. This phenomenon has been studied by [25, 26, 5, 17]. The Dirichlet energy was commonly used in these studies since it can indicate the oversmoothing level and measure the expressive power. Explanation paying attention to the structure of Laplacian has been undergone by [24, 27, 5, 4]. If the weight matrix [4] or the block structures [24] satisfy certain conditions, oversmoothing will make the features indistinguishable and hurt the classification accuracy. Hence, there are two main ideas to solve the problem. One is to relive the adjacent matrix by sparsification [28]. Another is to scale node representations to avoid features caught into the invariant regime. Adding residual connections [18] proves useful although it seems to violate intuition [5]. Many other attempts beyond the graph matrix analysis also emerged like GCON [29] using ODE dynamics and [7] using celluar sheaf theory. Besides, in the field of spectra analysis, GNNs' updating process can be viewed as tackling low-frequency information [11, 16]. This inspired FAGCN [16] to design different filters for low/high-frequency signals respectively.

Wavelet Analysis on Graphs [30] proposed a formal approach to spatial traffic analysis on the wavelet transform. This is the first exploration of wavelet analysis on graphs. Polynomials of a differential operator were used to build a multiscale tight frame by [10]. [31] gave the tight framelets framework on manifolds, which was then extended to graphs by [13, 32] with the fast decomposition and reconstruction algorithms on undecimated and decimated frames for graph signals. In the regime of signal processing, [33] established a tree-based wavelet system with localization properties, which is a milestone in the multi-reslution analysis. [34] apply harmonic analysis to semi-supervised learning and construct Haar-like bases for it. [35, 36] used the Haar-like wavelets system [37] to cope with deep learning tasks.

7 Conclusion

In this work, we develop framelet analysis on graphs and generalized the generic graph convolution to a framelet version in the spatial domain. Due to the energy difference between the low pass and high passes, we originally propose framelet augmentation which can increase the Dirichlet Energy associated with the graph and keep it at a relative high and steady value. In practice, we demonstrate the behaviors of framelet features during the training and the effectiveness of framelet augmentation to relieve the oversmoothing 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.

[2] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *NeurIPS*, 30, 2017.

[3] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lió, and Yoshua Bengio. Graph attention networks. *ICLR*, 2017.

[4] Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. *ICML Workshop on Graph Representation Learning*, 2020.

[5] Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *ICLR*, 2020.

[6] 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.

[7] 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.

[8] Bingbing Xu, Huawei Shen, Qi Cao, Yunqi Qiu, and Xueqi Cheng. Graph wavelet neural network. In *ICLR*, 2019.

[9] 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.

[10] Mauro Maggioni and Hrushikesh N Mhaskar. Diffusion polynomial frames on metric measure spaces. *Applied and Computational Harmonic Analysis*, 24(3):329–353, 2008.

[11] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *ICML*, pages 6861–6871, 2019.

[12] Qimai Li, Xiao-Ming Wu, Han Liu, Xiaotong Zhang, and Zhichao Guan. Label efficient semi-supervised learning via graph filtering. In *CVPR*, pages 9582–9591, 2019.

[13] Bin Dong. Sparse representation on graphs by tight wavelet frames and applications. *Applied and Computational Harmonic Analysis*, 42(3):452–479, 2017.

[14] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *AAAI*, 2018.

[15] 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.

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

[17] Hoang Nt and Takanori Maehara. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019.

[18] 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.

[19] Nitika Verma, Edmond Boyer, and Jakob Verbeek. Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *CVPR*, pages 2598–2606, 2018.

[20] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-GCN: geometric graph convolutional networks. In *ICLR*, 2020.

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

[22] Lingxiao Zhao and Leman Akoglu. PairNorm: tackling oversmoothing in GNNs. In *ICLR*, 2020.

[23] 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.

[24] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. DeepGCNs: Can GCNs go as deep as CNNs? In *CVPR*, pages 9267–9276, 2019.

[25] 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.

[26] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *ICLR*, 2019.

[27] 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.

[28] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. DropEdge: towards deep graph convolutional networks on node classification. In *ICLR*, 2020.

[29] T Konstantin Rusch, Benjamin P Chamberlain, James Rowbottom, Siddhartha Mishra, and Michael M Bronstein. Graph-coupled oscillator networks. *arXiv preprint arXiv:2202.02296*, 2022.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] Yu Guang Wang, Ming Li, Zheng Ma, Guido Montufar, Xiaosheng Zhuang, and Yanan Fan. Haar graph pooling. In *ICML*, pages 9952–9962, 2020.

[36] Ming Li, Zheng Ma, Yu Guang Wang, and Xiaosheng Zhuang. Fast haar transforms for graph neural networks. *Neural Networks*, 128:188–198, 2020.

[37] 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.

[38] Jakob Hansen and Thomas Gebhart. Sheaf neural networks. *arXiv preprint arXiv:2012.06333*, 2020.

[39] 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.

[40] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[41] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[42] Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Moritz Hardt, Ben Recht, and Ameet Talwalkar. Massively parallel hyperparameter tuning. 2018.

[43] 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.

A Theoretical Support

A.1 Framelet Dirichlet Energy Conservation

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

Proposition 5 *The Dirichlet energy is conserved under framelet decomposition:*

$$E_{total}(X) = E(X). \tag{14}$$

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

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

where $(UX)_i$ is the ith component of UX, and $\lambda_i, \lambda_{r,j}^i$ are the eigen-values of Λ, $\Lambda_{r,j}$. 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. ■

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. 3, where $(r,j) \in \{(r,j)|r = 1, \cdots, n; j = 1, \cdots, J\} \cup \{(0,J)\}$.*

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 ith eigenvalue of $\Lambda_{r,j}$. ■

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 6 *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

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

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

$$\|\widehat{V}_0\|^2 - \|\widehat{W}_1^1\|^2 - \|\widehat{W}_2^1\|^2 = \|\widehat{X}\|^2\Big(\big\|\widehat{\alpha}(\Lambda/2)\big\|^2 - \big\|\widehat{\beta}(\Lambda/2)\big\|^2 - \big\|\widehat{\beta}(\Lambda/4)\big\|^2\Big) \tag{15}$$

The RHS of (15) 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 $\|W_1^1\|^2 + \|W_2^1\|^2 \leq \|V_0\|^2$. ■

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 Dirichelet energy is enhanced with framelet augmentation.

Proposition 7 *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., $E^{\epsilon}_{total}(X) > E_{total}(X) = E(X)$.*

Proof.

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

By Proposition 6 and its specific framelet system, $\Lambda_{0,J}^2 - \sum_{r,j}\Lambda_{r,j}^2 \geq 0$, thus, $E^{\epsilon}_{total}(X) \geq E(X)$. ■

A.4 Asymptotic Behavior of EE-UFG

Proposition 8 *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 ($k = 1, 2, \cdots, n$), 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:*

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

where A^L and A^H are low-pass and high-passes adjacency matrices as defined in Eq. 8.

Proof. By Eq. 8, 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 $\epsilon \geq 0$. ■

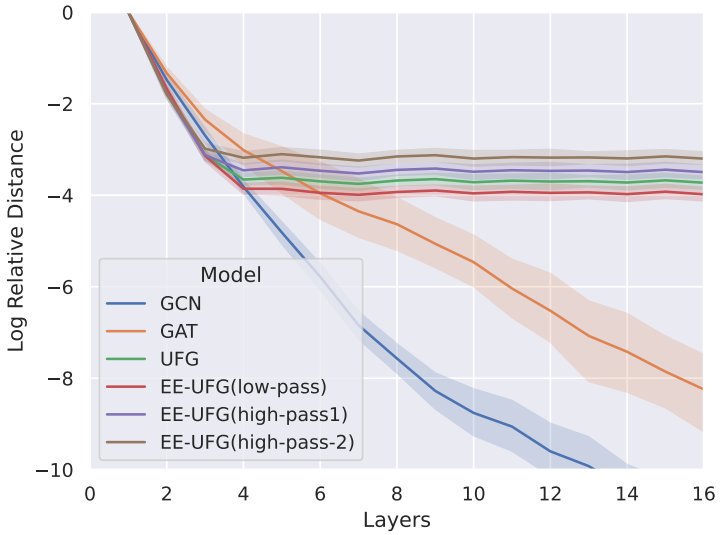


Figure 4: Layer-wise distances from the feature to the subspace $\mathcal{M}$. The result is the average of 100 runs. The Y-axis is the log relative distance, defined by $y^{(l)} = \log(d_{\mathcal{M}}(X^{(l)})/d_{\mathcal{M}}(X^{(0)}))$. $X^{(0)}$ is the initial feature representation, and $X^{(l)}$ 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 $\mathcal{M} = U \otimes \mathbb{R}^C = \left\{\sum_{m=1}^{M} e_m \otimes w_m | w_m \in \mathbb{R}^C, e_m \in U\right\} \subseteq \mathbb{R}^{N\times C}$, 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 3. The layer-wise outputs of GCN and GAT exponentially approaches the subspace $\mathcal{M}$. This subspace is invariant under any polynomial of 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 [5].

B An Example: Enhancing Energy for Sheaf Convolution

Our framelet systems and Dirichlet energy enhancing strategy are generalizable to other extension of (symmetric) Laplacian. Here we discuss Sheaf Lapacians 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 (5) and then the enhanced sheaflet convolution as (10). The latter can be proved to follow the same energy enhancement as the graph framelet convolution.

B.1 Sheaf Laplacian

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 $\mathcal{F}_{v\triangleleft e}$: $\mathcal{F}(v) \to \mathcal{F}(e)$ for each incident vertex-edge pair $v \triangleleft e$ of $\mathcal{G}$.

Construct the *Sheaf Laplacian* $L_{\mathcal{F}} : C^0(\mathcal{G}, \mathcal{F}) \to C^0(\mathcal{G}, \mathcal{F})$, 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-diagnoal 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, consists of a class of linear operators over the graph, thus allowing 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.2 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*

$$\begin{aligned} \phi_{j,p}(v) &:= \sum_{l=1}^{Nd} \widehat{\alpha}\left(\frac{\lambda_l}{2^j}\right) \overline{u_l(p)} u_l(v), \\ \psi_{j,p}^r(v) &:= \sum_{l=1}^{Nd} \widehat{\beta}\left(\frac{\lambda_l}{2^j}\right) \overline{u_l(p)} u_l(v), \quad r = 1, \ldots, n. \end{aligned} \tag{16}$$

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* $\mathrm{UFS}(\Psi, \eta; \mathcal{G})$ (starting from a scale J_1) as a non-homogeneous, stationary affine system:

$$\begin{aligned} \mathrm{UFS}_{J_1}^J(\Psi, \eta) &= \mathrm{UFS}_{J_1}^J(\Psi, \eta; \mathcal{G}) \\ &:= \{\phi_{J_1,p} : p \in V\} \cup \{\psi_{j,p} : p \in V, j = J_1, \ldots, J\}_{r=1}^n. \end{aligned} \tag{17}$$

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

$$V_0 = \langle \phi_{0,\cdot}, X \rangle = U^\top \widehat{\alpha}\left(\frac{\Lambda}{2}\right) UX \;\text{ and }\; W_j^r = \left\langle \psi_{j,\cdot}^r, X \right\rangle = U^\top \widehat{\beta^{(r)}}\left(\frac{\Lambda}{2^{j+1}}\right) UX, \tag{18}$$

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

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

B.3 Implementation Format

To reduce the computational complexity caused by eigendecomposition for Sheaf Laplacians, we use Chebyshev polynomials to approximate. Consider Chebyshev polynomials $\mathcal{T}_0, \cdots, \mathcal{T}_n$ of fixed degree t, and filter $a \approx \mathcal{T}_0$ and $b^{(r)} \approx \mathcal{T}_r$, then the above 3 can be approximated

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

$\mathcal{L}_F$ is the sheaf Laplacian.

B.4 From Sheaf Convolution to Sheaflet Convolution

Sheaf convolution [7, 38] is defined as follows,

$$Y = \sigma((I_{Nd} - L_{\mathcal{F}})(I_N \otimes W_1) X W_2) \in \mathbb{R}^{Nd \times f_2}, \tag{19}$$

where $(I_N \otimes W_1) X W_2 = \tilde{X} \in \mathbb{R}^{Nd \times f_2}$. Inheriting the characteristics of sheaf convolution in Eq. (19), we define *Sheaflet Convolution* based on the sheaf framelet system as

$$\begin{aligned} Y_{0,J} &= \sigma((I_{Nd} - \mathcal{L}_F)(I_N \otimes W_1) \mathcal{W}_{0,J} X W_{0,J}), \\ Y_{r,j} &= \sigma((I_{Nd} - \mathcal{L}_F)(I_N \otimes W_1) \mathcal{W}_{r,j} X W_{r,j}), \\ Y &= \mathcal{V}(Y_{0,J}; Y_{1,1}, \cdots, Y_{n,J}). \end{aligned} \tag{20}$$

B.5 Dirichlet Energy for Sheaflets

Applying our Dirichelet energy enhancement strategy to Sheaflet Convolution (Eq. 20), we obtain the following *Energy Enhanced Sheaflet Convolution*,

$$\begin{aligned} Y_{0,J} &= \sigma((I_{Nd} - \mathcal{L}_F^L)(I_N \otimes W_1)\mathcal{W}_{0,J} X W_{0,J}), \\ Y_{r,j} &= \sigma((I_{Nd} - \mathcal{L}_F^H)(I_N \otimes W_1)\mathcal{W}_{r,j} X W_{r,j}), \\ Y &= \mathcal{V}(Y_{0,J}; Y_{1,1}, \cdots, Y_{n,J}), \end{aligned} \tag{21}$$

where $\mathcal{L}_F^L$ and $\mathcal{L}_F^H$ are defined similarly as Eq. 9.

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

$$\begin{aligned} E_{0,J}^{\epsilon}(X) &= ((I_N \otimes W_1)\mathcal{W}_{0,J} X)^\top (\mathcal{L}_F + \epsilon D^{-1})((I_N \otimes W_1)\mathcal{W}_{0,J} X) \\ E_{r,j}^{\epsilon}(X) &= ((I_N \otimes W_1)\mathcal{W}_{r,j} X)^\top (\mathcal{L}_F - \epsilon D^{-1})((I_N \otimes W_1)\mathcal{W}_{r,j} X), \end{aligned} \tag{22}$$

where D is the degree matrix of the sheaf Laplacian. The total sheaflet Dirichelet energy $E_{\mathcal{F}}^{\epsilon}(X) = E_{0,J}^{\epsilon}(X) + \sum_{r,j} E_{r,j}^{\epsilon}(X)$. The original sheaf Dirichelet 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 [39] on NVIDIA Tesla A100 GPU with 6,912 CUDA cores and 80GB HBM2 mounted on an HPC cluster. PyTorch Geometric Library [40] is employed for all the benchmark datasets and baseline models. For each model, we run 300 epochs 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 [17], 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 [20]. 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 [21]. Nodes are the web pages and edges are the hyperlinks between them. Node features represent several informative nouns in the Wikipedia pages.

All benchmark datasets are available at `https://pytorch-geometric.readthedocs.io/en/latest/modules/datasets.html`.

C.3 Hyper-parameter and Model Implementation

We employ Adam [41] as our optimizer, and `ASHAScheduler` [42] as our scheduler. Each model is fine-tuned with `Ray` [43]. Table 3 provides the hyper-parameter search space for reproduction. All baseline models, except UFG, are available at `https://pytorch-geometric.readthedocs.io/en/latest/modules/nn.html`. We use the official implementation of UFG from the repository: `https://github.com/YuGuangWang/UFG`.

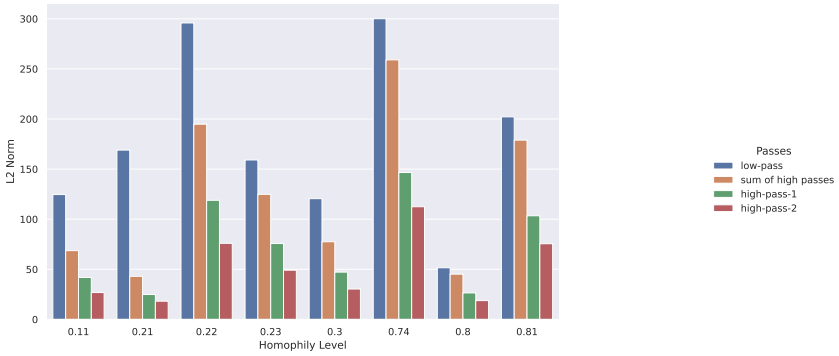


Figure 5: L_2 norm of framelet coefficients for the 8 datasets in Appendix C.2.

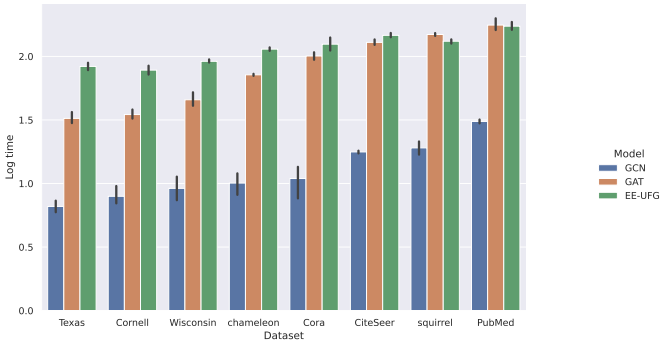


Figure 6: Running time comparison

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.4 Computational Complexity

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 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 the computational complexity close to GAT, especially when the number of nodes is large.