Dirichlet Energy Enhancement of Graph Neural Networks by Framelet Augmentation

Jialin Chen, Yuelin Wang, Cristian Bodnar, Rex Ying, Pietro Liò, Yuguang Wang

Submitted to LoG 2022

Limitation of GNNs

- **Over-smoothing issue:** the node representations tend to be **similar and indistinguishable** when we stack the layers of the model => model's performance **decays** rapidly w.r.t. **the number of layers**
- **Message-passing mechanism:** **aggregate** the neighboring information and then **update** the feature of the central node
- Especially detrimental in **heterophilous cases**, where the adjacent nodes are more likely to have different labels

Existing Methods

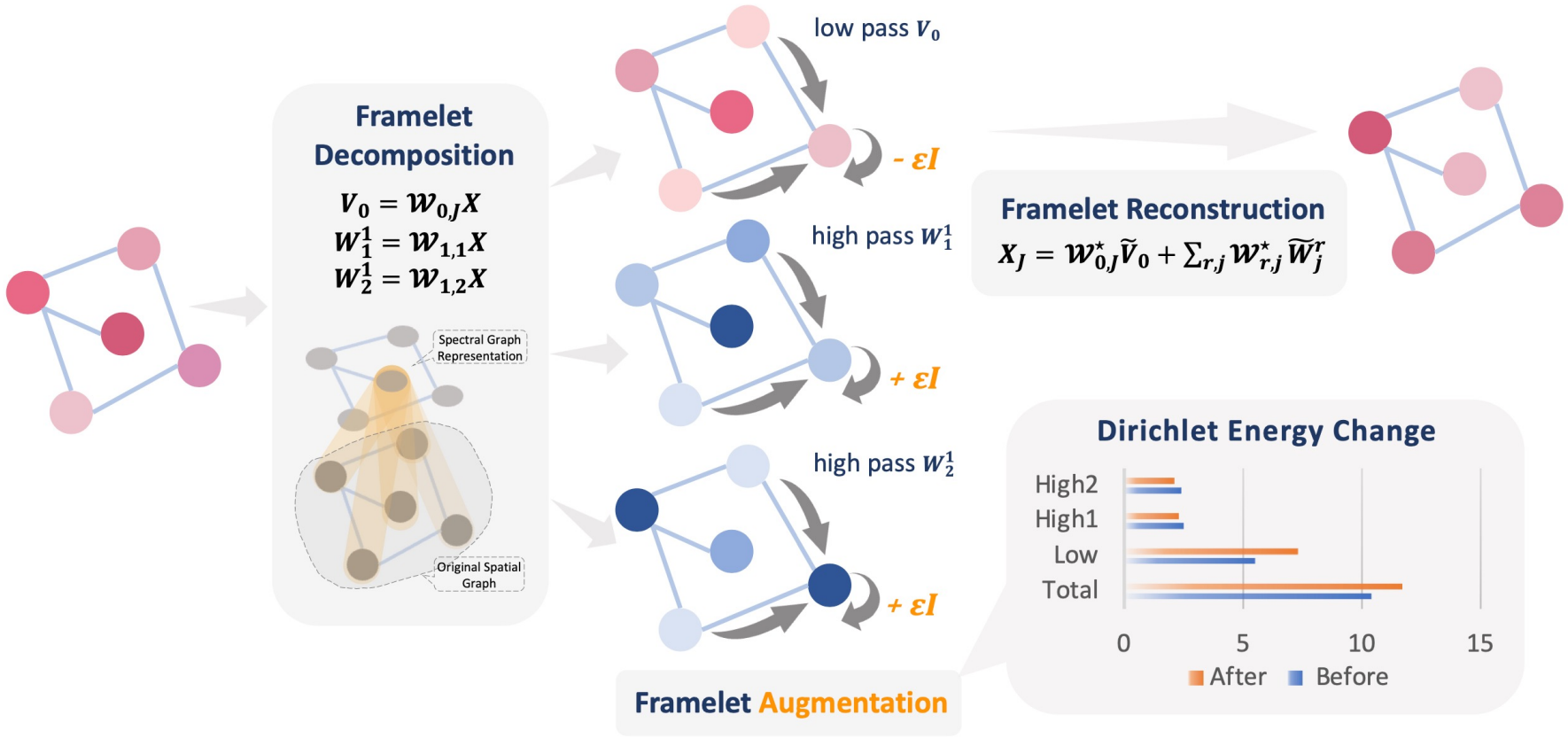
- **Techniques** in graph convolutional layers:
  - Residual connections[1,2] , weight normalization[3] , edge dropout[4], model simplification[5], etc.
- **Control Dirichlet energy** [6](a metric to measure the average feature distance between connected nodes)
  - Orthogonal weight controlling
  - Lower-bounded Residual Connection
  - Special activation
- **Limitation:**
  - Only consider spatial information
  - Sacrifice performance in deep-layers model to some extent

[1] DeepGCNs: Can GCNs Go as Deep as CNNs?
[2] Representation Learning on Graphs with Jumping Knowledge Networks
[3] PairNorm: tackling oversmoothing in GNNs
[4] DropEdge: towards deep graph convolutional networks on node classification
[5] Simplifying Graph Convolutional Networks
[6] Dirichlet Energy Constrained Learning for Deep Graph Neural Networks

Insightful Facts

- Graph convolution
  - works well for the case where the low-frequency components are sufficient for prediction
  - fails in the scenarios where the high-frequency information is also necessary, which often happens in real-world datasets
- **Main cause:** the denoising effect of graph convolution layers and insufficient attention to high-frequency components
- **Our work:** decouple low-pass and high-passes, characterize their different roles in graph prediction, strictly enhance Dirichlet energy by regulating the energy ratio of each frequency component.

Energy Enhanced Convolution



Step1: conduct framelet decomposition to obtain one low pass and two high passes.

Step2: apply the Framelet Augmentation by adding or subtracting an increment for low and high passes. The total Dirichlet energy will be lifted in this process.

Step3: A framelet reconstruction operator follows to resize the framelet coefficients to the original size.

Framelets for Graph

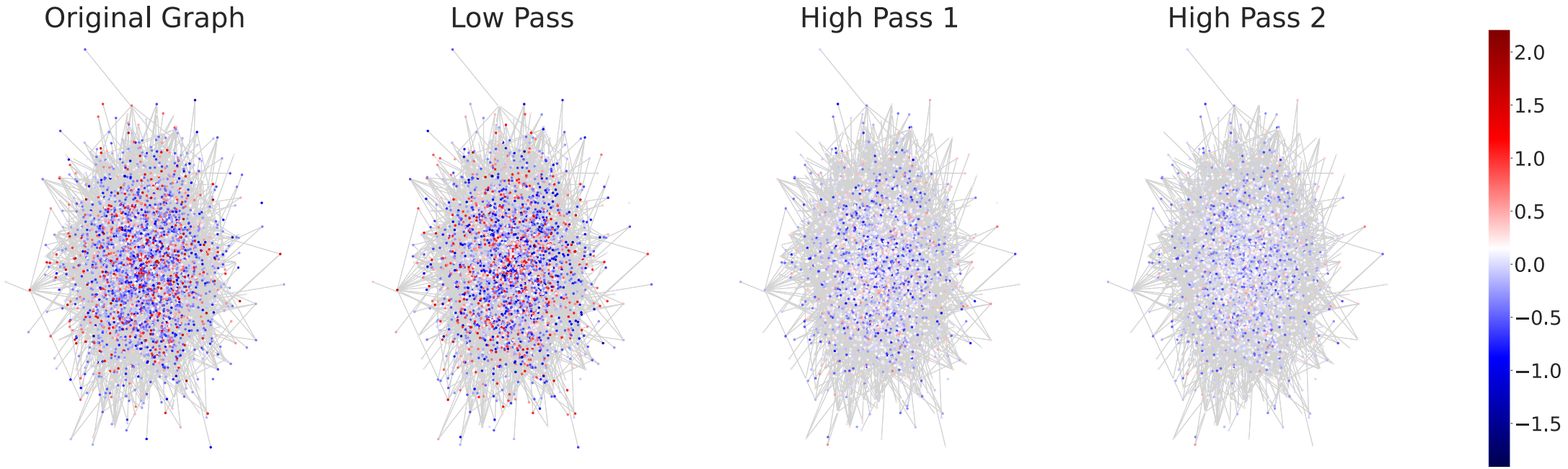
- For a Graph with N nodes and graph Laplacian Δ. $U = [u_1, \dots u_N]$ be the matrix of eigenvector of Δ, and $\Lambda = diag(\lambda_1, \dots, \lambda_N)$ be the diagonal matrix of the eigenvalues. Framelets over the graph is defined by a set of scaling functions $\Phi = \{\alpha; \beta^{(1)}, \dots, \beta^{(n)}\}$.
- Framelets for graph:

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

  - The framelet transforms take into account the global information and all hops of the graph into the multi-scale graph representation
- Extracted **low-pass node features and high-pass node features**:

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

Visualization of Framelet



- Low-pass node feature $V = \mathcal{W}_{0,J}X$; and high-pass node feature $W_j^r = \mathcal{W}_{r,j}X$ (V, W_j^r and X have the same size). In our implementation, we use the system of 2 scale level ($j = 1,2$) and 1 high-pass filter ($r = 1$).
- **Energy gap:** we have $\| \boldsymbol{W_1^1} \|^2 + \| \boldsymbol{W_2^1} \|^2 \leq \| \boldsymbol{V_0} \|^2$, which motivate us to consider the **energy imbalance** between low and high-passes.

Dirichlet Energy

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

- Over-smoothing representations produce a small value of Dirichlet energy
- Node features converge to a subspace of the eigenspace of graph Laplacian Δ during the feature propagation
  - Convergence rate is positively related to the second largest eigenvalue of the augmented adjacenc matrix $\hat{A}$ and singular value of the l-th layer weight matrix.
- **Key idea: enhance Dirichlet energy during the propagation and modify the feature convergence rate**

Framelet Dirichlet Energy

- We define **framelet Dirichlet energy** for low-pass and high-passes parts:

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

- The **total framelet Dirichlet energy** is defined as $E_{total} = \sum_{r,j} E_{r,j} + E_{0,J}$
- **Decomposability of Dirichlet energy:**

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

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

Our Algorithm

- We **add or subtract an increment** for low and high passes adjacency matrix respectively:

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

- **Layer-wise propagation rule** of Energy Enhanced Convolution:

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

Dirichlet Energy Enhancement

- **Modified Framelet Dirichlet Energy**

$$E^{\epsilon}_{0,J}(X,) = (\mathcal{W}_{0,J}X)^{\top}\tilde{\Delta}^{L}(\mathcal{W}_{0,J}X) = (\mathcal{W}_{0,J}X)^{\top}(\tilde{\Delta} + \epsilon\widehat{D}^{-1})(\mathcal{W}_{0,J}X)$$

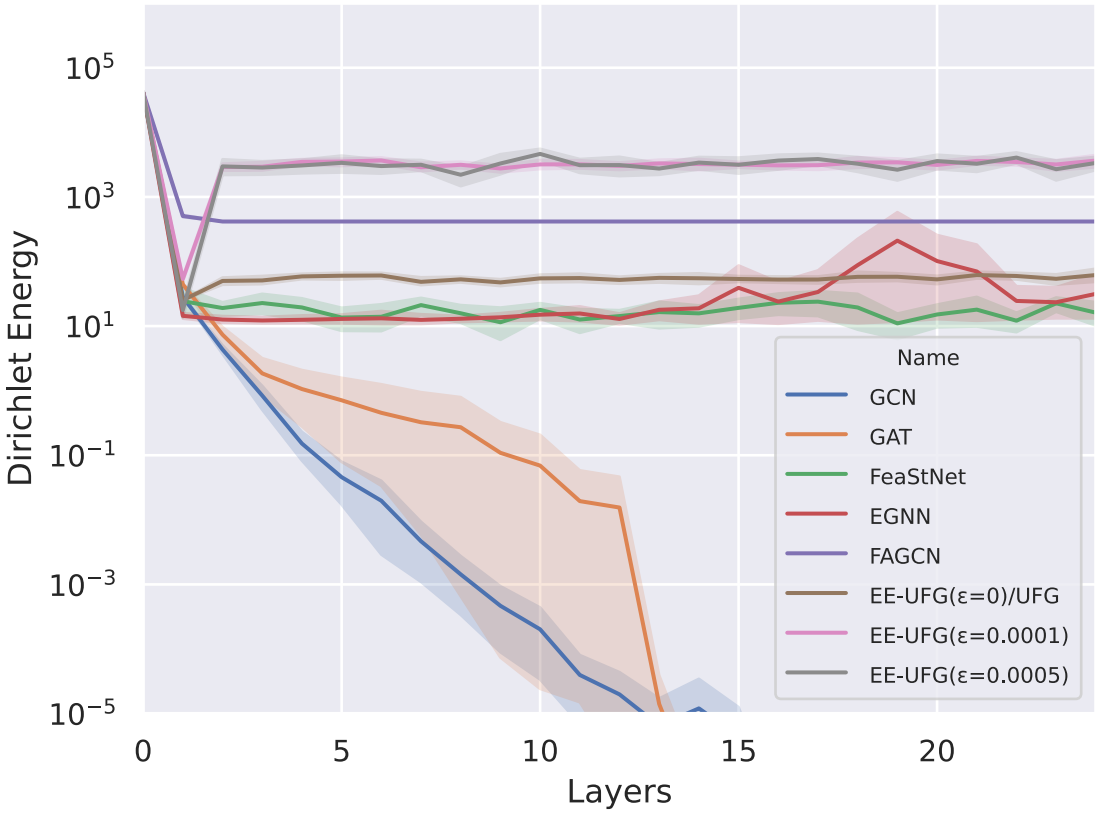
$$E^{\epsilon}_{r,j}(X) = (\mathcal{W}_{r,j}X)^{\top}\tilde{\Delta}^{H}(\mathcal{W}_{r,j}X) = (\mathcal{W}_{r,j}X)^{\top}(\tilde{\Delta} - \epsilon\widehat{D}^{-1})(\mathcal{W}_{r,j}X)$$

- Total framelet Dirichlet energy will be **enhanced**

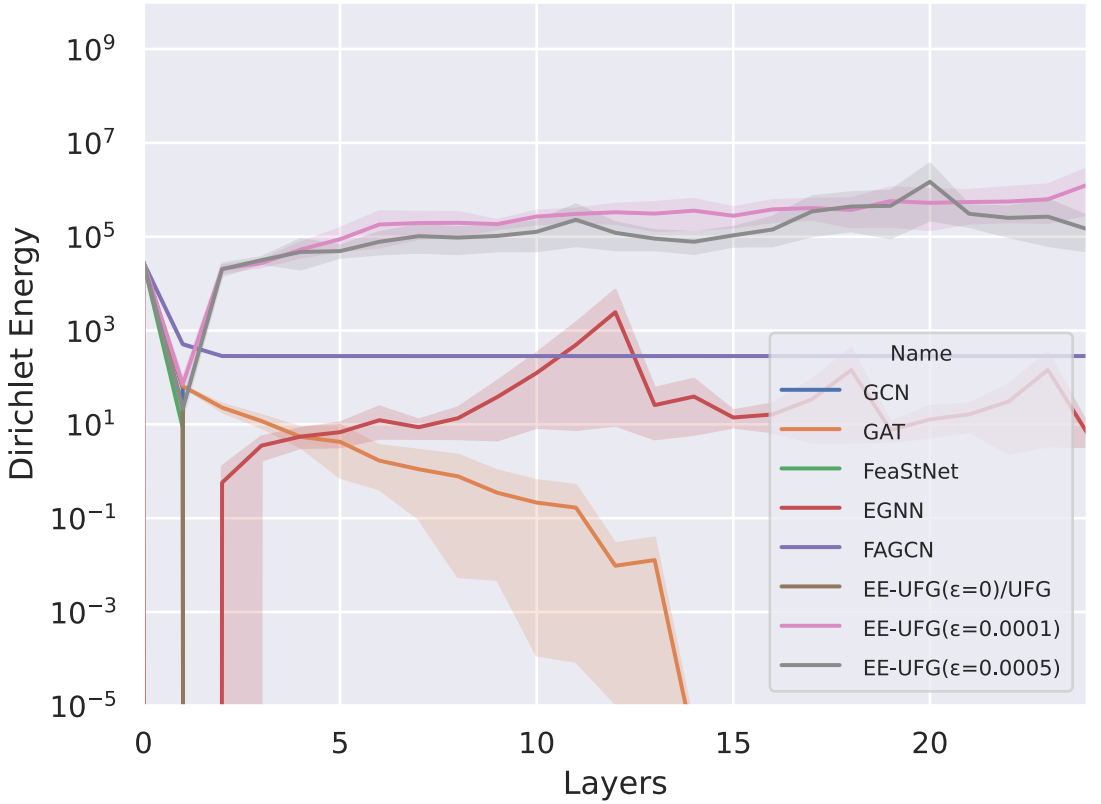
$$E^{\varepsilon}_{total}(X) = \sum_{r,j} E^{\varepsilon}_{r,j}(X) + E^{\varepsilon}_{0,J}(X) > E_{total}(X) = E(X) \text{ when } (\varepsilon > 0)$$

- **Self-attention and self-connection** to high-pass components
  - Strengthen the model's focus on high-frequency components of node itself
- **Self-impairment** to low-pass components
  - Increase the low-pass Dirichlet energy and obtain an overall enhanced Dirichlet energy (guaranteed by Enegry gap)

Dirichlet Energy Experiments

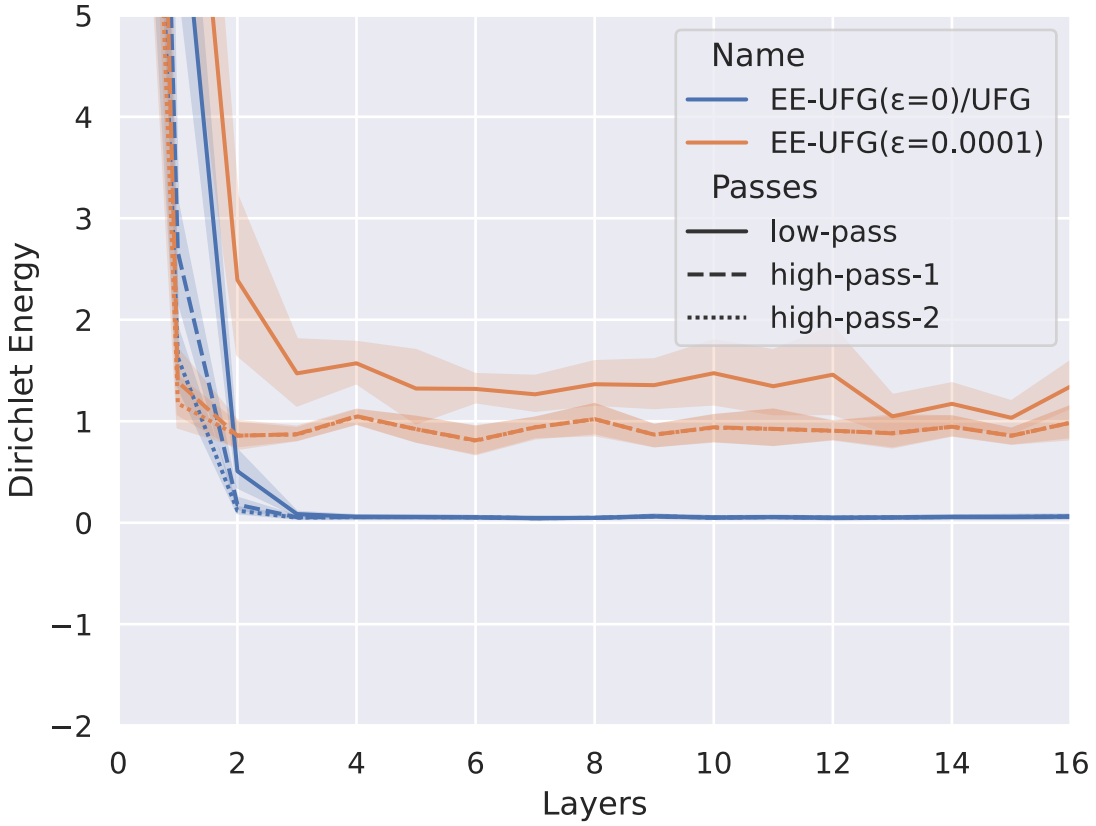


Dirichlet energy on Cora dataset with H=0.81

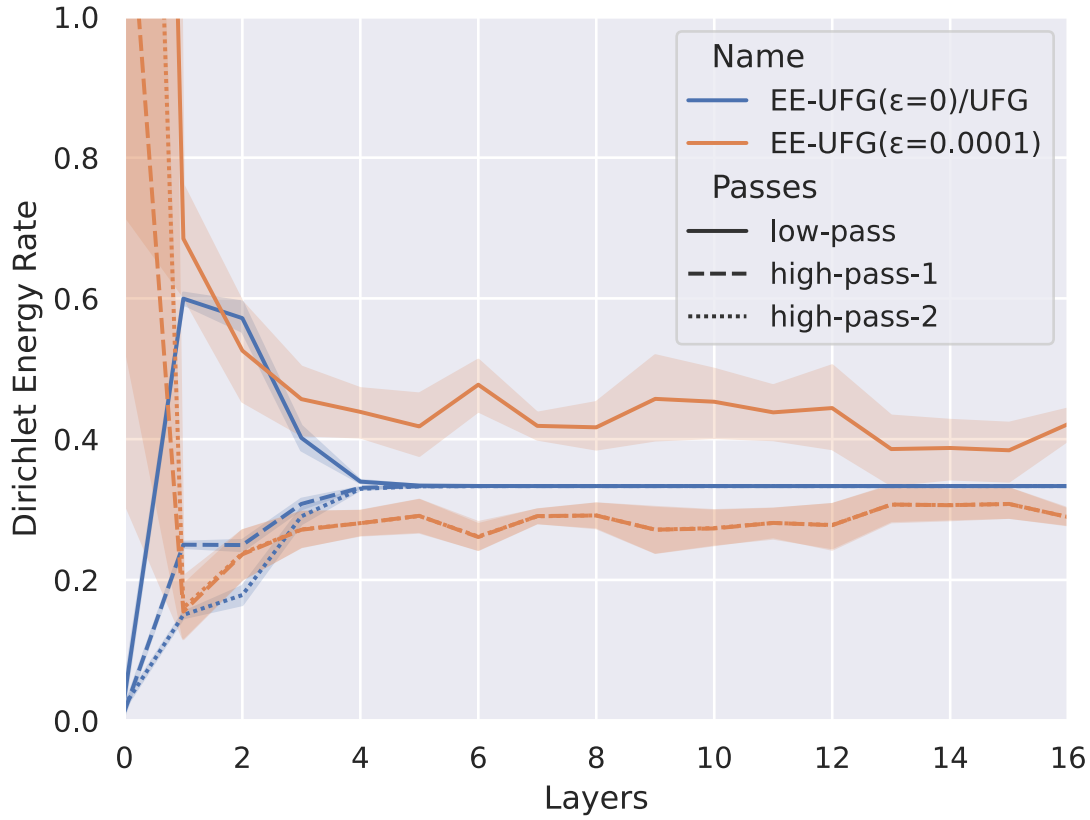


Dirichlet energy on Chameleon dataset with H=0.23

Dirichlet Energy Experiments

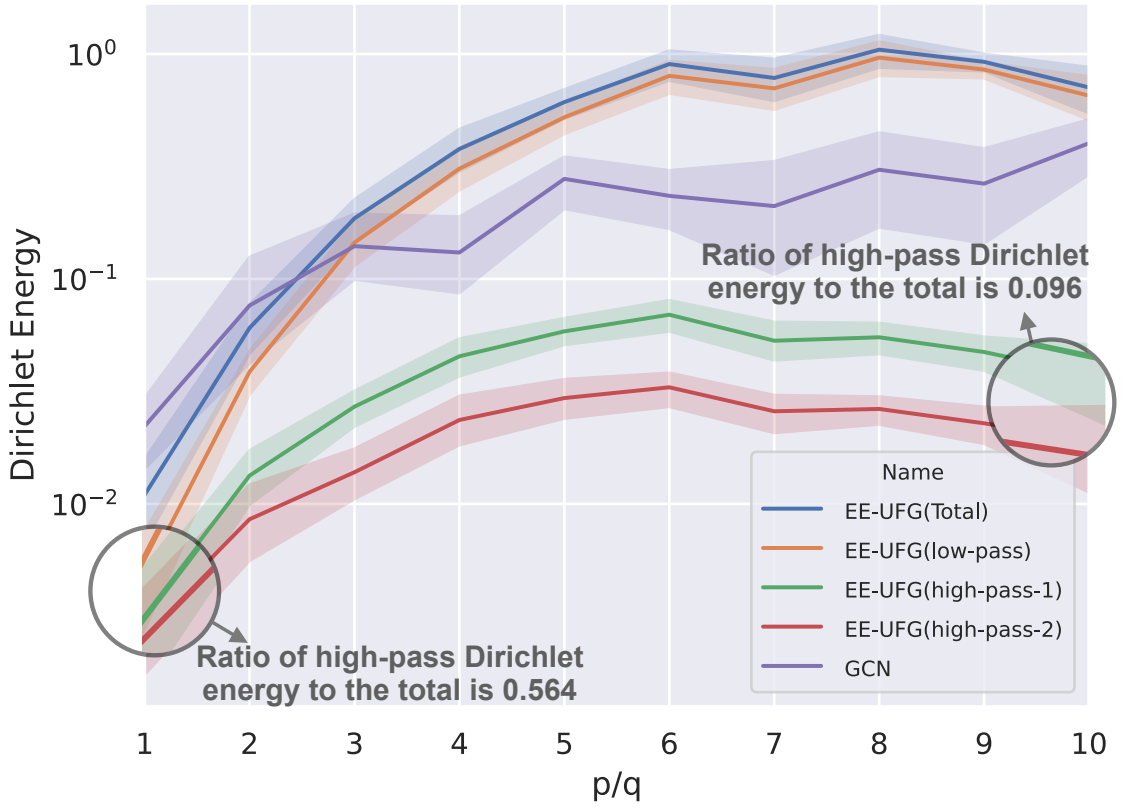


Framelet Dirichlet energy

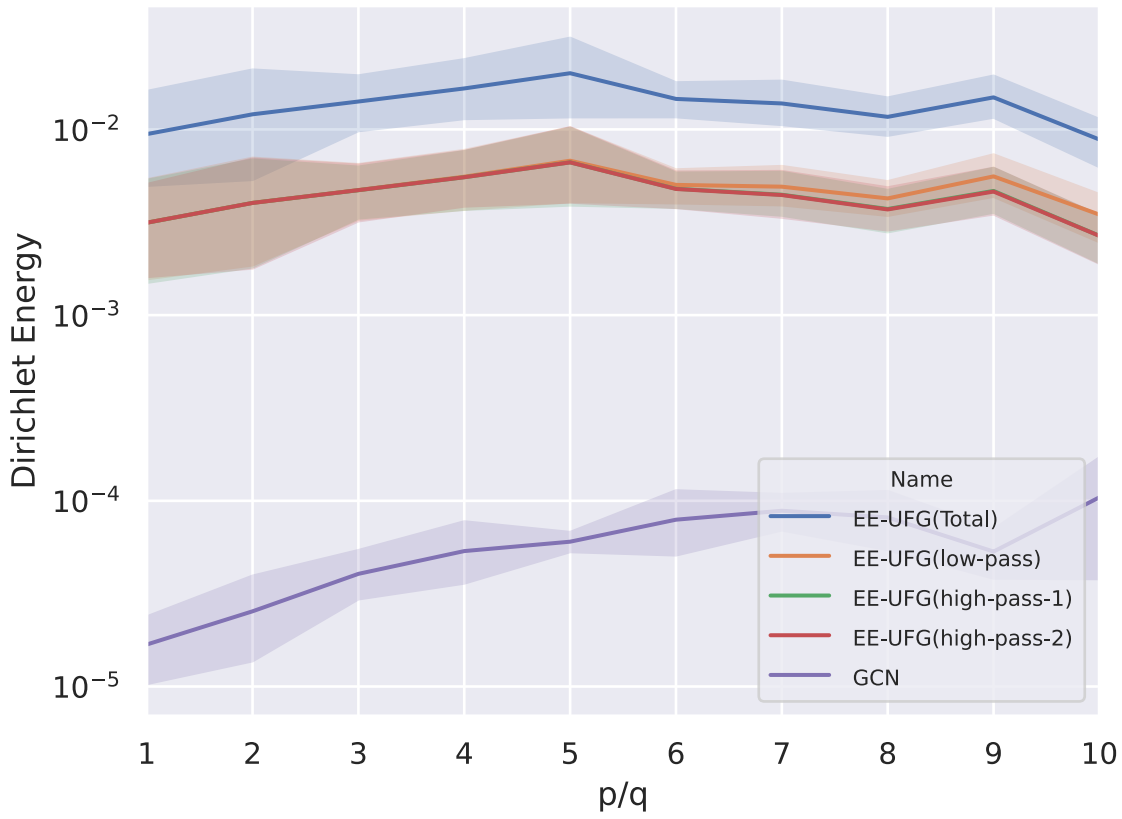


Framelet Dirichlet energy ratios

Dirichlet Energy Experiments



Dirichlet energy with different p/q ratio (3 Layers)



Dirichlet energy with different p/q ratio (8 Layers)

Real-world Datasets

Classic GNNs

GNNs for oversmoothing

GNNs for heterophily

	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	**82.4**±6.1	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	82.2±7.1	72.2±0.2	74.1±1.7	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	80.3±0.4	85.5±0.5	4.5
EGNN	81.0±0.8	**88.6**±3.2	48.3±2.3	62.7±2.6	**83.8**±4.6	72.7±1.2	70.4±2.8	80.1±3.6	**85.7**±3.7	3.3
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	53.3±1.5	66.9±1.1	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)	82.3±3.2	85.3±3.3	**55.3**±1.3	**68.0**±0.9	82.2±2.8	**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.

Over-smoothing Test

	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**.