

A Homophily Metrics

Here we review some commonly used metrics to measure homophily [23, 64]. We denote $\mathbf{z} \in \mathbb{R}^N$ as labels of nodes, and $\mathbf{Z} \in \mathbb{R}^{N \times C}$ as the one-hot encoding of labels, where C is the number of classes. Edge homophily H_{edge} and node homophily H_{node} are defined as follows:

$$H_{\text{edge}} = \frac{|\{e_{ij} | e_{ij} \in \mathcal{E}, \mathbf{z}_i = \mathbf{z}_j\}|}{|\mathcal{E}|}, \quad H_{\text{node}} = \frac{1}{|\mathcal{V}|} \sum_{u \in \mathcal{V}} \frac{|\{y_u = y_v : v \in \mathcal{N}(u)\}|}{|\mathcal{N}(u)|}. \quad (18)$$

Adjusted edge homophily H_{edge}^* considers classes imbalance and is defined as [50]:

$$H_{\text{edge}}^* = \frac{H_{\text{edge}} - \sum_c p^2(c)}{1 - \sum_c p^2(c)}. \quad (19)$$

Here $p(c) = \sum_{i: \mathbf{z}_i = c} \mathbf{D}_{ii} / (2|\mathcal{E}|)$, $c = 1 : C$, defines the degree-weighted distribution of class labels. The class homophily is also proposed to take class imbalance into account [65]:

$$H_{\text{class}} = \frac{1}{C-1} \sum_c \left[h_c - \frac{|\{v_i | \mathbf{z}_i = c\}|}{N} \right]_+ \quad (20)$$

$$h_c = \frac{\sum_{v_i: \mathbf{z}_i = c} |\{e_{ij} | e_{ij} \in \mathcal{E}, \mathbf{z}_i = \mathbf{z}_j\}|}{\sum_{v_i: \mathbf{z}_i = c} \mathbf{D}_{ii}}. \quad (21)$$

Label informativeness, which indicates the amount of information a neighbor's label provides about node's label, is defined as follows [50]:

$$LI = 2 - \frac{\sum_{c_1, c_2} p(c_1, c_2) \log p(c_1, c_2)}{\sum_c p(c) \log p(c)}, \quad (22)$$

where $p(c_1, c_2) = |\{e_{ij} | e_{ij} \in \mathcal{E}, \mathbf{z}_i = c_1, \mathbf{z}_j = c_2\}| / (2|\mathcal{E}|)$. The aggregation homophily $H_{\text{agg}}^M(\mathcal{G})$ measures the proportion of nodes that assign greater average weights to intra-class nodes than inter-class nodes. It is defined as follows [39]:

$$H_{\text{agg}}^M(\mathcal{G}) = \frac{1}{|\mathcal{V}|} |\{v_i | \text{Mean}_j(\{S(\hat{\mathbf{A}}, \mathbf{Z})_{ij} | \mathbf{z}_i = \mathbf{z}_j\}) \geq \text{Mean}_j(\{S(\hat{\mathbf{A}}, \mathbf{Z})_{ij} | \mathbf{z}_i \neq \mathbf{z}_j\})\}|, \quad (23)$$

where $S(\hat{\mathbf{A}}, \mathbf{Z}) = \hat{\mathbf{A}}\mathbf{Z}(\hat{\mathbf{A}}\mathbf{Z})^\top$ defines the post-aggregation node similarity, with $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, and $\text{Mean}_j(\{\cdot\})$ takes the average over node v_j of a given multiset of values. Under the homophily metrics mentioned above, a smaller value indicates a higher degree of heterophily. While H_{edge}^* can assume negative values, the other metrics fall within the range $[0, 1]$.

B Proof of Theorem

B.1 Proof of theorem 3.1

Proof. By Definition 3.1, we have

$$\begin{aligned} \mathbf{P}^{(\alpha, \gamma)} &= \mathbf{I} - \mathbf{L}^{(\alpha, \gamma)} = \mathbf{I} - \gamma[\gamma\mathbf{D} + (1-\gamma)\mathbf{I}]^{-\alpha} \mathbf{L} [\gamma\mathbf{D} + (1-\gamma)\mathbf{I}]^{\alpha-1} \\ &= [\gamma\mathbf{D} + (1-\gamma)\mathbf{I}]^{-\alpha} [\gamma\mathbf{D} + (1-\gamma)\mathbf{I} - \gamma\mathbf{L}] [\gamma\mathbf{D} + (1-\gamma)\mathbf{I}]^{\alpha-1} \\ &= [\gamma\mathbf{D} + (1-\gamma)\mathbf{I}]^{-\alpha} [\gamma\mathbf{A} + (1-\gamma)\mathbf{I}] [\gamma\mathbf{D} + (1-\gamma)\mathbf{I}]^{\alpha-1} \end{aligned}$$

It is easy to see that all elements in $\mathbf{P}^{(\alpha, \gamma)}$ are non-negative. Since $\mathbf{A}\mathbf{1} = \mathbf{D}\mathbf{1}$, we have

$$\mathbf{P}^{(1, \gamma)}\mathbf{1} = (\gamma\mathbf{D} + (1-\gamma)\mathbf{I})^{-1} (\gamma\mathbf{A} + (1-\gamma)\mathbf{I})\mathbf{1} = \mathbf{1},$$

completing the proof. \square

B.2 Proof of theorem 3.2

Proof. For any nonzero $\mathbf{x} \in \mathbb{R}^N$, write $\mathbf{y} := [\gamma \mathbf{D} + (1 - \gamma) \mathbf{I}]^{-1/2} \mathbf{x}$. Then we have

$$\begin{aligned} \frac{\mathbf{x}^\top \mathbf{L}^{(1/2, \gamma)} \mathbf{x}}{\mathbf{x}^\top \mathbf{x}} &= \frac{\mathbf{x}^\top \gamma [\gamma \mathbf{D} + (1 - \gamma) \mathbf{I}]^{-1/2} \mathbf{L} [\gamma \mathbf{D} + (1 - \gamma) \mathbf{I}]^{-1/2} \mathbf{x}}{\mathbf{x}^\top \mathbf{x}} \\ &= \frac{\gamma \mathbf{y}^\top \mathbf{L} \mathbf{y}}{\mathbf{y}^\top [\gamma \mathbf{D} + (1 - \gamma) \mathbf{I}] \mathbf{y}} = \frac{\frac{\gamma}{2} \sum_{ij} a_{ij} (y_i - y_j)^2}{\gamma \sum_{ij} a_{ij} y_i^2 + (1 - \gamma) \sum_i y_i^2} \end{aligned}$$

By the Rayleigh quotient theorem,

$$\lambda^{(0)}(\gamma) = \min_{\mathbf{y} \neq \mathbf{0}} \frac{\frac{\gamma}{2} \sum_{ij} a_{ij} (y_i - y_j)^2}{\gamma \sum_{ij} a_{ij} y_i^2 + (1 - \gamma) \sum_i y_i^2} = 0, \quad (24)$$

where the minimum is reached when \mathbf{y} is a multiple of $\mathbf{1}$ and

$$\begin{aligned} \lambda^{(N-1)}(\gamma) &= \max_{\mathbf{y} \neq \mathbf{0}} \frac{\frac{\gamma}{2} \sum_{ij} a_{ij} (y_i - y_j)^2}{\gamma \sum_{ij} a_{ij} y_i^2 + (1 - \gamma) \sum_i y_i^2} \\ &\leq \max_{\mathbf{y} \neq \mathbf{0}} \frac{\sum_{ij} a_{ij} (y_i^2 + y_j^2)}{\sum_{ij} a_{ij} y_i^2} \leq 2, \end{aligned}$$

leading to Eq. (11). The proof of showing $\lambda^{(1)}(\gamma) \neq 0$ if and only if \mathcal{G} is connected is similar to [31], thus we omit the details here.

By the Courant-Fischer min-max theorem, for $\gamma \neq 0$,

$$\begin{aligned} \lambda^{(i)}(\gamma) &= \min_{\{S: \dim(S)=i+1\}} \max_{\{\mathbf{x}: \mathbf{0} \neq \mathbf{x} \in S\}} \frac{\mathbf{x}^\top \mathbf{L}^{(1/2, \gamma)} \mathbf{x}}{\mathbf{x}^\top \mathbf{x}} \\ &= \min_{\{S: \dim(S)=i+1\}} \max_{\{\mathbf{y}: \mathbf{0} \neq \mathbf{y} \in S\}} \frac{\mathbf{y}^\top \mathbf{L} \mathbf{y}}{\mathbf{y}^\top [\mathbf{D} - \mathbf{I} + (1/\gamma) \mathbf{I}] \mathbf{y}}. \end{aligned}$$

It is obvious that the Rayleigh quotient $\frac{\mathbf{y}^\top \mathbf{L} \mathbf{y}}{\mathbf{y}^\top [\mathbf{D} - \mathbf{I} + (1/\gamma) \mathbf{I}] \mathbf{y}}$ is strictly increasing with respect to $\gamma \in (0, 1]$ if $\mathbf{L} \mathbf{y} \neq \mathbf{0}$, i.e., \mathbf{y} not a multiple of $\mathbf{1}$. Note that $\lambda^{(0)}(\gamma) = 0$ and it is reached when $\mathbf{L} \mathbf{y} = \mathbf{0}$, or equivalently \mathbf{y} is a multiple of $\mathbf{1}$. Thus, $\lambda^{(i)}(\gamma)$ is strictly increasing with respect to γ for $i = 1 : N - 1$.

From the eigendecomposition of the symmetric $\mathbf{L}^{(1/2, \gamma)}$ in (10), we can find the eigendecomposition of $\mathbf{L}^{(\alpha, \gamma)}$ as follows:

$$\begin{aligned} \mathbf{L}^{(\alpha, \gamma)} &= [\gamma \mathbf{D} + (1 - \gamma) \mathbf{I}]^{1/2 - \alpha} \mathbf{L}^{(1/2, \gamma)} [\gamma \mathbf{D} + (1 - \gamma) \mathbf{I}]^{\alpha - 1/2} \\ &= [\gamma \mathbf{D} + (1 - \gamma) \mathbf{I}]^{1/2 - \alpha} (\mathbf{U} \mathbf{\Lambda}^{(\gamma)} \mathbf{U}^\top) [\gamma \mathbf{D} + (1 - \gamma) \mathbf{I}]^{\alpha - 1/2} \\ &= \left([\gamma \mathbf{D} + (1 - \gamma) \mathbf{I}]^{1/2 - \alpha} \mathbf{U} \right) \mathbf{\Lambda}^{(\gamma)} \left([\gamma \mathbf{D} + (1 - \gamma) \mathbf{I}]^{1/2 - \alpha} \mathbf{U} \right)^{-1} \end{aligned}$$

Thus, $\lambda^{(i)}(\gamma)$ is also an eigenvalue of $\mathbf{L}^{(\alpha, \gamma)}$ for $i = 0 : N - 1$, and the i -th column of $[\gamma \mathbf{D} + (1 - \gamma) \mathbf{I}]^{1/2 - \alpha} \mathbf{U}$ is a corresponding eigenvector. \square

B.3 Proof of Theorem 3.3

Proof. The proof is similar to the proof of [8]. By [32], the diffusion distance at time t between node v_i and v_j can be expressed as:

$$d_t(v_i, v_j) = \left(\sum_{k=1}^{n-1} e^{-2t\lambda^{(k)}(\gamma)} (\phi_i^{(k)}(\gamma) - \phi_j^{(k)}(\gamma))^2 \right)^{\frac{1}{2}}, \quad (25)$$

where $\lambda^{(1)}(\gamma) \leq \lambda^{(2)}(\gamma) \leq \dots \leq \lambda^{(n-1)}(\gamma)$ are eigenvalues of $\mathbf{L}^{(1, \gamma)}$, and $\{\phi^{(1)}(\gamma), \phi^{(2)}(\gamma), \dots, \phi^{(n-1)}(\gamma)\}$ are the corresponding eigenvectors. We omit the zero $\lambda^{(0)}(\gamma)$.

The inequality $d_t(v_m, v_j) < d_t(v_i, v_j)$ is then equivalent as

$$\begin{aligned} & \left(\sum_{k=1}^{n-1} e^{-2t\lambda^{(k)}(\gamma)} (\phi_m^{(k)}(\gamma) - \phi_j^{(k)}(\gamma))^2 \right)^{\frac{1}{2}} \\ & < \left(\sum_{k=1}^{n-1} e^{-2t\lambda^{(k)}(\gamma)} (\phi_i^{(k)}(\gamma) - \phi_j^{(k)}(\gamma))^2 \right)^{\frac{1}{2}}. \end{aligned} \quad (26)$$

We can take out $\lambda^{(1)}(\gamma)$ and $\phi^{(1)}(\gamma)$ and rearrange the above inequality as:

$$\begin{aligned} & \sum_{k=2}^{n-1} e^{-2t\lambda^{(k)}(\gamma)} \left((\phi_m^{(k)}(\gamma) - \phi_j^{(k)}(\gamma))^2 - (\phi_i^{(k)}(\gamma) - \phi_j^{(k)}(\gamma))^2 \right) \\ & < e^{-2t\lambda^{(1)}(\gamma)} \left((\phi_i^{(1)}(\gamma) - \phi_j^{(1)}(\gamma))^2 - (\phi_m^{(1)}(\gamma) - \phi_j^{(1)}(\gamma))^2 \right). \end{aligned} \quad (27)$$

The left-hand side of Eq. (27) has an upper bound:

$$\begin{aligned} & \sum_{k=2}^{n-1} e^{-2t\lambda^{(k)}(\gamma)} \left| (\phi_m^{(k)}(\gamma) - \phi_j^{(k)}(\gamma))^2 - (\phi_i^{(k)}(\gamma) - \phi_j^{(k)}(\gamma))^2 \right| \\ & \leq e^{-2t\lambda^{(2)}(\gamma)} \sum_{k=2}^{n-1} \left| (\phi_m^{(k)}(\gamma) - \phi_j^{(k)}(\gamma))^2 - (\phi_i^{(k)}(\gamma) - \phi_j^{(k)}(\gamma))^2 \right|. \end{aligned} \quad (28)$$

Then Eq. (27) holds if:

$$\begin{aligned} & e^{-2t\lambda^{(2)}(\gamma)} \sum_{k=2}^{n-1} \left| (\phi_m^{(k)}(\gamma) - \phi_j^{(k)}(\gamma))^2 - (\phi_i^{(k)}(\gamma) - \phi_j^{(k)}(\gamma))^2 \right| \\ & \leq e^{-2t\lambda^{(1)}(\gamma)} \left((\phi_i^{(1)}(\gamma) - \phi_j^{(1)}(\gamma))^2 - (\phi_m^{(1)}(\gamma) - \phi_j^{(1)}(\gamma))^2 \right), \end{aligned} \quad (29)$$

which is equivalent to

$$\begin{aligned} & \log \left(\frac{(\phi_i^{(1)}(\gamma) - \phi_j^{(1)}(\gamma))^2 - (\phi_m^{(1)}(\gamma) - \phi_j^{(1)}(\gamma))^2}{\sum_{k=2}^{n-1} \left| (\phi_m^{(k)}(\gamma) - \phi_j^{(k)}(\gamma))^2 - (\phi_i^{(k)}(\gamma) - \phi_j^{(k)}(\gamma))^2 \right|} \right) \\ & \times \frac{1}{2(\lambda^{(1)}(\gamma) - \lambda^{(2)}(\gamma))} < t. \end{aligned} \quad (30)$$

Let the constant C be the left-hand side of Eq. (30), then if we take $t \geq \lfloor C \rfloor + 1$, we have $d_t(v_m, v_j) < d_t(v_i, v_j)$. Note that C exists if

$$\frac{(\phi_i^{(1)}(\gamma) - \phi_j^{(1)}(\gamma))^2 - (\phi_m^{(1)}(\gamma) - \phi_j^{(1)}(\gamma))^2}{\sum_{k=2}^{n-1} \left| (\phi_m^{(k)}(\gamma) - \phi_j^{(k)}(\gamma))^2 - (\phi_i^{(k)}(\gamma) - \phi_j^{(k)}(\gamma))^2 \right|} > 0, \quad (31)$$

which is satisfied since we assume $|\phi_i^{(1)}(\gamma) - \phi_j^{(1)}(\gamma)| > |\phi_m^{(1)}(\gamma) - \phi_j^{(1)}(\gamma)|$. The original theorem [8] is only based on $\phi^{(1)}$ and does not assume that v_m must satisfy $|\phi_i^{(1)} - \phi_j^{(1)}| > |\phi_m^{(1)} - \phi_j^{(1)}|$, which is necessary for the existence of C . In addition, the original theorem [8] assumes that v_m is obtained by taking a gradient step from v_i , i.e., $\phi_m - \phi_i = \max_{j: v_j \in \mathcal{N}(v_i)} (\phi_j - \phi_i)$, while this property is not needed for the proof. Therefore, Theorem 3.3 both extends and overcomes the shortcomings of [8]. \square

C Computational Complexity Analysis

The complexity of obtaining the first non-trivial eigenvector of the Laplacian is $O(|E|)$, aligning with previous research [8, 28]. Here, we further explain the algorithm, where full eigendecomposition is not required since we only need the first non-trivial eigenvector.

Suppose the symmetrically normalized Laplacian \mathbf{L}_s has eigen-pairs $\{(\mathbf{u}_i, \lambda_i)\}_{i=1:N}$ with $0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_N \leq 2$. The eigenvector corresponding to the smallest eigenvalue 0 is $\mathbf{u}_1 = \mathbf{D}^{\frac{1}{2}}\mathbf{e}$, where \mathbf{D} is the degree matrix and \mathbf{e} is a all-ones vector. To compute the first non-trivial eigenvector \mathbf{u}_2 of \mathbf{L}_s , we define $\tilde{\mathbf{L}}_s = 2\mathbf{I} - \mathbf{L}_s$ and denote the eigenvalues of $\tilde{\mathbf{L}}_s$ as $\bar{\lambda}_i = 2 - \lambda_i$. We know that $\tilde{\mathbf{L}}_s$ and \mathbf{L}_s shares eigenvectors, and $\bar{\lambda}_1 > \bar{\lambda}_2 \geq \dots \geq \bar{\lambda}_N$. Thus, we want to get the eigenvector associated with $\bar{\lambda}_2$.

Next, we define the deflated matrix $\tilde{\tilde{\mathbf{L}}}_s = \tilde{\mathbf{L}}_s - 2\mathbf{u}_1\mathbf{u}_1^\top / \|\mathbf{u}_1\|_2^2$. The largest eigenvalue of $\tilde{\tilde{\mathbf{L}}}_s$ is $\bar{\lambda}_2$, and the corresponding eigenvector is \mathbf{u}_2 . This eigenvector can be obtained using power method on $\tilde{\tilde{\mathbf{L}}}_s$. Note that during power iteration, we compute the multiplication of $\tilde{\tilde{\mathbf{L}}}_s$ with a vector \mathbf{b} , where we do not need to form the $\tilde{\tilde{\mathbf{L}}}_s$ explicitly and the complexity is $O(|E|)$ with a reasonable precision of stopping criterion: (1) $\tilde{\mathbf{L}}_s\mathbf{b}$ would take $O(|E|)$, considering the sparsity of the graph; (2) The multiplication of $2\mathbf{u}_1\mathbf{u}_1^\top / \|\mathbf{u}_1\|_2^2$ with \mathbf{b} takes $O(N)$ as we compute $\mathbf{u}_1^\top\mathbf{b}$ first. Thus, the overall complexity of computing \mathbf{u}_2 is in $O(|E|)$ given that $N < |E|$.

D Datasets

D.1 Real-world Datasets

The overall statistics of the real-world datasets are presented in Table D.1 and Table D.1 provides their heterophily levels calculated using various homophily metrics.

	#Nodes	#Edges	#Features	#Classes	Metric
cora	2,708	5,278	1,433	7	ACC
citeseer	3,327	4,552	3,703	6	ACC
pubmed	19,717	44,324	500	3	ACC
roman-empire	22,662	32,927	300	18	ACC
amazon-ratings	24,492	93,050	300	5	ACC
minesweeper	10,000	39,402	7	2	ROC AUC
tolokers	11,758	519,000	10	2	ROC AUC
questions	48,921	153,540	301	2	ROC AUC
squirrel-filtered	2,223	46,998	2,089	5	ACC
chameleon-filtered	890	8,854	2,325	5	ACC

Table 3: Statistics of the benchmark dataset. Following pre-processing, the graph has been transformed into an undirected and simple form, without self-loops or multiple edges.

	H_{node}	H_{edge}	H_{class}	$H_{\text{agg}}^M(\mathcal{G})$	H_{edge}^*	LI
cora	0.83	0.81	0.77	0.99	0.77	0.59
citeseer	0.71	0.74	0.63	0.97	0.67	0.45
pubmed	0.79	0.80	0.66	0.94	0.69	0.41
roman-empire	0.05	0.05	0.02	1.00	-0.05	0.11
amazon-ratings	0.38	0.38	0.13	0.60	0.14	0.04
minesweeper	0.68	0.68	0.01	0.61	0.01	0.00
tolokers	0.63	0.59	0.18	0.00	0.09	0.01
questions	0.90	0.84	0.08	0.00	0.02	0.00
squirrel-filtered	0.19	0.21	0.04	0.00	0.01	0.00
chameleon-filtered	0.24	0.24	0.04	0.25	0.03	0.01

Table 4: Heterophily levels of benchmark datasets. The $H_{\text{agg}}^M(\mathcal{G})$ stands for the aggregation homophily, calculated using $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$. The H_{edge}^* stands for the adjusted edge homophily, and the LI stands for the label informativeness. The definitions of these metrics can be found in Appendix A.

D.2 Synthetic Datasets

In addition to the real-world datasets, we also tested parameterized diffusion on synthetic graphs generated with different homophily levels ranging from 0 to 1 using the method proposed in [20]. Here we give a review of the generation process.

More specifically, when generating the output graph \mathcal{G} with a desired total number of nodes N , a total of C classes, and a homophily coefficient μ , the process begins by dividing the N nodes into C equal-sized classes. Then the synthetic graph \mathcal{G} (initially empty) is updated iteratively. At each step, a new node v_i is added, and its class z_i is randomly assigned from the set $\{1, \dots, C\}$. Whenever a new node v_i is added to the graph, we establish a connection between it and an existing node v_j in \mathcal{G} based on the probability p_{ij} determined by the following rules:

$$p_{ij} = \begin{cases} d_j \times \mu, & \text{if } z_i = z_j \\ d_j \times (1 - \mu) \times w_{d(z_i, z_j)}, & \text{otherwise} \end{cases} \quad (32)$$

where z_i and z_j are class labels of node i and j respectively, and $w_{d(z_i, z_j)}$ denotes the ‘‘cost’’ of connecting nodes from two distinct classes with a class distance of $d(z_i, z_j)$. For a larger μ , the chance of connecting with a node with the same label increases. The distance between two classes simply implies the shortest distance between the two classes on a circle where classes are numbered from 1 to C . For instance, if $C = 6$, $z_i = 1$ and $z_j = 5$, then the distance between z_i and z_j is 2. The weight exponentially decreases as the distance increases and is normalized such that $\sum_d w_d = 1$. In addition, the probability p_{ij} defined in Eq. (32) is also normalized over the existing nodes:

$$\bar{p}_{ij} = \frac{p_{ij}}{\sum_{k: v_k \in \mathcal{N}(v_i)} p_{ik}}$$

Lastly, the features of each node in the output graph are sampled from overlapping 2D Gaussian distributions. Each class has its own distribution defined separately.

E Hyperparameters

The following table lists the optimal γ and α used in models with the proposed methods.

		roman-empire	amazon-ratings	minesweeper	tolokers	questions	squirrel-filtered	chameleon-filtered
GCN (R(G))	γ	0.1	0.3	0.1	0.9	0.1	0.1	0.1
	α	0.6	0.3	0.3	0.9	0.2	0	0
GAT (R(G))	γ	0.8	0.2	0.2	0.5	0.5	0.2	0.1
	α	0.7	0	0.3	0.9	1	0.9	1
PD-GCN	γ	1	0.9	1	0.6	0.7	1	0.9
	α	0	0.9	1	0.4	0.8	0.1	0
PD-GAT	γ	0	0.9	0.4	1	0.1	0.4	0.6
	α	0	0.9	0.2	1	0.4	0.3	0.2
PD-GAT (R(G))	γ	0.5	0.4	0.3	1	0.5	0.7	0.8
	α	0.9	1	0	0.7	1	0.4	0.5
PD-GAT-sep	γ	0.9	0.1	0.9	0.5	0.2	0.5	0.1
	α	0.4	0.9	0	1	0.2	0.3	0.9
PD-GAT-sep (R(G))	γ	0.6	0.6	0.3	0.5	0.3	0.5	0.2
	α	0.8	0.2	0	0.9	0.4	0.9	0.9

Table 5: Optimal γ and α for real-world datasets.

For BernNet, LINKX, APPNP and ω GCN in real-world benchmark, we perform a grid search for learning rate $\in \{0.01, 0.05, 0.1\}$, weight decay $\in \{0, 5e - 7, 5e - 6, 1e - 5, 5e - 5, 1e - 4, 5e - 4, 1e - 3, 5e - 3, 1e - 2\}$, dropout $\in \{0, 0.1, 0.3, 0.5, 0.7\}$. Model specific parameters are: (1) BernNet: the propagation steps $K = 10$; (2) LINKX: the number of layers of MLP_A and MLP_X in $\{1, 2\}$; (3) APPNP: $\alpha \in \{0.1, 0.2, 0.5, 0.9\}$ and up to 10^{th} power of the adjacency is used; (4) PGSO: initialization $\in \{ \text{‘‘GCN’’}, \text{‘‘all zeros’’}, \text{‘‘SymLaplacian’’}, \text{‘‘RWLaplacian’’}, \text{‘‘Adjacency’’} \}$. According to [12], the weight decay is $5e - 4$, the learning rate is 0.005 for the exponential parameters and 0.01 for all other model parameters. We perform a grid search for dropout $\in \{0, 0.1, 0.3, 0.5, 0.7\}$; (5) DGN: We compute the first two non-trivial eigenvectors of \mathbf{L}_{rw} for DGN. We perform a grid search for the hyperparameters of DGN according to [8] for the learning rate in $\{10^{-5}, 10^{-4}\}$, the weight decay in $\{10^{-6}, 10^{-5}\}$, the dropout rate in $\{0.3, 0.5\}$, the aggregator in $\{ \text{‘‘mean-dir1-av’’}, \text{‘‘mean-dir1-dx’’}, \text{‘‘mean-dir1-av-dir1-dx’’} \}$, the net type in $\{ \text{‘‘complex’’}, \text{‘‘simple’’} \}$.

F Comparison with Message Passing with Virtual Nodes

This section further demonstrates the effectiveness of the parameterized diffusion with rewiring by comparing it to message passing with virtual nodes [51, 66], and the results are summarized in Table. The virtual node is an additional node attached to the original graph and connected to all nodes in the graph. While our rewiring strategy connects all nodes to the gradient node determined by $\mathbf{L}^{(\alpha, \gamma)}$. The difference between this and the virtual node approach is threefold: (1) the gradient node is selected from the original graph’s nodes, while the virtual node an artificially added node; (2) the selection of the gradient node is determined by α and γ ; and (3) the virtual node is incompatible with the proposed parameterized diffusion, as we cannot define the weights or features for edges connecting the virtual nodes based the first non-trivial eigenvector of the $\mathbf{L}^{(\alpha, \gamma)}$ from the original graph topology.

We apply the virtual nodes method to the baseline GNNs, with results provided in Table 6. The same experiment settings from Section 4.3 are used, and results for baselines and the proposed rewiring strategy are also reported in Table 2. In conclusion, the proposed rewiring approach yields a greater performance improvement over the baselines compared to the virtual node method.

	roman-empire	amazon-ratings	minesweeper	tolokers	questions	squirrel-filtered	chameleon-filtered
GCN	73.69 \pm 0.74	48.70 \pm 0.63	89.75 \pm 0.52	83.64 \pm 0.67	76.09 \pm 1.27	39.47 \pm 1.47	40.89 \pm 4.12
GAT	80.87 \pm 0.30	49.09 \pm 0.63	92.01 \pm 0.68	83.70 \pm 0.47	77.43 \pm 1.20	35.62 \pm 2.06	39.21 \pm 3.08
GAT-sep	88.75 \pm 0.41	52.70 \pm 0.62	93.91 \pm 0.35	83.78 \pm 0.43	76.79 \pm 0.71	35.46 \pm 3.10	39.26 \pm 2.50
GCN-Virtual Nodes	74.47 \pm 0.67	48.18 \pm 0.64	89.92 \pm 0.59	83.43 \pm 0.92	77.26 \pm 1.14	40.71 \pm 2.67	43.94 \pm 4.15
GAT-Virtual Nodes	79.35 \pm 0.26	46.85 \pm 0.56	92.62 \pm 0.77	84.17 \pm 0.70	78.47 \pm 0.81	39.07 \pm 1.81	43.47 \pm 3.59
GAT-sep-Virtual Nodes	88.07 \pm 0.61	48.92 \pm 0.50	93.79 \pm 0.46	84.11 \pm 0.50	78.00 \pm 0.96	37.76 \pm 1.57	40.96 \pm 3.54
PD-GAT ($\mathcal{R}(\mathcal{G})$)	87.27 \pm 0.64	48.03 \pm 0.58	93.27 \pm 0.56	84.74 \pm 0.59	79.55 \pm 0.81	42.09 \pm 2.65	44.16 \pm 4.20
PD-GAT-sep ($\mathcal{R}(\mathcal{G})$)	89.23 \pm 0.56	50.96 \pm 0.43	94.03 \pm 0.45	84.83 \pm 0.40	78.88 \pm 0.94	39.69 \pm 2.28	41.15 \pm 4.66

Table 6: Experiment results on heterophily datasets proposed by [50] for comparing baseline models using virtual nodes methods [51, 66] and the proposed parameterized diffusion with topology-guided rewiring.

G Comparison on Homophilic Datasets

The following table summarizes the results on homophilic datasets, where we perform a grid search for learning rate $\in \{0.01, 0.05, 0.1\}$, weight decay $\in \{0, 5e-7, 5e-6, 1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2\}$, dropout $\in \{0, 0.1, 0.3, 0.5, 0.7\}$, and 64 hidden states. Note that the implementation of baselines follows [50], where residual connections are adopted in each layer.

	Cora	Citeseer	Pubmed
ResNet	72.03 \pm 0.24	70.77 \pm 1.81	88.01 \pm 0.41
GCN	86.15 \pm 1.24	74.58 \pm 0.97	89.63 \pm 0.44
SAGE	85.12 \pm 1.64	74.47 \pm 1.93	89.69 \pm 0.51
GAT	86.46 \pm 1.02	74.13 \pm 1.85	88.87 \pm 0.65
GAT-sep	84.24 \pm 1.72	73.93 \pm 1.93	89.14 \pm 0.57
GT	86.19 \pm 0.99	74.23 \pm 1.12	89.48 \pm 0.52
GT-sep	86.3 \pm 1.13	74.14 \pm 0.91	89.63 \pm 0.52
DGN	85.16 \pm 1.17	72.70 \pm 1.17	87.35 \pm 0.53
ω GCN	86.13 \pm 1.38	74.74 \pm 1.39	88.65 \pm 0.42
PGSO	88.66 \pm 0.94	76.55 \pm 1.12	89.44 \pm 0.53
PD-GCN	86.91 \pm 1.45	75.17 \pm 1.24	89.70 \pm 0.45
PD-GAT	85.40 \pm 1.41	74.83 \pm 1.75	89.48 \pm 0.45

Table 7: Experiments on homophily datasets proposed in [67].

H Training

In training and evaluating a model using a node classification benchmark dataset with C distinct classes, each node $v_i \in \mathcal{V}$ has a label z_i associated with it. We denote $\mathbf{Z} \in \mathbb{R}^{N \times C}$ as the one-

hot encoding of labels. Moreover, nodes are divided into three sets: the training set $\mathcal{V}_{\text{train}}$, the validation set \mathcal{V}_{val} and the test set $\mathcal{V}_{\text{test}}$. In the training phase, the model uses features of all nodes under transductive learning. The model only has access to labels of nodes in $\mathcal{V}_{\text{train}}$ and in \mathcal{V}_{val} (for hyperparameter tuning), while labels of nodes in $\mathcal{V}_{\text{test}} = \mathcal{V} \setminus (\mathcal{V}_{\text{train}} \cup \mathcal{V}_{\text{val}})$ remain unknown to the model. The cost function used in node classification tasks is the standard categorical cross-entropy loss [26], which is commonly used for multi-class classification tasks:

$$\mathcal{L} = -\frac{1}{|\mathcal{V}_{\text{train}}|} \text{trace}(\mathbf{Z}^T \log \mathbf{Y}), \quad (33)$$

where \mathbf{Y} is the output from the model after softmax and $\log(\cdot)$ is applied element-wise.