Selective Blocking for Message-Passing Neural Networks on Heterophilic Graphs

Yoonhyuk Choi[1] **Taewook Ko**[2] **Jiho Choi**[3] **Chong-Kwon Kim**[4]

[1]Samsung SDS, Seoul, South Korea
[2]Samsung Electronics, Suwon, South Korea,
[3]KAIST, Seoul, South Korea,
[4]Korea Institute of Energy Technology, Naju, South Korea,

Abstract

Graph Neural Networks (GNNs) thrive on message passing (MP) but are vulnerable when the graph carries many heterophilic or misclassified edges. Prior analyses suggest that signed propagation can mitigate over-smoothing under low edge–error rates, yet they implicitly assume perfect edge labels and the presence of self-loops. We revisit this setting and show that, under high edge uncertainty, propagating any information may harm node separability even with signed weights. Our key insight is to decide not to propagate along uncertain edges adaptively. Concretely, we intentionally omit self-loops to isolate pure neighbor influence for a clearer theoretical analysis, adopt a row-stochastic (asymmetric) operator that matches the Markov–chain view of MP and simplifies spectral-radius proofs. Then, we dynamically estimate the local homophily b_i and edge-classification error e_t during training via an EM procedure. We prove that our selective blocking yields a sub-stochastic propagation matrix whose joint spectral radius exceeds that of signed GNNs under high e_t, guaranteeing reduced over-smoothing, and we supply a lemma showing that class-discriminative signals survive even when the operator is rank-deficient. Extensive experiments on six homophilic and heterophilic benchmarks confirm that the proposed adaptive blocking outperforms strong baselines.

1 INTRODUCTION

Graph Neural Networks (GNNs) have shown remarkable performance with the aid of a message-passing (MP), where the representation of each node is recursively updated using its neighboring nodes based on structural properties Defferrard et al. [2016], Kipf and Welling [2016], Velickovic et al. [2017]. Early GNNs rely on network homophily, assuming that connected nodes will likely share similar labels. However, many real-world graphs have low homophily (a.k.a. heterophilic) Zhu et al. [2020], Ma et al. [2021], Luan et al. [2022], where spectral GNNs Wang and Zhang [2022] achieve dismal performance under this condition since Laplacian smoothing only receives low-frequency signals from neighboring nodes Ko et al. [2023a,b].

Recent algorithms employ high-pass filters by adjusting edge weights during MP Velickovic et al. [2017], Brody et al. [2021], Chen et al. [2023], Liao et al. [2024] to separate the embeddings of connected but dissimilar nodes. Notably, flipping the sign of edges from positive to negative, known as signed propagation Chien et al. [2020], Bo et al. [2021], or blocking MP by assigning zero weights to heterophilic connections Luo et al. [2021], Hu et al. [2021], Tian et al. [2022] has recently achieved remarkable performance. These techniques represent significant advancements in the field, offering new ways to enhance the performance and applicability of GNNs in diverse graph structures.

Recently, efforts have been made to theoretically analyze the effect of various propagation schemes in terms of uncertainty Choi and Kim [2025a] and node separability Ma et al. [2021], Yan et al. [2021], Aseem et al. [2023], building upon the insights of the Contextual Stochastic Block Model (CSBM) Deshpande et al. [2018]. In detail, they measure the distance of node feature expectations from a decision boundary, demonstrating that signed propagation outperforms vanilla message-passing algorithms in binary classification tasks. More recently, Choi et al. [2023], Li et al. [2024b] provided new insights by extending the analysis to multiple classes and considering degree distributions. However, few efforts have been made to determine whether signed MP consistently improves separability compared to not using propagation.

In this paper, we focus on the work of Luo et al. [2021], which mitigates local smoothing by assigning zero weights

(blocking information) to heterophilic edges. Building on prior theoretical analysis, we propose the estimation of two latent parameters during training: homophily and edge classification error ratios. Based on this, we argue that MP (even with signed propagation) may result in poor performance compared to not propagating under specific conditions. Therefore, we suggest adaptively blocking information based on these values. Finally, since these values may not be easily measurable during the training phase, we propose an estimation strategy using the validation score. In summary, our contributions are as follows:

- Unlike previous methods that focused solely on signed propagation to prevent over-smoothing, we demonstrate that message passing may degrade the separability of graph neural networks under certain conditions.
- To mitigate the smoothing effect, we propose an adaptive propagation approach based on the estimated parameters. We show that blocking information can be more efficient depending on the homophily and edge classification error ratios.
- Extensive experiments on real-world benchmark datasets with state-of-the-art baselines show notable performance improvements, validating the proposed scheme and theoretical analysis.

2 RELATED WORK

Heterophilic GNNs. Starting from Laplacian decomposition Defferrard et al. [2016], spectral GNNs Kipf and Welling [2016], Wang and Zhang [2022], Du et al. [2022], Bo et al. [2023], Lu et al. [2024], Choi and Kim [2025b] have achieved remarkable performance on homophilic graphs. However, as the homophily Platonov et al. [2024] of the graph decreases, their performance sharply declines due to local smoothing Pei et al. [2020]. To address this limitation, spatial-based GNNs have emerged, developing many powerful schemes that adjust edge weights for message-passing Velickovic et al. [2017], Brody et al. [2021], Chen et al. [2023], Liao et al. [2024]. Specifically, some studies handle disassortative edges by capturing node differences or incorporating similar remote nodes as neighbors Derr et al. [2018], Huang et al. [2019], Zhu et al. [2020], Choi et al. [2022], Lei et al. [2022], Wang et al. [2022], Zhao et al. [2023], Zheng et al. [2023], Mao et al. [2024], Yan et al. [2024], Li et al. [2024a], Qiu et al. [2024], Tang et al. [2024]. Among these, methods that either change the sign of the edge Chien et al. [2020], Bo et al. [2021], Fang et al. [2022], Guo and Wei [2022] or opt not to transmit information Luo et al. [2021], Hu et al. [2021], Tian et al. [2022] have recently been proposed.

Over-smoothing in GNNs. In addition to the above methods, theoretical analyses have emerged explaining why each message-passing technique works well from the perspective of node separability (reduced smoothing effect). For example, Yan et al. [2021], Baranwal et al. [2021] analyzed the separability of the vanilla, signed, and blocked propagation after message-passing under binary class graphs. Recently, Choi et al. [2023] extended the theorems to multi-class scenarios, and Li et al. [2024b] suggested their degree-corrected version. However, these methods assume a fixed edge classification error, which may fail to induce the smoothing effect under precise condition.

3 PRELIMINARIES

Notations. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X)$ be a graph with $|\mathcal{V}| = n$ nodes and $|\mathcal{E}| = m$ edges. The node attribute matrix is $X \in \mathbb{R}^{n \times F}$, where F is the dimension of an input vector. Given X, the hidden representation of node features $H^{(l)}$ at the *l-th* layer is derived through message-passing. Here, node $i's$ feature is defined as h_i^l. The structural property of $\mathcal{G}$ is represented by its adjacency matrix $A \in \{0,1\}^{n \times n}$. A diagonal matrix with node degrees D is derived from A as $d_{ii} = \sum_{j=1}^{n} A_{ij}$. Each node has its label $Y \in \mathbb{R}^{n \times C}$ (C represents the number of classes).

Message-Passing (MP). Generally, GNNs employ alternate steps of propagation and aggregation recursively, during which the node features are updated iteratively. The widely known MP algorithm is GCN Kipf and Welling [2016], which can be represented as:

$$H^{(l+1)} = \sigma(\tilde{A}^l H^{(l)} W^{(l)}) \tag{1}$$

Here, $H^0 = X$ is the initial vector and H^l is nodes' hidden representations at the *l-th* layer. H^{l+1} is retrieved through message-passing ($\tilde{A} = D^{-1}A$) with an activation function σ. W^l is trainable weight matrices. The final prediction is produced by applying cross-entropy $\Phi(\cdot)$ (e.g., log-softmax) to $H^{(l)}$ and the loss function is defined as below:

$$\mathcal{L}_{GNN} = \mathcal{L}_{nll}(Y, \widehat{Y}). \tag{2}$$

The loss is computed through a negative log-likelihood $\mathcal{L}_{nll}$ between true labels (Y) and predictions, $\widehat{Y} = \Phi(H^{(L)})$.

Homophily. $\mathcal{H}_g$ stands for the global edge homophily ratio, which is defined as:

$$\mathcal{H}_g \equiv \frac{\sum_{(i,j)\in\mathcal{E}} 1(Y_i = Y_j)}{|\mathcal{E}|} \tag{3}$$

Likewise, the local homophily ratio, b_i, of node i is given as:

$$b_i \equiv \frac{\sum_{j=1}^{n} A_{ij} \cdot 1(Y_i = Y_j)}{d_{ii}} \tag{4}$$

Given a partially labeled training set $\mathcal{V}_L$, the goal of semi-supervised node classification is to correctly predict the classes of unlabeled nodes $\mathcal{V}_U = \{\mathcal{V} - \mathcal{V}_L\} \subset \mathcal{V}$.

Over-smoothing. As introduced in Oono and Suzuki [2019], Cai and Wang [2020], over-smoothing measures the distinguishability of node features as follows:

$$\mu(H^{(l)}) := ||H^{(l)} - 1\frac{1^T H^{(l)}}{N}||_F \tag{5}$$

The over-smoothing happens if $\lim_{l\to\infty} \mu(H^{(l)}) = 0$, where the node representation converges to zero after infinite propagation. Recently, Mao et al. [2024] proved that even the attention-based GNN Velickovic et al. [2017] loses the separability exponentially as l increases.

4 MOTIVATION

We first define various MP schemes, including vanilla, signed, and blocking (pruning) types. Then, we introduce the prior theorems on local smoothing using the Contextual Stochastic Block Models (CSBMs). Finally, we highlight their drawbacks and present our new insights based on the estimated parameters.

4.1 LOCAL SMOOTHING ON MESSAGE-PASSING SCHEMES

We first define the concept of three MP schemes Baranwal et al. [2021], Yan et al. [2021], Choi et al. [2023] below.

Definition 1 (**M**essage-**P**assing Schemes)**.** *Building upon the Laplacian-based degree normalization Kipf and Welling [2016], a.k.a. GCN, each propagation scheme modifies the adjacency matrix $\tilde{A} = D^{-1}A$ (Eq. 1) as follows:*

- ***Vanilla MP*** *inherits the original matrix, which only consists of positive edges.*

$$\forall(i,j) \in \mathcal{E}, \;\; \tilde{A} = D^{-1}A \geq 0 \tag{6}$$

- ***Signed MP*** *assigns negative values to the heterophilic edges, where $y_i \neq y_j$.*

$$\forall(i,j) \in \mathcal{E}, \;\; \tilde{A} \in \begin{cases} D^{-1}A, & y_i = y_j \\ -D^{-1}A, & y_i \neq y_j \end{cases} \tag{7}$$

- ***Blocked MP*** *blocks the information propagation for heterophilic edges by assigning zero.*

$$\forall(i,j) \in \mathcal{E}, \;\; \tilde{A} \in \begin{cases} D^{-1}A, & y_i = y_j \\ 0, & y_i \neq y_j \end{cases} \tag{8}$$

To analyze the smoothing effect of each MP scheme, we inherit several useful notations defined in Yan et al. [2021] as follows: (1) For all nodes $i = \{1, ..., n\}$, their degrees $\{d_i\}$ and latent features $\{h_i\}$ are *i.i.d.* random variables. (2) Each class has the same population. (3) The scale of each class distribution after initial embedding is identical, $||\mathbb{E}(h_i^{(0)}|y_i)|| = \mu$. Then, the feature distribution after a single hop propagation $\mathbb{E}(h^{(1)}|y_i)$ can be defined through a **C**ontextual **S**tochastic **B**lock **M**odel (CSBM) as follows.

Definition 2 (CSBM under binary class scenario)**.** *Assume a binary class $\mathbb{E}(h_i^{(0)}|y_i) \sim (\mu, \theta = \{0, \pi\})$ and node feature is sampled from Gaussian distribution (N) Deshpande et al. [2018]. If $y_i = 0$, the updated distribution $\mathbb{E}(h^{(1)}|y_i)$ is given by:*

$$\mathbb{E}(h^{(1)}|y_i) \sim N(\mu, \frac{1}{\sqrt{deg(i)}}) \tag{9}$$

Choi et al. [2023] extended binary CSBM (Eq. 9) to multiple (ternary) classes using additional angle ϕ as below:

$$\mathbb{E}(h_i^{(0)}|y_i) = (\mu, \phi, \theta), \tag{10}$$

where $\phi = \pi/2$ and $0 \leq \theta \leq 2\pi$. The above equation satisfies the binary case's origin symmetry as $(\mu, \pi/2, 0) = -(\mu, \pi/2, \pi)$. Now, we define the impact of three MP schemes using a multi-class CSBM below.

Lemma 3 (Multi-class CSBM, Vanilla MP)**.** *Given $y_i = 0$, let us assume ego $k \sim (\mu, \pi/2, 0)$ and aggregated neighbors $k' \sim (\mu, \pi/2, \theta')$. By replacing the $\mathbb{E}_p(h_i^{(1)}|y_i, d_i)$ as $\mathbb{E}_p(\cdot)$, the expectation after vanilla MP is as follows:*

$$\mathbb{E}_p(\cdot) = \frac{\{kb_i + k'(1-b_i)\}d_i' + k}{d_i + 1} \tag{11}$$

The k' always satisfies $||k'|| \leq ||\mu||$ regardless of the normalized degree and homophily ratio since $1 - b_i \leq 1$.

Lemma 4 (Multi-class CSBM, Signed MP)**.** *Similar to above, the expectation $\mathbb{E}_s(\cdot)$ after signed MP is given by:*

$$\mathbb{E}_s(\cdot) = \frac{(1-2e)\{b_i k + (b_i - 1)k'\}d_i' + k}{d_i + 1} \tag{12}$$

Remark. The notation e stands for the error ratio of incorrectly changing the sign of edges, which will be addressed throughout this paper. As introduced in Choi et al. [2023], the sign inconsistency caused by multi-hop propagation can be solved through jumping knowledge Xu et al. [2018].

Lemma 5 (Multi-class CSBM, Blocked MP)**.** *Lastly, the expectation $\mathbb{E}_b(\cdot)$ driven by blocked MP can be defined as:*

$$\mathbb{E}_b(\cdot) = \frac{\{(1-e)b_i k + e(1-b_i)k'\}d_i' + k}{d_i + 1} \tag{13}$$

Proof of Lemma 3 - 5 is in Appendix A.

4.2 MEANING OF BLOCKED MP

Based on the analyses above, we derive new insights into the smoothing effect of each propagation scheme. Following Yan et al. [2021], we assume that node separability

(discriminative power) is proportional to the coefficient of $\mathbb{E}_p(\cdot), \mathbb{E}_s(\cdot), \mathbb{E}_b(\cdot)$ as in the above lemma. Through this assumption, we first show that vanilla MP has lower separability than signed and blocked MPs.

Corollary 6 (Vanilla vs Other MPs)**.** *Since $d_i'/(d_i+1)$ is shared for all MPs (Eq. 11 - 13), we can compare the separability by omitting them as follows:*

- *(Vanilla vs Signed MP) By comparing Eq. 11 and 12:*

$$Z_1 = \mathbb{E}_p(\cdot) - \mathbb{E}_s(\cdot) = (2e-1)\{b_i k + (b_i - 1)k'\} \tag{14}$$

- *(Vanilla vs Blocked MP) By comparing Eq. 11 and 13:*

$$Z_2 = \mathbb{E}_p(\cdot) - \mathbb{E}_b(\cdot) = e b_i k + (1-e)(1-b_i)k' \tag{15}$$

As shown in the above equations, the $\iint_{e,b_i} Z_1$ and $\iint_{e,b_i} Z_2$ are always negative, regardless of the variables b_i and e. This indicates that ***vanilla MP has the lowest discrimination power among the three MPs****. Therefore, we can focus on the separability of the signed and blocked MPs to determine the optimal propagation scheme.*

Corollary 7 (Signed vs Blocked MP)**.** *The difference between signed and blocked MP is given by:*

$$Z_3 = \mathbb{E}_s(\cdot) - \mathbb{E}_b(\cdot) = (1-2e)k + (b_i - e)k' \tag{16}$$

Previous studies assume a perfect edge classification scenario ($e = 0$), leading to the conclusion that $Z_3 = 1 - b_i \geq 0$, where signed MP outperforms blocked MP. However, achieving this optimal condition is challenging in a semi-supervised setting with few training nodes. Therefore, we propose estimating two parameters, e and b_i, to select the most precise MP schemes. To facilitate understanding, we provide an example by varying e below.

Corollary 8 (Numerical example on edge error ratio)**.** *We assume three different edge error ratios, $e = \{1, 0.5, 0\}$, to compare the signed and blocked MPs. Assuming that the neighbors are i.i.d. ($k' = -k$), Eq. 16 simplifies to:*

$$Z_3 = (1-e-b_i)k = \begin{cases} -b_i k, & e = 1 \\ (-b_i + 0.5)k, & e = 0.5 \\ (1-b_i)k, & e = 0 \end{cases} \tag{17}$$

We can infer some useful insights from the above Corollary: (1) Under a high error ratio ($e = 1$, initial stage of training), it might be better to *not propagate* rather than using signed edges since $-b_i k \leq 0$ ($Z_3 \leq 0$). In addition, an error in signed propagation increases the uncertainty more than blocked GNNs, as shown in Lemma 9. (2) If the error is mediocre ($e = 0.5$), the sign of Z_3 is dependent on the homophily ratio b_i. In this condition, signed MP may perform well ($Z_3 \geq 0$) under heterophily ($b < 0.5$), but it is still advantageous not to propagate messages under homophily. (3) However, if the edges are perfectly classified ($e = 0$), signed MP might be the best option ($1 - b_i \geq 0$), as demonstrated in prior work Choi et al. [2023]. Therefore, we suggest the estimation of these two parameters, b_i and e, for precise training. In addition, the following theorem reveals the necessity of blocked MP under high e in terms of uncertainty.

Theorem 9 (Uncertainty)**.** *Under large values of e, signed MP exhibits higher entropy than the blocked ones.*

Proof. Let us assume an ego i with label k, and its neighbor node j is connected to i with a signed edge. Firstly, the true label probability (k) of node b $\widehat{y}_{b,k}$ increases, while other probabilities $\widehat{y}_{b,o}$ ($o \neq k$) decrease as follows:

$$\widehat{y}_p^{(t+1)} \in \begin{cases} \widehat{y}_{b,k}^t - \eta \nabla_b \mathcal{L}_{nll}(Y_i, \widehat{Y}_i)_k > \widehat{y}_{b,k}^t \\ \widehat{y}_{b,o}^t - \eta \nabla_b \mathcal{L}_{nll}(Y_i, \widehat{Y}_i)_o < \widehat{y}_{b,o}^t \quad \forall\, o \neq k. \end{cases} \tag{18}$$

Now, we analyze the partial derivative $\nabla_b \mathcal{L}_{nll}(Y_i, \widehat{Y}_i)_o$ for $\forall\, o \neq k$,

$$\begin{aligned} \nabla_b \mathcal{L}_{nll}(Y_i, \widehat{Y}_i)_o &= \frac{\partial \mathcal{L}_{nll}(Y_i, \widehat{Y}_i)_o}{\partial \widehat{y}_{i,o}} \cdot \frac{\partial \widehat{y}_{i,o}}{\partial h_{b,o}^{(L)}} \\ &= \frac{1}{\widehat{y}_{i,o}} \cdot (\widehat{y}_{i,o}(1 - \widehat{y}_{i,o})) = 1 - \widehat{y}_{i,o} > 0 \end{aligned} \tag{19}$$

On the contrary, the gradient of node s has a different sign from node p, where we can infer that:

$$\widehat{y}_s^{(t+1)} \in \begin{cases} \widehat{y}_{s,k}^t - \eta \nabla_s \mathcal{L}_{nll}(Y_i, \widehat{Y}_i)_k < \widehat{y}_{s,k}^t, \\ \widehat{y}_{s,o}^t - \eta \nabla_s \mathcal{L}_{nll}(Y_i, \widehat{Y}_i)_o > \widehat{y}_{s,o}^t, \ \forall\, o \neq k \end{cases} \tag{20}$$

Based on the above analysis, as the training epoch t proceeds, the expectation of signed MP prediction ($\widehat{y}_s$) exhibits higher entropy $\mathcal{H}(\mathbb{E}[\widehat{y}_s])$ compared to that ($\widehat{y}_b$) of blocked MP $\mathcal{H}(\mathbb{E}[\widehat{y}_b])$ as below:

$$\mathcal{H}(\mathbb{E}[\widehat{y}_s^{(t+1)}]) - \mathcal{H}(\mathbb{E}[\widehat{y}_b^{(t+1)}]) > \mathcal{H}(\mathbb{E}[\widehat{y}_s^t]) - \mathcal{H}(\mathbb{E}[\widehat{y}_b^t]). \tag{21}$$

5 METHODOLOGY

Selecting an appropriate MP scheme is crucial for reducing smoothing and uncertainty. However, this may not be tractable in a semi-supervised learning context. Thus, we employ an EM algorithm, where the **E-step** is used for parameter estimation, and the **M-step** is used for optimization.

5.1 (E-STEP) PARAMETER ESTIMATION

We start with the strategy of homophily (b_i) and edge error (e_t) estimation in Theorem 10 and 11, respectively.

Theorem 10 (Homophily estimation)**.** *The homophily ratio b_i can be inferred using the mechanism of MLP Wang et al. [2022] and EvenNet Lei et al. [2022] as follows:*

$$b_i = B_i B_i^T, \; B := \sigma\left(\sum_{l=0}^{L} XW^l + \sum_{l=0}^{\lfloor L/2 \rfloor} \tilde{A}^{2l} XW^l\right) \tag{22}$$

which can take advantage of both the heterophily robustness of MLP and the low variance of EvenNet.

Proof. The left term $\sum_{l=0}^{L} XW^l$ employs blocked MP (MLP), while the right term only receives messages from even-hop neighbors (EvenNet). This approach reduces the prediction variance caused by homophily changes while maintaining average performance. To begin, let us define the k-step homophily ratio $\mathbb{H}_k$, following Lei et al. [2022], under *i.i.d.* graphs as follows:

$$\mathbb{H}_k = \frac{1}{N} \sum_{l=0}^{K-1} \left(\Pi_{ll}^k - \sum_{m \neq l} \Pi_{lm}^k \right), \tag{23}$$

where $\Pi^k = Y^T \tilde{A}^k Y$. Since all k-step propagations share the first term, the above equation can be rewritten as:

$$\mathbb{H}_k \approx \sum_{i=0}^{k} \theta_i \mathbb{H}_i(\Pi) = \theta_0 + \theta_1 (I - \tilde{L}) + \cdots + \theta_k (I - \tilde{L})^k \tag{24}$$

Combining the notions of MLP and EvenNet, the equation becomes:

$$\mathbb{H}_0 + \mathbb{H}_{2k} \approx \theta_0 + \left(\theta_0 + \theta_2 (I - \tilde{L})^2 + \cdots + \theta_{2k} (I - \tilde{L})^{2k} \right) \tag{25}$$

By dividing the above equation by 2, we can infer that $\mathbb{E}[\mathbb{H}_0 + \mathbb{H}_{2k}] = \mathbb{E}[\mathbb{H}_{2k}]$, but $Var[\mathbb{H}_0 + \mathbb{H}_{2k}] < Var[\mathbb{H}_{2k}]$. This can enhance heterophily robustness (reduced variance) while maintaining overall performance (expectation).

Theorem 11 (Edge error estimation)**.** *Given the node classification accuracy of validation sets (α_{t-1}) at iteration $t-1$ and the number of classes (c), e_t can be inferred as below:*

$$e_t = 1 - \{ a_{t-1}^2 + \frac{(1 - a_{t-1})^2}{c - 1} \} \tag{26}$$

Approximation error. *Given the experimental settings in Kipf and Welling [2016], the number of validation sets satisfies $n_{val} = 1,080 > 30$ for Cora as in Table 1. Thus, it can be easily inferred that the feature distribution of randomly sampled validation sets ($\mathbb{E}[h_{val}]$) follows the total distribution ($\mathbb{E}[h]$) as below.*

$$Z = \frac{\mathbb{E}[h_{val}] - \mathbb{E}[h]}{var(h_{val})/\sqrt{n_{val}}} \sim N(0, 1) \tag{27}$$

Now, we demonstrate that the validation samples are less likely to be biased using the concentration theorem. Given that the output of the GNN (h_i) follows a Dirichlet distribution, Dir($h; c$) $= \frac{1}{B(c_1, \ldots, c_K)} \prod_{k=1}^{K} h_i^{c_k - 1}$, and assuming the classes are i.i.d., we can derive the following inequality:

$$\mathbb{P}(||h_{val} - \mathbb{E}[h_{val}]||_2 \geq \varepsilon) \leq 2 \cdot \exp(-2\varepsilon^2 n_{val}) \tag{28}$$

which means that the validation nodes are equally distributed around its center $\mathbb{E}[h_{val}]$.

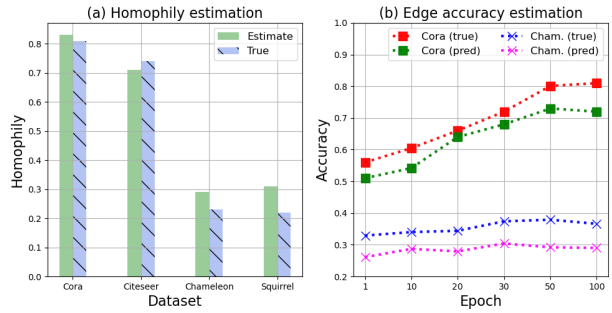

Figure 1: Empirical analysis on (a) homophily and (b) edge error estimation. More details can be found in Appendix B

Proof. We stated that the edge error ratio e_t can be inferred using the node classification accuracy of validation sets (α_{t-1}) at iteration $t-1$ and the number of classes (c). Thus,

$$e_t = 1 - p \text{ (correct edge classification)} \tag{29}$$

$$= 1 - \{ \underbrace{\left(\frac{a_{t-1}}{c}\right)^2 * c^2}_{y_i = y_j} + \underbrace{\left(\frac{1 - a_{t-1}}{c - 1}\right)^2 * (c - 1)}_{y_i \neq y_j} \} \tag{30}$$

$$= 1 - \{ a_{t-1}^2 + \frac{(1 - a_{t-1})^2}{c - 1} \} \tag{31}$$

If two nodes share the same label ($y_i = y_j$), the edge classification accuracy should be proportional to the product of their validation scores. Conversely, if they have different labels, the edge is correctly classified only if both nodes are incorrectly predicted to have the same label.

Empirical analysis. In Figure 1, we conduct empirical analyses to verify Theorem 10 (left) and 11 (right). In the figure (a), we illustrate the estimated and true homophily ratios under semi-supervised settings Kipf and Welling [2016]. In the figure (b), we compare the true and estimated edge errors (Eq. 26) for two graphs, Cora (homophilic) and Chameleon (heterophilic), during the training epochs (x-axis). Both figures show that the predicted values closely match the actual values. The minor differences likely arise because the node distribution in the benchmark graphs does not satisfy the i.i.d. assumption, which supports our theoretical analysis.

Algorithm 1 Pseudo-code of our method

Require: Adjacency matrix ($\tilde{A}$), node features (X), initialized parameters (θ), initialized validation ($\alpha_0 = 0$) and best validation score ($\alpha^* = 0$)

Ensure: Parameters with the best validation score (θ^*)

Homophily estimation b_i (Eq. 22)
for training epochs $t \geq 1$ **do**
  (E-Step)
  Get validation score a_{t-1}
  Edge error estimation e_t (Eq. 26)
  (M-Step)
  For all edges in $\tilde{A}$
  if $\tilde{A}_{ij} < 0 \ \wedge \ Z_t = 1 - b_i - e_t < 0$ **then**
    $\widehat{A}_{ij} = 0$
  else
    $\widehat{A}_{ij} = \tilde{A}_{ij}$
  Message-passing, $H^{(l+1)} = \sigma(\widehat{A}H^{(l)}W^{(l)})$
  Node classification, $\mathcal{L}_{GNN} = \mathcal{L}_{nll}(Y, \sigma(H^{(L)}))$
  Parameter update, $\theta^{(t+1)} = \theta^t - \eta\partial\mathcal{L}_{GNN}/\partial\theta^t$
  Get validation score α_t
  if $\alpha_t > \alpha^*$ **then**
    Save current parameters as best, $\theta^* = \theta^t$
    Update best validation score, $\alpha^* = \alpha_t$

5.2 (M-STEP) OPTIMIZATION WITH CALIBRATION

Referring to the separability of signed and blocked MP in Eq. 17, we can determine the propagation type based on the estimated values of b_i (Eq. 22) and e (Eq. 26). For each training step t, the discrimination gap between signed and blocked MP can be defined below:

$$Z_t = (1 - b_i - e_t)k \tag{32}$$

If $Z_t < 0$, we block signed messages to reduce the smoothing effect (Corollary 7) and uncertainty (Lemma 9). For a numerical definition, let us assume the adjacency matrix derived from the downstream task as $\tilde{A}$ as defined from Eq. 6 to 8. Then, we calibrate the edges based on the following conditions:

$$\widehat{A}_{ij} = \begin{cases} 0, & \tilde{A}_{ij} < 0 \ \wedge \ Z_t < 0 \\ \tilde{A}_{ij}, & \text{otherwise} \end{cases} \tag{33}$$

We replace the normalized adjacency matrix ($\tilde{A}$) in Eq. 1 with $\widehat{A}$ as follows:

$$H^{(l+1)} = \sigma(\widehat{A}H^{(l)}W^{(l)}) \tag{34}$$

Then, we employ Eq. 2 ($\mathcal{L}_{GNN}$) for optimization, where the above strategy can be applied to general signed MPs.

5.3 THEORETICAL JUSTIFICATION

We aim to show that the proposed method can relieve the smoothing effect of signed MP using the notion of spectral radius (Def. 12 to Thm. 16) and separability (Thm. 17).

Definition 12 (Spectral Radius)**.** *Let $\lambda_1, ..., \lambda_n$ be the eigenvalues of an adjacency matrix $A \in \mathcal{R}^{n\times n}$. Then, the spectral radius of A is given by:*

$$\rho(A) = \max\{|\lambda_1|, ..., |\lambda_n|\} \tag{35}$$

A^∞ is well known to converge if $\rho(A) < 1$ or $\rho(A) = 1$, where $\lambda_1 = 1$ is the only eigenvalue on the unit circle.

Definition 13 (Joint Spectral Radius)**.** *The generalization of a spectral radius from a single matrix to a finite set of matrices $\mathcal{M} = \{A_1, ..., A_n\}$:*

$$J_\rho(A) = \lim_{k\to\infty} \max\{||A_1 \cdots A_k||^{1/k} : A \in \mathcal{M}\} \tag{36}$$

where the over-smoothing occurs at an exponential rate if and only if $J_\rho(A) < 1$. In the following lemma, we first demonstrate that even attention-based MP cannot resolve the over-smoothing issue.

Lemma 14 (Attention is susceptible to over-smoothing)**.** *For all layers l, the attention matrix A^l cannot change the connectivity of the original graph Brody et al. [2021] as follows:*

$$0 < A^l_{ij} \leq 1, \ \ \forall(i,j) \in \mathcal{E} \tag{37}$$

It is well known that the infinite products of non-homogeneous row-stochastic matrices $\Pi_{l=1}^{\infty} A^l$ converge to the same vector Cowles and Carlin [1996], Seneta [2006], where we can easily infer that stacking the infinite softmax-based attention matrix also converges (irreducible). Recent work Wu et al. [2024] provided a tighter bound that GAT satisfies $J_\rho(A) < 1$ (Eq. 36), which means that $\lim_{l\to\infty} A^l$ converges to a rank-one matrix, causing over-smoothing.

Unlike an attention-based matrix, where the total sum of the edge weights connected to a neighbor node is 1, a signed adjacency matrix (Eq. 7) is sub-stochastic since the sum of rows is less than 1. Therefore, the spectral radius of the sub-stochastic matrix satisfies the following conditions.

Corollary 15 (Convergence of signed MP)**.** *Let $\tilde{A}$ be the irreducible sub-stochastic matrix obtained by applying signed message passing to the original adjacency matrix A (Eq. 7). Let $\boldsymbol{\pi}$ be the corresponding (right) Perron eigenvector of $\tilde{A}$, normalised such that $\|\boldsymbol{\pi}\|_1 = 1$. Define the row deficiency $\epsilon_i := 1 - \sum_{j=1}^{N} \tilde{A}_{ij}$ (> 0). Then, from Eq. 35,*

$$\rho(\tilde{A}) = \sum_{i=1}^{N}\sum_{j=1}^{N} \pi_i \tilde{A}_{ij} \tag{38}$$

$$\leq \sum_{i=1}^{N} \pi_i \Big(\sum_{j=1}^{N} \tilde{A}_{ij}\Big) = 1 - \sum_{i=1}^{N} \pi_i \epsilon_i < 1, \tag{39}$$

so $\tilde{A}$ is strictly convergent. In contrast, our edge-calibrated operator reduces $\rho(\cdot)$, alleviating over-smoothing.

Theorem 16 (Edge calibration reduces local smoothing)**.** *Based on Corollary 15, we can infer that a matrix with k-disconnected sub-stochastic components satisfies $|\lambda_1| = |\lambda_2| = \cdots = |\lambda_k| = 1$. According to Definition 12, $J_\rho(\tilde{A}) = 1$ with multiple λ's on the unit circle does not meet the convergence property compared to the graph attention as in Lemma 14, relieving the smoothing effect in Eq. 5.*

Lastly, we show that edge calibration enhances the separability of signed and blocked MP as follows.

Theorem 17 (Separability)**.** *Message-passing with edge calibration outperforms the original MPs as below:*

- *($0.5 \leq e_t \leq 1$) Blocked MP (w/ calib) vs Signed MP*

$$Z_4 = \mathbb{E}_b(\cdot) - \mathbb{E}_s(\cdot) = \int_0^1 \int_{0.5}^1 (1 - b_i - e_t + 2e_t b_i) \tag{40}$$

- *($0 \leq e_t < 0.5$) Signed MP (w/ calib) vs Blocked MP*

$$Z_5 = \mathbb{E}_s(\cdot) - \mathbb{E}_b(\cdot) = \int_0^1 \int_0^{0.5} (1 - e_t - b_i) \tag{41}$$

The equations satisfy $Z_4 > 0$ and $Z_5 > 0$, indicating that edge calibration improves MPs for both conditions.

5.4 TIME COMPLEXITY

To begin with, note that the edge calibration component of our model is proportional to baseline algorithms such as GPRGNN, FAGCN, or GGCN. Therefore, we focus on the complexity of the parameter estimation network, which consists of plain MLP and even-hop propagation networks. Firstly, it is well known that the cost of the MLP follows $\mathcal{O}(nz(X)F' + F'C)$, where $nz(\cdot)$ represents the non-zero elements of the inputs, F' denotes the hidden dimension, and C is the number of classes. The second part involves even-hop message passing (MP), which can be defined as $\mathcal{O}(L|\mathcal{E}|\theta_{GCN}/2)$. In summary, the overall cost should be $\mathcal{O}(nz(X)F' + F'C + L|\mathcal{E}|\theta_{GCN}/2)$, which may require double the computational cost compared to the baselines.

6 EXPERIMENTS

We conduct extensive experiments to answer the following research questions:

- **Q1:** Does the proposed method improve the node classification accuracy of signed GNNs?
- **Q2:** Does edge calibration enhance the inter-class separability of signed propagation?

Table 1: Details of the six benchmark graphs. We follow the experimental settings of GCN Kipf and Welling [2016]

Datasets	Cora	Citeseer	Pubmed	Actor	Cham.	Squirrel
# Nodes	2,708	3,327	19,717	7,600	2,277	5,201
# Edges	10,558	9,104	88,648	25,944	33,824	211,872
# Features	1,433	3,703	500	931	2,325	2,089
# Labels	7	6	3	5	5	5
# Training	140	120	60	100	100	100
# Validation	1,083	1,330	7,886	3,040	910	2,080

- **Q3:** How does performance change given the values Z_t in Eq. 32?
- **Q4:** How much does the method improve the quality of GNNs on large benchmark graphs?

Datasets. The statistical details of the datasets are in Table 1. (1) *Cora, Citeseer, Pubmed* Kipf and Welling [2016] are citation graphs where a node corresponds to a paper, and edges are citations between them. The labels represent the research topics of the papers. (2) *Actor* Tang et al. [2009] is a co-occurrence graph of actors appearing in the same movie. The labels represent five types of actors. (3) *Chameleon, Squirrel* Rozemberczki et al. [2019] are Wikipedia hyperlink networks. Each node is a webpage, and the edges are hyperlinks. Nodes are categorized into five classes based on monthly traffic.

Implementation Details. All methods, including baselines and ours, are implemented using *PyTorch Geometric*[1]. For a fair comparison, we set the hidden dimension for all methodologies to 64. ReLU with dropout is used for non-linearity and to prevent overfitting. We use log-Softmax as the cross-entropy function. The learning rate is set to $1e^{-3}$, and the Adam optimizer is used with a weight decay of $5e^{-4}$. For training, 20 nodes per class are randomly chosen, and the remaining nodes are divided into two parts for validation and testing following Kipf and Welling [2016].

Baselines. For the experiment, we classified the baselines into two main categories: signed GNNs and others. As shown in Table 2, we applied our method to the following three signed propagation techniques for evaluation: GPRGNN Chien et al. [2020], FAGCN Bo et al. [2021], and GGCN Yan et al. [2021].

6.1 PERFORMANCE ANALYSIS (Q1)

Table 2 shows the node classification accuracy (%) of several state-of-the-art methods. We analyze the results from the two perspectives below.

Signed and blocked MPs Outperform Vanilla MP

The experimental results demonstrate that GNNs with signed or blocked MP outperform traditional methods under

[1]https://pytorch-geometric.readthedocs.io/en/latest/modules/nn.html

Table 2: (Q1) Node classification performance (%) with standard deviation (±) on the six benchmark graphs. The gray-colored cells indicate top-3 performance. A symbol $*$ means that edge calibration (Eq. 33) is applied on a base method

Datasets	Cora	Citeseer	Pubmed	Actor	Chameleon	Squirrel
$\mathcal{H}_g$ (Eq. 3)	0.81	0.74	0.8	0.22	0.23	0.22
GCN Kipf and Welling [2016]	80.5 ±0.73	70.4 ±0.62	78.6 ±0.44	20.2 ±0.40	49.3 ±0.58	30.7 ±0.70
GAT Velickovic et al. [2017]	81.2 ±0.51	71.3 ±0.75	79.0 ±0.45	22.5 ±0.36	48.8 ±0.83	30.8 ±0.94
APPNP Gasteiger et al. [2018]	81.8 ±0.51	71.9 ±0.38	79.0 ±0.30	23.8 ±0.32	48.0 ±0.77	30.4 ±0.61
GCNII Chen et al. [2020]	81.3 ±0.69	70.7 ±1.28	78.5 ±0.50	25.9 ±1.21	48.6 ±0.76	30.4 ±0.90
H_2GCN Zhu et al. [2020]	79.4 ±0.45	71.2 ±0.79	78.1 ±0.31	25.6 ±1.15	47.5 ±0.82	31.0 ±0.74
PTDNet Luo et al. [2021]	81.1 ±0.79	71.4 ±1.12	78.8 ±0.67	21.5 ±0.75	50.4 ±1.06	32.2 ±0.75
P-reg Yang et al. [2021]	81.0 ±0.88	71.9 ±0.81	78.5 ±0.42	21.2 ±0.52	50.6 ±0.37	**33.1** ±0.40
ACM-GCN Luan et al. [2022]	81.6 ±0.85	71.3 ±1.01	78.4 ±0.53	24.9 ±2.17	49.6 ±0.59	31.2 ±0.44
HOG-GCN Wang et al. [2022]	81.7 ±0.41	72.2 ±0.67	79.0 ±0.24	21.3 ±0.56	47.9 ±0.45	30.2 ±0.50
JacobiConv Wang and Zhang [2022]	81.9 ±0.69	72.0 ±0.75	78.7 ±0.42	26.0 ±1.04	51.6 ±1.10	32.1 ±0.73
GloGNN Li et al. [2022]	82.0 ±0.40	71.8 ±0.52	79.4 ±0.28	**26.6** ±0.71	48.3 ±0.39	30.8 ±0.80
AERO-GNN Lee et al. [2023]	81.6 ±0.54	71.1 ±0.62	79.1 ±0.47	25.5 ±1.08	49.8 ±2.33	29.9 ±1.96
Auto-HeG Zheng et al. [2023]	81.5 ±1.06	70.9 ±1.41	**79.2** ±0.24	26.1 ±0.98	48.7 ±1.37	31.5 ±1.11
TED-GCN Yan et al. [2024]	81.8 ±0.88	71.4 ±0.56	78.6 ±0.30	26.0 ±0.95	50.4 ±1.21	**33.0** ±0.98
PCNet Li et al. [2024a]	81.5 ±0.76	71.2 ±1.20	78.8 ±0.26	26.4 ±0.85	48.1 ±1.69	31.4 ±0.56
GPRGNN Chien et al. [2020]	81.1 ±0.56	71.0 ±0.83	78.7 ±0.55	24.8 ±0.87	50.2 ±0.79	30.2 ±0.62
GPRGNN*	**82.5** ±0.37	**72.4** ±0.66	**79.3** ±0.35	**26.9** ±0.74	**52.5** ±0.54	32.2 ±0.49
Relative improv. (+ %)	+ 1.73 %	+ 1.97 %	+ 0.76 %	+ 8.47 %	+ 4.58 %	+ 6.62 %
FAGCN Bo et al. [2021]	81.4 ±0.51	72.2 ±0.69	78.9 ±0.62	25.3 ±0.77	49.1 ±1.20	30.3 ±0.96
FAGCN*	**82.8** ±0.45	**73.4** ±0.50	**79.6** ±0.33	**27.8** ±0.58	**51.8** ±0.91	32.5 ±0.77
Relative improv. (+ %)	+ 1.72 %	+ 1.66 %	+ 0.89 %	+ 9.88 %	+ 5.50 %	+ 7.26 %
GGCN Yan et al. [2021]	81.2 ±1.06	71.5 ±1.44	78.3 ±0.35	23.7 ±0.75	50.0 ±0.98	30.4 ±0.72
GGCN*	**82.4** ±0.87	**73.0** ±0.72	79.0 ±0.34	25.6 ±0.61	**52.0** ±0.72	**32.7** ±0.69
Relative improv. (+ %)	+ 1.48 %	+ 2.10 %	+ 0.89 %	+ 8.02 %	+ 4.00 %	+ 7.57 %

benchmark graphs. For both homophilic and heterophilic datasets, we can see that vanilla MP-based algorithms like GCN, GAT, and APPNP show inferior accuracy than others. Especially, the performance gap increases as the homophily decreases. Specifically, on the Actor dataset, GloGNN relatively outperforms GCN by over 30%. In addition, under Chameleon and Squirrel with many cyclic edges, Auto-HeG, signed (GPRGNN, FAGCN, and GGCN) or blocked (PTDNet) MPs achieve state-of-the-art performance. This highlights the efficacy of signed and blocked GNNs in capturing the complex relationships present in general graphs, thereby leading to superior improvement.

Edge Weight Calibration Enhances the Performance of Signed GNNs

The results underscore the critical importance of edge weight calibration in enhancing the performance of signed GNNs. Models that incorporate edge weight adjustments, such as GPRGNN* and FAGCN*, consistently outperform their counterparts across a wide range of datasets, highlighting the robustness and generalizability of this approach. For instance, GPRGNN* demonstrates remarkable performance improvements on both the Cora and Pubmed datasets compared to the baseline GPRGNN, underscoring its adaptability and effectiveness in diverse graph settings. Similarly, FAGCN*, which integrates edge weight calibration, consistently outperforms the original FAGCN model on both homophilic and heterophilic graphs, showcasing its versatility across different connectivity patterns. Furthermore, our method leverages blocked MP by effectively removing cyclic edges in challenging datasets such as Chameleon and Squirrel, further contributing to enhanced model performance. These advancements highlight the significance of incorporating edge weight calibration in mitigating the smoothing effect under edge uncertainty and achieving higher classification accuracy across diverse scenarios, ultimately unlocking the full potential of signed GNNs.

6.2 DISCRIMINATION POWER (Q2)

In Figure 2, we present the inter-class distances for three GNNs with signed MP (GPRGNN, FAGCN, and GGCN) across four benchmark graphs (Cora, Citeseer, Actor, and Squirrel) to investigate a neural collapse perspective Kothapalli et al. [2024]. The y-axis measures the average L2 distance between classes after the first layer projection. To ensure fairness, we removed parameter randomness from all baselines. The ensemble methods (indicated with an asterisk) consistently show an increase in inter-class distance compared to their standard counterparts across all datasets. For instance, on the Cora dataset, GPRGNN* and FAGCN*

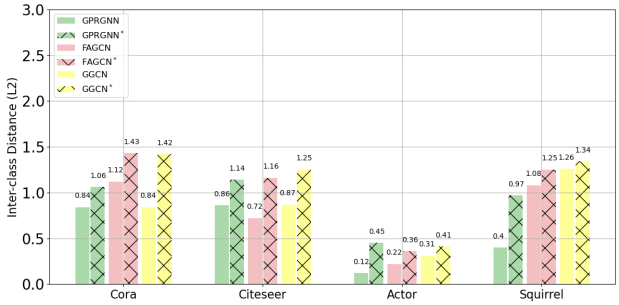

Figure 2: (Q2) We take three signed GNNs (GPRGNN, FAGCN, and GGCN) and measure the inter-class distances to show that our method improves the discrimination power

show increases to 1.06 and 1.43, respectively, compared to 0.84 and 1.12 for the standard models. The increase is even more substantial in heterophilic datasets like Actor and Squirrel, where the methods with edge calibration significantly improve separability compared to the standard versions, highlighting the effectiveness of blocked MP under certain conditions.

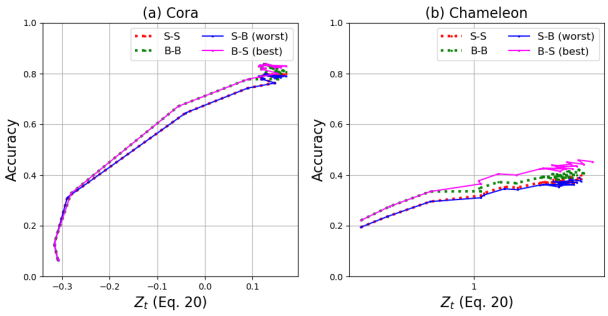

Figure 3: (Q3) Performance gain concerning signed/blocked propagation using FAGCN* model

6.3 IMPORTANCE OF MP SCHEMES (Q3)

We aim to show that selecting a proper propagation scheme between signed or blocked MPs is crucial for GNNs. As described in Figure 3, the x-axis stands for the Z_t in Eq. 32, and the y-axis represents node classification accuracy. For a fair comparison, we remove the randomness for all methods (e.g., parameter initialization) by fixing the seed. The left figure is (a) Cora, and the right one is (b) Chameleon. We have proved that blocked MP outperforms signed MP in the case of $Z_t < 0$, and vice versa. To verify this, we assume two types of MPs: signed (S) and blocked (B) under two different conditions: $Z_t < 0$ and $Z_t \geq 0$. For example, S-S means that signed MPs are used independent of Z_t, while B-S utilizes blocked MP if $Z_t < 0$ and signed MP for $Z_t \geq 0$. As illustrated, selecting the proper propagation scheme (B-S) achieves the best quality, where the performance gap becomes greater in the heterophilic graph (Chameleon). Conversely, improper MPs (S-B or S-S) when $Z_t < 0$ show low accuracy, showing that blocked MP can improve the prediction accuracy under this condition.

Table 3: (Q4) Node classification accuracy (%) on large heterophilic graphs. Penn94 has a binary class, while the arXiv-year and snap-patents have five classes

Datasets	Penn94	arXiv-year	snap-patents
$\mathcal{H}_g$ (Eq. 3)	0.046	0.272	0.1
GCN Kipf and Welling [2016]	81.3 $\pm$0.3	44.5 $\pm$0.3	43.9 $\pm$0.2
GAT Velickovic et al. [2017]	80.6 $\pm$0.6	45.0 $\pm$0.5	45.2 $\pm$0.6
GCNII Chen et al. [2020]	81.8 $\pm$0.7	46.1 $\pm$0.3	**47.5** $\pm$0.7
H_2GCN Zhu et al. [2020]	80.4 $\pm$0.6	**47.6** $\pm$0.2	OOM
GPRGNN Chien et al. [2020]	80.9 $\pm$0.3	43.6 $\pm$0.4	41.4 $\pm$0.1
GPRGNN*	**83.1** $\pm$0.4	44.9 $\pm$0.3	43.6 $\pm$0.3
Relative improv. (+ %)	2.72%	2.98%	5.3%

6.4 ANALYSIS ON LARGE GRAPHS (Q4)

We conduct experiments on large benchmarks Lim et al. [2021] and describe the results in Table 3. Due to the OOM issue, we apply our method to one of the memory-efficient signed GNNs, GPRGNN Chien et al. [2020]. As shown in the table, our method GPRGNN* demonstrates notable improvements in node classification accuracy across large graphs. Specifically, on the three datasets, GPRGNN* achieves an accuracy improvement of 2.72%, 2.98%, and 5.3% over the baseline GPRGNN, respectively. Although H_2GCN Zhu et al. [2020] and GCNII Chen et al. [2020] show the best performance on arXiv and snap, this is because the accuracy of the original GPRGNN is quite lower compared to the other methods. To summarize, these results highlight the effectiveness of our approach in enhancing the performance of GNNs on large-scale benchmarks.

7 CONCLUSION

This paper presents a comprehensive study on the impact of edge uncertainty and over-smoothing in signed Graph Neural Networks (GNNs). First, we scrutinize the smoothing effect under different values of Z_t (Eq. 32), offering a novel perspective that propagation schemes should account for varying homophily and edge classification error ratios. Second, we introduce an innovative training mechanism that dynamically selects between blocked and signed propagation based on these parameters, effectively mitigating over-smoothing and enhancing performance. Our theoretical analysis, supported by extensive experiments, demonstrates that blocking message propagation can be more effective than traditional message-passing schemes under certain conditions. This insight is crucial for improving node classification accuracy in both homophilic and heterophilic graphs. We hope that future work will explore further refinements of these techniques for more complex graph structures.

Acknowledgments

This work was supported by the KENTECH Research Grant (202200019A), by the Institute of Information & Communications Technology Planning & Evaluation (IITP)-Innovative Human Resource Development for Local Intellectualization program grant funded by the Korea government (MSIT) (IITP-2025-RS-2022-00156287), and by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2021-II212068, Artificial Intelligence Innovation Hub).

References

Baranwal Aseem, Fountoulakis Kimon, Jagannath Aukosh, and al et. Effects of graph convolutions in multi-layer networks. In *The Eleventh International Conference on Learning Representations*, 2023.

Aseem Baranwal, Kimon Fountoulakis, and Aukosh Jagannath. Graph convolution for semi-supervised classification: Improved linear separability and out-of-distribution generalization. *arXiv preprint arXiv:2102.06966*, 2021.

Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. *arXiv preprint arXiv:2101.00797*, 2021.

Deyu Bo, Chuan Shi, Lele Wang, and Renjie Liao. Specformer: Spectral graph neural networks meet transformers. *arXiv preprint arXiv:2303.01028*, 2023.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.

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

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, pages 1725–1735. PMLR, 2020.

Yuhan Chen, Yihong Luo, Jing Tang, Liang Yang, Siya Qiu, Chuan Wang, and Xiaochun Cao. Lsgnn: Towards general graph neural network in node classification by local similarity. *arXiv preprint arXiv:2305.04225*, 2023.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. *arXiv preprint arXiv:2006.07988*, 2020.

Yoonhyuk Choi and Chong-Kwon Kim. Hierarchical uncertainty-aware graph neural network. *arXiv preprint arXiv:2504.19820*, 2025a.

Yoonhyuk Choi and Chong-Kwon Kim. Specsphere: Dual-pass spectral-spatial graph neural networks with certified robustness. *arXiv preprint arXiv:2505.08320*, 2025b.

Yoonhyuk Choi, Jiho Choi, Taewook Ko, Hyungho Byun, and Chong-Kwon Kim. Finding heterophilic neighbors via confidence-based subgraph matching for semi-supervised node classification. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 283–292, 2022.

Yoonhyuk Choi, Jiho Choi, Taewook Ko, and Chong-Kwon Kim. Is signed message essential for graph neural networks. *arXiv preprint arXiv:2301.08918*, 2023.

Mary Kathryn Cowles and Bradley P Carlin. Markov chain monte carlo convergence diagnostics: a comparative review. *Journal of the American statistical Association*, 91 (434):883–904, 1996.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Tyler Derr, Yao Ma, and Jiliang Tang. Signed graph convolutional networks. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 929–934. IEEE, 2018.

Yash Deshpande, Subhabrata Sen, Andrea Montanari, and Elchanan Mossel. Contextual stochastic block models. *Advances in Neural Information Processing Systems*, 31, 2018.

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

Zheng Fang, Lingjun Xu, Guojie Song, Qingqing Long, and Yingxue Zhang. Polarized graph neural networks. In *Proceedings of the ACM Web Conference 2022*, pages 1404–1413, 2022.

Johannes Gasteiger, Aleksandar Bojchevski, Stephan Gunnemann, and Johns. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018.

Yuhe Guo and Zhewei Wei. Clenshaw graph neural networks. *arXiv preprint arXiv:2210.16508*, 2022.

Yang Hu, Haoxuan You, Zhecan Wang, Zhicheng Wang, Erjin Zhou, and Yue Gao. Graph-mlp: Node classification without message passing in graph. *arXiv preprint arXiv:2106.04051*, 2021.

Junjie Huang, Huawei Shen, Liang Hou, and Xueqi Cheng. Signed graph attention networks. In *International Conference on Artificial Neural Networks*, pages 566–577. Springer, 2019.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Taewook Ko, Yoonhyuk Choi, and Chong-Kwon Kim. A spectral graph convolution for signed directed graphs via magnetic laplacian. *Neural Networks*, 164:562–574, 2023a.

Taewook Ko, Yoonhyuk Choi, and Chong-Kwon Kim. Universal graph contrastive learning with a novel laplacian perturbation. In *Uncertainty in Artificial Intelligence*, pages 1098–1108. PMLR, 2023b.

Vignesh Kothapalli, Tom Tirer, and Joan Bruna. A neural collapse perspective on feature evolution in graph neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

Soo Yong Lee, Fanchen Bu, Jaemin Yoo, and Kijung Shin. Towards deep attention in graph neural networks: Problems and remedies. In *International Conference on Machine Learning*, pages 18774–18795. PMLR, 2023.

Runlin Lei, Zhen Wang, Yaliang Li, Bolin Ding, and Zhewei Wei. Evennet: Ignoring odd-hop neighbors improves robustness of graph neural networks. *arXiv preprint arXiv:2205.13892*, 2022.

Bingheng Li, Erlin Pan, and Zhao Kang. Pc-conv: Unifying homophily and heterophily with two-fold filtering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 13437–13445, 2024a.

Ting Wei Li, Qiaozhu Mei, and Jiaqi Ma. A metadata-driven approach to understand graph neural networks. *Advances in Neural Information Processing Systems*, 36, 2024b.

Xiang Li, Renyu Zhu, Yao Cheng, Caihua Shan, Siqiang Luo, Dongsheng Li, and Weining Qian. Finding global homophily in graph neural networks when meeting heterophily. *arXiv preprint arXiv:2205.07308*, 2022.

Ningyi Liao, Siqiang Luo, Xiang Li, and Jieming Shi. Ld2: Scalable heterophilous graph neural network with decoupled embeddings. *Advances in Neural Information Processing Systems*, 36, 2024.

Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. *Advances in Neural Information Processing Systems*, 34:20887–20902, 2021.

Qincheng Lu, Jiaqi Zhu, Sitao Luan, and Xiao-Wen Chang. Representation learning on heterophilic graph with directional neighborhood attention. *arXiv preprint arXiv:2403.01475*, 2024.

Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Revisiting heterophily for graph neural networks. *arXiv preprint arXiv:2210.07606*, 2022.

Dongsheng Luo, Wei Cheng, Wenchao Yu, Bo Zong, Jingchao Ni, Haifeng Chen, and Xiang Zhang. Learning to drop: Robust graph neural network via topological denoising. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 779–787, 2021.

Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. Is homophily a necessity for graph neural networks? *arXiv preprint arXiv:2106.06134*, 2021.

Haitao Mao, Zhikai Chen, Wei Jin, Haoyu Han, Yao Ma, Tong Zhao, Neil Shah, and Jiliang Tang. Demystifying structural disparity in graph neural networks: Can one size fit all? *Advances in neural information processing systems*, 36, 2024.

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

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*, 2020.

Oleg Platonov, Denis Kuznedelev, Artem Babenko, and Liudmila Prokhorenkova. Characterizing graph datasets for node classification: Homophily-heterophily dichotomy and beyond. *Advances in Neural Information Processing Systems*, 36, 2024.

Chenyang Qiu, Guoshun Nan, Tianyu Xiong, Wendi Deng, Di Wang, Zhiyang Teng, Lijuan Sun, Qimei Cui, and Xiaofeng Tao. Refining latent homophilic structures over heterophilic graphs for robust graph convolution networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8930–8938, 2024.

Benedek Rozemberczki, Ryan Davies, Rik Sarkar, and Charles Sutton. Gemsec: Graph embedding with self clustering. In *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining*, pages 65–72, 2019.

Eugene Seneta. *Non-negative matrices and Markov chains*. Springer Science & Business Media, 2006.

Boshi Tang, Zhiyong Wu, Xixin Wu, Qiaochu Huang, Jun Chen, Shun Lei, and Helen Meng. Simcalib: Graph neural network calibration based on similarity between nodes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15267–15275, 2024.

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

Yijun Tian, Chuxu Zhang, Zhichun Guo, Xiangliang Zhang, and Nitesh Chawla. Learning mlps on graphs: A unified view of effectiveness, robustness, and efficiency. In *The Eleventh International Conference on Learning Representations*, 2022.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *stat*, 1050:20, 2017.

Tao Wang, Di Jin, Rui Wang, Dongxiao He, and Yuxiao Huang. Powerful graph convolutional networks with adaptive propagation mechanism for homophily and heterophily. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4210–4218, 2022.

Xiyuan Wang and Muhan Zhang. How powerful are spectral graph neural networks. In *International Conference on Machine Learning*, pages 23341–23362. PMLR, 2022.

Xinyi Wu, Amir Ajorlou, Zihui Wu, and Ali Jadbabaie. Demystifying oversmoothing in attention-based graph neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pages 5453–5462. PMLR, 2018.

Yuchen Yan, Yuzhong Chen, Huiyuan Chen, Minghua Xu, Mahashweta Das, Hao Yang, and Hanghang Tong. From trainable negative depth to edge heterophily in graphs. *Advances in Neural Information Processing Systems*, 36, 2024.

Yujun Yan, Milad Hashemi, Kevin Swersky, Yaoqing Yang, and Danai Koutra. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. *arXiv preprint arXiv:2102.06462*, 2021.

Han Yang, Kaili Ma, and James Cheng. Rethinking graph regularization for graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4573–4581, 2021.

Kai Zhao, Qiyu Kang, Yang Song, Rui She, Sijie Wang, and Wee Peng Tay. Graph neural convection-diffusion with heterophily. *arXiv preprint arXiv:2305.16780*, 2023.

Xin Zheng, Miao Zhang, Chunyang Chen, Qin Zhang, Chuan Zhou, and Shirui Pan. Auto-heg: Automated graph neural network on heterophilic graphs. *arXiv preprint arXiv:2302.12357*, 2023.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems*, 33:7793–7804, 2020.

TECHNICAL APPENDIX

A PROOF OF LEMMA 3 - 5

In this section, we derive the expected distribution after three types of message-passing (MP) using a multi-class contextual stochastic block model (CSBM), respectively.

- **Vanilla MP** inherits the original matrix, which only consists of positive edges.

$$\forall (i,j) \in \mathcal{E}, \;\; \tilde{A} = D^{-1}A \geq 0 \tag{42}$$

- **Signed MP** assigns negative values to the heterophilic edges, where $y_i \neq y_j$.

$$\forall (i,j) \in \mathcal{E}, \;\; \tilde{A} \in \begin{cases} D^{-1}A, & y_i = y_j \\ -D^{-1}A, & y_i \neq y_j \end{cases} \tag{43}$$

- **Blocked MP** blocks the information propagation for heterophilic edges by assigning zero.

$$\forall (i,j) \in \mathcal{E}, \;\; \tilde{A} \in \begin{cases} D^{-1}A, & y_i = y_j \\ 0, & y_i \neq y_j \end{cases} \tag{44}$$

A.1. Proof of Lemma 3 (Vanilla MP)

Let's assume that $y_i = 0$, ego $k \sim (\mu, \pi/2, 0)$, and aggregated neighbors $k' \sim (\mu, \pi/2, \theta')$. Though each neighbor has multiple distributions proportional to the number of classes, their aggregation always satisfies $|k'| \leq \mu$ since the summation of coefficients $(1 - b_i)$ is lower than 1. Thus, we indicate k'_{aggr} as k' here for brevity. Given this, the expectation after vanilla MP $\mathbb{E}_p(h_i^{(1)}|y_i, d_i)$ can be retrieved as below:

$$\begin{aligned}
\mathbb{E}_p(h_i^{(1)}|y_i, d_i) &= \frac{k}{d_i+1} + \sum_{j \in \mathcal{N}_i} \left(\frac{k(1-e)+ke}{\sqrt{(d_i+1)(d_j+1)}} b_i + \frac{k'(1-e)+k'e}{\sqrt{(d_i+1)(d_j+1)}}(1-b_i) \right) && (45)\\
&= \frac{k}{d_i+1} + \sum_{j \in \mathcal{N}_i} \left(\frac{kb_i + k'(1-b_i)}{\sqrt{(d_i+1)(d_j+1)}} \right) && (46)\\
&= \frac{k}{d_i+1} + \frac{\{kb_i + k'(1-b_i)\}d'_i}{d_i+1} && (47)\\
&= \frac{\{kb_i + k'(1-b_i)\}d'_i + k}{d_i+1} && (48)\\
& && (49)
\end{aligned}$$

A.2. Proof of Lemma 4 (Signed MP) Similar to the above analysis, we can retrieve the expectation $\mathbb{E}_s(h_i^{(1)}|y_i, d_i)$ after signed MP. For this, we employ the edge classification error rate e, which determines the error ratio of finding heterophilic edges as follows:

$$\begin{aligned}
\mathbb{E}_s(h_i^{(1)}|y_i, d_i) &= \frac{k}{d_i+1} + \sum_{j \in \mathcal{N}_i} \left(\frac{k(1-e)-ke}{\sqrt{(d_i+1)(d_j+1)}} b_i + \frac{-k'(1-e)+k'e}{\sqrt{(d_i+1)(d_j+1)}}(1-b_i) \right) && (50)\\
&= \frac{k}{d_i+1} + \sum_{j \in \mathcal{N}_i} \left(\frac{k(1-2e)b_i - k'(1-2e)(1-b_i)}{\sqrt{(d_i+1)(d_j+1)}} \right) && (51)\\
&= \frac{k}{d_i+1} + \sum_{j \in \mathcal{N}_i} \left(\frac{(1-2e)\{kb_i + k'(b_i-1)\}}{\sqrt{(d_i+1)(d_j+1)}} \right) && (52)\\
&= \frac{k}{d_i+1} + \frac{(1-2e)\{kb_i + k'(b_i-1)\}d'_i}{d_i+1} && (53)\\
&= \frac{(1-2e)\{b_i k + (b_i-1)k'\}d'_i + k}{d_i+1}. && (54)
\end{aligned}$$

A.3. Proof of Lemma 5 (Blocked MP) Lastly, the expectation $\mathbb{E}_b(h_i^{(1)}|y_i, d_i)$ of assigning zero weights for heterophilic edges is given by:

$$\mathbb{E}_b(h_i^{(1)}|y_i, d_i) = \frac{k}{d_i+1} + \sum_{j\in\mathcal{N}_i}\left(\frac{k(1-e)}{\sqrt{(d_i+1)(d_j+1)}}b_i + \frac{k'e}{\sqrt{(d_i+1)(d_j+1)}}(1-b_i)\right) \tag{55}$$

$$= \frac{k}{d_i+1} + \sum_{j\in\mathcal{N}_i}\left(\frac{k(1-e)b_i + k'e(1-b_i)}{\sqrt{(d_i+1)(d_j+1)}}\right) \tag{56}$$

$$= \frac{k}{d_i+1} + \frac{\{(1-e)b_ik + e(1-b_i)k'\}d_i'}{d_i+1} \tag{57}$$

$$= \frac{\{(1-e)b_ik + e(1-b_i)k'\}d_i' + k}{d_i+1}. \tag{58}$$

B DETAILS OF EMPIRICAL ANALYSIS

Figure 1 presents empirical analyses to verify Theorems 10 (left) and 11 (right). For homophily estimation, we use early stopping, saving the predictions with the best validation score if there is no improvement for more than 100 epochs. The edge error estimation follows a slightly different approach, where the *(true)* values are based on the maximum predicted node class as shown below:

$$\forall(i,j)\in\mathcal{E},\ \ \tilde{\mathcal{E}}_{ij}^t \in \begin{cases}1, & \widehat{Y}_i^t = \widehat{Y}_j^t\\ 0, & \widehat{Y}_i^t \neq \widehat{Y}_j^t\end{cases} \tag{59}$$

where $\widehat{Y}^t$ stands for the predicted class at step t. The score s can be derived as follows:

$$s = 1\frac{1^T\sum_{i=1}^N\sum_{j=1}^N\tilde{\mathcal{E}}_{ij}^t}{|\mathcal{E}|} \tag{60}$$

C BASELINES

- **GCN** Kipf and Welling [2016] is a first-order approximation of Chebyshev polynomials Defferrard et al. [2016]. For all datasets, we simply take 2 layers of GCN.
- **APPNP** Gasteiger et al. [2018] combines personalized PageRank on GCN. We stack 10 layers and set the teleport probability (α) as $\{0.1, 0.1, 0.1, 0.5, 0.2, 0.3\}$ for Cora, Citeseer, Pubmed, Actor, Chameleon, and Squirrel.
- **GAT** Velickovic et al. [2017] calculates feature-based attention for edge coefficients. We construct 2 layers of GAT, the pair of (hidden dimension, head) is set as (8, 8) for the first layer, while the second layer is (1, # classes).
- **GCNII** Chen et al. [2020] integrates an identity mapping function on APPNP. We set $\alpha = 0.5$ and employ nine hidden layers. We increase the weight of identity mapping (β) that is inversely proportional to the heterophily of the dataset.
- **H_2GCN** Zhu et al. [2020] suggests the separation of ego and neighbors during aggregation. We refer to the publicly available *source code*[2] for implementation.
- **PTDNet** Luo et al. [2021] removes disassortative edges before a message-passing. We also utilize the open *source code*[3] and apply confidence calibration.
- **P-reg** Yang et al. [2021] ensembles a regularization term to provide additional information that training nodes might not capture.(*source code*[4])
- **HOG-GCN** Wang et al. [2022] adaptively controls the propagation mechanism by measuring the homophily degrees between two nodes. (*source code*[5])
- **JacobiConv** Wang and Zhang [2022] studies the expressive power of spectral GNN and establishes a connection with the graph isomorphism testing. (*source code*[6])

[2]https://github.com/GemsLab/H2GCN
[3]https://github.com/flyingdoog/PTDNet
[4]https://github.com/yang-han/P-reg
[5]https://github.com/hedongxiao-tju/HOG-GCN
[6]https://github.com/GraphPKU/JacobiConv

- **GloGNN** Li et al. [2022] receives information from global nodes, which can accelerate neighborhood aggregation. (*source code*[7])
- **ACM-GCN** Luan et al. [2022] suggests a local diversification operation through the adaptive channel mixing algorithm. (*source code*[8])
- **AERO-GCN** Lee et al. [2023] improves the deep graph attention to reduce the smoothing effect and improve the performance at deep layers (*source code*[9]).
- **Auto-HeG** Zheng et al. [2023] automatically build heterophilic GNN models with search space design, supernet training, and architecture selection *(source code)*[10].
- **TED-GCN** Yan et al. [2024] redefines GCN's depth L as a trainable parameter, which can control its signal processing capability to model both homophily/heterophily graphs.
- **PCNet** Li et al. [2024a] proposes a two-fold filtering mechanism to extract homophily in heterophilic graphs *(source code)*[11].
- **GPRGNN** Chien et al. [2020] generalized the personalized PageRank to deal with heterophily and over-smoothing. Referring to the open source *code*[12], we tune the hyper-parameters based on the best validation score for each dataset.
- **FAGCN** Bo et al. [2021] determines the sign of edges using the node features. We implement the algorithm based on the *sources*[13] and tune the hyper-parameters concerning their accuracy.
- **GGCN** Yan et al. [2021] proposes the scaling of degrees and the separation of positive/negative adjacency matrices. We simply take the publicly available *code*[14] for evaluation.

[7] https://github.com/RecklessRonan/GloGNN
[8] https://github.com/SitaoLuan/ACM-GNN
[9] https://github.com/syleeheal/AERO-GNN
[10] https://github.com/Amanda-Zheng/Auto-HeG
[11] https://github.com/uestclbh/PC-Conv
[12] https://github.com/jianhao2016/GPRGNN
[13] https://github.com/bdy9527/FAGCN
[14] https://github.com/Yujun-Yan/Heterophily_and_oversmoothing