Sign is Not a Remedy: Multiset-to-Multiset Message Passing for Learning on Heterophilic Graphs

Langzhang Liang [1] **Sunwoo Kim** [2] **Kijung Shin** [2] **Zenglin Xu** [3 4 1 5] **Shirui Pan** [6] **Yuan Qi** [3 4]

Abstract

Graph Neural Networks (GNNs) have gained significant attention as a powerful modeling and inference method, especially for homophilic graph-structured data. To empower GNNs in heterophilic graphs, where adjacent nodes exhibit dissimilar labels or features, Signed Message Passing (SMP) has been widely adopted. However, there is a lack of theoretical and empirical analysis regarding the limitations of SMP. In this work, we unveil some potential pitfalls of SMP and their remedies. We first identify two limitations of SMP: undesirable representation update for multi-hop neighbors and vulnerability against oversmoothing issues. To overcome these challenges, we propose a novel message-passing function called Multiset to Multiset GNN (M2M-GNN). Our theoretical analyses and extensive experiments demonstrate that M2M-GNN effectively alleviates the aforementioned limitations of SMP, yielding superior performance in comparison.

1. Introduction

Graph Neural Networks (GNNs) (Defferrard et al., 2016; Kipf & Welling, 2016; Hamilton et al., 2017; Veličković et al., 2018) have emerged as a prominent approach for machine learning on graph-structured data, such as social networks (Tang et al., 2013) and biomedical networks (Lin et al., 2020). In essence, GNNs employ message passing to iteratively aggregate information from neighboring nodes, ultimately yielding node embeddings for downstream tasks. Despite their success, the effectiveness of GNNs often diminishes on heterophilic graphs, where neighboring nodes typically have dissimilar labels or features. Surprisingly, GNNs can even be outperformed by graph-agnostic models like MLPs (Zhu et al., 2020; Zheng et al., 2022). This issue arises from the inherent smoothing effect of message passing, which tends to make adjacent nodes have similar embeddings.

To address this limitation, Signed Message Passing (SMP) has been widely adopted (Bo et al., 2021; Chien et al., 2020; Yan et al., 2022; Lee et al., 2023). SMP incorporates negative weights in message passing to push neighboring nodes apart in the embedding space. Consequently, SMP is also recognized as an effective solution for alleviating a key drawback of GNNs known as oversmoothing (Li et al., 2018). It refers to the tendency of node embeddings to become indistinguishable as multiple GNN layers are stacked.

However, in this work, we uncover two significant limitations of SMP. First, we prove that, even when the message-passing weights are "desirable" for direct neighbors, the cumulative weights for multi-hop neighbors may not necessarily be desirable. Second, we reveal that, even when message-passing weights are desirable, SMP can still be vulnerable to oversmoothing. More importantly, we point out that these limitations of SMP arise from its simple aggregator, which linearly combines the embeddings of all neighbors of each node into a single message vector.

Motivated by our findings, we propose a novel message passing scheme for GNNs, which we refer to as **M2M-GNN** (**M**ultiset **to** **M**ultiset GNN), to address the limitations of SMP. In essence, M2M-GNN aggregates neighborhood information by mapping the embeddings of neighbors, which can be considered a multiset of vectors, to another multiset. This differs from existing message passing schemes, where a multiset of vectors is mapped to a single message vector. We prove M2M-GNN's theoretical properties related to alleviating the limitations of SMP together with its empirical effectiveness. Our code is available at `https://github.com/Jinx-byebye/m2mgnn`

Our contributions are summarized as follows:

[1] Harbin Institute of Technology, Shenzhen, Shenzhen, China [2] Korea Advanced Institute of Science and Technology, Seoul, South Korea [3] Fudan University, Shanghai, China [4] Shanghai Academy of Artificial Intelligence for Science, Shanghai, China [5] Pengcheng Laboratory, Shenzhen, China [6] Griffith University, Gold Coast, Australia. Correspondence to: Langzhang Liang <lazylzliang@gmail.com>, Zenglin Xu <zenglin@gmail.com>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

- **Analysis:** Our theoretical analyses reveal two fundamental limitations of SMP.
- **Method:** We introduce M2M-GNN, a novel multiset-to-multiset message passing scheme, and illustrate how it mitigates the aforementioned limitations.
- **Experiments:** We conduct comprehensive experiments that demonstrate the superior performance of M2M-GNN.

2. Limitations of Signed Message Passing

In this section, we introduce two limitations of signed message passing (SMP): (1) undesirable representation update and (2) vulnerability to over-smoothing. To this end, we begin by introducing relevant notations and preliminaries. We provide all proofs in Appendix B.

2.1. Preliminaries

Basic terminologies. For any positive integer $N \in \mathbb{N}_+$, we denote the set $\{1, 2, \ldots, N\}$ as $[N]$. For any matrix $\mathbf{M}$, we use $\mathbf{M}_i = \mathbf{m}_i$ to denote its i-th row, and we use $\mathbf{M}_{ij}$ to denote the element in the i-th row and j-th column of $\mathbf{M}$. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ denote an undirected graph, where $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$ represents the node set and $\mathcal{E}$ represents the edge set. The nodes are characterized by a node feature matrix $\mathbf{X} \in \mathbb{R}^{N\times d}$, where d denotes the number of features per node. Let $\mathbf{y} \in [C]^N$ be the node label matrix where $y_i \in [C]$ denotes the label of node v_i, and C denotes the number of classes. A node pair with the same label is termed *homophilic nodes*, while a node pair with distinct labels is called *heterophilic nodes*. Here, the edge set $\mathcal{E}$ can be represented as an adjacency matrix $\mathbf{A} \in \{0, 1\}^{N\times N}$ where $\mathbf{A}_{ij} = 1$ if $\{v_i, v_j\} \in \mathcal{E}$ and $\mathbf{A}_{ij} = 0$ otherwise.

Message passing. Given a graph $\mathcal{G}$ with its adjacency matrix $\mathbf{A}$, *Message Passing* (MP) involves iteratively updating node embeddings by aggregating neighbors' information. The d'-dimensional node embeddings at each k-th layer, denoted as $\mathbf{H}^{(k)} \in \mathbb{R}^{N\times d'}$, can be expressed as follows:

$$\mathbf{H}^{(k)} = \mathtt{MP}(\mathbf{A}, \mathbf{H}^{(k-1)}, \mathbf{W}) = \sigma\left(\mathbf{A}\mathbf{H}^{(k-1)}\mathbf{W}\right), \quad (1)$$

where $\sigma(\cdot)$ denotes a (non-linear) activation function, and $\mathbf{W}$ denotes a learnable weight matrix. Instead of directly employing $\mathbf{A}$, many architectures utilize its variant denoted as $\mathcal{A}$, which we call a ***propagation matrix***. Examples of $\mathcal{A}$ include normalized $\mathbf{A}$ in Kipf & Welling (2016) ($\mathcal{A} = \mathbf{D}^{-0.5}\mathbf{A}\mathbf{D}^{-0.5}$ with $\mathbf{D}$ being the diagonal degree matrix) and learnable ones in Graph Attention Networks (Veličković et al., 2018) (where non-zero elements are computed using adjacent nodes' representations). We call the element $\mathcal{A}_{ij}$ the ***propagation coefficient*** for edge $\{v_i, v_j\}$. To clarify, Eq. (1) represents the message passing frameworks utilizing the mean aggregator as its aggregation function. There exist other design choices. We use this definition as Eq. (1) is sufficiently general to analyze SMP models.

Signed message passing. *Signed Message Passing* (SMP) enables $\mathcal{A}$ to incorporate both positive and negative values (Chien et al., 2020; Bo et al., 2021; Luan et al., 2022; Yan et al., 2022). By introducing more flexibility in $\mathcal{A}$, SMP facilitates adaptive updates of node embeddings in heterophilic graphs, as elaborated in Section 2.2.

2.2. Limitation 1: Undesirable Embedding Update

We first outline the mechanism of SMP and its particular benefits in heterophilic graphs. Subsequently, we show that, contrary to our expectation, this mechanism does not consistently yield beneficial updates of node embeddings.

Desirable embedding update. For accurate node classification, how should node embeddings be updated? In general, nodes with similar embeddings are more likely to be classified into the same class. Thus, intuitively, in an ideal scenario, embeddings of homophilic nodes would be closely located, while embeddings of heterophilic nodes would be relatively distant from each other. "Desirable" SMP (i.e., desirable $\mathcal{A}$) facilitates this circumstance by assigning positive and negative propagation coefficients to homophilic and heterophilic node pairs, respectively. This assignment is based on the understanding that positive coefficients tend to increase similarity between the embedding of pairs, while negative coefficients have the opposite effect. This phenomenon has been observed in numerous prior studies (Bo et al., 2021; Yan et al., 2022; Choi et al., 2023).

Based on the suggested ideal scenario, a desirable propagation matrix is defined as follows:

Definition 2.1 (Desirable matrix)**.** A matrix $\mathbf{M} \in \mathbb{R}^{N\times N}$ is ***desirable*** if and only if the following holds: for any $v_i, v_j \in \mathcal{V}$, $\mathbf{M}_{ij} \geq 0$, if $y_i = y_j$; $\mathbf{M}_{ij} \leq 0$, if $y_i \neq y_j$.

Message passing scheme. In practice, multi-layer GNNs are commonly employed to expand the receptive field of graph convolution. Following prior analyses (Chien et al., 2020; Bo et al., 2021) we consider linear SMP with multiple layers, formalized as:

$$\begin{aligned}\mathbf{H}^{(K)} &= \mathcal{A}^{(K)} \cdots \mathcal{A}^{(1)}\mathbf{H}^{(0)} \\ &= \prod\nolimits_{k=0}^{K-1} \mathcal{A}^{(K-k)}\mathbf{H}^{(0)} := \mathcal{T}\mathbf{H}^{(0)},\end{aligned} \quad (2)$$

where $\mathcal{A}^{(k)}$ is a (learnable) propagation matrix for the k-th layer, $\mathcal{T} := \prod_{k=0}^{K-1} \mathcal{A}^{(K-k)}$ is the ***cumulative propagation matrix***, and $\mathbf{H}^{(0)} = f_\theta(\mathbf{X})$ with $f_\theta(\cdot)$ being a neural network parameterized by θ.

Typically, matrices $\mathcal{A}^{(k)}$'s have the same sparsity pattern of the adjacency matrix $\mathbf{A}$. Note that the sparsity pattern identifies the specific coordinates within the matrix that contain non-zero elements. Thus, their product, $\mathcal{T}$, has the same sparsity pattern of $\mathbf{A}^K$, i.e., the K-th power of the adjacency matrix, where $\mathbf{A}^K_{ij} \neq 0$ only if v_i and v_j have a distance at most K. That is, $\mathcal{A}^{(k)}$ and $\mathcal{T}$ can have non-zero propagation coefficients for direct neighbors and K-hop reachable neighbors, respectively.

Theoretical result. Under the aforementioned design choice of SMP, one would expect that with desirable $\mathcal{A}^{(k)}, \forall k \in [K]$, the resulting $\mathcal{T}$ should also be desirable. Surprisingly, our analysis reveals the existence of counterexamples that are prevalent in the real worlds, specifically graphs with more than two distinct node classes.

Theorem 2.2 (Undesirability of SMP in multi-class cases)**.** *There exists a graph $\mathcal{G}$ with more than two distinct node classes (i.e., $C > 2$) where all propagation matrix $\mathcal{A}^{(1)}, \mathcal{A}^{(2)}, \cdots, \mathcal{A}^{(K)}$ are* ***desirable*** *(Def. 2.1), but the cumulative propagation matrix $\mathcal{T}$ is* ***not desirable****.*

The presence of undesirable $\mathcal{T}$ implies that, in the embedding space, nodes may end up being close to their heterophilic multi-hop neighbors, resulting in similar embeddings and, consequently, classification into the same class.

2.3. Limitation 2: Vulnerability to Oversmoothing

Next, we introduce another major limitation of SMP, i.e., its susceptibility to oversmoothing. Previous works have primarily focused on the potential advantages of SMP in mitigating oversmoothing, particularly its capacity to utilize negative propagation coefficients to separate nodes within the embedding space (Chien et al., 2020; Bo et al., 2021; Yan et al., 2022). However, through our theoretical and empirical investigation, we verify that even in a highly ideal setting, SMP is still susceptible to oversmoothing.

Below, we first introduce the analysis setup and then present our theoretical findings. Lastly, we review simulations designed to validate our theoretical findings.

Setup: (1) Propagation matrix and features. In this analysis, we assume a desirable propagation matrix $\mathcal{A}$ and node features $\mathbf{X}$ generated by a variant of the Contextual Stochastic Block Model (CSBM) (Deshpande et al., 2018), where each node feature is generated from a particular Gaussian distribution corresponding to its class, and edges are sampled independently from Bernoulli distributions. CSBMs have been widely employed for theoretical analyses of graphs (Wu et al., 2022; Lee et al., 2024)

Definition 2.3 (CSBM for $\mathcal{A}$ and $\mathbf{X}$)**.** Assume N nodes and C classes, each having an equal number of nodes. For each node $v_i \in \mathcal{V}$, its feature vector $\mathbf{x}_i$ is sampled from $\mathcal{N}(\mathbf{u}_{y_i}, \Sigma) \in \mathbb{R}^f$. For each node pair $\{v_i, v_j\} \in \binom{\mathcal{V}}{2}$, (1) $\mathcal{A}_{ij} = 1$, if $(y_i = y_j) \vee (\mathcal{U} < p)$ holds, and (2) $\mathcal{A}_{ij} = -1$, if $(y_i \neq y_j) \vee (\mathcal{U} < q)$ holds, where $\mathcal{U} \sim$ uniform(0,1) is sampled independently for each node pair. As a result, we have a desirable propagation matrix $\mathcal{A} \in \{-1, 0, 1\}^{N \times N}$ and a node feature matrix $\mathbf{X} \in \mathbb{R}^{N \times f}$.

Setup: (2) Message passing. Following prior oversmoothing analyses (Zhou et al., 2021; Keriven, 2022; Wu et al., 2022; Lee et al., 2023), we employ a simple message passing scheme, spec., the product of weighted (for SMP, signed) propagation matrices without non-linear activation. Formally, given a signed propagation matrix $\mathcal{A} \in \{-1, 0, 1\}^{N \times N}$, we normalize it as $\mathcal{P} := D^{-\frac{1}{2}} \mathcal{A} D^{-\frac{1}{2}}$, where D is a diagonal matrix with $D_{ii} = \sum_{j=1}^{N} |\mathcal{A}_{ij}|, \forall i \in [N]$. Then, the node embedding matrix at the K-th layer is:

$$\mathbf{H}^{(K)} = \mathcal{P}\mathbf{H}^{(K-1)} = \mathcal{P}^K \mathbf{H}^{(0)} := \mathcal{P}^K \mathbf{X}. \tag{3}$$

As SMP in our theoretical analysis, we use Eq. (3), which is very similar to FAGCN (Bo et al., 2021). It has a desirable property: $\mathcal{L} = \mathcal{I} - \mathcal{P}$ is a well-defined Laplacian of signed graphs (Atay & Tuncel, 2014), which ensures that the spectral norm of $\mathcal{P}^K$ is bounded, even when infinite layers are stacked. In Sec. 4, we present relevant experimental results using popular SMP-based methods and real-world graphs.

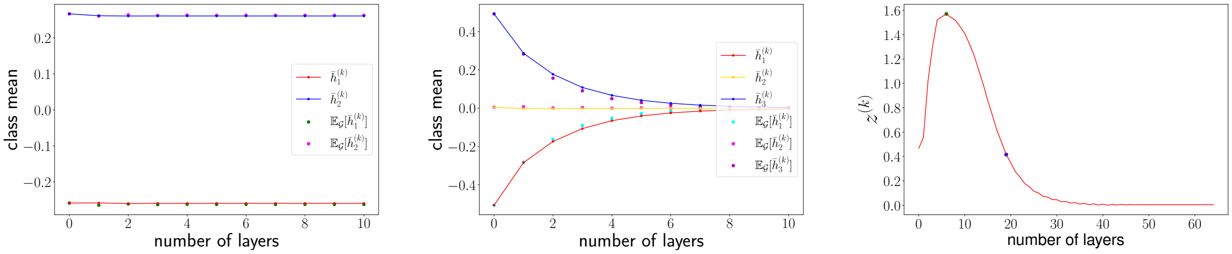
Theoretical result. With CSBM (Def. 2.3) and SMP (Eq. (3)), we present our theoretical finding: SMP is vulnerable to the over-smoothing problem.

Theorem 2.4 (Oversmoothing problem of SMP)**.** *Consider random variables $\mathcal{A}$ and $\mathbf{X}$ from CSBM (Def. 2.3), and recall that $\mathcal{A}$ is always desirable. Let $\mathcal{V}_c := \{v_j \in \mathcal{V} : y_j = c\}$ be the set of nodes belonging to each class $c \in [C]$, and let $\bar{\mathbf{h}}_c^{(K)} := \frac{C}{N} \sum_{v_i \in \mathcal{V}_c} \mathbf{h}_i^{(K)}$ be the mean of the K-th layer embeddings of $\mathcal{V}_c$. Then, for any two classes $a, b \in [C]$,*

$$\mathbb{E}_{\mathcal{G}}[\|\bar{\mathbf{h}}_a^{(K)} - \bar{\mathbf{h}}_b^{(K)}\|] = \left(\frac{p+q}{p+(C-1)q}\right)^K \|\mathbf{u}_a - \mathbf{u}_b\|. \tag{4}$$

Refer to Appendix B.1 for a concentration bound of $\|(\bar{\mathbf{h}}_a^{(K)} - \bar{\mathbf{h}}_b^{(K)}) - \mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(K)} - \bar{\mathbf{h}}_b^{(K)}]\|$. Theorem 2.4 states that, even when $\mathcal{A}$ is desirable, if $C > 2$ (i.e., multi-class), the expected distance between the average embeddings of any pair of classes can decrease exponentially as the number of layers K increases. That is, the separability of different class embeddings significantly decreases as more layers are stacked, revealing SMP's susceptibility to oversmoothing.

Simulation result. Fig. 1 shows the simulation results for binary-class and multi-class ($C = 3$) cases with 20 pairs of $\mathcal{A}$ and $\mathbf{X}$ from CSBM (Def. 2.3). For the binary case, we set $\mathbf{u}_1 = -0.25$, $\mathbf{u}_2 = 0.25$, and for the multi-class case, we set $\mathbf{u}_1 = -0.5$, $\mathbf{u}_2 = 0$, and $\mathbf{u}_3 = 0.5$. The other parameters are $N = 3000$, $p = 0.003$, and $q = 0.01$. One-dimensional features are used for illustrative purposes. We have several observations. (1) The expected and actual class means are close, validating that Theorem 2.4 is meaningful. (2) SMP



(a) binary case (b) multi-class ($C = 3$) case (c) score

Figure 1: Simulation results for (a) binary and (b) multi-class classification reveal that, as the number of layers increases, the mean embeddings of different classes converge in the multi-class case but not in the binary case. That is, SMP suffers from oversmoothing in the multi-class case, as also supported by (c) the drop of $z^{(k)}$, a score positively associated with the accuracy of the Bayes optimal classifier.

is robust to oversmoothing in the binary-class case, i.e., the difference between two class means does not decrease. (3) SMP suffers from oversmoothing in the multi-class case, with the differences between any two class means decreasing as the number of layers increases. (4) We further use a numerical score $z^{(k)}$ measuring the distinguishability between different classes, i.e., class 1 and class 2.[1] The score is positively correlated with the accuracy of the optimal Bayes classifier (Wu et al., 2022), where a higher value is desired, implying that the two classes are easier to distinguish. As the depth increases, we observe an initial rise in the score, followed by rapid decline and ultimate convergence to zero, illustrating the typical phenomenon of oversmoothing.

While our analysis of SMP's limitations is founded on linear models, these findings can be seamlessly applied to non-linear SMP models. See Appendix F for a discussion.

3. Multiset to Multiset Message Passing

In this section, we propose **M2M-GNN** (**M**ultiset **to** **M**ultiset GNN), our novel message-passing paradigm that provably mitigates the aforementioned limitations of SMP.

3.1. Motivation of M2M-GNN

Let us first recall the limitations we observed on SMP. First, a discrepancy arises between one-hop and multi-hop aggregation: even if we have a desirable one-hop propagation matrix $\mathcal{A}^{(k)}$s, the multi-hop propagation matrix $\mathcal{T}$ can still be undesirable (Theorem 2.2). Second, even with negative propagation coefficients between them, the embeddings of different classes can be mixed up, thereby leading to oversmoothing (Theorem 2.4). All proofs are in Appendix B.

Limitations of simple pooling approach. We point out that, even with desirable weights (or signs), the simple weighted summation scheme acts as the information bottleneck of SMP. Specifically, the weighted summation has a functional form of *multiset-to-element* (*m-2-e*). In the aggregation step, multiple elements (the embeddings of neighbors) are reduced (i.e., aggregated) into a single element (a single message vector), and the embeddings of heterophilic nodes are thus mixed up. Formally, an *m-2-e* message passing function can be represented as $\mathbf{m}_i = \phi(\mathcal{S}_i)$, where $\mathcal{S}_i := \{\mathbf{h}_j : v_j \in \mathcal{N}(v_i)\}$,[2] and $\phi(\cdot)$ is a mapping function defined as follows.

Definition 3.1. A mapping function $\phi(\cdot)$ takes a set of vectors as input and produces a single element as output, by applying a learnable weight matrix to each element of the set, followed by element-wise pooling (sum, mean, and max) to aggregate all elements.

For instance, using sum pooling and learnable weight matrix $\mathbf{W}$ yields a mapping $\phi(\mathcal{S}_i) = \sum_{\mathbf{h}_j \in \mathcal{S}_i}(\mathbf{h}_j\mathbf{W})$.

Multiset to multiset: Partition, mapping, and concatenation. Then, what if we aggregate embeddings into multiple elements instead? In other words, we can take a functional form of *multiset-to-multiset* (*m-2-m*). Specifically, given a multiset of node embeddings,[3] an *m-2-m* scheme (1) partitions the multiset into multiple subsets, (2) maps the features in each subset into an element (e.g., a vector), and (3) combines the elements into a multiset.

Definition 3.2 (*m-2-m* schemes)**.** Given a multiset $\mathcal{X}$ where each $x_i \in \mathcal{X}$ is a vector $x_i \in \mathbb{R}^d$, an *m-2-m* scheme f does the following: (1) it partitions $\mathcal{X} = \bigcup_{t=1}^{\omega} \mathcal{X}_t$ into ω disjoint subsets for some $\omega \in \mathbb{N}$, (2) it applies a mapping $\phi\colon 2^{\mathcal{X}} \mapsto \mathbb{R}^{d'}$ on each subset $\mathcal{X}_t$ to obtain an element (a vector) $\mathbf{z}_t = \phi(\mathcal{X}_t)$ for each subset, and (3) the obtained elements essentially form a multiset, and it concatenates the obtained elements into $f(\mathcal{X}) = \mathbf{z} = \|_{t=1}^{\omega}\mathbf{z}_t$, where $\|$ denotes the vector concatenation operator.

Desirable *m-2-m* schemes. There can be different *m-2-m* schemes, e.g., we have diverse ways of partitioning a given multiset $\mathcal{X}$. What kinds of *m-2-m* schemes are *desirable*?

[1] The score is defined as $z^{(k)} = (\bar{\mathbf{h}}_2^{(k)} - \bar{\mathbf{h}}_1^{(k)})/\sigma^{(k)}, \sigma^{(k)} = (\sigma_1^{(k)} + \sigma_2^{(k)})/2$, where $\sigma_1^{(k)}$ and $\sigma_2^{(k)}$ denote the variances of features from class 1 and class 2, respectively.

[2] Precisely, $\mathcal{S}_i^k$ consists of $k-1$ layer outputs of v_i neighbors. For simplicity, superscripts denoting layers are omitted.

[3] It is a multiset because different nodes may have identical embeddings, and each of them should be kept.

Recall that the limitations of SMP primarily arise from the intermixing of various classes. Hence, our high-level idea is to ensure that heterophilic node representations are not intertwined, i.e., when we partition $\mathcal{X}$, each $\mathcal{X}_k$ only consists of the embeddings of nodes from the same class. Specifically, we define *desirable m-2-m schemes* as follows.

Definition 3.3 (Desirable *m-2-m* schemes)**.** Assume for any multiset $\mathcal{X}$, each vector $x_i \in \mathcal{X}$ associated with a class y_i, we say an *m-2-m* scheme is *desirable*, iff, for any given $\mathcal{X}$, it always partitions $\mathcal{X}$ according to the class of x_i's, i.e., it partitions $\mathcal{X} = \bigcup_{t=1}^{\omega} \mathcal{X}_t$ such that, if two vectors $x_i, x_j \in \mathcal{X}$ are in the same $\mathcal{X}_t$, then we must have $y_i = y_j$. Note that we do *not* necessarily have x_i, x_j in the same $\mathcal{X}_t$ if $y_i = y_j$.

Based on the concept of desirable *m-2-m* schemes, we further define *desirable m-2-m message passing*.

Definition 3.4 (Desirable *m-2-m* message passing)**.** An *m-2-m message passing* operation f_{mp} is *desirable*, if given any node embedding matrix $\mathbf{H}$ and any node v_i, there exists a mapping $\phi(\cdot)$ such that the message vector of v_i can be represented as

$$f_{mp}(\mathbf{H}, \phi(\cdot); i) = \mathbf{m}_i = \|_{t=1}^{\omega} \phi(\mathcal{S}_{i,t}), \tag{5}$$

where (1) the union of $\mathcal{S}_{i,t}$'s is a subset of $\mathbf{H}$, i.e., $\bigcup_{t=1}^{\omega} \mathcal{S}_{i,t} \subseteq \{\mathbf{h}_j : j \in [N]\}$, and (2) each $\mathcal{S}_{i,t}$ contains embeddings of nodes from the same class,[4] i.e., if $\mathbf{h}_j, \mathbf{h}_{j'} \in \mathcal{S}_{i,t}$ then $y_j = y_{j'}$.

Moreover, we say f_{mp} is a ***one-hop*** *desirable m-2-m* message passing operation if there exists a mapping function $\phi(\cdot)$ satisfying

$$f_{mp}(\mathbf{H}, \phi(\cdot); i) = \mathbf{m}_i = \|_{t=1}^{C} \phi(\mathcal{S}_{i,t}), \tag{6}$$

where $\mathcal{S}_{i,t} \coloneqq \{\mathbf{h}_j : y_j = t, v_j \in \mathcal{N}(v_i)\}$.

Desirable *m-2-m* message passing alleviates the limitations. Below, we shall analyze several theoretical properties of *m-2-m* message passing and how it mitigates the limitations of SMP (i.e., *m-2-e*) discussed in Section 2.

First, regarding desirable properties, we claim that stacking one-hop *desirable m-2-m* message passing operations always gives us a desirable (multi-hop) message passing. Recall that this does not hold for SMP (Theorem 2.2).

Lemma 3.5 (Maintenance of desirable property)**.** *Assuming a **one-hop** desirable m-2-m message passing $f_{mp}^{(k)}$ is applied to each node at each layer k. Specifically, for each layer k, the embedding of each node v_i at the k-th layer is $\mathbf{h}_i^{(k)} = f_{mp}^{(k)}(\mathbf{H}^{(k-1)}, \phi^{(k)}(\cdot); i) = \|_{t=1}^{C} \phi^{(k)}(\mathcal{S}_{i,t}^{(k)})$, where $\mathcal{S}_{i,t}^{(k)} = \{\mathbf{h}_j^{(k-1)} : v_j \in \mathcal{N}(v_i), y_j = t\}$. Then, for any $d, k \in \mathbb{N}$, the d-hop message passing operation at the k-th layer $\rho^{(d,k)}$ defined by $\rho^{(d,k)}(\mathbf{H}^{(k)}, \Phi^{(k+d)}(\cdot); i) = \mathbf{h}_i^{(k+d)}$, is desirable. Specifically, $\rho^{(d,k)}$ stacks d one-hop message passing operations from $f_{mp}^{(k+1)}$ to $f_{mp}^{(k+d)}$, and it partitions the layer-k embeddings of the d-hop neighbors of each node, where each group contains the embeddings of d-hop neighbors in the same class.*

Lemma 3.5 demonstrates that unlike SMP (*m-2-e*), the proposed *m-2-m* message passing can generalize the desirable property from local to global. This is because, following our high-level idea, at each layer, each group consists of embeddings from the same class, and thus heterophilic node representations are not intertwined even when we stack multiple message passing operations.

Second, regarding oversmoothing, we show that the concatenation utilized in desirable *m-2-m* message passing can enhance the discriminability among message vectors. We present informal statements that describe the robustness of *m-2-m* to oversmoothing phenomena, with the formal statements and proofs detailed in Appendix B.4.

Lemma 3.6 (*m-2-m* is always no worse than *m-2-e* (informal))**.** *Given any two embedding multisets, the distance between the two resulting message vectors of m-2-m is greater than or equal to the distance between those of m-2-e.*

Theorem 3.7 (*m-2-m* never gets trapped by oversmoothing (informal))**.** *Under the same CSBM settings as in Sec. 2, a one-hop m-2-m message passing (Eq. (6) in Def. 3.4) can escape from oversmoothing even when the expected class means converge to the same point, i.e., even if $\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k-1)}] = \mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k-1)}]$ we have $\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k)}] \neq \mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k)}]$ as long as $p \neq q$, while SMP fails to achieve this.*

We refer the reader to Appendix A for a discussion on the discriminative power of *m-2-m*.

3.2. Proposed Method: M2M-GNN

Motivated by the good properties of *m-2-m* message passing, we present a specific instance it, termed **M2M-GNN**. An illustration of M2M-GNN is presented in Figure 2.

Core idea: Chunks. Each partition is represented as a *chunk*. Ideally, a model should assign a label to each chunk while ensuring a one-to-one correspondence between chunk labels and node classes, i.e., each chunk should only contain nodes belonging to the class associated with that chunk.

Method details. We elaborate on how M2M-GNN updates node embeddings using $\mathcal{C}$ chunks. Note that $\mathcal{C}$ may not necessarily be the same as C, which represents the number of classes. We focus on the embedding update at the k-th layer. For a given $(k-1)$-th layer output node embeddings $\mathbf{H}^{(k-1)} \in \mathbb{R}^{N \times d}$, M2M-GNN first projects the node embed-

[4] Or $\mathcal{S}_{i,t} = \emptyset$, and we let $\phi(\mathcal{S}_{i,t}) = \mathbf{0}$, so that all the $\mathbf{m}_i$'s have the same dimension.

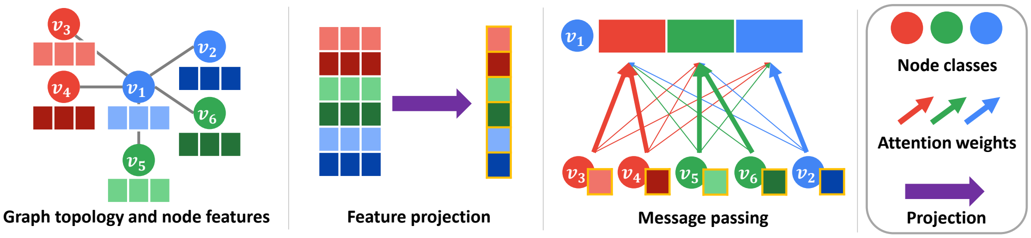


Figure 2: An illustration of the convolution layer of M2M-GNN, where the ego (central) node is v_1. For a given graph topology and node features (left), we first project node features from 3-dimension to 1-dimension (middle). Then, we construct a message vector for node v_1 by concatenating 3 chunks, each of which consists of a subset of neighbors' projected features using soft labels

.

dings using a learnable weight matrix $\mathbf{W}^{(k)} \in \mathbb{R}^{d\times(d/\mathcal{C})}$. Formally, $\hat{\mathbf{H}}^{(k)} = \mathbf{H}^{(k-1)}\mathbf{W}^{(k)}$.

Next, we describe the message passing scheme of M2M-GNN. We define v_i as an ego (central) node and $\mathcal{N}(v_i) := N_i$ as the set of neighbors of v_i. Since in practice, we typically do not have the ground-truth class information of nodes, we propose to utilize soft labels that are obtained via an attention function. We use the soft labels to assign each node (and its embedding) to chunks. In a nutshell, every node embedding of N_i is assigned to every chunk, but with different scores (i.e., weights). To learn the assigned scores, we employ the attention function inspired by Brody et al. (2021). Specifically, for $v_j \in \mathcal{N}_i$, its scores towards v_i are defined as follows:

$$\mathbf{s}^{(k)}(i,j) = \texttt{Softmax}(\texttt{ReLU}(\alpha\hat{\mathbf{h}}_i^{(k)} + \hat{\mathbf{h}}_j^{(k)})\mathbf{W}_{att}^{(k)}/\tau), \tag{7}$$

where τ is a temperature hyperparameter, $0 < \alpha < 1$ is a hyperparameter that weights the importance of the ego node v_i's representation, and $\mathbf{W}_{att}^{(k)} \in \mathbb{R}^{(d/\mathcal{C})\times\mathcal{C}}$ is the learnable attention weight. Note that $\mathbf{s}^{(k)}(i,j) \in \mathbb{R}^{\mathcal{C}}$ indicates scores of v_j during message passing towards v_i, where $\mathbf{s}_t^{(k)}(i,j)$ is the score assigned to the t-th chunk. Moreover, these scores can be interpreted as a soft label of v_j, and in the ideal case, M2M-GNN assigns a high score to the chunk that corresponds to the class of v_j, and nearly zero values to other chunks. We examine this property in Section 4.2.

$$\mathbf{C}_{it}^{(k)} = \sum_{v_j\in N_i} \mathbf{s}_t^{(k)}(i,j)\hat{\mathbf{h}}_j^{(k)} \in \mathbb{R}^{d/\mathcal{C}}, \forall t \in [\mathcal{C}]. \tag{8}$$

Then, the message vector of v_i, denoted by $\mathbf{m}_i^{(k)}$, is defined as the concatenation of $\mathbf{C}_{i1}^{(k)}, \cdots, \mathbf{C}_{i\mathcal{C}}^{(k)}$, i.e., $\mathbf{m}_i^{(k)} = \|_{t=1}^{\mathcal{C}}\mathbf{C}_{it}^{(k)}$. The dimension of $\mathbf{m}_i^{(k)}$ is equal to that of $\mathbf{h}_i^{(k-1)}$, which is $\mathbb{R}^d$.

Finally, the node representation of v_i is updated using the following weighted average:

$$\mathbf{h}_i^{(k)} = \texttt{ReLU}((1-\beta)\mathbf{h}_i^{(0)} + \beta\mathbf{m}_i^{(k)}), \tag{9}$$

where $\beta \in (0,1)$ is a hyperparameter, and $\mathbf{h}_k^{(0)}$ is the transformed original feature of v_i (i.e., $\mathbf{h}_i^{(0)} = \texttt{MLP}(\mathbf{x}_i)$).

Regularizing soft labels. In our preliminary study, we observed that every attention score is often collapsed into one particular chunk (i.e., there exists a $1 \leq t \leq \mathcal{C}$ such that $s_t^{(k)}(i,j) \approx 1$ for all edges $\{v_i, v_j\}$). This deviates from a desirable scenario, where attention scores of nodes are diversified to different chunks according to their respective classes. Motivated by Tsitsulin et al. (2023), we leverage a regularization term to help M2M-GNN diversify the attention scores. Unlike Tsitsulin et al. (2023), our objective does not involve balancing class sizes as this loss affects the attention weights rather than the predictions. Specifically, we employ L2-norm of the summation of $\mathbf{s}_t^{(k)}(i,j)$ across $\{v_i, v_j\} \in \mathcal{E}$. Formally,

$$\mathcal{L}_{reg} = \frac{1}{K}\sum_{k=1}^{K}\frac{\sqrt{\mathcal{C}}}{|\mathcal{E}|}\|\sum_{\{v_i,v_j\}\in\mathcal{E}} \mathbf{s}^{(k)}(i,j)\|_2^2 - 1, \tag{10}$$

where K is the number of layers of M2M-GNN. Note that the value of $\mathcal{L}_{reg}$ decreases as the distribution of $\mathbf{s}^{(k)}(i,j)$ varies across $\forall\{v_i, v_j\} \in \mathcal{E}$.

Training M2M-GNN. Finally, the parameters of K-layer M2M-GNN (spec., initial feature projector $\texttt{MLP}$, projection- and attention weights of each layer $\mathbf{W}^{(k)}, \mathbf{W}_{att}^{(k)}, \forall k \in [K]$) are trained via (1) the regularization loss $\mathcal{L}_{reg}$ (Eq. (10)) and (2) the loss designed for the particular learning task (e.g., node classification), which we denote as $\mathcal{L}_{task}$. Specifically, we use the loss function: $\mathcal{L} = \mathcal{L}_{task} + \lambda\mathcal{L}_{reg}$, where λ is a hyperparameter that controls the strength of the regularization term $\mathcal{L}_{reg}$.

The time complexity of M2M-GNN is analyzed in Appendix E, showing that it exhibits a time complexity similar to that of GCN (Kipf & Welling, 2016).

4. Experiments

In this section, we conduct a comprehensive evaluation of M2M-GNN on various benchmark datasets, including both homophilic and heterophilic graphs. We aim to answer four key research questions: (**RQ1**) How does M2M-GNN perform in node classification tasks? (**RQ2**) What patterns emerge in the attention scores, $\mathbf{s}^{(k)}(i,j)$'s? (**RQ3**) How ef-

	Edge Hom.	# Nodes	# Edges	# Features	# Classes
Texas	0.21	183	295	1,703	5
Wisconsin	0.11	251	466	1,703	5
Cornell	0.30	183	280	1,703	5
Actor	0.22	7,600	26,752	931	5
Squirrel	0.22	5,201	198,493	2,089	5
Chameleon	0.23	2,277	31,421	2,325	5
Cora	0.81	2,708	5,278	1,433	7
Citeseer	0.74	3,327	4,676	3,703	6
Pubmed	0.80	19,717	44,327	500	3
Penn94	0.47	41,554	1,362,229	5	2
Amazon-rat.	0.38	24,492	93,050	300	5

Table 1: Dataset statistics. The edge homophily ratio (Edge Hom.) is defined as $|\{(v_i, v_j) \in \mathcal{E}\} : y_i = y_j|/|\mathcal{E}|$.

fectively does M2M-GNN mitigate oversmoothing? (**RQ4**) Are all components of M2M-GNN necessary and impactful?

Datasets. We use 11 widely-used node classification benchmark datasets, where 8 are heterophilic (Texas, Wisconsin, Cornell, Actor, Squirrel, Chameleon, Penn94, and Amazon-rating), and the remaining 3 are homophilic (Cora, Citeseer, and Pubmed). In line with prior research, for training/validation/test splits, we employ the setting provided by Pytorch Geometric (PyG) (Platonov et al., 2022). For details of the datasets, including their sources and construction methods, refer to Appendix C.

Baseline models. We compare the performance of M2M-GNN against the following 12 baseline models:

1. **Classic models**: GCN (Kipf & Welling, 2016), GAT (Veličković et al., 2018), and MLP,
2. **SMP-based GNNs**: ACM-GCN (Luan et al., 2022), GPRGNN (Chien et al., 2020), FAGCN (Bo et al., 2021), GGCN (Yan et al., 2022), Goal (Zheng et al., 2023), and AERO-GNN (Lee et al., 2023),
3. **Concatenation-based GNNs**: Ordered GNN (Song et al., 2022) and H$_2$GCN (Zhu et al., 2020),
4. **Attention-vector-based GNN**: DMP (Yang et al., 2021).

For details of them, refer to Appendix C.

4.1. RQ1: Node Classification Performance Results

In Table 2, we provide the mean node classification accuracy for test nodes, along with their corresponding standard deviations, across 10 random data splits.

In summary, M2M-GNN achieves the best average ranking among all the methods, securing a position in the top 3 across all datasets. The following three points stand out.

First, M2M-GNN outperforms all SMP-based GNNs. This observation serves as empirical support for our theoretical finding, indicating that the limitations of SMP (Sec. 2) can adversely impact its effectiveness. Conversely, the theoretical properties of *m-2-m* (Sec. 3.1) contribute to the enhanced classification accuracy of M2M-GNN.

Second, M2M-GNN also outperforms DMP, an attention-vector-based GNN: This again demonstrates the superiority of *m-2-m* over *m-2-e*. DMP is a variant of SMP assigning distinct coefficients to each channel of node representations. Consequently, while employing weight vectors, DMP still follows the *m-2-e* paradigm in its message passing scheme.

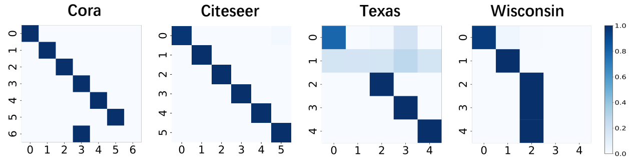


Figure 3: Visualization results of $\mathcal{S}$ for two homophilic (left two) and two heterophilic (right two) graphs. (Off-)Diagonal entries indicate (in)correct prediction.

Lastly, M2M-GNN outperforms concatenation-based GNNs. While H$_2$GCN and Ordered GNN employ concatenation to combine representations from different hops, they fail to address the mixing of features of heterophilic neighbors within the same hop. This validates the effectiveness of the novel concatenation design used in *m-2-m*.

4.2. RQ2: Attention Weight Analysis

We analyze the distribution of learned attention weights $\mathbf{s}^{(k)}(i,j)$'s. To this end, we consider directed graphs. In the case of an undirected graph, we convert it into a directed graph by splitting each undirected edge into two directed edges. For an edge from v_i to v_j, v_i and v_j are referred to as the source and target nodes of the edge, respectively.

As described in Section 3.2, the learned attention weight $\mathbf{s}^{(k)}(i,j)$ corresponds to the soft label of target node v_j. Let $\bar{\mathbf{S}} \in \mathbb{R}^{|\mathcal{E}| \times \mathcal{C}}$ be the average value across all layers, calculated as $\frac{1}{K}\sum_{k=1}^{K} \mathbf{S}^{(k)}$, where $\mathbf{S}^{(k)}$ is the matrix formed by all $\mathbf{s}^{(k)}(i,j) \in \mathbb{R}^{\mathcal{C}}$,$\{v_i, v_j\} \in \mathcal{E}$. To make a comparison, we also consider the ground truth edge labels $\hat{\mathbf{S}} \in \mathbb{R}^{|\mathcal{E}| \times \mathcal{C}}$, where $\hat{\mathbf{S}}_{ic} = 1$ if the target node of the i-th edge belongs to class c. A learned $\bar{\mathbf{S}}$ is considered accurate if, upon proper column index reordering, it is similar to $\hat{\mathbf{S}}$. We reorder the columns of $\bar{\mathbf{S}}$ to ensure their values are properly aligned to the order of class. Then we compute the matrix $\mathcal{S} = \hat{\mathbf{S}}^T \bar{\mathbf{S}} \in \mathbb{R}^{\mathcal{C} \times \mathcal{C}}$, followed by a row-wise softmax.

In the definition of $\mathcal{S}$, large diagonal entries imply predictions closely align with the ground truth, thus considered correct. As shown in Fig. 3, the learned $\bar{\mathbf{S}}$ and the ground truth $\hat{\mathbf{S}}$ are well matched. Specifically, 19 out of the 23 columns (each column corresponds to one class, e.g., Cora has 6 classes of nodes) of $\mathcal{S}$ are verified to be correct, (i.e., the diagonal entries are the largest in the columns), with a particularly high accuracy observed for the homophilic graphs where nearly all edges are correctly identified.

	Texas	Wiscon.	Cornell	Actor	Squir.	Chamel.	Cora	Citeseer	Pubmed	Penn94	Amazon-rat.	A.R.
MLP	80.81 ± 4.7	85.29 ± 3.3	81.89 ± 6.4	36.53 ± 0.7	28.77 ± 1.5	46.21 ± 2.9	75.69 ± 2.0	74.02 ± 1.9	87.16 ± 0.3	73.61 ± 0.4	42.87 ± 0.4	9.9
GCN	55.14 ± 5.1	51.76 ± 3.0	60.54 ± 5.3	27.32 ± 1.1	53.43 ± 2.0	64.82 ± 2.2	86.98 ± 1.2	76.50 ± 1.3	88.42 ± 0.5	82.47 ± 0.2	48.70 ± 0.6	9.1
GAT	52.16 ± 6.6	49.41 ± 4.0	61.89 ± 5.0	27.44 ± 0.8	40.72 ± 1.5	60.26 ± 2.5	87.30 ± 1.1	76.55 ± 1.2	86.33 ± 0.4	81.53 ± 0.5	49.09 ± 0.6	9.9
GPR-GNN	78.38 ± 4.3	82.94 ± 4.2	80.27 ± 8.1	34.63 ± 1.2	31.61 ± 1.2	46.58 ± 1.7	87.95 ± 1.1	77.13 ± 1.6	87.54 ± 0.3	81.38 ± 0.1	44.88 ± 0.3	8.6
FAGCN	77.00 ± 7.7	78.32 ± 6.3	82.41 ± 3.8	35.67 ± 0.9	42.20 ± 1.8	60.98 ± 2.3	87.42 ± 2.1	76.35 ± 1.7	87.83 ± 1.1	72.85 ± 0.5	44.12 ± 0.3	9.1
GGCN	84.86 ± 4.5	86.86 ± 3.2	85.68 ± 6.6	37.54 ± 1.5	55.17 ± 1.5	71.14 ± 1.8	87.95 ± 1.0	77.14 ± 1.4	89.15 ± 0.3	OOM	36.86 ± 0.4	6.0
ACM-GCN	87.84 ± 4.4	88.43 ± 3.2	85.14 ± 6.0	36.28 ± 1.0	54.40 ± 1.8	66.93 ± 1.8	87.91 ± 0.9	77.32 ± 1.7	90.00 ± 0.5	82.52 ± 0.9	38.62 ± 0.6	4.6
Goal	83.62 ± 6.7	86.98 ± 4.4	85.68 ± 6.2	36.46 ± 1.0	60.53 ± 1.6	71.65 ± 1.6	88.75 ± 0.8	77.15 ± 0.9	89.25 ± 0.5	84.18 ± 0.3	37.94 ± 0.3	4.7
AERO-GNN	84.35 ± 5.2	81.24 ± 6.8	84.80 ± 3.3	36.57 ± 1.1	61.76 ± 2.4	71.58 ± 2.4	88.12 ± 1.1	77.08 ± 1.5	89.95 ± 0.7	82.47 ± 0.7	45.71 ± 0.5	5.1
Ord. GNN	86.22 ± 4.1	88.04 ± 3.6	87.03 ± 4.7	37.99 ± 1.0	62.44 ± 1.9	72.28 ± 2.2	88.37 ± 0.7	77.31 ± 1.7	90.15 ± 0.3	83.65 ± 0.6	38.52 ± 0.4	2.7
H_2GCN	84.86 ± 7.2	87.65 ± 4.9	82.70 ± 5.2	35.70 ± 1.0	36.48 ± 1.8	60.11 ± 2.1	87.87 ± 1.2	77.11 ± 1.5	89.49 ± 0.3	81.31 ± 0.6	36.47 ± 0.2	7.8
DMP	66.08 ± 7.0	56.41 ± 5.5	62.73 ± 4.5	28.30 ± 2.7	34.19 ± 7.6	63.79 ± 4.1	82.56 ± 1.9	62.54 ± 1.5	73.12 ± 0.9	73.85 ± 0.7	35.84 ± 0.4	11.2
M2M-GNN	89.19 ± 4.5	89.01 ± 4.1	86.48 ± 6.1	36.72 ± 1.6	63.60 ± 1.7	75.20 ± 2.3	88.12 ± 1.0	77.20 ± 1.8	90.35 ± 0.6	85.94 ± 0.4	49.18 ± 0.6	1.7

Table 2: Node classification accuracy (%) on 11 datasets. The best, second-best, and third-best performance across each dataset are highlighted in blue, green, and yellow, respectively. A.R. and OOM denote average ranking and out-of-memory, respectively. Overall, our method **M2M-GNN** performs the best in terms of average ranking, achieving at least the third-best performance in every dataset.

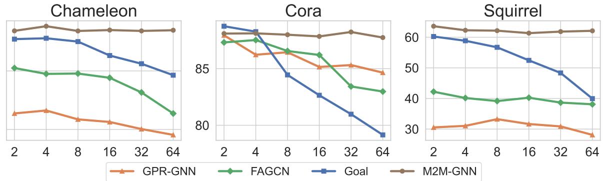


Figure 4: Performance comparison of M2M-GNN against SMP-based GNNs under various model depths. The X-axis has the number of layers, and the Y-axis has node classification accuracy.

4.3. RQ3: Over-smoothing Analysis

We evaluate the robustness of M2M-GNN against the over-smoothing problem by assessing its performance at different model depths ($K \in \{2, 4, 8, 16, 32, 64\}$). To provide a comparative analysis against other GNNs, we also evaluate three SMP-based GNNs: FAGCN, Goal, and GPR-GNN. We evaluate the performances of these methods on two heterophilic graphs and one homophilic graph.

The analysis presented in Fig. 4 illustrates that the performance of M2M-GNN remains relatively stable with varying numbers of layers, achieving its best performance when deeper layers are employed ($K = 32$) in the Cora dataset. In contrast, the SMP-based GNNs utilized in the experiment exhibit a substantial decrease in performance as the number of layers increases, indicating their empirical susceptibility to the oversmoothing problem.

4.4. RQ4: M2M-GNN Component Analysis

We examine the following: (1) the necessity of regularization loss $\mathcal{L}_{reg}$ (see Eq. (10)) and (2) the impact of the number of chunks $\mathcal{C}$. Specifically, we assess the performance of M2M-GNN with/without $\mathcal{L}_{reg}$ and across various values of $\mathcal{C}$. We focus on evaluating (1) the discriminative power of soft labels, (2) node classification accuracy, and (3) robustness against oversmoothing.

To this end, we use three metrics: (1) mixing score, which is defined as the proportion of heterophilic edges $\{v_i, v_j\}$ such that the argmax chunk is different for $(v_i \to v_j)-$ and $(v_j \to v_i)-$message passing, (2) node classification accuracy at the best hyperparameter setting for each, and (3) node classification accuracy at $K = 32$. Across all metrics, a higher value indicates better model performance. We utilized one heterophilic dataset (Wisconsin) and one homophilic dataset (Pubmed).

We present results in Fig. 5, where each tuple on the X-axis represents a pair of parameters $(\mathcal{C}, \lambda)$. Recall that λ is a scalar for $\mathcal{L}_{reg}$, and $\lambda = 0$ indicates the absence of the regularization loss. First, $\mathcal{L}_{reg}$ is demonstrated to be necessary, given that the models with $\mathcal{L}_{reg} \neq 0$ outperforms that with $\mathcal{L}_{reg} = 0$ across all metrics and datasets. Second, setting the number of chunks and classes equal leads to the best performance. While increasing $\mathcal{C}$ may help discriminate heterophilic node representations, this may cause excessive separation, which is harmful to smoothing homophilic node representations, leading to suboptimal node embeddings.

5. Related Work

Analysis of signed message passing. The theoretical analysis by Yan et al. (2022) explores the impact of signed messages in GNNs under a binary-class setting, introducing the desirable signs for homophilic and heterophilic edges, respectively. Choi et al. (2023) extend the theoretical framework to a multi-class setting, analyzing how node embeddings are updated in multi-class `SMP`.

Choi et al. (2023) investigate the impact of the error rate on performance by assigning positive and negative coefficients to heterophilic and homophilic edges, respectively. However, they do not discuss the suboptimality of `SMP` for GNNs with multiple layers, even when the error rate is zero (i.e., desirable).

Liang et al. (2023) reveal that for a triad in a graph (i.e., three connected nodes), their embedding update can be undesirable even when they are connected by edges with proper signs. However, they overlook the influence of other nodes

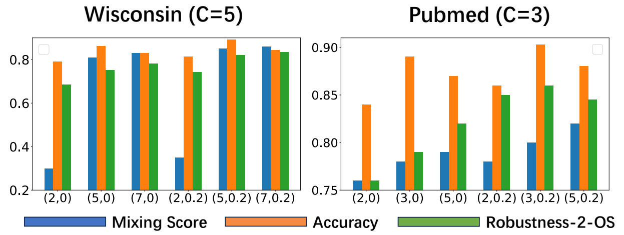


Figure 5: Ablation study. Each tuple on the X-axis indicates a pair of parameters: (number of chunks C, strength of regularization λ). For all the metrics, higher values indicate better performance.

and do not present a solution to mitigate this limitation, instead sidestepping it by employing a one-layer GNN.

Oversmoothing analysis. Among many studies of oversmoothing (Oono & Suzuki, 2019; Zhou et al., 2021; Keriven, 2022; Wu et al., 2023; 2022; Lee et al., 2023; Liang et al., 2024), we focus on two most related ones. Wu et al. (2022) present a non-asymptotic analysis of the oversmoothing problem on binary-class graphs, focusing on positive propagation coefficient settings. Lee et al. (2023) study oversmoothing phenomena in attention-based methods under asymptotic cases where the number of model layers goes infinity.

Unlike (Wu et al., 2022), we focus on multi-class settings with signed message passing, and compared to (Lee et al., 2023), we provide non-asymptotic results.

Concatenation-based GNNs. To avoid undesirable feature smoothing, many concatenation-based GNNs have been proposed (Hamilton et al., 2017; Xu et al., 2018b; Zhu et al., 2020; Song et al., 2022). They mainly (1) concatenate ego node features and neighbor features and/or (2) concatenate node features at different distances.

Note that LW-GCN (Dai et al., 2022) also focuses on separating node features from various classes. However, the authors do not investigate the constraints of SMP or establish theoretical properties for addressing these constraints. In contrast to LW-GCN's goal of accurately mapping nodes to chunks according to their labels, M2M-GNN adopts a more practical strategy by assigning nodes with different labels to separate chunks.

6. Conclusion

In this paper, we reveal two unexpected challenges associated with signed message passing (SMP): (a) undesirable embedding update and (b) vulnerability to oversmoothing. To mitigate these challenges, we develop a novel multiset-to-multiset (m-2-m) message passing scheme and prove several desirable theoretical properties of it. Motivated by the properties, we propose M2M-GNN, a novel GNN based on the m-2-m scheme. Through our comprehensive experiments, we demonstrate the superiority of M2M-GNN over existing SMP-based GNNs in node classification tasks.

Acknowledgements

We want to thank all the anonymous reviewers for their constructive comments. The authors also thank Fanchen Bu for his insightful feedback on Section 3. This work was partially supported by the Major Key Project of PCL (No. 2022ZD0115301), and an Open Research Project of Zhejiang Lab (NO.2022RC0AB04).

Impact Statement

This paper presents work whose goal is to advance graph representation learning with heterophily, a prevalent trend in real-world connections observed in domains like chemical compounds and social networks.

The effectiveness of our model may be compromised by inaccurate node labels, as labeled nodes are crucial for providing supervision signals during model training, potentially leading to adverse implications. For instance, utilizing a misled model to detect fraud within a trading network could introduce misclassifications and bias against legitimate users. To mitigate the vulnerability to noisy labels, established techniques can be integrated into our methodology. Additionally, it is essential to formulate specific ethical guidelines and impose constraints on the application of our model based on practical considerations.

Our model generates node representations using node attributes and linear transformation. Within the context of a social network represented as a graph, these attributes may encompass sensitive personal data like age and gender. To improve fairness, it may be helpful to incorporate the notion of counterfactual fairness, which emphasizes that a decision made about an individual is considered fair if altering the individual's sensitive attribute does not influence the decision outcome.

Unlike graph attention networks, which explicitly learn attention weights for each neighbor, our model makes it challenging to quantify the individual impact of each neighbor. Consequently, the interpretability of predictions in our model is not as straightforward as in the graph attention network. In order to enhance the transparency of predictions, existing graph explainers can be employed. For example, when focusing on a specific node v_i, masking individual edges allows us to pinpoint which edge exerts the most significant influence on its prediction. Analogous techniques can be applied to a particular node or attribute.

References

Atay, F. M. and Tuncel, H. On the spectrum of the normalized laplacian for signed graphs: Interlacing, contraction, and replication. *Linear Algebra and its Applications*, 442: 165–177, 2014.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Bo, D., Wang, X., Shi, C., and Shen, H. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 3950–3957, 2021.

Brody, S., Alon, U., and Yahav, E. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2021.

Chien, E., Peng, J., Li, P., and Milenkovic, O. Adaptive universal generalized pagerank graph neural network. In *International Conference on Learning Representations*, 2020.

Choi, Y., Choi, J., Ko, T., and Kim, C.-K. Is signed message essential for graph neural networks? *arXiv preprint arXiv:2301.08918*, 2023.

Chung, F. and Lu, L. Concentration inequalities and martingale inequalities: a survey. *Internet mathematics*, 3(1), 2006.

Dai, E., Zhou, S., Guo, Z., and Wang, S. Label-wise graph convolutional network for heterophilic graphs. In *Learning on Graphs Conference*, pp. 26–1. PMLR, 2022.

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

Deshpande, Y., Sen, S., Montanari, A., and Mossel, E. Contextual stochastic block models. *Advances in Neural Information Processing Systems*, 31, 2018.

Fey, M. and Lenssen, J. E. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30:1024–1034, 2017.

Keriven, N. Not too little, not too much: a theoretical analysis of graph (over) smoothing. *Advances in Neural Information Processing Systems*, 35:2268–2281, 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2016.

Lee, S. Y., Bu, F., Yoo, J., and Shin, K. Towards deep attention in graph neural networks: Problems and remedies. In *International Conference on Machine Learning*, volume 202, pp. 18774–18795. PMLR, 2023.

Lee, S. Y., Kim, S., Bu, F., Yoo, J., Tang, J., and Shin, K. Feature distribution on graph topology mediates the effect of graph convolution: Homophily perspective. *CoRR*, abs/2402.04621, 2024.

Li, Q., Han, Z., and Wu, X.-M. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Liang, L., Hu, X., Xu, Z., Song, Z., and King, I. Predicting global label relationship matrix for graph neural networks under heterophily. *Advances in Neural Information Processing Systems*, 36:10909–10921, 2023.

Liang, L., Xu, Z., Song, Z., King, I., Qi, Y., and Ye, J. Tackling long-tailed distribution issue in graph neural networks via normalization. *IEEE Trans. Knowl. Data Eng.*, 36(5):2213–2223, 2024.

Lim, D., Hohne, F., Li, X., Huang, S. L., Gupta, V., Bhalerao, O., and Lim, S. N. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. *Advances in Neural Information Processing Systems*, 34:20887–20902, 2021.

Lin, X., Quan, Z., Wang, Z.-J., Ma, T., and Zeng, X. Kgnn: Knowledge graph neural network for drug-drug interaction prediction. In *IJCAI*, pp. 2739–2745, 2020.

Luan, S., Hua, C., Lu, Q., Zhu, J., Zhao, M., Zhang, S., Chang, X.-W., and Precup, D. Revisiting heterophily for graph neural networks. *Advances in neural information processing systems*, 35:1362–1375, 2022.

Oliveira, R. I. Concentration of the adjacency matrix and of the laplacian in random graphs with independent edges. *arXiv preprint arXiv:0911.0600*, 2009.

Oono, K. and Suzuki, T. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2019.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Pei, H., Wei, B., Chang, K. C.-C., Lei, Y., and Yang, B. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*, 2020.

Platonov, O., Kuznedelev, D., Diskin, M., Babenko, A., and Prokhorenkova, L. A critical look at the evaluation of gnns under heterophily: Are we really making progress? In *International Conference on Learning Representations*, 2022.

Rozemberczki, B., Allen, C., and Sarkar, R. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021.

Song, Y., Zhou, C., Wang, X., and Lin, Z. Ordered gnn: Ordering message passing to deal with heterophily and over-smoothing. In *International Conference on Learning Representations*, 2022.

Tang, J., Hu, X., Gao, H., and Liu, H. Exploiting local and global social context for recommendation. In *IJCAI*, pp. 2712–2718, 2013.

Tsitsulin, A., Palowitch, J., Perozzi, B., and Müller, E. Graph clustering with graph neural networks. *Journal of Machine Learning Research*, 24(127):1–21, 2023.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Wu, X., Chen, Z., Wang, W. W., and Jadbabaie, A. A non-asymptotic analysis of oversmoothing in graph neural networks. In *International Conference on Learning Representations*, 2022.

Wu, X., Ajorlou, A., Wu, Z., and Jadbabaie, A. Demystifying oversmoothing in attention-based graph neural networks. *arXiv preprint arXiv:2305.16102*, 2023.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2018a.

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.-i., and Jegelka, S. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pp. 5453–5462. PMLR, 2018b.

Yan, Y., Hashemi, M., Swersky, K., Yang, Y., and Koutra, D. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. In *2022 IEEE International Conference on Data Mining (ICDM)*, pp. 1287–1292. IEEE, 2022.

Yang, L., Li, M., Liu, L., Wang, C., Cao, X., Guo, Y., et al. Diverse message passing for attribute with heterophily. *Advances in Neural Information Processing Systems*, 34: 4751–4763, 2021.

Yang, Z., Cohen, W. W., and Salakhutdinov, R. Revisiting semi-supervised learning with graph embeddings. In *International Conference on Machine Learning*, volume 48, pp. 40–48. PMLR, 2016.

Zheng, X., Liu, Y., Pan, S., Zhang, M., Jin, D., and Yu, P. S. Graph neural networks for graphs with heterophily: A survey. *CoRR*, abs/2202.07082, 2022.

Zheng, Y., Zhang, H., Lee, V. C., Zheng, Y., Wang, X., and Pan, S. Finding the missing-half: Graph complementary learning for homophily-prone and heterophily-prone graphs. In *International Conference on Machine Learning*, volume 202, pp. 42492–42505. PMLR, 2023.

Zhou, K., Huang, X., Zha, D., Chen, R., Li, L., Choi, S.-H., and Hu, X. Dirichlet energy constrained learning for deep graph neural networks. *Advances in Neural Information Processing Systems*, 34:21834–21846, 2021.

Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., and Koutra, D. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in neural information processing systems*, 33:7793–7804, 2020.

A. Discriminative Power of *m-2-m*

The work (Xu et al., 2018a) characterizes the discriminative power of GNNs by analyzing whether a GNN maps two different neighborhoods (i.e., two multisets) to the same representation.

m-2-m consistently yields higher discriminative power compared to *m-2-e*. The intuition behind this is that distinguishing between two multisets is typically expected to be easier compared to distinguishing between two single elements. This can be understood similarly to classifying multi-dimensional features versus classifying one-dimensional features: additional dimensions provide more evidence for accurate classification. Formally, we present the following result.

Lemma A.1. *Under the same setting as described in Theorem 3.6:*

(1) Any two distinct multisets that can be distinguished by m-2-e ($\mathbf{m}_a^{m2e} \neq \mathbf{m}_b^{m2e}$) can also be discriminated by m-2-m ($\mathbf{m}_a^{m2m} \neq \mathbf{m}_b^{m2m}$).

(2) There exist two different multisets such that m-2-m can differentiate, whereas m-2-e is unable to discriminate them: $\mathbf{m}_a^{m2e} = \mathbf{m}_b^{m2e}$.

Over-smoothing arises if two distinct multisets are mapped to the same embedding. The above result also suggests that *m-2-m* exhibits a higher level of robustness against over-smoothing compared to *m-2-e*.

B. Proof

B.1. Proof of Theorem 2.4 and The Concentration Bound

B.1.1. Proof of Theorem 2.4

Proof. Based on the definition of message passing scheme and CSBM defined in Sec. 2, the expectation $\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k)}]$ can be calculated as

$$\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k)}] = \frac{p\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k-1)}] - q\sum_{c\in[C],c\neq a}\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_c^{(k-1)}]}{p+(C-1)q}, \tag{11}$$

and similarly, we have

$$\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k)}] = \frac{p\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k-1)}] - q\sum_{c\in[C],c\neq a}\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_c^{(k-1)}]}{p+(C-1)q}. \tag{12}$$

By subtracting $\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k)}]$ from $\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k)}]$, we get

$$\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k)} - \bar{\mathbf{h}}_b^{(k)}] = \frac{p+q}{(p+(C-1)q)}\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k-1)} - \bar{\mathbf{h}}_b^{(k-1)}]. \tag{13}$$

Therefore, the sequence $(\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(0)} - \bar{\mathbf{h}}_b^{(0)}], \dots, \mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(K)} - \bar{\mathbf{h}}_b^{(K)}]$ is a geometric sequence. We write

$$\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(K)} - \bar{\mathbf{h}}_b^{(K)}] = (\frac{p+q}{p+(C-1)q})^K(\mathbf{u}_a - \mathbf{u}_b). \tag{14}$$

□

B.1.2. Concentration inequality

Let $\|\mathbf{M}\|$ denote the spectral norm of matrix $\mathbf{M}$. Next, we derive a concentration bound for $\bar{\mathbf{h}}_a^{(K)} - \bar{\mathbf{h}}_b^{(K)}$. We first present the Chernoff bound.

Lemma B.1. *(Chung & Lu, 2006) Let $\mathbf{X}_i \sim Bern(p_i)$ be independent and $\mathbf{Z} = \sum_{i=1}^n \mathbf{X}_i$, $\bar{\mathbf{Z}} = \mathbb{E}(\mathbf{Z})$. Then for any $\sigma > 0$,*

$$\mathbb{P}(\mathbf{Z} \geq (1+\sigma)\bar{\mathbf{Z}}) \leq e^{-\frac{\sigma^2}{2+\sigma}}\bar{\mathbf{Z}} \tag{15}$$

$$\mathbb{P}(\mathbf{Z} \leq (1-\sigma)\bar{\mathbf{Z}}) \leq e^{-\frac{\sigma^2}{2}}\bar{\mathbf{Z}} \tag{16}$$

Now we are prepared to establish the concentration inequality for node degrees.

Corollary B.2. *For any $\sigma > 0$ and $r > 0$, there exists a constant $\kappa(\sigma, r)$ such that when $\bar{d} \geq \kappa \log N$, the following holds with probability at least $1 - N^{-r}$,*

$$(1-\sigma)\bar{d} \leq d_i \leq (1+\sigma)\bar{d}, \quad \textit{for all } 1 \leq i \leq N.$$

Proof. For any $i \in [N]$, $d_i = \sum_{j=1}^{N} \mathbf{X}_j$, where $\mathbf{X}_j \sim Bern(p)$ if v_i and v_i share the same label and $\mathbf{X}_j \sim Bern(q)$ otherwise. Applying Lemma B.1, we get

$$\mathbb{P}(d_i \leq (1-\sigma)\bar{d}) \leq e^{-\frac{\sigma^2}{2}\bar{d}} \leq e^{-\frac{\sigma^2}{2}C\log N},\ \forall i \in [N]. \tag{17}$$

We then calculate the probability of the event that all $d_i \leq (1-\sigma)$. Since the degrees are independent, this probability is upper bounded by the following probability:

$$\sum_{i=1}^{N} P(d_i \leq (1-\sigma)\bar{d}) = Ne^{-\frac{\sigma^2}{2}\kappa \log N} = e^{(1-\frac{\sigma^2}{2}C)\log N}. \tag{18}$$

Therefore, the lower bound for all degrees can be derived as follows:

$$\mathbb{P}(d_i \geq (1-\sigma)\bar{d}) \geq 1 - e^{(1-\frac{\sigma^2}{2}\kappa)\log N} = 1 - N^{(1-\frac{\sigma^2}{2}\kappa)},\ \text{for all } 1 \leq i \leq N, \tag{19}$$

The desired lower bound is obtained by setting $\kappa = \frac{2(r+1)}{\sigma^2}$. Similarly, the upper bound can be derived, resulting in another $\kappa(\sigma, r)$. Selecting the larger value of κ completes the proof.

□

Afterward, we provide a concentration inequality for the signed adjacency matrix. Denote the expectation of $\mathcal{A}$ by $\mathbb{E}_{\mathcal{G}}[\mathcal{A}]$.

We introduce the following Lemma.

Lemma B.3. *(Oliveira, 2009) Let $\mathbf{X}_1, ..., \mathbf{X}_n \in \mathbb{R}^{n \times n}$ be symmetric independent random matrices defined on common probability space with zero means and $\mathbf{Z} = \sum_{i=1}^{n} \mathbf{X}_i$. There exists a constant m such that when the spectral norm of $\mathbf{X}_i$ is bounded by $\mathbf{X}_i \leq m$, for all $1 \leq i \leq n$, then for any $\sigma > 0$ $\bar{d} > 0$,*

$$P(\|\mathbf{Z}\| \geq \sigma\bar{d}) \leq 2Ne^{-\frac{\sigma^2\bar{d}^2}{8\lambda^2 + 4m\sigma\bar{d}}}, \tag{20}$$

where $\lambda^2 = \|\sum_{i=1}^{n} \mathbb{E}[\mathbf{X}_i^2]\|$.

The following corollary provides an upper bound for the spectral norm of $\mathcal{A} - \mathbb{E}_{\mathcal{G}}[\mathcal{A}]$.

Corollary B.4. *For any $\sigma > 0$ and $r > 0$, there exists a constant $\kappa(\sigma, r)$ such that when $\bar{d} \geq \kappa \log N$, the following holds with probability at least $1 - N^{-r}$,*

$$\mathbb{P}(\|\mathcal{A} - \mathbb{E}_{\mathcal{G}}[\mathcal{A}]\| \leq \sigma\bar{d}) \geq 1 - 2N^{-r}.$$

Proof. Let $\{\mathbf{e}\}_{i=1}^{N}$ be the canonical basis for $\mathbb{R}^N$. We can rewrite $\mathcal{A} - \mathbb{E}_{\mathcal{G}}[\mathcal{A}] = \sum_{1 \leq i \leq j \leq N} \mathbf{X}_{i,j}$, where

$$\mathbf{X}_{i,j} = \begin{cases} (I_{i,j} - b_{i,j})L_{i,j}(\mathbf{e}_i\mathbf{e}_j^T + \mathbf{e}_j\mathbf{e}_i^T) & i \neq j \\ (I_{i,j} - b_{i,j})\mathbf{e}_i\mathbf{e}_i^T & i = j \end{cases} \tag{21}$$

Let $L_{i,j} = 1$ and $b_{i,j} = p$ if v_i and v_j belong to the same class; otherwise, let $L_{i,j} = -1$ and $b_{i,j} = q$. Furthermore, let $I_{i,j} \sim \text{Bern}(b_{i,j})$.

Since $\mathbb{E}[I_{i,j} - b_{i,j}] = 0$ and the edges are sampled independently, Lemma B.3 can be applied here. By observing that

$$\|\mathbf{X}_{i,j}\| \leq \|\mathbf{e}_i\mathbf{e}_j^T + \mathbf{e}_j\mathbf{e}_i^T\| = 1, \tag{22}$$

we may set $m = 1$.

Notice that $\mathbf{e}_i^T\mathbf{e_j} = 0$ if $i \neq j$, the variances can be computed as follows.

$$\mathbb{E}[\mathbf{X}_{i,j}^2] = \begin{cases} b_{i,j}(1-b_{i,j})(\mathbf{e}_i\mathbf{e}_i^T + \mathbf{e}_j\mathbf{e}_j^T) & i \neq j \\ b_{i,j}(1-b_{i,j})\mathbf{e}_i\mathbf{e}_i^T & i = j \end{cases} \tag{23}$$

Then we have

$$\begin{aligned} \| \sum_{1\leq i\leq j} E[\mathbf{X}_{i,j}^2]\| &= \| \sum_i b_{i,i}(1-b_{i,i})\mathbf{e}\mathbf{e}^T + \sum_{i<j} b_{i,j}(1-b_{i,j})(\mathbf{e}_i\mathbf{e}_i^T + \mathbf{e}_j\mathbf{e}_j^T)\| \\ &= \| \sum_i b_{i,i}(1-b_{i,i})\mathbf{e}\mathbf{e}^T + \sum_{i\neq j} b_{i,j}(1-b_{i,j})\mathbf{e}_i\mathbf{e}_i^T\| \\ &= \| \sum_{i,j} b_{i,j}(1-b_{i,j})\mathbf{e}_i\mathbf{e}_i^T\| \ \leq \| \sum_{i,j} b_{i,j}\mathbf{e}_i\mathbf{e}_i^T\| \\ &= \| \sum_i \bar{d}\mathbf{e}_i\mathbf{e}_i^T\| = \bar{d} \end{aligned} \tag{24}$$

Now we can apply Lemma B.3 with $m = 1$ and $\lambda^2 = \bar{d}$, which gives

$$\mathbb{P}(\|\mathcal{A} - \mathbb{E}_{\mathcal{G}}[\mathcal{A}]\| \geq \sigma\bar{d}) \leq 2Ne^{-\frac{\sigma^2\bar{d}^2}{8\bar{d}+4\sigma\bar{d}}} \leq 2e^{(1-\frac{\sigma^2\kappa}{8+4\sigma})\log N} = 2N^{1-\frac{\sigma^2\kappa}{8+4\sigma}} \tag{25}$$

Setting $\kappa = \frac{(r+1)(8+4\sigma)}{\sigma^2}$, we get

$$\mathbb{P}(\|\mathcal{A} - \mathbb{E}_{\mathcal{G}}[\mathcal{A}]\| \leq \sigma\bar{d}) \geq 1 - 2N^{-r},$$

which completes the proof. □

Our concentration inequality for $\bar{\mathbf{h}}_a^{(k)} - \bar{\mathbf{h}}_b^{(k)}$ is given by the following theorem.

Theorem B.5. *For any $\sigma > 0$, and $r > 0$, there exists a constant $\kappa = (\sigma, r)$ such that when $\bar{d} \geq \kappa \log N$,*

$$\|(\bar{\mathbf{h}}_a^{(k)} - \bar{\mathbf{h}}_b^{(k)}) - \mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k)} - \bar{\mathbf{h}}_b^{(k)}]\| \leq 2k\sigma\sqrt{\frac{2C}{N}}\|\mathbf{U}\|$$

holds with probability at least $1 - O(N^{-r})$.

Proof. Let $\mathbb{E}_{\mathcal{G}}[\mathcal{P}] = \mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}\mathcal{A}\mathcal{D}^{-\frac{1}{2}}]$ be the expectation of $\mathcal{P} = \mathcal{D}^{-\frac{1}{2}}\mathcal{A}\mathcal{D}^{-\frac{1}{2}}$. Our objective is to bound $\|(\bar{\mathbf{h}}_a^{(k)} - \bar{\mathbf{h}}_b^{(k)}) - \mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k)} - \bar{\mathbf{h}}_b^{(k)}]\|$. To achieve this, we need to bound $\|\mathcal{P}^k - \mathbb{E}_{\mathcal{G}}[\mathcal{P}^k]\|$.

We consider the following spectral norm:

$$\|\mathcal{D}^{\frac{1}{2}}\mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}]\| = \max_{1\leq i\leq N}\frac{\sqrt{d_i}}{\sqrt{\bar{d}}}. \tag{26}$$

By Corollary B.2, we have

$$\mathbb{P}(\|\mathcal{D}^{\frac{1}{2}}\mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}]\| \leq \sqrt{1+\sigma}) \geq 1 - N^{-r}. \tag{27}$$

Similar to Oliveira (2009), we introduce an intermediate operator $\mathcal{O}$ defined as

$$\begin{aligned} \mathcal{O} = \mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}]\mathcal{A}\mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}] &= (\mathcal{D}^{\frac{1}{2}}\mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}])(\mathcal{D}^{-\frac{1}{2}}\mathcal{A}\mathcal{D}^{-\frac{1}{2}})(\mathcal{D}^{\frac{1}{2}}\mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}]) \\ &= (\mathcal{D}^{\frac{1}{2}}\mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}])\mathcal{P}(\mathcal{D}^{\frac{1}{2}}\mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}]) \end{aligned} \tag{28}$$

We then compute the difference between $\mathbf{P}$ and $\mathbf{O}$.

$$\begin{aligned} \|\mathcal{P} - \mathcal{O}\| &= \|(\mathcal{D}^{\frac{1}{2}}\mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}])\mathcal{P}(\mathcal{D}^{\frac{1}{2}}\mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}]) - \mathcal{P}\| \\ &= \|(\mathcal{D}^{\frac{1}{2}}\mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}] - \mathbf{I}_N)\mathcal{P}\mathcal{D}^{\frac{1}{2}}\mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}] + \mathcal{P}(\mathcal{D}^{\frac{1}{2}}\mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}] - \mathbf{I}_N)\| \\ &\leq \|(\mathcal{D}^{\frac{1}{2}}\mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}] - \mathbf{I}_N)\mathcal{P}\mathcal{D}^{\frac{1}{2}}\mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}]\| + \|\mathcal{P}(\mathcal{D}^{\frac{1}{2}}\mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}] - \mathbf{I}_N)\| \\ &\leq \|(\mathcal{D}^{\frac{1}{2}}\mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}] - \mathbf{I}_N)\| \cdot \|\mathcal{P}\| \cdot \|\mathcal{D}^{\frac{1}{2}}\mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}]\| + \|\mathcal{P}\| \cdot \|\mathcal{D}^{\frac{1}{2}}\mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}] - \mathbf{I}_N\| \\ &\leq (\sqrt{1+\sigma} - 1)(\sqrt{1+\sigma}) + (\sqrt{1+\sigma} - 1) \\ &= (\sqrt{1+\sigma} - 1)(\sqrt{1+\sigma} + 1) = \sigma. \end{aligned} \tag{29}$$

The inequality $\|\mathcal{P}\| \leq 1$ follows immediately from the fact that the spectrum of any Laplacian $\mathbf{L} = \mathbf{I} - \mathbf{P}$ lies in the interval $[0, 2]$, including signed Laplacian (Atay & Tuncel, 2014). The last inequality is obtained by utilizing Eq. (27). Consequently, it leads to

$$\mathbb{P}(\|\mathcal{P} - \mathcal{O}\| \leq \sigma) \geq 1 - N^{-r}. \tag{30}$$

We proceed with the proof by comparing $\mathcal{O}$ to $\mathbb{E}_{\mathcal{G}}[\mathcal{P}]$

$$\mathcal{O} - \mathbb{E}_{\mathcal{G}}[\mathcal{P}] = \mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}](\mathcal{A} - \mathbb{E}_{\mathcal{G}}[\mathcal{A}])\mathbb{E}_{\mathcal{G}}[\mathcal{D}^{-\frac{1}{2}}] = \sum_{i \leq j} \frac{\mathbf{X}_{i,j}}{\bar{d}} = \frac{1}{\bar{d}}(\mathcal{A} - \mathbb{E}_{\mathcal{G}}[\mathcal{A}]). \tag{31}$$

Applying Corollary B.4 we have

$$\mathbb{P}(\|\mathcal{O} - \mathbb{E}_{\mathcal{G}}[\mathcal{P}]\| \leq \sigma) \geq 1 - 2N^{-r} \tag{32}$$

Combining (30), (32), and

$$\|\mathcal{P} - \mathbb{E}_{\mathcal{G}}[\mathcal{P}]\| = \|\mathcal{P} - \mathcal{O} + \mathcal{O} - \mathbb{E}_{\mathcal{G}}[\mathcal{P}]\| \leq \|\mathcal{P} - \mathcal{O}\| + \|\mathcal{O} - \mathbb{E}_{\mathcal{G}}[\mathcal{P}]\|. \tag{33}$$

It leads to an upper bound for the norm as follows:

$$\mathbb{P}(\|\mathcal{P} - \mathbb{E}_{\mathcal{G}}[]\mathcal{P}]\| \leq 2\sigma) \geq 1 - O(N^{-r}). \tag{34}$$

Note that to ensure the simultaneous validity of both results, here κ should be set to the larger value from Corollaries B.2 and B.4.

It remains to show that the bound grows linearly with the number of layers. To achieve this, we rewrite

$$\begin{aligned}\mathcal{P}^k - \mathbb{E}_{\mathcal{G}}[\mathcal{P}^k] &= (\mathcal{P} - \mathbb{E}_{\mathcal{G}}[\mathcal{P}])(\mathcal{P}^{k-1} + \mathcal{P}^{k-2}\mathbb{E}_{\mathcal{G}}[\mathcal{P}] + ... + \mathbb{E}_{\mathcal{G}}[\mathcal{P}^{k-1}]) \\ &= (\mathcal{P} - \mathbb{E}_{\mathcal{G}}[\mathcal{P}])\mathcal{P}^{k-1} + (\mathcal{P} - \mathbb{E}_{\mathcal{G}}[\mathcal{P}])\mathcal{P}^{k-2}\mathbb{E}_{\mathcal{G}}[\mathcal{P}]... + (\mathcal{P} - \mathbb{E}_{\mathcal{G}}[\mathcal{P}])\mathbb{E}_{\mathcal{G}}[\mathcal{P}^{k-1}].\end{aligned} \tag{35}$$

Since the terms $\mathcal{P}^{k-1}$,$\mathcal{P}^{k-2}\mathbb{E}_{\mathcal{G}}[\mathcal{P}]$,..., $\mathbb{E}_{\mathcal{G}}[\mathcal{P}^{k-1}]$ have norms no more than 1, we have

$$\|\mathcal{P}^k - \mathbb{E}_{\mathcal{G}}[\mathcal{P}^k]\| \leq \|\mathcal{P} - \mathbb{E}_{\mathcal{G}}[\mathbf{P}]\| + \ldots + \|\mathcal{P} - \mathbb{E}_{\mathcal{G}}[\mathbf{P}]\| = k\|\mathcal{P} - \mathbb{E}_{\mathcal{G}}[\mathbf{P}]\|. \tag{36}$$

In turn, we get

$$\mathbb{P}(\|\mathcal{P}^k - \mathbb{E}_{\mathcal{G}}[\mathcal{P}^k]\| \leq 2k\sigma) \geq 1 - O(N^{-r}). \tag{37}$$

WLOG, we assume the nodes are properly ordered such that

$$\mathbf{U} = \begin{pmatrix} \mathbb{1}_{N/C}\mathbf{u}_1 \\ \vdots \\ \mathbb{1}_{N/C}\mathbf{u}_C \end{pmatrix}, \tag{38}$$

where $\mathbb{1}_{N/C}$ is a all 1 column vector of length N/C. For convenience, we further define a vector ζ as follows.

$$\zeta = \begin{pmatrix} \mathbb{1}_{N/C} \\ -\mathbb{1}_{N/C} \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{39}$$

Recall that $\mathbb{E}_{\mathcal{G}}[\mathcal{P}^k]$ is the k-th power of the expected propagation operator. We have

$$\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k)} - \bar{\mathbf{h}}_b^{(k)}] = \frac{C}{N}\zeta^T\mathbb{E}_{\mathcal{G}}[\mathcal{P}^k]\mathbf{U}, \tag{40}$$

and similarly

$$\bar{\mathbf{h}}_a^{(k)} - \bar{\mathbf{h}}_b^{(k)} = \frac{C}{N}\zeta^T\mathcal{P}^k\mathbf{U} \tag{41}$$

Hence, we can rewrite

$$(\bar{\mathbf{h}}_a^{(k)} - \bar{\mathbf{h}}_b^{(k)}) - \mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k)} - \bar{\mathbf{h}}_b^{(k)}] = \frac{C}{N}\zeta^T(\mathcal{P}^k - \mathbb{E}_{\mathcal{G}}[\mathcal{P}^k])\mathbf{U} \tag{42}$$

Our final result can be derived as follows:

$$\begin{aligned} \|(\bar{\mathbf{h}}_a^{(k)} - \bar{\mathbf{h}}_b^{(k)}) - \mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k)} - \bar{\mathbf{h}}_b^{(k)}]\| &= \|\frac{C}{N}\zeta^T(\mathcal{P}^k - \mathbb{E}_{\mathcal{G}}[\mathcal{P}^k])\mathbf{U}\| \\ &\leq \frac{C}{N}\|\zeta\|\|\mathcal{P}^k - \mathbb{E}_{\mathcal{G}}[\mathbf{P}^k]\|\|\mathbf{U}\| = 2k\sigma\sqrt{\frac{2C}{N}}\|\mathbf{U}\|, \end{aligned} \tag{43}$$

which proves the claim. □

B.2. Proof of Lemma 3.5

Proof. We will prove Lemma 3.5 by induction. Without loss of generality, we use sum as the pooling function. We prove that, for any node v_i and any d, the following holds.

$$\mathbf{h}_i^{(k+d)} = \rho^{(d,k)}(\mathbf{H}^{(k)}, \Phi^{(k+d)}(\cdot); i) = ||_{t=1}^{C^d}\Phi^{(k+d)}(\Gamma_{i,t}^{(k+d)}), \tag{44}$$

where $\Phi^{(k+d)}(\Gamma_{i,t}^{(k+d)}) = \sum \mathbf{h}_j^{(k)}\mathcal{W}^{(k+d)}, \Gamma_{i,t}^{(k+d)} = \{\mathbf{h}_j^{(k)} : dis(v_i, v_j) = d,\ t = \sum_{p=1}^{d}(y_p - 1)C^{d-p} + 1,\ y_j = ((t+2) \mod 3) + 1\}$. Here $v_1, v_2, ...v_d$ is a path from v_1 to v_j of length d, with $v_1 \in \mathcal{N}(v_i)$ and $v_d = v_j$, y_p is the label of v_p, $\forall p \in [d]$.

Base case: $d = 1$. It is trivial since $\rho^{(1,k)} = f_{mp}^{(k+1)}$ and $f_{mp}^{(k+1)}$ is desirable by definition.

By definition, $\rho^{(1,k)}(\mathbf{H}_i^{(k)}, \Phi^{(k+1)}(\cdot); i) = \mathbf{h}_i^{(k+1)} = f_{mp}^{(k+1)}(\mathbf{H}^{(k)}, \phi^{(k+1)}(\cdot); i) = ||_{t=1}^{C}\phi^{(k+1)}(S_{i,t}^{(k+1)})$, where $\phi^{(k+1)}(\mathcal{S}_{i,t}^{(k+1)}) = \sum \mathbf{h}_j^{(k)}\mathbf{W}^{(k+1)}, \mathcal{S}_{i,t}^{(k+1)} = \{\mathbf{h}_j^{(k)} : v_j \in \mathcal{N}(v_i), y_j = t\}$. Here, $\Gamma_{i,t}^{(k+1)} = \mathcal{S}_{i,t}^{(k+1)}$, and $\Phi^{(k+1)}(\cdot)$ is equivalent to $\phi^{(k+1)}(\cdot)$, defined as $\phi^{(k+1)}(\mathcal{S}) = \sum_{s\in S}\mathbf{s}\mathbf{W}^{(k+1)}$, for any set $\mathcal{S}$.

Inductive step: Assume this holds for d and any node v_j, i.e.,

$$\mathbf{h}_j^{(k+d)} = \rho^{(d,k)}(\mathbf{H}^{(k)}, \Phi^{(k+d)}; j) = ||_{t=1}^{C^d}\Phi^{(k+d)}(\Gamma_{j,t}^{(k+d)}), \tag{45}$$

where $\Phi^{(k+d)}(\Gamma_{j,t}^{(k+d)}) = \sum \mathbf{h}_{j'}^{(k)}\mathcal{W}^{(k+d)}, \Gamma_{j,t}^{(k+d)} = \{\mathbf{h}_{j'}^{(k)} : dis(v_j, v_{j'}) = d,\ t = \sum_{p=1}^{d}(y_p - 1)C^{d-p} + 1,\ y_{j'} = ((t+2) \mod 3) + 1\}$. Here $v_1, v_2, ...v_d$ is a path from v_1 to $v_{j'}$ of length d, with $v_1 \in \mathcal{N}(v_j)$ and $v_d = v_{j'}$, y_p is the label of v_p, $\forall p \in [d]$.

We shall show that this holds for any node v_i and $d + 1$. By definition of desirable one-hop *m-2-m* message passing,

$$\mathbf{h}_i^{(k+d+1)} = f_{mp}^{(k+d+1)}(\mathbf{H}_i^{(k+d)}, \phi^{(k+d+1)}(\cdot); i) = ||_{t=1}^{C}\phi^{(k+d+1)}(\mathcal{S}_{i,t}^{(k+d+1)}), \tag{46}$$

where $\mathcal{S}_{i,t}^{(k+d+1)} = \{\mathbf{h}_j^{(k+d)} : v_j \in \mathcal{N}(v_i), y_j = t\}$. We further define $N_{i,t} = \{v_j : v_j \in \mathcal{N}(v_i), y_j = t\}$. $N_{j,t}^{(d)} = \{v_{j'} : \mathbf{h}_{j'} \in \Gamma_{j,t}^{(k+d)}\}$. Replacing $\mathbf{h}_j^{(k+d)}$ with Eq. (45) and defining $\phi^{(k+d+1)}(\cdot)$ as the multiplication of a projection matrix on the right, followed by sum pooling, we get

$$\mathbf{h}_i^{(k+d+1)} = ||_{t=1}^{C}\sum_{\mathbf{h}_j^{(k+d)}\in\mathcal{S}_{i,t}^{(k+d+1)}}\mathbf{h}_j^{(k+d)}\mathcal{W}^{(k+d+1)} = ||_{t=1}^{C}\sum_{v_j\in N_{i,t}}(||_{t=1}^{C^d}\sum_{v_{j'}\in N_{j,t}^{(d)}}\mathbf{h}_{j'}^{(k)}\mathcal{W}^{(k+d)})\mathbf{W}^{(k+d+1)} \tag{47}$$

$$= ||_{t=1}^{C^{d+1}}\sum_{v_{j'}\in N_{j,t}^{(d)}, v_j\in N_{i,t}}\mathbf{h}_{j'}^{(k)}\mathcal{W}^{(k+d+1)} = ||_{t=1}^{C^{d+1}}\Phi^{(k+d+1)}(\Gamma_{i,t}^{(k+d+1)}), \tag{48}$$

where $\Phi^{(k+d+1)}(\Gamma_{i,t}^{(k+d+1)}) = \sum \mathbf{h}_{j'}^{(k)}\mathcal{W}^{(k+d+1)}, \Gamma_{i,t}^{(k+d+1)} = \{\mathbf{h}_{j'}^{(k)} : dis(v_i, v_{j'}) = d + 1,\ t = \sum_{p=1}^{d+1}(y_p - 1)C^{d+1-p} + 1,\ y_{j'} = ((t+2) \mod 3) + 1\}$. Here $v_1, v_2, ...v_{d+1}$ is a path from v_1 to $v_{j'}$ of length $d + 1$, with $v_1 \in \mathcal{N}(v_i)$ and $v_{d+1} = v_{j'}$, y_p is the label of v_p, $\forall p \in [d + 1]$. Note that when $\mathbf{W}^{(k+d+1)}$ is a block diagonal matrix of the form

$$\mathbf{W}^{(k+d+1)} = \begin{bmatrix} \mathbf{W} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{W} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{W} \end{bmatrix}, \tag{49}$$

where C^d $\mathbf{W}$'s with proper dimensions are on the diagonal, the following equality holds:

$$(||_{t=1}^{C^d} \sum_{v_{j'} \in N_{j,t}^{(d)}} \mathbf{h}_{j'}^{(k)} \mathcal{W}^{(k+d)})\mathbf{W}^{(k+d+1)} = ||_{t=1}^{C^d} \sum_{v_{j'} \in N_{j,t}^{(d)}} (\mathbf{h}_{j'}^{(k)} \mathcal{W}^{(k+d)} \mathbf{W}) \tag{50}$$

Therefore, we have $\mathcal{W}^{(k+d+1)} = \mathcal{W}^{(k+d)}\mathbf{W}$.

We have shown that $\mathbf{h}_i^{(k+d+1)} = ||_{t=1}^{C^{d+1}} \Phi^{(k+d+1)}(\Gamma_{i,t}^{(k+d+1)})$, $\Gamma_{i,t}^{(k+d+1)} = \{\mathbf{h}_{j'}^{(k)} : dis(v_i, v_{j'}) = d+1,\ t = \sum_{p=1}^{d+1}(y_p - 1)C^{d+1-p} + 1,\ y_{j'} = ((t+2) \mod 3) + 1\}$ and $\Phi^{(k+d+1)}(\mathcal{S}) = \sum_{s\in\mathcal{S}} \mathbf{s}\mathcal{W}^{(k+d)}\mathbf{W}$. All nodes in each set $\Gamma_{i,t}^{(k+d+1)}$ have the same label, and $\Gamma_{i,t}^{(k+d+1)}$ consists of the representations of v_i's $d+1$-hop neighbors, at the k-th layer. Combining these concludes the induction. □

B.3. Proof of Theorem 2.2

Proof. By matrix multiplication, we can express $\mathcal{T}_{ij}$ as $\mathcal{T}_{ij} = \sum_p \prod_{k=1}^K \mathcal{A}_{v_{k+1}v_k}^{(k)}$. In this expression, p represents a path of length $K+1$ from v_j to v_i, where $v_1 = v_j$, $v_{K+1} = v_i$, and $v_{k+1} \in \mathcal{N}(v_k), \forall k \in [K]$.

To prove Theorem 2.2, it suffices to provide a counterexample where all $\mathcal{A}_{v_{k+1}v_k}^{(k)}$ are desirable, but $\mathcal{T}_{ij}$ is not desirable. Let's consider a scenario with $K = 3$, $v_1v_2v_3$, $y_1 = 1, y_2 = 2, y_3 = 3$, and where there is only one such path. Since all the connected node pairs are heterophilic, the coefficients are all negative. Specifically, $\mathcal{T}_{ij} = \mathcal{A}_{v_3v_2}^{(2)}\mathcal{A}_{v_2v_1}^{(1)}$. As a result, $sign(\mathcal{T}_{ij}) = sign(\mathcal{A}_{v_3v_2}^{(2)})sign(\mathcal{A}_{v_2v_1}^{(1)}) = -1 \times -1 = 1$. However, v_i and v_j have diverse labels, thus $\mathcal{T}$ is not desirable.

The key point here is that the sign of $\mathcal{T}_{ij}$ is determined by the number of heterophilic node pairs along the path from v_i to v_j (an even number of heterophilic node pairs results in positive $\mathcal{T}_{ij}$ while odd number results in negative $\mathcal{T}_{ij}$). This aligns with the binary-class cases: after flipping a two-sided coin an even number of times, it will still land on the same side. However, this is not the case when we have multiple classes: after rolling a dice an even number of times, it could land on any of the possible numbers. □

B.4. Proof of Lemma 3.6 and Lemma A.1

We first present the formal form as follows:

Assume non-empty multisets $\mathcal{X}_a$ and $\mathcal{X}_b$ each of which consists of $x_i \in \mathbb{R}^d$ vectors. Let $\phi : 2^{\mathcal{X}} \mapsto \mathbb{R}^d$ be one of the elementwise-sum, -mean, or -max function. Consider a function $f : \mathcal{X} \mapsto \mathbb{R}^{\omega d}$ such that $f(\mathcal{X}) = ||_{k=1}^{\omega}\phi(\mathcal{X}'_k)$, where $\omega \in \mathbb{N}_+$, $\mathcal{X}'_k \subseteq \mathcal{X}$ are any disjoint subsets whose union is a full set. For message vectors: $\mathbf{m}_a^{m2m} := f(\mathcal{X}_a)$, $\mathbf{m}_b^{m2m} := f(\mathcal{X}_b)$ and $\mathbf{m}_a^{m2e}, := \phi(\mathcal{X}_a)$, $\mathbf{m}_b^{m2e}, := \phi(\mathcal{X}_b)$, the following holds: $\|\mathbf{m}_a^{m2m} - \mathbf{m}_b^{m2m}\|_2 \geq \|\mathbf{m}_a^{m2e} - \mathbf{m}_b^{m2e}\|_2$.

(1) Any two different multisets that can be discriminated by *m-2-e*, i.e., $\mathbf{m}_a^{m2e} \neq \mathbf{m}_b^{m2e}$, can be discriminated by *m-2-m* as well: $\mathbf{m}_a^{m2m} \neq \mathbf{m}_b^{m2m}$.

(2) There exist two different multisets, such that $\mathbf{m}_a^{m2m} \neq \mathbf{m}_b^{m2m}$, but *m-2-e* is unable to discriminate between them: $\mathbf{m}_a^{m2e} = \mathbf{m}_b^{m2e}$.

(3) We always have $\|\mathbf{m}_a^{m2m} - \mathbf{m}_b^{m2m}\|_2^2 \geq \|\mathbf{m}_a^{m2e} - \mathbf{m}_b^{m2e}\|_2^2$.

Proof. We prove (1) by contradiction. Assume there exist two different multisets that *m-2-m* cannot discriminate between them, i.e., $\mathbf{m}_a^{m2m} = \mathbf{m}_b^{m2m}$, but *m-2-e* determines they are different: $\mathbf{m}_a^{m2e} \neq \mathbf{m}_b^{m2e}$. $\mathbf{m}_a^{m2m} = \mathbf{m}_b^{m2m}$ indicates that $\phi(\mathcal{X}'_{a,k}) = \phi(\mathcal{X}'_{b,k}), \forall k \in [\omega]$ (assume the indices have been properly reordered). Let $\phi(\cdot)$ be the sum pooling. We have

$$\phi(\mathcal{X}'_{a,k}) = \phi(\mathcal{X}'_{b,k}) \Rightarrow \sum_{\mathbf{x}\in\mathcal{X}'_{a,k}} \mathbf{x} = \sum_{\mathbf{x}\in\mathcal{X}'_{b,k}} \mathbf{x} \tag{51}$$

We then sum over these subsets

$$\sum_{k=1}^{\omega} \phi(\mathcal{X}'_{a,k}) = \sum_{k}^{\omega} \sum_{\mathbf{x}\in\mathcal{X}'_{a,k}} \mathbf{x} = \sum_{\mathbf{x}\in\mathcal{X}'_a} \mathbf{x}. \tag{52}$$

Since $\phi(\mathcal{X}'_{a,k}) = \phi(\mathcal{X}'_{b,k}), \forall k \in [\omega]$, we can deduce

$$\sum_{k=1}^{\omega} \phi(\mathcal{X}'_{a,k}) = \sum_{k=1}^{\omega} \phi(\mathcal{X}'_{b,k}) \Rightarrow \sum_{\mathbf{x} \in \mathcal{X}'_a} \mathbf{x} = \sum_{\mathbf{x} \in \mathcal{X}'_b} \mathbf{x}. \tag{53}$$

That is to say $\phi(\mathcal{X}_a) = \phi(\mathcal{X}_b)$, which equals $\mathbf{m}_a^{m2e} = \mathbf{m}_b^{m2e}$. This contradicts to the assumption $\mathbf{m}_a^{m2e} \neq \mathbf{m}_b^{m2e}$. Hence, the assumption does not hold.

We can apply the same technique to prove the case of mean pooling with a minor modification. In the case of *m-2-m* mean pooling, the pooling operation is not performed within each subset individually. Specifically,

$$\phi(\mathcal{X}'_{a,k}) = \frac{1}{|\mathcal{X}_a|} \sum_{\mathbf{x} \in \mathcal{X}'_{a,k}} \mathbf{x}. \tag{54}$$

The subsequent steps remain the same as in the case of sum pooling.

Similarly, other weighted summation schemes can be handled in a similar manner by utilizing the same coefficient for every $\mathbf{x}$ as in *m-2-e*. which implies that $\sum_{k=1}^{\omega} \phi(\mathcal{X}'_{a,k}) = \phi(\mathcal{X}_a)$ and $\sum_{k=1}^{\omega} \phi(\mathcal{X}'_{b,k}) = \phi(\mathcal{X}_b)$. The key point here is the linearity property exhibited by the weighted average.

We now consider $\phi(\cdot)$ as an element-wise max pooling. As $\phi(\mathcal{X}_a) \neq \phi(\mathcal{X}_b)$, there must exist a component c for which we have $\mathbf{x} \in \mathcal{X}_a$ and $\mathbf{x}' \in \mathcal{X}_b$ such that $\mathbf{x}_c \neq \mathbf{x}'_c$. These values correspond to the maximum values of the c-th component among all vectors in their respective multisets. Without loss of generality, assuming $\mathbf{x}$ is in $\mathcal{X}'_{a,1}$ and $\mathbf{x}_c > \mathbf{x}'_c$, there does not exist a subset $\mathcal{X}'_{b,k}$ of $\mathcal{X}_b$ such that the c-th component of $\phi(\mathcal{X}'_{b,k})$ equals $\mathbf{x}_c$. Since we are guaranteed to have at least one different element, $\mathbf{m}_i^{m2m} \neq \mathbf{m}_{i'}^{m2m}$ is always ensured.

The proof of (2) can be demonstrated through examples. For sum and mean pooling, let us consider two multisets, namely $\{1, 3\}$ and $\{2, 2\}$ (one-dimensional vector). In the case of sum and mean pooling, both multisets are deemed equivalent since $1 + 3 = 2 + 2$ and $\frac{1+3}{2} = \frac{2+2}{2}$. However, for *m-2-m* pooling, they can be distinguished by placing two elements in different subsets. For max pooling, a similar example $\{1, 3\}$ and $\{2, 3\}$ can be used.

The proof of (3) follows directly the triangle inequality. For sum and mean pooling and other weighted summation schemes, we have

$$\sum_{k=1}^{\omega} \phi(\mathcal{X}'_{a,k}) = \phi(\mathcal{X}_a), \quad \sum_{k=1}^{\omega} \phi(\mathcal{X}'_{b,k}) = \phi(\mathcal{X}_b) \tag{55}$$

and

$$\mathbf{m}_a^{m2m} = [\phi(\mathcal{X}'_{a,1}) || \phi(\mathcal{X}'_{a,2}) || \ldots || \phi(\mathcal{X}'_{a,\omega})],\ \mathbf{m}_b^{m2m} = [\phi(\mathcal{X}'_{b,k}) || \phi(\mathcal{X}'_{b,k}) || \ldots || \phi(\mathcal{X}'_{b,\omega})] \tag{56}$$

By triangle inequality, we get

$$\begin{aligned} \|\mathbf{m}_a^{m2m} - \mathbf{m}_b^{m2m}\|_2^2 = \sum_{k=1}^{\omega} \|\phi(\mathcal{X}'_{a,k}) - \phi(\mathcal{X}'_{b,k})\|_2^2 \geq \| \sum_{k=1}^{\omega} (\phi(\mathcal{X}'_{a,k}) - \phi(\mathcal{X}'_{b,k}))\|_2^2 \\ = \|\phi(\mathcal{X}_a) - \phi(\mathcal{X}_b)\| = \|\mathbf{m}_a^{m2e} - \mathbf{m}_b^{m2e}\|_2^2. \end{aligned} \tag{57}$$

In the case of max pooling, the subsets of *m-2-m* also contain the maximum value of each component. Without loss of generality, we can rearrange the vectors such that the maximum values of all the components of $\mathcal{X}_a$ and $\mathcal{X}_b$ are located in $\mathcal{X}'_{a,1}$ and $\mathcal{X}'_{b,1}$, respectively. In other words, we have $\phi(\mathcal{X}'_{a,1}) = \phi(\mathcal{X}_a)$ and $\phi(\mathcal{X}'_{b,1}) = \phi(\mathcal{X}_b)$. It follows that

$$\|\mathbf{m}_a^{m2m} - \mathbf{m}_b^{m2m}\|_2^2 = \sum_{k=1}^{\mathcal{C}} \|\phi(\mathcal{X}'_{a,k}) - \phi(\mathcal{X}'_{b,k})\|_2^2 \geq \|\phi(\mathcal{X}'_{a,1}) - \phi(\mathcal{X}'_{b,1})\|_2^2 = \|\phi(\mathcal{X}_a) - \phi(\mathcal{X}_b)\| = \|\mathbf{m}_a^{m2e} - \mathbf{m}_b^{m2e}\|_2^2. \tag{58}$$

This completes the proof.

□

B.5. Proof of Theorem 2.4

We first present the formal form of Theorem 2.4:

Theorem 2.4 (Formal). Let $\mathcal{G}$ be a graph (a random variable) generated under the CSBM model (Def. 2.3). Denote the set of nodes belonging to each class $c \in [C]$ as $\mathcal{V}_c := \{v_j \in \mathcal{V} : y_j = c\}$. Denote the mean of K-th layer embeddings of $\mathcal{V}_c$ as $\bar{\mathbf{h}}_c^{(K)} := \frac{C}{N} \sum_{v_i \in \mathcal{V}_c} \mathbf{h}_i^{(K)}$. Assume any two classes $a, b \in [C]$, such that at the $k-1$-th layer, their expected class means converge to the same point:

$$\|E_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k-1)}] - E_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k-1)}]\| = 0. \tag{59}$$

Let $E_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k)}]$ and $E_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k)}]$ be the expected means obtained by using Eq. (6), with $\phi(\cdot)$ being a mean pooling. Then, we have:

$$\|\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k)}] - \mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k)}]\| > 0, \tag{60}$$

if $\|\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k-1)}]\| \neq 0$ or $E_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k-1)}] \neq 0$ and $p \neq q$.

Proof. According to the definition of Eq. (6), $\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k)}]$ can be calculated as:

$$\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k)}] = \frac{1}{p + (\mathcal{C}-1)q}[\cdots || p\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k-1)}] || q\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k-1)}] || \cdots] \tag{61}$$

Similarly, we get

$$\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k)}] = \frac{1}{p + (\mathcal{C}-1)q}[\cdots || q\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k-1)}] || p\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k-1)}] || \cdots] \tag{62}$$

Then, $\|\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k)}] - \mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k)}]\|$ is derived as follows:

$$\begin{aligned}\|\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k)}] - \mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k)}]\| &= \frac{1}{p + (\mathcal{C}-1)q}\|[\cdots || (p-q)\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k-1)}] || (q-p)\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k-1)}] || \cdots]\| \\ &\geq \frac{|p-q|}{p + (\mathcal{C}-1)q}\|E_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k-1)}]\| + \frac{|p-q|}{p + (\mathcal{C}-1)q}\|E_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k-1)}]\| > 0\end{aligned} \tag{63}$$

If we use a `SMP` as the same as Eq. (3), the expected class means can be calculated as follows:

$$\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k)}] = \frac{p\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k-1)}] - q\sum_{c \in [C], c \neq a} \mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_c^{(k-1)}]}{p + (C-1)q}, \tag{64}$$

and similarly,

$$\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k)}] = \frac{p\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k-1)}] - q\sum_{c \in [C], c \neq a} \mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_c^{(k-1)}]}{p + (C-1)q}. \tag{65}$$

By subtracting $\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k)}]$ from $\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k)}]$, we get

$$\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k)} - \bar{\mathbf{h}}_b^{(k)}] = \frac{p+q}{(p + (C-1)q)}\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k-1)} - \bar{\mathbf{h}}_b^{(k-1)}] = 0. \tag{66}$$

Thus, we have shown that *m-2-m* can escape from oversmoothing, while `SMP` get stuck in oversmoothing if $\mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_a^{(k-1)}] = \mathbb{E}_{\mathcal{G}}[\bar{\mathbf{h}}_b^{(k-1)}]$ provided.

□

$$\tag{67}$$

C. Dataset and Baseline Description

C.1. Datasets

The citation networks **Cora, Citeseer,** and **Pubmed** (Yang et al., 2016) are comprised of nodes representing scientific papers and edges representing citation relationships. The node features are represented as bag-of-words, and each label corresponds to the field to which the paper belongs.

Actor (Pei et al., 2020) is a co-occurrence network derived from the film-director-actor-writer network, where the nodes are characterized by bag-of-words representations of the actors' Wikipedia pages. The network's edges indicate the co-occurrence of two actors on the same web page. The task is to classify actors into five categories

Cornell, Texas, and Wisconsin (Pei et al., 2020) consist of web pages collected from computer science departments at various universities. In these datasets, the nodes represent web pages, while the edges symbolize the hyperlinks connecting them. We employ bag-of-words representations as feature vectors for the nodes. The objective is to classify the web pages into five categories: student, project, course, staff, and faculty.

Chameleon and Squirrel (Rozemberczki et al., 2021) are two subgraphs of web pages in Wikipedia. Nodes in this context symbolize web pages, while the edges represent the hyperlinks connecting them. The node features encapsulate various informative nouns extracted from the corresponding Wikipedia pages. The objective of this task is to forecast the average daily traffic received by each web page.

Penn94 (Lim et al., 2021) refers to a subgraph obtained from Facebook, focusing on students as its nodes. These nodes possess various features such as their majors, second majors/minors, dormitories/houses, academic years, and high schools. The gender of the students is utilized as the labels for the nodes.

Amazon-ratings (Platonov et al., 2022) This dataset is based on the Amazon product co-purchasing network metadata dataset2 from SNAP Datasets. Nodes are products (books, music CDs, DVDs, VHS video tapes), and edges connect products that are frequently bought together. The task is to predict the average rating given to a product by reviewers.

C.2. Baselines

GPR-GNN (Chien et al., 2020): The mathematical description of the process employed in **GPR-GNN** can be formulated as follows:

$$\mathbf{Z} = \sum_{k=0}^{K} \gamma_k \mathbf{P}^k \mathbf{H}^{(0)}. \tag{68}$$

Here, $\mathbf{P} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ represents the propagation matrix utilized in GCN (Kipf & Welling, 2016), and γ_k stands for a learned real value. The propagation coefficient for the connection between nodes i and j is calculated as $\gamma_k \mathbf{P}_{ij}^k$. Since y_k are real values, $\gamma_k \mathbf{P}_{ij}^k$ can be negative.

FAGCN (Bo et al., 2021): FAGCN combines both low-frequency and high-frequency signals by updating node representation as:

$$\mathbf{h}_i^{(l)} = \epsilon \mathbf{h}_i^{(l)} + \sum_{i \in \mathcal{N}(i)} \frac{\alpha_{ij}}{d_i d_j} \mathbf{h}_j^{(l-1)}, \tag{69}$$

where α_{ij} represents a learned propagation coefficient lies in $(-1, 1)$.

GGCN: (Yan et al., 2022) GGCN incorporates signed messages by computing cosine similarity between connected node pairs:

$$\mathbf{S}_{ij}^{(l)} = Cosine(\mathbf{h}_i^{(l-1)}, \mathbf{h}_j^{(l-1)}). \tag{70}$$

GGCN learns a weighted combination of self-representations, positive messages, and negative messages:

$$\mathbf{h}_i^{(l)} = \mathrm{ReLU}(\beta_1^{(l)} \mathbf{h}_i^{(l)} + \beta_2^{(l)} \sum_{j \in \mathcal{N}(i), \mathbf{S}_{ij}^{(l)} > 0} \mathbf{S}_{ij}^{(l)} \mathbf{h}_j^{(l)} + \beta_3^{(l)} \sum_{j \in \mathcal{N}(i), \mathbf{S}_{ij}^{(l)} < 0} \mathbf{S}_{ij}^{(l)} \mathbf{h}_j^{(l)}) \tag{71}$$

ACM-GCN (Luan et al., 2022): ACM-GCN includes a pair of filters that retain the low-frequency and high-frequency.

$$\mathbf{H}^{(l)} = \text{ReLU}(diag(\alpha_H^{(l)})(\mathbf{I}-\mathbf{P})\mathbf{H}^{(l-1)}\mathbf{W}_{HP}^{(l)} + diag(\alpha_L^{(l)})\mathbf{P}\mathbf{H}^{(l-1)}\mathbf{W}_{LP}^{(l)} + diag(\alpha_I^{(l)})\mathbf{H}^{(l-1)}\mathbf{W}_I^{(l)}). \tag{72}$$

Here, the signed message is incorporated through the term $\mathbf{I}-\mathbf{P}$.

Goal (Zheng et al., 2023): The node representations in the convolution layer of Goal are updated as:

$$\mathbf{H}^{(l+1)} = \text{ReLU}((\alpha\mathbf{I} + \beta\mathbf{A}_o - \gamma\mathbf{A}_t - \delta\mathbf{A}_{to})\mathbf{H}^{(l)}\mathbf{W}^{(l)}), \tag{73}$$

where $\mathbf{A}_o$ and $\mathbf{A}_t$ represent the constructed homophily graph and heterophily graph, respectively. $\mathbf{A}_{to}$ represents the combined graph. The utilization of signed messages occurs through the terms $-\gamma\mathbf{A}_t$ and $-\delta\mathbf{A}_{to}$.

AERO-GNN (Lee et al., 2023): AERO-GNN is an extension of GPR-GNN, distinguished by two significant differences. Firstly, while GPR-GNN employs the propagation matrix of GCN, AERO-GNN learns attention coefficients. Secondly, in AERO-GNN, the coefficient γ_k operates on a node-wise basis, whereas GPR-GNN adopts a layer-wise γ_k. Consequently, negative γ_k values are employed to introduce signed messages.

H$_2$GCN (Zhu et al., 2020) : There are three designs regarding concatenation in H$_2$GCN.

$$\mathbf{h}_i^{(k)} = [\mathbf{h}_i^{(k-1)} || \mathbf{m}_i^{(k)}], \tag{74}$$

where the ego- and neighbor-information are separated by concatenation. Second, the direct and second-order neighbor information are separated as it indicates that second-order neighbors expect to contain more homophilic neighbors in heterophilic graphs.

$$\mathbf{m}_i^{(k)} = [\texttt{Aggr}(\{h_j : v_j \in \mathcal{N}(v_i)\}) || \texttt{Aggr}(\{h_j : v_j \in \mathcal{N}_2(v_i)\})] \tag{75}$$

Lastly, the final out is obtained by combining the representations of different hops.

$$\mathbf{h}_i^{(\texttt{final})} = [\mathbf{h}_i^{(1)} || \mathbf{h}_i^{(2)} || ... || \mathbf{h}_i^{(K)}] \tag{76}$$

Ordered GNN (Song et al., 2022) : Ordered GNN employs a mechanism where the representations of different hops are separated and addresses the issue of dimension explosion. Specifically, the final representation of a node is obtained as follows:

$$\mathbf{h}_i^{(\texttt{final})} = \mathbf{h}_i^{(1)} \odot g_i^{(1)} + \mathbf{h}_i^{(2)} \odot (g_i^{(2)} - g_i^{(1)}) \ldots + \mathbf{h}_i^{(K)} \odot (g_i^{(K)} - g_i^{(K-1)}), \tag{77}$$

where $g_i^{(k)}$ is a vector with a split point $p_i^{(k)}$, such that the first $p_i^{(k)}$ elements of $g_i^{(k)}$ are ones while the remaining $1 - p_i^{(k)}$ elements are zeros. These $p_i^{(k)}$ values satisfy $p_i^{(k+1)} \geq p_i^{(k)}$. In some cases, when the models become deeper, $g_i^{(k)}$ at shallower layers approach being all-one vectors. Consequently, the deep representations $\mathbf{h}_i^{(k+1)}$ in later layers are rarely encoded in the final representation $\mathbf{h}_i^{(\texttt{final})}$, as $g_i^{(k+1)} - g_i^{(k)} \approx \mathbf{0}$ when $g_i^{(k)}$ is nearly an all-one vector. This can be confirmed by referring to the visualization results presented in the original paper.

DMP (Yang et al., 2021): DMP leverages a channel-wise attention mechanism:

$$\mathbf{h}_i^{(l)} = \text{ReLU}(\mathbf{c}_{ii}^{(l)} \odot \mathbf{h}_i^{(l-1)} + \sum_{j \in \mathcal{N}(i)} \mathbf{c}_{ij}^{(l)} \odot \mathbf{h}_j^{(l-1)}), \tag{78}$$

where $\odot$ represents the element-wise product of vectors, and $\mathbf{c}_{ij}^{(l)} = \tanh([\mathbf{h}_i^{(l-1)} || \mathbf{h}_j^{(l-1)}]\mathbf{W}_c^{(l)})$ is a vector with the same length as $\mathbf{h}_j^{(l-1)}$. The activation function $\tanh(c\cdot)$ suggests that the message can become negative.

D. Experimental Setup

We implemented M2M-GNN using PyTorch (Paszke et al., 2019). We applied LayerNorm (Ba et al., 2016) after each layer. Adam optimizer (Kingma & Ba, 2014) was used for optimizing the model parameters. Our experiments were conducted on an NVIDIA Tesla V100 with 32GB of memory. For dataset splitting, we utilized 10 random splits with training, validation,

Hyperparameter	Range
learning rate	$\{0.005, 0.01, 0.02\}$
weight decay	$\{5e-4, 5e-3, 1e-2\}$
dropout	$[0, 0.7]$
early stopping	$\{40, 100, 200\}$
β	$[0.0, 0.9]$
λ	$[0.0, 1.0]$
hidden channels d	$\{32, 64, 128, 256\}$
α	$(0, 1)$
τ	$\{0.1, 0.5, 1.0, 2.0\}$
number of layers	$\{2, 4, 8\}$

Table 3: Search space for hyperparameters.

and testing proportions of 48%, 32%, and 20%, respectively, as provided in Pei et al. (2020). These splits are available from PyTorch Geometry (Fey & Lenssen, 2019). If the results of the baseline methods on these benchmark datasets are publicly available, we report them directly. Otherwise, we executed these methods using the codes released by their authors and fine-tuned the hyperparameters based on the validation set. To identify the optimal hyperparameters, a grid search was performed on the validation set, and the search space is summarized in Table 3.

E. Time Complexity

This section presents an analysis of the time complexity of M2M-GNN. M2M-GNN involves several operations. Firstly, in each layer, a linear transformation $\mathbf{h}_i^{(k)}\mathbf{W}^{(k)}$ is performed using matrix multiplication between two matrices of sizes $N \times d$ and $d \times d/C$. This operation has a time complexity of $\mathcal{O}(Nd^2/C)$. Next, the computation of $\mathbf{s}^{(k)}(i,j)$ requires a vector addition with a time complexity of $\mathcal{O}(|\mathcal{E}|d/C)$, as well as a matrix multiplication with a time complexity of $\mathcal{O}(|\mathcal{E}|d)$. Following that, a message vector is constructed for each node according to the equation:

$$\mathbf{m}_i^{(k)} = \|_{t=1}^{\mathcal{C}}\mathbf{C}_{it}^{(k)}, \quad \mathbf{C}_{it}^{(k)} = \sum_{v_j \in N_i} \mathbf{s}_t^{(k)}(i,j)\hat{\mathbf{h}}_j^{(k)} \in \mathbb{R}^{d/\mathcal{C}}, \forall t \in [\mathcal{C}] \tag{79}$$

The time complexity of the computation of $\mathbf{C}_{it}^{(k)}$ is $\mathcal{O}(|\mathcal{E}|d/\mathcal{C})$. Since we have $\mathcal{C}$ chunks, the total time complexity for constructing message vectors is $\mathcal{O}(|\mathcal{E}|d)$. Additionally, the combination of $\mathbf{m}_i^{(k)}$ and $\mathbf{h}_i^{(0)}$ costs $\mathcal{O}(Nd)$. By combining all these operations, the derived time complexity of M2M-GNN is $\mathcal{O}(Nd^2/\mathcal{C} + |\mathcal{E}|d)$.

In comparison, the time complexity of the vanilla GCN (Kipf & Welling, 2016) is $\mathcal{O}(Nd^2 + |\mathcal{E}|d)$. It is worth noting that the time complexity of our proposed M2M-GNN is equivalent to that of GCN.

F. Impact of Non-linearity

The ReLU activation function has been known to exacerbate over-smoothing. In this study, we investigate the scenario of a GNN utilizing an activation function after k linear graph convolutions. We analyze the difference $\|\mathrm{ReLU}(\bar{\mathbf{h}}_a^{(k)}) - \mathrm{ReLU}(\bar{\mathbf{h}}_b^{(k)})\|$, and show that $\|\bar{\mathbf{h}}_a^{(k)} - \bar{\mathbf{h}}_b^{(k)}\| \geq \|\mathrm{ReLU}(\bar{\mathbf{h}}_a^{(k)}) - \mathrm{ReLU}(\bar{\mathbf{h}}_b^{(k)})\|$. Given that this activation operates element-wise, our analysis centers on the i-th element:

$$\mathrm{ReLU}(\bar{\mathbf{h}}_{a,i}^{(k)}) - \mathrm{ReLU}(\bar{\mathbf{h}}_{b,i}^{(k)}) = \begin{cases} 0, & \bar{\mathbf{h}}_{a,i}^{(k)} < 0, \bar{\mathbf{h}}_{b,i}^{(k)} < 0 \\ -\bar{\mathbf{h}}_{b,i}^{(k)}, & \bar{\mathbf{h}}_{a,i}^{(k)} < 0, \bar{\mathbf{h}}_{b,i}^{(k)} > 0 \\ \bar{\mathbf{h}}_{a,i}^{(k)}, & \bar{\mathbf{h}}_{a,i}^{(k)}, \bar{\mathbf{h}}_{b,i}^{(k)} < 0 \\ \bar{\mathbf{h}}_{a,i}^{(k)} - \bar{\mathbf{h}}_{b,i}^{(k)}, & \bar{\mathbf{h}}_{a,i}^{(k)} > 0, \bar{\mathbf{h}}_{b,i}^{(k)} > 0 \end{cases}. \tag{80}$$

$$|\bar{\mathbf{h}}_{a,i}^{(k)} - \bar{\mathbf{h}}_{b,i}^{(k)}| - |\mathrm{ReLU}(\bar{\mathbf{h}}_{a,i}^{(k)}) - \mathrm{ReLU}(\bar{\mathbf{h}}_{b,i}^{(k)}| = \begin{cases} |\bar{\mathbf{h}}_{a,i}^{(k)} - \bar{\mathbf{h}}_{b,i}^{(k)}|, & \bar{\mathbf{h}}_{a,i}^{(k)} < 0, \bar{\mathbf{h}}_{b,i}^{(k)} < 0 \\ |\bar{\mathbf{h}}_{a,i}^{(k)}|, & \bar{\mathbf{h}}_{a,i}^{(k)} < 0, \bar{\mathbf{h}}_{b,i}^{(k)} > 0 \\ |\bar{\mathbf{h}}_{b,i}^{(k)}|, & \bar{\mathbf{h}}_{a,i}^{(k)} > 0, \bar{\mathbf{h}}_{b,i}^{(k)} < 0 \\ 0, & \bar{\mathbf{h}}_{a,i}^{(k)} > 0, \bar{\mathbf{h}}_{b,i}^{(k)} > 0 \end{cases}. \tag{81}$$

Therefore, we always have $|\bar{\mathbf{h}}_{a,i}^{(k)} - \bar{\mathbf{h}}_{b,i}^{(k)}| \geq |\text{ReLU}(\bar{\mathbf{h}}_{a,i}^{(k)}) - \text{ReLU}(\bar{\mathbf{h}}_{b,i}^{(k)})|$, which leads to $(\bar{\mathbf{h}}_{a,i}^{(k)} - \bar{\mathbf{h}}_{b,i}^{(k)})^2 \geq (\text{ReLU}(\bar{\mathbf{h}}_{a,i}^{(k)}) - \text{ReLU}(\bar{\mathbf{h}}_{b,i}^{(k)}))^2$. Consequently, the following relationship can be derived: $\|\bar{\mathbf{h}}_{a}^{(k)} - \bar{\mathbf{h}}_{b}^{(k)}\| = \sqrt{\sum_i (\bar{\mathbf{h}}_{a,i}^{(k)} - \bar{\mathbf{h}}_{b,i}^{(k)})^2} \geq \sqrt{\sum_i (\text{ReLU}(\bar{\mathbf{h}}_{a,i}^{(k)}) - \text{ReLU}(\bar{\mathbf{h}}_{b,i}^{(k)}))^2} = \|\text{ReLU}(\bar{\mathbf{h}}_{a}^{(k)}) - \text{ReLU}(\bar{\mathbf{h}}_{b}^{(k)})\|$. This suggests that the ReLU activation function consistently increases the similarity between representations, thereby exacerbating over-smoothing.

Regarding Theorem 2.2, our conclusion remains valid even when nonlinear activation functions are applied. For instance, let's consider an `SMP` that incorporates a non-linear activation function in updating the representation matrix as $\mathbf{H}^{(l+1)} = \sigma(\mathcal{A}\mathbf{H}^{(l)}\mathbf{W})$, where $\sigma(\cdot)$ represents various non-linear activations, and $\mathcal{A}$ is a propagation matrix. Under this framework, for a given vector v, elements of the vector $\frac{\sigma(v)}{v}$ adhere to the condition $0 \leq \frac{\sigma(v_i)}{v_i} \leq 1$ (equal to 0 if $v_i = 0$) for each component i. It is worth noting that most common non-linear activation functions, such as ReLU, LeakyReLU, Tanh, and GeLU, conform to this requirement. Consequently, the i-th column of $\mathbf{H}^{(l+1)}$ can be redefined as $\mathbf{H}_{:,i}^{(l+1)} = \mathbf{N}_i\mathcal{A}(\mathbf{H}^{(l)}\mathbf{W})_{:,i}$, where $\mathbf{N}_i$ is a diagonal matrix with elements ranging between 0 and 1, acting as the non-linear activation factor. The modified coefficient matrix $\mathcal{A}' = \mathbf{N}\mathcal{A}$ inherits the sign of its elements from the original matrix $\mathcal{A}$ due to the non-negativity of the diagonal elements in $\mathbf{N}_i$. Therefore, non-linear activation does not alter the signs of the elements. According to Definition 2.1, the desirability of a matrix depends solely on the signs of its elements. Theorem 2.2 demonstrates the existence of desirable $\mathcal{A}^{(\|)}$ matrices whose product, denoted as $\mathcal{T}$, does not retain the desirability criteria. Furthermore, when any $\mathbf{N}_i\mathcal{T}$ resulting from applying a non-linear activation function satisfies $sign(\mathbf{N}_i\mathcal{T}) = sign(\mathcal{T})$, it indicates that $\mathbf{N}_i\mathcal{T}$ is considered an undesirable matrix as defined in Definition 2.1.