Feature Distribution on Graph Topology Mediates the Effect of Graph Convolution: Homophily Perspective

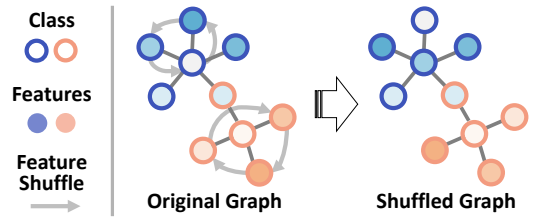
Soo Yong Lee [1] **Sunwoo Kim** [1] **Fanchen Bu** [1 2] **Jaemin Yoo** [2] **Jiliang Tang** [3] **Kijung Shin** [1 2]

Abstract

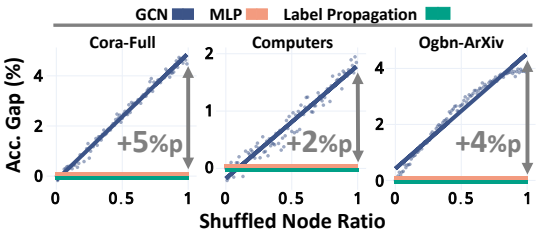
How would randomly shuffling feature vectors among nodes from the same class affect graph neural networks (GNNs)? The feature shuffle, intuitively, perturbs the dependence between graph topology and features (A-X dependence) for GNNs to learn from. Surprisingly, we observe a consistent and significant improvement in GNN performance following the feature shuffle. Having overlooked the impact of A-X dependence on GNNs, the prior literature does not provide a satisfactory understanding of the phenomenon. Thus, we raise two research questions. First, how should A-X dependence be measured, while controlling for potential confounds? Second, how does A-X dependence affect GNNs? In response, we (*i*) propose a principled measure for A-X dependence, (*ii*) design a random graph model that controls A-X dependence, (*iii*) establish a theory on how A-X dependence relates to graph convolution, and (*iv*) present empirical analysis on real-world graphs that align with the theory. We conclude that A-X dependence mediates the effect of graph convolution, such that smaller dependence improves GNN-based node classification.

1. Introduction

Graph neural networks (GNNs) are functions of graph topology and features. Understanding the conditions in which GNNs become powerful is the key to their improvement and effective applications. As such, many prior works have investigated the conditions that affect GNN effectiveness, especially from the node representation learning perspective (Oono & Suzuki, 2020; Abboud et al., 2021; You et al., 2021; Wang & Zhang, 2022; Wei et al., 2022; Wu et al., 2023; Baranwal et al., 2021; 2023).



(a) Visualization of the feature shuffle. Features of the same class nodes are shuffled. The shuffled node ratio is 0.6 in the example.



(b) Accuracy gaps between the original and shuffled graphs.

Figure 1: **An Intriguing Phenomenon.** GCN performance increases significantly over the feature shuffle, while those of MLP and label propagation remain stationary.

However, in this work, we report an intriguing phenomenon not well accounted for by the prior studies. How would randomly shuffling feature vectors among nodes from the same class affect GNNs? The feature shuffle, intuitively, disrupts the dependence between graph topology and node features (A-X dependence). Rather surprisingly, increasing the shuffled node ratio consistently improves GCN (Kipf & Welling, 2017) performance (Fig. 1). The performances of MLP and label propagation (LP), however, remain the same (for experiment details, refer to Sec. 5.1).

The prior studies on GNN theory do not provide a satisfactory understanding of the phenomenon. One line of studies indicates how class distribution on graph topology, such as class-homophily, can be critical for effective GNNs (Luan et al., 2022; 2023; Ma et al., 2022; Mao et al., 2023; Platonov et al., 2023a). The feature shuffle, however, does not intervene with label distribution because the class labels are not

[1]Kim Jaechul Graduate School of Artificial Intelligence, KAIST, Daejeon, Republic of Korea [2]School of Electrical Engineering, KAIST, Daejeon, Republic of Korea [3]Department of Computer Science and Engineering, Michigan State University, Michigan, US. Correspondence to: Kijung Shin <kijungs@kaist.ac.kr>.

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

shuffled, causing the LP performance to be unchanged.

Some studies point out feature informativeness for node class as another crucial factor for effective GNNs (Baranwal et al., 2021; 2023; Wei et al., 2022; Wu et al., 2023). These do not explain the reported phenomenon, either, since the feature shuffle is done only among nodes from the same class. Namely, feature informativeness for node class remains the same, leaving the MLP performance unaffected.

Others stress GNN's efficacy as a node signal denoiser (NT & Maehara, 2019; Ma et al., 2021). From such perspective, whether the signals are well denoised after graph convolution is important for effective node classification (Luan et al., 2023). However, they do not discuss conditions in which the convoluted features are well-denoised. The reason, thus, is vague for why the feature shuffle affects GNN performance.

The limitation of the prior works in understanding the observed phenomenon stems from overlooking the impact A-X dependence on GNNs. Thus, we *advance the findings from prior works* by investigating how A-X dependence affects GNNs. We raise two research questions (RQs).

- **RQ1**. How should A-X dependence be measured, while controlling for potential confounds?
- **RQ2**. How does A-X dependence affect GNNs?

In response, we propose a principled measure for A-X dependence, *class-controlled feature homophily* (CFH), that mitigates potential confounding by node class. We propose a random graph model, *CSBM-X*, to control CFH. With the measure, graph model, and feature shuffle, we establish a theory that CFH mediates the effect of graph convolution. Specifically, CFH moderates its force to pull each node feature toward the feature mean of the respective node class, with smaller CFH increasing the force.

2. Preliminaries

Graphs. A graph $G = (V, E)$ is defined by a *node set* $V = V(G)$ and an *edge set* $E = E(G) \subseteq \binom{V}{2}$. We denote an edge between two nodes u and v as $(u, v) \in E$, and $(u, v) = (v, u)$ holds unless otherwise stated.

Let $n = n(G)$ denote the number of nodes in G with $V = \{v_i : i \in [n]\}$. Let $X = X(G) \in \mathbb{R}^{n \times k}$ denote node feature matrix, where the i-th row corresponds to the feature vector $X_i \in \mathbb{R}^k$ of node $v_i \in V$, where $k = k(G)$ is the feature dimension. For each node $v_i \in V$, its class is $Y_i \in [c]$, where c is the number of node classes. Its neighbor set is $N_i = \{v_j \in V : (v_i, v_j) \in E\}$. Its degree is $d_i = |N_i|$, and its same- and different-class degrees are $d_i^+ = |\{v_j \in N_i : Y_j = Y_i\}|$ and $d_i^- = |\{v_j \in N_i : Y_j \neq Y_i\}|$, respectively, with $d_i = d_i^+ + d_i^-$.

We define $V_i' = V \setminus \{v_i\}$ as the set of nodes excluding v_i. Also, for each class $\ell \in [c]$, we use $C_\ell^+ = \{v_i \in V : Y_i = \ell\}$ to denote node set of class ℓ, and $C_\ell^- = V \setminus C_\ell^+$ denotes the rest.

Feature distance (FD). For measuring feature distance between classes, we adopt a simplified version of Bhattacharyya distance (Kailath, 1967). Specifically, given two data classes C_0 and C_1 with feature means $\mu_i \in \mathbb{R}^k$ and covariance matrices $\Sigma_i \in \mathbb{R}^{k \times k}$ with $i = 0$ or 1 respectively, we define the feature distance FD between C_0 and C_1 as:

$$\mathrm{FD}(C_0, C_1) \coloneqq \sqrt{(\mu_0 - \mu_1)^\top \left(\frac{\Sigma_0 + \Sigma_1}{2}\right)^{-1} (\mu_0 - \mu_1)}. \quad (1)$$

A higher feature distance FD indicates a larger (normalized) distance between the two classes, i.e., the two classes are more distinct. If both classes follow a Gaussian distribution, roughly speaking, the difficulty in classifying C_0 and C_1 decreases as feature distance $\mathrm{FD}(C_0, C_1) \in [0, \infty)$ increases (Kailath, 1967).

Homophily. From a network perspective, homophily (*love of the same*) refers to the *positive dependence* between node similarity and connection (McPherson et al., 2001). Heterophily (*love of the different*) is considered as the opposite, describing the *negative dependence* in that dissimilar nodes tend to connect (Rogers et al., 2014). Importantly, we distinguish *impartiality* from both for networks having *no dependence* between node similarity and their connection.

The vast majority of works on GNN-homophily connection focus specifically on class-homophily. We use $\mathbf{h}_c$ to denote the class-homophily defined by Lim et al. (2021):

$$\mathbf{h}_c = \frac{1}{c} \sum_{\ell \in [c]} \max\left(\frac{\sum_{v_i \in C_\ell^+} d_i^+}{\sum_{v_i \in C_\ell^+} d_i} - \frac{|C_\ell^+|}{|V|},\ 0\right) \quad (2)$$

Contextual stochastic block models (CSBMs). Stochastic block models (SBMs) are widely used graph models for network analysis (Holland et al., 1983), with distinct communities, or blocks, consisting of same-class nodes. CSBMs (Deshpande et al., 2018) supplement SBMs by considering node features. Recently, many researchers have used CSBMs and developed their variants for GNN analysis (Wei et al., 2022; Palowitch et al., 2022; Wu et al., 2023; Baranwal et al., 2021; 2023; Luan et al., 2023), where they directly control dependence between (*i*) topology and class (i.e., class-homophily $\mathbf{h}_c$) and (*ii*) features and class (i.e., feature distance FD). It is, however, non-trivial to control dependence between topology and features with the prior CSBMs, while holding the other two dependence (i.e., $\mathbf{h}_c$ and FD) constant.

3. Measure and Patterns

In this section, we address the first research question (**RQ1**) on the measure of A-X dependence (i.e., dependence between graph topology and features).

3.1. Design Goals and Intuition

We target two central goals in designing A-X dependence measure $\tilde{\mathbf{h}}(\cdot)$. First, the measure $\tilde{\mathbf{h}}(\cdot)$ should distinguish positive, negative, and no dependence. Second, if a third variable is available (i.e., node class Y), the measure $\tilde{\mathbf{h}}(\cdot)$ should control its potential effects on A-X dependence.

Their intuition is as follows. Let us first illustrate the examples of positive and negative dependence. Consider a *friendship network* (FN) of adults in the ages of 20-50s. People with similar *political inclinations* (PI) tend to become friends, so FN positively depends on PI. People with *care need* (CN) tend to seek friends without CN to receive their support, making FN negatively dependent on CN.

Now, we further illustrate no dependence and class-control. Consider the number of *wrinkles* (WK). People generally do not make friends based on their WK, but WK still positively depends on FN. This is because the *age group* (AG; i.e., node class) confounds the dependence between WK and FN. Controlling for the effect of AG on WK, the dependence between WK and FN should no longer exist.

3.2. Measure Design

To achieve the design goals, we propose **C**lass-controlled **F**eature **H**omophily (**CFH**) measure $\tilde{\mathbf{h}}(\cdot)$.

Class-controlled features. Assuming a linear relation between classes and features, we mitigate their association to define ***class-controlled features*** $X|Y$.

$$X_i|Y = X_i - \left(\frac{1}{|C^+_{Y_i}|} \sum_{v_j \in C^+_{Y_i}} X_j \right). \tag{3}$$

For intuition, consider the former example. Let AG be the class and WK be the feature. Since AG affects WK, WK distributions are different for each AG. However, the AG-controlled WK, obtained with Eq. (3), would have similar distributions across AGs. Namely, Eq. (3) mitigates the association between AG and WK. Eq. (3) is analogous to the variable control method of *partial* and *part correlation* (Stevens, 2012). We discuss their connection in Appendix B.2.

Measuring CFH. We measure CFH $\tilde{\mathbf{h}}(\cdot)$ with class-controlled features $X|Y$. Let us define a distance function.

D1) Distance function $\mathbf{d} : (V \times 2^V) \mapsto \mathbb{R}_{\geq 0}$:

$$\mathbf{d}(v_i, V') \coloneqq \frac{1}{|V'|} \sum_{v_j \in V'} \|(X_i \mid Y) - (X_j \mid Y)\|_2. \tag{4}$$

Recall that $V'_i = V \backslash \{v_i\}$. Given the distance function $\mathbf{d}(\cdot)$, we define ***homophily baseline*** $b(v_i) = \mathbf{d}(v_i, V'_i)$. Homophily baseline $b(v_i)$ can be interpreted as node v_i's expected (i.e., average) distance to random nodes or to its neighbors N_i when no A-X dependence is assumed.

Based on the distance functions, we define node pair-level, node-level, and graph-level CFHs as follows:

H1) Node pair-level CFH $\mathbf{h}^{(p)}_{ij}$:

$$\mathbf{h}^{(p)}_{ij} = \mathbf{h}((v_i, v_j)|X, Y, E) \coloneqq b(v_i) - \mathbf{d}(v_i, \{v_j\}). \tag{5}$$

H2) Node-level CFH $\mathbf{h}^{(v)}_i$:

$$\mathbf{h}^{(v)}_i = \mathbf{h}(v_i \mid X, Y, E) \coloneqq \frac{1}{|N_i|} \sum_{v_j \in N_i} \mathbf{h}^{(p)}_{ij}. \tag{6}$$

H3) Graph-level CFH $\mathbf{h}^{(G)}$:

$$\mathbf{h}^{(G)} = \mathbf{h}(G \mid X, Y, E) \coloneqq \frac{1}{|V|} \sum_{v_j \in V} \mathbf{h}^{(v)}_j. \tag{7}$$

Simply put, CFH $\mathbf{h}(\cdot)$ measures neighbor distance relative to homophily baseline $b(\cdot)$, and it meets the two design goals discussed in Sec. 3.1. With the homophily baseline $b(\cdot)$, $\mathbf{h}(\cdot)$ distinguishes homophily (positive dependence), heterophily (negative dependence), and impartiality (no dependence). At the same time, by measuring the distance with class-controlled features $X|Y$ (see Eq. (4)), CFH $\mathbf{h}(\cdot)$ mitigates potential confounding by node class.

Finally, we normalize $\mathbf{h}(\cdot)$ for good mathematical properties (Lemma 3.1- 3.3), which allows for its intuitive interpretation (discussed in Sec. 3.3). [1]

N1) Node-level normalization:

$$\tilde{\mathbf{h}}^{(v)}_i = \frac{\mathbf{h}^{(v)}_i}{\texttt{max}(b(v_i), \mathbf{d}(v_i, N_i))}. \tag{8}$$

N2) Graph-level normalization:

$$\tilde{\mathbf{h}}^{(G)} = \frac{\mathbf{h}^{(G)}}{\frac{1}{|V|} \texttt{max}(\sum_{v_i \in V} b(v_i), \sum_{v_i \in V} \mathbf{d}(v_i, N_i))}. \tag{9}$$

Lemma 3.1. *(Boundedness)* $\tilde{\mathbf{h}}^{(G)}$, $\tilde{\mathbf{h}}^{(v)}_i \in [-1, 1]$, *and the bound is tight, i.e.,* $\inf_G \tilde{\mathbf{h}}^{(G)} = -1$ *and* $\sup_G \tilde{\mathbf{h}}^{(G)} = 1$.

Lemma 3.2. *(Scale-Invariance)* $\tilde{\mathbf{h}}(v_i|\, X, \cdot) = \tilde{\mathbf{h}}(v_i|\, cX, \cdot)$ *and* $\tilde{\mathbf{h}}(G|\, X, \cdot) = \tilde{\mathbf{h}}(G|\, cX, \cdot)$, $\forall c \in \mathbb{R} \backslash \{0\}$.

Lemma 3.3. *(Monotonicity) Fix features of* $v_j \in V \backslash N_i$, $\tilde{\mathbf{h}}^{(v)}_i$ *is a monotonically decreasing function of* $\mathbf{d}(v_i, N_i)$.

All the proofs are in Appendix A.

[1] For completeness, if $b(\cdot) = 0$, we let $\tilde{\mathbf{h}}^{(v)}_i, \tilde{\mathbf{h}}^{(G)} = 0$.

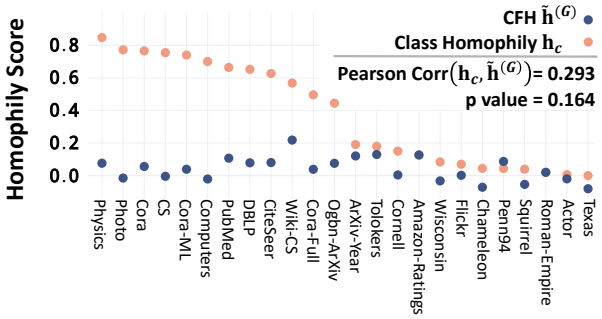


Figure 2: **Benchmark Graph Statistics.** Graph-level CFH scores $\tilde{\mathbf{h}}^{(G)}$ (*i*) are generally positive and small, with (*ii*) low correlation to class-homophily $\mathbf{h}_c$.

3.3. Measure Interpretation

We first focus on the node-level interpretation of CFH $\tilde{\mathbf{h}}(\cdot)$. Recall that $\mathbf{d}(v_i, N_i)$ and $b(v_i)$ respectively represent node v_i's distance to neighbors and random nodes.

Sign. Node-level CFH $\tilde{\mathbf{h}}_i^{(v)} > 0$ means that the node v_i is closer to its neighbors than to random nodes and, thus, *homophilic*. $\tilde{\mathbf{h}}_i^{(v)} < 0$ means that the node v_i is farther to its neighbors than to random nodes and, thus, *heterophilic*.

Zero. Node-level CFH $\tilde{\mathbf{h}}_i^{(v)} = 0$ indicates that the node v_i has the same distance to its neighbors and to random nodes, suggesting *impartiality* or *no* A-X *dependence*. Several different cases entail $\tilde{\mathbf{h}}_i^{(v)} = 0$ (in expectation). For example, (*i*) when the neighbors of v_i are chosen uniformly at random from all the other nodes regardless of their features or (*ii*) when all the nodes have the same feature, $\mathbb{E}[\tilde{\mathbf{h}}_i^{(v)}] = 0$.

Magnitude. Increasing a node v_i's distance to its neighbors reduces node-level CFH $\tilde{\mathbf{h}}_i^{(v)}$ (Lemma 3.3). We rephrase Eq. (8) as follows:

$$\tilde{\mathbf{h}}_i^{(v)} = \begin{cases} 1 - \frac{\mathbf{d}(v_i, N_i)}{b(v_i)}, & \text{if } \mathbf{d}(v_i, N_i) \le b(v_i), \\ \frac{b(v_i)}{\mathbf{d}(v_i, N_i)} - 1, & \text{if } \mathbf{d}(v_i, N_i) > b(v_i). \end{cases}$$

Intuitively, a node v_i is $\frac{|\tilde{\mathbf{h}}_i^{(v)}|}{1-|\tilde{\mathbf{h}}_i^{(v)}|}$ times closer (or farther) to its neighbors than to random nodes, if $\tilde{\mathbf{h}}_i^{(v)} > 0$ (or < 0).

Summary. In summary, for each node v_i, its distance to random nodes (i.e., $b(v_i)$) serves as an anchor to determine the sign and magnitude of its CFH $\tilde{\mathbf{h}}_i^{(v)}$, making it readily interpretable and comparable across different graphs.

Graph-level interpretation. A graph-level CFH $\tilde{\mathbf{h}}^{(G)}$ is an aggregation of node-level CFH $\mathbf{h}_i^{(v)}$'s. Details are in Appendix B.

3.4. Patterns in Benchmark Datasets

Here, we analyze node classification benchmark datasets using CFH $\tilde{\mathbf{h}}(\cdot)$. First, we measure graph-level CFH $\tilde{\mathbf{h}}^{(G)}$ in 24 datasets (Fig. 2). Most of the graphs (23 out of 24) have $\tilde{\mathbf{h}}^{(G)}$ scores below 0.13, and 16 graphs have positive $\tilde{\mathbf{h}}^{(G)}$ scores. Their mean $|\tilde{\mathbf{h}}^{(G)}|$ is 0.06. Recall that the full reachable range of $\tilde{\mathbf{h}}^{(G)}$ is $[-1, 1]$ (Lemma 3.1).

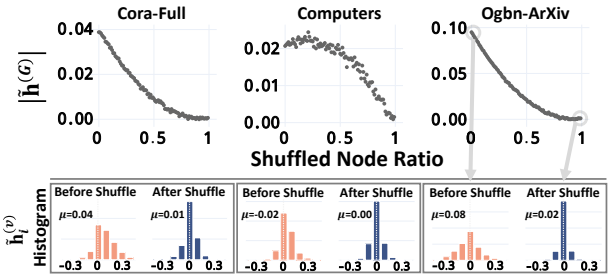


Figure 3: **The Effect of Feature Shuffle on CFH.** Both graph- and node-level CFH scores, $\tilde{\mathbf{h}}^{(G)}$ and $\tilde{\mathbf{h}}_i^{(v)}$, tend to approach zero over the feature shuffles.

Observation 1. *The real-world graph benchmarks tend to show small, positive CFH scores.*[2]

We further analyze the relation between CFH $\tilde{\mathbf{h}}^{(G)}$ and class-homophily $\mathbf{h}_c$ (Fig. 2). Surprisingly, their correlation is low (Pearson's $r = 0.293$, Kendall's $\tau = 0.196$) and not statistically significant (p value = 0.164 and 0.191, respectively).

Observation 2. *In the real-world graph benchmarks, CFH and class-homophily have a small, positive correlation.*

From Observations 1-2, we conclude that CFH $\tilde{\mathbf{h}}(\cdot)$ and class-homophily $\mathbf{h}_c$ show distinct patterns in the real-world benchmark graphs. We, thus, argue that investigating the impact of CFH $\tilde{\mathbf{h}}(\cdot)$ for GNNs has a unique significance.

Lastly, we examine how the feature shuffle (recall Fig. 1(a)) affects CFH. For graph-level CFH $\tilde{\mathbf{h}}^{(G)}$, increasing the shuffled node ratio reduces its magnitude $|\tilde{\mathbf{h}}^{(G)}|$ (Fig. 3). Also, the distribution of node-level CFH scores ($\tilde{\mathbf{h}}_i^{(v)}$'s) tends to center around 0 after the feature shuffle. We find similar results in 19 out of 24 datasets, while the remaining five do not fully obey the pattern.

Observation 3. *CFH scores tend to approach zero after shuffling the features of nodes from the same class.*

In later sections, Observation 3 serves to bridge a GNN theory and GNN performance in the real-world graphs, explicating the intriguing phenomenon (Fig. 1).

To further corroborate each Observation, we provide more in-depth analysis in Appendix C, together with dataset description and statistics.

[2] By small CFH scores, we mean the distances from nodes to their neighbors are *highly* close to their homophily baselines, numerically evidenced by the mean $|\tilde{\mathbf{h}}^{(G)}|$ of 0.06.

4. Graph Model and GNN Theory

We first address the second research question (**RQ2**) theoretically with a random graph model.

RQ2.1 [*Graph Model and Theory*]. How does A-X dependence affect graph convolution in a random graph model?

4.1. Graph Model: CSBM-X

CSBM-X overview. To control class-controlled feature homophily (CFH) $\tilde{\mathbf{h}}(\cdot)$ with a graph model, we propose CSBM-X. Compared to the previous CSBMs, CSBM-X is equipped with a new A-X dependence strength parameter τ. We provide a verbal description here, and its formal mathematical expression can be found in Appendix D.

CSBM-X uses $(n, \mu_0, \mu_1, \Sigma_0, \Sigma_1, d^+, d^-, \tau)$ as its parameters. It initializes n (assume even) number of nodes and equally divides them into two classes. For each node v_i, based on its class Y_i, CSBM-X samples its feature X_i from a Gaussian distribution with a mean vector μ_{Y_i} and a covariance matrix Σ_{Y_i} (i.e., $X_i = \mathcal{N}(\mu_{Y_i}, \Sigma_{Y_i})$).

Then, directed edges are sampled based on the node features X and classes Y, where parameter τ influences the sampling weights $\mathbb{P}_{ij}$'s. Specifically, CSBM-X first computes neighbor sampling weights $\mathbb{P}_{ij}$'s as follows:

$$\mathbb{P}_{ij} = \begin{cases} \dfrac{\exp(\tau \mathbf{h}_{ij}^{(p)})}{\sum_{t \in C^+_{Y_i} \setminus \{v_i\}} \exp(\tau \mathbf{h}_{it}^{(p)})} & \text{, if } Y_i = Y_j, \\ \dfrac{\exp(\tau \mathbf{h}_{ij}^{(p)})}{\sum_{t \in C^-_{Y_i}} \exp(\tau \mathbf{h}_{it}^{(p)})} & \text{, if } Y_i \neq Y_j. \end{cases}$$

A positive (or negative) τ exaggerates neighbor sampling weight $\mathbb{P}_{ij}$'s among the node pairs with higher (or lower) pair-level CFH $\mathbf{h}_{ij}^{(p)}$ (Eq. (5)). By the neighbor sampling weights $\mathbb{P}_{ij}$'s, for each node v_i, CSBM-X samples d^+ same-class (and d^- different-class) neighbors from the same-class node set $C^+_{Y_i} \setminus \{v_i\}$ (and different-class node set $C^-_{Y_i}$) without replacement. With its sampled nodes, neighbors, features, and classes, CSBM-X returns its graph $\mathcal{G} := \mathcal{G}(n, \mu_0, \mu_1, \Sigma_0, \Sigma_1, d^+, d^-, \tau) = (V, E, X, Y)$.

CSBM-X properties. The key innovation of CSBM-X involves satisfying good properties in controlling dependence among classes Y, features X, and graph topology A. First, the parameters $(\mu_0, \mu_1, \Sigma_0, \Sigma_1)$ control feature distance FD (Eq. (1); X-Y dependence). Second, the parameters (d^+, d^-) control $\mathbf{h}_c$ (Eq. (2); A-Y dependence). Last, the parameter τ controls CFH $\tilde{\mathbf{h}}(\cdot)$ (Eqs. (8)-(9); A-X dependence).

Existing CSBMs can also control X-Y and A-Y dependence (Deshpande et al., 2018; Abu-El-Haija et al., 2019; Chien et al., 2021; Palowitch et al., 2022; Baranwal et al., 2023; Luan et al., 2023; Wang et al., 2024). However, the proposed CSBM-X further controls A-X dependence (CFH $\tilde{\mathbf{h}}(\cdot)$), satisfying two additional good properties. [3]

Lemma 4.1 (τ controls CFH $\mathbf{h}(\cdot)$ *precisely*)**.** *Given* $0 < \max(d^+, d^-) < \frac{n}{2}$ *and fix the other parameters except for* τ*. (i)* $\mathbb{E}[\mathbf{h}^{(G)}]$ *strictly increases as* τ *increases. (ii) When* $\Sigma_0 = \Sigma_1 \neq \mathbf{0}$, $\mathbb{E}[\mathbf{h}^{(G)}] = 0$ *if and only if* $\tau = 0$.

Lemma 4.2 (τ controls CFH $\mathbf{h}(\cdot)$ *only*)**.** *Fix the other parameters except for* τ*, the FD and* $\mathbf{h}_c$ *of* $\mathcal{G}$ *are constant regardless of the value of* τ.

The proofs are in Appendix A. In concert, the above properties highlight that CSBM-X flexibly, yet precisely, controls the dependence among classes Y, features X, and topology A in the generated graph $\mathcal{G}$'s.

4.2. Graph Convolution in CSBM-X Graphs: Theory

In this section, we theoretically analyze the relationship between CFH $\tilde{\mathbf{h}}(\cdot)$ and graph convolution.

Analysis setting. For simplicity, we assume that the features are (*i*) 1-dimensional ($\mu_0, \mu_1, \Sigma_0, \Sigma_1 \in \mathbb{R}$) and (*ii*) symmetric with identical variances ($\mu_0 = -\mu_1 < 0$ and $\Sigma_0 = \Sigma_1 = 1$). We focus on an asymptotic setting with (*iii*) the number of nodes $n \to \infty$. (*iv*) The same- and different-class degree parameters respectively are $d^+ = np^+$ and $d^- = np^-$, with fixed $p^- \neq p^+ \in (0, \frac{1}{2})$.

Following some prior works on GNN theory (Wu et al., 2023; Luan et al., 2023), we define graph convolution as $D^{-1}AX$, a convolution of feature matrix X on an adjacency matrix A left-normalized by a (diagonal) degree matrix D.

Given the setting, after a graph convolution, the expected feature means of the two classes are constant and symmetric regardless of parameter τ. Specifically, the expected means are $\frac{d^- - d^+}{d^+ + d^-}\mu_1$ for class-0 and $\frac{d^+ - d^-}{d^+ + d^-}\mu_1$ for class-1. Thus, we consider a classifier $\mathcal{F}$ predicting node classes as follows:

$$\mathcal{F}(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Theoretical analysis. We analyze how parameter τ, controlling for CFH $\tilde{\mathbf{h}}(\cdot)$, affects the Bayes error rate of the classifier $\mathcal{F}$, given the *convoluted node features* (i.e., features after convolution $D^{-1}AX$). Formally, we denote the expected Bayes error rate of $\mathcal{F}$ for classifying the two classes in a CSBM-X graph $\mathcal{G}(\cdot, \tau)$ as $\mathcal{B}_{\mathcal{F}}(\mathcal{G}(\cdot, \tau))$.

Theorem 4.3. *Fix the other parameters except for* τ*, after a step of graph convolution,* $\mathcal{B}_{\mathcal{F}}(\mathcal{G}(\cdot, \tau))$ *is minimized at* $\tau = 0$ *and strictly increases as* $|\tau|$ *increases, i.e.,* $\arg\min_\tau \mathcal{B}_{\mathcal{F}}(\mathcal{G}(\cdot, \tau)) = 0$*;* $\mathcal{B}_{\mathcal{F}}(\mathcal{G}(\cdot, \tau_0)) < \mathcal{B}_{\mathcal{F}}(\mathcal{G}(\cdot, \tau_1))$ *for any* τ_0 *and* τ_1 *such that* $|\tau_0| < |\tau_1|$ *and* $\tau_0 \tau_1 > 0$.

[3] In theoretical statements, we focus on unnormalized CFH $\mathbf{h}(\cdot)$ for simplicity.

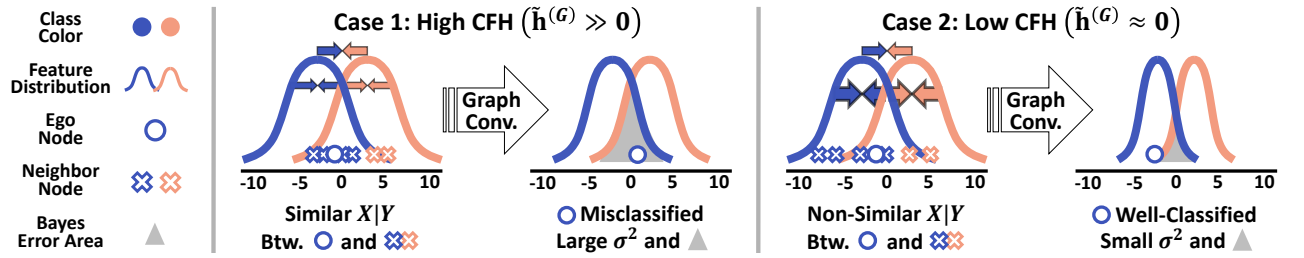


Figure 4: **Visual Intuition of Theorem 4.3.** When CFH is low ($\tilde{\mathbf{h}}^{(G)} \approx 0$), the feature distribution of each class shrinks faster (denoted by the arrows) by graph convolution, resulting in a lower Bayes error rate. Namely, the power to pull node features towards the feature mean of each class becomes stronger with decreasing $|\tilde{\mathbf{h}}^{(G)}|$.

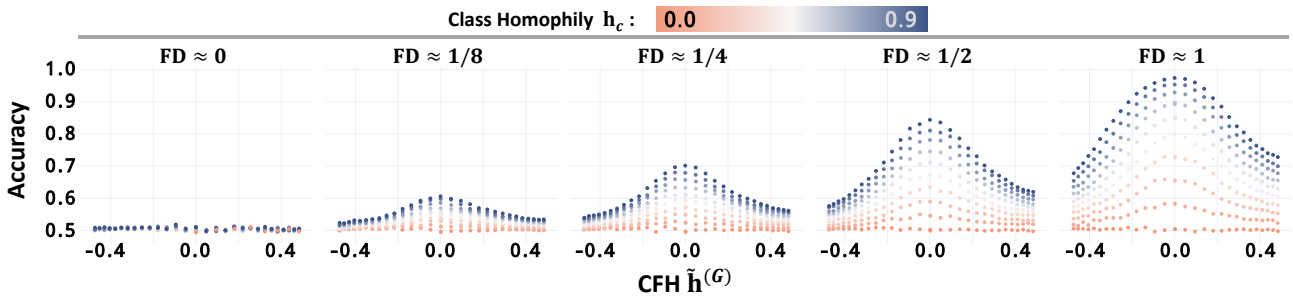


Figure 5: **The Simplified GNN Performance in CSBM-X Graphs.** Consistent with Theorem 4.3, for given feature distance FD > 0 and class homophily $\mathbf{h}_c > 0$, the simplified GNN performance increases as graph-level CFH $\tilde{\mathbf{h}}^{(G)} \to 0$ (i.e., $\tau \to 0$).

Proof sketch. WLOG, assume a node $v_i \in V$ is assigned to class-1 (i.e., $Y_i = 1$) and has class-controlled feature x_i (i.e., $X_i = \mu_1 + x_i$). [4]

We obtain closed-form formulae of the distributions of (*i*) edge sampling probabilities and, subsequently, (*ii*) neighbor sampling probabilities. This allows us to calculate the distribution of class-controlled, convoluted node features.

$$\mathbb{E}[x_N] = \int_{-\infty}^{\infty} x \Pr[x_N = x] dx$$
$$= \frac{\tau\left(\mathrm{erfc}\left(\frac{\tau - x_i}{\sqrt{2}}\right) - \exp(2\tau x_i)\mathrm{erfc}\left(\frac{\tau + x_i}{\sqrt{2}}\right)\right)}{\mathrm{erfc}\left(\frac{\tau - x_i}{\sqrt{2}}\right) + \exp(2\tau x_i)\mathrm{erfc}\left(\frac{\tau + x_i}{\sqrt{2}}\right)},$$

where x_N denotes the class-controlled feature of node v_i's neighbor and erfc denotes the complementary error function of the Gaussian error function.

Then, we obtain a closed-form formula of $\mathcal{B}_{\mathcal{F}}(\mathcal{G}(\cdot, \tau))$.

$$\mathcal{B}_{\mathcal{F}}(\mathcal{G}(\cdot, \tau)) = \int_{-\infty}^{\infty} \Pr[\mathcal{F}(x_i) = 0] \Pr[x_i \mid Y_i = 1] dx_i$$
$$\xrightarrow{n \to \infty} \int_{-\infty}^{\infty} \mathbf{1}[\mathbb{E}[x_N] < \frac{d^- - d^+}{d^- + d^+}\mu_1]\varphi[x_i] dx_i, \qquad (10)$$

where φ is the PDF of the standard Gaussian distribution. By analyzing the derivatives of the formulae, we show that $\mathcal{B}_{\mathcal{F}}(\mathcal{G}(\cdot, \tau))$ is minimized at $\tau = 0$ and increases as $|\tau|$ increases in both positive and negative directions. The full proof is in Appendix A. □

[4] Recall that class-controlled feature $x_i = (X_i|Y)$ in Eq. (3).

Summary. For each class ℓ's convoluted feature distribution, degree parameters (d^+, d^-) and the feature mean parameters (μ_0, μ_1) determine its mean, while A-X dependence strength parameter τ determines the distance between each node and the mean (or the distribution variance; see Eq. (10) and Fig. 4). Thus, the simplified GNN's Bayes error rate $\mathcal{B}_{\mathcal{F}}(\mathcal{G}(\cdot, \tau))$ decreases as $|\tau|$ decreases, reaching its minimum at $\tau = 0$ (Theorem 4.3). [5] With Lemma 4.1, we conclude that $\mathcal{B}_{\mathcal{F}}(\mathcal{G}(\cdot, \tau))$ is the lowest when CFH $\mathbf{h}^{(G)}$ is zero (i.e., no A-X dependence).

4.3. Empirical Elaboration on Theory

Here, we empirically validate and elaborate on Theorem 4.3.

Experiment setting. We generate CSBM-X graphs with various parameter configurations. We fix the number of nodes $n = 10000$ and node degree $(d^+ + d^-) = 20$, assuming sparse graph topology. The features in each class are sampled from a Gaussian distribution. The generated graphs have a wide range of feature distance FD, class-homophily $\mathbf{h}_c$, and CFH $\tilde{\mathbf{h}}(\cdot)$.

- FD: $|\mu_0 - \mu_1| \in \{0, 1/8, 1/4, 1/2, 1\}$; $\Sigma_0 = \Sigma_1 = 1$
- $\mathbf{h}_c$: $d^+ \in \{10, 11, \ldots, 18, 19\}$; $d^- = 20 - d^+$,
- $\tilde{\mathbf{h}}(\cdot)$: $\tau \in \{-1.5, -1.4, \ldots, -0.1, 0, 0.1, \ldots, 1.4, 1.5\}$.

[5] The graph convolution followed by the defined classifier $\mathcal{F}$ can be considered a simplified GNN (Wu et al., 2019).

On the CSBM-X graphs $\mathcal{G}$'s, we train a simplified GNN $D^{-1}AXW$, where $W \in \mathbb{R}$ is a learnable parameter. We report the test accuracy averaged over 5 trials, each with a train/val/test split of 50/25/25.

Finding 1 (*Effect of* $\tilde{\mathbf{h}}(\cdot)$). As shown in Fig. 5, given class-homophily $\mathbf{h}_c > 0$ (i.e., degree parameters $d^+ > d^-$) and feature distance FD > 0 (i.e., feature mean parameters $\mu_0 \neq \mu_1$), the simplified GNN achieves the best accuracy when graph-level CFH $\tilde{\mathbf{h}}^{(G)} \approx 0$, and the accuracy decreases as $|\tilde{\mathbf{h}}^{(G)}|$ increases, in both positive and negative directions.

Finding 2 (*Interplay among* FD, $\mathbf{h}_c$, *and* $\tilde{\mathbf{h}}(\cdot)$). Aligned with our theoretical outcomes (Eq. (10), Fig. 4), class-homophily $\mathbf{h}_c$ and feature distance FD moderate the beneficial effect of small $\tilde{\mathbf{h}}(\cdot)$ (Fig. 5). For understanding, recall that our theoretical findings roughly indicate that FD and $\mathbf{h}_c$ affect the mean, whereas $\tilde{\mathbf{h}}(\cdot)$ the variance, of the convoluted feature distribution of each class. Intuitively, consider the two cases below.

If feature distance FD and class homophily $\mathbf{h}_c$ are moderate-sized, the convoluted feature means of the two classes would be *somewhat distant*. Small $\tilde{\mathbf{h}}(\cdot)$, then, can significantly benefit GNNs, since small variances of the two feature distributions would markedly reduce their overlap. A very small (or large) FD and $\mathbf{h}_c$, on the contrary, would cause the convoluted feature distributions to be *too close* (or *too distant*). Then, reducing variances of the convoluted feature distributions may not significantly improve GNNs, mitigating the beneficial effect of $\tilde{\mathbf{h}}(\cdot)$.

In conclusion, the empirical outcomes are highly consistent with Theorem 4.3. In Appendix D, we report consistent results with (*i*) two graph convolution layers, (*ii*) symmetrically normalized graph convolution, (*iii*) high-dimensional X, (*iv*) imbalanced variances $\Sigma_0 \neq \Sigma_1$, (*v*) larger A-X dependence strength $|\tau|$'s, and (*vi*) a more complex version of CSBM-X reflecting a power-law degree distribution.

5. Feature Shuffle in Real-World Graphs

In this section, we finalize our investigation of the second research question (**RQ2**) with the feature shuffle.

RQ2.2 [*Feature Shuffle*]. In real-world graphs, how does reducing A-X dependence with feature shuffle affect GNNs?

5.1. Experiment Setting

For each class, we randomly choose the nodes to be shuffled by a given shuffled node ratio $\in \{0.00, 0.01, ..., 1.00\}$. For the chosen same-class nodes, their feature vectors are shuffled randomly. To ensure that the train/val/test split is not affected, shuffle is done only within the same split.

The feature shuffle can reduce A-X dependence (i.e., CFH $\tilde{\mathbf{h}}(\cdot)$; Observation 3) without perturbing X-Y and A-Y dependence, providing a suitable experimental setting to answer **RQ2**. Namely, the feature shuffle serves to generate synthetic versions of the benchmark graphs.

For each shuffled graph, we initialize, train, and evaluate a GNN model. We report the mean test accuracy over 5 trials, with a train/val/test split of 50/25/25. For the GNN model, we use GCNII (Chen et al., 2020), GPR-GNN (Chien et al., 2021), and AERO-GNN (Lee et al., 2023). We mainly use GCNII due to its (*i*) non-adaptive graph convolution layer and (*ii*) empirical strengths in both high and low class-homophily $\mathbf{h}_c$ graphs. For more training details, refer to Appendix G.

5.2. Connecting Theory and Real-World Graphs

High class-homophily graphs. As shown in Fig. 6, in all 12 high class-homophily $\mathbf{h}_c$ benchmark datasets, GCNII performance improves consistently over increasing shuffled node ratio (the mean increase of 4%p). The largest performance gain is 10%p in the Cora-Full dataset.

Low class-homophily graphs. Meanwhile, in low class-homophily $\mathbf{h}_c$ benchmark datasets, GCNII shows small to no performance improvement in 11/12 datasets (the mean increase of 0.5%p; Fig. 7). As demonstrated with CSBM-X (Fig. 5), low class-homophily $\mathbf{h}_c$ reduces the beneficial effect of small A-X dependence in real-world graphs. Unexpectedly, in the Roman-Empire dataset, GCNII suffers from a steady performance decline over increasing shuffled node ratio. The reason may relate to its abnormally large diameter of 6,824. We provide an in-depth analysis in Appendix C.4.

The role of FD. Increasing feature noise generally decreases feature distance FD. Specifically, we randomly chose 50% of all nodes and randomly permuted the feature vectors *irrespective of their classes* to add such noise. Fig. 8 shows that, after adding the noise, the slope of performance over the feature shuffles becomes smaller, suggesting that significantly increasing the feature noise reduces the beneficial effect of the feature shuffle. The finding echoes the results from the CSBM-X experiment (Fig. 5), such that FD moderates the beneficial effect of low A-X dependence.

Other GNN architectures. We use other GNN architectures to test if the effect of the feature shuffle relies on GNN architecture choice. Specifically, we use GPR-GNN, a decoupled GNN with an adaptive graph convolution. For an attention-based GNN, we use AERO-GNN, capable of stacking deep layers. In the considered models, the trends are similar to those of GCNII (Fig. 9), suggesting that the other GNN architectures also leverage small A-X dependence to improve their prediction. We also found consistent results with transformer- and neural-ODE-based GNNs (Gravina et al., 2023; Deng et al., 2024).

Proximity-based features. GNN node classification performance often degrades when using proximity-based information as the *only* node features (Duong et al., 2019; Cui et al., 2022; Lee et al., 2024). We find consistent results. However, we are astonished to find that, after the feature shuffle, a GNN trained with proximity-based features can be as competitive as the one trained with the original features (Fig. 10). The results highlight that, given some feature distance FD and class-homophily $\mathbf{h}_c$, reducing A-X dependence can improve GNN regardless of the feature types.

In summary, the results with the real-world graphs and advanced GNNs are well-aligned with the theoretical results, underscoring the validity and generalizability of the proposed theory. In Appendix E, we further report consistent results with lower train node ratios.

6. Discussion

In this work, we analyze the impact of A-X dependence (i.e., dependence between graph topology and node features) on GNNs with (*i*) a class-controlled feature homophily (CFH) measure $\tilde{\mathbf{h}}(\cdot)$, (*ii*) a random graph model CSBM-X, and (*iii*) the feature shuffle. In both CSBM-X and the real-world graphs, we demonstrate that A-X dependence, measured by CFH, significantly influences GNN performance.

GNN theory: the prior literature. The early studies found some failure cases of GNNs. NT & Maehara (2019) claimed that a graph convolution layer is simply a low-pass filter for node features. Under noisy features and non-linear feature spaces, they showed that GNN-based node classification may readily become ineffective. Oono & Suzuki (2020) further showed that over-smoothing of node features in GNNs may inevitably occur at infinite model depth.

The following works analyzed how GNNs behave at shallower model depths, demonstrating that the effect of graph convolution depends on feature informativeness, class-homophily, and node degree. Baranwal et al. (2021) focused on how they let GNNs obtain more linearly separable features for each class. Wei et al. (2022) studied how the factors interact with GNNs' non-linearity, and Wu et al. (2023) investigated their role in triggering over-smoothing.

Aligned with the theory, low class-homophily (often just called heterophily) has received significant attention as GNNs' 'nightmare.' A stream of empirical findings continued to show that GNN performance drops significantly in low class-homophily benchmark datasets (Pei et al., 2020; Zhu et al., 2020; 2021; Chien et al., 2021), and some works investigated the relationship between low class-homophily and over-smoothing (Bodnar et al., 2022; Yan et al., 2022).

However, studies began to discover that low class-homophily, per se, does not deteriorate GNN performance. Ma et al. (2022) and Platonov et al. (2023a) demonstrated that as long as the class distribution is informative w.r.t. node class, GNNs can effectively perform node classification even with low class-homophily.

Recently, studies have delved into how mesoscopic patterns of class-homophily affect GNNs. Luan et al. (2023) (roughly) argued that, for GNNs to well-classify a node class, its 'intra-class distance' should be smaller than the 'inter-class distance' after graph convolution. That is, low class-homophily may trigger the 'inter-class distance' to be smaller to degrade GNN performance. Mao et al. (2023) investigated mixed patterns of class-homophily and heterophily, showing that GNNs better classify the nodes with the majority pattern in the mixture. Lastly, Wang et al. (2024) investigated an array of low class-homophily patterns and showed that there exist good, mixed, and bad patterns for GNNs to learn from.

GNN theory: the present work. Not to mention that the role of A-X dependence (i.e., CFH $\tilde{\mathbf{h}}(\cdot)$) has not been adequately addressed by the prior literature, the present work can also be interpreted as an extension of the works on homophily-GNN connection to continuous feature domain. Intuitively, a large homophily slows feature mixing by graph convolution, and a small homophily accelerates it. From such a perspective, our conclusion that CFH should ideally be small, while class-homophily be large, is an intuitive outcome. To better classify node classes, the mixing between classes should occur at a slow rate, whereas the mixing within each class should occur faster.

To conclude, we argue that CFH mediates the effect of graph convolution by moderating the force to pull each node feature toward the feature mean of the respective node class. From the node classification perspective, even with high class-homophily and informative features, a large CFH can result in degraded GNN performance (Fig. 5).

The central implications are two-fold. In hindsight, our findings in concert suggest that the recent success of GNNs may have relied on the generally small CFH of the benchmark datasets. Looking forward, investigating the role of CFH on GNNs is a promising research direction.

Limitations and future works. We close with our discussion on limitations and potential research directions. First, we did not provide new benchmark datasets or GNNs that address varying levels of CFH. The current benchmark datasets generally have low and positive CFH (see Observation 1). According to our findings, the existing GNNs may significantly underperform in datasets with large CFH. Thus,

[6] CML, SQR, RM-Emp, and AMZ-Rts respectively stand for Chameleon, Squirrel, Roman-Empire, and Amazon-Ratings.

[7] Out-of-memory occurs when using adjacency matrix as the node features for Ogbn-ArXiv.

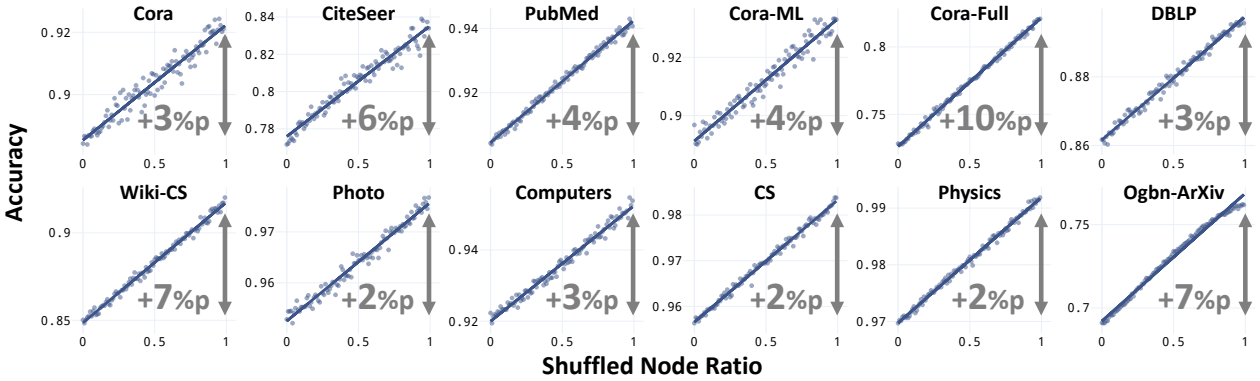


Figure 6: **GNN Performance After the Feature Shuffles: *High Class-Homophily*.** Consistent with the findings from CSBM-X (Theorem 4.3; Fig. 5; high $\mathbf{h}_c$ case), reducing A-X dependence with the feature shuffle improves GNN performance consistently and significantly in high class-homophily graphs.

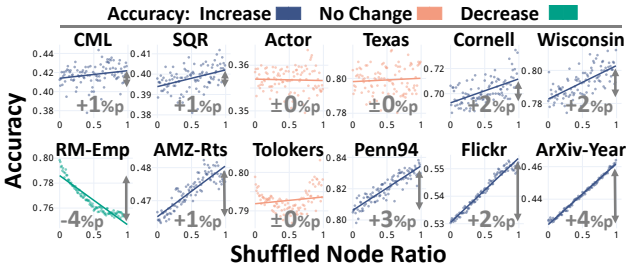


Figure 7: **GNN Performance After the Feature Shuffles: *Low Class-Homophily*.** Consistent with the findings from CSBM-X (Theorem 4.3; Fig. 5; low $\mathbf{h}_c$ case), the effect of the feature shuffle is smaller when class-homophily is low. [6]

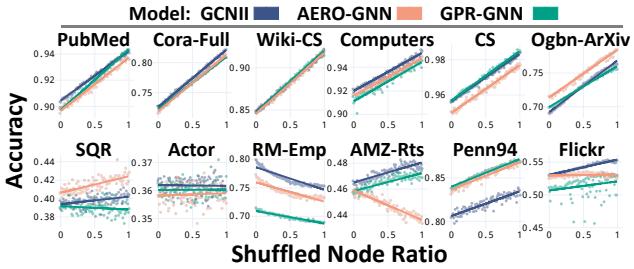


Figure 9: **GNN Performance After the Feature Shuffles: *Different Models*.** The adaptive convolution- and attention-based GNNs (GPR-GNN and AERO-GNN, respectively) generally show similar trends with GCNII.

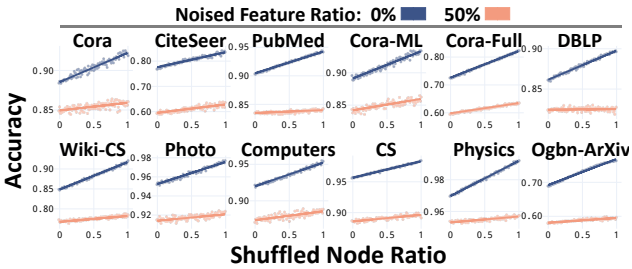


Figure 8: **GNN Performance After the Feature Shuffles: *Noisy Features*.** Consistent with the findings from CSBM-X (Theorem 4.3; Fig. 5; low FD case), the effect of the feature shuffle is smaller with noisy features.

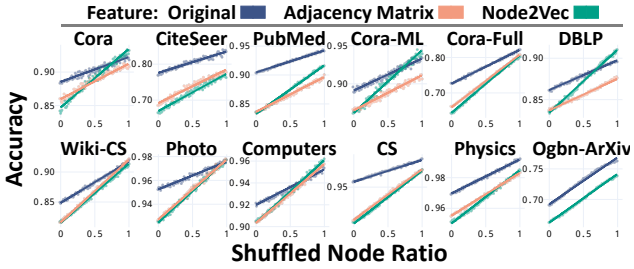


Figure 10: **GNN Performance After the Feature Shuffles: *Proximity-Based Features*.** Regardless of the feature types, the feature shuffle improves GNN node classification performance. [7]

proposing new datasets with a large and/or negative CFH and designing methods for such datasets would be valuable contributions.

Second, we did not explore potential applications of our findings. Low CFH can significantly contribute to GNN performance (see Fig. 6). The feature shuffle algorithm in Sec. 5, however, requires test labels, which are not known, and CFH values are also not known without test labels. Thus, a method that estimates and adaptively lowers CFH, while keeping class-homophily and feature informativeness intact, may substantially improve GNN performance. We explore a potential application in Appendix F.

Last, generalization of our theoretical findings is limited since both CFH measure $\tilde{\mathbf{h}}(\cdot)$ and CSBM-X, implicitly and explicitly, assume that the relation between node features and class is linear (see Eq. (3)) and that the node-level CFH distribution is symmetric and unimodal (see Eq. (7)). However, the patterns in the real-world graphs should be more complex. Exploring how more realistic patterns interact with GNNs would be a valuable next step.

Impact statement

We do not expect any immediate, negative societal impact of the present work. It delves into a theory of graph neural networks and, thereby, may indirectly benefit their applications to be more effective and reliable.

Acknowledgements

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00871, Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration) (No. 2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)). This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2024-00341425).

References

Abboud, R., Ceylan, I. I., Grohe, M., and Lukasiewicz, T. The surprising power of graph neural networks with random node initialization. In *IJCAI*, 2021.

Abu-El-Haija, S., Perozzi, B., Kapoor, A., Alipourfard, N., Lerman, K., Harutyunyan, H., Ver Steeg, G., and Galstyan, A. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *ICML*, 2019.

Baranwal, A., Fountoulakis, K., and Jagannath, A. Graph convolution for semi-supervised classification: Improved linear separability and out-of-distribution generalization. In *ICML*, 2021.

Baranwal, A., Fountoulakis, K., and Jagannath, A. Effects of graph convolutions in multi-layer networks. In *ICLR*, 2023.

Bodnar, C., Di Giovanni, F., Chamberlain, B., Liò, P., and Bronstein, M. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns. In *NeurIPS*, 2022.

Bojchevski, A. and Günnemann, S. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *ICLR*, 2018.

Chen, M., Wei, Z., Huang, Z., Ding, B., and Li, Y. Simple and deep graph convolutional networks. In *ICML*, 2020.

Chien, E., Peng, J., Li, P., and Milenkovic, O. Adaptive universal generalized pageRank graph neural network. In *ICLR*, 2021.

Cui, H., Lu, Z., Li, P., and Yang, C. On positional and structural node features for graph neural networks on non-attributed graphs. In *CIKM*, 2022.

Deng, C., Yue, Z., and Zhang, Z. Polynormer: Polynomial-expressive graph transformer in linear time. In *ICLR*, 2024.

Deshpande, Y., Montanari, A., Mossel, E., and Sen, S. Contextual stochastic block models. In *NeurIPS*, 2018.

Duong, C. T., Hoang, T. D., Dang, H. T. H., Nguyen, Q. V. H., and Aberer, K. On node features for graph neural networks. *arXiv preprint arXiv:1911.08795*, 2019.

Gravina, A., Bacciu, D., and Gallicchio, C. Anti-symmetric dgn: A stable architecture for deep graph networks. In *ICLR*, 2023.

Grover, A. and Leskovec, J. Node2vec: Scalable feature learning for networks. In *KDD*, 2016.

Holland, P. W., Laskey, K. B., and Leinhardt, S. Stochastic blockmodels: First steps. *Social Networks*, 5(2), 1983.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*, 2020.

Kailath, T. The divergence and bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, 15(1):52–60, 1967.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Lee, G., Lee, S. Y., and Shin, K. Villain: Self-supervised learning on hypergraphs without features via virtual label propagation. In *WWW*, 2024.

Lee, S. Y., Bu, F., Yoo, J., and Shin, K. Towards deep attention in graph neural networks: Problems and remedies. In *ICML*, 2023.

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. In *NeurIPS*, 2021.

Luan, S., Hua, C., Lu, Q., Zhu, J., Zhao, M., Zhang, S., Chang, X. W., and Precup, D. Revisiting heterophily for graph neural networks. In *NeurIPS*, 2022.

Luan, S., Hua, C., Xu, M., Lu, Q., Zhu, J., Chang, X.-W., Fu, J., Leskovec, J., and Precup, D. When do graph neural networks help with node classification? Investigating the impact of homophily principle on node distinguishability. In *NeurIPS*, 2023.

Ma, Y., Liu, X., Zhao, T., Liu, Y., Tang, J., and Shah, N. A unified view on graph neural networks as graph signal denoising. In *CIKM*, 2021.

Ma, Y., Liu, X., Shah, N., and Tang, J. Is homophily a necessity for graph neural networks? In *ICLR*, 2022.

Mao, H., Chen, Z., Jin, W., Han, H., Ma, Y., Zhao, T., Shah, N., and Tang, J. Demystifying structural disparity in graph neural networks: Can one size fit all? *arXiv preprint arXiv:2306.01323*, 2023.

McPherson, M., Smith-Lovin, L., and Cook, J. M. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

Mernyei, P. and Cangea, C. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901*, 2020.

NT, H. and Maehara, T. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019.

Oono, K. and Suzuki, T. Graph neural networks exponentially lose expressive power for node classification. In *ICLR*, 2020.

Palowitch, J., Tsitsulin, A., Mayer, B., and Perozzi, B. Graphworld: Fake graphs bring real insights for gnns. In *KDD*, 2022.

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

Platonov, O., Kuznedelev, D., Babenko, A., and Prokhorenkova, L. Characterizing graph datasets for node classification: Homophily-heterophily dichotomy and beyond. In *NeurIPS*, 2023a.

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 *ICLR*, 2023b.

Rogers, E. M., Singhal, A., and Quinlan, M. M. Diffusion of innovations. In *An integrated approach to communication theory and research*, pp. 432–448. Routledge, 2014.

Shchur, O., Mumme, M., Bojchevski, A., and Günnemann, S. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

Stevens, J. P. *Applied multivariate statistics for the social sciences*. Routledge, 2012.

Tang, J., Sun, J., Wang, C., and Yang, Z. Social influence analysis in large-scale networks. In *KDD*, 2009.

Wang, J., Guo, Y., Yang, L., and Wang, Y. Understanding heterophily for graph neural networks. *arXiv preprint arXiv:2401.09125*, 2024.

Wang, X. and Zhang, M. How powerful are spectral graph neural networks. In *ICML*, 2022.

Wei, R., Yin, H., Jia, J., Benson, A. R., and Li, P. Understanding non-linearity in graph neural networks from the bayesian-inference perspective. In *NeurIPS*, 2022.

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., and Weinberger, K. Simplifying graph convolutional networks. In *ICML*, 2019.

Wu, X., Chen, Z., Wang, W., and Jadbabaie, A. A non-asymptotic analysis of oversmoothing in graph neural networks. In *ICLR*, 2023.

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 *ICDM*, 2022.

Yang, Z., Cohen, W., and Salakhudinov, R. Revisiting semi-supervised learning with graph embeddings. In *ICML*, 2016.

You, J., Gomes-Selman, J. M., Ying, R., and Leskovec, J. Identity-aware graph neural networks. In *AAAI*, 2021.

Zeng, H., Zhou, H., Srivastava, A., Kannan, R., and Prasanna, V. Graphsaint: Graph sampling based inductive learning method. In *ICML*, 2020.

Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., and Koutra, D. Beyond homophily in graph neural networks: Current limitations and effective designs. In *NeurIPS*, 2020.

Zhu, J., Rossi, R. A., Rao, A., Mai, T., Lipka, N., Ahmed, N. K., and Koutra, D. Graph neural networks with heterophily. In *AAAI*, 2021.

A. Proofs and Additional Theoretical Results

A.1. Proofs for Measure $\tilde{\mathrm{h}}(\cdot)$

Throughout our proof w.r.t. measure $\tilde{\mathbf{h}}(\cdot)$, let us assume that a graph G has no isolated nodes. Also, recall that if $b(\cdot) = 0$ (i.e., all nodes have the same class-controlled features), we define $\tilde{\mathbf{h}}_i^{(v)}$ and $\tilde{\mathbf{h}}^{(G)}$ as 0.

Proof of Lemma 3.1 (Boundedness).

Proof. ***Bound of*** $\tilde{\mathbf{h}}_i^{(v)}$. The node-level CFH $\tilde{\mathbf{h}}_i^{(v)}$ is defined as follows:

$$\tilde{\mathbf{h}}_i^{(v)} = \frac{\mathbf{h}_i^{(v)}}{\max(b(v_i), \mathbf{d}(v_i, N_i))} = \frac{b(v_i) - \mathbf{d}(v_i, N_i)}{\max(b(v_i), \mathbf{d}(v_i, N_i))}.$$

L2 norm is non-negative, and thus, both $\mathbf{d}(\cdot), b(\cdot)$ are non-negative. Since $|b(v_i) - \mathbf{d}(v_i, N_i)| \leq \max(b(v_i), \mathbf{d}(v_i, N_i))$, $\tilde{\mathbf{h}}_i^{(v)} \in [-1, 1]$ holds, completing the proof of bound for node-level CFH $\tilde{\mathbf{h}}_i^{(v)}$. □

Proof. ***Bound of*** $\tilde{\mathbf{h}}^{(G)}$. The graph-level CFH $\tilde{\mathbf{h}}^{(G)}$ can be rewritten as:

$$\begin{aligned}
\tilde{\mathbf{h}}^{(G)} &= \frac{\mathbf{h}^{(G)}}{\frac{1}{|V|}\max(\sum_{v_i \in V} b(v_i), \sum_{v_i \in V} \mathbf{d}(v_i, N_i))} \\
&= \frac{\frac{1}{|V|}\sum_{v_i \in V} \mathbf{h}_i^{(v)}}{\frac{1}{|V|}\max(\sum_{v_i \in V} b(v_i), \sum_{v_i \in V} \mathbf{d}(v_i, N_i))} \\
&= \frac{\sum_{v_i \in V} b(v_i) - \sum_{v_i \in V} \mathbf{d}(v_i, N_i)}{\max(\sum_{v_i \in V} b(v_i), \sum_{v_i \in V} \mathbf{d}(v_i, N_i))}.
\end{aligned}$$

For the same reason as $\tilde{\mathbf{h}}_i^{(v)}$, $\tilde{\mathbf{h}}^{(G)} \in [-1, 1]$, completing the proof of bound for graph-level CFH $\tilde{\mathbf{h}}^{(G)}$. □

Proof. ***Existence Claim***. We show that the upper/lower bound is achievable under a non-asymptotic/asymptotic setting. First, we show that $\sup_G \tilde{\mathbf{h}}^{(G)} = 1$ holds. Consider a disconnected G such that class-controlled features of neighboring nodes are all equal (i.e., $X_i|Y = X_j|Y, \forall (v_i, v_j) \in E$), while that of disconnected nodes are different (i.e., $X_k|Y \neq X_\ell|Y$, where there does not exist a path between v_k and v_ℓ). In such a case, $b(v_i) \neq 0$ and $\mathbf{d}(v_i, N_i) = 0$ hold $\forall v_i \in V$. Thus, $\tilde{\mathbf{h}}_i^{(v)} = 1, \forall v_i \in V$ also holds, and consequently, $\max_G \tilde{\mathbf{h}}^{(G)} = \sup_G \tilde{\mathbf{h}}^{(G)} = 1$ holds.

Second, we show that $\inf_G \tilde{\mathbf{h}}^{(G)} = -1$ holds. Consider a case where $\mathbf{d}(v_i, N_i) \to \infty$ and $o(b(v_i)) < o(\mathbf{d}(v_i, N_i)), \forall v_i \in V$ hold. In such a case, the following holds:

$$\lim_{\mathbf{d}(v_i, N_i) \to \infty} \tilde{\mathbf{h}}_i^{(v)} = \frac{b(v_i) - \mathbf{d}(v_i, N_i)}{\mathbf{d}(v_i, N_i)} \equiv -\frac{\mathbf{d}(v_i, N_i)}{\mathbf{d}(v_i, N_i)} = -1, \forall v_i \in V. \tag{11}$$

Consequently, $\inf_G \tilde{\mathbf{h}}^{(G)} = -1$ hold. Note that the second result is derived under the asymptotic scenario, and thus, the result does not indicate the exact minimum. □

Proof of Lemma 3.2 (Scale-Invariance).

Proof. ***Scale-Invariance of*** $\tilde{\mathbf{h}}_i^{(v)}$. Denote the distance function (Eq. (4)) with node feature X as $\mathbf{d}'(v_i, V_i', X)$. Then, for any $c \in \mathbb{R} \backslash \{0\}$, the following holds:

$$\begin{aligned}
\mathbf{d}'(v_i, V_i', cX) &\coloneqq \frac{1}{|V_i'|} \sum_{v_j \in V_i'} \|(c \cdot X_i|Y) - (c \cdot X_j|Y)\|_2 \\
&= |c| \cdot \left(\frac{1}{|V_i'|} \sum_{v_j \in V_i'} \|(X_i|Y) - (X_j|Y)\|_2 \right) \\
&= |c| \cdot \mathbf{d}'(v_i, V_i', X).
\end{aligned}$$

Likewise, we denote a homophily baseline $b(v_i)$ with a node feature X as $b'(v_i, X)$. Then, since $b(v_i, cX)$ is a special case of Eq. (4), the following holds: $b'(v_i, cX) = |c|\, b'(v_i, X)$. Lastly, we denote $\tilde{\mathbf{h}}_i^{(v)}$ with a node feature X as $\tilde{\mathbf{h}}_i^{(v)}(X)$. Then, by showing the below, we finalize the proof for node-level CFH $\tilde{\mathbf{h}}_i^{(v)}$.

$$\begin{aligned}\tilde{\mathbf{h}}_i^{(v)}(cX) &= \frac{b'(v_i)(cX) - \mathbf{d}'(v_i, N_i, cX)}{\max(b'(v_i, cX), \mathbf{d}'(v_i, N_i, cX))} \\ &= \frac{|c| \cdot (b'(v_i, X) - \mathbf{d}'(v_i, N_i, X))}{|c| \cdot (\max(b'(v_i, X), \mathbf{d}'(v_i, N_i, X)))} \\ &= \frac{b'(v_i, X) - \mathbf{d}'(v_i, N_i, X)}{\max(b'(v_i, X), \mathbf{d}'(v_i, N_i, X))} = \tilde{\mathbf{h}}_i^{(v)}(X).\end{aligned}$$

□

Proof. ***Scale-Invariance of*** $\tilde{\mathbf{h}}^{(G)}$. We denote $\tilde{\mathbf{h}}^{(G)}$ with a node feature X as $\tilde{\mathbf{h}}^{(G)}(X)$. Then, we finalize the proof for graph-level CFH $\tilde{\mathbf{h}}^{(G)}$ by extending the above results.

$$\begin{aligned}\tilde{\mathbf{h}}^{(G)}(cX) &= \frac{\sum_{v_i \in V} b(v_i, cX) - \sum_{v_i \in V} \mathbf{d}(v_i, N_i, cX)}{\max(\sum_{v_i \in V} b(v_i, cX), \sum_{v_i \in V} \mathbf{d}(v_i, N_i, cX))} \\ &= \frac{|c| \cdot (\sum_{v_i \in V} b(v_i, X) - \sum_{v_i \in V} \mathbf{d}(v_i, N_i, X))}{|c| \cdot (\max(\sum_{v_i \in V} b(v_i, X), \sum_{v_i \in V} \mathbf{d}(v_i, N_i, X)))} \\ &= \frac{\sum_{v_i \in V} b'(v_i, X) - \sum_{v_i \in V} \mathbf{d}'(v_i, N_i, X)}{\max(\sum_{v_i \in V} b'(v_i, X), \sum_{v_i \in V} \mathbf{d}'(v_i, N_i, X))} = \tilde{\mathbf{h}}^{(G)}(X).\end{aligned}$$

□

Proof of Lemma 3.3 (Monotonicity).

Proof. First, since node features of $v_k \in V \backslash N_i$ are fixed, we rewrite homophily baseline $b(v_i)$ as $b(v_i) = \frac{|N_i|}{|V_i'|}\mathbf{d}(v_i, N_i) + C$, where C is a fixed constant. For simplicity, denote $\mathbf{d}(v_i, N_i)$ and $\frac{|N_i|}{|V_i'|}$ as K and a, respectively. Thus, the following holds: $b(v_i) := aK + C$. We break the rest of the proof down into two parts.

Case 1: $b(v_i) \geq \mathbf{d}(v_i, N_i)$. Node-level CFH $\tilde{\mathbf{h}}_i^{(v)}$ can be rewritten as

$$\begin{aligned}\tilde{\mathbf{h}}_i^{(v)} &= \frac{b(v_i) - \mathbf{d}(v_i, N_i)}{b(v_i)} = \frac{(a-1)K + C}{aK + C} \\ \frac{\partial \tilde{\mathbf{h}}_i^{(v)}}{\partial K} &= \frac{-1}{(aK + C)^2} < 0.\end{aligned} \tag{12}$$

Case 2: $b(v_i) < \mathbf{d}(v_i, N_i)$. Node-level CFH $\tilde{\mathbf{h}}_i^{(v)}$ can be rewritten as

$$\begin{aligned}\tilde{\mathbf{h}}_i^{(v)} &= \frac{b(v_i) - \mathbf{d}(v_i, N_i)}{\mathbf{d}(v_i, N_i)} = \frac{(a-1)K + C}{K} \\ \frac{\partial \tilde{\mathbf{h}}_i^{(v)}}{\partial K} &= \frac{a-2}{K^2} < 0, \quad \because a < 1.\end{aligned} \tag{13}$$

By merging the result of Eq (12) and Eq (13), the monotonic decreasing property is guaranteed. □

A.2. Proofs for CSBM-X Properties

Proof of Lemma 4.1 (τ controls CFH $\mathrm{h}(\cdot)$ *precisely*).

Proof. **Regarding claim (i).** When the other parameters are fixed, for each $\mathcal{X} = (X_i)_{i \in [n]} \in \mathbb{R}^{n \times k}$ and $\mathcal{Y} = (Y_i)_{i \in [n]} \in [c]^n$. The joint probability $\Pr[(\mathcal{X}, \mathcal{Y})]$ is fixed regardless of the value of τ. Moreover, for each node v_i, the numbers of

same-class and different-class neighbors are fixed. Now, let us fix any $\mathcal{X}$ and $\mathcal{Y}$, it suffices to show for each node v_i, $\mathbb{E}[\mathbf{h}_i^{(v)} \mid \tau_1] < \mathbb{E}[\mathbf{h}_i^{(v)} \mid \tau_2]$.

To see this, first, $\mathbf{h}_i^{(v)} = \sum_{v_j \in N_i}(\mathbf{d}(v_i, V_i') - \mathbf{d}(v_i, \{v_j\}))$, where $\mathbf{d}(v_i, V_i')$ is fixed when $\mathcal{X}$ is fixed. Hence, we only need to show that

$$\mathbb{E}[\sum_{v_j \in N_i} \mathbf{d}(v_i, \{v_j\})] = \sum_{v_j \in V_i'} \Pr[v_j \in N_i]\mathbf{d}(v_i, \{v_j\})$$

decreases as τ increases. Indeed, as τ increases, as long as $(\mathbf{d}(v_i, \{v_j\}))$'s for $v_j \in N_i$ are not all identical (since $\Sigma_0, \Sigma_1 > 0$, there must be cases satisfying this), there exists a threshold $\mathbf{d}_{th}$ such that all the v_j's with $(\mathbf{d}(v_i, \{v_j\})) < \mathbf{d}_{th}$ whose edge sampling weights (i.e., $\mathbb{P}_{ij}$'s) increase and all the v_j''s with $(\mathbf{d}(v_i, \{v_j'\})) > \mathbf{d}_{th}$ whose edge sampling weights decrease, which makes the "weighted sum" $\sum_{v_j \in V_i'} \Pr[v_j \in N_i]\mathbf{d}(v_i, \{v_j\})$ smaller.

Regarding claim (ii). Above, we have proved that $\mathbf{h}(\cdot)$ is a strictly increasing function of τ, which also means that $\mathbf{h}(\cdot)$ is an injective function of τ. Hence, it suffices to show that if $\tau = 0$, then $\mathbb{E}[\mathbf{h}(\cdot) = 0]$.

First, since we assume $\Sigma_0 = \Sigma_1 \neq 0$, the class controlled feature distributions of class-0 and class-1 are identical (i.e., $(X_i|Y) \sim \mathcal{N}(0, \Sigma_0), \forall v_i \in V$). Thus, the following holds:

$$\mathbb{E}[\mathbf{d}(v_i, C_{Y_i}^+ \backslash \{v_i\})] = \mathbb{E}[\mathbf{d}(v_i, C_{Y_i}^-)] = \mathbb{E}[\mathbf{d}(v_i, V_i')], \forall v_i \in V. \tag{14}$$

Recall that C_ℓ^+ denotes the node set of class ℓ, whereas C_ℓ^- denotes the set of the rest of the nodes.

Second, if $\tau = 0$, then $\phi_{ij} = 1, \forall (i,j) \in V \times V$. This means that the edge sampling probabilities are identical for all the same-class node pairs and for all the different-class node pairs, respectively. Then, for each node v_i, the same-class neighbor set N_i^+ is chosen from $C_{Y_i}^+ \backslash \{v_i\}$ uniformly at random. Likewise, the different-class neighbor set N_i^- is chosen from $C_{Y_i}^-$ uniformly at random. Thus,

$$\begin{aligned} &\mathbb{E}[\mathbf{d}(v_i, N_i^+)] = \mathbb{E}[\mathbf{d}(v_i, C_{Y_i}^+ \backslash \{v_i\})], \forall v_i \in V \\ &\mathbb{E}[\mathbf{d}(v_i, N_i^-)] = \mathbb{E}[\mathbf{d}(v_i, C_{Y_i}^-)], \forall v_i \in V. \end{aligned} \tag{15}$$

Combining Eqs. (14) and (15), the following holds if $\tau = 0$:

$$\mathbb{E}[\mathbf{d}(v_i, N_i^+)] = \mathbb{E}[\mathbf{d}(v_i, N_i^-)] = \mathbb{E}[\mathbf{d}(v_i, N_i)], \forall v_i \in V.$$

Since $\mathbb{E}[b(v_i)] = \mathbb{E}[\mathbf{d}(b_i, V_i')]$ by definition, the following holds if $\tau = 0$:

$$\begin{aligned} \mathbb{E}[\mathbf{h}_i^{(v)}] &= \mathbb{E}[b(v_i)] - \mathbb{E}[\mathbf{d}(v_i, N_i)] = 0, \\ \mathbb{E}[\mathbf{h}^{(G)}] &= \frac{1}{|V|} \sum_{v_j \in V} \mathbb{E}[\mathbf{h}_i^v] = 0. \end{aligned}$$

□

Proof of Lemma 4.2 (τ controls CFH $\mathbf{h}(\cdot)$ *only*).

Proof. It is straightforward, since the values of $\mathrm{FD}(\mathcal{G})$ and $\mathbf{h}_c(\mathcal{G})$ are directly controlled by the other parameters and are independent of the value of τ. □

A.3. Proofs for Graph Convolution in CSBM-X Graphs

Theorem 4.3. *Following the analysis setting, i.e., we assume (i) 1-dimensional node features $\mu_\ell, \Sigma_\ell, X_i \in \mathbb{R}$, (ii) symmetric feature means $\mu_0 = -\mu_1 \neq 0$ with identical variances $\Sigma_0 = \Sigma_1 = 1$, and we focus on asymptotic setting with (iii) fixed $p^- \neq p^+ \in (0, \frac{1}{2})$ and (iv) $n \to \infty$ with $d^+ = np^+$ and $d^- = np^-$. Use the prior distribution $\Pr[Y_i = 0] = \Pr[Y_i = 1] = 1/2$ and fix the other parameters except for τ, after a step of graph convolution $D^{-1}AX$, the Bayes error rate (BER) of $\mathcal{F}$, denoted by $\mathcal{B}_{\mathcal{F}}(\mathcal{G}(\cdot, \tau))$ is minimized at $\tau = 0$ and strictly increases as $|\tau|$ increases, i.e., $\arg\min_\tau \mathcal{B}_{\mathcal{F}}(\mathcal{G}(\cdot, \tau)) = 0$; $\mathcal{B}_{\mathcal{F}}(\mathcal{G}(\cdot, \tau_0)) < \mathcal{B}_{\mathcal{F}}(\mathcal{G}(\cdot, \tau_1))$ for any τ_0 and τ_1 such that $|\tau_0| < |\tau_1|$ and $\tau_0\tau_1 > 0$.*

Proof. To provide a high-level idea, after a step of graph convolution, higher $|\tau|$ makes more nodes have features far from the mean of its whole class and, thus, results in a higher error in classification.

For the simplicity of presentation, we assume $\mu_0 = -\mu_1$. Also, we illustrate with one-dimension node features $X_i \in \mathbb{R}$ for each node v_i here, but the reasoning can be extended to high-dimensional features in general.

WLOG, we assume $\Sigma_0 = \Sigma_1 = 1$, which can be ensured by feature normalization. As $n \to \infty$, the sample mean and variance of the node features in each class approach $\pm\mu$ and Σ. For a node v_i, WLOG (due to the symmetry), we assume $Y_i = 1$, and let its feature be $\mu + x_i$ (i.e., its class-controlled feature is x_i). Let φ be the PDF of standard normal distribution $\mathcal{N}(0,1)$, then the homophily baseline $b(v_i)$ of node v_i is

$$b(v_i) = \int_{-\infty}^{\infty} \varphi(x)\,|x - x_i|\,dx = \frac{1}{2}e^{-\frac{x_i^2}{2}}\left(e^{\frac{x_i^2}{2}}x_i\text{erf}\left(\frac{x_i}{\sqrt{2}}\right) - e^{\frac{x_i^2}{2}}x_i\text{erfc}\left(\frac{x_i}{\sqrt{2}}\right) + e^{\frac{x_i^2}{2}}x_i + 2\sqrt{\frac{2}{\pi}}\right) = \exp(-\frac{x_i^2}{2}) + x_i\text{erf}(\frac{x_i}{\sqrt{2}}),$$

where "erf" is the Gauss error function defined as

$$\text{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2}dt$$

and "erfc" is the complementary error function defined as

$$\text{erfc}(z) = 1 - \text{erf}(z).$$

Hence, the CFH between x_i and another node v_j with class-controlled feature x_j (i.e., v_j has feature $-\mu + x_j$ if $Y_j = 0$, and it has feature $\mu + x_j$ if $Y_j = 1$) is

$$\mathbf{h}_{ij}^{(p)} = b(v_i) - |x_i - x_j| = \exp(-\frac{x_i^2}{2})\sqrt{\frac{2}{\pi}} + x_i\text{erf}\left(\frac{x_i}{\sqrt{2}}\right) - |x_i - x_j|\,,$$

which gives

$$\phi_{ij} = \exp(\tau\mathbf{h}_{ij}^{(p)}) = \exp\left(\tau\left(-|x_i - x_j| + x_i\text{erf}\left(\frac{x_i}{\sqrt{2}}\right) + \sqrt{\frac{2}{\pi}}\exp(-\frac{x_i^2}{2})\right)\right)$$

Since $n \to \infty$, weighted sampling without replacement approaches weighted sampling with replacement approaches, and the probability of v_j being sampled as one of v_i's neighbors is

$$\Pr[v_j \in N_i] = \frac{\phi_{ij}}{\int_{-\infty}^{\infty}\phi_{ij'}\varphi(x_{j'})dx_{j'}} = \frac{2\exp(-\frac{1}{2}\tau(2|x_i - x_j| + \tau - 2x_i))}{\text{erfc}\left(\frac{\tau - x_i}{\sqrt{2}}\right) + \exp(2\tau x_i)\text{erfc}\left(\frac{\tau + x_i}{\sqrt{2}}\right)},$$

and in each sampling step, the sampled neighbor has a class-controlled feature equal to x_j is

$$\Pr[v_j \in N_i]\varphi(x_j).$$

In other words, let x_{nbr} denote the random variable of the class-controlled feature of a sampled neighbor, we have

$$\Pr[x_{nbr} = x^*] = \frac{2\exp(-\frac{1}{2}\tau(2|x_i - x^*| + \tau - 2x_i))}{\text{erfc}\left(\frac{\tau - x_i}{\sqrt{2}}\right) + \exp(2\tau x_i)\text{erfc}\left(\frac{\tau + x_i}{\sqrt{2}}\right)}\varphi(x^*) = \frac{\sqrt{\frac{2}{\pi}}\exp\left(-\frac{1}{2}\tau(2|x_i - x^*| + \tau - 2x_i) - \frac{x_i^2}{2}\right)}{\text{erfc}\left(\frac{\tau - x_i}{\sqrt{2}}\right) + \exp(2\tau x_i)\text{erfc}\left(\frac{\tau + x_i}{\sqrt{2}}\right)}, \forall x^* \in \mathbb{R}.$$

We can compute the closed-form expectation of x_{nbr}, which is

$$\mathbb{E}[x_{nbr}] = \int_{-\infty}^{\infty} x^*\Pr[x_{nbr} = x^*]dx^* = \frac{\tau\left(\text{erfc}\left(\frac{\tau - x_i}{\sqrt{2}}\right) - \exp(2\tau x_i)\text{erfc}\left(\frac{\tau + x_i}{\sqrt{2}}\right)\right)}{\text{erfc}\left(\frac{\tau - x_i}{\sqrt{2}}\right) + \exp(2\tau x_i)\text{erfc}\left(\frac{\tau + x_i}{\sqrt{2}}\right)},$$

and its variance $\text{Var}[x_{nbr}]$ does not depend on the value of n. After one step of graph convolution, the new node feature of v_i would be

$$\hat{x}_i = \frac{d^+\mu - d^-\mu + \sum_{t=1}^{d^+ + d^-} x_t}{d^+ + d^-},$$

where each x_t i.i.d. follows x_{nbr}. By the central limit theorem, as $n \to \infty$ and, thus, d^+ and d^- approaches infinity, $\hat{x}_i$ asymptotically follows $\mathcal{N}(\frac{d^+\mu - d^-\mu}{d^+ + d^-} + \mathbb{E}[x_{nbr}], \mathrm{Var}[x_{nbr}]/\sqrt{d^+ + d^-})$. WLOG, we assume $d^+ > d^-$ here (when $d^+ < d^-$, the classifier is flipped in a symmetric manner). The classifier would be equivalent to

$$\mathcal{F}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases},$$

and the probability that $\mathcal{F}$ misclassifies v_i is $\Pr[\hat{x}_i < 0]$, which would approach a binary function (since $\mathrm{Var}[x_{nbr}]/\sqrt{d^+ + d^-}$ approaches zero) $\mathbf{1}[\frac{d^+\mu - d^-\mu}{d^+ + d^-} + \mathbb{E}[x_{nbr}] < 0]$.

Now, we first claim that $\tau = 0$ asymptotically gives the lowest error probability 0. Indeed, when $\tau = 0$, $\mathbb{E}[x_{nbr}] = 0$ regardless of the value of x_i, and $\mathbf{1}[\frac{d^+\mu - d^-\mu}{d^+ + d^-} + \mathbb{E}[x_{nbr}] < 0] \to \mathbf{1}[\frac{d^+\mu - d^-\mu}{d^+ + d^-} < 0] = 0$.

Then, we claim that in both directions, the Bayes error rate (BER) of $\mathcal{F}$ increases as $|\tau|$ increases. First, by the symmetric prior, the BER can be written as

$$\mathcal{B}_{\mathcal{F}}(\mathcal{G}(\cdot, \tau)) = \frac{1}{2}\left(\Pr[\mathcal{F}(x_i) = 1 \mid Y_i = 0] + \Pr[\mathcal{F}(x_i) = 0 \mid Y_i = 1]\right)$$

Again, due to the symmetry, it is equal to

$$\Pr[\mathcal{F}(x_i) = 0 \mid Y_i = 1] = \int \Pr[\mathcal{F}(x_i) = 0] \Pr[x_i \mid Y_i = 1] dx_i.$$

By the above analysis, after a step of graph convolution, the BER is

$$\int \Pr[\mathcal{F}(x_i) = 0] \Pr[x_i \mid Y_i = 1] dx_i = \int_{-\infty}^{\infty} \Pr[\hat{x}_i < 0] \varphi[x_i] dx_i$$

approaching

$$\int_{-\infty}^{\infty} \mathbf{1}[\frac{d^+\mu - d^-\mu}{d^+ + d^-} + \mathbb{E}[x_{nbr}] < 0] \varphi[x_i] dx_i.$$

When $\tau > 0$, $\mathbb{E}[x_{nbr}]$ has the same sign as x_i. In such case, we only need to consider $x_i < 0$, since $\mathbb{E}[x_{nbr}] > 0$ when $x_i > 0$. We claim that for any fixed $x_i < 0$, $\mathbb{E}[x_{nbr}]$ (w.r.t. x_i and τ) is decreasing w.r.t $\tau > 0$, and thus $\mathbf{1}[\frac{d^+\mu - d^-\mu}{d^+ + d^-} + \mathbb{E}[x_{nbr}] < 0]$ is non-decreasing for all $x_i < 0$, which implies the increase in the BER. Indeed,

$$\frac{\partial}{\partial \tau}\mathbb{E}[x_{nbr}] = \frac{\exp(\tau x_i)\left(2\tau \exp(\tau^2 + x_i^2)\left(\sqrt{\frac{2}{\pi}} - 2x_i \exp(\frac{1}{2}(\tau + x_i)^2)\mathrm{erfc}\left(\frac{\tau + x_i}{\sqrt{2}}\right)\right)\mathrm{erfc}\left(\frac{\tau - x_i}{\sqrt{2}}\right) + \exp((\tau - x_i)^2 + \frac{1}{2}(\tau + x_i)^2)\mathrm{erfc}\left(\frac{\tau - x_i}{\sqrt{2}}\right)^2 - \exp((\tau + x_i)^2)\mathrm{erfc}\left(\frac{\tau + x_i}{\sqrt{2}}\right)\left(\exp(\frac{1}{2}(\tau + x_i)^2)\mathrm{erfc}\left(\frac{\tau + x_i}{\sqrt{2}}\right) + 2\sqrt{\frac{2}{\pi}}\tau\right)\right)}{\left(\mathrm{erfc}\left(\frac{\tau - x_i}{\sqrt{2}}\right) + \exp(2\tau x_i)\mathrm{erfc}\left(\frac{\tau + x_i}{\sqrt{2}}\right)\right)^2},$$

which is negative for all $x_i < 0$ (the denominator is always positive and the numerator is negative when $\tau > 0$ and $x_i < 0$).

Similarly, we claim that for any fixed $x_i > 0$, $\mathbb{E}[x_{nbr}](x_i; \tau)$ is decreasing w.r.t $\tau < 0$. When $\tau < 0$, $\mathbb{E}[x_{nbr}]$ has the opposite sign as x_i and we only need to consider $x_i > 0$ since $\mathbb{E}[x_{nbr}] > 0$ when $x_i < 0$. We claim that for any fixed $x_i > 0$, $\mathbb{E}[x_{nbr}]$ is decreases as $\tau < 0$ decreases (i.e., τ moves from 0 to $-\infty$), and thus $\mathbf{1}[\frac{d^+\mu - d^-\mu}{d^+ + d^-} + \mathbb{E}[x_{nbr}] < 0]$ is non-decreasing for all x_i values, which implies the increase in the BER. Indeed, the partial derivative is the same as above, where the denominator is always positive and the numerator is positive when $\tau < 0$ and $x_i > 0$.

When node features have higher dimensions, obtaining elegant closed-form equations as above would be challenging, but we still have the property that $\mathbb{E}[\boldsymbol{x}_{nbr}] = \mathbf{0}$ if and only if $\tau = 0$. Moreover, $\boldsymbol{x}_{nbr}$ moves further from $\mathbf{0}$ as $|\tau|$ increases, which increases the BER. Specifically, in the above reasoning, one needs to replace $x > 0$ with $\tilde{\boldsymbol{\mu}}^\top \boldsymbol{x} > 0$ with $\tilde{\boldsymbol{\mu}} = \frac{d^+\boldsymbol{\mu} - d^-\boldsymbol{\mu}}{d^+ - d^-} = \frac{p^+\boldsymbol{\mu} - p^-\boldsymbol{\mu}}{p^+ - p^-}$ (features that would be classified as the positive class, class-1), and similarly replace $x < 0$ with $\tilde{\boldsymbol{\mu}}^\top \boldsymbol{x} < 0$. □

B. In-Depth Analysis of Measure

B.1. $\tilde{\mathrm{h}}(\cdot)$ Interpretation

In this subsection, we discuss the details of $\tilde{\mathbf{h}}(\cdot)$ interpretation. For high-level ideas, refer to Sec. 3.3.

Magnitude: node-level CFH. We first rephrase node-level CFH $\tilde{\mathbf{h}}_i^{(v)}$:

$$\tilde{\mathbf{h}}_i^{(v)} = \frac{\mathbf{h}_i^{(v)}}{\max(b(v_i)\ ,\ \mathbf{d}(v_i, N_i))} = \frac{b(v_i) - \mathbf{d}(v_i, N_i)}{\max(b(v_i)\ ,\ \mathbf{d}(v_i, N_i))} = \begin{cases} 1 - \frac{\mathbf{d}(v_i, N_i)}{b(v_i)} & \text{, if } \tilde{\mathbf{h}}_i^{(v)} \geq 0 \\ \frac{b(v_i)}{\mathbf{d}(v_i, N_i)} - 1 & \text{, otherwise} \end{cases}$$

For positive (or negative) $\tilde{\mathbf{h}}_i^{(v)}$, the node v_i has $\frac{|\tilde{\mathbf{h}}_i^{(v)}|}{1-|\tilde{\mathbf{h}}_i^{(v)}|}$ times smaller (or larger) distance to neighbors $\mathbf{d}(v_i, N_i)$ than its homophily baseline $b(v_i)$. For example, if $\mathbf{d}(v_i, N_i) = 1$ and $b(v_i) = 10$, then $\tilde{\mathbf{h}}_i^{(v)} = 0.9$, indicating that $\mathbf{d}(v_i, N_i)$ is 9 times smaller than $b(v_i)$.

Magnitude: graph-level CFH. We also rephrase graph-level CFH $\tilde{\mathbf{h}}^{(G)}$:

$$\tilde{\mathbf{h}}^{(G)} = \frac{\mathbf{h}^{(G)}}{\frac{1}{|V|}\max(\sum_{v_i \in V} b(v_i)\ ,\ \sum_{v_i \in V} \mathbf{d}(v_i, N_i))} = \begin{cases} 1 - \frac{\sum_{v_i \in V} \mathbf{d}(v_i, N_i)}{\sum_{v_i \in V} b(v_i)} & \text{, if } \tilde{\mathbf{h}}^{(G)} \geq 0 \\ \frac{\sum_{v_i \in V} b(v_i)}{\sum_{v_i \in V} \mathbf{d}(v_i, N_i)} - 1 & \text{, otherwise} \end{cases}$$

Like in node-level interpretation, for positive (or negative) $\tilde{\mathbf{h}}^{(G)}$, the graph G has $\frac{|\tilde{\mathbf{h}}^{(G)}|}{1-|\tilde{\mathbf{h}}^{(G)}|}$ times smaller (or larger) mean distance to neighbors $\frac{1}{|V|}\sum_{v_i \in V} \mathbf{d}(v_i, N_i)$ than the mean homophily baseline $\frac{1}{|V|}\sum_{v_i \in V} b(v_i)$. For example, if $\frac{1}{|V|}\sum_{v_i \in V} \mathbf{d}(v_i, N_i) = 1$ and $\frac{1}{|V|}\sum_{v_i \in V} b(v_i) = 10$, then $\tilde{\mathbf{h}}^{(G)} = 0.9$, indicating that $\frac{1}{|V|}\sum_{v_i \in V} \mathbf{d}(v_i, N_i)$ is 9 times smaller than $\frac{1}{|V|}\sum_{v_i \in V} b(v_i)$.

Zero. If node v_i's feature is identical to all other nodes (i.e., $X_i = X_j, \forall v_j \in V$), $\tilde{\mathbf{h}}_i^{(v)} = 0$, because its $b(v_i) = \mathbf{d}(v_i, N_i) = 0$. [8] A fully connected node v_i has $\tilde{\mathbf{h}}_i^{(v)} = 0$, because its $b(v_i) = \mathbf{d}(v_i, N_i)$. For the same reason, a graph G has $\tilde{\mathbf{h}}^{(G)} = 0$ if (*i*) it is fully connected and/or (*ii*) has all identical node features.

If a node v_i chooses the non-zero number of neighbors by a random probability, $\mathbb{E}[\tilde{\mathbf{h}}_i^{(v)}] = 0$. For the same reason, a graph G has $\mathbb{E}[\tilde{\mathbf{h}}^{(G)}] = 0$ if each node $v_i \in V$ chooses a non-zero number of neighbors by a random probability.

It is important to note that there are many other conditions in which $\tilde{\mathbf{h}}_i^{(v)}$ and $\tilde{\mathbf{h}}^{(G)}$ become 0. That is, while $\tilde{\mathbf{h}}_i^{(v)}$ and $\tilde{\mathbf{h}}^{(G)}$ being 0 may suggest no A-X dependence, they are not conclusive. In-depth analysis of the microscopic patterns, such as distributions of $\tilde{\mathbf{h}}_i^{(v)}$ and $\tilde{\mathbf{h}}_{ij}^{(p)}$, may better elucidate the levels of A-X dependence.

B.2. On Class Control

Connection to Part and Partial Correlation. The class control mechanism in Eq. (3) is analogous to the variable control method of part and partial correlation. We focus on part correlation here.

The goal of its variable control is to control the effect of the third variable when analyzing the correlation between two variables. Let $X^{(P)}, A^{(P)} \in \mathbb{R}^N$ be two variables of interest and $Y^{(P)} \in \mathbb{R}^{N \times d}$ be the third variable, where N is the number of observed samples.

$$\beta^* = \arg\min_{\beta} \|(X^{(P)} - Y^{(P)}\beta)\|_2^2 \tag{16}$$

$$X|Y = X^{(P)} - (Y^{(P)}\beta^*), \tag{17}$$

where $\beta \in \mathbb{R}^d$ is a regression coefficient. Geometric interpretation of this mechanism is the projection of original $X^{(P)}$ onto the orthogonal space of $Y^{(P)}$ with the least approximation L2-error, expecting that the information of $X^{(P)}$ is maximally maintained given the removal of $Y^{(P)}$ intervention. In part correlation, correlation is measured between $X|Y$ and A.

Now, we show how Eq. (3) relates to the above equation. Let $Y^{(P)} \in \{0, 1\}^{|V| \times c}$ be the one-hot labeled class matrix for each node (i.e., $Y_{ij}^{(P)} = 1$ for $y_i = j, \forall v_i \in V$, 0 otherwise). Let $X^{(P)} \in \mathbb{R}^{|V|}$ be the original node feature. Now, in this analysis, we

[8] Recall that we define $\tilde{\mathbf{h}}_i^{(v)}$ and $\tilde{\mathbf{h}}^{(G)}$ to be 0, if $b(\cdot) = 0$.

let $X^{(P)} \coloneqq X$ and $Y^{(P)} \coloneqq Y$ for notational simplicity. We optimize Eq. (16) as below:

$$\beta^* = \arg\min_{\beta} \|X - Y\beta\|_2^2 = \arg\min_{\beta} (X - Y\beta)^T (X - Y\beta) \coloneqq \arg\min_{\beta} \mathcal{L} \tag{18}$$

$$\frac{\partial \mathcal{L}}{\partial \beta} = -2Y^T X + 2(Y^T Y)\beta = 0 \equiv \beta^* = (Y^T Y)^{-1} Y^T X. \tag{19}$$

As we take a closer look at the form of β^* in Eq. (19):

- $Y^T Y \in \mathbb{R}^{c \times c}$ is a diagonal matrix where each $i-$th diagonal entry indicates the number of nodes belonging to the class i (i.e., $(Y^T Y)_{ii} = |C_i^+|, \forall i \in [c]$).
- $Y^T X \in \mathbb{R}^c$ is a vector where $j-$th entry indicates the sum of node features that belong to the class i (i.e., $(Y^T X)_i = \sum_{v_k \in C_i^+} X_k, \forall i \in [c]$).

Thus, $\beta^* \in \mathbb{R}^c$ is a vector where $k-$ entry indicates the mean of node features that belong to the class i. In the given setting, by applying obtained β^*, Eq. (17) is equivalent to $(X|Y)_i = X_i - \frac{1}{|C_{y_i}^+|} \sum_{v_k \in C_{y_i}^+} X_k$, which is equal to Eq. (3). Therefore, we conclude that Eq. (3) is a special case of the variable control method of part correlation.

B.3. Generalizing $\tilde{\mathrm{h}}(\cdot)$

CFH $\tilde{\mathrm{h}}(\cdot)$ measures A-X dependence, while controlling for potential confounding by node class. However, with its good properties, we can generalize it to measure topology-feature and topology-class dependence *without* confound control.

Generalized distance function. Denote the distance function (Eq. (4)) with a matrix $\mathbf{X} \in \mathbb{R}^{n \times k}$ as

$$\mathbf{d}^*(v_i, V_i', \mathbf{X}) \coloneqq \frac{1}{|V_i'|} \sum_{v_j \in V_i'} \|\mathbf{X}_i - \mathbf{X}_j\|_2. \tag{20}$$

Eq. (4) is a special case of Eq. (20), where $\mathbf{X} = X|Y$. Likewise, we generalize homophily baseline as $b^*(v_i) = \mathbf{d}^*(v_i, V_i', \mathbf{X})$.

Generalized homophily measure. Based on $\mathbf{d}^*(\cdot)$, we propose a generalized homophily measure $\tilde{H}(\cdot)$.

G1) Generalized node pair-level homophily $H_{ij}^{(p)}$:

$$H_{ij}^{(p)}(\mathbf{X}) = H((v_i, v_j) \mid E, \mathbf{X}) \coloneqq b^*(v_i) - \mathbf{d}^*(v_i, \{v_j\}, \mathbf{X}) \tag{21}$$

G2) Generalized node-level homophily $H_i^{(v)}$:

$$H_i^{(v)}(\mathbf{X}) = H(v_i \mid E, \mathbf{X}) \coloneqq \frac{1}{|N_i|} \sum_{v_j \in N_i} H_{ij}^{(p)} \tag{22}$$

G3) Generalized graph-level homophily $H^{(G)}$:

$$H^{(G)}(\mathbf{X}) = H(G \mid E, \mathbf{X}) \coloneqq \frac{1}{|V|} \sum_{v_j \in V} H_j^{(v)} \tag{23}$$

G4) Generalized node-level normalization:

$$\tilde{H}_i^{(v)}(\mathbf{X}) = \frac{H_i^{(v)}}{\max(b^*(v_i), \mathbf{d}^*(v_i, N_i, \mathbf{X}))}. \tag{24}$$

G5) Generalized graph-level normalization: [9]

$$\tilde{H}^{(G)}(\mathbf{X}) = \frac{H^{(G)}}{\frac{1}{|V|} \max(\sum_{v_i \in V} b^*(v_i), \sum_{v_i \in V} \mathbf{d}^*(v_i, N_i, \mathbf{X}))}. \tag{25}$$

CFH measure $\tilde{\mathrm{h}}(\cdot)$ is a special case of the proposed generalized homophily measure $\tilde{H}(\cdot)$. With the generalized homophily measure, we can measure feature homophily $\tilde{H}^{(G)}(X)$ and class homophily $\tilde{H}^{(G)}(Y)$, where $Y \in \mathbb{R}^{n \times c}$ is a node class matrix.

[9] For completeness, if $b^*(\cdot) = 0$, we let $\tilde{H}_i^{(v)}(\cdot), \tilde{H}^{(G)}(\cdot) = 0$.

Table 1: Comparison of Graph-Level CFH Scores with Different Features

Dataset	Cora	CiteSeer	PubMed	Cora-ML	Cora-Full	DBLP	Wiki-CS	CS	Physics	Photo	Computers	Ogbn-ArXiv
$X^{(orig)}$	0.0562	0.0802	0.1072	0.0390	0.0388	0.0786	0.2182	-0.0042	0.0760	-0.0150	-0.0207	0.0755
$X^{(rand)}$	0.0204	0.0151	-0.0061	0.0036	0.0053	0.0031	-0.0018	0.0128	0.0031	0.0131	0.0067	0.0036
$X^{(conv)}(1)$	0.4612	0.5706	0.4399	0.4217	0.3991	0.4605	0.4442	0.3991	0.4603	0.3421	0.3278	0.3855
$X^{(conv)}(2)$	0.6238	0.7177	0.6095	0.5956	0.5621	0.6227	0.4956	0.5326	0.5835	0.5214	0.4878	0.4768
$X^{(conv)}(4)$	0.7221	0.8003	0.7181	0.7017	0.6513	0.7271	0.5120	0.5848	0.6479	0.6515	0.5995	0.5242

Dataset	Chameleon	Squirrel	Actor	Texas	Cornell	Wisconsin	RM-Emp	AMZ-Rts	Tolokers	Penn94	Flickr	ArXiv-Year
$X^{(orig)}$	-0.0714	-0.0538	-0.0199	-0.0803	0.0041	-0.0324	0.0199	0.1266	0.1296	0.0870	0.0018	0.1206
$X^{(rand)}$	-0.0368	0.0136	0.0060	-0.0398	0.0390	0.0165	0.0020	-0.0003	-0.0131	-0.0013	-0.0023	0.0037
$X^{(conv)}(1)$	0.3835	0.3529	0.2745	0.2279	0.2121	0.2590	0.4165	0.5893	0.5162	O.O.M.	0.2052	0.4709
$X^{(conv)}(2)$	0.5299	0.5316	0.3983	0.3111	0.3017	0.3584	0.5899	0.7667	0.6524	O.O.M.	0.2097	0.5784
$X^{(conv)}(4)$	0.6404	0.6227	0.4974	0.3704	0.3476	0.4237	0.6852	0.8563	0.6881	O.O.M.	0.3185	0.6386

$(*)$ $X^{(orig)}$ denotes the original node features. RM-Emp stands for Roman-Empire, and AMZ-Rts stands for Amazon-Ratings. O.O.M. denotes out-of-memory.

C. In-Depth Analysis of the Benchmark Datasets

We further analyze the benchmark datasets. Specifically, we buttress Observations 1-3 with additional results. We also briefly discuss the Roman-Empire dataset, delving into why GNN performance degrades consistently over the feature shuffles.

C.1. Observation 1: The Full Results

We focus on the lowness of CFH $\tilde{\mathbf{h}}(\cdot)$ in the benchmark graphs. Specifically, we further support Observation 1 with (*i*) comparison of CFH scores with different features and (*ii*) node-level analysis.

Comparison to different features. First, we investigate how low the CFH $\tilde{\mathbf{h}}(\cdot)$ scores are for the benchmark graphs, compared to different features. Recall that their mean $\left|\tilde{\mathbf{h}}^{(G)}\right| = 0.06$. For comparison, we consider two other node features.

- Random Baseline: Random node features $X_i^{(rand)} \sim \mathcal{N}(0,1), \forall v_i \in V$.
- Homophilic Baseline: Convoluted node features $X^{(conv)}(l) = ((D+I)^{-1}(A+I))^l X$,

where $I \in \mathbb{R}^{n \times n}$ is an identity matrix and $l \in \{1,2,4\}$.

In Table 1, we report the $\tilde{\mathbf{h}}^{(G)}$ for each feature type and dataset. Averaging the scores for all 24 datasets, $X^{(conv)}(4)$ has the mean $\tilde{\mathbf{h}}^{(G)}$ score of 0.61, while $X^{(rand)}$ has the mean near 0. The mean $\tilde{\mathbf{h}}^{(G)}$ of the original node feature $X^{(orig)}$ is much closer to that of the random features $X^{(rand)}$, further supporting Observation 1.

Node-level analysis. Second, we report that node-level CFH $\tilde{\mathbf{h}}_i^{(v)}$ scores also tend to be positive and low. As shown in Fig. 12(a), most nodes in most graphs have $|\mathbf{h}_i^{(v)}| < 0.3$. As observed at the graph level, each node's distance to the neighbors is close to its homophily baseline.

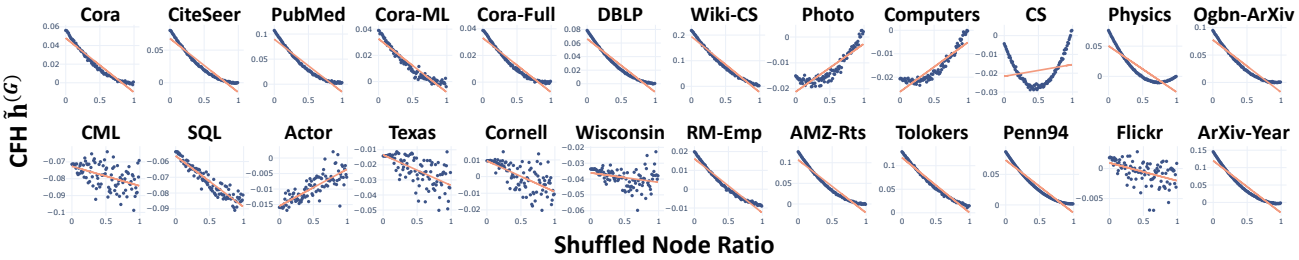
Conclusion. From the analyses, we conclude again that CFH $\tilde{\mathbf{h}}(\cdot)$ are generally positive and low in the benchmark datasets.

C.2. Observation 2: The Full Results

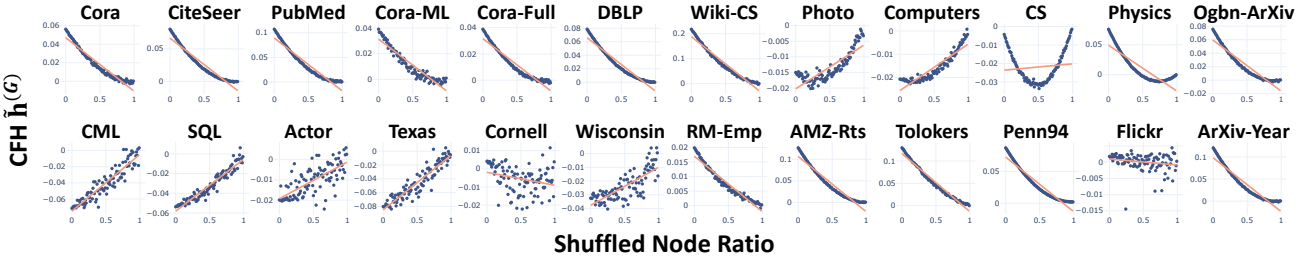
We further demonstrate that CFH and class-homophily have a small, positive correlation with two additional evidence. We (*i*) measure correlations between CFH and the other class-homophily measures and (*ii*) conduct node-level correlation analysis.

Other class-homophily measures. First, we complement Observation 2 by measuring correlations between CFH and different measures of class-homophily, defined by Pei et al. (2020) and Zhu et al. (2020). Class-homophily defined by Zhu et al. (2020) and CFH have correlation coefficients of 0.403 (Pearson's r) and 0.196 (Kendall's τ). Class-homophily defined by Pei et al. (2020) and CFH have correlation coefficients of 0.401 (Pearson's r) and 0.225 (Kendall's τ). While we find slightly stronger correlations between CFH and the other measures, the correlations are not consistently strong, such that there exist non-negligible gaps between Pearson's r and Kendall's τ. We, thus, do not find counter-evidence for Observation 2.

Node-level analysis. Now, we complement Observation 2 with node-level analysis. Specifically, we analyze the correlation



(a) Graph-level CFH score $\tilde{\mathbf{h}}^{(G)}$'s over the *class-wise* feature shuffles.



(b) Graph-level CFH score $\tilde{\mathbf{h}}^{(G)}$'s over the *non-class-wise* feature shuffles.

Figure 11: **Graph-Level CFH Statistics in Real-World Graphs and their Relationship to the Feature Shuffle.**

between node-level CFH $\tilde{\mathbf{h}}_i^{(v)}$ and node-level class-homophily $\tilde{H}_i^{(v)}(Y)$ (Eq. (24)). [10] We find that the Pearson correlations in most of the 24 graphs are very low, such that the absolute values of their Pearson's r scores are below 0.2 in 22/24 graphs (Table 2). Also, 19/24 have positive Pearson correlations.

Conclusion. From the analyses, we conclude again that CFH $\tilde{\mathbf{h}}(\cdot)$ has a small, positive correlation with class homophily.

C.3. Observation 3: The Full Results

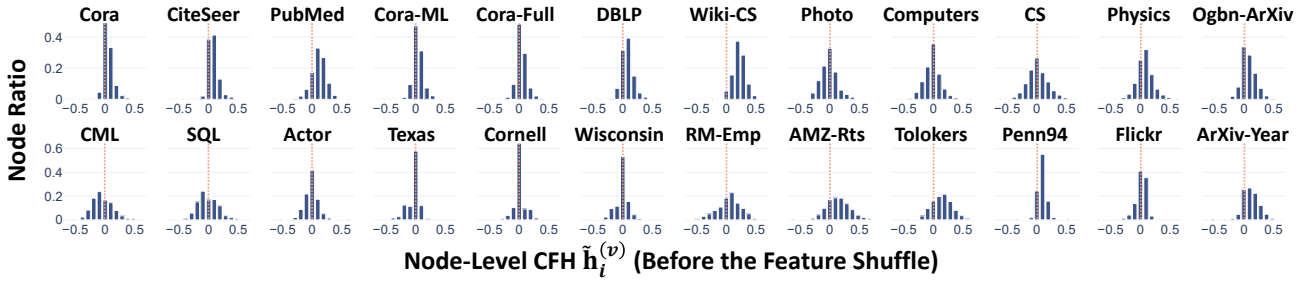
Here, we report how the feature shuffle affects CFH in all 24 benchmark datasets (Figs. 11-12). While most datasets follow the pattern reported in Observation 3, few of them (Chameleon, Squirrel, Texas, Wisconsin, and Cornell) do not fully obey it. Specifically, their graph-level CFH $\tilde{\mathbf{h}}^{(G)}$ score moves away from 0 over increasing shuffled node ratio. We reason their deviation by answering two questions: (*i*) Why do the $\tilde{\mathbf{h}}^{(G)}$ scores become larger after the feature shuffle?; (*ii*) Why do the $\tilde{\mathbf{h}}^{(G)}$ scores not approach 0 after the feature shuffle?

Answer to question (i). Node-level CFH $\tilde{\mathbf{h}}_i^{(v)}$ distributions before and after the feature shuffle (Fig. 12) reveals that the mean $|\tilde{\mathbf{h}}_i^{(v)}|$ decreases after the feature shuffle in all five datasets. The finding indicates that the magnitude in which the distance to neighbors (i.e., $\mathbf{d}(v_i, N_i)$) deviates from the homophily baseline (i.e., $b(v_i)$) becomes smaller after the feature shuffle. In short, the finding demonstrates (*i*) that A-X dependence is perturbed after the feature shuffle and (*ii*) an in-depth analysis of $\tilde{\mathbf{h}}(\cdot)$ is necessary to reveal the pattern. The graph-level CFH $\tilde{\mathbf{h}}^{(G)}$ does not fully capture the subtlety as it mean-aggregates the positive and negative $\tilde{\mathbf{h}}_i^{(v)}$ scores.

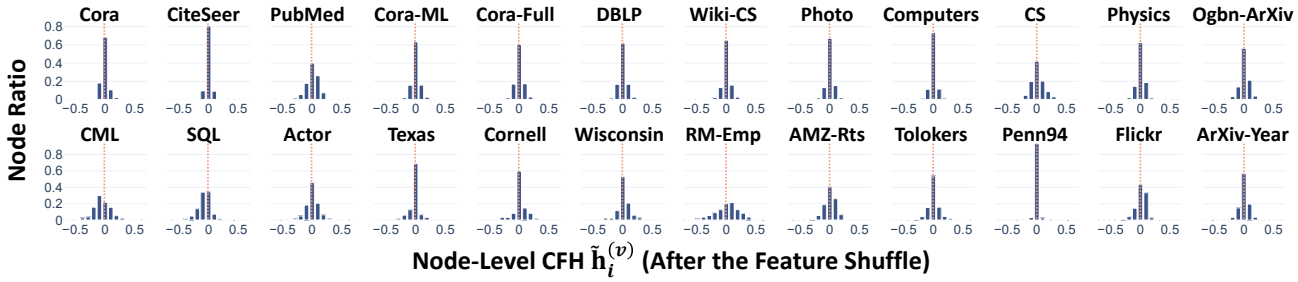
Answer to question (ii). The $\tilde{\mathbf{h}}^{(G)}$ scores may not approach 0 due to the imperfect class-control. We evidence our claim with *non-class-wise* feature shuffle, which means that the feature vectors of all nodes, irrespective of their class membership, are shuffled together. After *non-class-wise* feature shuffle, we find that the graph-level CFH $\tilde{\mathbf{h}}^{(G)}$'s approach 0 in 23/24 datasets (Fig. 11(b)). The finding suggests that feature distribution difference between node classes hinders CFH $\tilde{\mathbf{h}}^{(G)}$ from approaching 0. An advanced class-control method may mitigate such a problem, and we leave it up to future studies.

Conclusion. The series of analyses underscore the complexity of quantifying A-X dependence. We claim that while the feature shuffle effectively perturbs A-X dependence, $\tilde{\mathbf{h}}^{(G)}$ may not approach 0 due to (*i*) node-level discrepancies and (*ii*) the complex nature of feature distribution. Therefore, we further argue that a few datasets' deviations from Observation 3 do not undermine the integrity of our conclusion that A-X dependence mediates the effect of graph convolution.

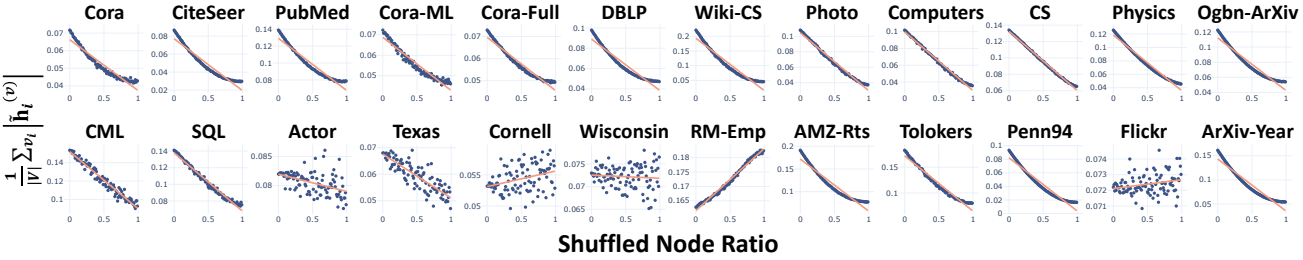
[10] We use $\tilde{H}_i^{(v)}(Y)$ as the node-level class-homophily measure because $\mathbf{h}_c$ has not been defined at node-level.



(a) Histogram of node-level CFH score $\tilde{\mathbf{h}}_i^{(v)}$'s before the feature shuffle.



(b) Histogram of node-level CFH score $\tilde{\mathbf{h}}_i^{(v)}$'s after the feature shuffle.



(c) The mean node-level CFH magnitude $|\mathbf{h}_i^{(v)}|$'s over the feature shuffles.

Figure 12: **Node-Level CFH Statistics in Real-World Graphs and Their Relationship to the Feature Shuffle.**

C.4. The Roman-Empire Dataset

The Roman-Empire dataset has an unusual, chain-like graph topology. Its number of nodes is 22,662, with a diameter of 6,824. In short, there is no small world effect observed, making the effect of the feature shuffle different from the rest of the datasets. A node-level analysis reveals its unique patterns of CFH $\tilde{\mathbf{h}}(\cdot)$ over the feature shuffle. In Fig. 12(a,b), we *uniquely* observe that its histograms of $\tilde{\mathbf{h}}_i^{(v)}$ before and after the feature shuffle are highly similar. In Fig. 12(c), we further *uniquely* observe that its mean $|\tilde{\mathbf{h}}_i^{(v)}|$ increase significantly (2%p) after the feature shuffle. The qualitative and quantitative uniqueness of the Roman-Empire dataset may have contributed to the degrading GNN performance over the feature shuffles.

C.5. Dataset Description

We provide a comprehensive description of the benchmark datasets, with their statistics in Table 2.

- The *Cora*, *CiteSeer*, *PubMed*, *Cora-ML*, *Cora-Full*, *DBLP*, and *Ogbn-ArXiv* (Yang et al., 2016; Bojchevski & Günnemann, 2018; Hu et al., 2020) datasets are citation networks. Each node represents a document, and two nodes are adjacent if a citation exists between the two corresponding articles. For each node, the features are the text features of the corresponding article, and the node class is the category of the research/subject domain of the document.
- The *Wiki-CS* dataset is a webpage network of Wikipedia (Mernyei & Cangea, 2020). Each node represents a Wikipedia webpage, and two nodes are adjacent if a hyperlink exists between the two webpages. For each node, the features are the GloVe word embeddings of the webpage, and the node class represents the article category of the webpage.
- The *Computer* and *Photo* datasets are Amazon co-purchase networks (Shchur et al., 2018). Each node represents a

Table 2: Statistics of the Benchmark Datasets

Dataset	Cora	CiteSeer	PubMed	Cora-ML	Cora-Full	DBLP	Wiki-CS	CS	Physics	Photo	Computers	Ogbn-ArXiv
# Nodes	2,708	3,327	19,717	2,995	19,793	17,716	11,701	18,333	34,393	7,650	13,752	169,343
# Edges	10,556	9,104	88,648	16,316	126,842	105,734	431,726	163,788	495,924	238,162	491,722	1,166,243
# Features	1,433	3,704	500	2,879	8,710	1,639	300	6,805	8,415	745	767	128
# Class	7	6	3	7	70	4	10	15	5	8	10	40
Class Homophily $\mathbf{h}_c$	0.7657	0.6267	0.6641	0.7401	0.4959	0.6522	0.5681	0.7547	0.8474	0.7722	0.7002	0.4445
CFH $\tilde{\mathbf{h}}^{(G)}$	0.0562	0.0802	0.1072	0.0390	0.0388	0.0786	0.2182	-0.0042	0.0760	-0.0150	-0.0207	0.0755
Pearson$(\tilde{\mathbf{h}}_i^{(v)}, \tilde{H}_i^{(v)}(Y))$	0.1158	0.1907	0.0660	0.1602	0.1090	0.1015	0.2993	0.1938	0.1344	0.1332	0.0287	0.2535

Dataset	Chameleon	Squirrel	Actor	Texas	Cornell	Wisconsin	RM-Emp	AMZ-Rts	Tolokers	Penn94	Flickr	ArXiv-Year
# Nodes	890	2,223	7,600	183	183	251	22,662	24,292	11,758	41,554	89,250	169,343
# Edges	17,708	93,996	30,019	325	298	515	65,854	186,100	1,038,000	2,724,458	899,756	1,166,243
# Features	2,325	2,089	932	1,703	1,703	1,703	300	300	10	4814	500	128
# Class	5	5	5	5	5	5	18	5	2	2	7	5
Class Homophily $\mathbf{h}_c$	0.0444	0.0398	0.0061	0.0000	0.1504	0.0839	0.0208	0.1266	0.1801	0.0460	0.0698	0.1910
CFH $\tilde{\mathbf{h}}^{(G)}$	-0.0714	-0.0538	-0.0199	-0.0803	0.0041	-0.0324	0.0199	0.1266	0.1296	0.0870	0.0018	0.1206
Pearson$(\tilde{\mathbf{h}}_i^{(v)}, \tilde{H}_i^{(v)}(Y))$	0.1390	-0.0759	-0.0272	0.0178	-0.1718	0.1539	0.1308	-0.0697	-0.0715	0.1523	0.0217	0.1721

$(*)$ For undirected graphs, their edges are counted as two directed edges.

$(**)$ $\tilde{H}_i^{(v)}(Y)$ is a node-level class-homophily measure, defined in Eq. (25) of Appendix B.

product, and two nodes are adjacent if the two products are frequently purchased together. For each node, the features are the bag-of-words of its customer reviews, and the node class is the product category.

- The *CS* and *Physics* datasets are coauthor networks (Shchur et al., 2018). Each node represents an author, and two nodes are adjacent if the two corresponding authors have coauthored a paper together. For each node, the features are the author's paper keywords, and the class is the most active field of the author's study.
- The *Chameleon* and *Squirrel* datasets are webpage networks of Wikipedia (Pei et al., 2020). Each node represents a webpage on Wikipedia, and two nodes are adjacent if mutual links exist between the two corresponding web pages. For each node, the features are informative nouns on the corresponding webpage, and the node class represents the category of the average monthly traffic of the corresponding webpage. We use the version of the datasets provided by Platonov et al. (2023b), which has filtered the possible duplicate nodes.
- The *Actor* dataset is the actor-only induced subgraph of a film-director-actor-writer network obtained from Wikipedia webpages (Tang et al., 2009; Pei et al., 2020). Each node represents an actor, and two nodes are adjacent if the two actors appear on the same Wikipedia webpage. For each node, the features are derived from the keywords on the Wikipedia webpage of the corresponding actor, and the node class is determined by the words on the webpage.
- The *Texas*, *Cornell*, and *Wisconsin* datasets are extracted from the WebKB dataset (Pei et al., 2020). Each node represents a webpage, and two nodes are adjacent if a hyperlink between the two webpages. For each node, the features are the bag-of-words features of the corresponding webpage, and the node class is the category of the webpage.
- The *Roman-Empire* dataset is a network of texts in a Wikipedia article (Platonov et al., 2023b). Each node represents each word in the article, and two nodes are adjacent if the words follow each other in the text or if one word depends on the other. For each node, the features is word embedding of the text, and the node class is the syntactic role of the text.
- The *Amazon-Ratings* dataset is a co-purchase network of Amazon products (Platonov et al., 2023b). Each node represents a product, and two nodes are adjacent if they are frequently purchased together. For each node, the features are text embedding of the product description, and the node class is its ratings.
- The *Tolokers* dataset is an online social network of Toloka crowdsourcing platform (Platonov et al., 2023b). Each node represents a worker, and two nodes are adjacent if they have worked on the same task. For each node, the features consist of the worker profile and task performance, and the node class is whether or not the worker has been banned.
- The *Penn94* dataset is a social network on Facebook (Lim et al., 2021). Each node represents a user, and two nodes are adjacent if they are friends. For each node, the features are the user profile, and the node class is the reported gender.
- The *Flickr* dataset is a network of images to Flickr website (Zeng et al., 2020). Each node represents an image, and two nodes are adjacent if the two share some common properties (e.g. the same geographic location). For each node, the features are bag-of-word representations of the image, and the class is the image's tag.
- The *ArXiv-Year* dataset (Lim et al., 2021) is a version of the *Ogbn-ArXiv* dataset, where the original node class is replaced with the article publication year.

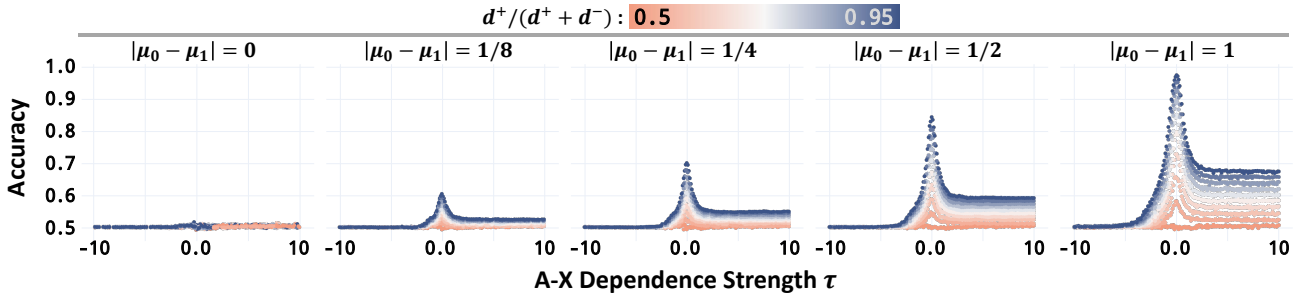


Figure 13: **The Simplified GNN Performance in CSBM-X Graphs: *Wider τ Range***. With $\tau \in \{-10, -9.9, \ldots, 9.9, 10\}$, the findings are consistent with those from Fig. 5.

D. In-Depth Analysis of CSBM-X

In this section, we provide the full experimental result with CSBM-X, together with its formal description. We use the same setting as in Sec. 4, unless otherwise specified.

D.1. Full Experiments: Large Range of τ

Fig. 13 shows the experimental results with larger range of $\tau \in \{-10, -9.9, \ldots, 9.9, 10\}$. That is, the CSBM-X graph $\mathcal{G}$ has extremely large CFH $\tilde{\mathbf{h}}(\cdot)$. Still, the results are consistent with the conclusion of Sec. 4.

D.2. Full Experiments: Feature Parameter Variations

High-dimensional features. Fig. 14 shows the experimental results with feature dimension $k \in \{4, 16\}$. Specifically, we let $\mu_0 = -\mu_1$, and all elements within each mean vector are identical (i.e., $\mu_0 = [c, c, \ldots, c] \in \mathbb{R}^k$ and $\mu_1 = [-c, -c, \ldots, -c] \in \mathbb{R}^k$, where c is a constant). To control FD, we generate CSBM-X graph $\mathcal{G}$'s with (*i*) $2c \in \{0,\ 1/8,\ 1/4,\ 1/2,\ 1\}$ and (*ii*) $\Sigma_0 = \Sigma_1 = \mathrm{diag}(\mathbf{1})$. The results are consistent with the conclusion of Sec. 4.

Imbalanced feature variances. Fig. 14 shows the experimental results with imbalanced feature variances (i.e., $\Sigma_0 \neq \Sigma_1$). To control FD with imbalanced feature variances, we generate CSBM-X graph $\mathcal{G}$'s with $\Sigma_0 = 1$ and $\Sigma_1 \in \{0.5, 0.25\}$. The results are consistent with the conclusion of Sec. 4.

D.3. Full Experiments: Graph Convolution Variations

The number of graph convolution layers. Fig. 15 shows the experimental results with two graph convolution layers. Specifically, we use $(D^{-1}A)^2X$ as the simplified GNN model. We find that with 2 layers, the beneficial effect of small τ is larger. The finding possibly relates to the sparse topology of the generated CSBM-X graph $\mathcal{G}$'s, [11] such that two convolution layers do not trigger over-smoothing. Overall, the results are consistent with the conclusion of Sec. 4.

Symmetric normalized graph convolution. Fig. 15 shows the experimental results with symmetrically normalized graph convolution. Specifically, we use $((D+I)^{-\frac{1}{2}}(A+I)(D+I)^{-\frac{1}{2}})X$ as the simplified GNN model, where $I \in \mathbb{R}^{n \times n}$ is the identity matrix. To do so, we conduct two pre-processing for CSBM-X graph $\mathcal{G}$. First, since the symmetric normalization assumes an undirected graph, we transform all its directed edges into (unweighted) undirected edges. Second, all nodes have added self-loops. Even with symmetric normalization, the results are consistent with the conclusion of Sec. 4.

The extensive experiments empirically support our conclusion that A-X dependence mediates the effect of graph convolution.

D.4. CSBM-X: Formal Description

Here, we provide a formal mathematical description of CSBM-X. Note that we use slightly different notations from Sec. 4.

Input. We consider a binary class setting (WLOG class 0 and 1). Each input parameter is as: number of nodes n (we assume that n is even), feature mean vector (μ_0, μ_1) and feature covariance matrix (Σ_0, Σ_1), each corresponds to the class ℓ, same- and different-class degree (d^+, d^-), respectively, and A-X dependence strength τ. Let $\mathcal{I} \coloneqq (n, \mu_0, \mu_1, \Sigma_0, \Sigma_1, d^+, d^-)$ denotes the set of input parameters.

[11] Recall that the number of nodes is 10,000, whereas the node degree is 20 for all nodes.

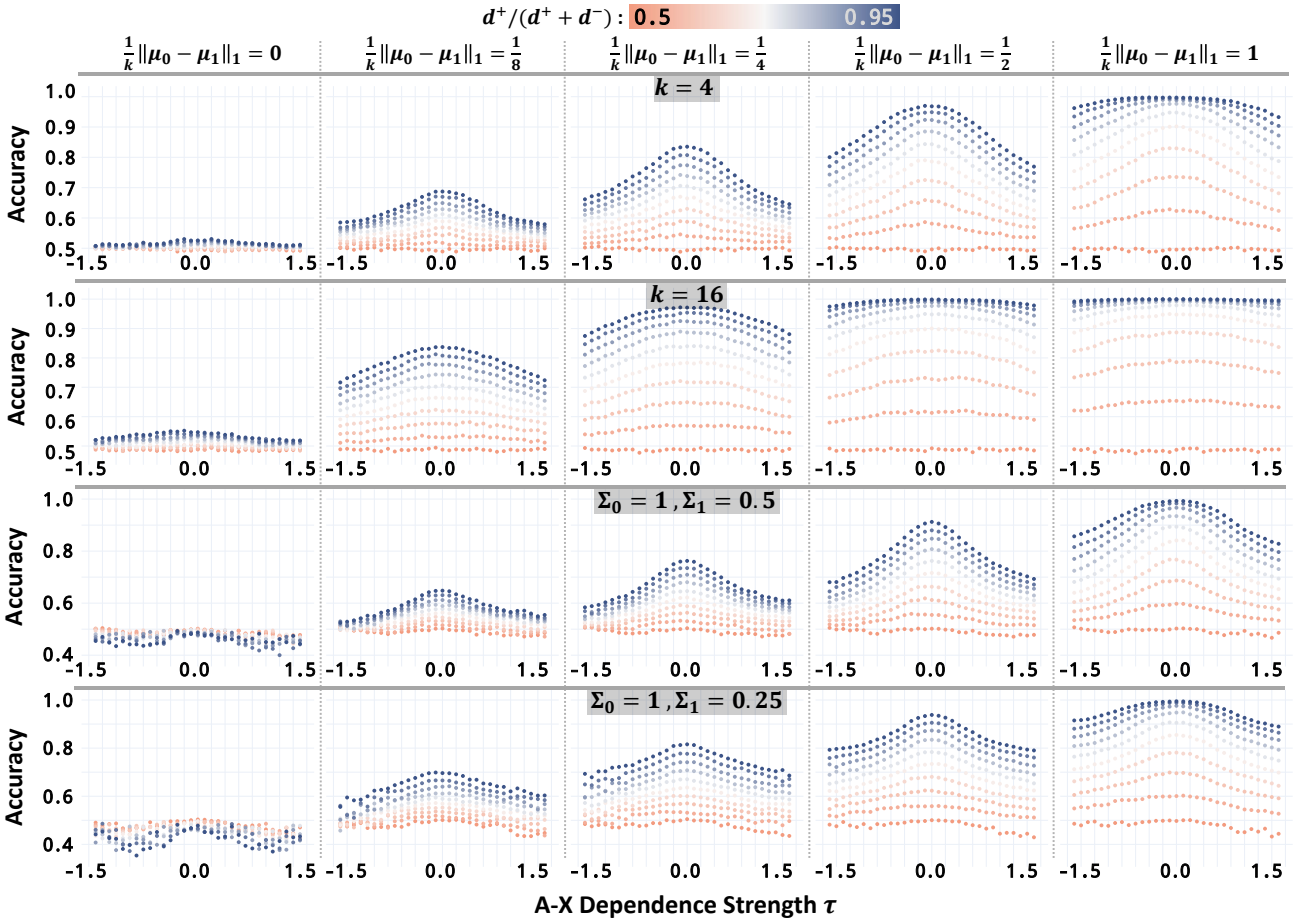


Figure 14: **The Simplified GNN Performance in CSBM-X Graphs: *Feature Variations*.** With different feature parameter configurations, including (*i*) high-dimensional features (i.e. $k > 1$) and (*ii*) imbalanced variances (i.e. $\Sigma_0 \neq \Sigma_1$), the findings are consistent with those from Fig. 5.

Node classes. Given input $\mathcal{I} = (n, \mu_0, \mu_1, \Sigma_0, \Sigma_1, d^+, d^-, \tau)$, the node set is (deterministically) $V = V(\mathcal{I}) = V(n) = [n]$, determined by n only. Hence, $v_i = i, \forall i \in [n]$. We assume that the numbers of nodes in two classes are the same, i.e., $\frac{n}{2}$. The node classes is represented by a vector

- $Y \in \mathcal{Y}_n \coloneqq \{\{0,1\}^n : \sum_{i \in [n]} Y_i = \frac{n}{2}\}$,

where $\mathcal{Y}_n$ is the possible set of node-class vectors. For each $Y \in \mathcal{Y}_n$,

- $\Pr[Y|\mathcal{I}] = \Pr[Y|n] = \frac{1}{|\mathcal{Y}_n|} = \frac{1}{\binom{n}{\frac{n}{2}}}$,

where the probability of Y is only decided by n, independent of the other parameters in the input $\mathcal{I}$, and all the possible Y's have the same probability (i.e., follow a uniform distribution on $\mathcal{Y}_n$).

Node features. Assume the feature dimension is $k \in \mathbb{N}$. Conditioned on the node classes Y, the features $X \in \mathbb{R}^{n \times k}$ follow the corresponding Gaussian distributions, where each node feature $X_i \in \mathbb{R}^k$ is an i.i.d. sample from a Gaussian with mean μ_{Y_i} and variance Σ_{Y_i}. Specifically,

- $\Pr[X_i|Y, \mathcal{I}] = \Pr[X_i|Y_i, \mu_{Y_i}, \Sigma_{Y_i}] = (2\pi)^{-k/2} \det(\Sigma_{Y_i})^{-1/2} \exp\left(-\frac{1}{2}(X_i - \mu_{Y_i})^{\mathrm{T}} \Sigma_{Y_i}^{-1} (X_i - \mu_{Y_i})\right)$,

which is the PDF of multivariate Gaussian $\mathcal{N}(\mu_{Y_i}, \Sigma_{Y_i})$.

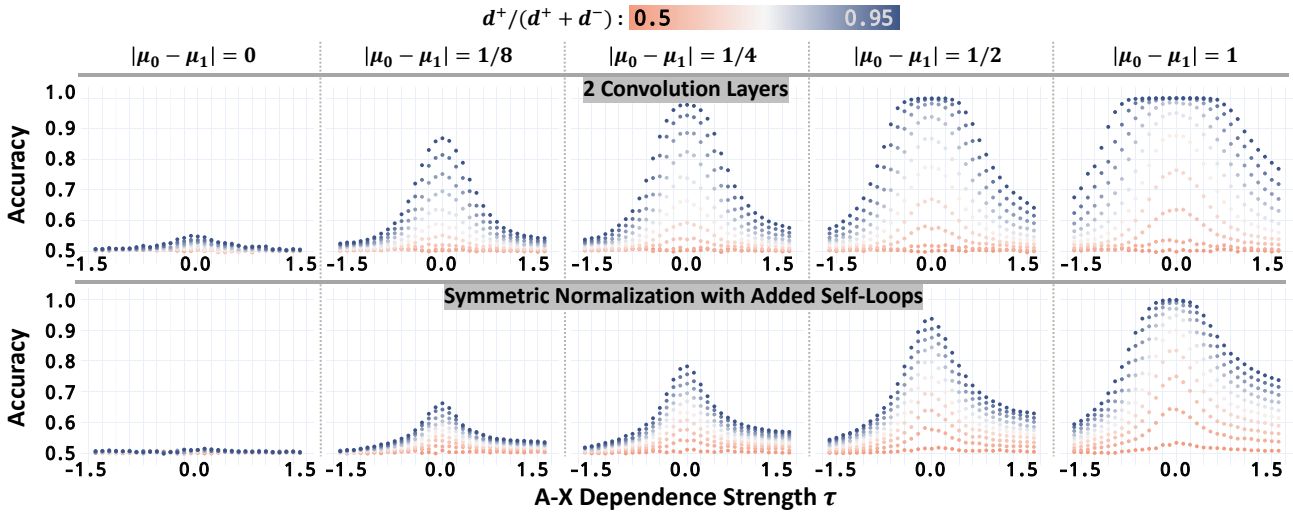


Figure 15: **The Simplified GNN Performance in CSBM-X Graphs: *GNN Variations*.** With two simplified GNN variations, including graph convolution with (*i*) two layers and (*ii*) symmetrically normalized adjacency matrix with self-loops, the findings are consistent with those from Fig. 5.

Edges. Conditioned on node classes and features, directed edges are sampled by weighted sampling without replacement. For each node $i \in [n]$, we define

- $\mathbf{C}_i^+ = \mathbf{C}_i^+(Y) = C_{Y_i}^+ \setminus \{v_i\} = \{\mathbf{C}_{i,1}^+, \mathbf{C}_{i,2}^+, \ldots, \mathbf{C}_{i,|\mathbf{C}_i^+|}^+\}$ (fix order of the nodes in $\mathbf{C}_i^+$),
- $\mathbf{C}_i^- = \mathbf{C}_i^-(Y) = C_{Y_i}^- = \{\mathbf{C}_{i,1}^-, \mathbf{C}_{i,2}^-, \ldots, \mathbf{C}_{i,|\mathbf{C}_i^-|}^-\}$,
- $\mathbf{\Phi}_i^+ = \mathbf{\Phi}_i^+(\mathbf{C}_i^+, X) = (\exp(\tau \mathbf{h}_{ij}^{(p)}), j = \mathbf{C}_{i,t}^+ : t \in [|\mathbf{C}_i^+|]) \in \mathbb{R}^{|\mathbf{C}_i^+|}$,
- $\mathbf{\Phi}_i^- = \mathbf{\Phi}_i^-(\mathbf{C}_i^-, X) = (\exp(\tau \mathbf{h}_{ij}^{(p)}), j = \mathbf{C}_{i,t}^- : t \in [|\mathbf{C}_i^-|]) \in \mathbb{R}^{|\mathbf{C}_i^-|}$.

Note that X is used here to compute $\mathbf{h}_{ij}^{(p)}$'s. Let $\mathbf{C}^+$ denote $(\mathbf{C}_i^+ : i \in [n])$, and $\mathbf{C}^-$, $\mathbf{\Phi}^+$, and $\mathbf{\Phi}^-$ are similarly defined.

For each node $i \in [n]$, let N_i denote its neighbor set, which is a random variable here. Recall that $N_i = \{j \in [n] : (i,j) \in E\}$. The set of all possible N_i's are

- $\mathcal{B}_i = \mathcal{B}_i(\mathbf{C}_i^+, \mathbf{C}_i^-, d^+, d^-) := \{N_i^+ \cup N_i^- : N_i^+ \in \mathcal{B}_i^+, N_i^- \in \mathcal{B}_i^-\}$, where
- $\mathcal{B}_i^+ = \mathcal{B}_i^+(\mathbf{C}_i^+, d^+) = \{N_i^+ : N_i^+ \subseteq \mathbf{C}_i^+, |N_i^+| = d^+\}$,
- $\mathcal{B}_i^- = \mathcal{B}_i^-(\mathbf{C}_i^-, d^-) = \{N_i^- : N_i^- \subseteq \mathbf{C}_i^-, |N_i^-| = d^-\}$.

For each possible $N_i^+ = \{N_{i,1}^+, N_{i,2}^+, \ldots, N_{i,d^+}^+\} \in \mathcal{B}_i^+$,

- $\Pr[N_i^+ | \mathbf{C}_i^+, \mathbf{\Phi}_i^+, d^+] = \sum_{\pi \in S_{d^+}} \prod_{t=1}^{d^+} (\mathbf{\Phi}_{N_{\pi(i,t)}^+}^+) / (1 - \sum_{t'=1}^{t-1} \mathbf{\Phi}_{N_{\pi(i,t')}^+}^+)$,
- $\Pr[N_i^- | \mathbf{C}_i^-, \mathbf{\Phi}_i^-, d^-] = \sum_{\pi \in S_{d^-}} \prod_{t=1}^{d^-} (\mathbf{\Phi}_{N_{\pi(i,t)}^-}^-) / (1 - \sum_{t'=1}^{t-1} \mathbf{\Phi}_{N_{\pi(i,t')}^-}^-)$,

where S_{d^+} and S_{d^-} are the sets of all permutations on $[d^+]$ and $[d^-]$, respectively.

For each possible $N_i = N_i^+ \cup N_i^- \in \mathcal{B}_i$,

- $\Pr[N_i | \mathbf{C}_i^+, \mathbf{\Phi}_i^+, \mathbf{C}^-, \mathbf{\Phi}_i^-, d^+, d^-] = \Pr[N_i^+ | \mathbf{C}_i^+, \mathbf{\Phi}_i^+, d^+] \Pr[N_i^- | \mathbf{C}^-, \mathbf{\Phi}_i^-, d^-]$.

The neighbor set of each node is sampled independently, i.e.,

- $\Pr[(N_i{:}i \in [n])|\mathbf{C}_i^+, \mathbf{\Phi}_i^+, \mathbf{C}^-, \mathbf{\Phi}_i^-, d^+, d^-] = \prod_{i\in[n]} \Pr[N_i|\mathbf{C}_i^+, \mathbf{\Phi}_i^+, \mathbf{C}^-, \mathbf{\Phi}_i^-, d^+, d^-]$.

Here, $N_i, \forall i \in [n]$ fully determine the topology of each generated graph.

D.5. CSMB-X2: a Multi-Community, Degree-Preserving CSBM-X

While the proposed CSBM-X can capture many properties of the real-world graphs, such as sparsity, class-homophily, and CFH, it does not reflect some of their key properties. Thus, we further propose CSBM-X2 to reflect a multi-community structure and varying degree distributions.

CSBM-X2 design. Two major differences are noticeable in comparison to CSBM-X. First, the number of classes can be larger than 2, allowing for a multi-community structure. Second, the vector $\mathbf{d} \in \mathbb{R}^n$ representing node degrees for each node serves as the degree parameter.

We describe the key differences of the CSBM-X2. Specifically, to allow for a multiple community structure, the nodes are equally divided into c classes (instead of 2 in CSBM-X). Also, for each node $v_i \in V$, CSBM-X2 assigns its node degree $\mathbf{d}_i$. By the same-class degree ratio parameter r, the node v_i's degree $\mathbf{d}_i$ is divided into the same-class and different-class degrees (d_i^+, d_i^-). Then, CSBM-X2 samples same- and different-class neighbors for each node by their sampling weights, like in CSBM-X.

Experiment setting. We generate CSBM-X2 graphs with various parameter configurations. The setting generally follows the one in Sec. 4.3. We fix the number of nodes $n = 10,000$ and number of classes $c = 10$. We use input node degrees $\mathbf{d}$ sampled from Pareto distribution (a power-law distribution), using Pareto's $\alpha = 1.5$. Since the sampled values are bounded by $[0, \infty)$, $d_{min} = 20$ is added to the sampled value, and $d_{max} = 1,000$ is used to clip it. Thus, the node degree is bounded $d_i \in [d_{min}, d_{max}], \forall v_i \in V$.

We set c-dimensional features, such that the feature mean vectors are $\mu_\ell \in \mathbb{R}^c$ and covariance matrices are $\Sigma_\ell \in \mathbb{R}^{c\times c}, \forall \ell \in [c]$. The generated CSBM-X2 graphs have a wide range of feature distance FD, class homophily $\mathbf{h}_c$, and CFH $\tilde{\mathbf{h}}(\cdot)$.

- FD: $\|\mu_{\ell_0} - \mu_{\ell_1}\|_1 \in \{0,\ 1/4,\ 1/2,\ 1,\ 2\}$; $\Sigma_\ell = \mathbf{I}$.
- $\mathbf{h}_c$: $r \in \{0.50, 0.55, ..., 0.95\}$.
- $\tilde{\mathbf{h}}(\cdot)$: $\tau \in \{-1.5, -1.4, \ldots, -0.1, 0, 0.1, \ldots, 1.4, 1.5\}$.

Experiment results. Figure 16 shows the experimental results with CSBM-X2. We observe consistent findings from the experiments with CSBM-X (Figure 5), such that the findings 1 and 2 from Sec. 4.3 are reproduced with CSBM-X2. Specifically, the simplified GNN performance gradually increases over decreasing CFH magnitude $|\tilde{\mathbf{h}}(\cdot)|$, and the effect of CFH $\tilde{\mathbf{h}}(\cdot)$ on GNN performance is moderated by feature distance FD and class homophily $\mathbf{h}_c$. Our results highlight the generalizability of our conclusion that A-X dependence mediates the effect of graph convolution.

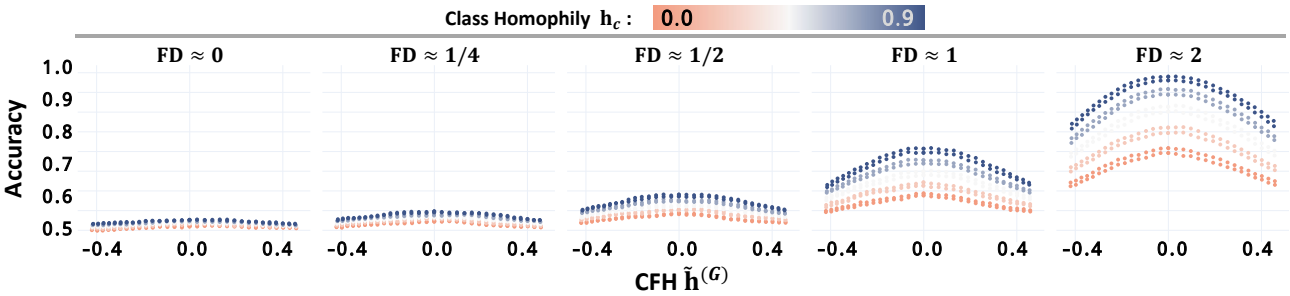
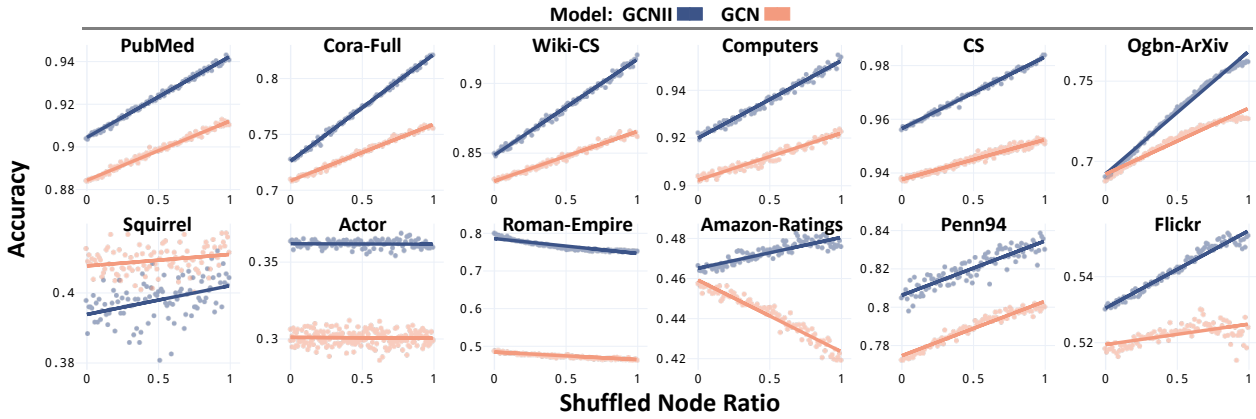
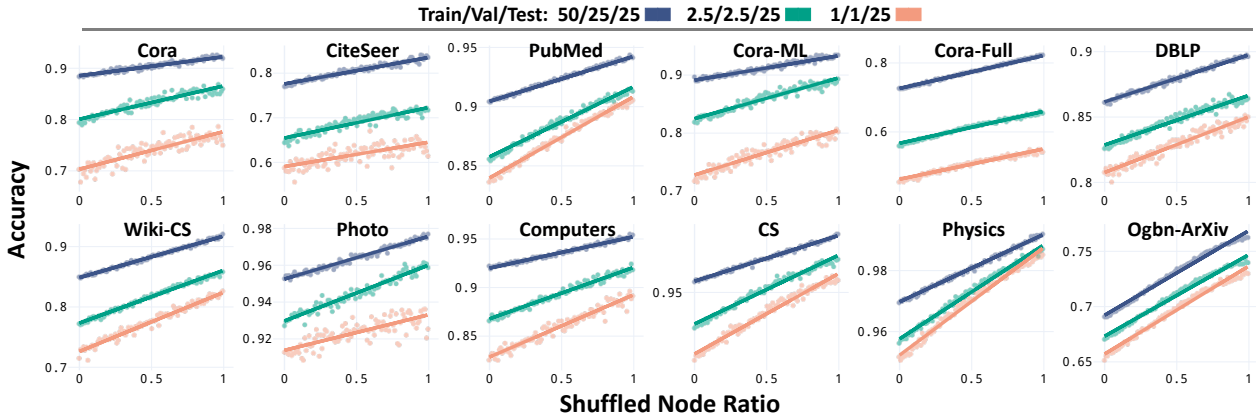


Figure 16: **The Simplified GNN Performance in CSBM-X2 Graphs.** The degrees follow the power-law distribution, and the graphs have multi-community structures. Consistent with Theorem 4.3 and Figure 5, for given feature distance FD > 0 and class homophily $\mathbf{h}_c > 0$, the simplified GNN performance increases as graph-level CFH $\tilde{\mathbf{h}}^{(G)} \to 0$ $(\tau \to 0)$.



(a) Additional results on the feature shuffle: Models



(b) Additional results on the feature shuffle: Train node ratio

Figure 17: **Additional Results on the Feature Shuffle.**

E. Additional Experimental Results with Feature Shuffle

In this section, we provide additional experimental results of the feature shuffle with real-world graphs. We use the same setting as in Sec. 5, unless otherwise specified.

Models. First, Fig. 17(a) shows GCN performance after the feature shuffle in 6 high $\mathbf{h}_c$ and 6 low $\mathbf{h}_c$ datasets. While GCN performance increases over the feature shuffles, GCNII benefits more from the feature shuffle (i.e., larger positive slopes by GCNII). This outcome may relate to the difference in their number of layers. We claim two complementary pieces of evidence. For one, CSBM-X experiment in Fig. 15 suggests that a larger number of layers can further improve the beneficial effect of small τ. Also, one of the main differences between the GCN and GCNII is their capability in stacking deeper layers. The relationship between GNN depth and A-X dependence, however, is beyond the scope of the present work, and we leave it up to future studies. Overall, consistent with the conclusion of Sec. 5, we conclude that all GNN models benefit from the feature shuffle.

Train ratios. Second, Fig. 17(b) shows GCNII performance over the feature shuffle with different splits. We use three different train/val splits while fixing the test split. Two findings are worth noting. First, model performance increases over the feature shuffles in all splits, highlighting that our conclusion is consistent with varying train and validation node ratios. Second, the performance gap between different splits generally reduces over the increasing shuffled node ratio. That is, the effect of feature shuffle may also interact with the number of train labels, hinting that A-X dependence may influence the generalization capacity of GNNs. Analysis of GNN generalization, however, is beyond the scope of the present work, and we leave it up to future studies.

The extensive experiments empirically support our conclusion that A-X dependence mediates the effect of graph convolution.

Table 3: Node Classification Performance of a GNN Enhanced with Pseudo-label-based Feature Shuffle Algorithm

Method	Cora	CiteSeer	PubMed	Cora-ML	DBLP	Wiki-CS	Cora-Full	Photo	Computers	CS	Physics	Ogbn-ArXiv
Vanila GCNII	79.70	67.26	76.23	82.05	75.42	73.37	60.94	90.44	81.10	91.31	92.97	51.60
Enhanced GCNII	80.83	68.02	76.26	82.77	77.07	73.16	61.58	90.50	81.66	91.27	93.31	50.80
Performance Gap	+1.13	+0.76	+0.03	+0.72	+1.65	-0.21	+0.64	+0.06	+0.56	-0.04	+0.34	-0.80

F. Potential Applications

In this section, we explore a potential application of our findings with node feature shuffle. As discussed in Section 5, the feature shuffle can significantly enhance GNN node classification. However, the algorithm requires test labels, which are not known. We, thus, propose an algorithm that extends the feature shuffle for practical scenarios where the labels of many nodes are unknown.

Algorithm. Here, we describe our proposed pseudo-label-based feature shuffle algorithm. The algorithm has three steps:

Step 1) Initial GNN training. Consider we are given a graph $G = (V, E)$, node features X, and node classes of train/val nodes $V^{(train/val)} \subset V$. We train a node classification GNN, which we denote as $\texttt{GNN} : (X, E) \mapsto \hat{\mathbf{Y}}$, where $\hat{\mathbf{Y}} \in [0,1]^{n \times c}$ is a node class prediction matrix.

Step 2) Pseudo-labeling. By using the trained GNN, we assign pseudo-labels to unlabeled nodes. First, we choose nodes to be labeled by using confidence scores of GNN. To this end, we use prediction matrix $\hat{\mathbf{Y}}$ of GNN. We label nodes whose prediction probability lies within the predefined confidence range. Specifically, we define labeling nodes $V^{(label)}$ as follows:

$$V^{(label)} = \{v_k \in V : o_l \leq \max_{j \in [c]} \hat{\mathbf{Y}}_{kj} \leq o_u\} \cup V^{(train/val)}, \tag{26}$$

where $0 \leq o_l \leq o_u \leq 1$ are hyperparameters. After choosing labeling nodes $V^{(label)}$, we assign pseudo-labels $y'_k, \forall v_k \in V^{(label)}$ as follows:

$$y'_k = \begin{cases} Y_k & \text{if } v_k \in V^{(train/val)} \\ \arg\max_{j \in [c]} \hat{\mathbf{Y}}_{kj} & \text{otherwise} \end{cases}, \forall v_k \in V^{(label)}. \tag{27}$$

Step 3) Classwise feature shuffle. Lastly, with obtained pseudo-labels $y_k, \forall v_k \in V^{(label)}$, we shuffle node features that share the same pseudo-label. Specifically, we shuffle the features of all nodes that share the same pseudo-labels, without considering the train/val/test split. Train/val/test node indices and true node labels are not shuffled. By performing the shuffle, we obtain new node features X'.

Step 4) GNN fine-tuning. Finally, we fine-tune the trained GNN with the shuffled node features X'. Specifically, $\texttt{GNN} : (X', E) \mapsto \hat{\mathbf{Y}}$. We make the test inference with the fine-tuned GNN and shuffled feature X'.

Experiment. We test the algorithm on the node classification benchmark datasets. We use 12 high class-homophily graphs. Similar experimental procedures and hyperparameter settings are used as in Sec. 5.1 and Appendix G. To focus on more practical scenarios, we use 20 train nodes and 30 validation nodes per class, whereas the rest of the nodes serve as the test nodes. We use GCNII as the GNN encoder for the pseudo-label-based feature shuffle algorithm. For pseudo-labeling hyperparameters, we fix $o_l = 0.7$ and $o_u = 1$. We compare the performance of vanilla GCNII versus GCNII enhanced with pseudo-label-based feature shuffle. We repeat 30 trials and report the mean performance in Table 3. In 9/12 datasets, the enhanced GCNII outperforms the vanilla GCNII by a small margin.

G. Experiment Settings: Pre-processing, Training, Hyperparameters, and Details

G.1. Dataset Pre-processing

Measurement. No dataset pre-processing is done when measuring class homophily $\mathbf{h}_c$ and feature distance FD. If the dataset has self-loops, they are removed when measuring CFH $\tilde{\mathbf{h}}(\cdot)$.

CSBM-X. For experiments with symmetrically normalized graph convolution in Fig. 15, (*i*) directed edges are converted into undirected edges (without edge weights) and (*ii*) self-loops are added. In other experiments, no dataset pre-processing is done.

The real-world graphs. All the considered GNN models assume undirected graph topology. Thus, directed edges are converted into undirected edges (without edge weights). Also, self-loops are added.

G.2. Model Training

All models are trained with Adam (Kingma & Ba, 2015) optimizer. We fix 500 train epochs. The best model is chosen based on early stopping, with a patience of 100. In feature shuffle experiments (Sec. 5, Appendix E), a new model is initialized and trained for each shuffled graph.

G.3. Hyperparameters

In CSBM-X experiments (Sec. 4, Appendix D), we do not tune hyperparameters since the coefficient $W \in \mathbb{R}$ is the only learnable parameter. In feature shuffle experiments (Sec. 5, Appendix E), we tune the hyperparameters on the *original graphs*. That is, the feature-shuffled graphs are unknown to the models during the hyperparameter search.

For all models, we set their hidden feature dimension as 64 and the learning rate as 0.01. Below, we provide the hyperparameter search space for each considered model.

1. **GCN**:

- Optimizer weight decay $\in \{5e-3, 1e-3, 5e-4, 1e-4\}$
- Dropout $\in \{0.5, 0.6, 0.7\}$
- Number of layers $\in \{2, 3, 4\}$

2. **GCN-II**:

- Optimizer weight decay $\in \{1e-3, 5e-4, 1e-4, 5e-5\}$
- Dropout $\in \{0.5, 0.6, 0.7\}$
- Number of layers $\in \{4, 8, 16\}$
- Residual connection weight $\alpha \in \{0.1, 0.3, 0.5\}$
- Weight decay $\lambda \in \{0.5, 1.0, 1.5\}$

3. **GPR-GNN**:

- Optimizer weight decay $\in \{5e-3, 1e-3, 5e-4, 1e-4\}$
- Dropout $\in \{0.5, 0.6, 0.7, 0.8\}$
- Number of layers $\in \{10\}$
- Return probability $\alpha \in \{0.1, 0.3, 0.5\}$

4. **AERO-GNN**:

- Optimizer weight decay $\in \{5e-3, 1e-3, 5e-4\}$
- Dropout $\in \{0.5, 0.6, 0.7\}$
- Number of MLP layers $\in \{1, 2\}$
- Number of convolution layers $\in \{4, 8, 16\}$
- Weight decay $\lambda \in \{0.5, 1.0, 1.5\}$

G.4. Other Details

Train/val/test split. For each node class, the train/val/test set is split randomly by the ratio of 50/25/25, unless otherwise specified. In CSBM-X experiments (Sec. 4, Appendix D), for each generated CSBM-X graph $\mathcal{G}$, we obtain 5 different splits. In feature shuffle experiments (Sec. 5, Appendix E), we use 5 different splits consistent across the shuffled node ratio.

Node2Vec. For Fig. 10, we use Node2Vec (Grover & Leskovec, 2016) as the node features. For each graph, the Node2Vec vector is 256-dimensional. To obtain the vector, we train the Node2Vec model with a walk length of 20, a context size of 10, walks per node of 10, and 100 epochs.

Shuffle and CFH $\tilde{\mathrm{h}}(\cdot)$. When measuring CFH after the feature shuffle, we average the outcomes over 5 trials.

Evaluation of WebKB datasets. The WebKB datasets (i.e., Texas, Cornell, and Wisconsin) have very small number of nodes, ranging from 183 to 251. Thus, the variance of GNN node classification accuracy on the datasets often tend to be very large, because mis-classifying one test node can drop near 2%p test accuracy. To enhance reliability of the empirical results in Sec. 5, we report the mean performance *over 30 trials* only for the WebKB datasets.