

---

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

---

000  
001  
002  
003  
004  
005  
006  
007  
008  
009  
010  
011  
012  
013  
014  
015  
016  
017  
018  
019  
020  
021  
022  
023  
024  
025  
026  
027  
028  
029  
030  
031  
032  
033  
034  
035  
036  
037  
038  
039  
040  
041  
042  
043  
044  
045  
046  
047  
048  
049  
050  
051  
052  
053  
054  
Anonymous Authors<sup>1</sup>

## Abstract

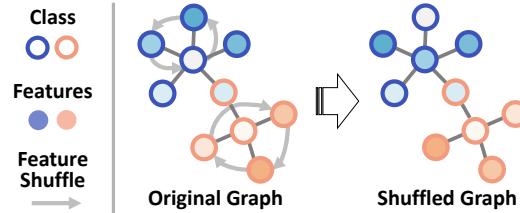
How would randomly shuffling feature vectors among nodes from the same class affect graph neural networks (GNNs)? The feature shuffle, intuitively, perturbs the dependence between graph topology and features (A-X dependence) for GNNs to learn from. Surprisingly, we observe a consistent and significant 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

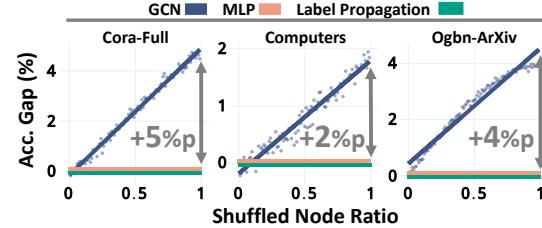
Graph neural networks (GNNs) are functions of graph topology and features. Understanding the conditions in which GNNs become powerful is the key to their improvement and effective applications. As such, many prior works have investigated the conditions that affect GNN effectiveness, especially from the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

<sup>1</sup>Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.



(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 (Kip & 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 satisfactory understanding of the phenomenon. One line of studies indicates how label 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 labels are not 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?

Our investigation concludes that A-X dependence mediates the effect of graph convolution. Specifically, A-X dependence moderates the force to pull each node feature toward the feature mean of the respective node class, with a smaller A-X dependence improving GNN-based node classification. Our key contributions are summarized as follows:

- **Measure.** We (i) propose a principled measure, *class-controlled feature homophily* (CFH), for A-X dependence, while mitigating potential confounding by node class.
- **Graph Model and GNN Theory.** We (ii) propose a *random graph model*, CSBM-X, that controls CFH. In CSBM-X graphs, we (iii) prove that the Bayes error rate of a simplified GNN is proportional to CFH.
- **Observations and Experiments.** In 24 real-world graphs, we (iv) observe that CFH is astonishingly small and (v) demonstrate that the feature shuffle interferes with CFH to improve GNN performance.

**Reproducibility:** The code is at [anonymous Github](#).

## 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  $\ell$ , and  $C_\ell^- = V \setminus C_\ell^+$  denotes the rest.

**Feature distance.** For measuring feature distance FD 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:

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

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**  $\text{FD}(C_0, C_1) \in [0, \infty)$  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. Measure Design

We target two central goals in designing A-X dependence measure  $\tilde{\mathbf{h}}(\cdot)$ . First, the measure  $\mathbf{h}(\cdot)$  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 the A-X dependence. To achieve the design goals, we propose Class-controlled Feature Homophily (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**  $X|Y$ .

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

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 the **class-controlled features**  $X|Y$ . Let us define a distance function.

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

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

Recall that  $V'_i = V \setminus \{v_i\}$ . 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 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  $\mathbf{h}_{ij}^{(p)}$ :

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

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

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

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

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

Simply put, CFH  $\mathbf{h}(\cdot)$  measures neighbor distance relative to homophily baseline  $b(\cdot)$ , and it meets the two discussed design goals. 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 CFH  $\mathbf{h}(\cdot)$  for good mathematical properties (Lemma 3.1–3.3), which allows for its intuitive interpretation (discussed in Sec. 3.2).<sup>1</sup>

**N1)** Node-level normalization:

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

**N2)** Graph-level normalization:

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

**Lemma 3.1.** (Boundedness)  $\tilde{\mathbf{h}}^{(G)}, \tilde{\mathbf{h}}_i^{(v)} \in [-1, 1]$ , 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  $\tilde{\mathbf{h}}(G | X, \cdot) = \tilde{\mathbf{h}}(G | cX, \cdot)$ ,  $\forall c \in \mathbb{R} \setminus \{0\}$ .

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

All the proofs are in Appendix A.

#### 3.2. 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 random nodes and, thus, *homophilic*.  $\tilde{\mathbf{h}}_i^{(v)} < 0$  means that the node  $v_i$  is farther to its neighbors than 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), e.g., (i) when the neighbors  $N_i$  of the node  $v_i$  are chosen uniformly at

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

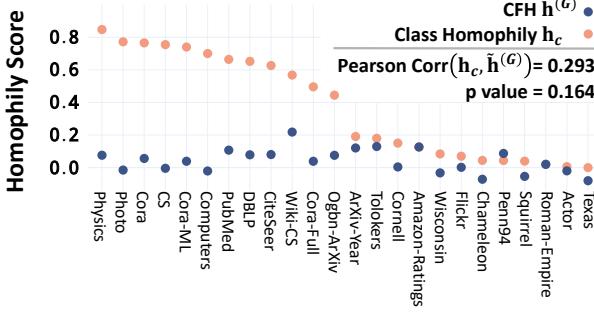


Figure 2: **Benchmark Graph Statistics.** Graph-level CFH  $\tilde{h}^{(G)}$  (i) are generally positive and small, with (ii) low correlation to class homophily  $h_c$ .

random from all the other nodes regardless of their features, or (ii) when all the nodes have the same feature.

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

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

Intuitively, when  $\tilde{h}_i^{(v)} > 0$  (or  $\tilde{h}_i^{(v)} < 0$ ), the node  $v_i$  is  $\frac{|\tilde{h}_i^{(v)}|}{1-|\tilde{h}_i^{(v)}|}$  times closer (or farther) to its neighbors than to random nodes. The detailed derivations are in Appendix B.1.

**Graph-level interpretation.** A graph-level CFH  $\tilde{h}^{(G)}$  is an aggregation of **node-level CFH**  $\tilde{h}_i^{(v)}$ 's. Details are in Appendix B.1.

### 3.3. Patterns in Benchmark Datasets

Here, we analyze node classification benchmark datasets using CFH  $\tilde{h}(\cdot)$ . The dataset details are in Appendix C.5.

First, we measure the graph-level CFH  $\tilde{h}^{(G)}$  in 24 datasets (Figure 2). Most of the graphs (23 out of 24) have  $\tilde{h}^{(G)}$  below 0.13, and 16 graphs have positive  $\tilde{h}^{(G)}$ . Their mean  $|\tilde{h}^{(G)}|$  is 0.06. Recall that the full reachable range of  $\tilde{h}^{(G)}$  is  $[-1, 1]$  (Lemma 3.1). We also analyze CFH at the node level (see Appendix C.1).

**Observation 1.** The benchmark graphs tend to show small, positive CFH  $\tilde{h}(\cdot)$ .<sup>2</sup>

We further analyze the relationship between the **graph-level CFH**  $\tilde{h}^{(G)}$  and class homophily  $h_c$  (Figure 2). 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). We find consistent results in node-level correlation analysis, described in Appendix C.2.

**Observation 2.** In the benchmark graphs, CFH  $\tilde{h}(\cdot)$  and

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

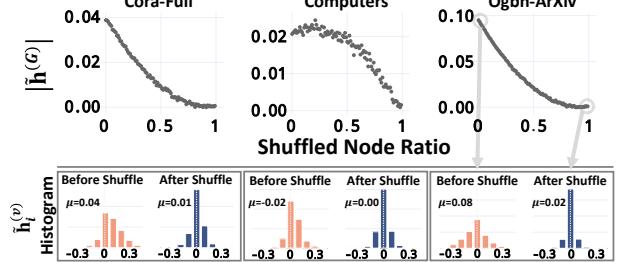


Figure 3: **The Effect of Feature Shuffle on CFH.** Both graph- and node-level CFH,  $\tilde{h}^{(G)}$  and  $\tilde{h}_i^{(v)}$ , tend to approach zero over the feature shuffles.

class homophily  $h_c$  have a small, positive correlation.

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

Lastly, we examine how the feature shuffle (recall Figure 1(a)) affects CFH  $\tilde{h}(\cdot)$ . For graph-level CFH  $\tilde{h}^{(G)}$ , increasing the shuffled node ratio reduces its magnitude  $|\tilde{h}^{(G)}|$  (Figure 3). Also, the distribution of node-level CFH  $(\tilde{h}_i^{(v)})$ '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. In Appendix C.3, we provide more details with analyses and reasonings.

**Observation 3.** CFH  $\tilde{h}(\cdot)$  tends to approach zero after shuffling the features of nodes from the same class.

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

## 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)  $\tilde{h}(\cdot)$  with a graph model, we propose CSBM-X (Algorithm 1). Compared to the previous CSBMs, CSBM-X is equipped with a new parameter  $\tau$  that controls the strength of A-X dependence. We provide verbal description and algorithm 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 (Lines 2-3). For each node  $v_i$ , based on its class  $Y_i$ , CSBM-X samples its fea-

ture  $X_i$  from a Gaussian distribution with a mean vector  $\mu_{Y_i}$  and a covariance matrix  $\Sigma_{Y_i}$  (Line 5). Then, directed edges are sampled based on the node features and classes, where the sampling weights are determined by the node features and  $\tau$ . A positive (or negative)  $\tau$  exaggerates edge sampling weights among the node pairs with higher (or lower) pair-level CFH  $\mathbf{h}_{ij}^{(p)}$  (Lines 8-9). Finally, for each node,  $d^+$  same-class and  $d^-$  different-class neighbors are sampled with weighted sampling without replacement (WS<sub>wr</sub>; Lines 10-12), returning a CSBM-X graph  $\mathcal{G}$  (Line 14).

**CSBM-X properties.** The key innovation of CSBM-X involves satisfying good properties w.r.t. its control over the dependence among classes  $Y$ , features  $X$ , and graph topology  $A$ . First, the parameters  $\mu_\ell$  and  $\Sigma_\ell$  control **feature distance** FD (Eq. (1); X-Y dependence). Second, the parameters  $d^+$  and  $d^-$  control **class homophily**  $\mathbf{h}_c$  (Eq. (2); A-Y dependence). Last, the parameter  $\tau$  controls CFH  $\tilde{\mathbf{h}}(\cdot)$  (Eq. (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 (or CFH  $\tilde{\mathbf{h}}(\cdot)$ ), satisfying two additional good properties.

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

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

The proofs of the above two lemmas are in Appendix A. In concert, the above properties highlight that CSBM-X flexibly, yet precisely, controls the dependence among classes, features, and topology in the generated graphs.

## 4.2. Graph Convolution in CSBM-X Graphs: Theory

**Analysis setting.** We assume that the features are (i) 1-dimensional ( $\mu_\ell, \Sigma_\ell \in \mathbb{R}$ ) and (ii) symmetric with identical variances ( $\mu_0 = -\mu_1 < 0$  and  $\Sigma_0 = \Sigma_1 = \mathbf{1}$ ). We use asymptotic setting with (iii) the number of nodes  $n \rightarrow \infty$ . (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 step of graph convolution, the

<sup>3</sup>In CSBM-X,  $(v_i, v_j)$  denotes a directed edge from  $v_i$  to  $v_j$ .

### Algorithm 1 CSBM-X

```

1: Input: number of nodes  $n$ , feature mean vector  $\mu_\ell$ 's and covariance matrix  $\Sigma_\ell$ 's, same- and different-class degree  $d^+$  and  $d^-$ , and A-X dependence strength  $\tau$ .
2: /* Step 0. Initialize nodes, edges, and class */
3:  $V \leftarrow [n]$ ,  $E \leftarrow \emptyset$ 
4:  $Y \leftarrow$  a random permutation of  $[\mathbf{0}_{\frac{n}{2}} \| \mathbf{1}_{\frac{n}{2}}]$ 
5: /* Step 1. Sample node features */
6: for  $v_i \in V$  do
7:    $X_i \sim \mathcal{N}(\mu_{Y_i}, \Sigma_{Y_i})$ 
8: end for
9: /* Step 2. Sample directed edges */
10: for  $v_i \in V$  do
11:    $\mathbf{C}_i^+ \leftarrow [j]_{v_j \in C_{Y_i}^+ \setminus \{v_i\}}$ ,  $\Phi_i^+ \leftarrow [e^{\tau \mathbf{h}_{ij}^{(p)}} : j = \mathbf{C}_{i,t}^+]_{t=1}^{|C_{Y_i}^+|-1}$ 
12:    $\mathbf{C}_i^- \leftarrow [j]_{v_j \in C_{Y_i}^-}$ ,  $\Phi_i^- \leftarrow [e^{\tau \mathbf{h}_{ij}^{(p)}} : j = \mathbf{C}_{i,t}^-]_{t=1}^{|C_{Y_i}^-|}$ 
13:    $N_i^+ \leftarrow \text{WS}_{\text{wr}}(\mathbf{C}_i^+, \Phi_i^+, d^+)$ 
14:    $N_i^- \leftarrow \text{WS}_{\text{wr}}(\mathbf{C}_i^-, \Phi_i^-, d^-)$ 
15:    $E \leftarrow E \cup \{(v_i, v_j)\}, \forall v_j \in (N_i^+ \cup N_i^-)$  3
16: end for
17: return  $\mathcal{G} := \mathcal{G}(n, \mu_\ell, \Sigma_\ell, d^+, d^-, \tau) = (V, E)$  (**)
```

(\*) WS<sub>wr</sub>( $\mathbf{C}, \Phi, d$ ) denotes weighted sampling without replacement given a vector of population  $\mathbf{C}$ , a sample weight vector  $\Phi$ , and the sample size  $d$ .

(\*\*) Each node  $v_i \in V$  has class  $Y_i$  and features  $X_i$ .

feature means of the two classes are constant and symmetric regardless of parameter  $\tau$ . 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  $\tau$ , 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 the classifier  $\mathcal{F}$  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  $\tau$ , 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  $\tau_0$  and  $\tau_1$  such that  $|\tau_0| < |\tau_1|$  and  $\tau_0 \tau_1 > 0$ .

**Proof sketch.** WLOG (due to symmetry), we 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.  $X_i = \mu_1 + x_i$ ).<sup>4</sup>

<sup>4</sup>Recall that  $x_i = (X_i | Y)$  in Eq. (3).

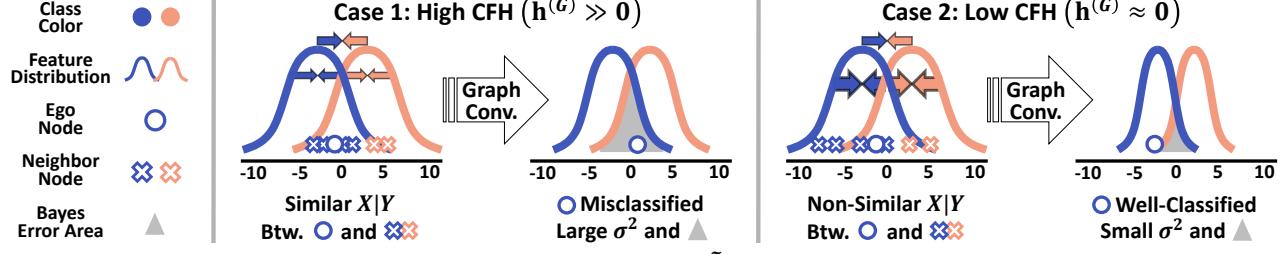


Figure 4: **Visual Intuition of Theorem 4.3.** When CFH is low ( $\tilde{h}^{(G)} \approx 0$ ), the feature distribution of each class shrinks faster (denoted by the arrows) by graph convolution, resulting in a lower Bayes error rate. Namely, the power to pull node features towards the feature mean of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

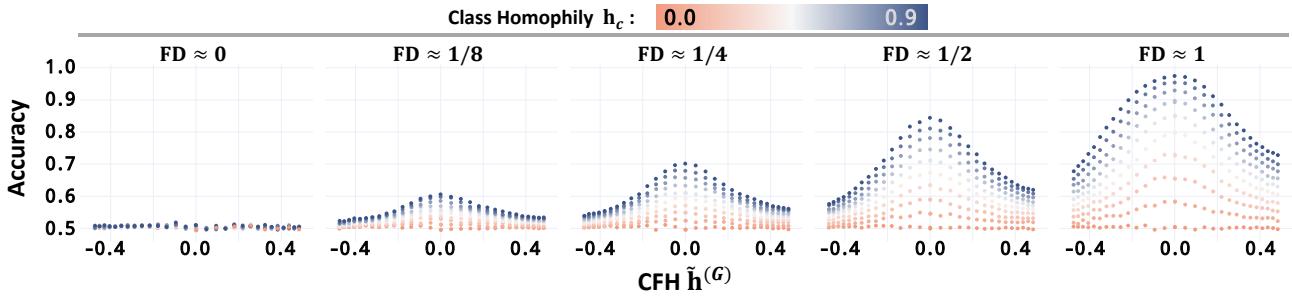


Figure 5: **The Simplified GNN Performance in CSBM-X Graphs.** Consistent with Theorem 4.3, for given **feature distance**  $FD > 0$  and **class homophily**  $h_c > 0$ , the simplified GNN performance increases as **graph-level CFH**  $\tilde{h}^{(G)} \rightarrow 0$  ( $\tau \rightarrow 0$ ).

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.

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

where  $x_N$  denotes the class-controlled feature of each neighbor of  $v_i$  and  $\operatorname{erfc}$  denotes the complementary error function of the Gaussian error function.

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

$$\begin{aligned}\mathcal{B}_F(\mathcal{G}(\cdot, \tau)) &= \int_{-\infty}^{\infty} \Pr[\mathcal{F}(x_i) = 0] \Pr[x_i | Y_i = 1] dx_i \\ &\xrightarrow{n \rightarrow \infty} \int_{-\infty}^{\infty} \mathbf{1}[\mathbb{E}[x_N] < \frac{d^- - d^+}{d^- + d^+} \mu_1] \varphi(x_i) dx_i,\end{aligned}\quad (10)$$

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

**Summary.** For each class  $\ell$ 's convoluted feature distribution, **degree parameters**  $(d^+, d^-)$  and **the feature mean parameter**  $\mu_\ell$  determine the distribution mean, while **A-X**

**dependence strength parameter**  $\tau$  determines the distance between each node and the mean (or the distribution variance; see Eq. (10) and Figure 4). Thus, the simplified GNN's Bayes error rate  $\mathcal{B}_F(\mathcal{G}(\cdot, \tau))$  decreases as  $|\tau|$  decreases, reaching its minimum at  $\tau = 0$  (Theorem 4.3).<sup>5</sup> With Lemma 4.1, we conclude that  $\mathcal{B}_F(\mathcal{G}(\cdot, \tau))$  is the lowest when CFH  $\tilde{h}^{(G)}$  is near 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 follow univariate Gaussian distributions. The CSBM-X graphs have a wide range of **feature distance**  $FD$ , **class homophily**  $h_c$ , and **CFH**  $\tilde{h}(\cdot)$ .

- **FD:**  $|\mu_0 - \mu_1| \in \{0, 1/8, 1/4, 1/2, 1\}; \Sigma_\ell = 1$
- **$h_c$ :**  $d^+ \in \{10, 11, \dots, 18, 19\}; d^- = 20 - d^+$ ,
- **$\tilde{h}(\cdot)$ :**  $\tau \in \{-1.5, -1.4, \dots, -0.1, 0, 0.1, \dots, 1.4, 1.5\}$ .

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.

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

**Finding 1** (*Effect of  $\tilde{h}(\cdot)$* ). Figure 5 shows that, given **class homophily**  $h_c > 0$  (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 highest accuracy when **graph-level CFH**  $\tilde{h}^{(G)} \approx 0$ . The accuracy decreases as  $|\tilde{h}^{(G)}|$  increases, in both positive and negative directions.

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

If **feature distance**  $FD$  and **class homophily**  $h_c$  are moderate-sized, the convoluted feature means of the two classes would be *somewhat distant*. Small CFH  $\tilde{h}(\cdot)$ , 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  $h_c$ , 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 the small  $\tilde{h}(\cdot)$ .

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 features, (iv) imbalanced variances  $\Sigma_0 \neq \Sigma_1$ , and (v) larger **A-X dependence strength**  $|\tau|$ 's.

## 5. Feature Shuffle in Real-World Graphs

In this section, we finalize our investigation of the second research question (**RQ2**) with **feature shuffle**. The feature shuffle can reduce A-X dependence without perturbing X-Y and A-Y dependence, providing a suitable experimental setting to answer **RQ2**. Thus, the feature shuffle serves to generate synthetic versions of the benchmark graphs.

**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  $\in \{0.00, 0.01, \dots, 1.00\}$ . 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. Thereby, the feature shuffle perturbs A-X dependence, reducing **CFH**  $|\tilde{h}(\cdot)|$  (**Observation 3**). Recall that **class homophily**  $h_c$  and **feature distance**  $FD$  remain the same after the feature shuffle.

For each shuffled graph, we initialize, train, and evaluate GNNs. We report mean test accuracy over 5 trials, with a train/val/test split of 50/25/25. For the GNN model, we use GCN, 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) use of non-adaptive convolution and (ii) empirical strengths in various levels of **class homophily**  $h_c$ . For more training details, refer to Appendix F.

### 5.2. Connecting Theory and Real-World Graphs

**High class-homophily graphs.** As shown in Figure 6, in all 12 high **class-homophily**  $h_c$  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**  $h_c$  benchmark datasets, GCNII shows small to no performance improvement in 11/12 datasets (the mean increase of 0.5%p; Figure 7). As demonstrated in CSBM-X experiment (Figure 5), low  $h_c$  reduces the beneficial effect of small A-X dependence in real-world graphs. Unexpectedly, in one of the datasets (Roman-Empire), GCNII shows a steady performance decline. 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**  $FD$  between node classes. Figure 8 shows that significantly increasing feature noise reduces the beneficial effect of the feature shuffle. The finding echoes the results from the CSBM-X (Figure 5), such that  $FD$  moderates the beneficial effect of low A-X dependence. For details about feature noise, refer to Appendix F. The results for all 12 high **class-homophily**  $h_c$  datasets are in Appendix E.

**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 adaptive graph convolution. For an attention-based GNN, we use AERO-GNN, capable of stacking deep layers. In all the considered models, we generally find similar trends as those of GCNII (Figure 9). The results with GCN are also available in Appendix E.

**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). 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 (Figure 10). Our results highlight that reducing A-X dependence can improve GNN regardless of the feature types. The results for all 12 high **class-homophily**  $h_c$  datasets are in Appendix E.

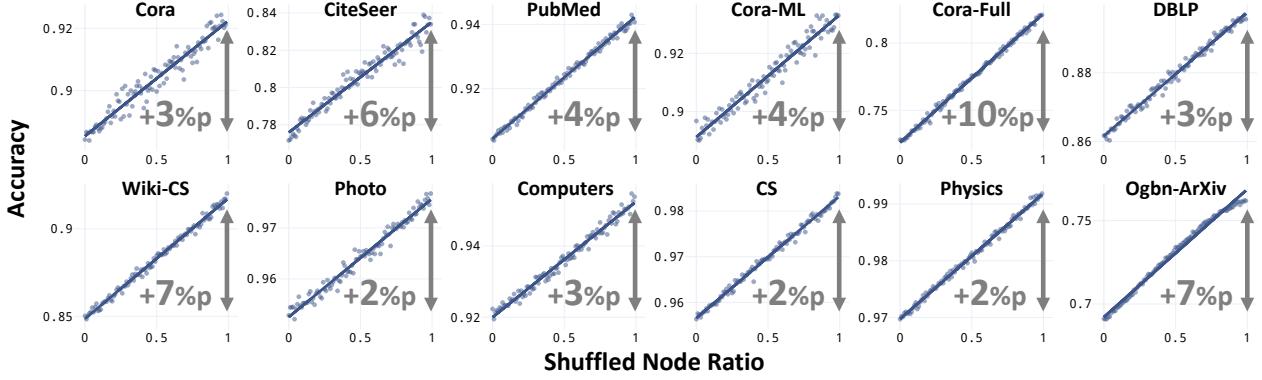


Figure 6: **GCNII Perf. After the Feature Shuffles: High Class Homophily.** Consistent with our findings from CSBM-X (Theorem 4.3; Fig. 5; high  $h_c$  case), reducing A-X dependence with the feature shuffle improves GNN performance consistently and significantly in high class homophily graphs.

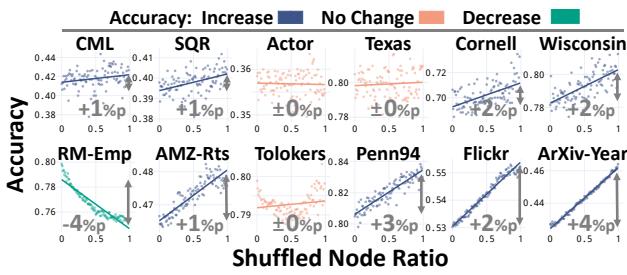


Figure 7: **GCNII Perf. After the Feature Shuffles: Low Class Homophily.** Consistent with our findings from CSBM-X (Theorem 4.3; Fig. 5; low  $h_c$  case), the effect of the feature shuffle is smaller in low class homophily graphs.<sup>6</sup>

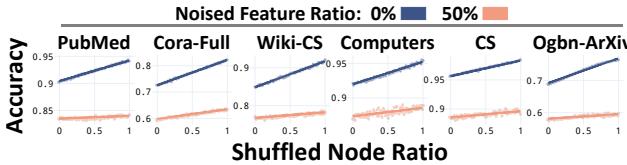


Figure 8: **GCNII Perf. After the Feature Shuffles: Noisy Features.** Consistent with the findings from CSBM-X (Theorem 4.3; Fig. 5; low FD case), the effect of the feature shuffle is smaller with noisy features.

## 6. Discussion

In this work, we present a theory on how A-X dependence (i.e. dependence between graph-topology and node features) affects GNNs. With (i) class-controlled feature homophily (CFH) measure  $\bar{h}(\cdot)$ , (ii) a graph model CSBM-X and theoretical analysis, and (iii) the feature shuffle on real-world graphs, we comprehensively analyze the impact of A-X dependence on GNNs. We conclude that A-X dependence mediates the beneficial effect of graph convolution. In Appendix G, we provide an in-depth discussion.

<sup>6</sup>CML, SQR, RM-Emp, and AMZ-Rts respectively stand for Chameleon, Squirrel, Roman-Empire, and Amazon-Ratings.

<sup>7</sup>Out-of-memory occurs when using adjacency matrix as the node features for Ogbn-ArXiv.

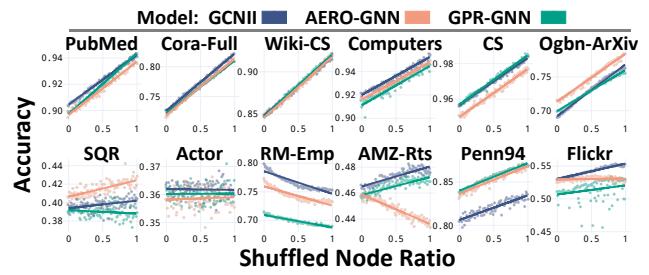


Figure 9: **Other GNNs Perf. After the Feature Shuffles.** The adaptive convolution- and attention-based GNNs (GPR-GNN and AERO-GNN, respectively) generally show similar trends with GCNII.

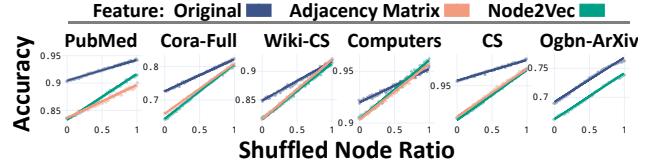


Figure 10: **GCNII Perf. After the Feature Shuffles: Proximity-Based Features.** Regardless of the feature types, the feature shuffle improves GNN. The perf. with proximity-based features degrades potentially due to their high CFH.<sup>7</sup>

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 A-X dependence on GNNs is a promising research direction.

**Limitations and future works.** Generalization of our findings is limited since CSBM-X assumes a monotonic relationship between (class-controlled) feature distance and graph topology, which is also uniform across all nodes. However, the patterns in the real-world graphs are likely to be more complex. Exploring how more realistic patterns interact with GNNs would be a valuable next step.

440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
**Potential Broader Impact**

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.

447  
448  
449  
450  
451  
452  
**References**

- 453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
8010  
8011  
8012  
8013  
8014  
8015  
8016  
8017  
8018  
8019  
8020  
8021  
8022  
8023  
8024  
8025  
8026  
8027  
8028  
8029  
8030  
8031  
8032  
8033  
8034  
8035  
8036  
8037  
8038  
8039  
8040  
8041  
8042  
8043  
8044  
8045  
8046  
8047  
8048  
8049  
8050  
8051  
8052  
8053  
8054  
8055  
8056  
8057  
8058  
8059  
8060  
8061  
8062  
8063  
8064  
8065  
8066  
8067  
8068  
8069  
8070  
8071  
8072  
8073  
8074  
8075  
8076  
8077  
8078  
8079  
8080  
8081  
8082  
8083  
8084  
8085  
8086  
8087  
8088  
8089  
8090  
8091  
8092  
8093  
8094  
8095  
8096  
8097  
8098  
8099  
80100  
80101  
80102  
80103  
80104  
80105  
80106  
80107  
80108  
80109  
80110  
80111  
80112  
80113  
80114  
80115  
80116  
80117  
80118  
80119  
80120  
80121  
80122  
80123  
80124  
80125  
80126  
80127  
80128  
80129  
80130  
80131  
80132  
80133  
80134  
80135  
80136  
80137  
80138  
80139  
80140  
80141  
80142  
80143  
80144  
80145  
80146  
80147  
80148  
80149  
80150  
80151  
80152  
80153  
80154  
80155  
80156  
80157  
80158  
80159  
80160  
80161  
80162  
80163  
80164  
80165  
80166  
80167  
80168  
80169  
80170  
80171  
80172  
80173  
80174  
80175  
80176  
80177  
80178  
80179  
80180  
80181  
80182  
80183  
80184  
80185  
80186  
80187  
80188  
80189  
80190  
80191  
80192  
80193  
80194  
80195  
80196  
80197  
80198  
80199  
80200  
80201  
80202  
80203  
80204  
80205  
80206  
80207  
80208  
80209  
80210  
80211  
80212  
80213  
80214  
80215  
80216  
80217  
80218  
80219  
80220  
80221  
80222  
80223  
80224  
80225  
80226  
80227  
80228  
80229  
80230  
80231  
80232  
80233  
80234  
80235  
80236  
80237  
80238  
80239  
80240  
80241  
80242  
80243  
80244  
80245  
80246  
80247  
80248  
80249  
80250  
80251  
80252  
80253  
80254  
80255  
80256  
80257  
80258  
80259  
80260  
80261  
80262  
80263  
80264  
80265  
80266  
80267  
80268  
80269  
80270  
80271  
80272  
80273  
80274  
80275  
80276  
80277  
80278  
80279  
80280  
80281  
80282  
80283  
80284  
80285  
80286  
80287  
80288  
80289  
80290  
80291  
80292  
80293  
80294  
80295  
80296  
80297  
80298  
80299  
80300  
80301  
80302  
80303  
80304  
80305  
80306  
80307  
80308  
80309  
80310  
80311  
80312  
80313  
80314  
80315  
80316  
80317  
80318  
80319  
80320  
80321  
80322  
80323  
80324  
80325  
80326  
80327  
80328  
80329  
80330  
80331  
80332  
80333  
80334  
80335  
80336  
80337  
80338  
80339  
80340  
80341  
80342  
80343  
80344  
80345  
80346  
80347  
80348  
80349  
80350  
80351  
80352  
80353  
80354  
80355  
80356  
80357  
80358  
80359  
80360  
80361  
80362  
80363  
80364  
80365  
80366  
80367  
80368  
80369  
80370  
80371  
80372  
80373  
80374  
80375  
80376  
80377  
80378  
80379  
80380  
80381  
80382  
80383  
80384  
80385  
80386  
80387  
80388  
80389  
80390  
80391  
80392  
80393  
80394  
80395  
80396  
80397  
80398  
80399  
80400  
80401  
80402  
80403  
80404  
80405  
80406  
80407  
80408  
80409  
80410  
80411  
80412  
80413  
80414  
80415  
80416  
80417  
80418  
80419  
80420  
80421  
80422  
80423  
80424  
80425  
80426  
80427  
80428  
80429  
80430  
80431  
80432  
80433  
80434  
80435  
80436  
80437  
80438  
80439  
80440  
80441  
80442  
80443  
80444  
80445  
80446  
80447  
80448  
80449  
80450  
80451  
80452  
80453  
80454  
80455  
80456  
80457  
80458  
80459  
80460  
80461  
80462  
80463  
80464  
80465  
80466  
80467  
80468  
80469  
80470  
80471  
80472  
80473  
80474  
80475  
80476  
80477  
80478  
80479  
80480  
80481  
80482  
80483  
80484  
80485  
80486  
80487  
80488  
80489  
80490  
80491  
80492  
80493  
80494  
80495  
80496  
80497  
80498  
80499  
80500  
80501  
80502  
80503  
80504  
80505  
80506  
80507  
80508  
80509  
80510  
80511  
80512  
80513  
80514  
80515  
80516  
80517  
80518  
80519  
80520  
80521  
80522  
80523  
80524  
80525  
80526  
80527  
80528  
80529  
80530  
80531  
80532  
80533  
80534  
80535  
80536  
80537  
80538  
80539  
80540  
80541  
80542  
80543  
80544  
80545  
80546  
80547  
80548  
80549  
80550  
80551  
80552  
80553  
80554  
80555  
80556  
80557  
80558  
80559  
80560  
80561  
80562  
80563  
80564  
80565  
80566  
80567  
80568  
80569  
80570  
80571  
80572  
80573  
80574  
80575  
80576  
80577  
80578  
80579  
80580  
80581  
80582  
80583  
80584  
80585  
80586  
80587  
80588  
80589  
80590  
80591  
80592  
80593  
80594  
80595  
80596  
80597  
80598  
80599  
80600  
80601  
80602  
80603  
80604  
80605  
80606  
80607  
80608  
80609  
80610  
80611  
80612  
80613  
80614  
80615  
80616  
80617  
80618  
80619  
80620  
80621  
80622  
80623  
80624  
80625  
80626  
80627  
80628  
80629  
80630  
80631  
80632  
80633  
80634  
80635  
80636  
80637  
80638  
80639  
80640  
80641  
80642  
80643  
80644  
80645  
80646  
80647  
80648  
80649  
80650  
80651  
80652  
80653  
80654  
80655  
80656  
80657  
80658  
80659  
80660  
80661  
80662  
80663  
80664  
80665  
80666  
80667  
80668  
80669  
80670  
80671  
80672  
80673  
80674  
80675  
80676  
80677  
80678  
80679  
80680  
80681  
80682  
80683  
80684  
80685  
80686  
80687  
80688  
80689  
80690  
80691  
80692  
80693  
80694  
80695  
80696  
80697  
80698  
80699  
80700  
80701  
80702  
80703  
80704  
80705  
80706  
80707  
80708  
80709  
80710  
80711  
80712  
80713  
80714  
80715  
80716  
80717  
80718  
80719  
80720  
80721  
80722  
80723  
80724  
80725  
80726  
80727  
80728  
80729  
80730  
80731  
80732  
80733  
80734  
80735  
80736  
80737  
80738  
80739  
80740  
80741  
80742  
80743  
80744  
80745  
80746  
80747  
80748  
80749  
80750  
80751  
80752  
80753  
80754  
80755  
80756  
80757  
80758  
80759  
80760  
80761  
80762  
80763  
80764  
80765  
80766  
80767  
80768  
80769  
80770  
80771  
80772  
80773  
80774  
80775  
80776  
80777  
80778  
80779  
80780  
80781  
80782  
80783  
80784  
80785  
80786  
80787  
80788  
80789  
80790  
80791  
80792  
80793  
80794  
80795  
80796  
80797  
80798  
80799  
80800  
80801  
80802  
80803  
80804  
80805  
80806  
80807  
80808  
80809  
80810  
80811  
80812  
80813  
80814  
80815  
80816  
80817  
80818  
80819  
80820  
80821  
80822  
80823  
80824  
80825  
80826  
80827  
80828  
80829  
80830  
80831  
80832  
80833  
80834  
80835  
80836  
80837  
80838  
80839  
80840  
80841  
80842  
80843  
80844  
80845  
80846  
80847  
80848  
80849  
80850  
80851  
80852  
80853  
80854  
80855  
80856  
80857  
80858  
80859  
80860  
80861  
80862  
80863  
80864  
80865  
80866  
80867  
80868  
80869  
80870  
80871  
80872  
80873  
80874  
80875  
80876  
80877  
80878  
80879  
80880  
80881  
80882  
80883  
80884  
80885  
80886  
80887  
80888  
80889  
80890  
80891  
80892  
80893  
80894  
80895  
80896  
80897  
80898  
80899  
80900  
80901  
80902  
80903  
80904  
80905  
80906  
80907  
80908  
80909  
80910  
80911  
80912  
80913  
80914  
80915  
80916  
80917  
80918  
80919  
80920  
80921  
80922  
80923  
80924  
80925  
80926  
80927  
80928  
80929  
80930  
80931  
80932  
80933  
80934  
80935  
80936  
80937  
80938  
80939  
80940  
80941  
80942  
80943  
80944  
80945  
80946  
80947  
80948  
80949  
80950  
80951  
80952  
80953  
80954  
80955  
80956  
80957  
80958  
80959  
80960  
80961  
80962  
80963  
80964  
80965  
80966  
80967  
80968  
80969  
80970  
80971  
80972  
80973  
80974  
80975  
80976  
80977  
80978  
80979  
80980  
80981  
80982  
80983  
80984  
80985  
80986  
80987  
80988  
80989  
80990  
80991  
80992  
80993  
80994  
80995  
80996  
80997  
80998  
80999  
80100  
80101  
80102  
80103  
80104  
80105  
80106  
80107  
80108  
80109  
80110  
80111  
80112  
80113  
80114  
80115  
80116  
80117  
80118  
80119  
80120  
80121  
80122  
80123  
80124  
80125  
80126  
80127  
80128  
80129  
80130  
80131  
80132  
80133  
80134  
80135  
80136  
80137  
80138  
80139  
80140  
80141  
80142  
80143  
80144  
80145  
80146  
80147  
80148  
80149  
80150  
80151  
80152  
80153  
80154  
80155  
80156  
80157  
80158  
80159  
80160  
80161  
80162  
80163  
80164  
80165  
80166  
80167  
80168  
80169  
80170  
80171  
80172  
80173  
80174  
80175  
80176  
80177  
80178  
80179  
80180  
80181  
80182  
80183  
80184  
80185  
80186  
80187  
80188  
80189  
80190  
80191  
80192  
80193  
80194  
80195  
80196  
80197  
80198  
80199  
80200  
80201  
80202  
80203  
80204  
80205  
80206  
80207  
80208  
80209  
80210  
80211  
80212  
80213  
80214  
80215  
80216  
80217  
80218  
80219  
80220  
80221  
80222  
80223  
80224  
80225  
80226  
80227  
80228  
80229  
80230  
80231  
80232  
80233  
80234  
80235  
80236  
80237  
80238  
80239  
80240  
80241  
80242  
80243  
80244  
80245  
80246  
80247  
80248  
80249  
80250  
80251  
80252  
80253  
80254  
80255  
80256  
80257  
80258  
80259  
80260  
80261  
80262  
80263  
80264  
80265  
80266  
80267  
80268  
80269  
80270  
80271  
80272  
80273  
80274  
80275  
80276  
80277  
80278  
80279  
80280  
80281  
80282  
80283  
80284  
80285  
80286  
80287  
80288  
80289  
80290  
80291  
80292  
80293  
80294  
80295  
80296  
80297  
80298  
80299  
802100  
802101  
802102  
802103  
802104  
802105  
802106  
802107  
802108  
802109  
802110  
802111  
802112  
802113  
802114  
802115  
802116  
802117  
802118  
802119  
802120  
802121  
802122  
802123  
802124  
802125  
802126  
802127  
802128  
802129  
802130  
802131  
802132  
802133  
802134  
802135  
802136  
802137  
802138  
802139  
802140  
802141  
802142  
802143  
802144  
802145  
802146  
802147  
802148  
802149  
802150  
802151  
802152  
802153  
802154  
802155  
802156  
802157  
802158  
802159  
802160  
802161  
802162  
802163  
802164  
802165  
802166  
802167  
802168  
802169  
802170  
802171  
802172  
802173  
802174  
802175  
802176  
802177  
802178  
802179  
802180  
802181  
802182  
802183  
802184  
802185  
802186  
802187  
802188  
802189  
802190  
802191  
802192  
802193  
802194  
802195  
802196  
802197  
802198  
802199  
802200  
802201  
802202  
802203  
802204  
802205  
802206  
802207  
802208  
802209  
802210  
802211

- 495 Luan, S., Hua, C., Xu, M., Lu, Q., Zhu, J., Chang, X.-W.,  
 496 Fu, J., Leskovec, J., and Precup, D. When do graph neural  
 497 networks help with node classification? Investigating the  
 498 impact of homophily principle on node distinguishability.  
 499 In *Advances in Neural Information Processing Systems*,  
 500 2023.
- 501 Ma, Y., Liu, X., Zhao, T., Liu, Y., Tang, J., and Shah, N. A  
 502 unified view on graph neural networks as graph signal  
 503 denoising. In *ACM CIKM International Conference on*  
 504 *Information & Knowledge Management*, 2021.
- 505 Ma, Y., Liu, X., Shah, N., and Tang, J. Is homophily a  
 506 necessity for graph neural networks? In *International*  
 507 *Conference on Learning Representations*, 2022.
- 508 Mao, H., Chen, Z., Jin, W., Han, H., Ma, Y., Zhao, T.,  
 509 Shah, N., and Tang, J. Demystifying structural disparity  
 510 in graph neural networks: Can one size fit all? *arXiv*  
 511 preprint *arXiv:2306.01323*, 2023.
- 512 McPherson, M., Smith-Lovin, L., and Cook, J. M. Birds of  
 513 a feather: Homophily in social networks. *Annual Review*  
 514 of *Sociology*, 27(1):415–444, 2001.
- 515 Mernyei, P. and Cangea, C. Wiki-cs: A wikipedia-based  
 516 benchmark for graph neural networks. *arXiv preprint*  
 517 *arXiv:2007.02901*, 2020.
- 518 NT, H. and Maehara, T. Revisiting graph neural  
 519 networks: All we have is low-pass filters. *arXiv preprint*  
 520 *arXiv:1905.09550*, 2019.
- 521 Oono, K. and Suzuki, T. Graph neural networks exponentially  
 522 lose expressive power for node classification. In *International*  
 523 *Conference on Learning Representations*, 2020.
- 524 Palowitch, J., Tsitsulin, A., Mayer, B., and Perozzi, B.  
 525 Graphworld: Fake graphs bring real insights for gnns.  
 526 In *ACM SIGKDD International Conference on*  
 527 *Knowledge Discovery & Data Mining*, 2022.
- 528 Pei, H., Wei, B., Chang, K. C.-C., Lei, Y., and Yang, B.  
 529 Geom-gcn: Geometric graph convolutional networks. In *International*  
 530 *Conference on Learning Representations*, 2020.
- 531 Platonov, O., Kuznedelev, D., Babenko, A., and  
 532 Prokhorenkova, L. Characterizing graph datasets for node  
 533 classification: Homophily-heterophily dichotomy and be-  
 534 yond. In *Advances in Neural Information Processing Systems*, 2023a.
- 535 Platonov, O., Kuznedelev, D., Diskin, M., Babenko, A., and  
 536 Prokhorenkova, L. A critical look at the evaluation of  
 537 gnns under heterophily: Are we really making progress?  
 538
- 539 In *International Conference on Learning Representations*,  
 540 2023b.
- 541 Ribeiro, L. F., Saverese, P. H., and Figueiredo, D. R.  
 542 struc2vec: Learning node representations from structural  
 543 identity. In *ACM SIGKDD International Conference on*  
 544 *Knowledge Discovery & Data Mining*, 2017.
- 545 Rogers, E. M., Singhal, A., and Quinlan, M. M. Diffusion of  
 546 innovations. In *An integrated approach to communication*  
 547 *theory and research*, pp. 432–448. Routledge, 2014.
- 548 Rozemberczki, B., Allen, C., and Sarkar, R. Multi-scale at-  
 549 tributed node embedding. *Journal of Complex Networks*,  
 550 9(2):cnab014, 2021.
- 551 Shchur, O., Mumme, M., Bojchevski, A., and Günnemann,  
 552 S. Pitfalls of graph neural network evaluation. *arXiv*  
 553 preprint *arXiv:1811.05868*, 2018.
- 554 Stevens, J. P. *Applied multivariate statistics for the social*  
 555 *sciences*. Routledge, 2012.
- 556 Tang, J., Sun, J., Wang, C., and Yang, Z. Social influence  
 557 analysis in large-scale networks. In *ACM SIGKDD Inter-*  
 558 *national Conference on Knowledge Discovery & Data*  
 559 *Mining*, 2009.
- 560 Wang, J., Guo, Y., Yang, L., and Wang, Y. Understanding  
 561 heterophily for graph neural networks. *arXiv preprint*  
 562 *arXiv:2401.09125*, 2024.
- 563 Wang, X. and Zhang, M. How powerful are spectral graph  
 564 neural networks. In *International Conference on Machine*  
 565 *Learning*, 2022.
- 566 Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., and Yu,  
 567 P. S. Heterogeneous graph attention network. In *The Web*  
 568 *Conference*, 2019.
- 569 Wei, R., Yin, H., Jia, J., Benson, A. R., and Li, P. Under-  
 570 standing non-linearity in graph neural networks from the  
 571 bayesian-inference perspective. In *Advances in Neural*  
 572 *Information Processing Systems*, 2022.
- 573 Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., and Wein-  
 574 berger, K. Simplifying graph convolutional networks. In *International*  
 575 *Conference on Machine Learning*, 2019.
- 576 Wu, X., Chen, Z., Wang, W., and Jadbabaie, A. A non-  
 577 asymptotic analysis of oversmoothing in graph neural  
 578 networks. In *International Conference on Learning Rep-*  
 579 *resentations*, 2023.
- 580 Yadati, N., Nimishakavi, M., Yadav, P., Nitin, V., Louis, A.,  
 581 and Talukdar, P. Hypergcn: A new method for training  
 582 graph convolutional networks on hypergraphs. In *Ad-*  
 583 *vances in Neural Information Processing Systems*, 2019.

- 550 Yan, Y., Hashemi, M., Swersky, K., Yang, Y., and Koutra,  
551 D. Two sides of the same coin: Heterophily and over-  
552 smoothing in graph convolutional neural networks. In  
553 *IEEE International Conference on Data Mining*, 2022.
- 554 Yang, Z., Cohen, W., and Salakhudinov, R. Revisiting  
555 semi-supervised learning with graph embeddings. In  
556 *International Conference on Machine Learning*, 2016.
- 557 Yin, H., Benson, A. R., Leskovec, J., and Gleich, D. F.  
558 Local higher-order graph clustering. In *ACM SIGKDD  
559 International Conference on Knowledge Discovery &  
560 Data Mining*, 2017.
- 561 You, J., Gomes-Selman, J. M., Ying, R., and Leskovec, J.  
562 Identity-aware graph neural networks. In *AAAI Conference  
563 on Artificial Intelligence*, 2021.
- 564 Zeng, H., Zhou, H., Srivastava, A., Kannan, R., and  
565
- 566
- 567
- 568
- 569
- 570
- 571
- 572
- 573
- 574
- 575
- 576
- 577
- 578
- 579
- 580
- 581
- 582
- 583
- 584
- 585
- 586
- 587
- 588
- 589
- 590
- 591
- 592
- 593
- 594
- 595
- 596
- 597
- 598
- 599
- 600
- 601
- 602
- 603
- 604
- Prasanna, V. Graphsaint: Graph sampling based inductive  
learning method. In *International Conference on  
Machine Learning*, 2020.
- Zhang, B., Luo, S., Wang, L., and He, D. Rethinking the  
expressive power of gnns via graph biconnectivity. In  
*International Conference on Learning Representations*,  
2023.
- 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 *Advances in  
Neural Information Processing Systems*, 2020.
- Zhu, J., Rossi, R. A., Rao, A., Mai, T., Lipka, N., Ahmed,  
N. K., and Koutra, D. Graph neural networks with het-  
erophily. In *AAAI Conference on Artificial Intelligence*,  
2021.

## A. Proofs and Additional Theoretical Results

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

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  $\tilde{\mathbf{h}}_i^{(v)}$ .  $\square$

*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  $\tilde{\mathbf{h}}^{(G)}$ .  $\square$

*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  $v_\ell$ ). 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  $\mathbf{d}(v_i, N_i) \rightarrow \infty$  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:

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

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.  $\square$

#### 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  $c \in \mathbb{R} \setminus \{0\}$ , the following holds:

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

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  $v_k \in V \setminus N_i$  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 ( $\tau$ controls CFH $\tilde{\mathbf{h}}(\cdot)$ 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  $\tau$ . 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  $v_i$ ,  $\mathbb{E}[\mathbf{h}_i^{(v)} | \tau_1] < \mathbb{E}[\mathbf{h}_i^{(v)} | \tau_2]$ .

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

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

decreases as  $\tau$  increases. Indeed, as  $\tau$  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.,  $\phi_{ij}$ '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  $\tilde{\mathbf{h}}(\cdot)$  is a strictly increasing function of  $\tau$ , which also means that  $\tilde{\mathbf{h}}(\cdot)$  is an injective function of  $\tau$ . Hence, it suffices to show that if  $\tau = 0$ , then  $\mathbb{E}[\tilde{\mathbf{h}}(\cdot)] = 0$ .

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:

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

Recall that  $C_\ell^+$  denotes the node set of class  $\ell$ , whereas  $C_\ell^-$  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  $C_{Y_i}^+ \setminus \{v_i\}$  uniformly at random. Likewise, the different-class neighbor set  $N_i^-$  is chosen from  $C_{Y_i}^-$  uniformly at random. Thus,

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

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  $\tau = 0$ :

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

$\mathbf{h}_i^{(v)}$  is the numerator when calculating  $\tilde{\mathbf{h}}_i^{(v)}$ , and  $\mathbf{h}^{(G)}$  is the numerator when calculating  $\tilde{\mathbf{h}}^{(G)}$ . Thus, both  $\mathbb{E}[\tilde{\mathbf{h}}_i^{(v)}]$  and  $\mathbb{E}[\tilde{\mathbf{h}}^{(G)}]$  become 0s, completing the proof.  $\square$

### Proof of Lemma 4.2 ( $\tau$ controls CFH $\tilde{\mathbf{h}}(\cdot)$ only).

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

### 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)  $n \rightarrow \infty$  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  $\tau$ , 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  $\tau_0$  and  $\tau_1$  such that  $|\tau_0| < |\tau_1|$  and  $\tau_0 \tau_1 > 0$ .*

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

772 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  
 773 each node  $v_i$  here, but the reasoning can be extended to high-dimensional features in general.  
 774

775 WLOG, we assume  $\Sigma_0 = \Sigma_1 = 1$ , which can be ensured by feature normalization. As  $n \rightarrow \infty$ , the sample mean and variance  
 776 of the node features in each class approach  $\pm\mu$  and  $\Sigma$ . For a node  $v_i$ , WLOG (due to the symmetry), we assume  $Y_i = 1$ , and  
 777 let its feature be  $\mu + x_i$  (i.e., its class-controlled feature is  $x_i$ ). Let  $\varphi$  be the PDF of standard normal distribution  $\mathcal{N}(0, 1)$ ,  
 778 then the homophily baseline  $b(v_i)$  of node  $v_i$  is  
 779

$$780 \quad b(v_i) = \int_{-\infty}^{\infty} \varphi(x) |x - x_i| dx = \frac{1}{2} e^{-\frac{x_i^2}{2}} \left( e^{\frac{x_i^2}{2}} x_i \operatorname{erf}\left(\frac{x_i}{\sqrt{2}}\right) - e^{\frac{x_i^2}{2}} x_i \operatorname{erfc}\left(\frac{x_i}{\sqrt{2}}\right) + e^{\frac{x_i^2}{2}} x_i + 2\sqrt{\frac{2}{\pi}} \right) = \exp(-\frac{x_i^2}{2}) + x_i \operatorname{erf}\left(\frac{x_i}{\sqrt{2}}\right),$$

783 where ‘‘erf’’ is the Gauss error function defined as  
 784

$$785 \quad \operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$$

786 and ‘‘erfc’’ is the complementary error function defined as  
 787

$$788 \quad \operatorname{erfc}(z) = 1 - \operatorname{erf}(z).$$

789 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  
 790 it has feature  $\mu + x_j$  if  $Y_j = 1$ ) is  
 791

$$792 \quad h_{ij}^{(p)} = b(v_i) - |x_i - x_j| = \exp(-\frac{x_i^2}{2}) \sqrt{\frac{2}{\pi}} + x_i \operatorname{erf}\left(\frac{x_i}{\sqrt{2}}\right) - |x_i - x_j|,$$

793 which gives  
 794

$$795 \quad \phi_{ij} = \exp(\tau h_{ij}^{(p)}) = \exp\left(\tau\left(-|x_i - x_j| + x_i \operatorname{erf}\left(\frac{x_i}{\sqrt{2}}\right) + \sqrt{\frac{2}{\pi}} \exp(-\frac{x_i^2}{2})\right)\right)$$

800 Since  $n \rightarrow \infty$ , weighted sampling without replacement approaches weighted sampling with replacement approaches, and the  
 801 probability of  $v_j$  being sampled as one of  $v_i$ ’s neighbors is  
 802

$$803 \quad \Pr[v_j \in N_i] = \frac{\phi_{ij}}{\int_{-\infty}^{\infty} \phi_{ij'} \varphi(x_{j'}) dx_{j'}} = \frac{2 \exp(-\frac{1}{2}\tau(2|x_i - x_j| + \tau - 2x_i))}{\operatorname{erfc}\left(\frac{\tau - x_i}{\sqrt{2}}\right) + \exp(2\tau x_i) \operatorname{erfc}\left(\frac{\tau + x_i}{\sqrt{2}}\right)},$$

806 and in each sampling step, the sampled neighbor has a class-controlled feature equal to  $x_j$  is  
 807

$$808 \quad \Pr[v_j \in N_i] \varphi(x_j).$$

809 In other words, let  $x_{nbr}$  denote the random variable of the class-controlled feature of a sampled neighbor, we have  
 810

$$811 \quad \Pr[x_{nbr} = x^*] = \frac{2 \exp(-\frac{1}{2}\tau(2|x_i - x^*| + \tau - 2x_i))}{\operatorname{erfc}\left(\frac{\tau - x_i}{\sqrt{2}}\right) + \exp(2\tau x_i) \operatorname{erfc}\left(\frac{\tau + x_i}{\sqrt{2}}\right)} \varphi(x^*) = \frac{\sqrt{\frac{2}{\pi}} \exp(-\frac{1}{2}\tau(2|x_i - x^*| + \tau - 2x_i) - \frac{x_i^2}{2})}{\operatorname{erfc}\left(\frac{\tau - x_i}{\sqrt{2}}\right) + \exp(2\tau x_i) \operatorname{erfc}\left(\frac{\tau + x_i}{\sqrt{2}}\right)}, \forall x^* \in \mathbb{R}.$$

815 We can compute the closed-form expectation of  $x_{nbr}$ , which is  
 816

$$817 \quad \mathbb{E}[x_{nbr}] = \int_{-\infty}^{\infty} x^* \Pr[x_{nbr} = x^*] dx^* = \frac{\tau \left( \operatorname{erfc}\left(\frac{\tau - x_i}{\sqrt{2}}\right) - \exp(2\tau x_i) \operatorname{erfc}\left(\frac{\tau + x_i}{\sqrt{2}}\right) \right)}{\operatorname{erfc}\left(\frac{\tau - x_i}{\sqrt{2}}\right) + \exp(2\tau x_i) \operatorname{erfc}\left(\frac{\tau + x_i}{\sqrt{2}}\right)},$$

820 and its variance  $\operatorname{Var}[x_{nbr}]$  does not depend on the value of  $n$ . After one step of graph convolution, the new node feature of  
 821  $v_i$  would be  
 822

$$823 \quad \hat{x}_i = \frac{d^+ \mu - d^- \mu + \sum_{t=1}^{d^+ + d^-} x_t}{d^+ + d^-},$$

where each  $x_t$  i.i.d. follows  $x_{nbr}$ . By the central limit theorem, as  $n \rightarrow \infty$  and, thus,  $d^+$  and  $d^-$  approaches infinity,  $\hat{x}_i$  asymptotically follows  $\mathcal{N}(\frac{d^+\mu-d^-\mu}{d^++d^-} + \mathbb{E}[x_{nbr}], \text{Var}[x_{nbr}]/\sqrt{d^++d^-})$ . 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  $\text{Var}[x_{nbr}]/\sqrt{d^++d^-}$  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  $\mathbf{1}[\frac{d^+\mu-d^-\mu}{d^++d^-} + \mathbb{E}[x_{nbr}] < 0] \rightarrow \mathbf{1}[\frac{d^+\mu-d^-\mu}{d^++d^-} < 0] = 0$ .

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

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

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  $\tau$ ) 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,

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

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  $x_i > 0$ ,  $\mathbb{E}[x_{nbr}]$  decreases as  $\tau < 0$  decreases (i.e.,  $\tau$  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  $\mathbb{E}[x_{nbr}] = \mathbf{0}$  if and only if  $\tau = 0$ . Moreover,  $x_{nbr}$  moves further from  $\mathbf{0}$  as  $|\tau|$  increases, which increases the BER. Specifically, in the above reasoning, one needs to replace  $x > 0$  with  $\tilde{\mu}^\top x > 0$  with  $\tilde{\mu} = \frac{d^+\mu-d^-\mu}{d^+-d^-} = \frac{p^+\mu-p^-\mu}{p^+-p^-}$  (features that would be classified as the positive class, class-1), and similarly replace  $x < 0$  with  $\tilde{\mu}^\top x < 0$ .  $\square$

## B. In-Depth Analysis of Measure

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

In this subsection, we discuss the details of  $\tilde{h}(\cdot)$  interpretation. For high-level ideas, refer to Sec. 3.2.

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

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

For positive (or negative)  $\tilde{h}_i^{(v)}$ , the node  $v_i$  has  $\frac{|\tilde{h}_i^{(v)}|}{1-|\tilde{h}_i^{(v)}|}$  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  $\tilde{h}_i^{(v)} = 0.9$ , 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  $\tilde{h}^{(G)}$ :

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

Like in node-level interpretation, for positive (or negative)  $\tilde{h}^{(G)}$ , the graph  $G$  has  $\frac{|\tilde{h}^{(G)}|}{1-|\tilde{h}^{(G)}|}$  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  $\tilde{h}^{(G)} = 0.9$ , 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 features are identical to all other nodes (i.e.  $X_i = X_j, \forall v_j \in V$ ),  $\tilde{h}_i^{(v)} = 0$ , because its  $b(v_i) = \mathbf{d}(v_i, N_i) = 0$ .

<sup>8</sup> A fully connected node  $v_i$  has  $\tilde{h}_i^{(v)} = 0$ , because its  $b(v_i) = \mathbf{d}(v_i, N_i)$ . For the same reason, a graph  $G$  has  $\tilde{h}^{(G)} = 0$  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 probability  $p_e$ ,  $\mathbb{E}[\tilde{h}_i^{(v)}] = 0$ . For the same reason, a graph  $G$  has  $\mathbb{E}[\tilde{h}^{(G)}] = 0$  if each node  $v_i \in V$  chooses a non-zero number of neighbors by a probability  $p_e$ .

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

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

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

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

<sup>8</sup>Recall that we define  $\tilde{h}_i^{(v)}$  and  $\tilde{h}^{(G)}$  to be 0, if  $b(\cdot) = 0$ .

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

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

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

941 As we take a closer look at the form of  $\beta^*$  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.,  $(Y^T X)_i = \sum_{v_k \in C_i^+} X_k$ ,  $\forall i \in [c]$ ).

942 Thus,  $\beta^* \in \mathbb{R}^c$  is a vector where  $k$ -th entry indicates the mean of node features that belong to the class  $i$ . In the given setting,  
 943 by applying obtained  $\beta^*$ , 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  
 944 conclude that Eq. (3) is a special case of the variable control method of part correlation.

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

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

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

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

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

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

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

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

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

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

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

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

953 **G4) Generalized node-level normalization:**

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

954 **G5) Generalized graph-level normalization:**<sup>9</sup>

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

955 CFH measure  $\tilde{h}(\cdot)$  is a special case of the proposed generalized homophily measure  $\tilde{H}(\cdot)$ . With the generalized homophily  
 956 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  
 957 matrix.

958 <sup>9</sup>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 24 node classification benchmark datasets. Specifically, we buttress Obs. 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  $\tilde{h}(\cdot)$  in the benchmark graphs. Specifically, we further support Obs. 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  $\tilde{h}(\cdot)$  scores are for the benchmark graphs, compared to different features. Recall that their mean  $|\tilde{h}^{(G)}| = 0.06$ . For comparison, we consider two other node features.

**X1) Random Baseline:** Random node features  $X^{(rand)}$

$$X_i^{(rand)} \sim \mathcal{N}(0, 1), \forall v_i \in V. \quad (26)$$

**X2) Homophilic Baseline:** Convolved node features  $X^{(conv)}$

$$X^{(conv)}(l) = ((D + I)^{-1}(A + I))^l X, \quad (27)$$

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

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

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

**Conclusion.** From the series of analyses, we conclude again that CFH  $\tilde{h}(\cdot)$  scores are generally positive and low in the benchmark graphs.

### 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 Obs. 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.

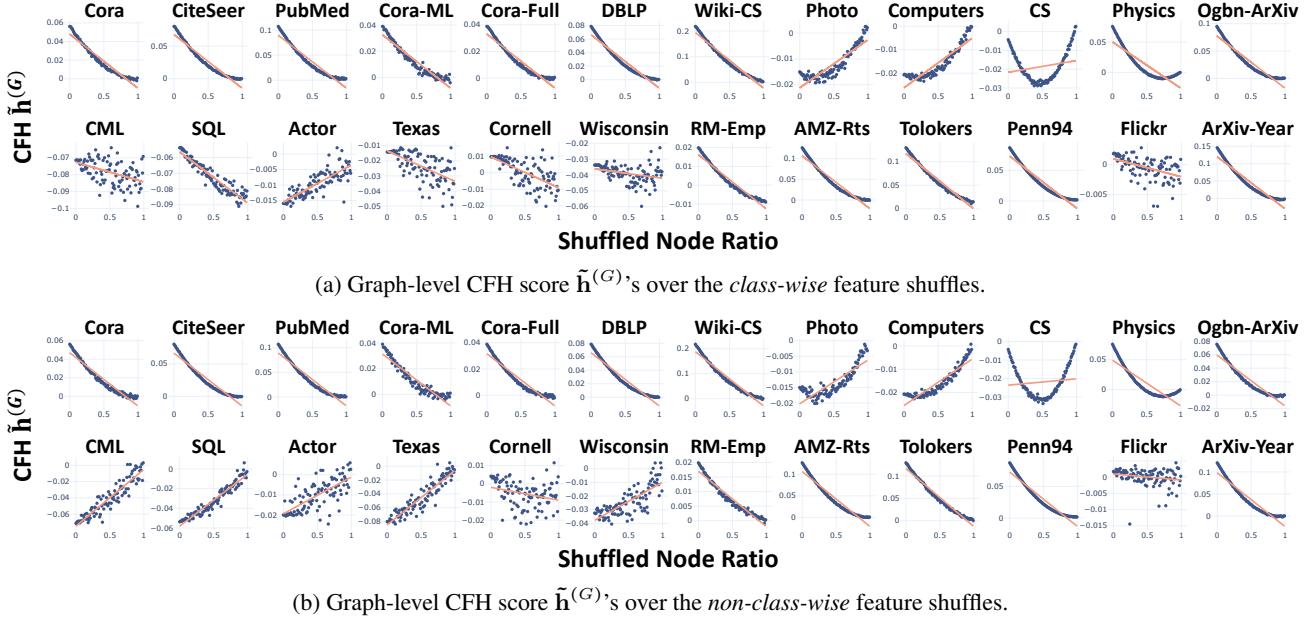


Figure 11: Graph-Level CFH Statistics in Real-World Graphs and their Relationship to the Feature Shuffle.

(2020) and CFH have correlation coefficients of 0.403 (Pearson's  $r$ ) and 0.196 (Kendall's  $\tau$ ). Class homophily defined by Pei et al. (2020) and CFH have correlation coefficients of 0.401 (Pearson's  $r$ ) and 0.225 (Kendall's  $\tau$ ). 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  $\tau$ . We, thus, do not find counter-evidence for Obs. 2.

**Node-level analysis.** Now, we complement Obs. 2 with node-level analysis. Specifically, we analyze the correlation between node-level CFH  $\tilde{h}_i^{(v)}$  and node-level class homophily  $\tilde{H}_i^{(v)}(Y)$  (Eq. (24)).<sup>10</sup> 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  $\tilde{h}(\cdot)$  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 Obs. 3, few of them (Chameleon, Squirrel, Texas, Wisconsin, and Cornell) do not fully obey it. Specifically, their graph-level CFH  $\tilde{h}^{(G)}$  score moves away from 0 over increasing shuffled node ratio. We reason their deviation by answering two questions: (i) Why do the  $\tilde{h}^{(G)}$  scores become larger after the feature shuffle?; (ii) Why do the  $\tilde{h}^{(G)}$  scores not approach 0 after the feature shuffle?

**Answer to question (i).** Node-level CFH  $\tilde{h}_i^{(v)}$  distributions before and after the feature shuffle (Fig. 12) reveals that the mean  $|\tilde{h}_i^{(v)}|$  decreases after the feature shuffle in all five datasets. The finding indicates that the magnitude in which the distance to neighbors (i.e.  $d(v_i, N_i)$ ) 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  $\tilde{h}(\cdot)$  is necessary to reveal the pattern. The graph-level CFH  $\tilde{h}^{(G)}$  does not fully capture the subtlety as it mean-aggregates the positive and negative  $\tilde{h}_i^{(v)}$  scores.

**Answer to question (ii).** The  $\tilde{h}^{(G)}$  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  $\tilde{h}^{(G)}$ 's approach 0 in 23/24 datasets (Fig. 11(b)). The finding suggests that feature distribution difference between node classes hinders CFH  $\tilde{h}^{(G)}$  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

<sup>10</sup>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.

Feature Distribution on Graph Topology Mediates the Effect of Graph Convolution

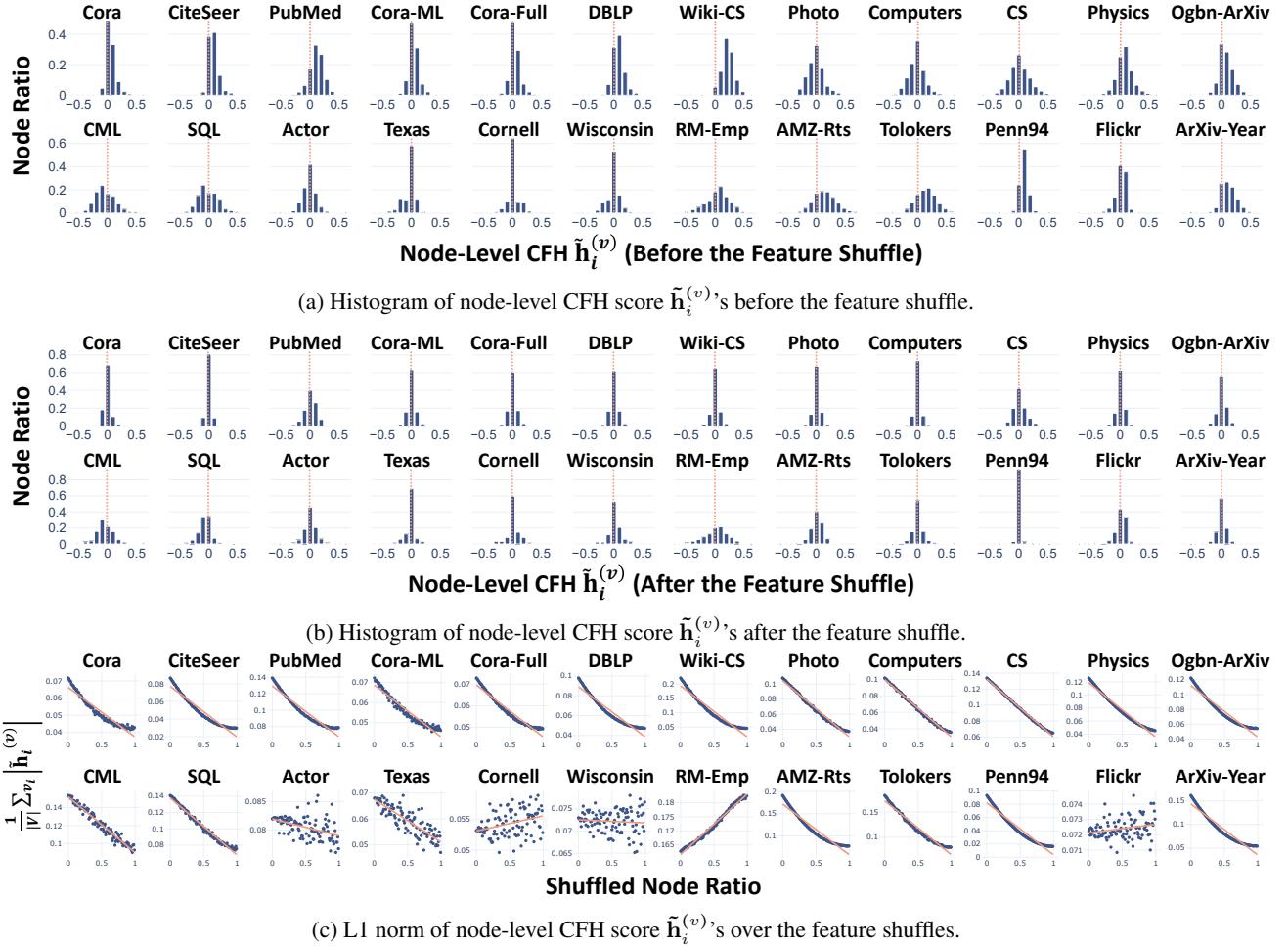


Figure 12: Node-Level CFH Statistics in Real-World Graphs and their Relationship to the Feature Shuffle.

feature shuffle effectively perturbs A-X dependence,  $\tilde{h}^{(G)}$  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 Obs. 3 do not undermine the integrity of our conclusion that A-X dependence mediates the effect of graph convolution.

#### 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  $\tilde{h}(\cdot)$  over the feature shuffle. In Fig. 12(a,b), we *uniquely* observe that its histograms of  $\tilde{h}_i^{(v)}$  before and after the feature shuffle are highly similar. In Fig. 12(c), we further *uniquely* observe that its mean  $|\tilde{h}_i^{(v)}|$  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 33 benchmark datasets, which are partly borrowed from Lee et al. (2023). The dataset statistics are provided 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 node features are the text features of the corresponding article, and the node class is the category of the research/subject domain of the document.

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 $h_c$	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 $\tilde{h}^{(G)}$	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( $\tilde{h}_i^{(v)}, \tilde{H}_i^{(v)}(Y)$ )	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 $h_c$	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 $\tilde{h}^{(G)}$	-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( $\tilde{h}_i^{(v)}, \tilde{H}_i^{(v)}(Y)$ )	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.

(\*\*) RM-Emp stands for Roman-Empire, and AMZ-Rts stands for Amazon-Ratings.

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

- The *Wiki-CS* dataset is a webpage network of Wikipedia (Mernyei & Cangea, 2020). Each node represents a Wikipedia webpage related to computer science, and two nodes are adjacent if a hyperlink exists between the two corresponding webpages. For each node, the node features are the GloVe word embeddings of the corresponding 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 product, and two nodes are adjacent if the two corresponding products are frequently purchased together. For each node, the node features are the bag-of-words features of the customer reviews of the corresponding product, and the node class is the category of the product.
- 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 node 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 node features are informative nouns on the corresponding webpage, and the node label 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 corresponding actors appear on the same Wikipedia webpage. For each node, the node features are derived from the keywords on the Wikipedia webpage of the corresponding actor, and the node label 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 corresponding webpages. For each node, the node 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 node 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 node 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 node features consist of the worker profile and task performance statistics, and the node class is whether or not the worker has been banned.
- The *Penn94* dataset is an online social network on Facebook (Lim et al., 2021). Each node represents a student user, and two nodes are adjacent if they are friends. For each node, the node features are the user profile, and the node class is the reported gender.
- The *Flickr* dataset is a network of uploaded 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, comments by the same user, etc). For each node, the node 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.
- The *Twitch* dataset (Rozemberczki et al., 2021) is a network of Twitch users. Each node represents an Twitch user, and two nodes are adjacent if the two have mutual friendships. For each node, the node features are games liked, location, and streaming habits. The node class indicates whether the user uses explicit language. We use DE version of the Twitch dataset.
- The *GitHub* dataset (Rozemberczki et al., 2021) is a network of GitHub users. Each node represents a GitHub user, and two nodes are adjacent if the two have mutual follower relationships. For each node, the node features are locations, starred repositories, employer, and email address. The node class is the user profession, either web or machine learning developer.
- The *USA* and *Europe* datasets (Ribeiro et al., 2017) is a network of airports. Each node represents an airport, and two nodes are adjacent if the two airports are connected by commercial flights. For each node, its class is airport activity level. The node features are not provided.
- The *Email* dataset (Yin et al., 2017) is a network of emails. Each node represents an members of an research institution, and two nodes are adjacent if the two exchanged email at least once. For each node, its class is the membership to one of 42 departments. The node features are not provided.
- The *Cora-CA* and *DBLP-P* datasets (Yadati et al., 2019) is a network of academic papers. Each node represents a publication, and the nodes are connected by a hyperedge if they are written by the same author. For each node, the node features are bag-of-words of the corresponding publication, and the class is the academic category of the publication.
- The *IMDB* dataset (Wang et al., 2019) is a network of movies. Each node represents an movie, and the nodes are connected by a hyperedge if the same actor appears in the movies. For each node, the node features are bag-of-words of the movie plot, and the class is movie genre.

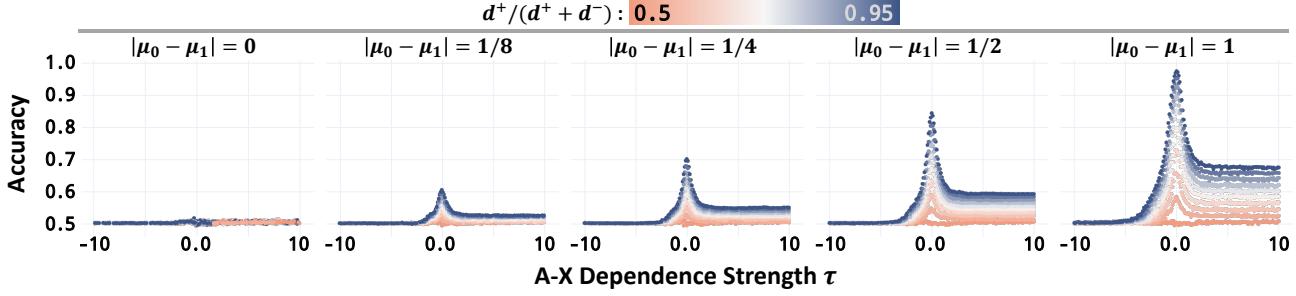


Figure 13: **The Simplified GNN Performance in CSBM-X Graphs: Wider  $\tau$  Range.** With  $\tau \in \{-10, -9.9, \dots, 9.9, 10\}$ , the findings are consistent with those from Fig. 5.

## D. In-Depth Analysis of CSBM-X

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

### D.1. Full Experiments: Large Range of $\tau$

Fig. 13 shows the experimental results with larger range of  $\tau \in \{-10, -9.9, \dots, 9.9, 10\}$ . 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.  $\mu_0 = [c, c, \dots, c] \in \mathbb{R}^k$  and  $\mu_1 = [-c, -c, \dots, -c] \in \mathbb{R}^k$ , 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)  $\Sigma_0 = \Sigma_1 = \text{diag}(1)$ . 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

**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  $\tau$  is larger. The finding possibly relates to the sparse topology of the generated CSBM-X graph  $\mathcal{G}$ 's,<sup>11</sup> 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

**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  $\mu_\ell$  and feature covariance matrix  $\Sigma_\ell$ , each corresponds to the class  $\ell$ , same- and different-class degree  $d^+$  and  $d^-$ , respectively, and A-X dependence strength  $\tau$ . Let  $\mathcal{I} := (n, \mu_0, \mu_1, \Sigma_0, \Sigma_1, d^+, d^-, \tau)$  denotes the set of input parameters.

**Node classes.** Given input  $\mathcal{I} = (n, \mu_0, \mu_1, \Sigma_0, \Sigma_1, d^+, d^-, \tau)$ , the node set is (deterministically)  $V = V(\mathcal{I}) = V(n) = [n]$ ,

<sup>11</sup>Recall that the number of nodes is 10,000, whereas the node degree is 20 for all nodes.

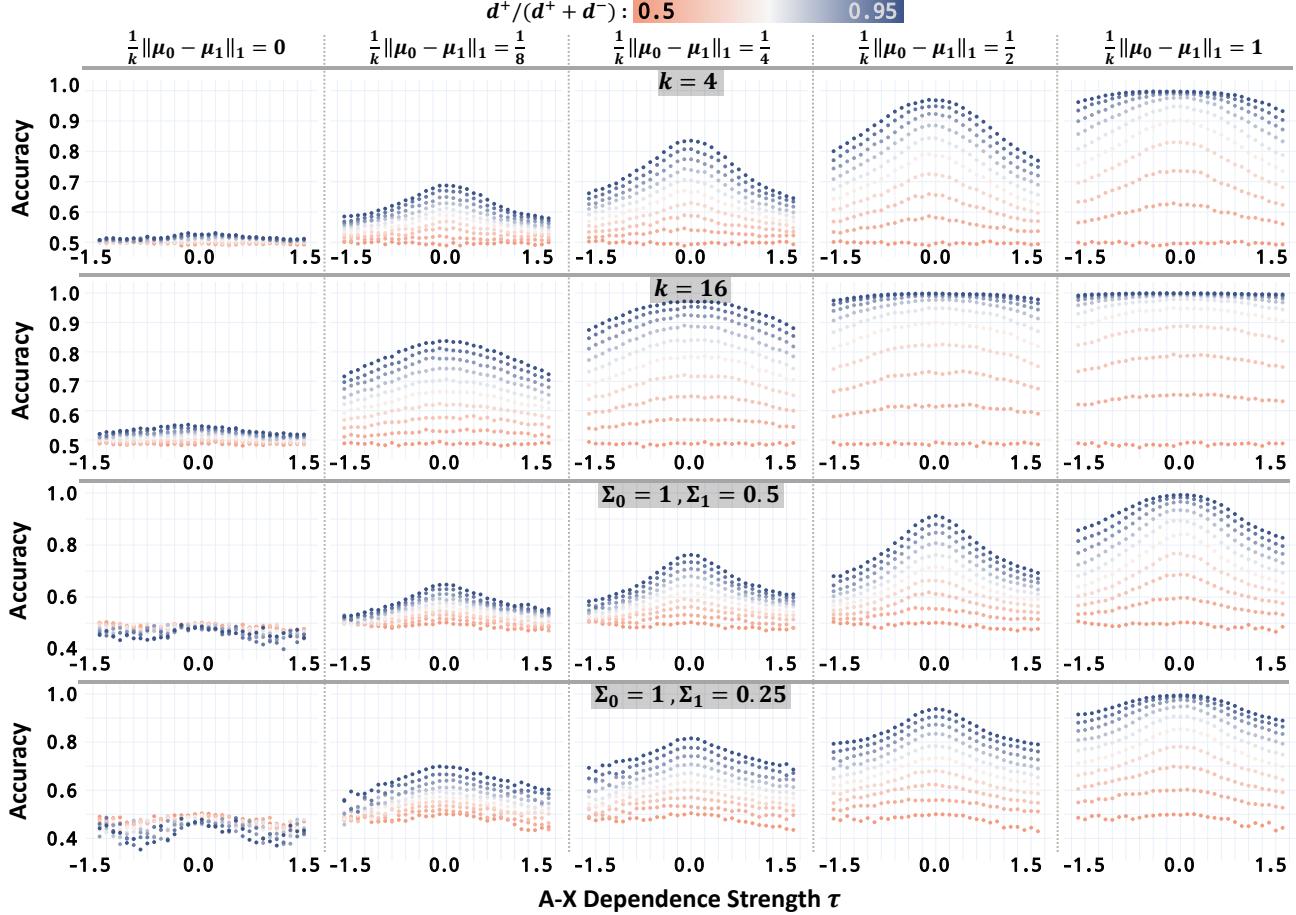


Figure 14: **The Simplified GNN Performance in CSBM-X Graphs: Feature Variations.** With different feature parameter configurations, including (i) high-dimensional features (i.e.  $k > 1$ ) and (ii) imbalanced variances (i.e.  $\Sigma_0 \neq \Sigma_1$ ), the findings are consistent with those from Fig.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

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  $Y \in \mathcal{Y}_n := \{\{0, 1\}^n : \sum_{i \in [n]} Y_i = \frac{n}{2}\}$ , 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 node 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  $\mu_{Y_i}$  and variance  $\Sigma_{Y_i}$ . Specifically,  $\Pr[X_i | Y, \mathcal{I}] = \Pr[X_i | Y_i, \mu_{Y_i}, \Sigma_{Y_i}] = (2\pi)^{-k/2} \det(\Sigma_{Y_i})^{-1/2} \exp\left(-\frac{1}{2}(X_i - \mu_{Y_i})^\top \Sigma_{Y_i}^{-1} (X_i - \mu_{Y_i})\right)$ , which is the PDF of multivariate Gaussian  $\mathcal{N}(\mu_{Y_i}, \Sigma_{Y_i})$ .

**Edges.** Conditioned on node classes and features, edges are sampled by weighted sampling without replacement. For each node  $i \in [n]$ , we define  $\mathbf{C}_i^+ = \mathbf{C}_i^+(Y) = C_{Y_i}^+ \setminus \{i\} = \{\mathbf{C}_{i,1}^+, \mathbf{C}_{i,2}^+, \dots, \mathbf{C}_{i,|\mathbf{C}_i^+|}^+\}$  (fix order of the nodes in  $\mathbf{C}_i^+$ ) and  $\mathbf{C}_i^- = \mathbf{C}_i^-(Y) = C_{Y_i}^- = \{\mathbf{C}_{i,1}^-, \mathbf{C}_{i,2}^-, \dots, \mathbf{C}_{i,|\mathbf{C}_i^-|}^-\}$ , and we also define the sampling weights  $\Phi_i^+ = \Phi_i^+(\mathbf{C}_i^+, X) = (e^{\tau h_{ij}^{(p)}}, j = \mathbf{C}_{i,t}^+; t \in [\mathbf{C}_i^+]) \in \mathbb{R}^{|\mathbf{C}_i^+|}$  and  $\Phi_i^- = \Phi_i^-(\mathbf{C}_i^-, X, \mathcal{I}) = (e^{\tau h_{ij}^{(p)}}, j = \mathbf{C}_{i,t}^-; t \in [\mathbf{C}_i^-]) \in \mathbb{R}^{|\mathbf{C}_i^-|}$ . Note that  $X$  is used here to compute  $h_{ij}^{(p)}$ 's. Let  $\mathbf{C}^+$  denote  $(\mathbf{C}_i^+ : i \in [n])$ , and  $\mathbf{C}^-, \Phi_i^+$ , and  $\Phi_i^-$  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  $\mathcal{B}_i^+ = \mathcal{B}_i^+(\mathbf{C}_i^+, d^+) = \{N_i^+ : N_i^+ \subseteq \mathbf{C}_i^+, |N_i^+| = d^+\}$  and  $\mathcal{B}_i^- = \mathcal{B}_i^-(\mathbf{C}_i^-, d^-) = \{N_i^- : N_i^- \subseteq \mathbf{C}_i^-, |N_i^-| = d^-\}$ . For

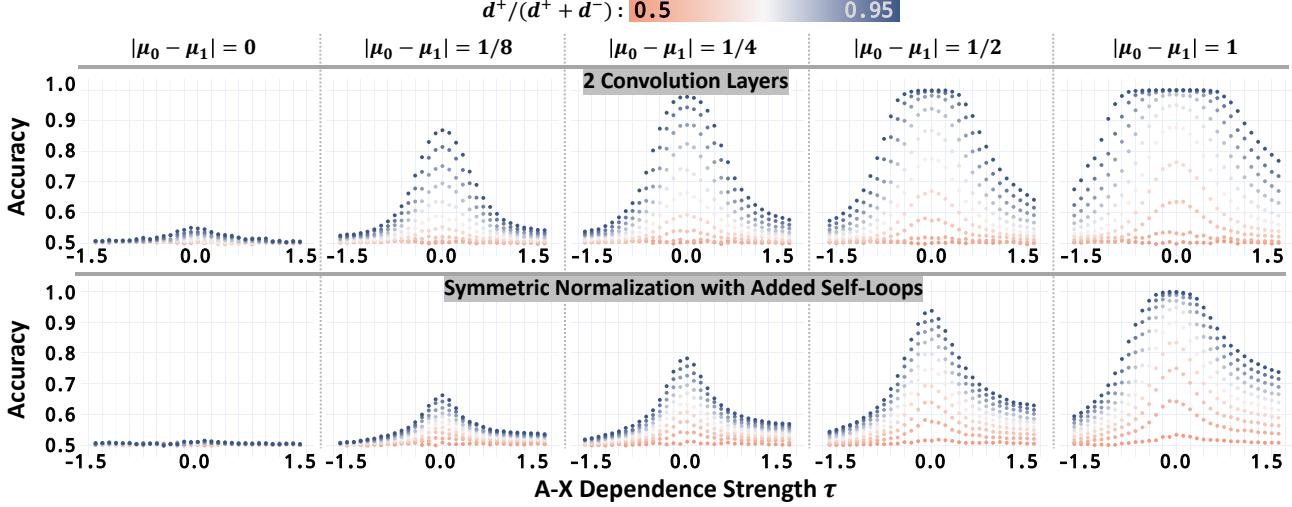


Figure 15: **The Simplified GNN Performance in CSBM-X Graphs: GNN Variations.** With two simplified GNN variations, including graph convolution with (i) two layers and (ii) symmetrically normalized adjacency matrix with self-loops, the findings are consistent with those from Fig. 5.

each possible  $N_i^+ = \{N_{i,1}^+, N_{i,2}^+, \dots, N_{i,d^+}^+\} \in \mathcal{B}_i^+$ ,  $\Pr[N_i^+ | \mathbf{C}_i^+, \Phi_i^+, d^+] = \sum_{\pi \in S_{d^+}} \prod_{t=1}^{d^+} \frac{\Phi_{N_{\pi(i,t)}^+}^+}{1 - \sum_{t'=1}^{t-1} \Phi_{N_{\pi(i,t')}^+}^+}$ , where  $S_{d^+}$  is the set of all permutations on  $[d^+]$ ;  $\Pr[N_i^- | \mathbf{C}_i^-, \Phi_i^-, d^-]$  is defined similarly. For each possible  $N_i = N_i^+ \cup N_i^- \in \mathcal{B}_i$ ,  $\Pr[N_i | \mathbf{C}_i^+, \Phi_i^+, \mathbf{C}^-, \Phi_i^-, d^+, d^-] = \Pr[N_i^+ | \mathbf{C}_i^+, \Phi_i^+, d^+] \Pr[N_i^- | \mathbf{C}_i^-, \Phi_i^-, d^-]$ . The neighbor set of each node is sampled independently, i.e.,  $\Pr[(N_i : i \in [n]) | \mathbf{C}_i^+, \Phi_i^+, \mathbf{C}^-, \Phi_i^-, d^+, d^-] = \prod_{i \in [n]} \Pr[N_i | \mathbf{C}_i^+, \Phi_i^+, \mathbf{C}^-, \Phi_i^-, d^+, d^-]$ . Here,  $N_i, \forall i \in [n]$  fully determine the topology of each generated graph.

### Summary.

$$\bullet \quad \mathcal{I} = (n, \mu_0, \mu_1, \Sigma_0, \Sigma_1, d^+, d^-, \tau)$$

$$\bullet \quad V = V(\mathcal{I}) = [n]$$

$$\bullet \quad \Pr[Y | \mathcal{I}] = \frac{1}{\binom{n}{2}}, \forall Y \in \mathcal{Y}_n \doteq \{\{0, 1\}^n : \sum_{i \in [n]} Y_i = \frac{n}{2}\}$$

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

$$\bullet \quad \Pr[X | Y, \mathcal{I}] = \prod_{i \in [n]} \Pr[X_i | Y_i, \mathcal{I}]$$

$$\bullet \quad \mathbf{C}_i^+(Y) = C_{Y_i}^+ \setminus \{i\} = \{\mathbf{C}_{i,1}^+, \mathbf{C}_{i,2}^+, \dots, \mathbf{C}_{i,|\mathbf{C}_i^+|}^+\}$$

$$\bullet \quad \mathbf{C}_i^-(Y) = C_{Y_i}^- = \{\mathbf{C}_{i,1}^-, \mathbf{C}_{i,2}^-, \dots, \mathbf{C}_{i,|\mathbf{C}_i^-|}^-\}$$

$$\bullet \quad \Phi_i^*(\mathbf{C}_i^*, X, \mathcal{I}) = (e^{\tau \mathbf{h}_{ij}^{(p)}}, j = \mathbf{C}_{i,t}^* : t \in [|C_i^*|]), \forall * \in \{+, -\}$$

$$\bullet \quad \mathcal{B}_i^*(\mathbf{C}_i^*, d^*) = \{N_i^* : N_i^* \subseteq \mathbf{C}_i^*, |N_i^*| = d^*\}, \forall * \in \{+, -\}$$

$$\bullet \quad \mathcal{B}_i(\mathbf{C}_i^+, \mathbf{C}_i^-, d^+, d^-) \doteq \{N_i^+ \cup N_i^- : N_i^+ \in \mathcal{B}_i^+, N_i^- \in \mathcal{B}_i^-\}$$

$$\bullet \quad \Pr[N_i^* | \mathbf{C}^*, \Phi^*, \mathcal{I}] = \sum_{\pi \in S_{d^*}} \prod_{t=1}^{d^*} \frac{\Phi_{N_{\pi(i,t)}^*}^*}{1 - \Phi_{N_{\pi(i,1)}^*}^* - \dots - \Phi_{N_{\pi(i,t-1)}^*}^*}, \forall * \in \{+, -\}$$

$$\bullet \quad \Pr[N_i | \mathbf{C}^+, \Phi^+, \mathbf{C}^-, \Phi^-, \mathcal{I}] = \Pr[N_i^+ | \mathbf{C}^+, \Phi^+, \mathcal{I}] \Pr[N_i^- | \mathbf{C}^-, \Phi^-, \mathcal{I}], \forall N_i = N_i^+ \cup N_i^- \in \mathcal{B}_i$$

$$\bullet \quad \Pr[(N_i : i \in V = [n]) | \mathbf{C}^+, \Phi^+, \mathbf{C}^-, \Phi^-, \mathcal{I}] = \prod_{i \in [n]} \Pr[N_i | \mathbf{C}_i^+, \Phi_i^+, \mathbf{C}_i^-, \Phi_i^-, \mathcal{I}]$$

---

## 1430 E. Additional Experimental Results with Feature Shuffle

1431 In this section, we provide additional experimental results of the feature shuffle with real-world graphs. We use the same  
 1432 setting as in Sec. 5, unless otherwise specified.

1433 **Models.** First, Fig. 16(a) shows GCN performance after the feature shuffle in 6 high  $\mathbf{h}_c$  and 6 low  $\mathbf{h}_c$  datasets. While GCN  
 1434 performance increases over the feature shuffles, GCNII benefits more from the feature shuffle (i.e. larger positive slopes  
 1435 by GCNII). This outcome may relate to the difference in their number of layers. We claim two complementary pieces of  
 1436 evidence. For one, CSBM-X experiment in Fig. 15 suggests that a larger number of layers can further improve the beneficial  
 1437 effect of small  $\tau$ . Also, one of the main differences between the GCN and GCNII is their capability in stacking deeper  
 1438 layers. The relationship between GNN depth and A-X dependence, however, is beyond the scope of the present work, and  
 1439 we leave it up to future studies. Overall, consistent with the conclusion of Sec. 5, we conclude that all GNN models benefit  
 1440 from the feature shuffle.

1441 **Train ratios.** Second, Fig. 16(b) shows GCNII performance over the feature shuffle with different splits. We use three  
 1442 different train/val splits while fixing the test split. Two findings are worth noting. First, model performance increases over  
 1443 the feature shuffles in all splits, highlighting that our conclusion is consistent with varying train and validation node ratios.  
 1444 Second, the performance gap between different splits generally reduces over the increasing shuffled node ratio. That is, the  
 1445 effect of feature shuffle may also interact with the number of train labels, hinting that A-X dependence may influence the  
 1446 generalization capacity of GNNs. Analysis of GNN generalization, however, is beyond the scope of the present work, and  
 1447 we leave it up to future studies.

1448 **Noisy features.** Third, Fig. 16(c) shows the full results of Fig. 10 for all 12 high  $\mathbf{h}_c$  datasets. We draw the same conclusion  
 1449 as in Sec. 5.

1450 **Proximity-based features.** Last, Fig. 16(d) shows the full results of Fig. 10 for all 12 high  $\mathbf{h}_c$  datasets. We draw the same  
 1451 conclusion as in Sec. 5.

1452 The extensive experiments empirically support our conclusion that A-X dependence mediates the effect of graph convolution.

1453

1454

1455

1456

1457

1458

1459

1460

1461

1462

1463

1464

1465

1466

1467

1468

1469

1470

1471

1472

1473

1474

1475

1476

1477

1478

1479

1480

1481

1482

1483

1484

**Feature Distribution on Graph Topology Mediates the Effect of Graph Convolution**

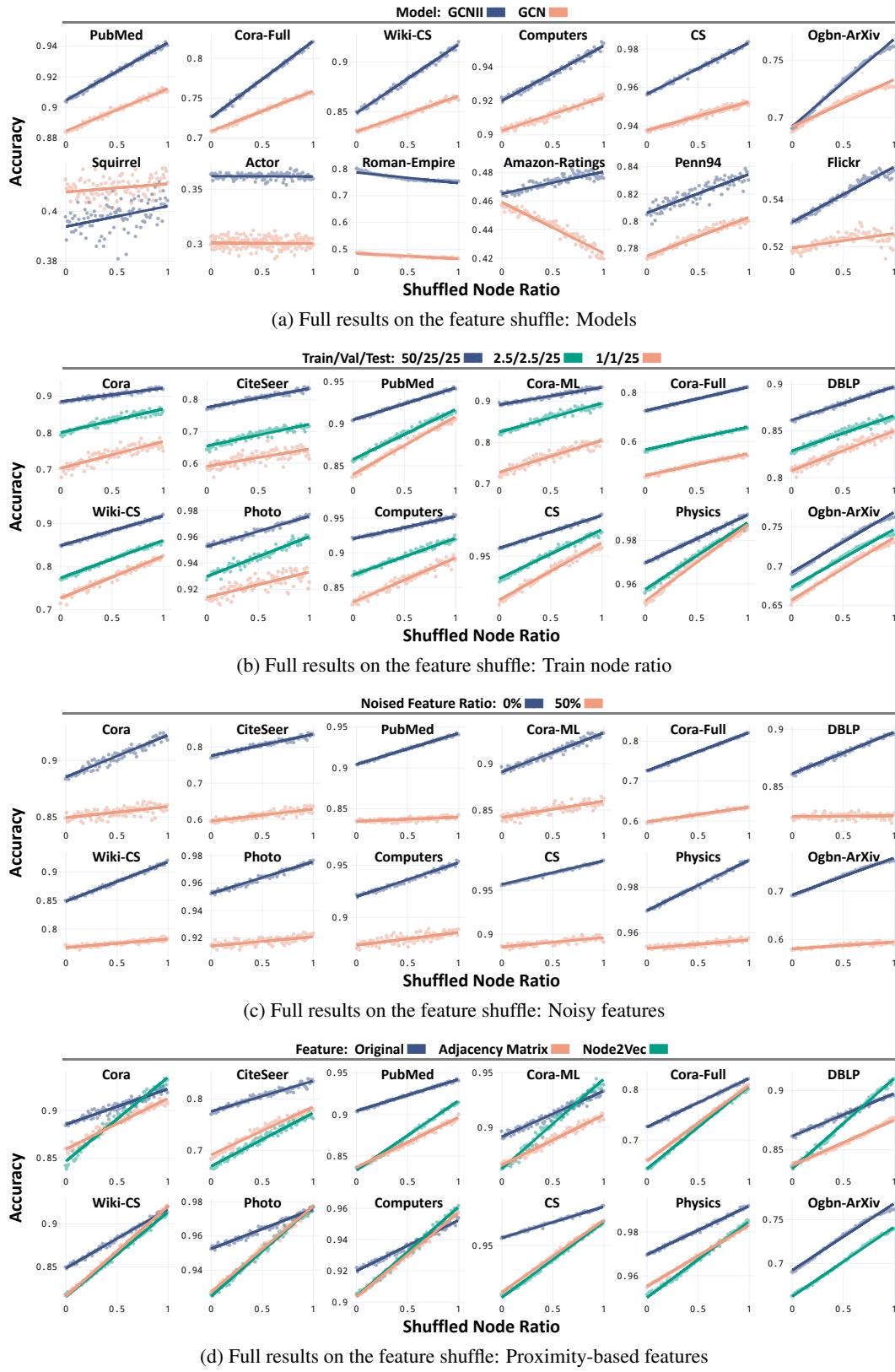


Figure 16: **Full Experimental Results on the Feature Shuffle.**

---

**F. Experiment Settings: Feature Shuffle, Pre-processing, Training, Hyperparameters, and Details**
**F.1. On the Feature Shuffle**

The feature shuffle reduces A-X dependence without perturbing X-Y and A-Y dependence (feature distance FD and classhomophily  $h_c$ , respectively), providing a suitable experimental setting to answer our research question. Thus, the feature shuffle serves to generate synthetic versions of the benchmark graphs.

**F.2. Dataset Pre-processing**

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

**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.

**F.3. 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.

**F.4. 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  $\in \{5e - 3, 1e - 3, 5e - 4\}$

- 1595 • Dropout  $\in \{0.5, 0.6, 0.7\}$
- 1596 • Number of MLP layers  $\in \{1, 2\}$
- 1597 • Number of convolution layers  $\in \{4, 8, 16\}$
- 1598 • Weight decay  $\lambda \in \{0.5, 1.0, 1.5\}$

## 1600 F.5. Other Details

1601 **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.

1602 **Noisy features.** To obtain noisy features for Figs. 10 and 16, we (i) randomly chose 50% of all nodes and (ii) randomly 1603 permute their feature vectors *irrespective of their class*. Thereby, we contaminate the dependence between node features 1604 and class. This is distinguished from the feature shuffle since the feature shuffle is done only among the same class nodes 1605 (Fig. 1).

1606 **Node2Vec.** For Figs. 10 and 16, we use Node2Vec (Grover & Leskovec, 2016) as the node features. For each graph, the 1607 Node2Vec vector is 256-dimensional. To obtain the vector, we train the Node2Vec model with a walk length of 20, a context 1608 size of 10, walks per node of 10, and 100 epochs.

1609 **Shuffle and CFH  $\tilde{h}(\cdot)$ .** When measuring CFH after the feature shuffle, we average the outcomes over 5 trials.

## 1610 G. In-Depth Discussion

1611 **On feature shuffle.** Our finding that feature shuffle improves GNN-based node classification may seem contradictory to some established findings (e.g. node feature smoothing on graph topology improves node classification). However, our 1612 conclusions are complementary to the existing findings. Smoothing makes neighbor node features to be similar, thereby 1613 *increasing A-X dependence*. When class-homophily is assumed, the smoothness is a good quality for the *final node 1614 embeddings*. The feature shuffle randomizes neighbor node feature distribution, thereby *reducing* the A-X dependence. As 1615 shown in Theorem 4.3 and Figures 5-10, the low A-X dependence is a good quality for the *input node features* such that it 1616 encourages *effective smoothing* by the graph convolution.

1617 **On practical applications: GNN engineering.** Our conclusion provides immediate insights and guides for GNN engineering. 1618 In practice, obtaining and preprocessing node features is often an art, heavily relying on engineers' intuition and resources. 1619 For engineers interested in node classification with GNNs, our findings provide a guide on obtaining the 'right' features. 1620 Specifically, the features should have low CFH  $\tilde{h}(\cdot)$ , while being informative for node class (i.e. high feature distance FD).

1621 We first describe the setting for a thought experiment. We are interested in using GNNs to predict user *biological sex* 1622 (class  $Y$ ; potentially non-binary) in an online *friendship network* (graph topology  $A$ ). Biological sex  $Y$  often displays 1623 homophily, so we would assume medium-to-high  $h_c$ . In obtaining the node features  $X$ , we may choose to collect either 1624 user *height* (feature  $X$  candidate 1) or *political inclination* (feature  $X$  candidate 2). We assume both of them have similar 1625 feature distance FD w.r.t. biological sex  $Y$ , because both *heights* and *political inclination* have been shown to correlate with 1626 biological sex  $Y$ .

1627 In such a case, our study suggests that using user *height* will return higher GNN performance. This is because people do not 1628 tend make friends based on their *heights*, but by their *biological sex*. That is, after controlling for the effect of biological sex 1629  $Y$ , we would find small-to-no dependence between *height* and friendship network  $A$ , which makes its class-controlled feature 1630 homophily (CFH) small. Meanwhile, people tend to make friends by their *political inclinations*, even after controlling for 1631 biological sex  $Y$ . Thus, CFH for *political inclination* would be high.

1632 The finding from this example is highly intuitive. Even for an extremely tall woman, the average height among all her 1633 friends would be similar to an extremely short woman. Meanwhile, the average height would be somewhat distant from the 1634 mean height among men (or the other non-binary sexes). This would make it easier to classify their biological sex  $Y$  after 1635 graph convolution.

1636 **GNN theory.** Many works have investigated GNN theory from a node classification perspective. We review the studies in 1637 (semi-) chronological order and discuss how the present work fills their gap.

1638 The early studies found some failure cases of GNNs. NT & Maehara (2019) show that a graph convolution layer is simply 1639

1650 a low-pass filter for node features. They claim that under noisy features and non-linear feature spaces, GNN-based node  
 1651 classification may readily become ineffective. [Oono & Suzuki \(2020\)](#) further show that over-smoothing of node features in  
 1652 GNNs inevitably occurs at infinite model depth.

1653 The following works analyzed how GNNs behave at shallower model depths, demonstrating that the effect of graph  
 1654 convolution depends on feature informativeness, class homophily, and node degree. [Baranwal et al. \(2021\)](#) focus on how  
 1655 they let GNNs obtain more linearly separable features for each class. [Wei et al. \(2022\)](#) study how the factors interact with  
 1656 GNNs' non-linearity, and [Wu et al. \(2023\)](#) investigate their role in triggering over-smoothing.  
 1657

1658 Aligned with the theory, low class homophily (often just called heterophily) has received significant attention as GNNs'  
 1659 'nightmare'. A stream of empirical findings continued to show that GNN performance drops significantly in low class  
 1660 homophily benchmark datasets ([Pei et al., 2020](#); [Zhu et al., 2020, 2021](#); [Chien et al., 2021](#)), and some works investigated the  
 1661 relationship between low class homophily and over-smoothing ([Bodnar et al., 2022](#); [Yan et al., 2022](#)).

1662 However, studies began to demonstrate that low class homophily, *per se*, does not deteriorate GNN performance. [Ma et al.](#)  
 1663 ([2022](#)) and [Platonov et al. \(2023a\)](#) demonstrate that as long as the class distribution is informative w.r.t. node class, GNNs  
 1664 can effectively perform node classification even with low class homophily.  
 1665

1666 Recently, studies have delved into how microscopic patterns of class homophily affect GNNs. [Luan et al. \(2023\)](#) (roughly)  
 1667 argue that, for GNNs to well-classify a node class, its 'intra-class distance' should be smaller than the 'inter-class distance.'  
 1668 That is, low class homophily may trigger the 'inter-class distance' to be smaller to degrade GNN performance. [Mao et al.](#)  
 1669 ([2023](#)) delve into mixed patterns of class homophily and heterophily. Specifically, they show that GNNs better classify  
 1670 the nodes with the majority pattern in the mixture. Lastly, [Wang et al. \(2024\)](#) investigate an array of low class homophily  
 1671 patterns and show that there exist good, mixed, and bad patterns for GNNs to learn from.  
 1672

1673 Not to mention that the role of A-X dependence has not been adequately addressed by the prior literature, the present  
 1674 work can also be interpreted as an extension of the works on homophily-GNN connection to continuous feature domain.  
 1675 Intuitively, a large homophily slows feature mixing by graph convolution, and a small homophily accelerates it. From such a  
 1676 perspective, our conclusion that CFH should ideally be small, while class homophily be large, is an intuitive outcome. To  
 1677 better classify node classes, the mixing between classes should occur at a slow rate, whereas the mixing within each class  
 1678 should occur faster.  
 1679

1680 In summary, we find that the question of how A-X dependence may affect GNNs has been unanswered. Our work highlights  
 1681 that the beneficial effect of graph convolution relies on small A-X dependence. Even with high class homophily or  
 1682 informative features, a large A-X dependence can result in degraded GNN performance (Fig. 5). Considering the current  
 1683 landscape of GNN theory research, we expect the present work to inspire new study directions.  
 1684  
 1685  
 1686  
 1687  
 1688  
 1689  
 1690  
 1691  
 1692  
 1693  
 1694  
 1695  
 1696  
 1697  
 1698  
 1699  
 1700  
 1701  
 1702  
 1703  
 1704

1705 **Algorithm 2** Degree-Preserving CSBM-X

---

1706 1: **Input:** number of nodes  $n$ , number of classes  $c$ , feature mean vector  $\mu_\ell$ 's and covariance matrix  $\Sigma_\ell$ 's, node degree vector  $\mathbf{d}$ , and A-X dependence strength  $\tau$ .

1707   /\* Step 0. Initialize nodes, edges, and class \*/

1708 2:  $V \leftarrow [n]$ ,  $E \leftarrow \emptyset$

1709 3:  $Y \leftarrow$  a random permutation of  $[\mathbf{0}_{\frac{n}{c}} \| \mathbf{1}_{\frac{n}{c}} \| \dots \| \mathbf{c}_{\frac{n}{c}}]$

1710   /\* Step 1. Sample node features and degrees \*/

1711 4: **for**  $v_i \in V$  **do**

1712   5:  $X_i \sim \mathcal{N}(\mu_{Y_i}, \Sigma_{Y_i})$

1713   6:  $d_i^+, d_i^- \leftarrow \lfloor \mathbf{d}_i r \rfloor, \lfloor \mathbf{d}_i (1-r) \rfloor$

1714 7: **end for**

1715   /\* Step 2. Sample directed edges \*/

1716 8: **for**  $v_i \in V$  **do**

1717   9:  $\mathbf{C}_i^+ \leftarrow [j]_{v_j \in C_{Y_i}^+ \setminus \{v_i\}}, \Phi_i^+ \leftarrow [e^{\tau \mathbf{h}_{ij}^{(p)}} : j = \mathbf{C}_{i,t}^+]_{t=1}^{|C_{Y_i}^+|-1}$

1718 10:  $\mathbf{C}_i^- \leftarrow [j]_{v_j \in C_{Y_i}^-}, \Phi_i^- \leftarrow [e^{\tau \mathbf{h}_{ij}^{(p)}} : j = \mathbf{C}_{i,t}^-]_{t=1}^{|C_{Y_i}^-|}$

1719 11:  $N_i^+ \leftarrow \text{WS}_{\text{wr}}(\mathbf{C}_i^+, \Phi_i^+, d_i^+)$  (\*)

1720 12:  $N_i^- \leftarrow \text{WS}_{\text{wr}}(\mathbf{C}_i^-, \Phi_i^-, d_i^-)$

1721 13:  $E \leftarrow E \cup \{(v_i, v_j)\}, \forall v_j \in (N_i^+ \cup N_i^-)$

1722 14: **end for**

1723 15: **return**  $\mathcal{G}(n, c, r, d_{min}, d_{max}, \alpha, \mu_\ell, \Sigma_\ell, \tau) = (V, E)$

(\*)  $\text{WS}_{\text{wr}}(\mathbf{C}, \Phi, d)$  denotes weighted sampling without replacement given a vector of population  $\mathbf{C}$ , a sample weight vector  $\Phi$ , and the sample size  $d$ .

---

## H. Rebuttal: Additional Experiments

In this section, we provide additional experimental results, responding to Reviewer comments.

### H.1. CSBM-X Model Extension

**Model design.** While the proposed CSBM-X (Algorithm 1) 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 propose CSBM-X2 (Algorithm 2). 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 node degrees follow the power-law distribution.

Specifically, the nodes are equally divided into  $c$  classes (line 3). Also, for each node  $v_i$ , CSBM-X2 samples its initial degree  $u_i$  from the Pareto distribution (line 6). Since the sampled initial degree  $u_i$  is bounded by  $[0, \infty)$ ,  $d_{min}$  is added and  $d_{max}$  is used to clipping if the  $u_i$  is large (line 7), bounding the node degree  $d_i \in [d_{min}, d_{max}]$ . By the parameter  $r$ , the degree is divided into the same-class and different-class degrees ( $d_i^+, d_i^-$ ). Thereby, CSBM-X2 reflects the key properties of real-world graphs, including power-law degree distribution, multiple communities, and multiple node classes.

**Experiment setting.** We generate CSBM-X2 graphs with various parameter configurations, generally following the setting in Sec. 4.3. We fix the number of nodes  $n = 10,000$ , number of classes  $c = 10$ ,  $\alpha = 1.5$ , and minimum and maximum degrees ( $d_{min} = 20, d_{max} = 1,000$ ). 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 CSBM-X2 graphs have a wide range of feature distance FD, class homophily  $\mathbf{h}_c$ , and CFH  $\tilde{\mathbf{h}}(\cdot)$ .

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

**Experiment results.** Figure ?? 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 generalizability of our conclusion that A-X dependence mediates the effect of graph convolution.

### H.2. Feature Shuffle

**Additional datasets and GNNs.** The experimental setting is identical to the one in Sec. 5.1. However, we conduct additional experiments with more graph datasets and GNNs to further test the generalizability of our findings.

**Datasets.** We add ten more datasets. The datasets comprise of six *homogeneous* graphs (Twitch, Github, Facebook, Email, USA, Europe), one *heterogeneous* graph (Aminer), and three *hypergraphs* (Cora-CA, DBLP-CA, IMDB).<sup>12</sup> Thereby, we use a total of 34 benchmark graph datasets, including (1) bibliographic networks, (2) e-commercial networks, (3) social networks, (4) web-page networks, (5) geographic networks, (6) online communication networks, (7) text semantics network, (8) and synthetic networks. Their details can be found in Appendix C.5.

**GNNs.** In addition to the GNNs used in Sec. 5 (GCNII, AERO-GNN, GPR-GNN), we adopt four more GNNs: A-DGN (Gravina et al., 2023), Polynormer (Deng et al., 2024), HCHA (Bai et al., 2021), GraphSAGE (Hamilton et al., 2017). Thereby, the GNNs consist of *spectral* (GCNII), *spatial* (GraphSAGE), *adaptive-filter-based* (GPR-GNN), *transformer-based* (Polynormer), *graph-attention-based* (AERO-GNN), *neural-ODE-based* (A-DGN), and

<sup>12</sup>Because USA, Europe, Email, and Aminer datasets are not equipped with external node features, we use their adjacency matrices as the input features.

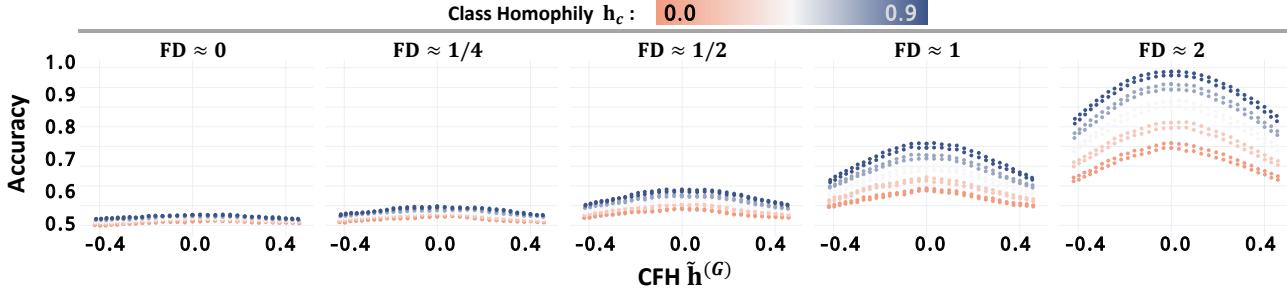
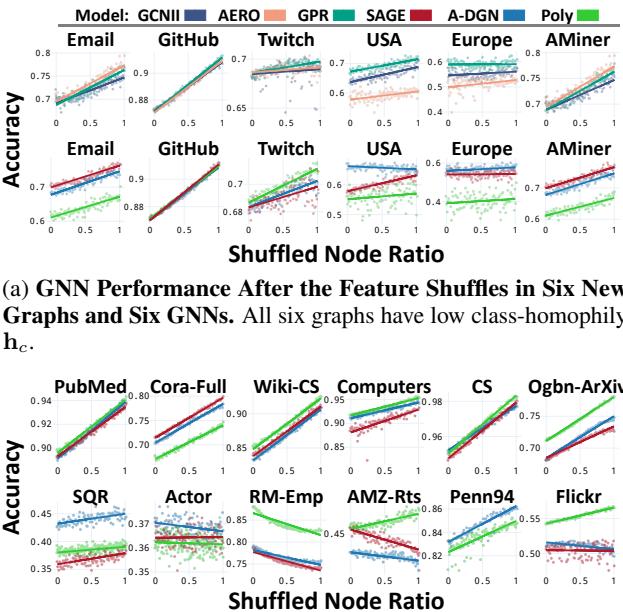
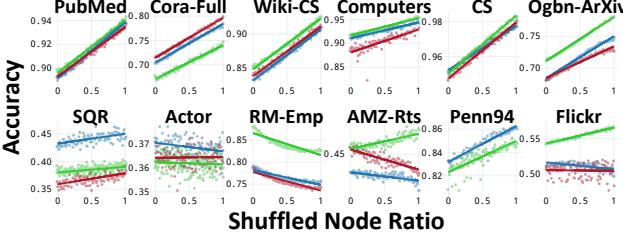


Figure 17: **The Simplified GNN Performance in Degree-Preserving CSBM-X Graphs.** The degrees follow the power-law distribution. Consistent with Theorem 4.3 and Figure 5, for given feature distance  $FD > 0$  and class homophily  $h_c > 0$ , the simplified GNN performance increases as graph-level CFH  $h^{(G)} \rightarrow 0$  ( $\tau \rightarrow 0$ ).



(a) **GNN Performance After the Feature Shuffles in Six New Graphs and Six GNNs.** All six graphs have low class-homophily  $h_c$ .



(b) **GNN Performance After the Feature Shuffles with Six New GNNs.** The graphs in the first row have high class-homophily  $h_c$ , whereas the ones in the second row has low class-homophily  $h_c$ .

Figure 18: **GNN Performance After the Feature Shuffles with Additional GNNs and Datasets.** The results with 6 more datasets and 3 more GNNs are largely consistent with the ones in Figure 6-9. over the feature shuffles, (1) GNN performances generally increase and (2) the rates of increase is generally smaller than those in high class-homophily graphs.

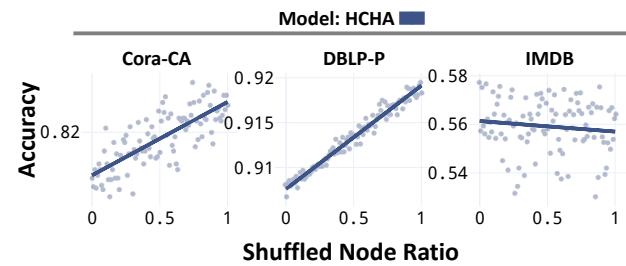


Figure 19: **GNN Performance After the Feature Shuffles in Hypergraphs.** Cora-CA and DBLP-P have high class-homophily  $h_c$ , whereas IMDB has low class-homophily  $h_c$ . The results with hypergraphs and hypergraph neural networks (HCHA) are largely consistent with the ones in Figure 6-9. Over the feature shuffles, (1) performances generally increase in all six GNN models and (2) the rates of increase is smaller than those in high class-homophily graphs.

*hypergraph-convolution-based (HCHA) models.*

**Experiment results.** Figure ?? shows the results with the newly added 10 datasets, and the results with the added GNN models are displayed in Figure ???. Our findings from Sec. 5 are consistently reproduced with different datasets and GNNs. Again, our results highlight the generalizability of our conclusion.

### H.3. Application

#### Algorithm.

#### Experiment results.