Heterophily and Graph Neural Networks: Past, Present and Future

Jiong Zhu[†] Yujun Yan[‡] Mark Heimann[§] Lingxiao Zhao[‖] Leman Akoglu[‖]
Danai Koutra[†]
[†] University of Michigan {jiongzhu, dkoutra}@umich.edu
[‡] Dartmouth College yujun.yan@dartmouth.edu
[§] Lawrence Livermore National Laboratory heimann2@llnl.gov
[‖] Carnegie Mellon University {lingxia1, lakoglu}@andrew.cmu.edu

Abstract

Recently, there has been interest in understanding the performance of Graph Neural Networks (GNNs) on input graphs exhibiting heterophily, or the tendency for nodes of different classes to connect. Initial findings showed that many standard GNN models struggled on certain benchmark datasets exhibiting high heterophily, prompting research into existing and novel GNN designs that improved learning in these contexts. However, further analyses revealed that certain highly heterophilous settings did not challenge GNNs without these specialized designs, raising questions about the true factors causing performance degradation. In this work, we first review various GNN designs proposed for handling graphs with heterophily, and examine their connections to other GNN research objectives such as robustness, fairness, and oversmoothing avoidance. Next, we conduct an empirical study to investigate the specific heterophilous graph conditions under which GNNs can and cannot perform effectively. Our analysis reveals that although high heterophily does not universally impede conventional GNNs, unique challenges in heterophilous graphs, particularly the intertwined effects with low-degree nodes and complex compatibility patterns, warrant GNN designs specifically tailored to heterophily. In conclusion, we discuss future research directions aimed at advancing the understanding of the impact of heterophily on GNNs across a broader range of contexts.

1 Introduction

Graph Neural Networks (GNNs) [51, 43] have gained prominence in recent years due to their remarkable theoretical and empirical potential for learning powerful representations of graph-structured data. Many real-world graphs or networks exhibit homophily, where nodes predominantly connect with others belonging to the same class [27, 58]. While early GNNs demonstrated promise on graphs with this property, they faced challenges on graphs exhibiting heterophily, where the majority of nodes connect to those of different classes [1, 29, 58]. This prompted investigations into GNN design choices conducive to learning on graphs with heterophily and sparked interest in developing new GNN models tailored for this property [58, 46, 6, 49, 24, 46, 52, 33].

Beyond improving the effectiveness of GNNs on heterophilous datasets, recent research has shown that the challenges posed by graphs with heterophily are closely connected to other GNN challenges, including oversmoothing [19, 5], algorithmic bias [18, 40], and sensitivity to adversarial attacks [60, 8, 44, 42, 17, 25]. Designs addressing heterophily often improve the ability of GNNs to handle these challenges as well, leading to significant advances in overall GNN capabilities [6, 46, 23, 55, 4].

Another line of work, however, has revisited whether early GNN designs were as ill-suited for learning from heterophilous graphs as initially thought. On some heterophilous networks, basic Graph Convolutional Networks (GCNs) [16] have proven competitive with, or even outperformed, models specifically designed for heterophily [26, 24]. This has led to the proposition that the challenges posed by some graph datasets are not best captured by the traditional homophily ratio. Consequently, other works have focused on analyzing the properties of heterophilous graphs that challenge early GNNs and designing generalized homophily metrics that offer more insight into the difficulties a graph dataset may present [26, 24]. Thus, a valid debate exists over whether "heterophily" is a real problem that GNNs face.

Our work revisits this debate with additional analysis. First, we provide a concise review of recent designs for graphs with heterophily, their connections to other GNN research objectives, as well as generalized heterophily metrics. We then examine the heterophilous conditions under which conventional GNNs have been shown to be competitive with those tailored for heterophily. Our analysis reveals that while conventional GNNs can sometimes succeed in learning on heterophilous graphs without specialized designs, such condition is often broken when the underlying data has low-degree nodes and complex heterophilous patterns ("compatibility matrices"). Thus, we believe that continuing to develop GNNs that can learn across the spectrum of low-to-high homophily remains an important theoretical and empirical problem. We summarize our contributions as follows:

- We review and summarize recent designs proposed for graphs with heterophily (§3.1), providing a unifying intuition. Moreover, we discuss their use in subsequent GNN works and their implications for other objectives of GNN research (§3.2), such as fairness, robustness, and reducing oversmoothing.
- We conduct an empirical analysis on the conditions under which conventional GNNs can succeed on heterophilous datasets (§4). Our analysis demonstrates the unique challenges in achieving high separability of Neighborhood Label Distribution (NLD) when low-degree nodes (§4.2.2) or complex heterophilous patterns (§4.2.3) are present. These challenges hinder the effectiveness of conventional GNNs and are best addressed by GNN designs specifically tailored for heterophily (§4.2.4).
- We discuss future research directions aimed at enhancing our understanding of how heterophily impacts GNNs across a broader range of contexts (§5). These include moving beyond node classification and global homophily, introducing more diverse graph datasets and applications, and exploring the connections between heterophily and heterogeneity.

2 Notation and Preliminaries

In this section, we give the key notations and definitions that we use throughout our paper. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected, unweighted graph with node set $\mathcal{V}$ and edge set $\mathcal{E}$. We denote a general neighborhood centered around v as $N(v)$ ($\mathcal{G}$ may have self-loops), the corresponding neighborhood that does *not* include the ego (node v) as $\bar{N}(v)$, and the general neighbors of node v at exactly i hops/steps away (minimum distance) as $\bar{N}_i(v)$. For example, as shown in Fig. 1, $\bar{N}_1(v) = \{u : (u, v) \in \mathcal{E}\}$ are the immediate neighbors of v. We represent the graph by its adjacency matrix $\mathbf{A} \in \{0, 1\}^{n \times n}$ and its node feature matrix $\mathbf{X} \in \mathbb{R}^{n \times F}$, where the vector $\mathbf{x}_v$ corresponds to the *ego-feature* of node v, and $\{\mathbf{x}_u : u \in \bar{N}(v)\}$ to its *neighbor-features*. We further represent the degree of a node v by d_v, which denotes the number of neighbors in its immediate neighborhood $\bar{N}_1(v)$.

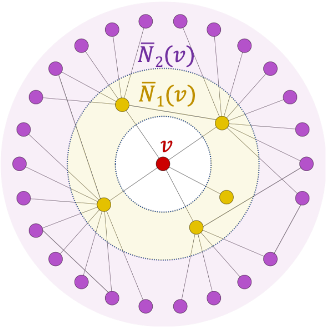


Figure 1: Neighborhoods.

We further assume a class label vector $\mathbf{y}$, which for each node v contains a unique class label $y_v \in \mathcal{Y}$, and the one-hot encoding $\mathrm{onehot}(y_v)$ forms the row vectors of label encoding matrix $\mathbf{Y} \in \{0, 1\}^{n \times |\mathcal{Y}|}$. We further define $\mathcal{V}_i$ as the set of nodes $v \in \mathcal{V}$ with label $y_v = i$. The goal of semi-supervised node classification is to learn a mapping $\ell : \mathcal{V} \to \mathcal{Y}$, given a set of labeled nodes $\mathcal{T}_{\mathcal{V}} = \{(v_1, y_1), (v_2, y_2), ...\}$ as training data.

Graph Neural Networks (GNNs). From a probabilistic perspective, most GNN models assume the following local Markov property on node features: for each node $v \in \mathcal{V}$, there exists a neighborhood $N(v)$ such that y_v only depends on the ego-feature $\mathbf{x}_v$ and neighbor-features $\{\mathbf{x}_u : u \in N(v)\}$. Most models derive the class label y_v via the following representation learning approach:

$$\mathbf{r}_v^{(k)} = f\left(\mathbf{r}_v^{(k-1)}, \{\mathbf{r}_u^{(k-1)} : u \in N(v)\}\right), \ \mathbf{r}_v^{(0)} = \mathbf{x}_v, \text{ and } y_v = \arg\max\{\text{softmax}(\mathbf{r}_v^{(K)})\mathbf{W}\}, \tag{1}$$

where the embedding function f is applied repeatedly in K total rounds, node v's representation (or hidden state vector) at round k, $\mathbf{r}_v^{(k)}$, is learned from its ego- and neighbor-representations in the previous round, and a softmax classifier with learnable weight matrix $\mathbf{W}$ is applied to the final representation of v. Most existing models differ in their definitions of neighborhoods $N(v)$ and embedding function f. A typical definition of neighborhood is $N_1(v)$—i.e., the 1-hop neighbors of v. As for f, in graph convolutional networks (GCN) [16] each node repeatedly averages its own features and those of its neighbors to update its own feature representation. Using an attention mechanism, GAT [37] models the influence of different neighbors more precisely as a weighted average of the ego- and neighbor-features. GraphSAGE [12] generalizes the aggregation beyond averaging, and models the ego-features distinctly from the neighbor-features in its subsampled neighborhood.

Homophily and heterophily. In this work, we focus on heterophily in class labels. We first define the edge homophily ratio h as a measure of the graph homophily level, and use it to define graphs with strong homophily/heterophily:

Definition 2.1 (Edge Homophily Ratio [1, 58]) *The edge homophily ratio $h = \frac{|\{(u,v):(u,v)\in\mathcal{E}\wedge y_u=y_v\}|}{|\mathcal{E}|}$ is the fraction of edges in a graph which connect nodes that have the same class label (i.e., intra-class edges).*

Definition 2.2: Graphs with strong homophily have high edge homophily ratio $h \to 1$, while graphs with strong heterophily (i.e., low/weak homophily) have small edge homophily ratio $h \to 0$.

The edge homophily ratio in Dfn. 2.1 gives an *overall trend* for all the edges in the graph. The actual level of homophily may vary within different pairs of node classes, i.e., there is different tendency of connection between each pair of classes. For instance, in an online purchasing network [28] with three classes—fraudsters, accomplices, and honest users—, fraudsters connect with higher probability to accomplices and honest users. Moreover, within the same network, it is possible that some classes exhibit homophily, while others exhibit heterophily; we give an example in Figure 2. To capture the tendency of connection between each pair of classes, we define the empirical *class compatibility matrix* $\mathbf{H}$ as follows:

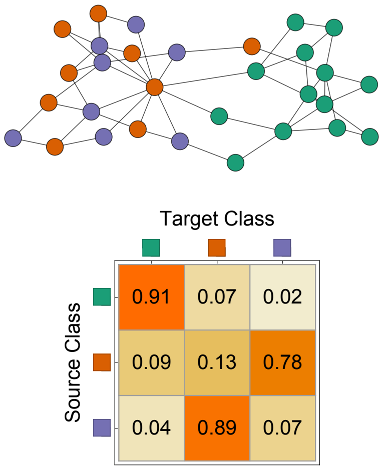


Figure 2: An example graph (top) and its empirical class compatibility matrix $\mathbf{H}$ (bottom). It demonstrates mixed homophily and heterophily, with node colors represent class labels: nodes in green show strong homophily, while nodes in orange and purple show strong heterophily.

Definition 2.3 (Empirical Class Compatibility Matrix [58, 57]) *The empirical class compatibility matrix $\mathbf{H}$ has entries $[\mathbf{H}]_{i,j}$ that capture the fraction of edges from a node in class i to a node in class j:*

$$[\mathbf{H}]_{i,j} = \frac{|\{(u,v) : (u,v) \in \mathcal{E} \wedge y_u = i \wedge y_v = j\}|}{|\{(u,v) : (u,v) \in \mathcal{E} \wedge y_u = i\}|}$$

By definition, the class compatibility matrix is a stochastic matrix, with each row summing up to 1.

Heterophily $\neq$ Heterogeneity. We remark that heterophily, which we study in this work, is a distinct network concept from heterogeneity. Formally, a network is heterogeneous [35] if it has at least two types of nodes and different relationships between them, and homogeneous if it has a single type of nodes (e.g., users) and a single type of edges (e.g., friendship). The type of nodes in heterogeneous graphs does *not* necessarily match the class labels y_v, therefore both homogeneous and heterogeneous networks may have different levels of homophily.

3 Progress for Addressing Heterophily in GNNs

In this section, we first present a concise overview of effective design strategies proposed to enhance GNN performance under heterophily (§3.1), and then discuss the implications of these designs for other GNN research in robustness, fairness, and reducing oversmoothing (§3.2).

3.1 Effective Designs for Graph Neural Networks on Heterophilous Graphs

We present an overview of the effective design strategies that have been recently proposed to enhance GNN performance on heterophilous graphs. We initiate our discussion with widely adopted designs (D1-D4) in GNN architectures for heterophily, three of which were initially explored in [58]. Subsequently, we examine two emerging designs (D5-D6) introduced by Yan et al. [46], which offer a novel unified approach to address two significant challenges faced by GNNs: oversmoothing and heterophily. Our primary focus in this section is to discuss the design principles and their underlying intuition for improving learning under heterophily without delving into specific models; we direct interested readers to a comprehensive survey by Zheng et al. [52] for more details about particular models.

3.1.1 Ego- and Neighbor-embedding Separation

Zhu et al. [58] identified three designs for improving the performance of GNNs on heterophilous graphs and provided theoretical justifications. At a high level, the first design entails encoding each ego-embedding (i.e., a node's embedding) *separately* from the aggregated embeddings of its neighbors, since they are likely to be dissimilar in heterophily settings. Formally, the representation (or hidden state vector) learned for each node v at GNN layer with depth k is given as:

$$\mathbf{r}_v^{(k)} = \texttt{COMBINE}\left(\mathbf{r}_v^{(k-1)},\ \texttt{AGGR}(\{\mathbf{r}_u^{(k-1)} : u \in \bar{N}(v)\})\right), \tag{2}$$

the neighborhood $\bar{N}(v)$ does *not* include v (no self-loops), the `AGGR` function aggregates representations *only* from the neighbors (in some way—e.g., average), and `AGGR` and `COMBINE` may be followed by a non-linear transformation. For heterophily, after aggregating the neighbors' representations, the definition of `COMBINE` (akin to 'skip connection' between layers) is critical: the ego-embedding and the aggregated neighbor-embedding should be processed by different sets of weight matrices under `COMBINE`. A simple way to combine the ego- and the aggregated neighbor-embeddings without 'mixing' them is with concatenation as in GraphSAGE [12]—rather than averaging *all* of them as in the GCN model by Kipf and Welling [16]. Intuitively, [58] argues that choosing a `COMBINE` function that separates the representations of each node v and its neighbors $\bar{N}(v)$ allows for more expressiveness, where the skipped or non-aggregated representations can evolve separately over multiple rounds of propagation without becoming prohibitively similar to representations aggregated from neighbors.

While this design was first discussed in [58] as the most critical design in the context of improving GNN performance under heterophily, it had already been proposed and adopted in prior GNN models such as GraphSAGE [12], without addressing the problem of heterophily. GCN-Cheby [9] and MixHop [1] also feature a variant of this design, with the `AGGR` function operating on $N(v)$ (with self-loops) instead of $\bar{N}(v)$ (no self-loops), while still featuring a separate channel for the ego-embedding. Following H$_2$GCN proposed in [58], this design has gained wide adaptation for GNNs designed with heterophilous graphs in mind, such as CPGNN [57], GPR-GNN [6], FAGCN [49], FSGNN [50], JacobiConv [20], GGCN [46], GBK-GNN [10], ACM [24], and OrderedGNN [33]. More recently, Platonov et al. [30] conducted benchmark experiments on additional heterophilous datasets and showed that GNNs featuring this design, including GAT [37] and UniMP [32] modified to include this design, achieve the best results in nearly all cases, which further validates the importance of the ego & neighbor embedding separation.

3.1.2 Higher-order Neighborhoods

The second design in [58] involves explicitly aggregating information from higher-order neighborhoods in each GNN layer, beyond the immediate neighbors of each node:

$$\mathbf{r}_v^{(k)} = \mathtt{COMBINE}\left(\mathbf{r}_v^{(k-1)},\ \mathtt{AGGR}_1(\{\mathbf{r}_u^{(k-1)} : u \in \bar{N}_1(v)\}),\ \mathtt{AGGR}_2(\{\mathbf{r}_u^{(k-1)} : u \in \bar{N}_2(v)\}), \ldots\right), \tag{3}$$

where $\bar{N}_i(v)$ denotes the neighbors of v at *exactly* i hops away, and the $\mathtt{AGGR}_i$ functions applied to different neighborhoods can be the same or different. This design—first employed in GCN-Cheby [9] and MixHop [1]—augments the *implicit* aggregation over higher-order neighborhoods that most GNN models achieve through multiple layers of first-order propagation based on variants of Eq. equation 2. Zhu et al. [58] attribute the effectiveness of this design to observations that even though the immediate neighborhoods may be heterophilous, the higher-order neighborhoods may show homophily in certain datasets (e.g., binary attribute prediction on 2-partite graphs [2, 7]) and thus provide more relevant context to GNNs.

Early implementations of this design, such as GCN-Cheby [9] and MixHop [1], extract embeddings from higher-order neighborhoods $\bar{N}_i(v)$ within each layer by employing "Delta Operators" [1]. These operators differentiate the aggregated embeddings in different orders of the (normalized) adjacency matrices $\mathbf{A}^i$ and $\mathbf{A}^{i-1}$ for improved computational efficiency. In contrast, H$_2$GCN [58], UGCN [13], TDGNN [39], and OrderedGNN [33] precisely compute the i-hop neighborhoods $\bar{N}_i(v)$ for each node v before applying the $\mathtt{AGGR}_i$ functions to prevent mixing nodes from different hops. Notably, the recent approach by Song et al. [33] achieves state-of-the-art classification accuracy on heterophilous datasets by modeling message passing within higher-order neighborhoods using a rooted-tree hierarchy, and aligning segments of variable length in the resulting node embeddings with specific neighborhood orders.

3.1.3 Combination of Intermediate Representations

The third design proposed in [58] combines the intermediate representations of each node at the final layer:

$$\mathbf{r}_v^{(\text{final})} = \mathtt{COMBINE}\left(\mathbf{r}_v^{(1)}, \mathbf{r}_v^{(2)}, \ldots, \mathbf{r}_v^{(K)}\right). \tag{4}$$

This approach explicitly captures both local and global information using $\mathtt{COMBINE}$ functions that process each representation individually, such as concatenation or LSTM-attention [45]. This design was initially introduced in jumping knowledge networks [45] and demonstrated to enhance the representation power of GCNs under *homophily*. Intuitively, each GNN layer gathers information with varying degrees of locality—earlier layers focus on local information, while later layers increasingly capture global information (implicitly, through propagation). Similar to D2 (which models explicit neighborhoods), this design models the distribution of neighbor representations in low-homophily networks more accurately. It also allows the class prediction to leverage different neighborhood ranges in different networks, adapting to their structural properties.

The application of this design is often linked to graph spectral theory: Zhu et al. [58] provided a theoretical justification for this design from the perspective of graph spectral filtering. Building upon this foundation, GPR-GNN [6], FAGCN [3], and ACM [24] further enhance GNN performance under heterophily by developing additional graph filters and mixture mechanisms to utilize embeddings generated with varying frequency components at the final layer, in conjunction with this design.

3.1.4 Similarity-based Attention and Neighbor Discovery

The designs identified in [58] focus on boosting the effectiveness of message passing on heterophilous graphs without modifying the underlying structure. An alternative approach, however, is to go beyond the original graph adjacency and discover additional connections between the nodes in the graph, based on the similarity their original or latent features (e.g., structural embeddings), which replace or augment the original heterophilous structure of the graph in the message passing. Specifically, UGCN [13], SimP-GCN [14], NL-GNN [22], HOG-GCN [38] and GPNN [47] update the message-passing graph for GNNs by removing or downweighting the heterophilous edges in the original graph (i.e., edges that connect nodes with dissimilar features or structural embeddings), while introducing newly discovered connections that exhibit strong homophily. On the other hand, Geom-GCN [29] and WRGNN [36] leverage for each node both its original graph neighborhood and the derived "structural neighborhood" based on proximity of structural node embeddings in order to augment the message passing and aggregation process.

3.1.5 Signed Messages & Gated Kernel

In most GNN models [16, 37], messages are positively aggregated from neighbors and transformed using a single kernel or weight matrix. However, in heterophilous graphs, this may degrade GNN performance when messages from neighbors of different classes are mixed [58, 46]. Although attention-based GNNs, such as GAT [37], can theoretically reduce aggregation weights on heterophilous edges, they may still accumulate noise in the generated embeddings in practice.

An intuitive solution to address this issue is to learn signed messages (e.g., GGCN [46] and GReTo [54]) or gated kernels (e.g., GBK [10]) that separate message passing between homophilous intra-class and heterophilous inter-class edges. Yan et al. [46] suggested that ideally, messages from neighbors of a different class should be multiplied by a negative sign ("negative messages"), while messages from neighbors of the same class should remain unchanged. However, ground truth node labels are inaccessible in real scenarios, and any approximated sign function may introduce errors. To identify conditions when signed messages can enhance node classification performance, [46] introduced the concept of "error rate" that quantifies the portion of non-ideal messages and analyzed node classification performance under various error rates and homophily levels. The benefits of using signed messages can also be interpreted from the perspective of graph spectrum: signed messages allow negative mixture of certain frequency components [49, 6], helping models better capture high-frequency components in node features. This is especially beneficial for learning on heterophilous graphs as they contain abundant high-frequency components in their node features, unlike homophilous graphs [58].

From the perspective of practical model design, GGCN [46] and GReTo [54] used proximity between node features to approximate the sign function. As an alternative to signed messages, Du et al. [10] proposed a gated bi-kernel design that applies separately to the message passing of homophilous and heterophilous edges, and adopted a learnable gate function to distinguish between the two types of edges based on the node features.

3.1.6 Degree Corrections

Zhu et al. [58] first noted that the performance divide between low- and high-degree nodes is exacerbated on heterophilous graphs (c.f. Figure 5). Later, Yan et al. [46] provided a thorough theoretical and empirical analysis of how the interplay of degrees and homophily levels affects the node classification accuracy. Specifically, two node-level properties were defined: relative degree $\bar{\theta}_u$, which evaluates the degree of a node compared to its neighbors' degrees; and node-level homophily h_u, which captures the tendency of a node to have the same class as its neighbors. Formally, the relative degree of a node u is defined as $\bar{\theta}_u \equiv \mathbb{E}_{\mathbf{A}|d_u}(\frac{1}{d_u}\sum_{v\in N_1(u)}\theta_{uv}|d_u)$, where $\theta_{uv} \equiv \sqrt{\frac{d_u+1}{d_v+1}}$; and the node-level homophily h_u is defined as $h_u \equiv \mathbb{P}(y_u = y_v | v \in N_1(u))$. The authors discovered that nodes with higher relative degrees outperform the nodes with lower ones under certain conditions of node features when the design of signed messages (D5) is employed. To improve the performance of nodes with lower relative degrees, they proposed a degree correction strategy which learns to virtually increase the relative degree of the nodes via structure-based edge attention weights $\tau_{uv}^{l} = \texttt{softplus}\left(\lambda_0^l\left(\frac{1}{\theta_{uv}} - 1\right) + \lambda_1^l\right)$, where λ_0^l and λ_1^l are the learnable parameters at the l-th GNN layer. If θ_{uv} is small, a large τ_{uv}^{l} is learned, which compensates for the current relative degree.

3.2 Heterophily and Other Objectives of GNN Research

Numerous studies have demonstrated that tackling the limitations of GNNs under heterophily not only enhances their performance on heterophilous datasets, but also improves their properties in other aspects of GNN research. In this section, we provide an overview of the connections that have been investigated between heterophily and adversarial robustness, algorithmic fairness, and oversmoothing, all of which are also important for deployment.

Heterophily & Robustness. Recent works have shown that GNNs have a high sensitivity to adversarial attacks [60, 8, 44, 42, 17, 25]. While most previous works have focused on naturally-occurring heterophily, heterophilous interactions may also be introduced as adversarial noise: as many GNNs exploit homophilous correlations, they can be sensitive to changes that render the data more heterophilous. This relation between adversarial structural attacks and the change of homophily level was first suggested though empirical analyses on homophilous graphs [42, 15], and was later formalized by Zhu et al. [55] with theoretical and additional empirical analyses. Specifically, Zhu et al. [55] showed that on homophilous graphs, effective structural attacks lead to increased heterophily, while, on heterophilous graphs, they alter the homophily level contingent on node degrees: for low-degree nodes, attacks increasing the heterophily are still effective, but for high-degree nodes, attacks decreasing the heterophily will be effective. By leveraging these relations, the authors further demonstrated that some key architectural designs for effectively handling heterophily—separate aggregators for ego- and neighbor-embeddings (D1) and Combination of Intermediate Representations (D3)—also improve the robustness of GNNs against attacks. Following these relations, a follow-up work proposed a defense framework called CHAGNN that improves the robustness of GNNs against Graph Injection Attacks (GIA) by iteratively pruning the heterophilous edges in the graph and retraining the GNN model [59].

Heterophily & Fairness. Algorithmic fairness is a critical aspect of machine learning that ensures a model does not disproportionately underperform for certain input classes. In the context of link prediction in networks, fairness is desirable to prevent the prediction accuracy from being influenced by sensitive node attributes, such as race or religion in a social network context. To promote fairness in Graph Neural Networks (GNNs), previous research has suggested learning a fair reweighting or rewiring of the graph structure alongside the parameters of the GNN [18, 34]. Theoretical analysis has shown that the effectiveness of these approaches depends on the weights of the intergroup edges (essentially, heterophilous edges according to sensitive attribute), along with the group sizes and other structural attributes of the graph. For node classification, the global homophily ratio of a graph has revealed to be crucial in providing bounds for group fairness concerning a sensitive attribute [40]. Other research has examined GNN fairness with respect to local homophily ratios within individual node neighborhoods [23], revealing that variations in local homophily can impact model fairness, and that GNN designs for heterophily can empirically enhance group fairness.

Heterophily & Oversmoothing. The oversmoothing problem relates to the degenerated performance of GNNs with an increasing number of layers [19]. Though both the heterophily and oversmoothing problems are associated to the unsatisfactory performance of GNNs, they do not appear to be related at a first glance. However, evidence from both empirical [5, 6] and theoretical analysis [46, 4] has found that the two problems may share the same root causes and may be addressed with the same approaches. Chen et al. [5] addressed the oversmoothing problem via initial residual and identity mapping, but their designs were found empirically to help improve the node classification performance on heterophilous graphs. Vice versa, Chien et al. [6] addressed the heterophily problem via generalized PageRank, but they showed that their designs are also effective for addressing the oversmoothing problem. Yan et al. [46] are the first to explicitly analyze the relationship between the two problems. They found that the two problems can be jointly explained by analyzing the changes in the node representations over the layers, and proposed two designs, namely signed messages (D5) and degree corrections (D6), to address the two problems jointly. Later, Bodnar et al. [4] used cellular sheaves theory to explain the two problems jointly. They found that the underlying geometry of the graph is related to the performance of GNNs in heterophilous settings and their oversmoothing behavior, and many GNNs implicitly assume a graph with a trivial underlying sheaf. These observations and analyses have shown promising results in addressing the two problems jointly, which is an interesting direction to explore further.

4 Revisiting When is Heterophily Challenging for GNNs

While many works have focused on designing new GNN models with improved performance under heterophily, few of them have probed whether heterophily persistently presents challenges for GNNs. Some of these works have found that GNNs without the aforementioned heterophilous designs (e.g., SGC [41], GCN [16], GAT [37]) can exhibit better or equivalent performance to GNNs possessing such designs on certain datasets [26, 24]. In this section, we first summarize the main findings of these works, and show that the complexity of heterophily can be measured based on the distinguishability of the Neighborhood Label Distributions (NLDs).[1] We then highlight two key factors, low-degree nodes and complex compatibility matrices, which deteriorate the distinguishability of the neighborhood label distributions when coupled with heterophily, thus making heterophily a unique challenge for GNNs in most cases.

4.1 Improved Measures for Complexity of Heterophily

While many works measure the level of homophily/heterophily by the ratio of edges that connect nodes with the same class label (e.g., edge homophily in Dfn. 2.1, node homophily [29], or class homophily [21]), recent works have shown that graphs with high heterophily are not always challenging for GNNs without heterophilous designs. Through independent analyses, Ma et al. [26] and Luan et al. [24] arrive at the conclusion that the complexity of heterophily is closely related to the distinguishability of the neighborhood label distributions, which we define next.

Definition 4.1 (Neighborhood Label Distribution (NLD)) *Given $\mathbf{Y}$ as the label encoding matrix defined in §2 for nodes $\mathcal{V}$ in graph $\mathcal{G}$, the neighborhood label distribution of node v is defined as $\mathbf{D}(v) = \frac{1}{|N_1(v)|}\sum_{u \in N_1(v)} \mathbf{Y}_u$, where $\mathbf{Y}_u = \text{onehot}(y_u)$ is the v-th row of the label encoding matrix $\mathbf{Y}$.*

We now rephrase the two metrics proposed by Ma et al. [26] and Luan et al. [24] with the above definition, both of which measure the complexity of heterophily by quantifying the distinguishability of $\mathbf{D}(v)$.

Definition 4.2 (Class Neighborhood Similarity (CNS) [26]) *The class neighborhood similarity between classes $i, j \in \mathcal{Y}$ is defined as the average cosine similarity between the NLDs $\mathbf{D}(v), \mathbf{D}(u)$ of nodes v, u in class i and j, respectively, i.e.,*

$$S(i,j) = \frac{1}{|\mathcal{V}_i||\mathcal{V}_j|} \sum_{v \in \mathcal{V}_i} \sum_{u \in \mathcal{V}_j} \text{sim}_{\cos}(\mathbf{D}(v), \mathbf{D}(u)), \tag{5}$$

[1] In parallel with these studies, Yan et al. [46] conducted a theoretical analysis of performance degradation in heterophilous networks under the "non-swapping" condition. This condition emerges when the neighboring representations for each node are insufficient to cause the interchange of node representations from two distinct classes across their separation plane in the latent space. Conversely, the case of "easy heterophily" [26, 24] that we address in this section corresponds to the "swapping" condition as articulated in [46].

where $\mathcal{V}_i$ and $\mathcal{V}_j$ are the sets of nodes with class label i and j, and $\text{sim}_{\text{cos}}(\cdot)$ is the function of cosine similarity. We refer to the case of $i = j$ as intra-class neighborhood similarity (intra-CNS) and the case of $i \neq j$ as inter-class neighborhood similarity (inter-CNS).

Definition 4.3 (Graph Aggregation Homophily [24]) *Define the average similarity score of a node $v \in \mathcal{V}$ to nodes $\mathcal{V}_i$ with class label $i \in \mathcal{Y}$ as $g(v, i) = \text{mean}\left(\{\text{sim}(\mathbf{D}(v), \mathbf{D}(u)) : v, u \in \mathcal{V}, y_u = i\}\right)$, where sim is a function (e.g., dot product) that measures the similarity between two neighborhood label distributions. The graph aggregation homophily is then defined as the ratio of nodes $v \in \mathcal{V}$ where the neighborhood label distribution $\mathbf{D}(v)$ is more similar for nodes in the same class than for nodes in any other class, i.e.,*

$$h_{\text{agg}} = \frac{1}{|\mathcal{V}|} \left| \left\{ v \in \mathcal{V} : g(v, y_v) \geq \max_{j \neq y_v \in \mathcal{Y}} g(v, j) \right\} \right|. \tag{6}$$

We note that while h_{agg} measures the *proportion* of nodes with NLD exhibiting greater similarity (regardless of the extent) to nodes within the same class compared to nodes from different classes, it does not quantify the *degree of similarity* between NLDs of nodes within the same class or across different classes, which is captured by the CNS metric. Consequently, as we show in our empirical analysis in §4.2.4, CNS provides a more comprehensive and accurate assessment of the complexity of heterophily on synthetic datasets, and thus we focus on CNS in our empirical analysis below.

4.2 Factors Determining the Complexity of Heterophily

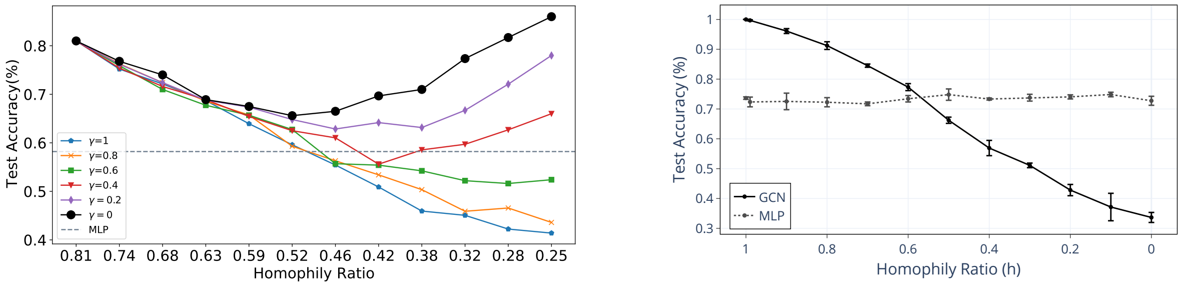
It has been shown that it is possible to have graphs with high level of heterophily but low complexity for GNNs as measured by CNS or aggregation homophily [26, 24]: when nodes in the same class have strong similarity with respect to neighborhood label distributions, and nodes from different classes have weak or no similarity, GCN models are able to perform well due to the high distinguishability of the neighborhood label distributions, even when the graphs are heterophilous. These are important findings, but this type of analysis does not provide a complete picture of the complexity of heterophily for GNNs, as the high distinguishability of the class label distributions under heterophily is largely dependent on key graph properties, such as degree distributions and the compatibility matrices that drive the generation of the graph. In this section, we provide a detailed analysis of the above two factors that determine how challenging the data heterophily is for GNNs.

4.2.1 Motivating Example: Differences in Synthetic Datasets

Prior research exploring the impact of heterophily on GNN performance frequently incorporates experiments on synthetic datasets with controlled homophily/heterophily levels [1, 58, 26, 24]. In line with this research, in this section we provide a motivating example based on synthetic data that showcases the role of the two factors—namely, degree distribution and compatibility matrices—in characterizing how challenging heterophily is for GNN models. We analyze the seemingly contradictory observations arising from the results on two distinct synthetic datasets based on the `Cora` dataset: `syn-cora` [58] and `necessity-cora` [26].

Before diving into the analysis of the factors that affect the complexity of heterophily, we first provide a brief overview on the setup and key results on the two synthetic datasets [58, 26].

Data Generation: `syn-cora` vs. `necessity-cora`. While both synthetic datasets are generated based on `Cora`, their generation processes are largely different. The `syn-cora` dataset [58, 57] follows a modified preferential attachment process. In this process, the probability of a new node u with class label c to attach to existing node v with class label c' is proportional to: (1) the ratio $\mathcal{D}_{c,c'}$ specified in the underlying compatibility matrix $\mathcal{D}$, which determines the homophily level in the resulting graph, as empirically measured by the edge homophily ratio h (Dfn. 2.1) and the compatibility matrix $\mathbf{H}$ (Dfn. 2.3), and (2) the degree d_v of the existing



(a) Accuracy on `necessity-cora` under different noise levels γ. Figure is reproduced from [26]; shared with permission by the authors.

(b) Accuracy on `syn-cora` for GCN and MLP reported by [58]. Figure is adapted from [58].

Figure 3: Semi-supervised node classification accuracy of GCN and MLP observed in [26] and [58] on `necessity-cora` and `syn-cora`, respectively, under increasing level of heterophily (i.e., decrease of the edge homophily ratio h).

node v. This process results in a power-law degree distribution in the generated graph. On the other hand, `necessity-cora` [26] varies the level of homophily by adding heterophilous (cross-label) edges on top of the existing (homophilous) edges in `Cora`. To control the randomness of the added heterophilous edges, `necessity-cora` adds: (1) non-random heterophilous edges based on an underlying compatibility matrix $\mathcal{D}$, and (2) random edges that do not follow the underlying compatibility matrix $\mathcal{D}$, but are controlled by a noise parameter γ.

Observations: `syn-cora` vs. `necessity-cora`. When the level of heterophily is varied, largely different observations are reported on the two sets of synthetic graphs with respect to the GNN performance: [26] shows that on synthetic graphs with none or few randomly-added heterophilous connections (i.e., with the noise parameter γ close to 0), the performance of GCNs can even increase as the level of heterophily in the graph gets stronger (i.e., when the edge homophily ratio h decreases), as shown in Figure 3(a); on the other hand, [58] shows that the performance of GCNs significantly decreases as the heterophily increases, which we show in Figure 3(b). As our follow-up analysis below shows, these seemingly contradictory results are due to the different processes used to generate the synthetic graphs, which lead to very different graph properties (i.e., degree distribution, class compatibility matrix); these in turn affect the model performance. We analyze the effects of the (F1) degree distribution in §4.2.2 and the (F2) compatibility matrix in §4.2.3.

4.2.2 Factor (F1): Degree Distributions & Heterophily

In §4.1, we revisit the findings from recent works [26, 24] that the complexity of heterophily for GNNs is largely determined by the distinguishability of the Neighborhood Label Distributions (NLDs) of nodes with different class labels. Under the generation process of `necessity-cora` with noise $\gamma = 0$ (§4.2.1), when classes are different $c \neq c'$ and the distributions $\mathcal{D}_c$ and $\mathcal{D}_{c'}$ are distinguishable from each other, one would state that the GCN models can perform well to distinguish the nodes with class label c from the nodes with class label c' from the perspective of NLD distinguishability.

However, we argue that the aforementioned statement ignores the critical factor of degree for each node $v \in \mathcal{V}_c$ that impacts the quality of the samples of the distribution $\mathcal{D}_c$: when all the nodes have sufficiently large degrees, it is expected that $\mathcal{D}_c$ can be recovered well in the node neighborhoods due to sufficient samples of the distributions; however, when many low-degree nodes are present in the graph (which is the case for many real-world graphs, which usually follow power-law degree distributions), $\mathcal{D}_c$ may not be consistently recovered in the neighborhood of the low-degree nodes under heterophily due to the insufficiency of the samples. This affects the intra-class and inter-class similarity of the NLDs in heterophilous settings.

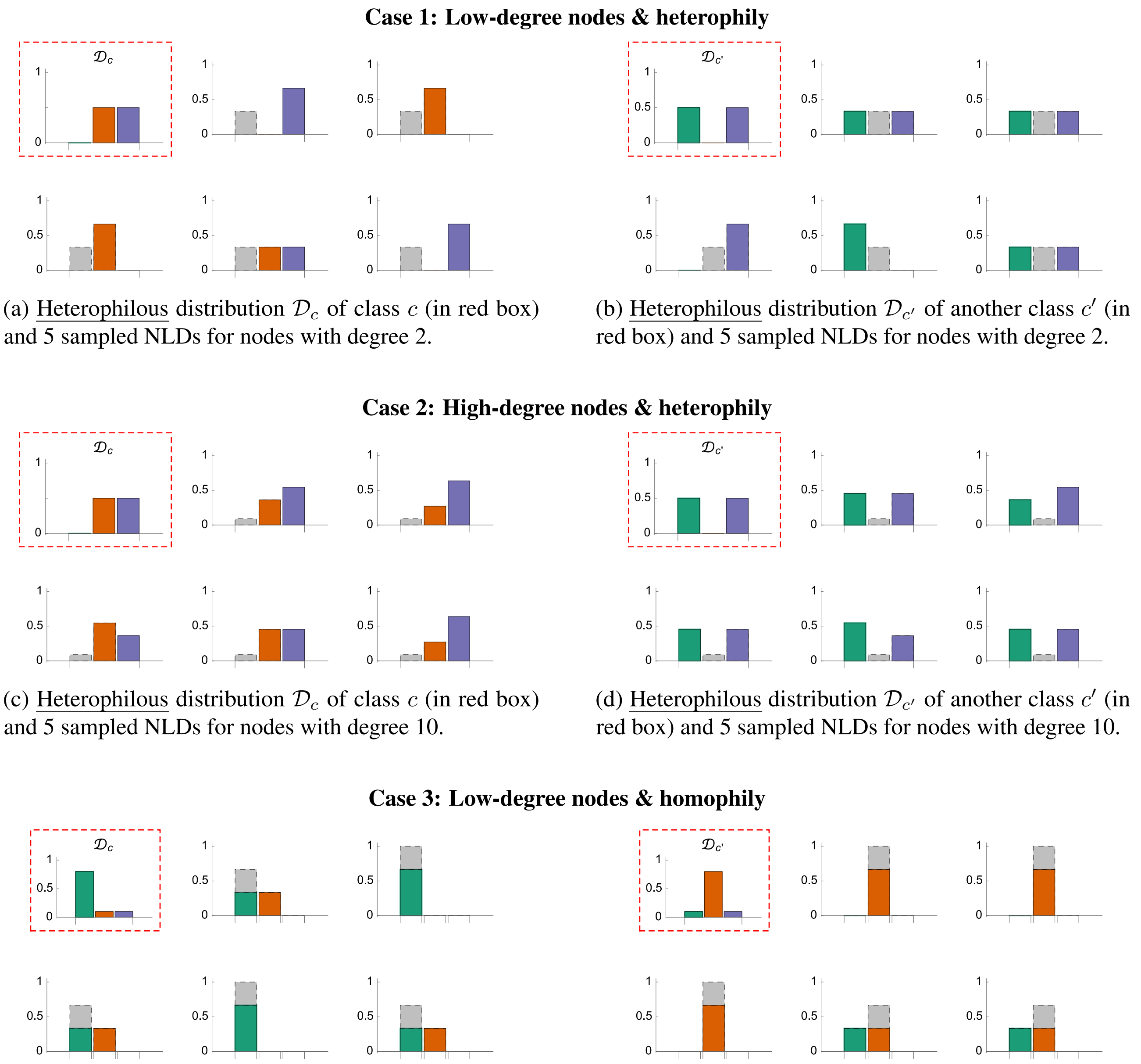
Empirical Analysis. We can further explain how low-degree nodes affect the intra-class and inter-class distinguishability of the NLD for GCNs under heterophily with a simple empirical analysis: Suppose the neighborhood label distributions $\mathcal{D}_c$ and $\mathcal{D}_{c'}$ for two classes $c \neq c'$ are given in the red dashed boxes in Figure 4. Following the distributions $\mathcal{D}_c$ and $\mathcal{D}_{c'}$, we randomly generate the NLDs of 200 nodes with degree 2 for both classes c and c'; then, we sample the labels of their 2 neighbors, and we visualize a random set of 5 of the 200 synthetic NLDs in Figure 4(a)-(b), for $\mathcal{D}_c$ and $\mathcal{D}_{c'}$, respectively. We note that since GCN aggregators additionally consider a self-loop for each node, the NLD observed by GCN models should be considered with self-loops added to the graphs, even when $\mathcal{D}_c$ and $\mathcal{D}_{c'}$ dictate purely heterophilous connections. To visualize the contributions of self-loops in the NLDs, we show them in gray in Figure 4.

- *Case 1: Low-degree nodes & heterophily.* Figure 4(a)-(b) show that the existence of low-degree nodes reduces the distinguishability of the NLDs. Specifically, we observe that: (1) the intra-class NLDs have a high variance and can be very different from the corresponding ground-truth distributions (even when not considering the self-loops). In fact, the mean and standard deviation of the intra-class pairwise cosine similarity among the 200 synthetic neighborhoods of class c and c' are 0.79 ± 0.24, where the high standard deviation reflects the strong variance among the sampled neighborhood distributions. (2) Many of the NLDs from nodes in class c, c' are the same when considering the self-loops, which affects their distinguishability across different classes; the inter-class pairwise cosine similarity for our synthetic neighborhoods of class c and c' is 0.79 ± 0.17 in our analysis, which is the same as the intra-class pairwise similarity with even smaller standard deviation.

- *Case 2: High-degree nodes & heterophily.* On the other hand, when the node degrees are high, the NLDs are more similar to the underlying distributions $\mathcal{D}_c$ and $\mathcal{D}_{c'}$ (even with self-loops considered) and thus have much smaller variances. In our example, we randomly sampled NLDs of another 200 nodes with degree 10 (instead of 2), and illustrate them for 5 randomly selected nodes in Figure 4(c)-(d). The mean and standard deviation of the pairwise cosine similarity among the 200 generated neighborhoods of class c and c' are 0.93 ± 0.09, while the inter-class pairwise similarity is only 0.64 ± 0.15. These changes in intra-class and inter-class similarities can also be observed in the sampled distributions shown in Figure 4(c)-(d).

- *Case 3: Low- / high-degree nodes & strong homophily.* We note that the presence of low-degree nodes does not affect the similarity of NLDs as much in strong homophilous settings as in the heterophilous settings. To show this empirically, we similarly generate the neighborhood label distributions of 200 nodes with degrees of 2 for class c, c', but this time with distributions $\mathcal{D}_c$ and $\mathcal{D}_{c'}$ showing strong homophily. In Figure 4(e)-(f), we observe that, unlike the heterophilous settings, almost all synthetic distributions of nodes from the same class c (or c') are close to the expected distribution $\mathcal{D}_c$ (or $\mathcal{D}_{c'}$); most neighbors (considering self-loops) have the same class label c (or c') as the ego node, even for low-degree nodes. Numerically, the intra-class pairwise cosine similarity among the synthetic neighborhoods of class c and c' is 0.88 ± 0.15 and 0.91 ± 0.13, respectively. On the other hand, the inter-class pairwise similarity is 0.21 ± 0.29, which shows good separability. This example shows that **the presence of low-degree nodes is a challenge that is more pronounced in heterophilous settings than in homophilous settings.**

Summary & connections to other works. From the above analysis, we see that **the existence of low-degree nodes can lead to weak distinguishability of inter-class NLDs, thus affecting the performance of GCNs**. The significant performance gap between low-degree and high-degree nodes is also observed in [58], as shown



(a) Heterophilous distribution $\mathcal{D}_c$ of class c (in red box) and 5 sampled NLDs for nodes with degree 2.

(b) Heterophilous distribution $\mathcal{D}_{c'}$ of another class c' (in red box) and 5 sampled NLDs for nodes with degree 2.

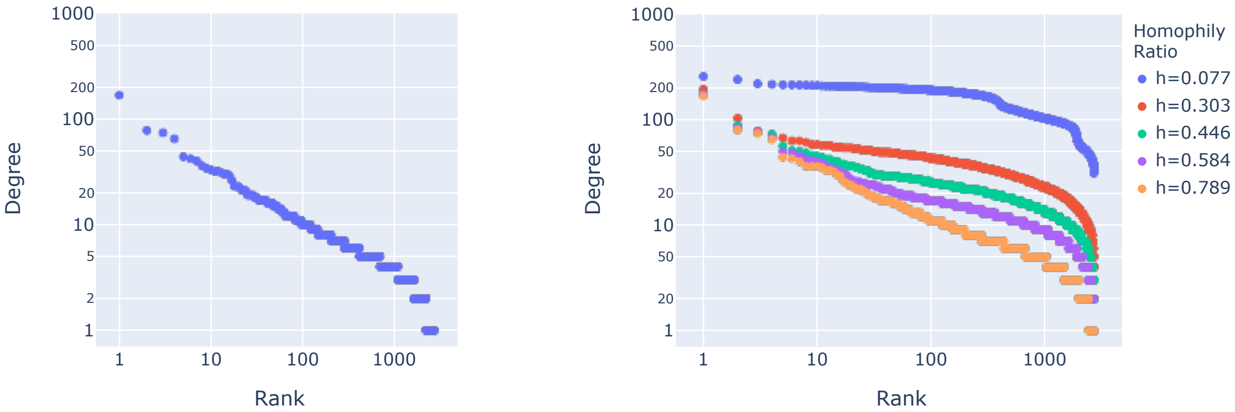
(c) Heterophilous distribution $\mathcal{D}_c$ of class c (in red box) and 5 sampled NLDs for nodes with degree 10.

(d) Heterophilous distribution $\mathcal{D}_{c'}$ of another class c' (in red box) and 5 sampled NLDs for nodes with degree 10.

(e) Homophilous distribution $\mathcal{D}_c$ of class c (in red box) and 5 sampled NLDs for nodes with degree 2.

(f) Homophilous distribution $\mathcal{D}_{c'}$ of another class c' (in red box) and 5 sampled NLDs for nodes with degree 2.

Figure 4: (Factor F1) Degree Distributions: Per case, we sample 200 NLDs from distribution $\mathcal{D}_c$ (and $\mathcal{D}_{c'}$) for nodes with specific degrees, and visualize 5 sampled NLDs. The gray parts correspond to the contributions of self-loops in the NLDs aggregated by GCN. **(a)-(b) Case 1:** Low degrees reduce the distinguishability of NLDs: for all synthetic NLDs of c and c', inter-CNS is $S(c, c') = 0.79 \pm 0.17$, with even smaller standard deviation than intra-CNS $S(c, c) = S(c', c') = 0.79 \pm 0.24$. **(c)-(d) Case 2:** Higher node degrees improve the distinguishability of NLDs: for all synthetic NLDs of c and c', inter-CNS is $S(c, c') = 0.64 \pm 0.15$, which is smaller than the intra-CNS $S(c, c) = S(c', c') = 0.93 \pm 0.09$. **(e)-(f) Case 3:** Low node degrees matter less under homophily: for all synthetic NLDs of c and c', inter-CNS is $S(c, c') = 0.21 \pm 0.29$, which is significantly smaller than the intra-CNS $S(c, c) = 0.88 \pm 0.15$ and $S(c', c') = 0.91 \pm 0.13$.



(a) Degree distribution for `cora`: it follows a typical power-law degree distribution.

(b) Degree distribution of `necessity-cora`, the cora-based synthetic graphs in [26] with $\gamma = 0$ and edge homophily ratio $h \in \{0.077, 0.303, 0.446, 0.584, 0.789\}$.

Figure 6: Degree distributions of `cora` and `necessity-cora`. As the level of heterophily increases (i.e., edge homophily ratio h decreases), the degrees for all the nodes increase in `necessity-cora`, and the degree distributions move further away from the original degree distribution of `cora`. The shift in degree distribution explains the increase of GCN performance with the level of heterophily for the $\gamma = 0$ case in Figure 3(a).

in Figure 5. As a follow-up to our analysis[2], Ma et al. [26] formalized the effects of node degrees on the distinguishability of NLDs for Contextual Stochastic Block Model (CSBM), and derived a lower bound of node degrees for GCN-style aggregation to improve the distinguishability of NLDs.

Revisiting the `necessity-cora` dataset. The degree distributions of `necessity-cora` also explain why GCN performance starts to increase as the level of homophily h decreases in the range of $h < 0.5$ (for noise $\gamma < 0.5$): the `necessity-cora` graphs with homophily ratio $h < 0.5$ have a significantly higher average degree compared to their corresponding base graph `cora`, as a large amount of edges needs to be added in order to decrease the edge homophily ratio in the (strongly homophilous) base graph. In Figure 6, we show the degree distribution of base graph `cora` in comparison to the degree distributions of the `necessity-cora` graphs with $\gamma = 0$ (i.e., when all the heterophilous edges are added according to the underlying compatibility matrix $\mathcal{D}$, without any randomness) for varying edge homophily ratio h. We see that as h decreases, the degrees for all nodes in the graph increase, and the degree distributions move further away from the degree distribution of `cora`; for the $h = 0.077$ instance, even the minimum node degree in the `necessity-cora` graph has exceeded the degree of most nodes in `cora`. In our additional empirical analysis (§4.2.4), we show that the lack of low-degree nodes is indeed a necessary condition that contributes to the observed high performance of GCNs on `necessity-cora`.

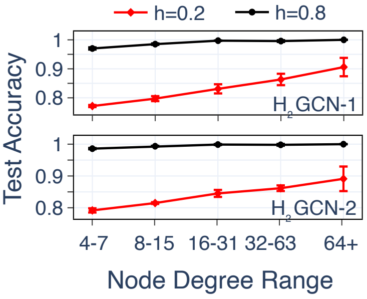
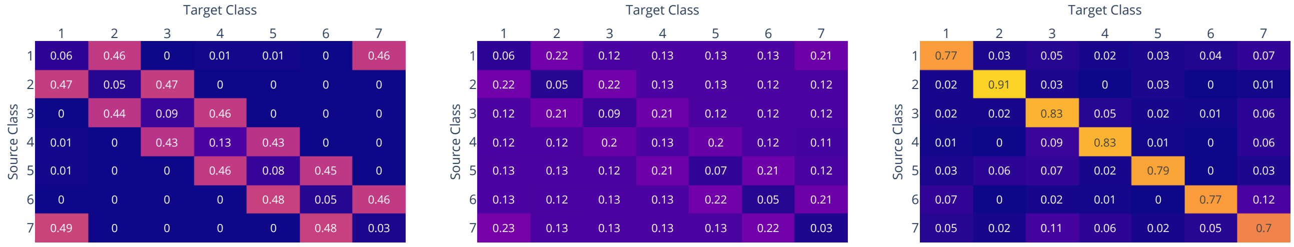


Figure 5: H_2GCN accuracy per degree range on synthetic heterophilous ($h = 0.2$) and homophilous ($h = 0.8$) graphs. Figure from [58].

[2] These analyses were first made available in the form of a blog post: Zhu, J. and Koutra, D. (2021) Revisiting the problem of heterophily for GNNS. Available at: https://www.jiongzhu.net/revisiting-heterophily-gnns/.



	Target Class						
Source Class	1	2	3	4	5	6	7
1	0.06	0.46	0	0.01	0.01	0	0.46
2	0.47	0.05	0.47	0	0	0	0
3	0	0.44	0.09	0.46	0	0	0
4	0.01	0	0.43	0.13	0.43	0	0
5	0.01	0	0	0.46	0.08	0.45	0
6	0	0	0	0	0.48	0.05	0.46
7	0.49	0	0	0	0	0.48	0.03

(a) Compatibility matrix with $\gamma = 0$ and $h = 0.16$ in `necessity-cora`.

	Target Class						
Source Class	1	2	3	4	5	6	7
1	0.06	0.22	0.12	0.13	0.13	0.13	0.21
2	0.22	0.05	0.22	0.13	0.13	0.12	0.12
3	0.12	0.21	0.09	0.21	0.12	0.12	0.12
4	0.12	0.12	0.2	0.13	0.2	0.12	0.11
5	0.13	0.13	0.12	0.21	0.07	0.21	0.12
6	0.13	0.12	0.13	0.13	0.22	0.05	0.21
7	0.23	0.13	0.13	0.13	0.13	0.22	0.03

(b) Compatibility matrix with $\gamma = 0.8$ and $h = 0.16$ in `necessity-cora`.

	Target Class						
Source Class	1	2	3	4	5	6	7
1	0.77	0.03	0.05	0.02	0.03	0.04	0.07
2	0.02	0.91	0.03	0	0.03	0	0.01
3	0.02	0.02	0.83	0.05	0.02	0.01	0.06
4	0.01	0	0.09	0.83	0.01	0	0.06
5	0.03	0.06	0.07	0.02	0.79	0	0.03
6	0.07	0	0.02	0.01	0	0.77	0.12
7	0.05	0.02	0.11	0.06	0.02	0.05	0.7

(c) Compatibility matrix of `cora` with strong homophily.

Figure 7: Comparison of compatibility matrices $\mathbf{H}$ of different synthetic graphs in `necessity-cora` with homophily ratio $h = 0.16$ but different noise ratio γ, with comparison to the compatibility matrix of `cora` (the homophilous graph which `necessity-cora` is based on).

4.2.3 Factor (F2): Compatibility Matrices & Heterophily

Another factor that affects the distinguishability of NLDs is the distinguishability of the compatibility matrices for different classes: Under the generation process of `necessity-cora` (§4.2.1), when the node degrees are sufficiently high in the generated graphs (Case 2 of §4.2.2), the NLDs for nodes $v \in \mathcal{V}_c$ are expected to be similar to $\mathcal{D}_c$. In this case, the distinguishability of NLDs between nodes in class c and c' mostly depends on the distinguishability of $\mathcal{D}_c$ and $\mathcal{D}_{c'}$, which can also be observed empirically in the compatibility matrices $\mathbf{H}$ of the generated graphs. In this section, we discuss how complex compatibility patterns in the rows of $\mathbf{H}$ can reduce the distinguishability of NLDs and contribute to the complexity of heterophily, in addition to (F1) node degrees.

Observations on heterophilous datasets: `necessity-cora` under different noise levels. The differences in the observed performance of GCN on `necessity-cora` under different noise levels γ (i.e., randomness of the heterophilous edges) in Figure 3(a) can be explained by the differences in the distinguishability of the empirical compatibility matrices $\mathbf{H}$ (and the underlying compatibility matrices $\mathcal{D}$ by extension). In Figure 7, we visualize the compatibility matrices of graphs from `necessity-cora` with homophily ratio $h = 0.16$, and compare between graphs with noise levels $\gamma = 0$ and $\gamma = 0.8$: (1) when $\gamma = 0$, the compatibility matrices in `necessity-cora` are formulated to resemble a “loop”, where almost all connections for nodes of class c are limited to the two adjacent classes in the “circle” of classes (e.g., nodes in Class 2 almost exclusively connect to nodes in Class 1 and 3 in Figure 7(a)). This “loop”-pattern helps maintain the high distinguishability of compatibility patterns among different classes, and thus provides an easier node classification problem for GCNs compared to more general heterophilous patterns. (2) In comparison, when $\gamma = 0.8$, the heterophilous connections of class c are distributed to all classes $c' \neq c$, and the rows $\mathbf{H}_c$ of the compatibility matrix are less distinct, which is similar to the case of `syn-cora` (c.f. Figure 8(h)). The high similarity of $\mathbf{H}_c$ among different classes makes it challenging to distinguish different classes from the NLDs even for high-degree nodes, as many heterophilous connections from class c are uniformly distributed to other classes $c' \neq c$. This explains the decrease of GCN accuracy under the same homophily level (and degree distribution[3]) in `necessity-cora` when γ increases, as shown in Figure 3(a).

Observations on homophilous datasets. We also note that, echoing the observation in [26], the **distinguishability** of the rows in compatibility matrix $\mathbf{H}$ **is guaranteed for graphs with strong homophily**, as the largest entries in the distributions are concentrated on the diagonal elements of the compatibility matrix $\mathbf{H}$ as shown

[3] Graphs with the same edge homophily ratio h in `necessity-cora` also have highly similar degree distributions by extension, since the level of homophily is varied by edge addition.

in Figure 7(c). **Thus, the distinguishability of the compatibility matrices is also a challenge specific to heterophilous settings.**

4.2.4 The Interplay of Degree Distribution, Compatibility Matrices & NLDs

In Sections §4.2.2–4.2.3, we discussed two key factors that determine how challenging heterophily is for GNN models. Here, we explore the interplay of these factors and NLDs via an empirical study. To this end, we construct additional synthetic data with properties complementary to those in [58, 26].

Data generation: "loop"-style schema & power-law degree distribution. To study the interplay of factors (F1) & (F2), we generate synthetic graphs which have: (1) (mostly) "loop"-style compatibility matrices (where nodes in each class only connect to nodes in its nearby classes, as if all classes are arranged in a circle), e.g., Figure 7(a). This schema is similar to that used for `necessity-cora`; we leverage the same or similar compatibility matrices as specified in [26] in the generation process. (2) the same power-law degree distribution as `syn-cora` by following the same modified preferential attachment generation process as in [58, 57].

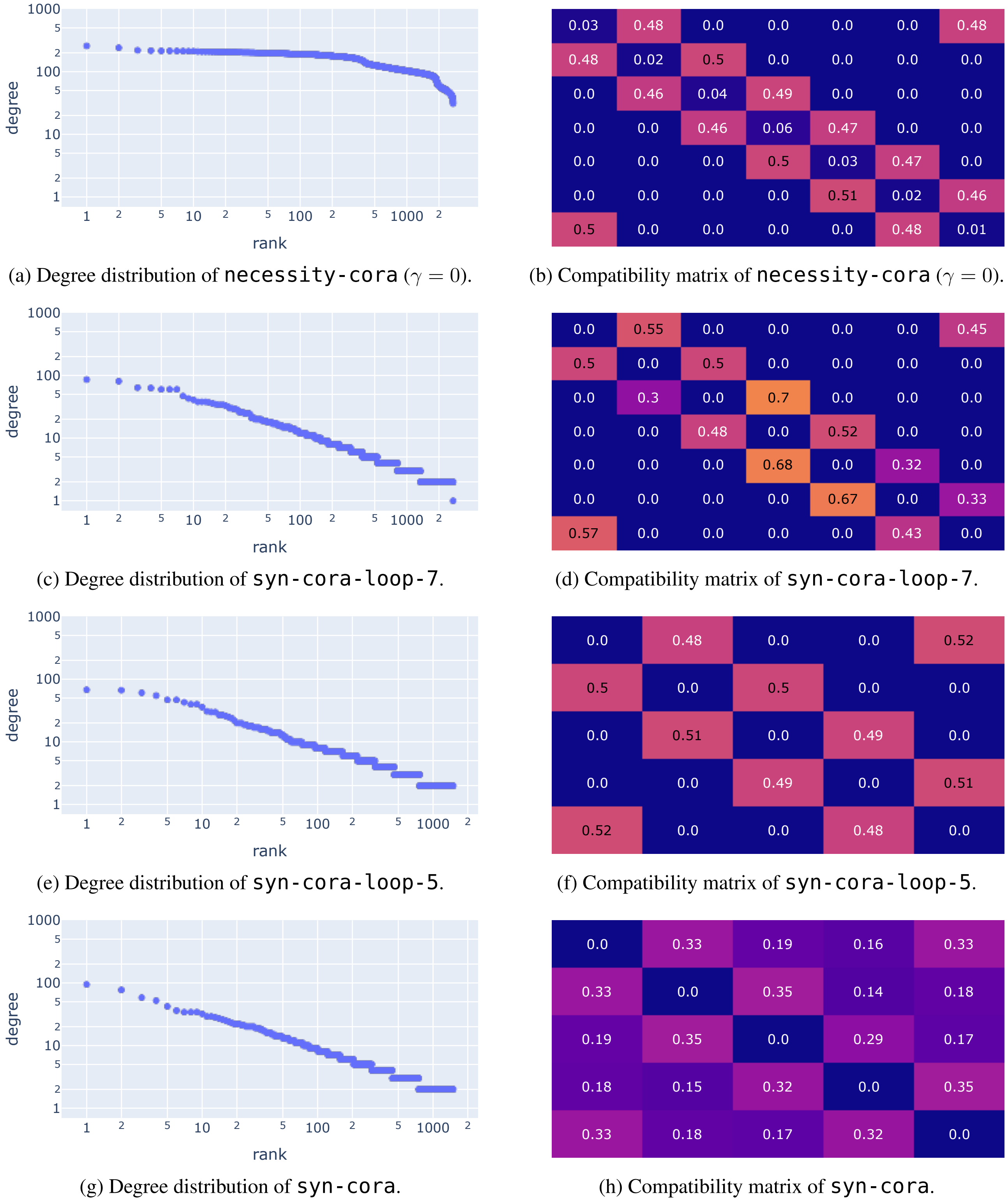
We refer to these synthetic graphs as `syn-cora-loop`, and consider two variants: `syn-cora-loop-7` with 7 classes as in `necessity-cora`, and `syn-cora-loop-5` with 5 classes as in `syn-cora`. In Figure 8, we visualize the degree distributions and compatibility matrices of our `syn-cora-loop` datasets, along with the visualizations for `necessity-cora` and `syn-cora`.

Models. We assess the influence of degree distributions and compatibility matrices on the performance of three GNN models: H_2GCN [58], GCN [16], and MLP. H_2GCN represents GNN models that incorporate one or more heterophilous designs as discussed in §3.1; we examine two variants of H_2GCN, namely H_2GCN-1 and H_2GCN-2, with one or two layers of aggregation respectively. In contrast, GCN serves as the GNN baseline model which does not incorporate any heterophilous designs, while MLP functions as the graph-agnostic baseline that does not consider the graph structure. For GCN, we adopt the same hyperparameter tuning as in [26], and further tune the dimension of hidden embeddings between 16 and 64. For H_2GCN, we only tune a subset of the hyperparameters that we tune for GCN (16 vs. 112 combinations), which are more hyperparameter combinations than those explored in [58]. For each model, we present the mean and standard deviation of the classification accuracy under five runs with different random seeds per dataset.

Data setup. Our experiments incorporate four sets of synthetic graphs: `necessity-cora` provided by Ma et al. [26], `syn-cora` from [58], and the newly generated `syn-cora-loop-7` and `syn-cora-loop-5`. For `syn-cora`, we select the graph with homophily level $h = 0$; for `necessity-cora`, we select the graph with noise parameter $\gamma = 0$ and h nearest to 0 (i.e., $h = 0.03$) as permitted by its generation process. We generate `syn-cora-loop-7` and `syn-cora-loop-5` with $h = 0$. In Table 1, we present the statistics for each dataset; the degree distributions and compatibility matrices for all datasets are visualized in Figure 8. For the train/validation/test splits, we utilize the provided splits for `necessity-cora` [26], and create splits for the other datasets using identical sizes as in `necessity-cora`. Specifically, we randomly select 20 nodes per class for the training set, 500 nodes throughout the graph for the validation set, and allocate the remaining nodes to the test set[4].

GNN performance & graph properties. In Table 1, we list the performance of each model along with the corresponding properties of the graphs. We observe the effects of the interplay between low-degree nodes and more complex compatibility matrices to the performance of GNN models when the graphs share similar edge homophily ratio h (0 or as close to 0 as the generation process allows):

[4] This setup is identical to [16], but differs from [58, 57] (where Figure 3(b) was generated), which utilized a larger training set.



(a) Degree distribution of `necessity-cora` ($\gamma = 0$).

0.03	0.48	0.0	0.0	0.0	0.0	0.48
0.48	0.02	0.5	0.0	0.0	0.0	0.0
0.0	0.46	0.04	0.49	0.0	0.0	0.0
0.0	0.0	0.46	0.06	0.47	0.0	0.0
0.0	0.0	0.0	0.5	0.03	0.47	0.0
0.0	0.0	0.0	0.0	0.51	0.02	0.46
0.5	0.0	0.0	0.0	0.0	0.48	0.01

(b) Compatibility matrix of `necessity-cora` ($\gamma = 0$).

(c) Degree distribution of `syn-cora-loop-7`.

0.0	0.55	0.0	0.0	0.0	0.0	0.45
0.5	0.0	0.5	0.0	0.0	0.0	0.0
0.0	0.3	0.0	0.7	0.0	0.0	0.0
0.0	0.0	0.48	0.0	0.52	0.0	0.0
0.0	0.0	0.0	0.68	0.0	0.32	0.0
0.0	0.0	0.0	0.0	0.67	0.0	0.33
0.57	0.0	0.0	0.0	0.0	0.43	0.0

(d) Compatibility matrix of `syn-cora-loop-7`.

(e) Degree distribution of `syn-cora-loop-5`.

0.0	0.48	0.0	0.0	0.52
0.5	0.0	0.5	0.0	0.0
0.0	0.51	0.0	0.49	0.0
0.0	0.0	0.49	0.0	0.51
0.52	0.0	0.0	0.48	0.0

(f) Compatibility matrix of `syn-cora-loop-5`.

(g) Degree distribution of `syn-cora`.

0.0	0.33	0.19	0.16	0.33
0.33	0.0	0.35	0.14	0.18
0.19	0.35	0.0	0.29	0.17
0.18	0.15	0.32	0.0	0.35
0.33	0.18	0.17	0.32	0.0

(h) Compatibility matrix of `syn-cora`.

Figure 8: Synthetic networks used to study the interplay of factors (F1) and (F2). `syn-cora-loop` datasets have the "loop"-style structure of `necessity-cora` graphs and the power law degree distribution of the `syn-cora` graphs.

Table 1: Dataset statistics and effectiveness of models for node classification. We report the min, median, and max values for the intra-class and inter-CNS, and the mean accuracy ± standard deviation for each model. The best result for each dataset is highlighted in blue.

	necessity-cora	**syn-cora-loop-7**	**syn-cora-loop-5**	**syn-cora**
#Nodes	2,708	2,708	1,490	1,490
#Edges	132,196	5,394	2,968	2,968
# Classes	7	7	5	5
Edge Hom. h	0.03	0	0	0
(F1) High-degree Nodes Only	✓	✗	✗	✗
(F2) Compatibility "Loop"	✓	✓	✓	✗
Heterophily Type	Easy	Challenging	Challenging	Most challenging
Agg. Hom. h_{agg}	1.00	0.90	1.00	0.41
Intra-CNS (min/median/max)	0.97/0.99/1.00	0.79/0.82/0.84	0.78/0.80/0.80	0.62/0.63/0.64
Inter-CNS (min/median/max)	0.00/0.08/0.52	0.00/0.31/0.57	0.30/0.39/0.47	0.53/0.57/0.60
H_2GCN-2	99.26 ± 0.28	88.45 ± 1.26	87.98 ± 1.49	68.95 ± 1.88
H_2GCN-1	93.48 ± 0.93	80.85 ± 1.69	82.40 ± 1.77	66.82 ± 2.13
GCN	100.00 ± 0.00	65.10 ± 1.80	59.26 ± 2.17	27.27 ± 1.72
MLP	59.16 ± 0.52	58.20 ± 2.05	64.16 ± 1.61	63.84 ± 2.17

(1) On `necessity-cora`, with no low-degree nodes and the simpler "loop"-style compatibility matrices (Figure 8(a)-(b)), models like GCN and H_2GCN-2 can achieve near-perfect accuracy.

(2) On `syn-cora-loop` variants, where we keep the "loop"-style compatibility matrices but modify the degree distributions to follow a power law, we observe 34.90% to 40.74% decrease in accuracy for GCN, which falls below the accuracy of H_2GCN. As we discussed in §4.2.2, this heterophilous case is challenging; this is also confirmed by the performance drop for the graph-aware methods, including H_2GCN.

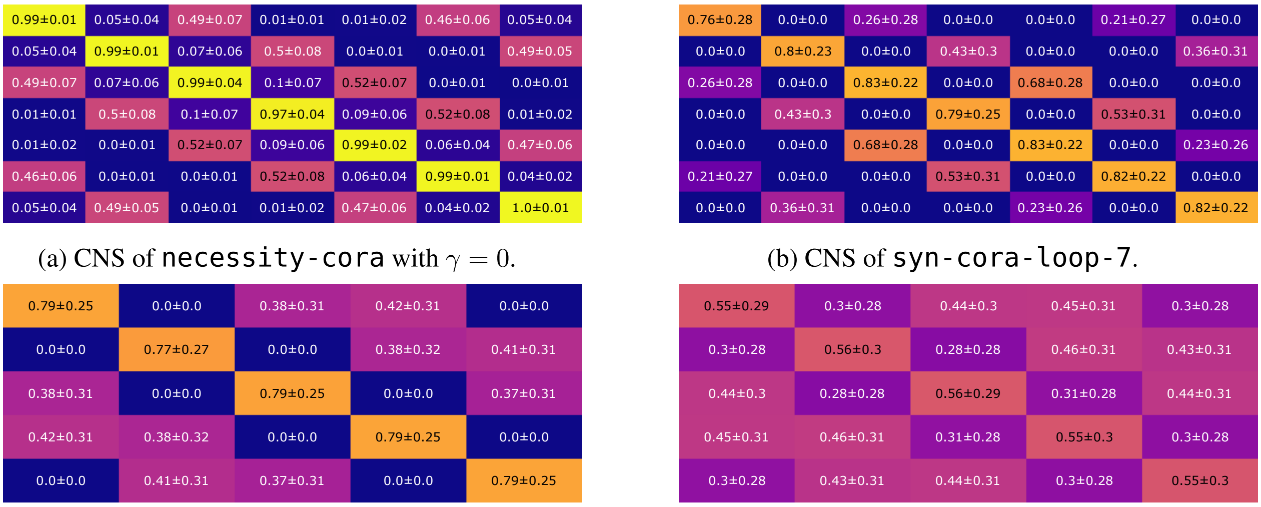
(3) On `syn-cora`, which further strips the "loop"-style compatibility matrices for heterophilous connections and has more complex connectivity patterns across different classes, the performance of GCN further decreases by 31.99% and falls much below the performance of the graph-agnostic MLP in this case; though the accuracy of H_2GCN variants also decreases significantly in this challenging case, they still outperform MLP in this case.

NLD distinguishability & graph properties. The significant changes in the accuracy of GCN can also be explained by the changes in the distinguishability of NLDs caused by different graph properties. In Table 1, we report the Class Neighborhood Similarity (CNS) and Graph Aggregation Homophily h_{agg} for the all synthetic graphs (as defined in §4.1, where we consider self-loops in accordance with GCN aggregation). We also visualize the CNS and its standard deviation (following Eq. equation 5) between pairs of classes on each dataset in Figure 9.

Based on the intra-class and inter-CNS in Table 1, we observe that:

(1) With the presence of low-degree nodes, the `syn-cora-loop` variants have reduced intra-CNS with higher variances compared to `necessity-cora`, while the inter-CNS also increases, though they share similar "loop"-style compatibility matrices;

(2) The removal of the "loop" pattern in the compatibility matrices of `syn-cora` further reduces the intra-CNS to a level similar to the inter-CNS, which leads to weak distinguishability of the neighborhood label distributions



0.99±0.01	0.05±0.04	0.49±0.07	0.01±0.01	0.01±0.02	0.46±0.06	0.05±0.04
0.05±0.04	0.99±0.01	0.07±0.06	0.5±0.08	0.0±0.01	0.0±0.01	0.49±0.05
0.49±0.07	0.07±0.06	0.99±0.04	0.1±0.07	0.52±0.07	0.0±0.01	0.0±0.01
0.01±0.01	0.5±0.08	0.1±0.07	0.97±0.04	0.09±0.06	0.52±0.08	0.01±0.02
0.01±0.02	0.0±0.01	0.52±0.07	0.09±0.06	0.99±0.02	0.06±0.04	0.47±0.06
0.46±0.06	0.0±0.01	0.0±0.01	0.52±0.08	0.06±0.04	0.99±0.01	0.04±0.02
0.05±0.04	0.49±0.05	0.0±0.01	0.01±0.02	0.47±0.06	0.04±0.02	1.0±0.01

(a) CNS of `necessity-cora` with $\gamma = 0$.

0.76±0.28	0.0±0.0	0.26±0.28	0.0±0.0	0.0±0.0	0.21±0.27	0.0±0.0
0.0±0.0	0.8±0.23	0.0±0.0	0.43±0.3	0.0±0.0	0.0±0.0	0.36±0.31
0.26±0.28	0.0±0.0	0.83±0.22	0.0±0.0	0.68±0.28	0.0±0.0	0.0±0.0
0.0±0.0	0.43±0.3	0.0±0.0	0.79±0.25	0.0±0.0	0.53±0.31	0.0±0.0
0.0±0.0	0.0±0.0	0.68±0.28	0.0±0.0	0.83±0.22	0.0±0.0	0.23±0.26
0.21±0.27	0.0±0.0	0.0±0.0	0.53±0.31	0.0±0.0	0.82±0.22	0.0±0.0
0.0±0.0	0.36±0.31	0.0±0.0	0.0±0.0	0.23±0.26	0.0±0.0	0.82±0.22

(b) CNS of `syn-cora-loop-7`.

0.79±0.25	0.0±0.0	0.38±0.31	0.42±0.31	0.0±0.0
0.0±0.0	0.77±0.27	0.0±0.0	0.38±0.32	0.41±0.31
0.38±0.31	0.0±0.0	0.79±0.25	0.0±0.0	0.37±0.31
0.42±0.31	0.38±0.32	0.0±0.0	0.79±0.25	0.0±0.0
0.0±0.0	0.41±0.31	0.37±0.31	0.0±0.0	0.79±0.25

(c) CNS of `syn-cora-loop-5`.

0.55±0.29	0.3±0.28	0.44±0.3	0.45±0.31	0.3±0.28
0.3±0.28	0.56±0.3	0.28±0.28	0.46±0.31	0.43±0.31
0.44±0.3	0.28±0.28	0.56±0.29	0.31±0.28	0.44±0.31
0.45±0.31	0.46±0.31	0.31±0.28	0.55±0.3	0.3±0.28
0.3±0.28	0.43±0.31	0.44±0.31	0.3±0.28	0.55±0.3

(d) CNS of `syn-cora`.

Figure 9: Class neighborhood similarities (CNS) of the synthetic datasets in Table 1.

between nodes from different classes. These observations explain the decrease of GCN performance observed in our experiments, and show how that the distinguishability of the neighborhood label distributions can depend on other properties like degree distributions and class compatibility matrices in the underlying graphs.

Additionally, we note that while the Aggregation Homophily h_{agg} is a good indicator of the performance of GCN on `necessity-cora` and `syn-cora`, it does not correlate well with the performance changes of GCN on `syn-cora-loop` variants. While h_{agg} is defined as the ratio of nodes with NLD more similar to nodes from the same class than nodes from other classes (Eq. equation 6), it does not measure the level of similarity between the NLDs of nodes from the same or different classes as CNS does. Therefore, we believe that CNS is a more accurate and comprehensive indicator of the complexity of heterophily, as Table 1 shows.

Effectiveness of heterophilous GNN designs. With H$_2$GCN as an example that incorporates three heterophilous designs (D1), (D2) and (D3) (discussed in §3.1), we observe that these heterophilous designs can largely improve the performance of GNNs compared to GCNs even when the heterophilous connections do not have the ideal distinguishability in the NLDs as in the `necessity-cora` ($\gamma = 0$) case. When the distinguishability of NLDs among different classes is low (i.e., when intra-CNS is low and inter-CNS is high), the H$_2$GCN variants largely outperform GCN under heterophilous settings. While our experiments focus more on the effects of graph properties to NLD distinguishability and GNN performance and only considered H$_2$GCN as an example for heterophilous GNNs, more comprehensive experiments have been conducted in recent works [30, 46, 21] which support the effectiveness of these heterophilous GNN designs.

5 Conclusion & Future Directions

In this work, we revisited the debate of whether heterophily is a challenge for GNNs. We first reviewed representative architectural designs that have been proposed in the literature for improving the performance of GNNs on heterophilous data, and then discussed the connections with other objectives of GNN research, such as robustness, fairness, and reducing oversmoothing. To address the debate and reconcile seemingly contradictory statements in the literature, we conducted an extensive empirical analysis that aimed to provide a better understanding of when heterophily is challenging and when it does not pose significant additional challenges compared to handling graphs with homophily. We also considered recently proposed measures for quantifying the complexity of heterophily and evaluated their effectiveness across synthetic datasets based on different generation processes. Our analysis revealed two key factors that increase the complexity of heterophily: (F1) the presence of low-degree nodes, and (F2) the complexity of the class compatibility matrices of the underlying graphs. These factors present unique challenges for GNNs under heterophilous settings, and necessitate architectural designs that can improve the performance of GNNs. We hope that our review and empirical analysis will inspire future research on better understanding the unique challenges of heterophily in GNNs and developing more effective GNN models that can handle well both graphs with homophily and heterophily (of variable complexity).

Future Directions. There are many promising research directions towards understanding the unique challenges that heterophily poses to GNN models. Next we discuss some representative open problems:

- **Beyond node classification and global homophily.** Most existing works on GNNs and heterophily (including the ones we review in this work) focus on node classification, where heterophily can be defined and measured with respect to the agreement of class labels for connected nodes. However, many important applications on graphs, such as recommendation systems, query matching, and the prediction of molecular properties, are based on other learning tasks such as link prediction and graph classification. It is thus important to understand the effects of heterophily on these tasks and inform the design of tailored GNN models that can handle heterophily. While few works have discussed heterophily in the settings of link prediction [53, 56] and graph classification [48], their definition of heterophily is still based on node class labels, which are often not available for these tasks. Measuring homophily in the absence of node is an interesting problem for these graph learning tasks. Moreover, going beyond a global perspective and exploring the effect of different mixing patterns across different neighborhoods is an important research direction that has started to gain reaction [23].

- **More datasets & applications.** Despite recent efforts in collecting and introducing new datasets that address the drawbacks of existing heterophilous ones [21, 30], we believe that the call for more heterophilous graph datasets and applications is still important and timely. Many existing works on GNN and heterophily rely on the six heterophilous graph datasets which were first adopted by Pei et al. [29]. While these datasets were useful during the early stages of research on GNNs and heterophily, multiple works [58, 21, 30] have pointed out the drawbacks of these commonly adopted benchmark datasets, namely their small sizes, artificial class labels, imbalanced class sizes, unusual network structure, and even leakage of test nodes in the training set. In light of these, Lim et al. [21] and Platonov et al. [30] proposed a set of mid- to large-scale social, citation and web networks with more diverse node features and realistic class labels, but these datasets have yet to gain widespread adoption, and the relationship between the (heterophilous) links and the class labels is often ambiguous (e.g., predicting product ratings on Amazon based on edges connecting frequently bought items). Thus, we believe that there is still a need for datasets that have naturally-occurring heterophilous connections that align better with defined node class labels. In terms of application domains, it would be useful to go beyond social, citation, and webpage networks and introduce benchmarks that capture molecular or protein structures, which could also aid the investigation of more graph learning tasks that we discuss above.

- **Connections between heterophily & heterogeneity.** Although we highlighted in §2 that heterophily and heterogeneity are two distinct concepts that should not be confused, heterogeneity may introduce unique forms and challenges of heterophily that are worth investigating: connected nodes of different types could imply dissimilarity in their embeddings, resembling the concept of heterophily, while the level of homophily may also vary across different local mixing patterns. As a result, GNN models operating on heterogeneous graphs have already adopted designs similar to those tailored for heterophily, such as the separation of ego- and neighbor-embeddings and the use of type-specific kernels in message passing [31], in order to address the challenges of heterogeneity. Moreover, recently, Guo et al. [11] also discussed how enhancing the homophily level in the meta-paths of heterogeneous graphs can improve GNN performance. Therefore, we believe that further research on the connections between heterophily and heterogeneity can help better understand the connections between the methodologies and findings of these two settings, which in turn may lead to the development of more effective GNNs for both scenarios.

Acknowledgements

We thank Yao Ma, Xiaorui Liu, Neil Shah and Jiliang Tang for sharing the synthetic datasets generated in [26] and the constructive discussions regarding the takeaways of our analysis in §4. This material is based upon work supported by the National Science Foundation under IIS 2212143, CAREER Grant No. IIS 1845491, IIS 1452425, an Adobe Digital Experience research faculty award, an Amazon faculty award, a Google faculty award, and AWS Cloud Credits for Research. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Quadro P6000 GPU used for this research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or other funding parties.

References

[1] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. 2019. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In international conference on machine learning. PMLR, 21–29.

[2] Kristen M Altenburger and Johan Ugander. 2018. Monophily in social networks introduces similarity among friends-of-friends. Nature human behaviour 2, 4 (2018), 284–290.

[3] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. 2021. Beyond Low-frequency Information in Graph Convolutional Networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35. 3950–3957.

[4] Cristian Bodnar, Francesco Di Giovanni, Benjamin Chamberlain, Pietro Lio, and Michael Bronstein. 2022. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns. Advances in Neural Information Processing Systems 35 (2022), 18527–18541.

[5] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. 2020. Simple and deep graph convolutional networks. In ICML.

[6] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. 2021. Adaptive Universal Generalized PageRank Graph Neural Network. In International Conference on Learning Representations.

[7] Alex Chin, Yatong Chen, Kristen M. Altenburger, and Johan Ugander. 2019. Decoupled smoothing on graphs. In Proceedings of the 2019 World Wide Web Conference. 263–272.

[8] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. 2018. Adversarial attack on graph structured data. In ICML.

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In Advances in Neural Information Processing Systems (NeurIPS). 3844–3852.

[10] Lun Du, Yujie Li, Yifan Zhang, Xianglong Liu, and Meng Wang. 2022. GBK-GNN: Gated Bi-Kernel Graph Neural Networks for Modeling Both Homophily and Heterophily. In Proceedings of The Web Conference 2022.

[11] Jiayan Guo, Lun Du, Wendong Bi, Qiang Fu, Xiaojun Ma, Xu Chen, Shi Han, Dongmei Zhang, and Yan Zhang. 2023. Homophily-oriented Heterogeneous Graph Rewiring. In Proceedings of the ACM Web Conference 2023. 511–522.

[12] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In Advances in neural information processing systems (NeurIPS). 1024–1034.

[13] Di Jin, Zhizhi Yu, Cuiying Huo, Rui Wang, Xiao Wang, Dongxiao He, and Jiawei Han. 2021. Universal graph convolutional networks. Advances in Neural Information Processing Systems 34 (2021), 10654–10664.

[14] Wei Jin, Tyler Derr, Yiqi Wang, Yao Ma, Zitao Liu, and Jiliang Tang. 2021. Node similarity preserving graph convolutional networks. In Proceedings of the 14th ACM international conference on web search and data mining. 148–156.

[15] Wei Jin, Yaxing Li, Han Xu, Yiqi Wang, Shuiwang Ji, Charu Aggarwal, and Jiliang Tang. 2021. Adversarial Attacks and Defenses on Graphs: A Review, A Tool and Empirical Studies. SIGKDD Explor. Newsl. 22, 2 (Jan 2021), 19–34.

[16] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In International Conference on Learning Representations (ICLR).

[17] Jia Li, Honglei Zhang, Zhichao Han, Yu Rong, Hong Cheng, and Junzhou Huang. 2020. Adversarial attack on community detection by hiding individuals. In WebConf.

[18] Peizhao Li, Yifei Wang, Han Zhao, Pengyu Hong, and Hongfu Liu. 2021. On dyadic fairness: Exploring and mitigating bias in graph connections. In International Conference on Learning Representations.

[19] Qimai Li, Zhichao Han, and Xiaoming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. AAAI (2018).

[20] Yujie Li, Lun Du, Yifan Zhang, Xianglong Liu, and Meng Wang. 2021. Jacobi Convolutional Neural Networks. In International Conference on Machine Learning. PMLR, 6219–6228.

[21] Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser-Nam Lim. 2021. Large Scale Learning on Non-Homophilous Graphs: New Benchmarks and Strong Simple Methods. In Advances in Neural Information Processing Systems.

[22] Meng Liu, Zhengyang Wang, and Shuiwang Ji. 2021. Non-local graph neural networks. IEEE transactions on pattern analysis and machine intelligence 44, 12 (2021), 10270–10276.

[23] Donald Loveland, Jiong Zhu, Mark Heimann, Ben Fish, Michael T Schaub, and Danai Koutra. 2022. On graph neural network fairness in the presence of heterophilous neighborhoods. In the 8th International Workshop on Deep Learning on Graphs (DLG-KDD'22).

[24] Sitao Luan, Yuchen Zhang, Ziqi Wang, Yujie Li, Yifan Zhang, Xinyu Zhang, Zhiyuan Liu, and Maosong Sun. 2022. Revisiting Heterophily For Graph Neural Networks. In Advances in Neural Information Processing Systems.

[25] Jiaqi Ma, Shuangrui Ding, and Qiaozhu Mei. 2020. Towards More Practical Adversarial Attacks on Graph Neural Networks. In NeurIPS.

[26] Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. 2022. Is Homophily a Necessity for Graph Neural Networks?. In International Conference on Learning Representations. https://openreview.net/forum?id=ucASPPD9GKN

[27] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a Feather: Homophily in Social Networks. Annual Review of Sociology 27, 1 (2001), 415–444.

[28] Shashank Pandit, Duen Horng Chau, Samuel Wang, and Christos Faloutsos. 2007. NetProbe: A Fast and Scalable System for Fraud Detection in Online Auction Networks. In Proceedings of the 16th international conference on World Wide Web. ACM, 201–210.

[29] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. 2020. Geom-GCN: Geometric Graph Convolutional Networks. In International Conference on Learning Representations (ICLR). https://openreview.net/forum?id=S1e2agrFvS

[30] Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. 2023. A critical look at evaluation of GNNs under heterophily: Are we really making progress?. In International Conference on Learning Representations.

[31] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15. Springer, 593–607.

[32] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. 2021. Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21). International Joint Conferences on Artificial Intelligence Organization, 1548–1554.

[33] Yunchong Song, Chenghu Zhou, Xinbing Wang, and Zhouhan Lin. 2023. Ordered GNN: Ordering Message Passing to Deal with Heterophily and Over-smoothing. In The Eleventh International Conference on Learning Representations. https://openreview.net/forum?id=wKPmPBHSnT6

[34] Indro Spinelli, Simone Scardapane, Amir Hussain, and Aurelio Uncini. 2021. Fairdrop: Biased edge dropout for enhancing fairness in graph representation learning. IEEE Transactions on Artificial Intelligence 3, 3 (2021), 344–354.

[35] Yizhou Sun and Jiawei Han. 2012. Mining Heterogeneous Information Networks: Principles and Methodologies. Morgan & Claypool Publishers.

[36] Susheel Suresh, Vinith Budde, Jennifer Neville, Pan Li, and Jianzhu Ma. 2021. Breaking the Limit of Graph Neural Networks by Improving the Assortativity of Graphs with Local Mixing Patterns. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 1541–1551.

[37] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. International Conference on Learning Representations (ICLR) (2018). https://openreview.net/forum?id=rJXMpikCZ

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

[39] Yu Wang and Tyler Derr. 2021. Tree decomposed graph neural network. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2040–2049.

[40] Yu Wang, Yuying Zhao, Yushun Dong, Huiyuan Chen, Jundong Li, and Tyler Derr. 2022. Improving fairness in graph neural networks via mitigating sensitive attribute leakage. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 1938–1948.

[41] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In International conference on machine learning. PMLR, 6861–6871.

[42] Huijun Wu, Chen Wang, Yuriy Tyshetskiy, Andrew Docherty, Kai Lu, and Liming Zhu. 2019. Adversarial Examples for Graph Data: Deep Insights into Attack and Defense. In IJCAI. https://doi.org/10.24963/ijcai.2019/669

[43] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems 32, 1 (2020), 4–24.

[44] Kaidi Xu, Hongge Chen, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Mingyi Hong, and Xue Lin. 2019. Topology attack and defense for graph neural networks: an optimization perspective. In IJCAI.

[45] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation Learning on Graphs with Jumping Knowledge Networks. In Proceedings of the 35th International Conference on Machine Learning, ICML, Vol. 80. PMLR, 5449–5458.

[46] Yujun Yan, Milad Hashemi, Kevin Swersky, Yaoqing Yang, and Danai Koutra. 2022. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. In 2022 IEEE International Conference on Data Mining (ICDM). IEEE, 1287–1292.

[47] Zhiqing Yang, Yiqi Wang, Carl Yang, and Yuandong Tian. 2022. Graph Pointer Neural Networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36. 8832–8839.

[48] Wei Ye, Jiayi Yang, Sourav Medya, and Ambuj Singh. 2022. Incorporating Heterophily into Graph Neural Networks for Graph Classification. arXiv preprint arXiv:2203.07678 (2022).

[49] Yifan Zhang, Yuxin Chen, Nan Du, Xianglong Liu, and Meng Wang. 2021. Beyond Low-frequency Information in Graph Convolutional Networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35. 10842–10850.

[50] Yifan Zhang, Yuxin Chen, Nan Du, Xianglong Liu, and Meng Wang. 2021. Improving Graph Neural Networks with Simple Architecture Design. In arXiv preprint arXiv:2105.07634.

[51] Z. Zhang, P. Cui, and W. Zhu. 2020. Deep Learning on Graphs: A Survey. IEEE Transactions on Knowledge and Data Engineering (TKDE) (2020).

[52] Xin Zheng, Yixin Liu, Shirui Pan, Miao Zhang, Di Jin, and Philip S Yu. 2022. Graph neural networks for graphs with heterophily: A survey. arXiv preprint arXiv:2202.07082 (2022).

[53] Shijie Zhou, Zhimeng Guo, Charu Aggarwal, Xiang Zhang, and Suhang Wang. 2022. Link Prediction on Heterophilic Graphs via Disentangled Representation Learning. arXiv preprint arXiv:2208.01820 (2022).

[54] Zhengyang Zhou, qihe huang, Gengyu Lin, Kuo Yang, LEI BAI, and Yang Wang. 2023. GReTo: Remedying dynamic graph topology-task discordance via target homophily. In The Eleventh International Conference on Learning Representations. https://openreview.net/forum?id=8duT3mi_5n

[55] Jiong Zhu, Junchen Jin, Donald Loveland, Michael T Schaub, and Danai Koutra. 2022. How does Heterophily Impact the Robustness of Graph Neural Networks? Theoretical Connections and Practical Implications. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22).

[56] Jiong Zhu, Aishwarya Reganti, Edward Huang, Charles Dickens, Nikhil Rao, Karthik Subbian, and Danai Koutra. 2023. Simplifying Distributed Neural Network Training on Massive Graphs: Randomized Partitions Improve Model Aggregation. arXiv preprint arXiv:2305.09887 (2023).

[57] Jiong Zhu, Ryan A Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K Ahmed, and Danai Koutra. 2021. Graph Neural Networks with Heterophily. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35. 11168–11176.

[58] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. 2020. Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs. Advances in Neural Information Processing Systems 33 (2020).

[59] Zhihao Zhu, Chenwang Wu, Min Zhou, Hao Liao, Defu Lian, and Enhong Chen. 2023. Resisting Graph Adversarial Attack via Cooperative Homophilous Augmentation. In Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part III. Springer, 251–268.

[60] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. 2018. Adversarial attacks on neural networks for graph data. In KDD.