Refining Latent Homophilic Structures over Heterophilic Graphs for Robust Graph Convolution Networks

Chenyang Qiu[1], Guoshun Nan[1*], Tianyu Xiong[1], Wendi Deng[1], Di Wang[1], Zhiyang Teng[2], Lijuan Sun[1], Qimei Cui[1], Xiaofeng Tao[1]

[1]Beijing University of Posts and Telecommunications, China
[2]Nanyang Technological University, Singapore
{cyqiu, nanguo2021, tyxiong, dengwendi, wdwdwd, sunlijuan, cuiqimei, taoxf}@bupt.edu.cn, chihyangteng@gmail.com

Abstract

Graph convolution networks (GCNs) are extensively utilized in various graph tasks to mine knowledge from spatial data. Our study marks the pioneering attempt to quantitatively investigate the GCN robustness over omnipresent heterophilic graphs for node classification. We uncover that the predominant vulnerability is caused by the structural out-of-distribution (OOD) issue. This finding motivates us to present a novel method that aims to harden GCNs by automatically learning **L**atent **H**omophilic **S**tructures over heterophilic graphs. We term such a methodology as **LHS**. To elaborate, our initial step involves learning a latent structure by employing a novel self-expressive technique based on multi-node interactions. Subsequently, the structure is refined using a pairwisely constrained dual-view contrastive learning approach. We iteratively perform the above procedure, enabling a GCN model to aggregate information in a homophilic way on heterophilic graphs. Armed with such an adaptable structure, we can properly mitigate the structural OOD threats over heterophilic graphs. Experiments on various benchmarks show the effectiveness of the proposed LHS approach for robust GCNs.

Introduction

Graph-structured spatial data, such as social networks (Qiu et al. 2022) and molecular graphs, is ubiquitous in numerous real-world applications (Li et al. 2022). Graph convolution networks (GCNs) (Kipf and Welling 2017), following a neighborhood aggregation scheme, are well-suited to handle these relational and non-Euclidean graph structures, and have been widely applied in various graph tasks, including node classification and recommender systems. Recently, there has been a surge in GCN approaches for challenging heterophilic graphs (Zhu et al. 2020), where most neighboring nodes have different labels or features. These methods can be divided into two categories: 1) Multi-hop-based approaches (Abu-El-Haija et al. 2019; Jin et al. 2021a; Wang and Derr 2021); 2) Ranking-based approaches (Liu, Wang, and Ji 2021; Yuan and Ji 2021; Yang et al. 2022). The former group learns node representations based on multi-hop aggregations, while the latter performs selective node aggregations by a sorting mechanism. These GCN methods continue to advance the state-of-the-art performance for node classification and have enabled various downstream applications (Lin, Lan, and Li 2021; Qiu et al. 2023).

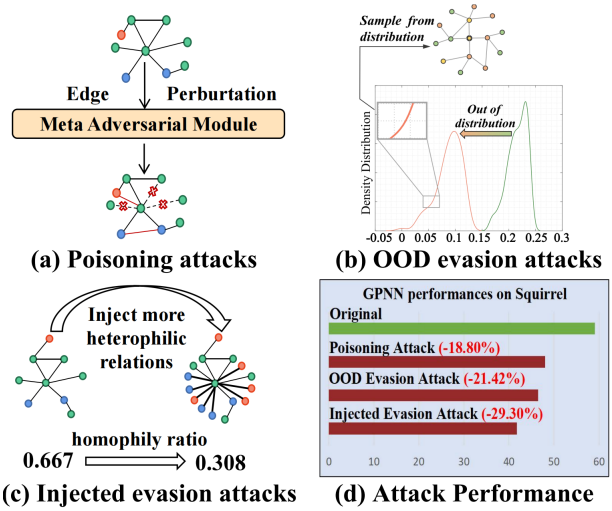


Figure 1: Illustration for the vulnerability of existing GPNN under various threats on Squirrel, including a poisoning attack, and another two adapted from evasion attacks. The subfigures (a), (b), and (c) depict how we generate the above three attacks, and (d) reports the significant performance degradation under each attack.

Despite the significant success of the current GCN methods on heterophilic graphs, these approaches are **extremely vulnerable to malicious threats** that aim to distort the structure of the target graph during testing. We conduct experiments to attack the state-of-the-art GPNN (Yang et al. 2022) method, which was trained on the popular Squirrel (Pei et al. 2020) benchmark for heterophilic graphs, using samples created by various attacks. Fig.1 demonstrates that the accuracy of node classification can be greatly reduced under three different types of destructive attacks, including a well-known poisoning attack (Jin et al. 2020), and two attacks adapted from evasion attacks (Biggio et al. 2013; Zhang et al. 2016). Specifically, as shown in Fig.1 (a), the poisoning attack produces adversarial structural perturba-

*Guoshun Nan is the corresponding author.

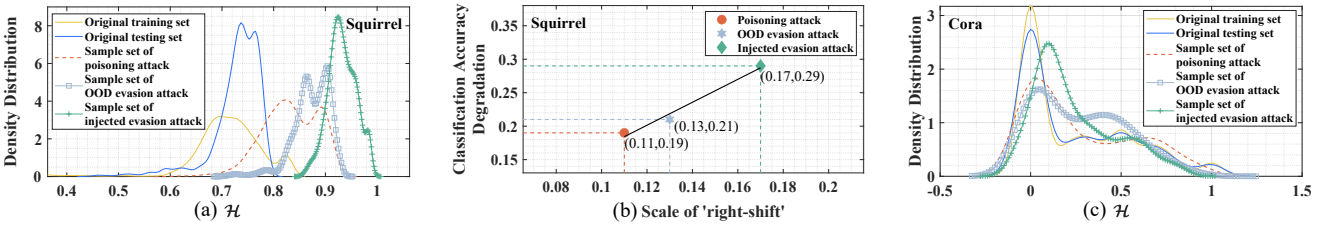


Figure 2: Illustration of $\mathcal{H}$ distributions and the "right-shift". (a) $\mathcal{H}$ distributions of various data over Squirrel, including crafted sample sets of three attacks. (b) correlation between the "right-shift" of $\mathcal{H}$ distributions and node classification degradation. (c) $\mathcal{H}$ distributions of various data, including the original train and test set of a homophilic dataset Cora, and crafted sample sets of three attacks.

tions to the edges of the graph, fooling GPNN to make incorrect predictions. The proposed two evasion-based attacks are referred to as "OOD evasion attacks" and "injected evasion attacks", respectively. Fig.1 (b) and Fig.1 (c) demonstrate how the sample sets for these two attacks are created. The first generates a graph with a node distribution that is vastly different from that of the target testing set, while the second manipulates the target graph by injecting more heterophilic edges. Under these three attacks, Fig.1 (d) shows that classification accuracy of GPNN is sharply decreased by 18.90%, 21.42%, and 29.30%, respectively.

To analyze the reasons why GCN methods are fragile on heterophilic graphs, we further depict the $\mathcal{H}$ distributions (Zheng et al. 2022) of the crafted data from the aforementioned three attacks, as well as the distributions of the original train and test sets of Squirrel in Fig.2 (a). Here $\mathcal{H}$ represents the node-level heterophily, which is the proportion of a node's neighbors that have a different class[1]. Fig.2 (a) demonstrates that the distributions of three attack samples are all located to the right of the training set, with the most destructive sample for the GPNN method being the furthest to the right. This observation led us to investigate the correlation between the "right-shift" of the $\mathcal{H}$ distribution relative to the train set and the vulnerability of GCN approaches. This correlation is visualized in Fig.2 (b) and it is shown that the scale of "right-shift" is strongly proportional to the degradation of node classification performance. We refer to this phenomenon as "**structural out-of-distribution (OOD)**" in GCN methods for graphs of spatial data.

To investigate the underlying cause of the aforementioned structural OOD, we attacked another GPNN model that was trained on the homophilic graph Cora (Yang, Cohen, and Salakhudinov 2016) and depicted the resulting $\mathcal{H}$ distributions in Fig.2 (c). Interestingly, the shifts of the three attacks relative to the training set of Cora are very small. This minor "right-shift" enables the GPNN model trained on Cora to be more robust. We attribute this to the strong homophily present in the Cora dataset and believe that more homophily will result in less "right-shift" under attacks, even for heterophilic graphs, and hence alleviate the structural OOD.

In light of the above discussion, a critical question arises: "*How can a GCN model automatically learn an appropriate* ***homophilic structure*** *over* ***heterophilic graphs*** *to reduce the scale of "right-shift" in $\mathcal{H}$ distributions?*" This could help to make the model more resistant to malicious attacks on heterophilic graphs. Achieving this goal is challenging. Despite the success of many structure learning-related methods (Jin et al. 2021b,a; He et al. 2022), they also tend to strengthen the heterophily or only focus on the *local* relations between two nodes rather than considering the *global* connections. These methods still suffer from vulnerability issues under attacks (as seen in Figure 4 and Table 1), and they are hardly able to address the challenge.

We address the above challenging question with a novel method called LHS. The key components of the proposed LHS are: 1) a self-expressive generator that automatically induces a latent homophilic structure over heterophilic graphs via multi-node interactions, and 2) a dual-view contrastive learner that refines the latent structure in a self-supervised manner. LHS iteratively refines this latent structure during the learning process, enabling the model to aggregate information in a homophilic way on heterophilic graphs, thereby reducing the "right-shift" and increasing robustness. It should be noted that the original graph Experiments on five benchmarks of heterophilic graphs show the superiority of our method. We also verify the effectiveness of our LHS on three public homophilic graphs. Additionally, the induced structure can also be applied to other graph tasks such as clustering. Our contributions are as follows:

- We quantitatively analyze the robustness of GCN methods over omnipresent heterophilic graphs for node classification, and reveal that the "right-shift" of $\mathcal{H}_{node}$ distributions is highly proportional to the model's vulnerability, i.e., the structural OOD. To the best of our knowledge, this is the first study in this field.
- We present LHS, a novel method that strengthens GCN against various attacks by learning latent homophilic structures on heterophilic graphs.
- We conduct extensive experiments on various spatial datasets to show the effectiveness of the proposed LHS in mitigating the structural OOD issue.

Related Work

Graph Convolution Networks

There is a line of early studies in graph convolution networks (GCNs) (Kipf and Welling 2016, 2017; Hamilton, Ying, and

[1]A more formal definition is given in Preliminaries Section, and higher $\mathcal{H}$ values indicate a node with strong heterophily.

Leskovec 2017; Veličković et al. 2018). Recent GCN approaches over heterophilic graphs can be grouped into multi-hop-based ones (Abu-El-Haija et al. 2019; Zhu et al. 2020; Jin et al. 2021b; Wang and Derr 2021; Wang et al. 2022b), ranking-based ones (Liu, Wang, and Ji 2021; Wang et al. 2022a; Yang et al. 2022), and the ones using GCN architecture refinement (Bo et al. 2021; Yang et al. 2021; Suresh et al. 2021a; Yan et al. 2021; Luan et al. 2022; Xu et al. 2023; Li, Kim, and Wang 2023; Zheng et al. 2023). These methods have achieved remarkable success in graph node classification. However, robustness is yet to be explicitly considered on challenging heterophilic graphs.

Robust Graph Convolution Networks

Recently we have witnessed a surge in the robustness of GCN on heterophilic graphs. These methods can be categorized into the structure learning-based ones (Jin et al. 2020, 2021a; He et al. 2022; Zhu et al. 2022; Liu et al. 2023), and the ones based on adversarial training (Dai et al. 2018; Zhu et al. 2019; Zhang, Zhang, and Cheng 2020; Zhang and Zitnik 2020; Suresh et al. 2021b). The most related to our work is ProGNN (Jin et al. 2020) which explores the low-rank and sparsity of the graph structure, and SimP-GCN (Jin et al. 2021b) which relies on a similarity preservation scheme for structure learning. Our work differs from the above methods in two aspects: 1) We focus on the structural OOD issue of GCN approaches over heterophilic graphs. To the best of our knowledge, this problem is largely ignored in previous works. 2) We iteratively refine the latent structure of heterophilic graphs by a novel self-expressive method and a dual-view contrastive learning scheme, enabling a GCN model to effectively aggregate information in a homophilic way on heterophilic graphs.

Preliminaries

We denote the graph as $G = (V, E, \mathbf{X})$, where $V \in \mathbb{R}^{N\times N}$ is the set of nodes, E is the set of edges between nodes, $\mathbf{X}$ is the node features and (V, E) can form the original network structure $\mathbf{A}$. We aim to generate a latent homophilic structure for robust GCN in node classification. For convenience, we give the following edge definition:

Definition 1. *(Positive/Negative Edge) A **positive** edge indicates that the two nodes in the link have the same type, while **negative** one refers to a link that connects two nodes with different types.*

Node-level Heterophily: We use $\mathcal{H}$ to represent the node-level heterophily, which is the proportion of a node's neighbors that have a different class. We refer to (Zheng et al. 2022) to give a formal definition, and it is a fine-grained metric to measure the edge heterophily in a graph.

Definition 2. *Node-level Heterophily ($\mathcal{H}$):*

$$\mathcal{H}(v_i) = \frac{|(v_i, v_j) \mid y(v_i) \neq y(v_j)|}{|\mathcal{E}(v_i)|}, \forall v_i, v_j \in V, \qquad (1)$$

where $(v_i, v_j) \in \mathcal{E}(v_i)$, $\mathcal{E}(v_i)$ is the edge set of v_i, $y(v_i)$ is the node class of v_i, and $|\cdot|$ represents the number of edges. The nodes with strong heterophily have higher $\mathcal{H}$ (closer to 1), whereas nodes with strong homophily have smaller $\mathcal{H}$ (closer to 0). It also provides an edge distribution sampling set for quantitative analysis over heterophilic graphs.

OOD Formulation: We rigorously formulate the ego-graph edge distribution by utilizing the proposed node-level heterophily, and the formulation further enables the multi-layer edge distribution analyses. The 'right-shift' phenomenon found in the heterophilic graphs also motivates us to propose latent homophilic structure refinement. Theoretical analysis from a spectral-domain view is given in Appendix 1 [2] to further elaborate the rationale of the proposed LHS.

Edge Distribution Formulation: Given a random node $v_i \in V$, we define v_i's k-hop neighbors as $N_{v_i}(k)$ (where k is an arbitrary positive integer) and the nodes in $N_{v_i}(k)$ form an ego-graph substructure called $A_{v_i}(k)$, which consists of a local adjacency matrix represented as $A_{v_i}(k) = \{a_{v_i u} \mid u \in N_{v_i}(k)\}$. In this way, we can study the distribution of the k-hop substructure A_v via $p(\mathcal{H} \mid A_{v_i}(k)) = p(\mathcal{H} \mid A_{v_i}(1)A_{v_i}(2)...A_{v_i}(k))$. It's worth noting that the ego-graph can be seen as a Markov blanket for the centered node v_i, meaning that the conditional distribution $p(\mathcal{H} \mid A_{v_i}(k))$ can be decomposed as a product of independent and identical marginal distributions $p(\mathcal{H} \mid A_{v_i}(i), i \in k)$ for each of the $A_{v_i}(j), j \leq k$. We also provide more empirical observations about the "right-shift" phenomenons on heterophilic graphs, which are available in Appendix 3.

Methodology

Overview

In this section, we present the proposed LHS. Our goal is to learn an appropriate latent homophilic structure from heterophilic graphs, so as to reduce the scale of "right-shift" in $\mathcal{H}$ distributions. Inspired by the analysis in the Introduction Section that more homophily of a graph can reduce the "right-shift", our latent structure tends to encourage positive edge connections by increasing the edge weights for pairs of nodes with the same type, and suppresses negative edge connections by reducing the edge weight for nodes with different types. Fig. 3 shows the architecture of LHS.

Structure Inducer

The proposed structure inducer involves a self-expressive generator and a dual-view contrastive learner.

Self-Expressive Generator. Our proposed self-expressive generator produces a latent homophilic structure over heterophilic graphs in three steps:

Step 1: Capturing multi-node information. Given the node features $\mathbf{X}$, we aim to capture the multi-node feature information by expressing one node feature via a linear or affine combination of other node features. Differently from the existing structure learning method with pair-wise similarity matrix (Jin et al. 2021a), our inducer can generate fine-grained latent structure $S^* \in \mathbb{R}^{N\times N}$ by discovering the global information in low-dimension subspace. Specifically, for $\forall v_i \in V$, we express it by a linear sum of multi-node features $\mathbf{x}_j, v_j \neq v_i$, which can be expressed

[2] Appendices are available in the preprint version.

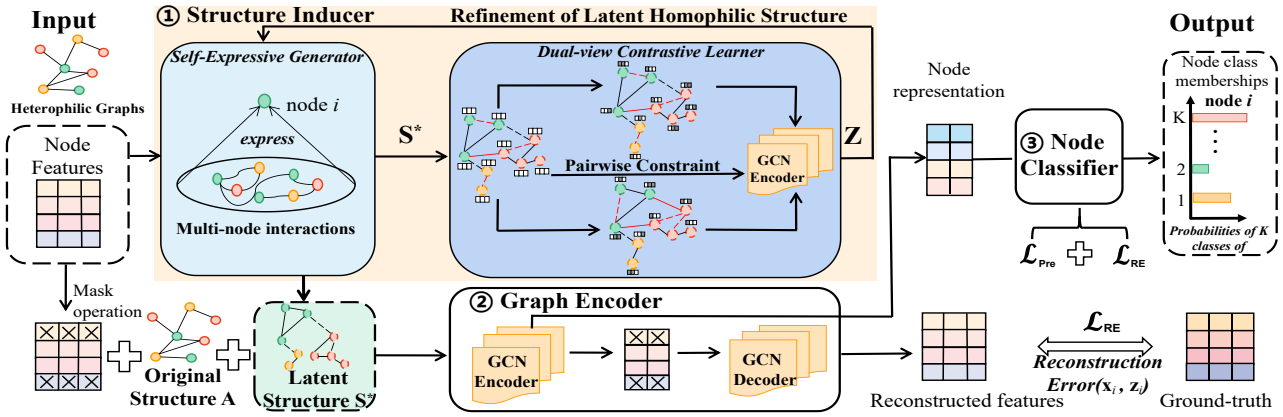


Figure 3: The architecture of the proposed LHS, which consists of three modules: structure inducer, graph encoder, and node classifier. The key ingredient structure inducer involves two components, i.e., the self-expressive generator and dual-view contrastive. The former learns a latent homophilic structure by multi-node interactions and then the latter refines the structure. We iteratively perform such a procedure to learn a better homophilic structure on heterophilic graphs. The refined structure will be fed to the graph encoder for representation aggregation, and finally, the classifier for node classification.

as $\mathbf{x}_i = \sum_{v_j \in V} q_{ij}\mathbf{x}_j$, where q_{ij} is the (i, j) th element of a coefficient matrix Q.

Step 2: Optimizing the generator loss. We use the coefficient matrix Q to generate latent structure. The optimization problem to solve Q can be formulated as follows:

$$\min_{Q} \|Q\|_F \quad s.t. \quad \mathbf{X} = Q\mathbf{X};\ \mathrm{diag(Q)} = 0 \tag{2}$$

where $\|Q\|_F$ is the Frobenius matrix norm (Böttcher and Wenzel 2008) of Q and $\mathrm{diag(Q)}$ denotes the diagonal entries of Q. Eq. 2 optimizes a block-diagonal matrix Q to generate the latent structure S^*. Each block of Q contains the nodes which belong to the same class, thus mitigating the "right-shift" phenomenon. We relax the hard constraint $\mathbf{X} = Q\mathbf{X}$ with a soft constraint $(\mathbf{X} - Q\mathbf{X})$, as the exact reconstruction of $\mathbf{X}$ may be impractical. The relaxation formulation is:

$$\min_{Q} \mathcal{L}_{SE} = \|\mathbf{X} - Q\mathbf{X}\|_F^2 + \lambda_1 \|Q\|_F^2, s.t. \mathrm{diag(Q)} = 0, \tag{3}$$

where λ_1 is a weight hyperparameter of optimization.

Step 3: Generating latent homophilic structure. We construct the latent homophilic structure S^* by $Q + Q^T$, while this structure still has noise and outliers. Therefore, we rely on Algorithm 1 to generate S^*. Specifically, the SVD decomposition in Algorithm 1 aims to filter noisy information during the structure generation. In each iteration, we refine the latent structure S^*. We employ the scalable randomized SVD (Halko, Martinsson, and Tropp 2011) to improve the computation efficiency for large-scale graphs. Details are available in Appendix 2.1.

Dual-view Contrastive Learner. So far we have obtained the latent structure S^*, and in the previous step, we focus on learning S^* based on the node features. To refine such a structure, we further explore the enriched structural information of the graph and propose a novel dual-view contrastive learner. We take four steps for such a refinement.

Step 1: Generating the dual views of latent structure. We denote the graph as $G = (S^*, \mathbf{X})$, where S^* is the learnable latent homophilic structure. Based on G, we generate two graphs G_1 and G_2 via the corruption function (Velickovic et al. 2019) to refine the structure in a self-supervised manner. Specifically, the corruption function randomly removes a small portion of edges from S^* and also randomly masks a fraction of dimensions with zeros in node features $\mathbf{X}$.

Algorithm 1: The generation algorithm of S^*

Input: Coefficient matrix Q, subspaces dimension $K = 4$, rank $r = Kd+1$, where d is the number of node classes.
Output: Latent structure $\mathbf{S}^*$.

1: **Initialization**: $Q' = \frac{1}{2}\left(Q + Q^T\right)$.
2: **Compute**: the r rank **SVD** of Q' via $Q' = U\Sigma V^T$.
3: **Compute**: $L = U\Sigma^{\frac{1}{2}}$ and normalize each row of L.
4: **Update**: $L' \leftarrow$ set the negative values in L to zero.
5: **Obtain**: $S^* = (L' + L'^T) / \|L\|_\infty$, where $s_{ij} \in [0, 1]$.

Step 2: Aggregating information on latent structure. For efficient aggregation on S^*, we devise a truncated threshold GCN to control the sparsity of the structure. For S^*, we introduce a threshold σ to decide if there exists a soft connection with continuous values between two nodes and then form a new structure S^*_σ. Such a way is quite different from the previous hard-coding operations (Liu et al. 2022) that only have 0 or 1, and our S^*_σ can be flexibly applied to various benchmarks. We employ the truncated threshold GCN on three graphs, including G, G_1, and G_2. The proposed truncated threshold GCN on graph G can generate the representations as follows:

$$\begin{aligned} S^* &= \{s^*_{ij} \geq \sigma \mid s^*_{ij} \in S^*\} \\ Z = \mathrm{GCN}(\mathbf{X}, S^*) &= \hat{S}^* \mathrm{ReLU}(\hat{\mathrm{S}}^* \mathbf{X} \mathrm{W}^{(0)}) \mathrm{W}^{(1)} \end{aligned} \tag{4}$$

where ReLU is an activation function, $W^{(0)}$ and $W^{(1)}$ are the trainable weight matrices, $\tilde{S}^* = S^* + I$, $I \in \mathbb{R}^{|V| \times |V|}$ is

the identity matrix and the degree diagonal matrix $\tilde{D}_{ii}$ with $\tilde{D}_{ii} = \sum_{j \in V} \tilde{S}^*_{ij}, \forall i \in V$. We set $\hat{S}^* = \tilde{D}^{-\frac{1}{2}} \tilde{S}^* \tilde{D}^{-\frac{1}{2}}$. $W^{(0)}$ and $W^{(1)}$ are trainable parameter matrices of GCN. Z_1 and Z_2 denote the node embedding matrices for the two views G_1 and G_2, and these node embeddings are generated from the proposed GCN encoder.

Step 3: Sampling the contrastive samples. For a node $v_i \in V$, let us denote the corresponding nodes in G_1 and G_2 as $G_1(v_i)$ and $G_2(v_i)$ respectively. Then we introduce the node-pair sampling rules for the contrastive learning as follows: a) **positive example** is the node pair from the same node of different graph views, that is, $\forall i \in V$, the pair $(G_1(i), G_2(i))$. b) **negative example** is the node pair from the different nodes of the same or different graph views, that is, $\forall i \in V$, $V_{-i} = \{j \in V \mid j \neq i\}$. Both $(G_1(i), G_1(j))$ and $(G_1(i), G_2(j))$ are the negative examples.

Step 4: Optimizing the contrastive loss. In addition to the above dual-view optimization, we also propose a novel pairwise constraint to optimize the original graph view, which can further improve the quality of the learned homophilic structure. Specifically, we sample the node pairs from the training set with labels. The same-class node pairs are positive samples noted as (u, v), while the different-classes node pairs are negative samples noted as (u, v_n), where u, v, and v_n belong to the training node set and $y(u) = y(v)$, $y(u) \neq y(v_n)$. Here $y(\cdot)$ is the node label. We formally propose the loss function as:

$$\begin{aligned} \mathcal{L}_{refine} = \sum_{i \in V} \Bigg[& -\frac{\cos\left(z_{1i}, z_{2i}\right)}{\tau} \\ & + \log \left(\sum_{j \in V_{-i}} e^{\frac{\cos(z_{1i}, z_{1j})}{\tau}} + e^{\frac{\cos(z_{1i}, z_{2j})}{\tau}} \right) \Bigg] \\ & - \lambda_2 [\log\left(\sigma\left(z_u^\top z_v\right)\right) - \log\left(\sigma\left(-z_u^\top z_{v_n}\right)\right)] \end{aligned} \tag{5}$$

where z_{1i} and z_{2i} denote embeddings node i on Z_1 and Z_2 respectively, $z_{(v)}$ denotes embedding of node v on Z, $\cos(\cdot)$ is the cosine similarity between the two embeddings, τ is a temperature parameter, and λ_2 is a hyperparameter of the second term. The first term of Eq. 5 encourages the consistent information between positive samples, while the second term penalizes the inconsistent information between dual views. The last term of Eq. 5 makes sure that the same-class nodes have more similar representations.

Structure Refinement. The node embedding matrix $\mathbf{Z}$ generated by Eq. 5 has incorporated the refined structure and node features. Finally, we feed $\mathbf{Z}$ into the structure inducer again to iteratively refine the structure S^*. Equipped with both the original graph A and the refined one S^*, we use a structure bootstrapping mechanism $S^* \leftarrow \zeta A + (1 - \zeta) S^*$ to update S^* with a slow-moving of A, where ζ is a hyperparameter to balance the information between A and S^*. Specifically, the input graphs with high heterophily will lead to smaller ζ, while the ones with high homophily have larger ζ. By doing so, we can reduce the scale of "right-shift" over heterophilic graphs and thus potentially mitigate the structural OOD issue under malicious attacks as discussed in Fig. 1 in the Introduction Section.

Graph Encoder

Our graph encoder consists of a GCN encoder and a GCN decoder, where the former encodes the masked features, and the latter aims to generate the reconstructed features $\hat{\mathbf{X}}$. We feed the masked node features $\tilde{\mathbf{X}}$ and S^* to the graph encoder. Then we use a scaled cosine error to optimize the encoder as follows:

$$\mathcal{L}_{\mathrm{Re}} = \frac{1}{|\widetilde{\mathcal{V}}|} \sum_{v_i \in \widetilde{\mathcal{V}}} \left(1 - \frac{\mathbf{x}_i^T \hat{\mathbf{x}}_i}{\|\mathbf{x}_i\| \cdot \|\hat{\mathbf{x}}_i\|}\right)^\gamma, \gamma \geq 1 \tag{6}$$

where $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$ are the feature and reconstructed feature of node i, γ is a scale factor.

Classifier and Loss Functions

Finally, our classifier outputs predictions. We generate classification representations via a fully-connected layer $F(\cdot)$, that is $y_{\text{pred}} = \mathrm{softmax}(F(\mathbf{h}_i))$. Then the loss of the classifier can be expressed as $\mathcal{L}_{Pre} = \sum_{i=1}^{N_l} y_i \log y_{\text{pred}\,i}$. We jointly train the graph encoder and classifier with $\mathcal{L}$, which can be expressed as:

$$\mathcal{L} = \mathcal{L}_{Pre} + \beta \mathcal{L}_{Re} \tag{7}$$

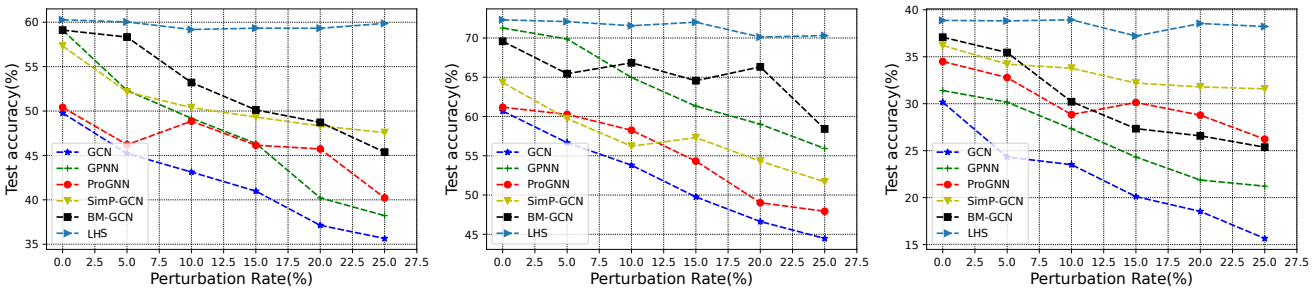
where β is a hyperparameter of loss weight.

Experiments

Datasets, Baselines and Settings

Datasets: We experiment on nine benchmarks. For six heterographic spatial datasets including **Cornell**, **Texas**, **Wisconsin** (Pei et al. 2020), **Chameleon**, **Squirrel** (Rozemberczki, Allen, and Sarkar 2021), nodes are web pages and edges are hyperlinks between these pages, and **Actor** (Tang et al. 2009), nodes are actors and edges denote co-occurrences on same web pages. For the three homophilic datasets including **Cora** and **Citeseer** (Yang, Cohen, and Salakhudinov 2016), nodes refer to articles, and edges are the citations between articles. Due to space limitations, we provide detailed descriptions in Appendix 4.1.

Baselines: We follow the previous works (Jin et al. 2021b; He et al. 2022) to use eleven baselines. We categorize these methods into three groups: **1) multi-hop-based approaches** MixHop (Abu-El-Haija et al. 2019) and H2GCN (Zhu et al. 2020), which mix the multi-hop neighbors for aggregation; **2) ranking-based approaches** NLGNN (Liu, Wang, and Ji 2021), GEOM-GCN (Pei et al. 2020), Node2Seq (Yuan and Ji 2021) and GPNN (Yang et al. 2022) that aim to search on the network structure and then perform selective aggregation; **3) structure learning approaches** ProGNN (Jin et al. 2020), UGCN (Jin et al. 2021a), BM-GCN (He et al. 2022) and GREET (Liu et al. 2023) that automatically learn graph structures for aggregations. Specifically, **ProGNN** preserves the low-rank and sparsity characteristics of the graph structure for robust GCN. **UGCN** and **SimP-GCN** employ a similarity preservation scheme for structure learning on heterophilic graphs and **BM-GCN** employs a selective aggregation on structure via a block-guided strategy. We also compare our model with a recently proposed spectral-based method ALT-GCN (Xu et al. 2023).



(a) Squirrel (b) Chameleon (c)Actor

Figure 4: Comparisons of node classification under a poisoning attack. We repeat three times and report the mean values.

	Wisconsin		Chameleon		Squirrel		Actor	
	OOD	Injected	OOD	Injected	OOD	Injected	OOD	Injected
H2GCN	48.24±2.1	41.53±1.7	47.20±1.5	37.21±2.3	48.27±3.1	36.34±1.6	21.33±2.3	14.17±1.4
GPNN	52.78±0.8	40.21±1.4	54.62±2.0	48.49±1.6	50.23±0.6	40.55±1.3	20.83±2.7	16.28±1.5
UGCN	72.37±2.7	44.58±2.2	57.23±3.1	40.39±2.8	52.45±3.3	41.79±3.2	23.37±1.9	15.57±0.8
SimP-GCN	73.34±2.1	61.43±2.4	61.28±1.6	54.57±2.6	54.34±1.0	54.01±2.3	28.96±1.8	24.31±2.9
BM-GCN	76.58±0.5	69.78±1.3	62.37±2.6	52.58±2.3	57.30±1.0	59.82±0.7	26.17±1.4	18.92±1.9
LHS	**82.31**±0.5	**73.34**±1.1	**67.33**±0.9	**60.10**±2.2	**68.76**±3.1	**62.13**±2.7	**32.23**±2.6	**33.42**±1.8

Table 1: Robustness comparisons in terms of classification accuracy(%) under 2 evasion-based attacks OOD and Injected.

Settings: We implement our method by Pytorch and Pytorch Geometric and use Adam Optimizer on all datasets with the learning rate as 0.001. We configure epochs as 1000 and apply early stopping with the patience of 40. We configure the hidden size as 64 and the batch size as 256. We perform the structure learning for 2 rounds. More detailed hyperparameters are available in Appendix 4.3.

Main Results

Comparisons under Poisoning Attacks. We compare the robustness of our LHS with five baseline approaches under a popular poisoning attack (Jin et al. 2020) on three benchmarks including Squirrel, Chameleon, and Actor. Under various perturbation rates ranging from 0 to 25%, Figure 4 shows that our LHS consistently performs best among all baselines. For example, ours yields higher classification accuracy of up to 20 points compared to ProGNN. These results confirm the superiority of our latent structure learning scheme against poisoning attacks. The existing structure learning methods, including BM-GCN, SimP-GCN, and ProGNN, are also extremely vulnerable under large positioning perturbation rates. Nevertheless, they are better than the other two, showing the promise of structure learning over heterophilic graphs. We also observe that the positioning perturbations, which can significantly degrade the baselines at a large rate (i.e., 25%), have a very slight impact on our method. We attribute such gains to the latent structure that can be resistant to the structural OOD issue discussed in the Introduction Section, which will also be illustrated in the first question of the Discussion Section.

Comparisons under Evasion Attacks. We presented two evasion-based attacks (Zhang et al. 2016), i.e., "OOD evasion attack (OOD)" and "injected evasion attack (Injected)"

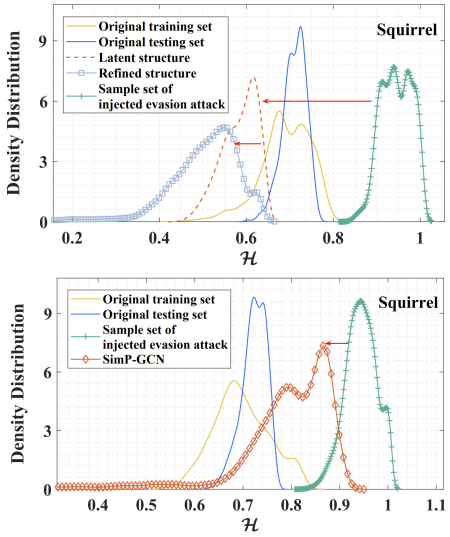


Figure 5: Comparisons of the "right-shift" of $\mathcal{H}_E$.

in Fig. 1 (b) and Fig. 1(c), to craft attack samples with destructive structural perturbations to the edges of the graph. Here we compare our method with five baselines on five heterophilic graphs, and report the results in Table 1. We chose these five baselines because they are representative of different types of GCN and have been widely used in previous studies. For two nodes with different classes, the "Injected" attacks manipulate to inject a connection with a 0.9 probability. We repeat our experiments three times and report the mean and variance values in Table 1. Under the two attacks, Table 1 shows that our method consistently achieves the best among all baselines on five benchmarks. Compared

	Wisconsin	Texas	Cornell	Chameleon	Squirrel	Actor	Cora	Citeseer
MixHop	75.88±4.9	77.84±7.7	73.51±6.2	60.50±2.5	43.80±1.4	32.22±2.3	81.90±0.8	71.40±1.3
H2GCN	86.67±4.6	84.86±6.7	82.16±6.0	57.11±1.6	37.90±2.0	35.86±1.0	87.81±1.3	77.07±1.6
NLGNN	87.30±4.3	85.40±3.8	84.90±5.7	70.10±2.9	59.00±1.2	37.90±1.3	88.50±1.8	76.20±1.6
GEOM-GCN	65.10±6.5	67.84±5.8	60.00±6.5	65.81±1.6	45.49±1.3	31.94±1.0	85.65±1.7	**79.41**±1.7
Node2Seq	60.30±7.0	63.70±6.1	58.70±6.8	69.40±1.6	58.80±1.4	31.40±1.0	-	-
GPNN	86.86±2.6	85.23±6.4	85.14±6.0	71.27±1.8	59.11±1.3	37.08±1.4	-	-
UGCN	69.89±5.2	71.72±6.2	69.77±6.7	54.07±1.7	34.39±1.9	-	84.00±0.9	74.08±1.2
SimP-GCN	85.49±3.5	81.62±6.5	84.05±5.3	-	-	36.20±1.3	-	-
BM-GCN	-	85.13±4.6	-	69.58±2.9	51.41±1.1	-	87.99±1.2	76.13±1.9
GREET	84.90±3.3	**87.00**±4.2	85.10±4.9	63.60±1.2	42.30±1.3	36.60±1.2	83.81±0.9	73.08±0.8
ALT-GCN	76.40±3.9	70.90±4.3	73.90±5.1	65.80±0.9	52.40±0.8	-	81.20±0.5	71.40±0.4
LHS	**88.32**±2.3	86.32±4.5	**85.96**±5.1	**72.31**±1.6	**60.27**±1.2	**38.87**±1.0	**88.71**±0.7	78.53±1.5

Table 2: Comparisons of node classification without any attacks.

with "OOD" attacks, "Injected" attacks are much more constructive as they significantly increase of heterophily of the testing set. Compared to the state-of-the-art structure learning method BM-GCN, our LHS achieves an 11.33 point accuracy under "OOD" attacks. Overall, these results suggest that LHS is more robust against both attacks compared to all the considered baselines. The key to these improved results is the ability of our LHS to perform global searches of the homophilic structure learned by the structure inducer.

Comparisons without attacks. We have shown that our model is more robust than existing methods under various attacks. To further investigate the performance without attacks, we conduct experiments on the five heterophilic graphs and compare ours with baseline approaches. Table 2 shows that the proposed LHS performs best and we attribute this to the information aggregation in a homophilic way on heterophilic graphs. Additionally, we also achieve better or comparable classification results on three benchmarks for homophilic graphs. This suggests that improving homophily for both homophilic and heterophilic graphs benefits node classification, and this also remotely aligns with a previous work (Yan et al. 2021), showing that our method can handle both types of graphs in a unified manner.

Discussion

Can LHS reduce the scale of "right-shift" of $\mathcal{H}$ distributions? We have discussed that the "right-shift" phenomenon, i.e., the structural OOD, is the cause of performance degradation under attacks in the Introduction Section. To answer this question, we visualize how our method reduces the "right-shift" for experiments in Table 1 on Squirrel. Under the "injected evasion attack", Figure 5 shows that our latent structure can greatly move the $\mathcal{H}$ distribution of the attacking sample to the left, thus reducing the "right-shift" (see the red arrow). We also observe that the second round of refinement can further move distribution to the left side, further improving the model's robustness. However, we find that existing SimP-GCN can slightly move the distribution, visually explaining why LHS is more robust than Simp-GCN. This further confirms our hypothesis that reducing "right-shift" can harden the GCN over heterophilic graphs.

Can the learnable homophilic structure be applied to other tasks? To answer this question, we also apply the homophilic structure learned on four graphs, including Wisconsin, Squirrel, Chameleon, and Cora, to the graph clustering task. We use vanilla GCN (Kipf and Welling 2017) and the proposed structure inducer of LHS to develop "GCN + Structure Inducer". Even on the vanilla GCN, Table 3 shows that our "GCN + Structure Inducer" outperforms all other baselines on heterophilic graphs. For example, ours outperforms the Simp-GCN on Squirrel by 2.79 points. We attribute such again to our homophilic structure.

	Wisconsin	Squirrel	Chameleon
SimP-GCN	58.42	38.57	46.44
BMGCN	54.92	40.26	50.17
AGC	43.71	32.98	35.78
GCN + Inducer	**61.32**	**41.36**	**52.37**

Table 3: Performance Comparisons on graph clustering

Conclusion

This paper studies robust graph convolution networks over heterophilic graphs. We take the first step towards quantitatively analyzing the robustness of GCN approaches over omnipresent heterophilic graphs for node classification, and reveal that the vulnerability is mainly caused by the structural out-of-distribution (OOD). Based on this crucial observation, we present LHS, a novel method that aims to harden GCN against various attacks by learning latent homophilic structures on heterophilic graphs. Our LHS can iteratively refine the latent structure during the learning process, facilitating the model to aggregate information in a homophilic way on heterophilic graphs. Extensive experiments on various benchmarks show the effectiveness of our approach. We believe our structure can also benefit more graph tasks for better representation learning. Future work could focus on the development of novel adversarial training methods based on the structural OOD.

Acknowledgments

This work was partially supported by the National Key R&D Program of China (Grant No.2022YFB2902200), Major Projects of National Natural Science Foundation of China

(Grant No.72293583), and the Joint Funds for Regional Innovation and Development of the National Natural Science Foundation of China (No. U21A20449).

References

Abu-El-Haija, S.; Perozzi, B.; Kapoor, A.; Alipourfard, N.; Lerman, K.; Harutyunyan, H.; Ver Steeg, G.; and Galstyan, A. 2019. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *international conference on machine learning*, 21–29. PMLR.

Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Šrndić, N.; Laskov, P.; Giacinto, G.; and Roli, F. 2013. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*. Springer.

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

Böttcher, A.; and Wenzel, D. 2008. The Frobenius norm and the commutator. *Linear algebra and its applications*, 429(8-9): 1864–1885.

Dai, H.; Li, H.; Tian, T.; Huang, X.; Wang, L.; Zhu, J.; and Song, L. 2018. Adversarial attack on graph structured data. In *International conference on machine learning*, 1115–1124. PMLR.

Halko, N.; Martinsson, P.-G.; and Tropp, J. A. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2): 217–288.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*.

He, D.; Liang, C.; Liu, H.; Wen, M.; Jiao, P.; and Feng, Z. 2022. Block modeling-guided graph convolutional neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 4022–4029.

Jin, D.; Yu, Z.; Huo, C.; Wang, R.; Wang, X.; He, D.; and Han, J. 2021a. Universal graph convolutional networks. *Advances in Neural Information Processing Systems*, 34: 10654–10664.

Jin, W.; Derr, T.; Wang, Y.; Ma, Y.; Liu, Z.; and Tang, J. 2021b. Node similarity preserving graph convolutional networks. In *Proceedings of the 14th ACM international conference on web search and data mining*, 148–156.

Jin, W.; Ma, Y.; Liu, X.; Tang, X.; Wang, S.; and Tang, J. 2020. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 66–74.

Kipf, T. N.; and Welling, M. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Li, H.; Xu, W.; Qiu, C.; and Pei, J. 2022. Fast Markov clustering algorithm based on belief dynamics. *IEEE Transactions on Cybernetics*.

Li, S.; Kim, D.; and Wang, Q. 2023. Restructuring Graph for Higher Homophily via Adaptive Spectral Clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 8622–8630.

Lin, W.; Lan, H.; and Li, B. 2021. Generative Causal Explanations for Graph Neural Networks. In *Proceedings of the 38th International Conference on Machine Learning*.

Liu, M.; Wang, Z.; and Ji, S. 2021. Non-local graph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, Y.; Zheng, Y.; Zhang, D.; Chen, H.; Peng, H.; and Pan, S. 2022. Towards unsupervised deep graph structure learning. In *Proceedings of the ACM Web Conference 2022*, 1392–1403.

Liu, Y.; Zheng, Y.; Zhang, D.; Lee, V. C.; and Pan, S. 2023. Beyond smoothing: Unsupervised graph representation learning with edge heterophily discriminating. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 4516–4524.

Luan, S.; Hua, C.; Lu, Q.; Zhu, J.; Zhao, M.; Zhang, S.; Chang, X.-W.; and Precup, D. 2022. Is Heterophily A Real Nightmare For Graph Neural Networks on Performing Node Classification?

Pei, H.; Wei, B.; Chang, K. C.-C.; Lei, Y.; and Yang, B. 2020. Geom-GCN: Geometric Graph Convolutional Networks. In *International Conference on Learning Representations*.

Qiu, C.; Geng, Y.; Lu, J.; Chen, K.; Zhu, S.; Su, Y.; Nan, G.; Zhang, C.; Fu, J.; Cui, Q.; et al. 2023. 3D-IDS: Doubly Disentangled Dynamic Intrusion Detection. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1965–1977.

Qiu, C.; Huang, Z.; Xu, W.; and Li, H. 2022. VGAER: graph neural network reconstruction based community detection. *arXiv preprint arXiv:2201.04066*.

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

Suresh, S.; Budde, V.; Neville, J.; Li, P.; and Ma, J. 2021a. Breaking the Limit of Graph Neural Networks by Improving the Assortativity of Graphs with Local Mixing Patterns. In *KDD*.

Suresh, S.; Li, P.; Hao, C.; and Neville, J. 2021b. Adversarial graph augmentation to improve graph contrastive learning. *Advances in Neural Information Processing Systems*, 34: 15920–15933.

Tang, J.; Sun, J.; Wang, C.; and Yang, Z. 2009. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 807–816.

Velickovic, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep graph infomax. *ICLR (Poster)*, 2(3): 4.

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

Wang, T.; Jin, D.; Wang, R.; He, D.; and Huang, Y. 2022a. Powerful Graph Convolutional Networks with Adaptive Propagation Mechanism for Homophily and Heterophily. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 4210–4218.

Wang, Y.; and Derr, T. 2021. Tree decomposed graph neural network. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*.

Wang, Y.; Yi, K.; Liu, X.; Wang, Y. G.; and Jin, S. 2022b. ACMP: Allen-cahn message passing with attractive and repulsive forces for graph neural networks. In *The Eleventh International Conference on Learning Representations*.

Xu, Z.; Chen, Y.; Zhou, Q.; Wu, Y.; Pan, M.; Yang, H.; and Tong, H. 2023. Node Classification Beyond Homophily: Towards a General Solution. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2862–2873.

Yan, Y.; Hashemi, M.; Swersky, K.; Yang, Y.; and Koutra, D. 2021. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. *arXiv preprint arXiv:2102.06462*.

Yang, L.; Li, M.; Liu, L.; bingxin niu; Wang, C.; Cao, X.; and Guo, Y. 2021. Diverse Message Passing for Attribute with Heterophily. In Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*.

Yang, T.; Wang, Y.; Yue, Z.; Yang, Y.; Tong, Y.; and Bai, J. 2022. Graph pointer neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 8832–8839.

Yang, Z.; Cohen, W.; and Salakhudinov, R. 2016. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, 40–48. PMLR.

Yuan, H.; and Ji, S. 2021. Node2Seq: Towards Trainable Convolutions in Graph Neural Networks. *CoRR*.

Zhang, F.; Chan, P. P. K.; Biggio, B.; Yeung, D. S.; and Roli, F. 2016. Adversarial Feature Selection Against Evasion Attacks. *IEEE Transactions on Cybernetics*, 46.

Zhang, K.; Zhang, Y.; and Cheng, H. 2020. Self-supervised structure learning for crack detection based on cycle-consistent generative adversarial networks. *Journal of Computing in Civil Engineering*, 34(3): 04020004.

Zhang, X.; and Zitnik, M. 2020. Gnnguard: Defending graph neural networks against adversarial attacks. *Advances in neural information processing systems*, 33: 9263–9275.

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

Zheng, Y.; Zhang, H.; Lee, V.; Zheng, Y.; Wang, X.; and Pan, S. 2023. Finding the Missing-half: Graph Complementary Learning for Homophily-prone and Heterophily-prone Graphs. *arXiv preprint arXiv:2306.07608*.

Zhu, D.; Zhang, Z.; Cui, P.; and Zhu, W. 2019. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 1399–1407.

Zhu, J.; Jin, J.; Loveland, D.; Schaub, M. T.; and Koutra, D. 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*, 2637–2647.

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