

# Graph Neural Networks with Soft Association between Topology and Attribute

Yachao Yang<sup>1</sup>, Yanfeng Sun<sup>1\*</sup>, Shaofan Wang<sup>1</sup>, Jipeng Guo<sup>2</sup>,  
Junbin Gao<sup>3</sup>, Fujiao Ju<sup>1</sup>, Baocai Yin<sup>1</sup>

<sup>1</sup>Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Beijing Institute of Artificial Intelligence, Faculty of Information Technology, Beijing University of Technology, Beijing, China

<sup>2</sup>College of Information Science and Technology, Beijing University of Chemical Technology, China

<sup>3</sup>Discipline of Business Analytics, The University of Sydney Business School, The University of Sydney, Camperdown, NSW 2006, Australia.

yangyc@emails.bjut.edu.cn, {yfsun, wangshaofan, jfj2017, ybc}@bjut.edu.cn, guojipeng@buct.edu.cn, junbin.gao@sydney.edu.au

## Abstract

Graph Neural Networks (GNNs) have shown great performance in learning representations for graph-structured data. However, recent studies have found that the interference between topology and attribute can lead to distorted node representations. Most GNNs are designed based on homophily assumptions, thus they cannot be applied to graphs with heterophily. This research critically analyzes the propagation principles of various GNNs and the corresponding challenges from an optimization perspective. A novel GNN called Graph Neural Networks with Soft Association between Topology and Attribute (GNN-SATA) is proposed. Different embeddings are utilized to gain insights into attributes and structures while establishing their interconnections through *soft association*. Further as integral components of the soft association, a Graph Pruning Module (GPM) and Graph Augmentation Module (GAM) are developed. These modules dynamically remove or add edges to the adjacency relationships to make the model better fit with graphs with homophily or heterophily. Experimental results on homophilic and heterophilic graph datasets convincingly demonstrate that the proposed GNN-SATA effectively captures more accurate adjacency relationships and outperforms state-of-the-art approaches. Especially on the heterophilic graph dataset Squirrel, GNN-SATA achieves a 2.81% improvement in accuracy, utilizing merely 27.19% of the original number of adjacency relationships. Our code is released at <https://github.com/wwwfadecom/GNN-SATA>.

## Introduction

A growing trend has recently been observed in generating data characterized by intricate relationships and interdependencies between objects, frequently represented as graph data. Illustrative examples encompass social networks (Guo et al. 2022), citation networks (Zhao et al. 2021), and financial networks (Bi et al. 2022). Graph neural networks (GNNs) have been proven highly effective in various graph data applications, including recommender systems (Fan et al. 2019), data mining (Wu et al. 2020a), and natural language processing tasks (Wu et al. 2020b).

GNNs can be broadly classified into two categories: spectral-based GNNs and spatial-based GNNs. Spectral-based GNNs utilize the eigendecomposition of the Laplacian matrix of the graph to update node representations by transforming and aggregating the eigenvectors. Spatial-based GNNs perform information propagation and aggregation directly in the graph structure. Some researchers (Ma et al. 2021; Yang et al. 2021; Zhu et al. 2021) have shown that spectral-based GNNs are designed to optimize a unified objective optimization framework, including a graph Laplacian regularization and feature-fitting term. The Laplacian regularization term utilizes the topology smooth feature to acquire a stable embedding that eliminates noise, and the feature fitting term establishes a link between the node embedding and the node attributes.

However, recent studies have revealed that the interference between topology and attribute can have a detrimental impact on the performance of GNNs. On the one hand, topology frequently influences the representation of attributes. GNNs that prioritize topological smoothness tend to minimize differences between interconnected nodes, leading to over-smoothing issues when multiple layers are stacked and resulting in the loss of node attribute information (Jin et al. 2021). The challenge becomes more pronounced in the context of heterophilic graphs, as relying on inaccurate topological relations for transfer can yield entirely distinct representations for two seemingly similar nodes. On the other hand, inaccurate or incomplete node attributes may distort the encoding of topology. In tasks like node link prediction, GNNs that rely on both attribute and topology often exhibit inferior performance compared to encoding methods that exclusively hinge on topology, such as matrix factorization (Mnih and Salakhutdinov 2007) and random block model (Airoldi et al. 2008).

In a recent development, Yang et al. (Yang et al. 2022) introduced the utilization of the Hilbert-Schmidt Independence Criterion (HSIC) (Gretton et al. 2005) to enforce the mutual exclusivity of attribute and topology. By doing so, they effectively eliminate shared information and derive two independent representations. This approach serves to mitigate the interference that arises from the interplay between attribute and topology. Compared with the previous methods, Yang et al.'s model exhibits impressive perfor-

\*Corresponding author.

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

mance on heterophilic data, although its performance on homophilic data is generally consistent. This distinction can be attributed to the relatively dependable relationship between attributes and structures in homophilic data. Consequently, opting for a straightforward learning of mutually exclusive representations may not be the optimal choice in such scenarios.

In order to alleviate the interference between topological structures and attributes, this paper proposes a novel graph neural network model called GNN-SATA, which aims to address the problem of node embedding and topological distortion caused by attribute and topological interference. It employs different representations for learning node attributes and topology, and recognizes the interconnectedness of features and structures by a soft association. Further, we introduce two modules, GPM and GAM, to explore an adaptive graph relationship in which GPM is used to prune the original adjacency relationship according to the established association, and GAM is used to enhance the adjacency relationship. Finally, the model is trained based on the cross-entropy loss of the semi-supervised node classification. The contributions of this paper can be summarized as follows:

- We proposed GNN-SATA, a novel GNN model that addresses the interference between attribute and topology by establishing a soft association between node attributes and topology.
- To investigate more reliable graph relations, we propose two modules, namely GAM and GPM, to explore the topological relationships among nodes dynamically. Experimental results demonstrate the effectiveness of these modules in enhancing the homophily of graph data and improving the model’s performance.
- Compared with state-of-the-art methods, extensive comparative experiments are conducted on six benchmark datasets highlight the superiority of GNN-SATA.

## Relate Works

### Typical Graph Neural Networks

The information propagation of GNNs involves the transformation and aggregation of node characteristics at a specific depth based on topological information. Various GNN models employ different propagation techniques. The following are the message passing formulae of Graph Convolutional networks (GCN) (Kipf and Welling 2017) and Simplifying Graph Convolutional networks (SGC) (Wu et al. 2019):

$$\text{GCN} : \mathbf{Z}^{k+1} = \sigma(\bar{\mathbf{A}}\mathbf{Z}^k\mathbf{W}^k) \quad (1)$$

$$\text{SGC} : \mathbf{Z}^{k+1} = \bar{\mathbf{A}}\mathbf{Z}^k \quad (2)$$

where  $\mathbf{X}$  is node feature matrix,  $\mathbf{Z}^0 = \mathbf{X}$ ,  $\bar{\mathbf{A}}$  is the regularized adjacency matrix which includes a self-loop, and  $\sigma = \text{Relu}$  and  $\mathbf{W}$  are the non-linear activation function and the weight matrix of the corresponding layer, respectively. To learn the representation of nodes, GCN applies linear transformation and non-linear activation  $K$  times. SGC eliminates the non-linear activation and weight matrix, which lessens the computational complexity of GCN

and improves performance for some straightforward graph shapes while being brief and effective.

GCN and SGC learn the data representation  $\mathbf{Z}$  by optimizing the following objective:

$$\min_{\mathbf{Z}} \text{tr}(\mathbf{Z}^T \tilde{\mathbf{L}}\mathbf{Z}) = \sum_{i,j}^N \bar{\mathbf{A}}_{i,j} \|\mathbf{Z}_i - \mathbf{Z}_j\|^2, \mathbf{Z} = \mathbf{X}\mathbf{W}^* \quad (3)$$

where  $\mathbf{W}^* = \mathbf{W}^0 \dots \mathbf{W}^{k-1}$ . and  $N$  is the number of nodes. In essence, this acts as a low-pass filter on the input node by smoothing two adjacency nodes to assign them with a more comparable representation. However, when numerous GCN or SGC layers are stacked together, the over-smoothing issue arises. A few networks focused on solving the over-smoothing problem were suggested, such as APPNP (Gasteiger, Bojchevski, and Günnemann 2019), GCNII (Chen et al. 2020), JKNet (Xu et al. 2018) and DAGNN (Liu, Gao, and Ji 2020). Their inter-layer connection may be described as follows:

$$\text{APPNP} : \mathbf{Z}^{k+1} = (1 - \lambda)\bar{\mathbf{A}}\mathbf{Z}^k + \lambda\mathbf{X}$$

$$\text{GCNII} : \mathbf{Z}^{k+1} = (1 - \lambda)(\mathbf{Z}^k\mathbf{W}) + \lambda\mathbf{X}$$

$$\text{JKNet} : \mathbf{Z}^{k+1} = \sum_{t=1}^{k+1} \lambda_t \bar{\mathbf{A}}^t \mathbf{X}\mathbf{W}^* \quad (4)$$

$$\text{DAGNN} : \mathbf{Z}^{k+1} = \sum_{t=0}^{k+1} \lambda_t \bar{\mathbf{A}}^t \mathbf{H}, \mathbf{H} = f_{\theta}(\mathbf{X})$$

where  $f_{\theta}(\mathbf{X})$  is the non-linear feature transformation utilizing an MLP before the propagation process,  $\lambda$  and  $\lambda_t$  are the appropriate balancing coefficients. By creating a link between the convolutional layer’s output and the original input, APPNP and GCNII reduce the over-smoothing problem and improve node discrimination. JKNet and DAGNN alleviate the over-smoothing issue by fusing multi-scale information from different layers. The optimization objective of the above four methods can be expressed as:

$$\min_{\mathbf{Z}} \|\mathbf{Z} - \mathbf{X}\|_F^2 + \alpha_1 \text{tr}(\mathbf{Z}^T \tilde{\mathbf{L}}\mathbf{Z}) \quad (5)$$

compared with Eq.(3), the above formula retains more original feature information  $\mathbf{X}$  to alleviate the over-smoothing problem.

### Homophily and Heterophily

In graph networks, homophily refers to a situation in which similar pairs of nodes may be more likely to be linked than other pairs, and heterophily refers to a situation in which dissimilar pairings are more likely to be connected. Smooth-based GNNs and their variations are built on the homophily assumption while omitting the fact that heterophilic graph architectures are frequently found in the real world. To address the challenges posed by heterophilic graphs, Pei et al. (Pei et al. 2020) employed geometric transformations in conjunction with graph neural networks to better capture interactions between nodes in heterophilic graphs. Zhu et al. (Zhu et al. 2020a) introduced the concept of hypergraphs,

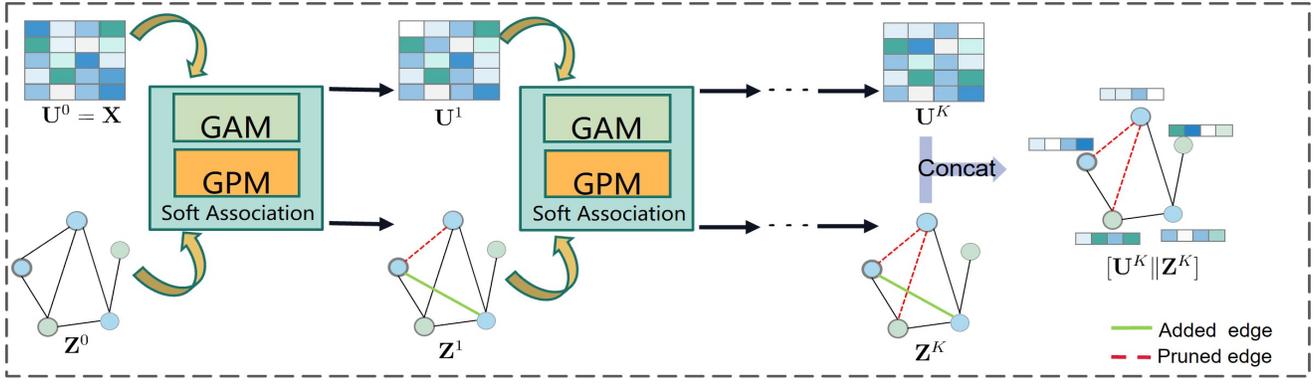


Figure 1: An overview of the proposed GNN-SATA model.

which enable edges to connect multiple nodes, enhancing their capacity to represent and manage diverse connections. Yang et al. (Yang et al. 2022) iteratively update node representations by collecting semantic information and removing common information to improve the model’s performance in heterophilic networks. Additionally, Huang et al. (Huang et al. 2023) proposed a straightforward and efficient mid-pass filter with increased resilience. These approaches offer promising insights into handling heterophily in graph networks and contribute to advancing graph neural network research.

### Model Formulation

Let  $\mathcal{G} = \{\mathbf{X}, \mathcal{V}, \mathcal{E}, \mathbf{A}\}$  be an attribute graph with  $C$  class nodes.  $\mathcal{V} = \{v_1, \dots, v_N\}$  and  $\mathcal{E}$  represent the sets of nodes and edges respectively,  $N = |\mathcal{V}|$  is the number of nodes.  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times d}$  is node feature matrix,  $d$  is the dimension of features.  $\mathcal{V}_L$  is labeled training set, and the corresponding labels are  $\mathcal{Y}_L = \{y_1, y_2, \dots, y_L\}$ .  $\mathbf{1} = \{1\}^{N \times N}$ ,  $\mathbf{A} = (a_{ij})_{N \times N} \in \mathbb{R}^{N \times N}$  is an adjacency matrix of  $\mathcal{G}$ ,  $a_{ij} = 1$  if a connecting edge (i.e.,  $(v_i, v_j) \in \mathcal{E}$ ) between node  $v_i$  and  $v_j$  exists, else  $a_{ij} = 0$ . The degree matrix of  $\mathbf{A}$  is denoted as  $\mathbf{D}$ , and  $\bar{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$  is the regularized adjacency matrix, where  $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$  is an adjacency matrix with self-connections and  $\tilde{\mathbf{D}} = \mathbf{D} + \mathbf{I}$ .  $\tilde{\mathbf{L}} = \mathbf{I} - \bar{\mathbf{A}}$  is the Laplacian matrix that represents a normalized symmetric positive semi-definite network. The node classification task is to classify  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  into  $C$  categories.

### The Proposed GNN-SATA

The objective of Eq.(5) is to acquire a representation  $\mathbf{Z}$  that closely aligns with the attributes of a node while also satisfying the topological constraints. However, it is challenging to extract optimal attribute and structural embedding simultaneously (Yang et al. 2022). Furthermore, in the case of high heterophilic graph data, a feature smooth representation based on inaccurate graph structure may not effect node representation extraction and may have the opposite influence (Zhu et al. 2020a; Jin et al. 2021; Yang et al. 2023). Consequently, we propose to learn feature-structure representation separately while simultaneously establishing the

soft association between attribute and topology via Eq.(6). That is:

$$\min_{\mathbf{Z}, \mathbf{U}} \|\mathbf{U} - \mathbf{X}\|_F^2 + \alpha_1 \text{tr}(\mathbf{Z}^\top \tilde{\mathbf{L}} \mathbf{Z}) + \alpha_2 \|\mathbf{U} \mathbf{U}^\top - \mathbf{A}\|_F^2 \quad (6)$$

where  $\mathbf{U}$  and  $\mathbf{Z}$  denote the attribute and topology representation of graph data, respectively, which are associated with each other by soft association via the third term. Specifically, we use the representation  $\mathbf{U}$  extracted from the features to construct a similarity matrix, bringing this matrix close to the original adjacency matrix  $\mathbf{A}$  to establish connections between features and structures. The third term is balanced by the weight coefficient  $\alpha_2$ , which acts as an appropriate constraint in the learning process of  $\mathbf{U}$  and  $\mathbf{Z}$ .

In order to further enhance the versatility of the model on homophilic and heterophilic graphs, the GNN-SATA model has been proposed, consisting of the Graph Pruning Module (GPM) and the Graph Augmentation Module (GAM). The GPM is responsible for removing irrelevant or false edges from the original graph to ensure these edges do not affect the network. Simultaneously, the GAM module investigates the absence of edge connections and supplements the original graph.

The GPM achieves this by introducing a learnable matrix  $\mathbf{E}_P$  to prune the original adjacency matrix  $\mathbf{A}$ , using  $\mathbf{A} \odot \mathbf{E}_P$  instead of  $\mathbf{A}$ . Pruning edges is guided by the soft association between attribute and downstream tasks. Nodes with low feature similarity are likelier to have no connections, and the feedback from downstream tasks aids the model in making decisions. However, for graphs with high heterophily, simply deleting edges may cause the GNN to rely on fewer or no edges during iteration. To address this, we incorporated a mechanism to the model attempting to add edges while removing untrusted edges. Like edge pruning, an adaptive matrix  $\mathbf{E}_A$  is introduced to act on the complementary matrix  $(\mathbf{1} - \mathbf{A})$  of  $\mathbf{A}$ . The objective function of GNN-SATA can be expressed as:

$$\min_{\mathbf{Z}, \mathbf{U}, \mathbf{E}_P, \mathbf{E}_A} \|\mathbf{U} - \mathbf{X}\|_F^2 + \alpha_1 \text{tr}(\mathbf{Z}^\top \tilde{\mathbf{L}} \mathbf{Z}) + \alpha_2 \|\mathbf{U} \mathbf{U}^\top - (\mathbf{A} \odot \mathbf{E}_P + (\mathbf{1} - \mathbf{A}) \odot \mathbf{E}_A)\|_F^2 \quad (7)$$

where  $\odot$  represents element-wise multiplication. And note

**Algorithm 1: Optimization Algorithm of GNN-SATA**

**Input:** Feature matrix  $\mathbf{X}$ , adjacency matrix  $\mathbf{A}$ , iterations  $T = 200$  and  $K \in \{2, 3, 4\}$ , Labels  $\mathcal{Y}_L$ , Hyper-parameter  $\alpha_1, \alpha_2$ .

**Parameter:**  $\mathbf{W}_L, \mathbf{W}_U$ .

**Output:** Classification results  $\mathbf{M}'$ .

- 1: Initializing  $\mathbf{E}_P$  and  $\mathbf{E}_A$ .
- 2: **for**  $iter1 = 1, 2, \dots, T$  **do**
- 3:   **for**  $iter2 = 1, 2, \dots, K$  **do**
- 4:     Calculate  $\mathbf{U}^{k+1}, \mathbf{Z}^{k+1}, \mathbf{E}_A^{k+1}, \mathbf{E}_P^{k+1}$  by Eq.(15), (16),(17),(18).
- 5:   **end for**
- 6:   Calculate  $\mathbf{M} = [\mathbf{U}^K \parallel \mathbf{Z}^K]$
- 7:   Calculate classification results  $\mathbf{M}'$  by Eq.(8).
- 8:   Calculate classification loss by Eq.(9)
- 9:   Update parameter  $\mathbf{W}_L, \mathbf{W}_U$  by gradient descent.
- 10: **end for**

that when the model applies to non-boolean adjacency matrix  $\mathbf{A}$ , we replace  $\mathbf{1} - \mathbf{A}$  with  $\mathbf{1} - \mathbb{B}(\mathbf{A})$  where  $\mathbb{B}(\cdot)$  denotes the elementwise boolean operator.

GNN-SATA utilizes the concatenated representation of  $\mathbf{U}^K$  and  $\mathbf{Z}^K$  for classification tasks. The spliced encoding is expressed as  $\mathbf{M} = [\mathbf{U}^K \parallel \mathbf{Z}^K]$  and is used for semi-supervised node classification tasks after linear transformation and softmax function:

$$\mathbf{M}' = \text{softmax}(\mathbf{W}_L \cdot \mathbf{M} + \mathbf{b}) \quad (8)$$

where  $\mathbf{M}'_{ik}$  represents the possibility that the  $i$ -th node belongs to the  $k$ -th class. Similarly, the loss function of labeled data can be expressed as:

$$\mathcal{L} = \sum_{v_i \in \mathcal{V}_L} \ell(\mathbf{m}'_i, \mathbf{y}_i) \quad (9)$$

where  $\mathbf{y}_i$  is the one-hot encoding of  $y_i$  and  $\ell(\cdot)$  is the cross entropy function. Algorithm 1 shows the entire optimization process of GNN-SATA.

## Model Optimization

To determine  $\mathbf{U}$  and  $\mathbf{Z}$ , we use the alternating direction optimization by vanishing the partial derivative of the target function of Eq.(7) for all variables, i.e.  $\mathbf{U}, \mathbf{Z}, \mathbf{E}_P, \mathbf{E}_A$ . For convenience, let  $\mathbf{A}' = (\mathbf{1} - \mathbf{A})$  and  $\mathcal{A} = \mathbf{A} \odot \mathbf{E}_P + \mathbf{A}' \odot \mathbf{E}_A$ , then

$$2(\mathbf{U} - \mathbf{X}) + 2\alpha_2(2\mathbf{U}\mathbf{U}^\top - \bar{\mathcal{A}})\mathbf{U} = 0 \quad (10)$$

$$2\alpha_1 \bar{\mathcal{L}}_A \mathbf{Z} = 0 \quad (11)$$

$$\mathbf{V}_P + 2\alpha_2(-\mathbf{A} \odot \hat{\mathbf{U}}_P + \mathbf{A} \odot \mathbf{A} \odot \mathbf{E}_P) = 0 \quad (12)$$

$$\mathbf{V}_A + 2\alpha_2(-\mathbf{A}' \odot \hat{\mathbf{U}}_A + \mathbf{A}' \odot \mathbf{A}' \odot \mathbf{E}_A) = 0 \quad (13)$$

where  $\bar{\mathcal{A}}$  is the regularized adjacency matrix like  $\bar{\mathbf{A}}, \mathbf{V}_{Pij} = \frac{1}{2}\alpha_1 \mathbf{A}_{ij} \|\mathbf{Z}_i - \mathbf{Z}_j\|_2^2, \bar{\mathbf{A}}, \mathbf{V}_{Aij} = \frac{1}{2}\alpha_1 \mathbf{A}'_{ij} \|\mathbf{Z}_i - \mathbf{Z}_j\|_2^2, \hat{\mathbf{U}}_P = \mathbf{U}\mathbf{U}^\top - \mathbf{A}' \odot \mathbf{E}_A, \hat{\mathbf{U}}_A = \mathbf{U}\mathbf{U}^\top - \mathbf{A} \odot \mathbf{E}_P$ . The iterative relationship between the  $k$ -th layer and the  $k+1$ -th layer of

Dataset	Nodes	Features	Classes	Edges	$\mathcal{H}(\mathbf{A})$
Cora	2708	1433	7	5429	0.809
Citeseer	3327	3703	6	4732	0.721
Photo	7650	745	8	119081	0.824
Computer	13752	767	10	245861	0.791
Squirrel	5201	2089	5	217073	0.203
Chameleon	2277	2325	5	36101	0.233

Table 1: The statistics of six datasets.

the model can be expressed as:

$$\mathbf{U}^{k+1} = \mathbf{X} - \alpha_2(2\mathbf{U}^k \mathbf{U}^{k\top} - \bar{\mathcal{A}})\mathbf{U}^k \quad (14)$$

$$\mathbf{Z}^{k+1} = \bar{\mathcal{A}}\mathbf{Z}^k \quad (15)$$

$$\mathbf{E}_P^{k+1} = (2\alpha_2 \mathbf{A} \odot \hat{\mathbf{U}}_P^k - \mathbf{V}_P^k) / (2\alpha_2 \mathbf{A} \odot \mathbf{A}) \quad (16)$$

$$\mathbf{E}_A^{k+1} = (2\alpha_2 \mathbf{A}' \odot \hat{\mathbf{U}}_A^k - \mathbf{V}_A^k) / (2\alpha_2 \mathbf{A}' \odot \mathbf{A}') \quad (17)$$

where  $\mathbf{V}_{ij}^k = \frac{1}{2}\alpha_1 \mathbf{A}_{ij} \|\mathbf{Z}_i^{k+1} - \mathbf{Z}_j^{k+1}\|_2^2, \hat{\mathbf{U}}^k = \mathbf{U}^{k+1} \mathbf{U}^{k+1\top} - \mathbf{A}' \odot \mathbf{E}_A$ . And we introduced additional training parameters  $\mathbf{W}_U$  and the activation function  $\sigma = \text{Relu}$  to Eq.(14) to enhance the model's capacity for representation and optimize the training process while simultaneously repeatedly updating  $\mathbf{U}^{k+1}$  and  $\mathbf{Z}^{k+1}$  to minimize the optimization objective. The following is the optimized iteration method of Eq (14):

$$\mathbf{U}^{k+1} = \mathbf{X} - \alpha_2 \sigma[(2\mathbf{U}^k \mathbf{U}^{k\top} - \bar{\mathcal{A}})\mathbf{U}^k \mathbf{W}_U] \quad (18)$$

in Eq (18),  $\mathbf{U}^k \mathbf{U}^{k\top} - \bar{\mathcal{A}}$  represents the difference between the feature-based similarity relationship and the adaptive matrix  $\mathcal{A}$ , and  $\mathbf{X} - \alpha_2 \sigma[(2\mathbf{U}^k \mathbf{U}^{k\top} - \bar{\mathcal{A}})\mathbf{U}^k \mathbf{W}_U]$  represents the component that eliminates the difference while retaining the commonality, thereby establishing a soft association between features and structures. The feature-based representation  $\mathbf{U}$  allows us to discover the structure's commonalities while preserving the original features' uniqueness. In Eq (15),  $\bar{\mathcal{A}}\mathbf{Z}^k$  is the label propagation algorithm (LPA) (Raghavan, Albert, and Kumara 2007) in community detection.

## Experiments

This section describes the datasets, comprised methods, and experimental parameter configurations to evaluate GNN-SATA. In addition, we analyze the experimental results and demonstrate the significance of the proposed soft association, GPM and GAM through ablation studies.

### Datasets

The proposed GNN-SATA is evaluated on four high homophilic datasets (Cora, Citeseer, Photo, Computer) and two heterophilic (Squirrel, Chameleon) datasets. The attribute statistics of datasets are shown in Table 1.

The training process iterates for 200 epochs on a machine with RTX 3090 Ti GPU. In order to assess the effectiveness of the proposed model in classification tasks, we partitioned the nodes of each class in all datasets into three sets: 60%

for training, 20% for validation, and 20% for testing like other baselines. We utilized Accuracy (ACC), a commonly employed and significant evaluation metric to measure the model’s performance. A higher value of this metric signifies the superior performance of the model. In comparison, the results of the baseline methods are reported according to their original references.

## Competitive Methods

We demonstrate the efficacy of GNN-SATA by comparing it to 13 classical and state-of-the-art baselines. According to their role, all methods can be divided into three categories:

**Classical GCN methods:** GCN (Kipf and Welling 2017), GAT (Veličković et al. 2017) and GraphSAGE (Hamilton, Ying, and Leskovec 2017) are three classical GNNs.

**Methods for alleviating the over-smoothing problem:** GCNII (Chen et al. 2020) introduces a personal-ity initialization method. APPNP (Gasteiger, Bojchevski, and Günnemann 2019) incorporates personalized features. JKNet (Xu et al. 2018) alleviates over-smoothing with skip connections.

**Methods for handling heterophilic graphs:** Geom-GCN (Pei et al. 2020) incorporates geometric info for effective graph convolutions. GPRGNN (Chien et al. 2021) adapts weights using graph pagerank for feature extraction. FAGCN (Bo et al. 2021) uses adaptive gating for signal integration in message passing. H2GCN (Zhu et al. 2020b) learns from heterophilic graphs with multiple modalities. Ordered GNN (Song et al. 2023) utilizes an ordered gating mechanism to prevent the mixing of node features within a hop to enhance the robustness of the model.

## Analysis of Results

The classification results of both the proposed method and the comparison methods on six benchmark datasets are presented in Table 2. Table 2 reveals the following observations:

GNN-SATA demonstrates strong performance across all datasets. On average, the proposed method improves by 1.82% compared to the optimal baseline results across the six datasets, with a remarkable 2.81% improvement specifically on the Squirrel dataset, confirming the method’s superiority. GNN-SATA employs a more straightforward and efficient process by concatenating the final representations of attributes and structures, in contrast to the comparative method H2GCN which enhances the representation ability by integrating self-encoding, neighbor encoding, high-order adjacency encoding, and intermediate layer encoding to address the challenges posed by heterophilic graphs. However, this approach requires significant memory resources and becomes unfeasible for large graphs.

The MLP-based methods outperform several classic and advanced graph neural network methods, including GCN, GAT, GraphSAGE, and Geom-GCN, on the highly heterophilic Squirrel and Chameleon datasets. These graph neural network approaches rely too much on homophily assumption which leads to poor results on datasets with heterophily due to topology and attribute interference. Furthermore, although GNN-BC achieves suboptimal results on most graph data with low homophily, it exhibits moderate

performance on homophilic data. This can be attributed to the authors’ treating features and structures as opposing entities, enforcing HSIC constraints that render the representations of features and structures mutually exclusive. Consequently, the interconnection between features and structures must be considered for effective results. In contrast, GNN-SATA enhances the independence of attribute and topology by learning their respective information separately and capturing their commonality via establishing soft associations between attribute and topology. Therefore, the best results have been achieved in both homophilic and heterophilic graphs.

Besides, the proposed model can significantly improve on datasets with heterophily, especially the Squirrel and Chameleon datasets. We analyze that this is due to the critical role played by the proposed GAM and GPM models. For datasets with high heterophily, the pruning and enhancement of the adjacency matrix can alleviate the negative impact caused by inaccurate structure to a certain extent so that the learned structural information is more accurate and better results can be obtained. And the **Ablation Study** substantiates this statement.

## Ablation Study

Ablation experiments are conducted to assess the individual contributions of different modules in our model. The four variants tested were: SATA-S, SATA-P, SATA-A, and SATA-AP, and the experimental results are shown in Table 3.

- **SATA-S:** The proposed variant model, denoted as SATA-S, is essentially GNN-SATA with the exclusion of soft association. To avoid a trivial solution where  $\mathbf{U} = \mathbf{X}$ , a two-layer MLP is employed for nonlinear transformation of features.
- **SATA-P:** The SATA-P variant refers to the GNN-SATA model without the GPM, which implies that the model eliminates the edge pruning in the soft association component and solely preserves the edge augmentation.
- **SATA-A:** The SATA-A variant refers to the GNN-SATA model without the GAM, which implies that the model eliminates add edges in the soft association component and solely preserves the edge pruning.
- **SATA-AP:** The SATA-AP variant pertains to the GNN-SATA model that lacks GAM and GPM. Consequently, the edges in the soft association component of the model remain fixed and cannot be modified. The guidance relations become unidirectional, generated solely by topology guidance features.

The SATA-S exhibits a decreased performance on all datasets. The variant model independently acquires knowledge of feature representation and topology structure and subsequently merges the representation. It is illogical to consider the attribute and topology of identical data as two unrelated entities. Moreover, the SATA-S outperforms the GCN and MLP, indicating the indispensability of integrating attribute and topology learning.

The effect of SATA-AP is decreased in all datasets, but the decline was more evident in datasets with heterophily.

Dataset	Cora	Citeseer	Photo	Computer	Squirrel	Chameleon
GCN	85.77 ± 0.2	73.68 ± 0.31	90.54 ± 0.21	82.52 ± 0.32	23.96 ± 0.26	28.18 ± 0.23
GAT	86.37 ± 0.30	74.32 ± 0.27	90.09 ± 0.27	81.95 ± 0.38	30.03 ± 0.25	42.93 ± 0.28
GraphSAGE	87.77 ± 1.04	71.09 ± 1.30	90.51 ± 0.25	83.11 ± 0.23	36.28 ± 1.73	49.24 ± 1.68
MLP	74.82 ± 2.22	70.94 ± 0.39	78.69 ± 0.30	70.48 ± 0.28	37.04 ± 0.46	49.67 ± 0.78
GCNII	88.49 ± 2.78	77.08 ± 1.21	90.98 ± 0.93	86.13 ± 0.51	37.85 ± 2.76	60.61 ± 2.00
APPNP	87.87 ± 0.85	76.53 ± 1.33	91.11 ± 0.26	81.99 ± 0.26	33.29 ± 1.72	54.30 ± 0.34
JKNet	88.93 ± 1.35	74.37 ± 1.53	87.70 ± 0.70	77.80 ± 0.97	44.24 ± 2.11	62.31 ± 2.76
Geom-GCN-I	85.19 ± 1.13	<u>77.99 ± 1.23</u>	NA	NA	33.32 ± 1.59	60.31 ± 1.77
Geom-GCN-P	84.93 ± 0.51	75.14 ± 1.50	NA	NA	38.14 ± 1.23	60.90 ± 1.13
Geom-GCN-S	85.27 ± 1.48	74.71 ± 1.17	NA	NA	36.24 ± 1.05	59.96 ± 2.03
GPRGNN	88.65 ± 1.37	<u>77.99 ± 1.64</u>	91.93 ± 0.26	82.90 ± 0.37	49.93 ± 1.34	67.48 ± 1.98
FAGCN	87.77 ± 1.69	<u>74.66 ± 2.27</u>	91.96 ± 0.71	86.09 ± 0.40	40.88 ± 2.02	61.12 ± 1.95
H2GCN-1	86.92 ± 1.37	77.07 ± 1.64	OOM	OOM	36.42 ± 1.89	57.11 ± 1.58
H2GCN-2	87.81 ± 1.35	76.88 ± 1.77	OOM	OOM	37.90 ± 2.02	59.39 ± 1.98
GNN-BC	88.75 ± 1.21	76.70 ± 0.77	<u>93.17 ± 0.67</u>	<u>89.60 ± 0.89</u>	61.41 ± 1.55	74.63 ± 0.93
Ordered GNN	88.37 ± 0.75	77.31 ± 1.73	NA	NA	62.44 ± 1.96	72.28 ± 2.29
GNN-SATA	<b>91.24 ± 0.69</b>	<b>78.03 ± 0.75</b>	<b>93.62 ± 0.55</b>	<b>91.34 ± 0.29</b>	<b>65.25 ± 1.33</b>	<b>77.12 ± 0.96</b>

Table 2: The classification results on all six datasets. The best classification results are bolded, and the second-best results are underlined.

Examples include the Squirrel and Chameleon datasets. We conclude that this is because edge pruning and augmenting are more critical for heterophily datasets. On the one hand, untrue edges can be deleted. On the other hand, connections can be established based on soft associations to explore reliable edges. The next Section **Analysis of Adaptive Matrix  $\mathcal{A}$**  illustrates that the proposed method significantly reduces the heterophily of nodes.

Simultaneously, the outcomes of SATA-A and SATA-P exhibit superior performance on all datasets compared to SATA-AP, signifying the contribution of both GAM and GPM modules in enhancing the model’s efficacy. The experimental findings reveal that SATA-A yields higher results than SATA-P, indicating the greater significance of GPM in adapting topology relationships. Adding edges requires exploring edges that don’t exist, making it more challenging than pruning existing edges. Consequently, GAM’s effect on improving the model’s performance is comparatively lesser than GPM’s.

### Analysis of Adaptive Matrix $\mathcal{A}$

Initially, this study introduces a homophily rate  $\mathcal{H}(\cdot)$  proposed by Pei et al. (Pei et al. 2020) to measure the proportion of edges connecting nodes with the same class in a graph.

Dataset	SATA-S	SATA-A	SATA-P	SATA-AP	GNN-SATA
Cora	86.33	<b>91.24</b>	<u>90.05</u>	89.23	<b>91.24</b>
Citeseer	75.01	<u>76.60</u>	76.30	76.00	<b>78.33</b>
Photo	91.33	<u>93.30</u>	92.33	92.12	<b>93.62</b>
Computer	82.32	90.76	90.26	90.14	<b>91.34</b>
Squirrel	35.23	<u>63.38</u>	61.95	61.23	<b>65.25</b>
Chameleon	50.44	<u>75.60</u>	73.76	72.89	<b>77.12</b>

Table 3: The ablation experiment results of five modules. The best classification results are bolded, and the second-best results are underlined.

The matching rate of the adaptive adjacency matrix  $\mathcal{H}(\mathcal{A})$  can be expressed as:

$$\mathcal{H}(\mathcal{A}) = \frac{|\{(u, v) : (u, v) \in \mathcal{E}_{\mathcal{A}} \wedge y_u = y_v\}|}{|\mathcal{E}_{\mathcal{A}}|}. \quad (19)$$

To analyze the  $\mathcal{A}$  matrix more specifically, statistical analysis was conducted on the edges of the matrix, as presented in Table 4. From Table 4, we can get the following conclusions:

It can be seen from  $|\mathcal{E}_{\mathcal{A}}|/|\mathcal{E}_{\mathcal{A}}|$  that datasets with high homophily tend to learn more edges. This is because it is easier to add edges to datasets with high homophily, and the homophily rate can be increased by adding more homophilic edges so that the edge-increasing effect of GAM is more prominent. On the contrary, graph data with heterophily contains more heterophilic edges, and GPM enhances the homophily ratio by removing such edges. For instance, in the Squirrel dataset, utilizing merely 27.19% of the original number of adjacency relationships and achieve better results than the baseline methods.

Furthermore,  $\mathcal{H}(\mathcal{A})/\mathcal{H}(\mathbf{A}) > 1$  on all datasets demonstrates that the proposed model has enhanced the homophily rate for all datasets, and the improvement effect is more obvious for datasets with lower homophily.

### Analysis of Parameters

This section examines the two hyperparameters  $\alpha_1$  and  $\alpha_2$  in Eq (7). The purpose of parameter  $\alpha_1$  is to balance the loss associated with topological smoothness constraints, while parameter  $\alpha_2$  is used to balance the loss related to soft association constraints. The value of parameter  $\alpha_1$  is chosen from the set  $[1, 3, 5, 7, 10]$ , while parameter  $\alpha_2$  is selected from the set  $[0.001, 0.01, 0.1, 1]$ . From Figure 3, it is evident that GNN-SATA demonstrates robustness in response to variations in  $\alpha_1$ . However, changes in  $\alpha_2$  have a pronounced impact on the model. Similar results across both homophilic and heterophilic graphs, as observed in cases like Cora and

Dataset	Cora	Citeseer	Photo	Computers	Squirrel	Chameleon
$ \{(u, v) : (u, v) \in \mathcal{E}_A \wedge y_u = y_v\} $	5858	5044	97966	102265	203679	4154
$ \{(u, v) : (u, v) \in \mathcal{E}_A \wedge y_u \neq y_v\} $	1118	1310	20956	51888	44973	9939
$ \mathcal{E}_A $	6976	6354	123221	255567	59024	14039
$ \mathcal{E}_A / \mathcal{E}_A $	1.286	1.35	1.035	1.04	0.271	0.389
$\mathcal{H}(\mathcal{A})$	0.84	0.794	0.83	0.797	0.238	0.296
$\mathcal{H}(\mathcal{A})/\mathcal{H}(\mathbf{A})$	1.038	1.101	1.007	1.008	1.172	1.265

Table 4: The statistical data pertaining to  $\mathcal{A}$ .

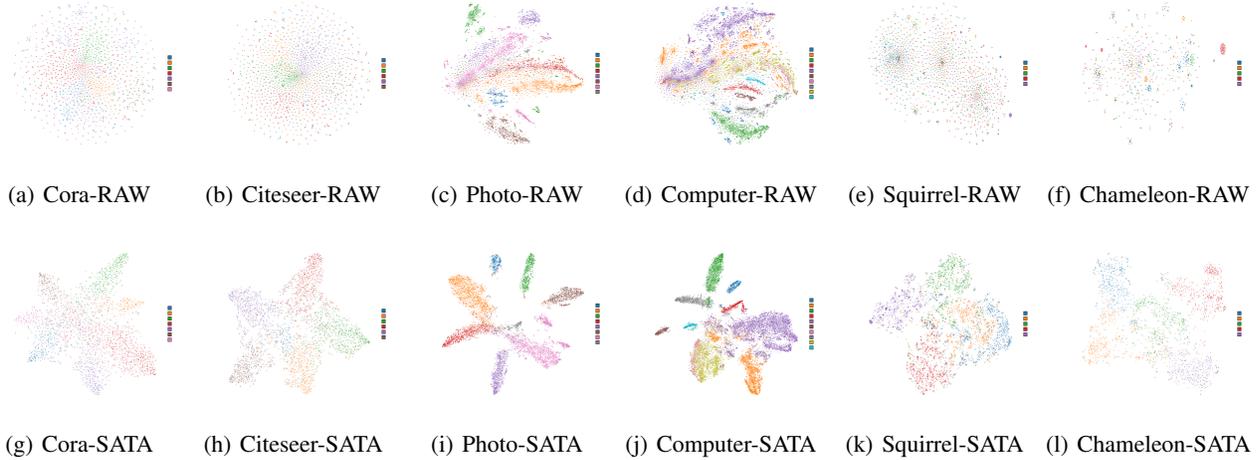


Figure 2: The visualization of classification results.

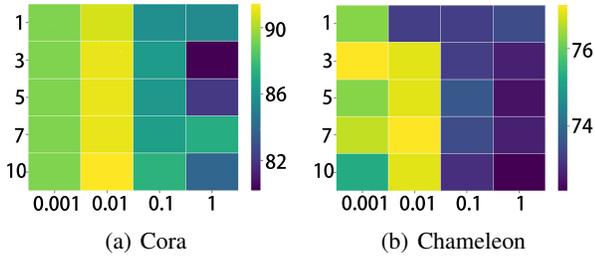


Figure 3: The accuracy of node classification with different parameters  $\alpha_1$  and  $\alpha_2$ .

Chameleon. For parameter  $\alpha_1$ , the model achieves superior results when its value is set to 3 or 5, indicating that appropriate topology smoothing constraints lead to improved performance.

Additionally, parameter  $\alpha_2$  yields the best results when its value is set to 0.01, while a value of 1 produces poor results. This discrepancy arises because the proposed soft association constraint aims to establish a more flexible association between topology and attribute, thereby mitigating issues related to over-smoothing and distortion of node representations. Setting large values of  $\alpha_2$  causes the model to degenerate to an approximation to Eq.(3). Consequently, avoiding setting the  $\alpha_2$  too high is advisable.

### Classification Result Visualization

Figure 2 visually presents the classification results in a two-dimensional space using the T-SNE (Van der Maaten and Hinton 2008) algorithm to provide a more intuitive verification of the proposed model’s effectiveness. In contrast to the original raw irregular data distribution (-RAW), GNN-SATA (-SATA) effectively categorizes the data into distinct groups, providing compelling evidence of the proposed model’s capability in classifying various irregular data.

### Conclusion

This paper proposes a novel graph neural network for semi-supervised node classification called GNN-SATA. The network is designed based on the optimization principle of graph neural networks and incorporates a soft association constraint. The soft associations aim to address the issues of node representation distortion arising from the compromise between attribute and topology. Additionally, GNN-SATA utilizes the GAM and GPM to adaptively adjust the soft association by adding or removing edges based on the topology, node attribute, and classification tasks. The analysis of the adjacency matrix  $\mathcal{A}$  demonstrates that the GAM and GPM techniques effectively enhance homophily and significantly impact the addition and removal of adjacency matrix edges. Furthermore, the performance of node classification serves as evidence of the efficacy of the GNN-SATA model.

## Acknowledgments

This research was supported by the National Key R&D Program of China (No.2021ZD0111902), NSFC (62172023, U21B2038, U19B2039), Beijing Natural Science Foundation (4222021) and the China Postdoctoral Science Foundation under Grant 2023M740201.

## References

- Airoldi, E. M.; Blei, D.; Fienberg, S.; and Xing, E. 2008. Mixed membership stochastic blockmodels. *Advances in Neural Information Processing Systems*, 21.
- Bi, W.; Xu, B.; Sun, X.; Wang, Z.; Shen, H.; and Cheng, X. 2022. Company-as-tribe: Company financial risk assessment on tribe-style graph with hierarchical graph neural networks. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2712–2720.
- Bo, D.; Wang, X.; Shi, C.; and Shen, H. 2021. Beyond low-frequency information in graph convolutional networks. In *AAAI Conference on Artificial Intelligence*, volume 35, 3950–3957.
- Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; and Li, Y. 2020. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, 1725–1735.
- Chien, E.; Peng, J.; Li, P.; and Milenkovic, O. 2021. Adaptive universal generalized pagerank graph neural network. In *International Conference on Learning Representations*.
- Fan, W.; Ma, Y.; Li, Q.; He, Y.; Zhao, E.; Tang, J.; and Yin, D. 2019. Graph neural networks for social recommendation. In *The World Wide Web Conference*, 417–426.
- Gasteiger, J.; Bojchevski, A.; and Günnemann, S. 2019. Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations*.
- Gretton, A.; Bousquet, O.; Smola, A.; and Schölkopf, B. 2005. Measuring statistical dependence with Hilbert-Schmidt norms. In *International Conference on Algorithmic Learning Theory*, 63–77. Springer.
- Guo, Z.; Yu, K.; Jolfaei, A.; Li, G.; Ding, F.; and Beheshti, A. 2022. Mixed graph neural network-based fake news detection for sustainable vehicular social networks. *IEEE Transactions on Intelligent Transportation Systems*, 1–13.
- Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30.
- Huang, J.; Du, L.; Chen, X.; Fu, Q.; Han, S.; and Zhang, D. 2023. Robust Mid-Pass Filtering Graph Convolutional Networks. *arXiv:2302.08048*.
- Jin, W.; Derr, T.; Wang, Y.; Ma, Y.; Liu, Z.; and Tang, J. 2021. Node similarity preserving graph convolutional networks. In *ACM International Conference on Web Search and Data Mining*, 148–156.
- Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.
- Liu, M.; Gao, H.; and Ji, S. 2020. Towards deeper graph neural networks. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 338–348.
- Ma, Y.; Liu, X.; Zhao, T.; Liu, Y.; Tang, J.; and Shah, N. 2021. A unified view on graph neural networks as graph signal denoising. In *ACM International Conference on Information & Knowledge Management*, 1202–1211.
- Mnih, A.; and Salakhutdinov, R. R. 2007. Probabilistic matrix factorization. *Advances in Neural Information Processing Systems*, 20.
- 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*.
- Raghavan, U. N.; Albert, R.; and Kumara, S. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3): 036106.
- Song, Y.; Zhou, C.; Wang, X.; and Lin, Z. 2023. Ordered GNN: Ordering Message Passing to Deal with Heterophily and Over-smoothing. In *International Conference on Learning Representations*.
- Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11).
- Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv:1710.10903*.
- Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying graph convolutional networks. In *International Conference on Machine Learning*, 6861–6871. PMLR.
- Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Philip, S. Y. 2020a. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1): 4–24.
- Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Philip, S. Y. 2020b. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1): 4–24.
- Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; Kawarabayashi, K.-i.; and Jegelka, S. 2018. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*, 5453–5462.
- Yang, L.; Wang, C.; Gu, J.; Cao, X.; and Niu, B. 2021. Why do attributes propagate in graph convolutional neural networks? In *AAAI Conference on Artificial Intelligence*, volume 35, 4590–4598.
- Yang, L.; Zhou, W.; Peng, W.; Niu, B.; Gu, J.; Wang, C.; Cao, X.; and He, D. 2022. Graph Neural Networks Beyond Compromise Between Attribute and Topology. In *ACM Web Conference*, 1127–1135.
- Yang, Y.; Sun, Y.; Ju, F.; Wang, S.; Gao, J.; and Yin, B. 2023. Multi-graph Fusion Graph Convolutional Networks with pseudo-label supervision. *Neural Networks*, 158: 305–317.
- Zhao, T.; Liu, Y.; Neves, L.; Woodford, O.; Jiang, M.; and Shah, N. 2021. Data augmentation for graph neural networks. In *AAAI Conference on Artificial Intelligence*, volume 35, 11015–11023.

- Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; and Koutra, D. 2020a. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems*, 33: 7793–7804.
- Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; and Koutra, D. 2020b. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems*, 33: 7793–7804.
- Zhu, M.; Wang, X.; Shi, C.; Ji, H.; and Cui, P. 2021. Interpreting and unifying graph neural networks with an optimization framework. In *ACM Web Conference*, 1215–1226.