Multiplex Graph Representation Learning via Bi-level Optimization

Yudi Huang^{1,2}, **Yujie Mo**^{3*}, **Yujing Liu**^{1,2}, **Ci Nie**^{1,2}, **Guoqiu Wen**^{1,2}, **Xiaofeng Zhu**^{1,2}

^{1}School of Computer Science and Engineering, Guangxi Normal University, Guilin 541004, China
^{2}Guangxi Key Lab of Multisource Information Mining Security, Guilin 541004, China
^{3}School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

Abstract

Many multiplex graph representation learning (MGRL) methods have been demonstrated to 1) ignore the globally positive and negative relationships among node features; and 2) usually utilize the node classification task to train both graph structure learning and representation learning parameters, and thus resulting in the problem of edge starvation. To address these issues, in this paper, we propose a new MGRL method based on the bi-level optimization. Specifically, in the inner level, we optimize the self-expression matrix to capture the globally positive and negative relationships among nodes, as well as complement them with the local relationships in graph structures. In the outer level, we optimize the parameters of the graph convolutional layer to obtain discriminative node representations. As a result, the graph structure optimization does not depend on the node classification task, which solves the edge starvation problem. Extensive experiments show that our model achieves the superior performance on node classification tasks on all datasets.

1 Introduction

Multiplex graph consists of multiple graph structures and shared node features, and each graph reflects a specific relationship among nodes. Regarding the complex relationships in the multiplex graph, traditional graph representation learning (GRL) methods generally lack the capability to process it effectively [Fan *et al.*, 2019; Yang *et al.*, 2023; Liang *et al.*, 2024]. Therefore, multiplex graph representation learning (MGRL) methods are proposed to learn the low-dimensional node representations by mining the hidden information in the multiplex graph [Yu *et al.*, 2021], and have been adapted to various real-world applications, *e.g.,* social media analysis, community anomaly detection, and recommendation systems [Chen *et al.*, 2022; Xie *et al.*, 2022].

Existing MGRL methods can be broadly categorized into two groups, *i.e.,* node feature-free methods and node feature-based methods. Specifically, the former generally employ the structural information only to obtain node representations. For example, Metapath2vec [Dong *et al.*, 2017] employs random walk to obtain the sequences on metapaths and generate node representations. MNE [Zhang *et al.*, 2018] unifies structural information among various relationships into a shared representation space by random walk. However, these methods ignore the discriminative information in the node features, degrading node representations. Therefore, recent works are proposed to consider both node feature information and structural information in the multiplex graph. For example, MGCN [Ghorbani *et al.*, 2019] combines structural information with node features under the framework of graph convolutional networks. HAN [Wang *et al.*, 2019] aggregates node features using node-level attention and semantic-level attention and MAGNN [Fu *et al.*, 2020] aggregates meta-path ends and intermediates node features.

Despite effectiveness, most of previous MGRL methods rely on a basic assumption, *i.e.,* the original graph structure is reliable. This is generally unrealistic as graph structures are inevitably noisy or missing [Kang *et al.*, 2019; Zhang and He, 2023]. To mitigate it, some MGRL works adopt graph structure learning in traditional GRL methods to improve the model performance. For example, HGSL [Zhao *et al.*, 2021] utilizes the feature similarity to obtain the superior compositional graph structure. However, these methods still have the following limitations. First, these works optimize the graph structure with the feature similarity to measure whether nodes belong to the same class or not, so it cannot effectively capture negative relationships among nodes. Moreover, the feature similarity focuses on single pairs of nodes and cannot effectively capture the global relationships among each node and all nodes simultaneously. Second, these works optimize the parameters of graph structure learning by directly depending on the classification task. As a result, this may lead to edge starvation problem, *i.e.,* the representation of edges is obtained by receiving less help of supervision information, leading to poor generalization ability [Fatemi *et al.*, 2021]. Based on the above analysis, capturing globally positive and negative relationships among nodes, as well as separately conducting graph structure learning and representation learning, helps to enhance the MGRL performance.

To address the above challenges, in this paper, we propose a new MGRL framework, *i.e.,* **M**ultiplex **G**raph Representation Learning via **B**i-level **O**ptimization (MGBO), as shown

*Corresponding author.

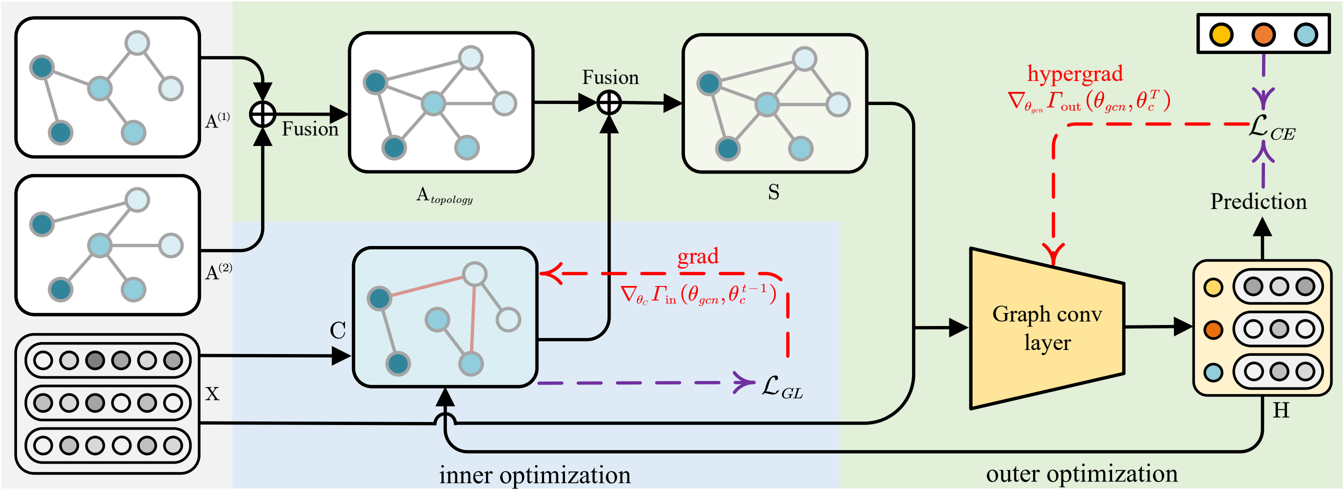


Figure 1: The flowchart of the proposed MGBO. Specifically, in the inner level (blue part), MGBO first generates a self-expression matrix $\mathbf{C}$ based on node features, and then investigates the graph learning loss $\mathcal{L}_{GL}$ to optimize $\mathbf{C}$ to capture the globally positive and negative relationships among nodes. MGBO further aggregates the graph structures and the self-expression matrix to obtain the aggregated graph $\mathbf{S}$. Finally, in the outer level (green part), $\mathbf{S}$ and node features are inputted to the graph convolutional layers to generate the final node representation $\mathbf{H}$, which is optimized by the cross-entropy loss $\mathcal{L}_{CE}$.

in Figure 1. Specifically, we utilize the self-expression matrix to adaptively capture the globally positive and negative relationships among nodes and complement them with the local relationships in graph structures. Moreover, we consider the learning of self-expression matrix and the representation learning as a bi-level optimization problem. That is, the inner level optimizes the self-expression matrix by learning the globally positive and negative relationships among nodes, while the outer level optimizes the representation learning parameters by the classification task based on the optimized graph structure. In addition, the node representations generated in the outer level is used in the inner level to adaptively optimize the self-expression matrix. In this way, graph structure learning is detached from the directly supervision of the node classification task, so edge starvation problem are solved, and the interdependence between graph structure learning and representation learning is fully utilized.

Compared to previous MGRL methods, the contributions of the proposed method can be summarized as:

- We utilize the self-expression property to adaptively capture the globally positive and negative relationships among nodes, achieving the effectiveness MGRL.
- To the best of our knowledge, we propose the first bi-level optimization for MGRL, where the self-expression matrix and the representations learning are optimized in the inner and outer level of the model, respectively, to solve the edge starvation problem.
- We investigate a new algorithm for the proposed bi-level optimization. Comprehensive experimental results on multiple benchmark datasets demonstrate the effectiveness of our method on node classification tasks, compared to nine comparison methods.

2 Related Work

2.1 Graph Structure Learning

Graph structure learning aims to optimize the graph structure for better adaptation to downstream tasks. Early methods such as LDS [Franceschi *et al.*, 2019] typically modeled each edge using Bernoulli random variables and created graph structures by sampling from these distributions. However, these direct parameterizations of the adjacency matrix are time and space consuming [Zhao *et al.*, 2023]. An alternative approach is to provide complementary information to the graph structure based on node feature similarity. For example, IDGL [Chen *et al.*, 2020] iteratively learns metrics to generate graph structures from node features and GNN embeddings. GLCN [Jiang *et al.*, 2019] combines graph learning and graph convolution to learn graph structure through the smoothness of node features. ProGNN [Jin *et al.*, 2020] iteratively reconstructs the clean graph by preserving the low rank, sparsity, and feature smoothness properties of a graph. In addition, MV-GCN [Yuan *et al.*, 2021] proposes to improve the performance of the model by using local structure, global structure, and feature similarity. Recently, HGSL proposed graph structure learning for heterogeneous graphs via feature similarity graphs, feature propagation graphs and semantic similarity graphs.

2.2 Multiplex Graph Representation Learning

MGRL aims to obtain low-dimensional and discriminative node representations by mining information from multiplex graph structures [Mo *et al.*, 2023b]. Depending on whether or not node features are used when performing graph representation learning, previous MGRL works can be categorized into two types, *i.e.,* node feature-free methods and node feature-based methods.

Early MGRL methods usually fall into the category of node feature-free methods. For example, Metapath2vec [Dong *et al.*, 2017] utilizes metapath-based random wandering with skip-gram method to obtain node representations.GATNE-T [Cen *et al.*, 2019]fuses base embeddings and edge embeddings to form holistic embeddings and learns them through the skip-gram method. In addition, MNE [Zhang *et al.*, 2018] unifies information from various relation types into a shared representation space by metapath-based random walk. However, the performance of these MGRL methods is always unsatisfactory because these methods ignore the discriminative information contained in the node features.

Therefore, subsequent work attempts to improve model performance by considering both node features and structural information. For example, HAN [Wang *et al.*, 2019] uses the attention mechanism to aggregate node features to form node-level representations and semantic-level representations. FAME [Liu *et al.*, 2020] captures meta-paths with multiple inter-node relationships and higher-order topologies, and combines node features to learn low-dimensional representations of nodes. MAGNN [Fu *et al.*, 2020] proposes to aggregate node features within and across meta-paths to improve the performance. MHGCN [Yu *et al.*, 2022] uses multilayer graph convolution modules to automatically capture short and long metapath interactions across multiple relations to learn node representation. In addition, BPHGNN [Fu *et al.*, 2023] captures locally and globally relevant information about a node from both depth and breadth patterns.

3 Method

Notations. Let $\mathcal{G} = \{\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \ldots, \mathcal{G}^{(\mathcal{R})}\} = \{\mathcal{V}, \mathcal{E}, \mathrm{X}\}$ to denote the multiplex graph, where $\mathcal{G}^{(r)} = \{\mathcal{V}, \mathcal{E}^{(r)}, \mathrm{X}\}$ is the r-th graph in the multiplex graph, and $\mathcal{R}$ is the number of graphs, $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$ and $\mathcal{E} = \bigcup \mathcal{E}^{(r)} \subseteq \mathcal{V} \times \mathcal{V}$ represent the nodes set and the edges set of the multiplex graph, respectively, $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N} \in \mathbb{R}^{N \times F}$ denotes the node feature matrix, where $\mathbf{x}_i \in \mathbb{R}^F$ is the feature of the node v_i out of N nodes, and denote the set of structural information as $\mathcal{A} = \{\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(\mathcal{R})}\}$, where $\mathbf{A}^{(r)} \in \{0,1\}^{N \times N}$ is the adjacency matrix of the network $\mathcal{G}^{(r)}$, and $a_{ij}^{(r)} = 1$ if $(v_i, v_j) \in \mathcal{E}^{(r)}$. The goal of MGRL is to obtain fused discriminative node representations $\mathbf{H} \in \mathbb{R}^{N \times d}$ with node features and multiple graph structures, where $d \ll F$ is the dimension of the representation space.

3.1 Motivation

Previous MGRL methods usually adopt the graph structure learning in traditional GRL methods to optimize the graph structure by directly utilizing similarity between node features, thus improving the performance of the model. However, they usually neglect to capture the globally positive and negative relationships among nodes, resulting in sub-optimal model performance. Taking a movie multiplex graph dataset containing co-director relationship and co-actor relationship as an example. Different classes of movies can be connected by common actors and directors, but the edges between different classes of movies are not conducive for representation learning. However, the method based on the similarity of node features is only able to find the edges between movies of the same classes but not able to eliminate the edges between movies of different classes. As a result, node information propagation between two nodes from different classes may reduce node discriminability.

Furthermore, previous work usually uses a node classification task to optimize both graph structure learning and representation learning parameters, and the update of the graph structure receives supervision from the node classification task, resulting in uneven training of the graph structure. Take a training scenario with a two-layer GCN as an example, node information can only be propagated for a range of two hops. If there are no labeled nodes within two hops of an unlabeled node, the edges around the unlabeled node will lack supervision information. As a result, the updated graph structures are more inclined to fit the training set data, leading to edge starvation problem.

To address these issues, our proposed MGBO captures the globally positive and negative relationships among nodes through a self-expression matrix.

In addition, we use bi-level optimization to solve the edge starvation problem. We show the framework of the proposed method in Figure 1 and introduce the details as follows.

3.2 Self-Expression Learning

As mentioned above, previous MGRL works usually capture the relationships among nodes with the node feature similarity to mitigate the graph structure reliability assumption, as the correlation information among node features has been shown to provide complementarity to the graph structure [Yuan *et al.*, 2021]. However, directly utilizing the similarity graph fails to capture negative relationships between two nodes because the similarity between nodes is always non-negative. Moreover, the similarity graph is usually derived by calculating the similarity between every node and its neighbors and cannot capture the global relationships among nodes. As a result, previous MGRL methods may lose discriminative information that captures the intrinsic structure of the data [Tenenbaum *et al.*, 2000]. To solve these issues, in this paper, we utilize the self-expression property among node features to adaptively capture globally positive and negative relationships among nodes.

To do this, given the node feature matrix $\mathbf{X}$, we initialize the similarity graph by the K-Nearest Neighbor (KNN) algorithm [Wu *et al.*, 2020]. Specifically, we first obtain the feature similarity matrix $\mathbf{M}$ between each node pair by calculating the distance among all node pairs, *i.e.,*

$$\mathbf{M}_{ij} = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{|\mathbf{x}_i|\,|\mathbf{x}_j|}, \tag{1}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are node features of v_i and v_i. Then we select k nodes with the highest similarity to each node as their neighbors in the distance matrix $\mathbf{M}$, and thus obtaining the initial feature similarity graph $\mathbf{C}$.

To further capture the negative relationships among nodes with different labels and the positive relationships among nodes with the same label, we propose to optimize the similarity graph with the self-expression property of node features. Specifically, the self-expression property assumes that each data point can be linearly reconstructed from a weighted combination of all data points. More formally, given features $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ of the nodes set $\mathcal{V}$, for any $v_j \in \mathcal{V}$, there exists a coefficient $c_{ij} \in \mathbb{R}$ such that:

$$\mathbf{x}_j = \sum_{i \neq j} \mathbf{x}_i c_{ij}, \tag{2}$$

where c_{ij} is the self-expression coefficient corresponding to v_i for reconstructing v_j from all other nodes. As a result, each node can be represented as linear combinations of all other nodes with positive or negative coefficients.

Therefore, we propose to optimize the similarity graph to capture globally positive and negative relationships among nodes with the self-expression property, *i.e.,*

$$\min_{\mathbf{C}} \left\| \mathbf{X}^\top - \mathbf{X}^\top \mathbf{C} \right\|_F^2, \tag{3}$$

As a result, Eq. (3) enforces the self-expression matrix to describe each node by all nodes.

However, the self-expression matrix may easily obtain the trivial solution, *i.e.,* the identity matrix. Therefore, we further consider two regularization terms in the above objective function to avoid the trivial solution, *i.e.,*

$$\mathcal{L}_{GL} = \left\| \mathbf{X}^\top - \mathbf{X}^\top \mathbf{C} \right\|_F^2 + \alpha \|\mathbf{C}\|_F^2 + \sum_{r=1}^{\mathcal{R}} \beta^{(r)} \left\| \mathbf{C} - \mathbf{A}^{(r)} \right\|_F^2, \tag{4}$$

where α and $\beta^{(r)}$ are two non-negative parameter to balance the second term and the third term. In Eq. (4), the second term aims to regularize the sparsity of the self-expression matrix, while the third term aims to optimize the self-expression matrix with the guidance of graph structures to avoid the trivial solution. After that, we let $\mathbf{C} = (\mathbf{C} + \mathbf{C}^\top)/2$ to ensure that the learned self-expression matrix is symmetric.

In this way, each element of the self-expression matrix is not constrained to be non-negative, thus adaptively capturing positive and negative relationships between two nodes. Moreover, the self-expression matrix indeed relies on all data points to describe each node, unlike previous MGRL works that only calculate the similarity between the node and its nearby nodes. As a result, the self-expression matrix also captures the global relationships of the nodes.

3.3 Graph Fusion

In the multiplex graph, an edge may connect two nodes from different classes and two nodes in the same class may not be connected. To solve these issues, in this section, we propose to fuse the graph structure with the self-expression matrix in our method. We expect to use the positive relationship among two nodes to connect two nodes in the same class and use the negative relationship among two nodes to disconnect two nodes from different classes.

To do this, we first fuse all graph structures in the multiplex graph. A simple approach would be to use the average pooling to aggregate information from different graph structures. However, such an approach is counter-intuitive because different graph structures represent different relationships among nodes, thus the importance of different graph structures is different. Therefore, we utilize the attention mechanism to learn the weights of different graph structures and perform aggregation.

Specifically, given multiple graph structures $\mathcal{A} = \{\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(\mathcal{R})}\}$, we obtain the fused graph structure $\mathbf{A}_{topology}$ with the graph-level attention mechanism, *i.e.,*

$$\mathbf{A}_{topology} = \Psi_1 \left(\left[\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(\mathcal{R})} \right] \right), \tag{5}$$

where $\left[\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(\mathcal{R})} \right]$ is the stacking matrix of all graph structures. Ψ_1 denotes a channel attention layer whose weight matrix $\mathbf{W}_{\Psi 1} \in \mathbb{R}^{1 \times 1 \times \mathcal{R}}$ indicates the importance of different graph structures.

After the attention mechanism, we further employ an attention layer to fuse the graph structure $\mathbf{A}_{topology}$ and the self-expression matrix $\mathbf{C}$ by:

$$\mathbf{S} = \Psi_2 \left([\mathbf{A}_{topology}, \mathbf{C}] \right), \tag{6}$$

where $[\mathbf{A}_{topology}, \mathbf{C}]$ is a stacked matrix fused the graph structure with the self-expression matrix. Ψ_2 is the channel attention layer whose weight matrix $\mathbf{W}_{\Psi 1} \in \mathbb{R}^{1 \times 1 \times 2}$ indicates the importance of fused graph structure and self-expression matrix. Therefore, we obtain the aggregated graph $\mathbf{S}$ and further let $\mathbf{S} = (\mathbf{S} + \mathbf{S}^\top)/2$ to ensure that the aggregated graph is symmetric.

3.4 Representation Learning

Given the node feature matrix $\mathbf{X}$ and the aggregated graph $\mathbf{S}$, we employ the graph convolutional layer $g : \mathbb{R}^{N \times F} \times \mathbb{R}^{N \times N} \to \mathbb{R}^{N \times d}$ to obtain node representations $\mathbf{H}$ by:

$$\mathbf{H} = \sigma(\widehat{\mathbf{S}} \mathbf{X} \Theta), \tag{7}$$

where σ is the activation function, and Θ is the weight matrix of the encoder g. $\widehat{\mathbf{S}} = \hat{\mathbf{D}}^{-1/2} (\mathbf{S} + w\mathbf{I}_N) \hat{\mathbf{D}}^{-1/2} \in \mathbb{R}^{N \times N}$ is the symmetrically normalized graph structure of aggregated graph, $\hat{\mathbf{D}}$ is the degree matrix of $\mathbf{S} + w\mathbf{I}_N$, and w is the weight of identity matrix $\mathbf{I}_N$.

In Eq. (7), $\mathbf{H}$ aggregates neighbors information from original topology and node features space with $\mathbf{S}$. Given the node representations $\mathbf{H}$, we further train our model by minimizing the cross-entropy loss between true and predicted labels. To do this, we employ a fully connected layer to obtain the predicted class based on the node representation $\mathbf{H}$, *i.e.,*

$$\hat{\mathbf{Y}} = softmax(\mathbf{W}\mathbf{H}), \tag{8}$$

where $\mathbf{W}$ indicates the parameter of the fully connected layer, and $\hat{\mathbf{Y}}$ is the class of nodes predicted by the classifier. Therefore, the cross-entropy loss can be formulated as:

$$\mathcal{L}_{CE} = - \sum_{l \in \mathbf{Y}_L} \mathbf{Y}_l \cdot ln(\hat{\mathbf{Y}}_l), \tag{9}$$

where $\mathbf{Y}_L$ is the set of node indices with labels, $\mathbf{Y}_l$ and $\hat{\mathbf{Y}}_l$ are the labels and prediction class of labeled nodes. As a result, the label information is adopted to guide the training process with the cross-entropy loss so that meaningful node representations $\mathbf{H}$ can be learned.

Based on the optimized node representations $\mathbf{H}$, we consider replacing the original node features in Eq. (4) with the learned $\mathbf{H}$. The reason can be attributed as follows. On the one hand, the noise or redundancy in the original node features may make it difficult for the self-expression matrix to capture the positive and negative relationships among nodes accurately [Lin *et al.*, 2024]. On the other hand, the original node features lack sufficient discriminability, which affects the quality of the self-expression matrix. As a result, with the optimized node representations $\mathbf{H}$, the self-expression learning module can be designed in a dynamically optimized mode, and Eq. (4) can be reformulated as:

$$\mathcal{L}_{GL} = \left\|\mathbf{H}^\top - \mathbf{H}^\top\mathbf{C}\right\|_F^2 + \alpha\|\mathbf{C}\|_F^2 + \sum_{r=1}^{\mathcal{R}} \beta^{(r)} \left\|\mathbf{C} - \mathbf{A}^{(r)}\right\|_F^2. \tag{10}$$

Therefore, the self-expression matrix $\mathbf{C}$ can be optimized by minimizing $\mathcal{L}_{GL}$ based on the learned node representation $\mathbf{H}$.

3.5 Bi-level Optimization

With the graph structure learning in Section 3.2 and the representation learning in Section 3.4, previous MGRL works generally optimize the parameters of them jointly. As a result, these methods may lead to the edge starvation problem, as the representation learning of edges far from the labeled nodes cannot be supervised by the node classification task, while the representation learning of edges close to the labeled nodes are supervised. The resulting learned graph structure tends to fit the data of labeled nodes (*i.e.,* the training set), leading to poor generalization to unlabeled nodes.

To address the above issues, the proposed method considers optimizing the parameters of graph structure learning and the parameters of representation learning through a bi-level manner, which has been widely used in meta-learning and model selection [Franceschi *et al.*, 2018].

To do this, we formulate the representation learning and the self-expression learning, respectively, as an outer optimization task and an inner optimization task. Specifically, denoting the parameters as θ_{gcn} and θ_c, respectively, the outer optimization and the inner optimization can be optimized by the following objective functions, *i.e.,*

$$\mathbf{\Gamma}_{\text{out}}(\theta_{gcn}, \theta_c) = \mathcal{L}_{CE}, \quad \mathbf{\Gamma}_{\text{in}}(\theta_{gcn}, \theta_c) = \mathcal{L}_{GL}, \tag{11}$$

where $\mathbf{\Gamma}_{\text{out}}$ aims to optimize parameters of the representation learning module and $\mathbf{\Gamma}_{\text{in}}$ aims to optimize the parameters of the self-expression learning module. Particularly, the above objective functions can be formulated as the following bi-level optimization problem, *i.e.,*

$$\min_{\theta_{gcn}} \mathbf{\Gamma}_{\text{out}}(\theta_{gcn}, \theta_c) \quad s.t. \quad \theta_c = \arg\min_{\theta_c} \mathbf{\Gamma}_{\text{in}}(\theta_{gcn}, \theta_c). \tag{12}$$

For the above bi-level optimization, existing literature [Liu *et al.*, 2021] usually first unrolls the inner optimization dynamics T steps, and then compute the hypergradient (outer gradient) from the unrolled dynamics. In this paper, we conduct the inner optimization process by the stochastic gradient descent method [Amari, 1993], *i.e.,*

$$\begin{gathered}\theta_c^t = \psi_t\left(\theta_c^{t-1}; \theta_{gcn}\right), t = 1, \cdots, T, \\ \text{where} \\ \psi_t\left(\theta_c^{t-1}; \theta_{gcn}\right) = \theta_c^{t-1} - \eta_t \nabla_{\theta_c} \mathbf{\Gamma}_{\text{in}}(\theta_{gcn}, \theta_c^{t-1}),\end{gathered} \tag{13}$$

where ψ_t denotes the update scheme at the t-th step of the inner optimization process, T denotes the total number of iterations of the inner optimization, and η_{t} is the learning rate for the inner optimization. After T steps, we can formulate the inner parameters as:

$$\theta_c^T = \psi_T \circ \cdots \circ \psi_2 \circ \psi_1(\theta_{gcn}). \tag{14}$$

where $\circ$ denotes the composite dynamics operation for the entire iteration. Therefore, the outer optimization is:

$$\min_{\theta_{gcn}} \mathbf{\Gamma}_{\text{out}}(\theta_{gcn}, \theta_c^T). \tag{15}$$

Based on Eq. (15), to update the outer parameters (*i.e.,* θ_{gcn}), we need to compute the hypergradient $\nabla_{\theta_{gcn}} \mathbf{\Gamma}_{\text{out}}(\theta_{gcn}, \theta_c^T)$. Recalling that $\theta_c^t = \psi_t\left(\theta_c^{t-1}; \theta_{gcn}\right)$, the operation ψ_t is obviously dependent on θ_{gcn} directly. Moreover, the operation ψ_t is dependent on θ_{gcn} indirectly through θ_c^{t-1}, which is updated based on θ_{gcn}. Thus, the hypergradient of Eq. (15) can be computed by the chain rule, *i.e.,*

$$\begin{aligned}\nabla_{\theta_{gcn}} \mathbf{\Gamma}_{\text{out}}(\theta_{gcn}, \theta_c^T) \quad = \quad & \frac{\partial \mathbf{\Gamma}_{\text{out}}(\theta_{gcn}, \theta_c^T)}{\partial \theta_{gcn}} + \\ & \frac{\partial \mathbf{\Gamma}_{\text{out}}(\theta_{gcn}, \theta_c^T)}{\partial \theta_c^T} * \left.\frac{\partial \theta_c^T}{\partial \theta_{gcn}}\right|_{(\theta_{gcn}, \theta_c^T)}.\end{aligned} \tag{16}$$

In Eq. (16), the first term indicates the direct gradient, which can be obtained directly. The second term indicates the indirect gradient, which is difficult to obtain by direct computation, especially the parameter Jacobian $\frac{\partial \theta_c^T}{\partial \theta_{gcn}}$. To address this issue, we further expand it as:

$$\frac{\partial \theta_c^T}{\partial \theta_{gcn}} = \frac{\partial \psi_T(\theta_c^{T-1}; \theta_{gcn})}{\partial \theta_c^{T-1}} \frac{\partial \theta_c^{T-1}}{\partial \theta_{gcn}} + \frac{\partial \psi_T(\theta_c^{T-1}; \theta_{gcn})}{\partial \theta_{gcn}}. \tag{17}$$

Obviously, Eq. (17) can be solved by the superposition of the gradients of T rounds.

By optimizing Eq. (13) and Eq. (15), the bi-level optimization considers the graph structure learning and the representation learning, respectively, as the inner optimization and the outer optimization. Moreover, the graph structure learning module is only optimized by the self-expression property at the inner optimization instead of the node classification task at the outer optimization. As a result, the graph structure module avoids the overfitting issue for labelled nodes and the underfitting issue for unlabeled nodes. Therefore, the proposed method avoids the generation of starved edges to effectively solve the edge starvation problem. This improves the generalization of the model over unlabeled nodes.

4 Experiments

In this section, we conduct extensive experiments on 4 public benchmark datasets to evaluate effectiveness of the proposed method, compared to 9 comparison methods, on the node classification task.

Method	ACM		DBLP		IMDB		Freebase	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
GCN	89.9 ± 0.3	89.7 ± 0.2	89.0 ± 0.2	89.7 ± 0.1	49.2 ± 0.3	49.3 ± 0.4	49.5 ± 0.2	50.3 ± 0.4
GAT	90.1 ± 0.2	90.4 ± 0.3	89.7 ± 0.3	90.5 ± 0.3	49.7 ± 0.3	55.7 ± 0.3	53.4 ± 0.1	57.9 ± 0.5
HAN	91.0 ± 0.1	90.9 ± 0.2	92.8 ± 0.4	93.4 ± 0.2	58.6 ± 0.3	58.5 ± 0.4	54.3 ± 0.5	56.3 ± 0.6
GTN	91.3 ± 0.2	91.2 ± 0.2	93.5 ± 0.4	94.1 ± 0.3	57.9 ± 0.2	57.8 ± 0.2	52.6 ± 0.2	55.7 ± 0.2
MAGNN	91.4 ± 0.1	91.1 ± 0.3	93.3 ± 0.2	93.8 ± 0.1	59.3 ± 0.2	59.7 ± 0.3	57.6 ± 0.4	59.0 ± 0.3
MAGNN-AC	92.3 ± 0.3	92.2 ± 0.2	93.6 ± 0.2	94.1 ± 0.4	59.7 ± 0.2	60.0 ± 0.2	59.8 ± 0.2	62.3 ± 0.2
HGSL	92.6 ± 0.2	92.5 ± 0.2	93.7 ± 0.3	94.2 ± 0.2	58.8 ± 0.2	58.5 ± 0.2	61.4 ± 0.6	65.7 ± 0.5
MGDCR	92.4 ± 0.3	92.2 ± 0.4	93.9 ± 0.2	94.3 ± 0.3	59.6 ± 0.1	60.2 ± 0.2	61.8 ± 0.7	67.3 ± 0.4
MHGCN	92.7 ± 0.2	92.4 ± 0.1	94.2 ± 0.1	94.5 ± 0.4	60.3 ± 0.3	60.5 ± 0.4	**62.5 ± 0.6**	67.1 ± 0.5
MGBO	**93.3 ± 0.3**	**93.2 ± 0.4**	**95.2 ± 0.3**	**95.6 ± 0.3**	**60.7 ± 0.3**	**61.4 ± 0.2**	61.5 ± 0.7	**67.5 ± 0.8**

Table 1: Classification performance (*i.e.,* Macro-F1 and Micro-F1) of all methods on all datasets.

4.1 Experimental Setup

Datasets

The used datasets include 2 citation multiplex graph datasets, *i.e.,* ACM [Jin *et al.*, 2021] and DBLP [Jin *et al.*, 2021], and two movie multiplex graph datasets, *i.e.,* IMDB [Jin *et al.*, 2021] and Freebase [Mo *et al.*, 2023a].

Comparison Methods

The comparison methods include 2 single-view graph methods and 7 multiplex graph methods. Single-view graph methods include 2 baseline methods, *i.e.,* GCN [Kipf and Welling, 2017] and GAT [Velickovic *et al.*, 2018]. Multiplex graph methods include HAN [Wang *et al.*, 2019], GTN [Yun *et al.*, 2019], MAGNN [Fu *et al.*, 2020], MAGNN-AC [Jin *et al.*, 2021], HGSL [Zhao *et al.*, 2021], MGDCR[Mo *et al.*, 2023a], and MHGCN [Yu *et al.*, 2022]). For single-view graph methods, we follow previous works [Wang *et al.*, 2019] to separately learn the representations for every graph and report the best performance.

4.2 Effectiveness Analysis

We evaluate the effectiveness of the proposed method on the node classification task by reporting the results (*i.e.,* Macro-F1, and Micro-F1) of all methods on four datasets in Table 1. Obviously, our method achieves the best effectiveness on the node classification task.

First, our method performs substantially better than the single-view graph methods (*i.e.,* GCN and GAT). For example, our MGBO outperforms the best single-view graph method (*i.e.,* GAT) by an average of 6.2%, in terms of Macro-F1 and Micro-F1, on all datasets. This validates the superiority of multiplex graph methods as they can efficiently capture the complex relationships among multiple graphs.

Second, the proposed method achieves the best performance among all MGRL methods, followed by MHGCN, MGDCR, HGSL, MAGNN-AC, MAGNN, GTN, and HAN. For example, the proposed method achieves 0.5% average improvement, compared to the best performing semi-supervised method (*i.e.,* MHGCN), in terms of Macro-F1 and Micro-F1, on all datasets. This suggests that the correlations learned from node features by the proposed method can provide complementary information to graph structures and help to learn distinguishable node representations.

Third, the proposed method outperforms previous MGRL method that utilizes the graph structure learning (*i.e.,* HGSL) by an average of 1.3%, in terms of Macro-F1 and Micro-F1, on all datasets. The reason can be attributed to the fact that the proposed method effectively captures the globally positive and negative relationships among nodes through the self-expression matrix as well as solves the edge starvation problem with bi-level optimization.

4.3 Ablation Study

The proposed method utilizes cross-entropy loss (*i.e.,* $\mathcal{L}_{CE}$) and graph structure loss (*i.e.,* $\mathcal{L}_{GL}$) to optimize the node representation and self-expression matrix. Moreover, MGBO further addresses edge starvation problem through bi-level optimization. To demonstrate the effectiveness of different parts of the proposed framework, we investigate the classification performance of different combinations of these components in the model on all datasets and report the results in Table 2 . In addition, we also investigate if the self-expression matrix captures the globally positive and negative relationships in Figure 2 and if the bi-level optimization explores the edge starvation problem in Figure 3.

Effectiveness of Self-Expression Matrix

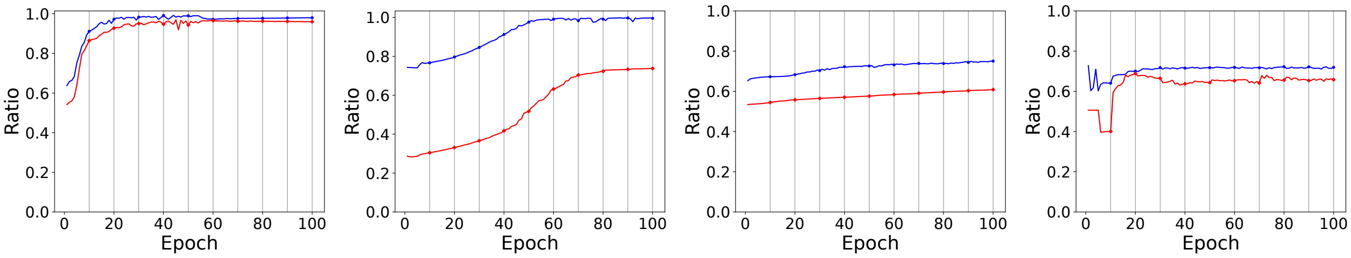
To verify the effectiveness of the self-expression matrix, we investigate the performance of the variant method with/without the graph structure loss (*i.e.,* $\mathcal{L}_{GL}$) on all datasets and report the results in Table 2. Obviously, the variant method with the $\mathcal{L}_{GL}$ achieves superior performance and obtains an average improvement of 2.1%, compared to the method without $\mathcal{L}_{GL}$. This indicates that the globally positive and negative relationships of nodes learned from the self-expression matrix can indeed provide complementary information about the graph structure and contribute to the learning of discriminative node representations.

Effectiveness for Self-Expression Matrix To Capture the Globally Positive and Negative Relationships

Intuitively, in the self-expression matrix, the edges with positive values indicate that the connected nodes may be from same class, while the edges with negative values indicate that the connected nodes may be from different classes. To verify that the self-expression matrix indeed captures the globally

$\mathcal{L}_{CE}$	$\mathcal{L}_{GL}$	ACM		DBLP		IMDB		Freebase	
		Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
√	–	90.5 ± 0.2	90.6 ± 0.2	92.9 ± 0.3	93.4 ± 0.2	58.3 ± 0.2	59.0 ± 0.2	56.8 ± 0.7	59.9 ± 0.8
√	√	**92.0 ± 0.4**	**91.9 ± 0.3**	**94.7 ± 0.4**	**95.0 ± 0.3**	**59.3 ± 0.4**	**59.8 ± 0.5**	**60.1 ± 0.8**	**65.4 ± 0.6**

Table 2: Classification performance (*i.e.,* Macro-F1 and Micro-F1) of variants with/without graph structure loss on all datasets.



(a) ACM (b) DBLP (c) IMDB (d) Freebase

Figure 2: Ratios of node pairs connected by edges with positive/negative values in the self-expression matrix belonging to same/different classes on all datasets, denoted as red and blue lines, respectively.

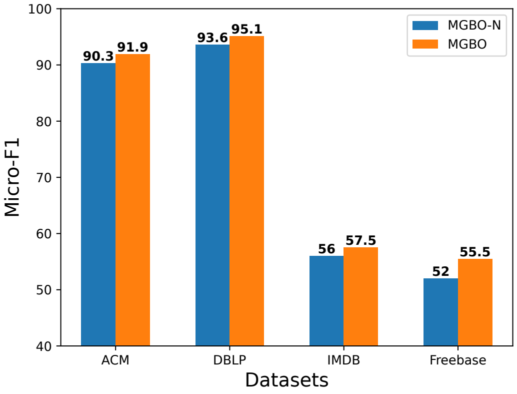


Figure 3: Classification performance (*i.e.,* Micro-F1) of variants with/without bi-level optimization on nodes with starved edges.

positive and negative relationships among nodes, we calculate the ratio of node pairs belonging to the same/different classes in edges with positive/negative values in the self-expression matrix and reported results on all datasets in Figure 2. Obviously, the globally positive and negative relationships among nodes can be captured using the self-expression matrix, and the ratios can continue to increase as the bi-level optimization proceeds. This verifies the effectiveness of the self-expression matrix as well as the bi-level optimization.

Effectiveness of the Bi-Level Optimization To Solve the Edge Starvation Problems

To verify that the bi-level optimization indeed solves the edge starvation problems, we first filter out nodes that were not connected to any labeled nodes, and the edges around these nodes were starved edges that lacked supervision information. We then test the performance of the variant methods with/without bi-level optimization (*i.e.,* MGBO and MGBO-N) on these nodes on all datasets and report the results in figure 3. As a result, the method with bi-level optimization is significantly better than the method without it, achieving an average improvement of 2.5%. This indicates that the bi-level optimization is effective in mitigating the edge starvation problems, thus improving the model's performance.

5 Conclusion

In this paper, we proposed a multiplex graph representation learning framework based on bi-level optimization. Specifically, we proposed to capture the globally positive and negative relationships among nodes by learning the self-expression matrix based on node features. In addition, we updated the parameters of the self-expression learning and the parameters of the representation learning separately through bi-level optimization, which effectively solves the edge starvation problem. Extensive experimental results show that the method consistently achieves state-of-the-art performance on the node classification task.

Acknowledgements

This work was supported in part by the National Key Research & Development Program of China under Grant 2022YFA1004100, Guangxi Natural Science Foundation (2023GXNSFBA026010), Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security (22-A-03-02), Innovation Project of Guangxi Graduate Education (XJCY2022009).

References

[Amari, 1993] Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, pages 185–196, 1993.

[Cen *et al.*, 2019] Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. Representation learning for attributed multiplex heterogeneous network. In *KDD*, pages 1358–1368, 2019.

[Chen *et al.*, 2020] Yu Chen, Lingfei Wu, and Mohammed Zaki. Iterative deep graph learning for graph neural networks: Better and robust node embeddings. *NeurIPS*, 33:19314–19326, 2020.

[Chen *et al.*, 2022] Bo Chen, Jing Zhang, Xiaokang Zhang, Yuxiao Dong, Jian Song, Peng Zhang, Kaibo Xu, Evgeny Kharlamov, and Jie Tang. Gccad: Graph contrastive learning for anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[Dong *et al.*, 2017] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*, pages 135–144, 2017.

[Fan *et al.*, 2019] Shaohua Fan, Junxiong Zhu, Xiaotian Han, Chuan Shi, Linmei Hu, Biyu Ma, and Yongliang Li. Metapath-guided heterogeneous graph neural network for intent recommendation. In *KDD*, pages 2478–2486, 2019.

[Fatemi *et al.*, 2021] Bahare Fatemi, Layla El Asri, and Seyed Mehran Kazemi. Slaps: Self-supervision improves structure learning for graph neural networks. *NeurIPS*, 34:22667–22681, 2021.

[Franceschi *et al.*, 2018] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *ICML*, pages 1568–1577, 2018.

[Franceschi *et al.*, 2019] Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning discrete structures for graph neural networks. In *ICML*, pages 1972–1982, 2019.

[Fu *et al.*, 2020] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *WWW*, pages 2331–2341, 2020.

[Fu *et al.*, 2023] Chaofan Fu, Guanjie Zheng, Chao Huang, Yanwei Yu, and Junyu Dong. Multiplex heterogeneous graph neural network with behavior pattern modeling. In *KDD*, pages 482–494, 2023.

[Ghorbani *et al.*, 2019] Mahsa Ghorbani, Mahdieh Soleymani Baghshah, and Hamid R Rabiee. Mgcn: semi-supervised classification in multi-layer graphs with graph convolutional networks. In *ASONAM*, pages 208–211, 2019.

[Jiang *et al.*, 2019] Bo Jiang, Ziyan Zhang, Doudou Lin, Jin Tang, and Bin Luo. Semi-supervised learning with graph learning-convolutional networks. In *CVPR*, pages 11313–11320, 2019.

[Jin *et al.*, 2020] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *KDD*, pages 66–74, 2020.

[Jin *et al.*, 2021] Di Jin, Cuiying Huo, Chundong Liang, and Liang Yang. Heterogeneous graph neural network via attribute completion. In *WWW*, pages 391–400, 2021.

[Kang *et al.*, 2019] Zhao Kang, Haiqi Pan, Steven CH Hoi, and Zenglin Xu. Robust graph learning from noisy data. *IEEE Transactions on Cybernetics*, 50(5):1833–1843, 2019.

[Kipf and Welling, 2017] N. Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, pages 1–14, 2017.

[Liang *et al.*, 2024] Ke Liang, Sihang Zhou, Meng Liu, Yue Liu, Wenxuan Tu, Yi Zhang, Liming Fang, Zhe Liu, and Xinwang Liu. Hawkes-enhanced spatial-temporal hypergraph contrastive learning based on criminal correlations. In *AAAI*, pages 8733–8741, 2024.

[Lin *et al.*, 2024] Han Lin, Yingjian Li, Zheng Zhang, Lei Zhu, and Yong Xu. Learning with noisy labels by semantic and feature space collaboration. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[Liu *et al.*, 2020] Zhijun Liu, Chao Huang, Yanwei Yu, Baode Fan, and Junyu Dong. Fast attributed multiplex heterogeneous network embedding. In *ACM CIKM*, pages 995–1004, 2020.

[Liu *et al.*, 2021] Risheng Liu, Jiaxin Gao, Jin Zhang, Deyu Meng, and Zhouchen Lin. Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):10045–10067, 2021.

[Mo *et al.*, 2023a] Yujie Mo, Yuhuan Chen, Yajie Lei, Liang Peng, Xiaoshuang Shi, Changan Yuan, and Xiaofeng Zhu. Multiplex graph representation learning via dual correlation reduction. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[Mo *et al.*, 2023b] Yujie Mo, Yajie Lei, Jialie Shen, Xiaoshuang Shi, Heng Tao Shen, and Xiaofeng Zhu. Disentangled multiplex graph representation learning. In *ICML*, 2023.

[Tenenbaum *et al.*, 2000] Joshua B Tenenbaum, Vin de Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[Velickovic *et al.*, 2018] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, pages 1–12, 2018.

[Wang *et al.*, 2019] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *WWW*, pages 2022–2032, 2019.

[Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions On Neural Networks and Learning Systems*, 32(1):4–24, 2020.

[Xie *et al.*, 2022] Yaochen Xie, Zhao Xu, Jingtun Zhang, Zhengyang Wang, and Shuiwang Ji. Self-supervised learning of graph neural networks: A unified review. *IEEE*

Transactions On Pattern Analysis and Machine Intelligence, 45(2):2412–2429, 2022.

[Yang *et al.*, 2023] Xihong Yang, Yue Liu, Sihang Zhou, Siwei Wang, Wenxuan Tu, Qun Zheng, Xinwang Liu, Liming Fang, and En Zhu. Cluster-guided contrastive graph clustering network. In *AAAI*, pages 10834–10842, 2023.

[Yu *et al.*, 2021] Junliang Yu, Hongzhi Yin, Min Gao, Xin Xia, Xiangliang Zhang, and Nguyen Quoc Viet Hung. Socially-aware self-supervised tri-training for recommendation. In *KDD*, pages 2084–2092, 2021.

[Yu *et al.*, 2022] Pengyang Yu, Chaofan Fu, Yanwei Yu, Chao Huang, Zhongying Zhao, and Junyu Dong. Multiplex heterogeneous graph convolutional network. In *KDD*, pages 2377–2387, 2022.

[Yuan *et al.*, 2021] Jinliang Yuan, Hualei Yu, Meng Cao, Ming Xu, Junyuan Xie, and Chongjun Wang. Semi-supervised and self-supervised classification with multi-view graph neural networks. In *ACM CIKM*, pages 2466–2476, 2021.

[Yun *et al.*, 2019] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *NeurIPS*, 32, 2019.

[Zhang and He, 2023] Zheng Zhang and Wen-Jue He. Tensorized topological graph learning for generalized incomplete multi-view clustering. *Information Fusion*, 100:101914, 2023.

[Zhang *et al.*, 2018] Hongming Zhang, Liwei Qiu, Lingling Yi, and Yangqiu Song. Scalable multiplex network embedding. In *IJCAI*, volume 18, pages 3082–3088, 2018.

[Zhao *et al.*, 2021] Jianan Zhao, Xiao Wang, Chuan Shi, Binbin Hu, Guojie Song, and Yanfang Ye. Heterogeneous graph structure learning for graph neural networks. In *AAAI*, pages 4697–4705, 2021.

[Zhao *et al.*, 2023] Jianan Zhao, Qianlong Wen, Mingxuan Ju, Chuxu Zhang, and Yanfang Ye. Self-supervised graph structure refinement for graph neural networks. In *WSDM*, pages 159–167, 2023.