Multiplex Graph Representation Learning with Homophily and Consistency

Yudi Huang[1], Ci Nie[1], Hongqing He[1], Yujie Mo[2], Yonghua Zhu[1,3*], Guoqiu Wen[1], Xiaofeng Zhu[1,2]

[1]Guangxi Key Lab of Multisource Information Mining Security, Guangxi Normal University, Guilin, China
[2]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China
[3]Information Systems Technology Design Pillar, Singapore University of Technology and Design, Singapore
yhzhu9423@gmail.com

Abstract

Although unsupervised multiplex graph representation learning (UMGRL) has been a hot research topic, existing UMGRL methods still has limitations to be addressed. For example, previous works either preserve structural information by ignoring the impact of heterophily in the graph structure or only focus on node-level consistency by ignoring class-level consistency. To address these issues, in this paper, we propose a new UMGRL method to explore both homophily and consistency in the multiplex graph. Specifically, we propose to restructure the multi-order relationships of every graph between every node and its multi-order neighbors to improve the homophily and reduce the impact of the heterophily in the graph structure. We also design a contrastive loss based on a self-expression matrix of the node representation to achieve node-level and class-level consistency. Furthermore, we theoretically prove our method to achieve class-level consistency. Extensive experimental results on real datasets verify the effectiveness of the proposed method with respect to node classification tasks, compared to SOTA methods.

1 Introduction

Multiplex graph consists of multiple graphs to share the same node representation. Benefiting from the richness and variety of graph structures, the relationships between two nodes in the multiplex graph can be fully and comprehensively described. Hence, multiplex graph learning has been popularly designed to conduct representation learning in various real-world applications such as social networks and biological networks (Wan et al. 2020; Peng et al. 2023). Recently, Unsupervised Multiplex Graph Representation Learning (UMGRL), as an effective method for modeling complex relationships of nodes without using node labels, has shown great potentiality in a wide range of applications such as anomaly detection and recommendation systems (Chu et al. 2019; Peng et al. 2022).

Existing UMGRL methods can be broadly categorized into two groups, *i.e.,* traditional unsupervised methods and self-supervised methods. Traditional UMGRL methods such as PMNE (Liu et al. 2017) and Metapath2vec (Dong, Chawla, and Swami 2017) aim to learn node representations without the help of labels by random walk strategies. However, these works only focus on learning structural information and ignore the discriminative information in the node representation. To address these issues, Self-supervised methods are proposed to make full use of node features and structural information for representation learning. Previous self-supervised methods can be categorized into two subgroups, *i.e.,* intra-graph contrastive learning methods and inter-graph contrastive learning methods. Intra-graph contrastive learning methods aims to improve the quality of node representations by capturing global attributes. For example, HDMI (Jing, Park, and Tong 2021) encourages improved mutual information between node representations and global representations to capture global attributes. However, this method ignores the intrinsic correlation among different graphs. Therefore, inter-graph contrastive learning methods have been proposed to address these issues by capturing consistent information across different graphs. For example, CKD (Wang et al. 2022) models the correlation among different graphs by performing collaborative distillation.

Although existing UMGRLs have achieved good results in many tasks, these works still have limitations to be addressed. First, previous UMGRL works mainly focus on exploring the homophily where two connected nodes come from the same class, but ignoring the impact of heterophily of the graph structure, where two connected nodes come from different classes. For example, previous UMGRL works (Mo et al. 2023b; Peng, Wang, and Zhu 2023) rely on either homophilic GNNs (*e.g.,* graph convolutional layers) or node representation to reconstruct the original graph for preserving the structural information. Their key idea is to keep representation smooth on the graph structure, which implicitly assumes the homophily of the graph structure. However, heterophily has been found to be prevalent in real-world scenarios. If the heterophily in the graph structure is ignored, the model will inevitably introduce noise (Zheng et al. 2022). Hence, existing methods significantly limit the rationality of real applications. Second, previous works mainly focus on exploring the consistency of the same node across different graphs, *i.e.,* node-level consistency, to obtain consistent information among graphs, which has been widely used to conduct node identification (Zhu et al. 2022).

*Corresponding author.

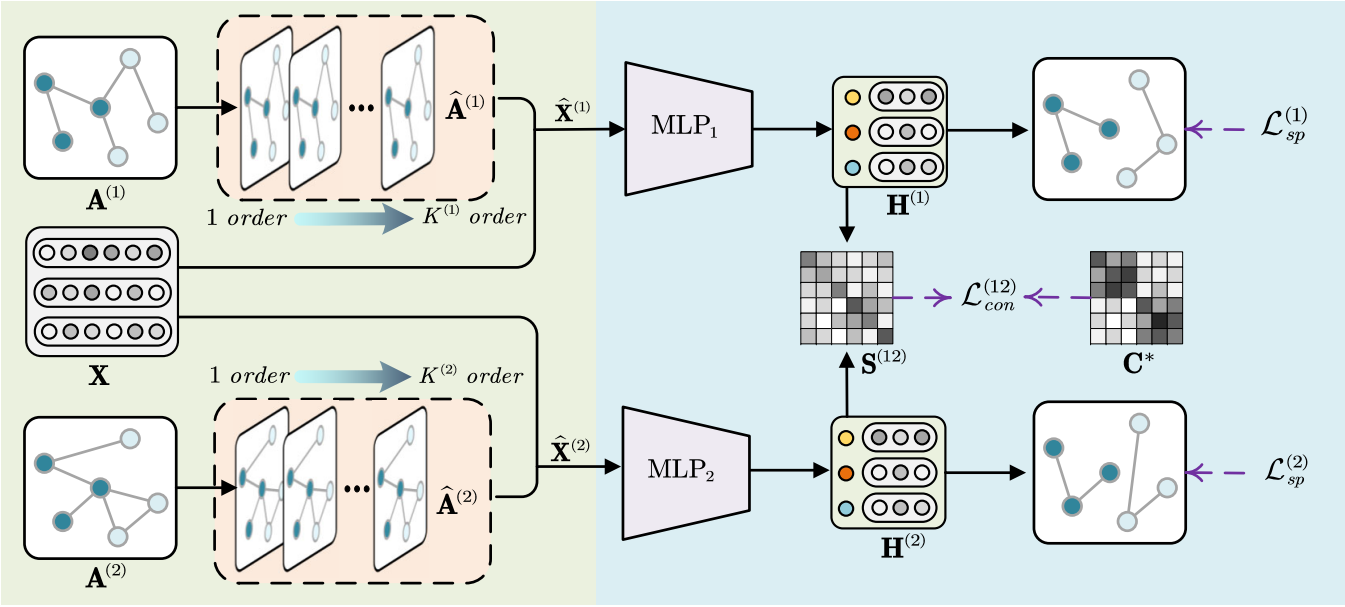


Figure 1: The flowchart of the proposed MGHC. Specifically, we first employ ASGC (Chanpuriya and Musco 2022) on $\mathbf{X}$ for every graph to generate its new representation $\widehat{\mathbf{X}}^{(r)}$ $(r = 1, ..., \mathcal{R})$ and obtain the correlation coefficients between every node and its multi-order neighbors, which improve the homophily and reduces the impact of heterophily. We then employ MLPs on $\widehat{\mathbf{X}}^{(r)}$ to further obtain its node representation $\mathbf{H}^{(r)}$. After that, we investigate the structure preservation loss $\mathcal{L}_{sp}^{(r)}$ to capture the structural information by $\widehat{\mathbf{A}}^{(r)}$. Meanwhile, we first concatenate all node representation $\mathbf{H}^{(r)}$ to obtain full-graph representation $\mathbf{H}$, which is then designed to capture the relationship of nodes from the same class by a self-expression matrix $\mathbf{C}^*$. We further compute the cross-graph similarity matrix $\mathbf{S}^{(rr')}$ $(r' = 1, ..., \mathcal{R})$, which and $\mathbf{C}^*$ are used to construct the contrastive loss $\mathcal{L}_{con}^{(rr')}$ for achieving node-level consistency and class-level consistency.

For example, HeCo aligns the node representation of the network schema view with the meta-path view to capture the node-level consistency (Wang et al. 2021). However, these methods are unable to make the distance between two nodes from the same class close, *i.e.,* ignoring the class-level consistency. This will increase the intra-class variance of node representation to affect node representation learning. Based on the above analysis, the UMGRL should be conducted by (1) capturing the homophily but reducing the impact of heterophily in graph structure and (2) exploring both the node-level consistency and the class-level consistency.

To address the above issues, in this paper, we propose a new UMGRL framework, *i.e.,* **M**ultiplex **G**raph Representation Learning with **H**omophily and **C**onsistency, MGHC for short as shown in Figure 1. Specifically, we first calculate the correlation coefficient between each node and its multi-order neighbors, which is then used to restructure the multi-order relationship of every graph. The correlation coefficient reflects the homophilic/heterophilic relationship between every node and its multi-order neighbors. The restructure process automatically updates the correlation coefficient to improve the homophily and reduce the heterophily, thus exploring the first issue in previous methods. We further investigate to capture the relationships between two nodes from the same class by learning a self-expression matrix, which is then used to construct a contrastive loss. As a result, the contrastive loss is able to make the same nodes across different graphs close as well as the distance between two nodes from the same class close, thus simultaneously achieving node-level and class-level consistency.

Compared to previous UMGRL methods, our main contributions are summarized as follows:

- We propose to restructure the graph structure based on the correlation coefficients between every node and its multi-order neighbors so that our method improves the homophily and reduces the impact of the heterophily of the graph structure.
- We investigate a contrastive loss based on the self-expression matrix to capture the relationship between two nodes from the same class, thus achieving both the node-level consistency and the class-level consistency.
- We theoretically demonstrate that the learned self-expression matrix has the grouping effect, thus capturing the relationships between two nodes from the same class, *i.e.,* class-level consistency.

2 Related Work

2.1 Unsupervised Multiple Graph Representation Learning

UMGRL aims to extract valuable information from multiple relationships among nodes without relying on label infor-

mation. Existing UMGRL methods can be categorized into two groups, *i.e.,* traditional unsupervised methods and self-supervised methods. Traditional unsupervised methods usually obtain node representation through random walk strategy. For example, MNE (Zhang et al. 2018) performs random walk in different meta-paths to extract various types of relationships into a common representation space. Metapath2vec (Dong, Chawla, and Swami 2017) obtains meta-path sequences by random walk and generates node representation by the skip-gram method. However, these works only focus on learning the structural information by ignoring discriminative information in node representation.

To solve these problems, self-supervised methods have been proposed. Self-supervised methods can be further categorized into intra-graph contrastive learning methods and inter-graph contrastive learning methods. Intra-graph contrastive learning methods encourage capturing global attributes to improve the quality of node representations. For example, DMGI (Park et al. 2020) and HDMI (Jing, Park, and Tong 2021) perform contrastive learning between node representations and global representations to capture global attributes. However, intra-graph contrastive learning methods learn information about each graph independently and fail to capture the intrinsic correlation among every graph, thus leading to sub-optimal model performance. Therefore, inter-graph contrastive learning methods have been proposed to address these issues. inter-graph contrastive learning methods aim to capture consistent information among different graphs. For example, HeCo (Wang et al. 2021) performs contrastive learning between network schema view and meta-path views to obtain local and higher-order structural information. STENCIL obtains inter-graph consistency by reducing the distance between each graph and the aggregated graph. Further, CPIM (Mo et al. 2023b) and CoCoMG (Peng, Wang, and Zhu 2023) achieve contrastive learning in different graphs without negative samples and successfully extract the consistency among different graphs.

2.2 Heterophilic Graph Neural Networks

Heterophilic graph neural networks (GNNs) aim to reduce the impact of heterophily in graph structure while performing data mining. Existing methods for heterophilic GNNs can be broadly classified into three groups, *i.e.,* non-local neighbor extension methods, GNN architecture refinement methods, and hybrid methods.

Non-local neighbor extension methods extend local neighbors to non-local neighbors to alleviate the impact of local heterophily by incorporating richer homophilic information from higher-order neighborhoods. For example, MixHop (Abu-El-Haija et al. 2019) considers multi-hop neighbors for message propagation and then integrates messages obtained from different hops. TDGNN (Wang and Derr 2021) directly aggregates information from neighbors of multi-order in the aggregation step to promote information fusion among different layers.

GNN architecture refinement methods boost the expressive ability of GNNs for heterophilic graphs by encouraging distinguishable and discriminative node representations. For example, GPRGNN (Chien et al. 2020) assigns learnable weights to adaptively combine the representations of each layer through PageRank. FAGCN (Bo et al. 2021) designs low-pass and high pass filters to achieve expressive performance of heterophilic graphs through adaptive frequency signal learning

Hybrid methods simultaneously extend non-local neighbor sets and improve heterophily-guided GNN architectures. For example, BM-GCN (He et al. 2022) explores block-guided neighbors and performs different types of classification aggregation on homophilic and heterophilic nodes. However, most of the above methods are applied to single-graph data and are not extended to multiplex graph data.

3 Method

Notations. Given the multiplex graph $\mathcal{G} = \{\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \ldots, \mathcal{G}^{(\mathcal{R})}\} = \{\mathcal{V}, \mathcal{E}, \mathrm{X}\}, \mathcal{G}^{(r)} = \{\mathcal{V}, \mathcal{E}^{(r)}, \mathrm{X}\}$ 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. We denote $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N} \in \mathbb{R}^{N \times F}$ as the original node representation matrix, where $\mathbf{x}_i \in \mathbb{R}^F$ is the representation of the node v_i out of N nodes, and denote the set of structural information of $\mathbf{X}$ 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 the proposed MGHC is to learn the encoders to map the original inputs into low-dimensional representation $\mathbf{Z} \in \mathbb{R}^{N \times d}$, where $d \ll F$ is the dimension of the representation space.

3.1 Homophily Improvement and Heterophily Reduction

Based on the homophily assumption of the graph structure, Previous UMGRL methods usually rely either homophilic GNNs or using node representation to reconstruct the graph structure for capturing the structural information. However, real-world graphs do not always obey the homophily assumption and often contain both homophily and heterophily. For example, in online transaction networks, fraudsters are likely to establish connections with customers instead of other fraudsters. In molecular networks, protein structures are likely to consist of different types of amino acids linked together. As a result, previous work may capture the structural information with noise to reduce the distinguishability of the node representation. To address this issue, we propose to improve homophily and reduce the impact of heterophily by first restructuring the multi-order relationships of every graph and then capturing the structural information through the structure preservation loss.

Specifically, given node representation $\mathbf{X}$ and multiple graph structures $\{\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(\mathcal{R})}\}$, we first integrate all multi-order neighbor information to obtain new node representation $\widehat{\mathbf{X}}^{(r)}$ for every graph $\mathbf{A}^{(r)}$, *i.e.,*

$$\widehat{\mathbf{X}}^{(r)} = \widehat{\mathbf{A}}^{(r)}\mathbf{X}, \tag{1}$$

where $\widehat{\mathbf{A}}^{(r)} = \sum_{k=0}^{K^{(r)}} \beta_k^{(r)} \mathbf{A}^{(r)k}$. $K^{(r)}$ is the maximum propagation order of $\mathbf{A}^{(r)}$. In this paper, $\mathbf{A}^{(r)k} =$

$\mathbf{A}^{(r)k-1}\mathbf{P}^{(r)}$, where the transition matrix $\mathbf{P}^{(r)}$ is defined as $\mathbf{P}^{(r)} = (\mathbf{D}^{(r)})^{-1}\mathbf{A}^{(r)}$ and $\mathbf{D}^{(r)}$ is the degree matrix of $\mathbf{A}^{(r)}$. $\beta_k^{(r)} \in \mathbb{R}^N$ is the correlation coefficient, representing the correlation between every node and its k-th order neighbor of the graph $\mathbf{A}^{(r)}$.

Based on ASGC (Chanpuriya and Musco 2022), Eq. (1) obtains the correlation coefficient $\beta_k^{(r)}$ by $\min_\beta \|(\sum_{k=0}^{K} \beta_k^{(r)}\mathbf{A}^{(r)k})\mathbf{X} - \mathbf{X}\|_2^2$. Intuitively, if $\mathbf{A}^{(r)k}$ is homophilic, every node in the r-th graph and its neighbors tend to have similar representation. As a result, the correlation coefficient $\beta_k^{(r)}$ between the node representation $\mathbf{X}$ and the representation of the k-th order neighbor $\mathbf{A}^{(r)k}\mathbf{X}$ is positive. If a graph is heterophilic, the representation of each node and its neighbors are significantly different. As a result, the correlation coefficient is negative. Hence, the correlation coefficient directly reflects the homophilic or heterophilic relationship between each node and its multi-order neighbors. Obviously, restructuring the multi-order relationship of each graph by the correlation coefficient (*i.e.,* $\widehat{\mathbf{A}}^{(r)}$) can automatically introduce homophilic neighbors (*i.e.,* positive $\beta^{(r)}$) as well as removing heterophilic neighbors (*i.e.,* negative $\beta^{(r)}$) for every node.

To calculate $\widehat{\mathbf{X}}^{(r)}$ and $\beta_k^{(r)}$, we need to determine the maximum propagation order $K^{(r)}$ of $\mathbf{A}^{(r)}$. If $K^{(r)}$ is set too small, it will not be able to aggregate useful information from the higher-order neighbors. By contrast, if $K^{(r)}$ is set too large, it will lead to over-smoothing. Previous work has usually set the same maximum propagation order for all graphs. However, this method is counter-intuitive because different graphs imply different structural information. In this paper, we design a heuristic algorithm to compute the maximum propagation order for each graph. To do this, we follow (Wang and Tu 2012) to define the "degrees of freedom" (DoF) of the graph $\mathbf{A}^{(r)}$ as follows.

$$DoF(\mathbf{A}^{(r)k}) = Tr(\mathbf{A}^{(r)k}). \tag{2}$$

Based on Eq. (2), with the increase of the values of k, $DoF(\mathbf{A}^{(r)k})$ begins to smoothly evolve, so that the similarity of $\mathbf{A}^{(r)k}$ has the most discriminative ability. Moreover, we set the stopping criteria as:

$$K^{(r)} = \min\{k : \nabla DoF(\mathbf{A}^{(r)k}) \leq \varepsilon\}, \tag{3}$$

where $\nabla DoF(\mathbf{A}^{(r)k})$ is a smoothness indicator of $DoF(\mathbf{A}^{(r)k})$, and ∇ is the Laplace operator, and. ε is a manually set threshold. Theorem 3.1 is further used to estimates the maximum propagation order $K^{(r)}$.

Theorem 3.1. *If we denote the eigenvalues of the transition matrix $\mathbf{P}^{(r)}$ as $\lambda_1 > \lambda_2 \ldots, > \lambda_n$, then the maximum propagation order $K^{(r)}$ can be estimated as:*

$$K^{(r)} = \lceil \frac{\ln(\frac{\varepsilon}{(N - DoF(\mathbf{P}^{(r)}))\|\mathbf{A}^{(r)} - \mathbf{A}^{(r)}\mathbf{P}^{(r)} - \mathbf{I}_N\|})}{\ln(\lambda_2)} \rceil, \tag{4}$$

where $\lceil \cdot \rceil$ is the ceiling function. $\mathbf{I}_N$ is the identity matrix. Based on Eq. (4) and Eq. (1), we obtain $\widehat{\mathbf{X}}^{(r)}$, which is further transferred to its new representation $\mathbf{H}^{(r)}$ by an MLP encoder, *i.e.,* $\mathcal{F}_{\theta^{(r)}}^{(r)} : \mathbb{R}^{N\times F} \to \mathbb{R}^{N\times d}$:

$$\mathbf{H}^{(r)} = \mathcal{F}_{\theta^{(r)}}^{(r)}(\widehat{\mathbf{X}}^{(r)}), \tag{5}$$

where $\theta^{(r)}$ is the parameters of the MLP. In this paper, the encoder parameters for each graph are not shared because each encoder is used to learn specific structural information.

Given the node representation $\mathbf{H}^{(r)}(r = 1, \ldots, \mathcal{R})$ of each graph, previous work usually encourages that the similarity among node representation converges to the original graph structure, thus capturing the structural information. For example, CoCoMG (Peng, Wang, and Zhu 2023) captures the structural information through the following structure preservation loss:

$$\mathcal{L}_{sp}^{(r)} = -\sum_{i=1}^{N} log\left(\mathbf{a}_{ij}^{(r)}\mathcal{D}(\mathbf{H}_i^{(r)}, \mathbf{H}_j^{(r)})\right), \tag{6}$$

where $\mathcal{D}(\mathbf{H}_i^{(r)}, \mathbf{H}_j^{(r)}) = \exp(\cos(\mathbf{H}_i^{(r)}, \mathbf{H}_j^{(r)}))$ is used to calculate the similarity between node v_i and node v_j. In Eq. (6), if node v_i and node v_j are connected in the graph $\mathbf{A}^{(r)}$, $\mathbf{H}_i^{(r)}$ is similar to $\mathbf{H}_j^{(r)}$, *i.e.,* thus capturing the structural information. However, due to the heterophily in the original graph structure, the captured structural information inevitably contains noise, which reduces the distinguishability of the node representation.

To solve this issue, we propose to replace $\mathbf{a}_{ij}^{(r)}$ in Eq. (6) with $\widehat{\mathbf{a}}_{ij}^{(r)}$ because $\widehat{\mathbf{A}}^{(r)}$ contains multi-order neighbor information and is able to distinguish the homophilic ($\widehat{\mathbf{a}}_{ij}^{(r)} > 0$) or heterophilic ($\widehat{\mathbf{a}}_{ij}^{(r)} < 0$) relationship between every node and its neighbors. Hence, our proposed structural preservation loss is defined as follows.

$$\mathcal{L}_{sp}^{(r)} = -\sum_{i=1}^{N} log\left(\widehat{\mathbf{a}}_{ij}^{(r)}\mathcal{D}(\mathbf{H}_i^{(r)}, \mathbf{H}_j^{(r)})\right). \tag{7}$$

Obviously, the structure preservation loss in Eq. (7) improves the similarity between node v_i and node v_j if the relationship between node v_i and node v_j is homophilic (*i.e.,* $\widehat{\mathbf{a}}_{ij}^{(r)} > 0$), and reduces the similarity between node v_i and node v_j if the relationship between node v_i and node v_j is heterophilic (*i.e.,* $\widehat{\mathbf{a}}_{ij}^{(r)} < 0$). In this way, two nodes with large $\mathbf{a}_{ij}^{(r)}$ obtain large similarity, *i.e.,* $\mathcal{D}(\mathbf{H}_i^{(r)}, \mathbf{H}_j^{(r)})$.

3.2 Node-level and Class-level Consistency

The high-quality structural information can be obtained by restructuring the multi-order relationships of every graph, which improve homophily and reduce the impact of heterophily in graph structures. However, learning the information of every graph independently cannot capture the consistency among graphs because the consistency among graphs reflects the intrinsic correlation among graphs.

To capture consistency among graphs, previous UMGRL methods align the representation of the same sample in different graphs, *i.e.,* node-level consistency. However, these works ignore the consistency among samples from the same class in different graphs, *i.e.,* class-level consistency. As a

result, previous work could not make the distance among samples from the same class close so that they may increase the intra-class variance of the node representation to affect node representation learning. To solve this issue, we investigate a contrastive loss based on a self-expression matrix to capture the relationship between two nodes from the same class. This makes our method achieving both the node-level consistency and the class-level consistency.

Specifically, we first obtain the full-graph representation $\mathbf{H}$ by concatenating the node representation of every graph to make it linearly separable, *i.e.,*

$$\mathbf{H} = [\mathbf{H}^{(1)}; \mathbf{H}^{(2)}; \dots; \mathbf{H}^{(\mathcal{R})}] \in \mathbb{R}^{N \times (\mathcal{R} \times d)}. \tag{8}$$

We then utilize the self-expression property of the full-graph representation $\mathbf{H}$ to capture the relationship between two nodes in the same class. Specifically, given the full-graph representation $\mathbf{H} = \{\mathbf{h}_1, \dots, \mathbf{h}_N\}$ of the nodes set $\mathcal{V}$, for any node $v_i \in \mathcal{V}$, the self-expression property assumes that each nodes can be linearly reconstructed by a weighted combination of all nodes, *i.e.,*

$$\mathbf{h}_i = c_{i1}\mathbf{h}_1 + \dots + c_{iN}\mathbf{h}_N, \tag{9}$$

where c_{ij} is the self-expression coefficient corresponding to node v_j for reconstructing node v_i. based on the literature (Chapelle, Weston, and Schölkopf 2002; Zhou et al. 2003), where nodes with similar representations are more likely to belong to the same class. Therefore, Each node in the self-expression matrix sets the nodes with similar representations as its neighbours, thus capturing the relationship between two nodes from the same class. We can compute the self-expression matrix $\mathbf{C}$ by optimizing the following objective:

$$\min_{\mathbf{C}} \ \|\mathbf{H} - \mathbf{C}\mathbf{H}\|_F^2 + \beta\|\mathbf{C}\|_F^2, \tag{10}$$

where β is a non-negative parameter. The second term aims to achieve smoothing of the self-expression matrix $\mathbf{C}$ in order to avoid obtaining a trivial solution, *i.e.,* the identity matrix. In fact, we can obtain the self-expression matrix by calculating the closed-form solution of Eq.(10) as follows.

$$\mathbf{C}^* = \mathbf{H}(\mathbf{H})^T(\mathbf{H}(\mathbf{H})^T + \beta\mathbf{I}_N)^{-1}, \tag{11}$$

where $\mathbf{I}_N$ is the identity matrix. In addition, we prove that the self-expression matrix has the grouping effect, thereby capturing the relationship between two nodes from the same class. To do this, we first follow (Li et al. 2020) to define the grouping effect as follows:

Definition 3.2. *Given the nodes set $\mathcal{V} = \{v_i\}_{i=1}^N$, a matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ has the grouping effect if*

$$(v_i \to v_j) \Rightarrow (|\mathbf{w}_{ip} - \mathbf{w}_{jp}| \to 0, \forall 1 \le p \le N).$$

Definition 3.2 states that when $v_i \to v_j$ (*i.e.,* $\|\mathbf{h}_i - \mathbf{h}_j\|_2 \to 0$, node v_i and node v_j may belong to the same class), each element in $\mathbf{w}_i$ and $\mathbf{w}_j$ will be aligned if the matrix $\mathbf{W}$ has the grouping effect. Therefore, the alignment between $\mathbf{w}_i$ and $\mathbf{w}_j$ reflects the relationship of nodes from the same class. Next, we follow (Mo et al. 2024) to prove that the self-expression matrix $\mathbf{C}^*$ has the grouping effects by Theory 3.3.

Theorem 3.3. *The self-expressive matrix $\mathbf{C}^* \in \mathbb{R}^{N \times N}$ has the grouping effect for any two nodes v_i and v_j that hold the condition $v_i \to v_j$,* i.e.,

$$(v_i \to v_j) \Rightarrow (\left|c_{ip}^* - c_{jp}^*\right| \to 0, \forall 1 \le i, j, p \le N). \tag{12}$$

Theorem 3.3 proves that the self-expression matrix $\mathbf{C}^*$ has the grouping effect. If $v_i \to v_j$, then every element in c_i^* and c_j^* will be aligned, *i.e.,* c_i^* and c_j^* are similar. Therefore, all nodes from the same class have the same neighbors, thus capturing the relationship of nodes from the same class.

To capture the consistency among graphs, we first calculate the cross-graph similarity matrix $\mathbf{S}^{(rr')}$ $(r, r' = 1, ..., \mathcal{R})$ between $\mathbf{h}^{(r)}$ and $\mathbf{h}^{(r')}$ by:

$$s_{ij}^{(rr')} = \frac{\mathbf{h}_i^{(r)} \cdot \mathbf{h}_j^{(r')}}{\|\mathbf{h}_i^{(r)}\|\|\mathbf{h}_j^{(r')}\|}. \tag{13}$$

Based on cross-graph similarity matrix $\mathbf{S}^{(rr')}$, previous works usually set the same nodes in different graphs as positive samples, and the others as negative samples, and then utilize the contrastive loss (*e.g.,* JSD (Wang et al. 2022), InfoNCE (Wang et al. 2021)) to improve the similarity of the same nodes in every graph, thus achieving node-level consistency. However, they cannot capture the relationships among nodes from the same class.

Since the self-expression matrix $\mathbf{C}^*$ contains the relationships among nodes from the same class, we set two nodes with positive relationship in $\mathbf{C}^*$ as positive samples and two nodes with negative relationship $\mathbf{C}^*$ as negative samples to design the contrastive loss as follows:

$$\begin{aligned} \mathcal{L}_{con}^{(rr')} = & -\sum_{i=1}^{N} \log\left(\frac{\exp\left(s_{ii}^{(rr')}\right)}{\sum \exp\left(s_{ij}^{(rr')}\right)}\right) \\ & -\sum_{i=1, j\neq i}^{N} c_{ij}^* \log\left(\frac{\exp\left(s_{ij}^{(rr')}\right)}{\sum \exp\left(s_{ij}^{(rr')}\right)}\right). \end{aligned} \tag{14}$$

In Eq. (14), the first term performs contrastive learning on the same node in different graphs by setting the same nodes in different graphs as positive samples and other nodes as negative samples. Hence, the first term makes the distance among positive samples close and pushes away the negative samples. As a result, the representation of the same node in different graphs tends to be consistent, thus achieving node-level consistency. The second term performs contrastive learning on nodes from the same class in different graphs. To do this, it sets the nodes from the same class and different classes in different graphs as positive and negative samples, respectively, through the self-expression matrix $\mathbf{C}^*$, so it makes the distance among positive samples close and pushes away the negative samples. As a result, the representations of nodes from the same class in different graphs tend to be consistent, thus achieving class-level consistency.

3.3 Objective Function

Combining the structure preservation loss (*i.e.,* Eq. (7)) with the contrastive loss (*i.e.,* Eq. (14)), the objective function of the proposed MGHC is:

$$\mathcal{L}_{total} = \sum_{r=1,r'\neq r}^{\mathcal{R}} \mathcal{L}_{con}^{(rr')} + \omega \sum_{r=1}^{\mathcal{R}} \mathcal{L}_{sp}^{(r)}, \tag{15}$$

where ω is a non-negative parameter achieving the trade-off between $\mathcal{L}_{con}^{(rr')}$ and $\mathcal{L}_{sp}^{(r)}$.

In the optimization process, $\mathbf{H}$ captures homophily in the graph structure and reduces the impact of heterophily as well as achieves both node-level consistency and class-level consistency. We then directly average the node representation of each graph to obtain the final node representation $\mathbf{Z}$.

$$\mathbf{Z} = \frac{1}{\mathcal{R}} \sum_{r=1}^{\mathcal{R}} \mathbf{H}^{(r)}. \tag{16}$$

The fused representation $\mathbf{Z}$ can be used for downstream tasks.

4 Experiments

In this section, we conduct extensive experiments on 4 public benchmark datasets to evaluate the effectiveness of our method, compared to 12 comparison methods, on the node classification.

4.1 Experimental Setup

Datasets. The used datasets include two citation multiplex graph datasets (Wang et al. 2019), *i.e.,* ACM and DBLP, and two movie multiplex graph datasets (Mo et al. 2023a), *i.e.,* IMDB and Freebase. We follow (Zhu et al. 2020) to compute the homophily ratio and define the heterophily ratio=1- homophily ratio. As a result, the average heterophily ratio in ACM, DBLP, IMDB, and Freebase is 0.26, 0.38, 0.39 and 0.21, respectively. Obviously, heterophily is widespread in graph datasets

Comparison Methods. The comparison methods include 3 single graph methods and 7 multiplex graph methods.

Single graph methods include 2 traditional semi-supervised methods (*i.e.,* GCN (Kipf and Welling 2016) and GAT (Veličković et al. 2017)) and a self-supervised method (*i.e.,* DGI (Veličković et al. 2018)).

Multiplex graph methods include a semi-supervised method (*i.e.,* HAN (Wang et al. 2019)) and 8 self-supervised methods, *i.e.,* DMGI (Park et al. 2020), DMGIattn (Park et al. 2020), HDMI (Jing, Park, and Tong 2021), HeCo (Wang et al. 2021), CKD (Wang et al. 2022),MGDCR (Mo et al. 2023b),DMG (Mo et al. 2023a) and CoCoMG (Peng, Wang, and Zhu 2023).

To conduct fair comparison, for single graph methods, we separately learn the representation for every graph and further concatenate them for downstream tasks.

4.2 Effectiveness Analysis

We evaluate the effectiveness of the proposed method on node classification tasks 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 node classification tasks.

First, our method greatly outperforms the single graph methods (*i.e.,* GCN, GAT, and DGI). For example, our MGHC outperforms the best single graph method (GAT) by an average of 5.8% on all datasets in terms of macro-F1 and micro-F1. This verifies the superiority of multiplex graph methods which effectively capture complex relationships among multiple graphs to learn discriminative node representation. Second, among all the MGRL methods, the method proposed in this paper has the best performance, followed by DMG, CoCoMG, MGDCR, CKD, HDMI, HeCo, DMGI, DMGIattn and HAN. Compared to the best semi-supervised method (*i.e.,* HAN), our method achieves an average improvement of 4.6% improvement on all the datasets in terms of Macro-F1 and Micro-F1. This indicates that the UMGRL method can efficiently mine the hidden information in multiplex graph without using labels. Compared to the best unsupervised method (*i.e.,* DMG), the proposed method achieves 0.2% improvement, which can be attributed to the fact that the proposed method captures homophily in the graph structure, reduces the impact of heterophily as well as achieves both node-level consistency and class-level consistency. As a result, this helps the model learn discriminative representation, benefiting for downstream tasks.

4.3 Ablation Study

The proposed method captures homophily in the graph structure using structure preservation loss (*i.e.,* $\mathcal{L}_{sp}^{(r)}$) and achieves both node-level consistency and class-level consistency through contrastive loss (*i.e.,* $\mathcal{L}_{con}^{(rr')}$). 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.

First, our method with complete loss improves by 17.4% and 2.8% over methods with only contrastive loss or structure preservation loss, respectively, which demonstrates that both structure preservation loss and contrastive loss are necessary for our method. Second, the method with structure preservation loss achieves higher performance than the method with only contrastive loss. This can be attributed to the fact that restructuring multi-order relationships of graph improve homophily and reduce the impact of heterophily in the graph structure, and therefore, we can capture high-quality structural information, which effectively improves the model performance.

4.4 Visualization of The Self-expression Matrix

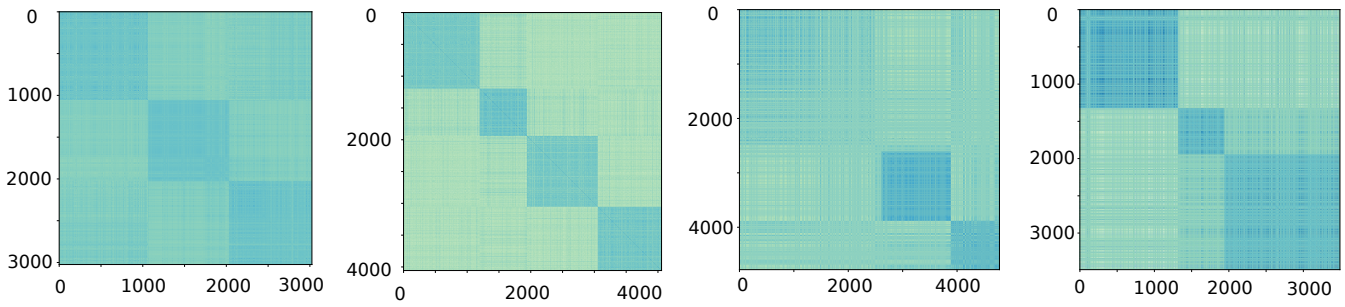
To verify that the self-expression matrix indeed captures capture the relationship between two nodes from the same class, we visualize the heatmaps of self-expressive matrix on the all datasets in Figure 2, where rows and columns are

Method	ACM		DBLP		IMDB		Freebase	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
GCN	85.7 ± 0.4	86.4 ± 0.2	88.6 ± 0.2	89.7 ± 0.1	45.8 ± 0.3	49.1 ± 0.3	50.2 ± 0.4	53.0 ± 0.4
GAT	85.1 ± 0.2	84.4 ± 0.1	90.7 ± 0.3	91.5 ± 0.2	49.7 ± 0.3	53.7 ± 0.3	55.4 ± 0.1	58.9 ± 0.5
DGI	89.3 ± 0.2	88.7 ± 0.3	90.2 ± 0.3	91.1 ± 0.4	45.7 ± 0.3	46.7 ± 0.4	54.2 ± 0.2	57.9 ± 0.3
HAN	89.6 ± 0.3	89.5 ± 0.3	91.7 ± 0.4	92.3 ± 0.3	49.6 ± 0.2	53.1 ± 0.5	54.3 ± 0.3	57.1 ± 0.4
DMGI	89.2 ± 0.2	89.5 ± 0.2	92.0 ± 0.2	92.6 ± 0.3	52.0 ± 0.2	53.6 ± 0.2	54.5 ± 0.2	57.2 ± 0.3
DMGIattn	88.3 ± 0.4	88.6 ± 0.5	90.5 ± 0.3	91.1 ± 0.3	52.9 ± 0.2	53.3 ± 0.5	55.6 ± 0.5	58.7 ± 0.6
HDMI	90.4 ± 0.1	90.1 ± 0.3	91.5 ± 0.4	92.4 ± 0.2	55.3 ± 0.2	57.7 ± 0.3	56.6 ± 0.4	59.1 ± 0.3
HeCo	88.6 ± 0.4	88.8 ± 0.2	91.3 ± 0.2	91.8 ± 0.4	49.6 ± 0.5	50.7 ± 0.4	58.8 ± 0.3	61.3 ± 0.5
CKD	90.3 ± 0.2	90.7 ± 0.3	92.1 ± 0.3	92.3 ± 0.3	54.8 ± 0.2	57.5 ± 0.3	60.4 ± 0.4	62.7 ± 0.6
MGDCR	91.5 ± 0.3	91.5 ± 0.4	92.4 ± 0.4	92.9 ± 0.2	56.4 ± 0.4	57.3 ± 0.3	60.7 ± 0.3	64.7 ± 0.3
DMG	91.0 ± 0.2	90.8 ± 0.4	**93.0 ± 0.2**	**93.8 ± 0.3**	57.5 ± 0.2	58.6 ± 0.3	**62.2 ± 0.8**	**65.6 ± 0.7**
CoCoMG	92.0 ± 0.2	92.2 ± 0.2	91.9 ± 0.1	92.7 ± 0.2	57.6 ± 0.9	58.9 ± 0.3	60.5 ± 0.2	63.5 ± 0.9
MGHC	**92.3 ± 0.3**	**92.4 ± 0.4**	**93.0 ± 0.3**	93.6 ± 0.5	**57.8 ± 0.3**	**59.2 ± 0.2**	61.2 ± 0.5	64.1 ± 0.6

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

$\mathcal{L}_{con}^{(rr')}$	$\mathcal{L}_{sp}^{(r)}$	ACM		DBLP		IMDB		Freebase	
		Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
√	–	78.0 ± 0.4	78.2 ± 0.5	64.7 ± 0.4	65.7 ± 0.4	48.3 ± 0.3	51.1 ± 0.3	43.4 ± 0.3	44.8 ± 0.9
–	√	90.4 ± 0.3	90.4 ± 0.3	89.1 ± 0.6	90.8 ± 0.5	53.3 ± 0.5	55.3 ± 0.4	60.3 ± 1.1	63.3 ± 1.4
√	√	**92.3 ± 0.3**	**92.4 ± 0.4**	**93.0 ± 0.3**	**93.6 ± 0.5**	**57.8 ± 0.3**	**59.2 ± 0.2**	**61.2 ± 0.5**	**64.1 ± 0.6**

Table 2: Classification performance (*i.e.,* Macro-F1 and Micro-F1) of each component in our proposed method on all datasets.



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

Figure 2: The heatmaps of the self-expressive matrix on all datasets.

reordered by node labels. Each pixel in the heatmap represents the correlation between the corresponding nodes. The darker the pixel, the higher the node correlation.

The heatmap shows a similar block diagonal structure. This indicates that in the self-expression matrix, nodes in the same class show a larger correlation with each other, while nodes from different classes show a smaller correlation. This verifies our argument above, *i.e.,* the self-expression matrix effectively captures the relationship between two nodes from the same class. Therefore, the contrast loss based on the self-expression matrix can achieve node-level consistency and class-level consistency.

5 Conclusion

In this paper, we proposed an unsupervised multiplex graph representation learning framework to explore the homophily and consistency in multiplex graph. Specifically, we propose to restructure the multi-order relationships of every graph by correlation coefficients between nodes and multi-order neighbors, thus improving homophily and reducing the impact of heterophily in the graph structure. In addition, we propose to capture the relationship between two nodes from the same class by learning a self-expression matrix and constructing a contrastive loss to achieve node-level and class-level consistency. Extensive experimental results verify the effectiveness of the proposed method with respect to various downstream tasks.

Acknowledgements

This work was supported in part by National Key Research and Development Program of China under Grant 2022YFA1004100 and the Guangxi Natural Science Foundation 2023GXNSFBA026010.

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 *ICML*, 21–29.

Bo, D.; Wang, X.; Shi, C.; and Shen, H. 2021. Beyond low-frequency information in graph convolutional networks. In *AAAI*, 5, 3950–3957.

Chanpuriya, S.; and Musco, C. 2022. Simplified graph convolution with heterophily. *NeurIPS*, 27184–27197.

Chapelle, O.; Weston, J.; and Schölkopf, B. 2002. Cluster kernels for semi-supervised learning. *NeurIPS*, 15.

Chien, E.; Peng, J.; Li, P.; and Milenkovic, O. 2020. Adaptive universal generalized pagerank graph neural network. *arXiv preprint arXiv:2006.07988*.

Chu, X.; Fan, X.; Yao, D.; Zhu, Z.; Huang, J.; and Bi, J. 2019. Cross-network embedding for multi-network alignment. In *WWW*, 273–284.

Dong, Y.; Chawla, N. V.; and Swami, A. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*, 135–144.

He, D.; Liang, C.; Liu, H.; Wen, M.; Jiao, P.; and Feng, Z. 2022. Block modeling-guided graph convolutional neural networks. In *AAAI*, 4, 4022–4029.

Jing, B.; Park, C.; and Tong, H. 2021. Hdmi: High-order deep multiplex infomax. In *WWW*, 2414–2424.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Li, X.; Kao, B.; Shan, C.; Yin, D.; and Ester, M. 2020. CAST: a correlation-based adaptive spectral clustering algorithm on multi-scale data. In *KDD*, 439–449.

Liu, W.; Chen, P.-Y.; Yeung, S.; Suzumura, T.; and Chen, L. 2017. Principled multilayer network embedding. In *ICDMW*, 134–141.

Mo, Y.; Lei, Y.; Shen, J.; Shi, X.; Shen, H. T.; and Zhu, X. 2023a. Disentangled multiplex graph representation learning. In *ICML*, 24983–25005.

Mo, Y.; Nie, F.; Hu, P.; Shen, H. T.; Zhang, Z.; Wang, X.; and Zhu, X. 2024. Self-Supervised Heterogeneous Graph Learning: a Homophily and Heterogeneity View. In *The Twelfth International Conference on Learning Representations*.

Mo, Y.; Wu, Z.; Chen, Y.; Shi, X.; Shen, H. T.; and Zhu, X. 2023b. Multiplex graph representation learning via common and private information mining. In *AAAI*, 8, 9217–9225.

Park, C.; Kim, D.; Han, J.; and Yu, H. 2020. Unsupervised attributed multiplex network embedding. In *AAAI*, 04, 5371–5378.

Peng, L.; Mo, Y.; Xu, J.; Shen, J.; Shi, X.; Li, X.; Shen, H. T.; and Zhu, X. 2023. GRLC: Graph representation learning with constraints. *IEEE transactions on neural networks and learning systems*.

Peng, L.; Wang, N.; Dvornek, N.; Zhu, X.; and Li, X. 2022. Fedni: Federated graph learning with network inpainting for population-based disease prediction. *IEEE Transactions on Medical Imaging*, 42(7): 2032–2043.

Peng, L.; Wang, X.; and Zhu, X. 2023. Unsupervised Multiplex Graph Learning with Complementary and Consistent Information. In *ACM MM*, 454–462.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Veličković, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2018. Deep graph infomax. *arXiv preprint arXiv:1809.10341*.

Wan, G.; Du, B.; Pan, S.; and Haffari, G. 2020. Reinforcement learning based meta-path discovery in large-scale heterogeneous information networks. In *AAAI*, 04, 6094–6101.

Wang, B.; and Tu, Z. 2012. Affinity learning via self-diffusion for image segmentation and clustering. In *CVPR*, 2312–2319.

Wang, C.; Zhou, S.; Yu, K.; Chen, D.; Li, B.; Feng, Y.; and Chen, C. 2022. Collaborative knowledge distillation for heterogeneous information network embedding. In *WWW*, 1631–1639.

Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; and Yu, P. S. 2019. Heterogeneous graph attention network. In *WWW*, 2022–2032.

Wang, X.; Liu, N.; Han, H.; and Shi, C. 2021. Self-supervised heterogeneous graph neural network with co-contrastive learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 1726–1736.

Wang, Y.; and Derr, T. 2021. Tree decomposed graph neural network. In *CIKM*, 2040–2049.

Zhang, H.; Qiu, L.; Yi, L.; and Song, Y. 2018. Scalable multiplex network embedding. In *IJCAI*, volume 18, 3082–3088.

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

Zhou, D.; Bousquet, O.; Lal, T.; Weston, J.; and Schölkopf, B. 2003. Learning with local and global consistency. *NeurIPS*, 16.

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. *NeurIPS*, 7793–7804.

Zhu, Y.; Xu, Y.; Cui, H.; Yang, C.; Liu, Q.; and Wu, S. 2022. Structure-enhanced heterogeneous graph contrastive learning. In *SDM*, 82–90. SIAM.