SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2025, Vol. 68, Iss. 3, 132102:1–132102:14
https://doi.org/10.1007/s11432-023-3939-1

Unsupervised multiplex graph diffusion networks with multi-level canonical correlation analysis for multiplex graph representation learning

Sichao FU[1], Qinmu PENG[1], Yange HE[2], Baokun DU[2], Bin ZOU[3], Xiao-Yuan JING[4] & Xinge YOU[1*]

[1]*School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China;*
[2]*Platform Operation and Marketing Center, JD Retail, Beijing 100176, China;*
[3]*Faculty of Mathematics and Statistics, Hubei Key Laboratory of Applied Mathematics, Hubei University, Wuhan 430062, China;*
[4]*School of Computer Science, Wuhan University, Wuhan 430072, China*

Received 25 April 2023/Revised 25 July 2023/Accepted 21 September 2023/Published online 24 December 2024

Abstract In recent years, unsupervised multiplex graph representation learning (UMGRL) has received increasing research interest, which aims to learn discriminative node features from the multiplex graphs supervised by data without the guidance of labels. Although these designed UMGRL methods have obtained great success in various graph-related tasks, most existing UMGRL models still have the following issues: highly depending on complex self-supervised strategies (i.e., data augmentation, pretext tasks, and negative pairs sampling), restricted receptive fields, and only aggregating low-frequency information between nodes. In this paper, we propose a simple unsupervised multiplex graph diffusion network (UMGDN) with the aid of multi-level canonical correlation analysis to solve the above issues. Specifically, we first decouple the feature transform and propagation processes of the graph convolution layer to further improve the generalization of the learnable parameters. And then, we propose adaptive diffusion propagation to capture long-range dependency relationships between nodes, not the local neighborhood interactions. Finally, a multi-level canonical correlation analysis loss on both the feature transform and propagation processes is proposed to maximize the correlation of the same node features from multiple graphs for guiding model optimization. Compared to the existing UMGRL models, our proposed UMGDN does not need to introduce any data augmentation, negative pairs sampling techniques, complex pretext tasks, and also adaptively aggregates the optimal frequency information between nodes to generate more robust node embeddings. Extensive experiments on four popular datasets and two graph-related tasks demonstrate the effectiveness of the proposed method.

Keywords unsupervised multiplex graph representation learning, graph neural networks, node classification, node clustering

Citation Fu S C, Peng Q M, He Y G, et al. Unsupervised multiplex graph diffusion networks with multi-level canonical correlation analysis for multiplex graph representation learning. Sci China Inf Sci, 2025, 68(3): 132102, https://doi.org/10.1007/s11432-023-3939-1

1 Introduction

Owing to the powerful description ability of graph structure, it is often used to describe and model complex interaction relationships between various objects [1–4]. By carrying out multi-angle and multi-level analyses for the graph structure, the model can better mine discriminative embeddings hidden behind the data, and then make it apply to arbitrary downstream tasks. In the past few years, graph representation learning (GRL) [5–8] that aims to preserve the geometric distribution of non-Euclidean data by dimensionality reduction has obtained prosperous development and also has achieved superior performance in various graph-related tasks, such as node classification [9–11], and node clustering [12,13]. Most existing GRL methods follow the assumption that only exists single interaction relationships among nodes in the graph. However, the graph structure in real life usually exists multiple pairs or even far more complex connection relationships. Thus, how to reasonably explore and exploit the existing diversification connection relationships among nodes for improving the performance of the GRL is important [14].

* Corresponding author (email: youxg@hust.edu.cn)

In recent years, massive multiplex graph representation learning (MGRL) [15] methods are proposed to solve this problem, which aggregates neighbor embeddings from multiple graphs to update its node features by the message-passing scheme [16]. Existing MGRL methods can be roughly divided into two categories: semi-supervised MGRL and unsupervised MGRL.

Semi-supervised MGRL methods aim to utilize a small amount of label information provided in the training dataset to train a robust MGRL model. For example, R-GCN [17] generalized the widely-used GCN [18] to graph-related tasks with large-scale interaction relationships by introducing the basis and block-diagonal-decomposition techniques to relieve its parameters' rapid growth problem. GTN [19] adopted matrix multiplication operation to adaptively learn a reasonable weight coefficient for multiple interaction relationships between nodes, and furtherly generated meta-paths that better describe the local geometric distribution. HGSL [20] dynamically optimized the connection relationships of the raw graph structure to alleviate the influence of inevitably noisy or incomplete connections for MGRL performance. MHGCN [21] introduced the weighted combination operation on the provided multiple graph structures to generate a more accurate adjacency matrix. After then, MHGCN fused the output embeddings of the multi-layer graph convolution module for arbitrary downstream tasks.

The main purpose of unsupervised MGRL (UMGRL) is to train an MGRL model with a better generalization ability by introducing popular self-supervised technologies including cross-view and cross-scale contrastive learning. These proposed UMGRL architectures can discover more supervision signals from the data itself without labels for guiding model optimization. For example, DMGI [22] maximized the average mutual information between the graph-level summary and the local patch representations on each graph for generating relation-type specific embeddings. HeCo [23] learned a high-level embedding matrix by maximizing the agreement of node feature representations from network schema with local structure and meta-path with high-order structure. MCGC [24] conducted contrastive learning at the graph structure level, not the node level to acquire a consensus graph that more accurately preserves the data distribution on high dimensional space. BGRL [25] trained an effective MGRL model by maximizing the cosine similarity loss between the same input node with stochastic graph augmentation techniques including random node feature masking and edge masking.

Although a large number of UMGRL approaches have been designed and also have seen emerging success in numerous graph-related tasks, the superior performance of most existing UMGRL can be influenced by the complex model designs or architectures including complex data augmentation, complex pretext tasks, and complex negative pairs construction. For example, GCA [26] introduced node centrality measures on the graph structure and more noise on the unimportant node attributes to augment the structure and attribute information from multiple graphs, respectively. DMGI [22] made a corruption on each node's attributes and regarded it as negative node embeddings. HeCo [23] regarded the other nodes of all graph structures, not its graph structure as the negative samples. HDMI [27] simultaneously designed two pretext tasks including contrastive between node representations and global summary, contrastive between node representations and node attributes for parameter updating, which all need some complex operations (such as additional projector and mutual information estimator). Moreover, these proposed UMGRL methods still exist the following problems: (1) They only consider the interaction relationships between each node and its direct neighbors in the node features aggregating process while neglecting the long-range dependency relationships, i.e., restricted receptive fields. (2) They only preserve the commonality information of node embeddings by the low-pass filter with fixed parameters while neglecting the fine-grained or other messages of node embeddings.

To solve the above-mentioned problems, in this paper, we propose a multi-level canonical correlation analysis guided unsupervised multiplex graph diffusion network (UMGDN) for multiplex graph representation learning. Specifically, for the given graph-related tasks with multiple graph structures, we first remove the complex data augmentation, complex pretext tasks, and complex negative pairs construction operations for expanding the scope of application of UMGRL in real life. And then, we divide the widely-used graph convolution layer into two modules: the feature transform module including multi-layer perceptron, and the feature propagation module including diffusion propagation mechanism with learnable parameters. In the feature propagation module, our designed diffusion propagation mechanism not only effectively captures long-range dependency relationships between all nodes in the graph, but also adaptively generates an optimal filter to learn more robust node embeddings for the current graph-related task while updating its aggregating other node embeddings process. Inspired by the success of canonical correlation analysis in many areas [28], a multi-level canonical correlation analysis loss on both the feature transform and propagation processes is proposed, which aims to provide more supervision information

for the proposed UMGDN by maximizing the correlation of the same node features from different graphs. Compared with the existing UMGRL, our proposed method is without any complex self-supervised operations and further improves the generalization of the model parameters by the decoupling operation.

The main contributions of the proposed UMGDN can be summarized three fold:

• The widely-used graph convolution layer is divided into feature transform and propagation processes for further improving the generalization of the model parameters. More importantly, a diffusion propagation mechanism with learnable parameters is proposed to learn more robust node embeddings by capturing long-range dependency relationships among nodes and generating an optimal filter.

• We remove some self-supervised operations including complex data augmentation, complex pretext tasks, and complex negative pairs construction for expanding the scope of the application. And then, a simple pretext task is designed to provide more supervision information for UMGDN, i.e., simultaneously maximizing the correlation of the same node features from multiple graphs on both the feature transform and propagation processes.

• Extensive experiments on four popular benchmarks and two graph-related tasks (node classification and node clustering) are conducted to demonstrate the effectiveness of the proposed UMGDN. Experiment results show that our proposed UMGDN without complex designs outperforms the existing UMGRL.

The rest of this paper is organized as follows. Section 2 briefly summarizes the related work on graph diffusion convolution and deep canonical correlation analysis. In Section 3, we describe our proposed UMGDN framework in detail. Extensive experimental analyses on various datasets and graph-related tasks are shown in Section 4. Finally, we summarize this work in Section 5.

2 Related work

In this section, we briefly outline the related work about the proposed UMGDN model, such as graph diffusion convolution and deep canonical correlation analysis. In addition, many existing MGRL studies have been discussed in Section 1.

2.1 Graph diffusion convolution

In the past few years, graph representation learning [5,6], especially the graph convolution networks, has demonstrated a successful technique in various graph-based applications. Existing graph representation learning can be roughly divided into two categories: spectral-based graph representation learning and spatial-based graph representation learning. In spectral-based graph representation learning including GCN [18], Hypergraph GCN [29], Dynamic GCN [30], and SGHFP [31], the graph convolution is defined as removing noise process by designing different graph filters according to graph Fourier transform theory. Spatial-based graph representation learning utilizes the spatial relations between each node and its neighbors to update the feature representations of each node by information propagation, such as GAT [32], GraphSAGE [33], FastGCN [34], and GIN [35]. In this subsection, we summarize the development process of the spectral graph convolution in detail.

The spectral graph convolution is defined as the multiplication of a signal x and a filter g_θ with fixed parameters θ in the graph Fourier transform domain, i.e.,

$$g_\theta(L') * x = U g_\theta(\Lambda) U^{\mathrm{T}} x, \tag{1}$$

where $L' = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \in \mathbb{R}^{N_1 \times N_1}$ denotes the normalized graph Laplacian matrix. I and D are the identity matrix and degree matrix $D_{ii} = \sum_{j=1}^{j=N_1}(A_{ij})$ about adjacency matrix $A \in \mathbb{R}^{N_1 \times N_1}$, respectively. $U = [u_0, u_1, \ldots, u_{N_1}] \in \mathbb{R}^{N_1 \times N_1}$ and $\Lambda = [\lambda_1, \lambda_2, \ldots, \lambda_{N_1}] \in \mathbb{R}^{N_1 \times N_1}$ represent the generated eigenvectors matrix and eigenvalues matrix of normalized graph Laplacian matrix L' according to spectral decomposition. * denotes the convolution operation in the spectral domain.

Owing to the spectral decomposition with expensive computational complexity, the above method cannot well address the large-scale graph structure. To solve this issue, Defferrard et al. [36] introduced the K-order Chebyshev polynomials $T_k()$ with a truncated expansion to well approximate $g_\theta(\Lambda)$. Thus,

the above spectral graph convolution can be written as follows:

$$\begin{aligned} g_\theta(L') * x &= U\left(\sum_{k=0}^{K-1}\theta_k T_k(\tilde{\Lambda})\right)U^{\mathrm{T}}x \\ &= \sum_{k=0}^{K-1}\theta_k T_k(\tilde{L})x, \end{aligned} \tag{2}$$

where $\tilde{L} = \frac{1}{2\lambda_{\max}}L' - I$ and $\lambda_{\max}$ is the largest eigenvalue of the graph Laplacian matrix L'. K-order Chebyshev polynomials $T_k(x)$ are recursively defined as the following form: $T_0(x) = 1$, $T_1(x) = x$, and $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$. To further improve the computing efficiency of the ChebNet [36], GCN [18] utilized a localized first-order approximation of spectral graph convolutions to optimize the above definition, i.e., $\lambda_{\max} = 2$, $K = 1$, and $\theta_0 = -\theta_1 = \theta$. The final spectral graph convolution is represented as follows:

$$g_\theta(L') * x = \tilde{D}^{-\frac{1}{2}}(A + I)\tilde{D}^{-\frac{1}{2}}x\theta, \tag{3}$$

where $\tilde{D}$ is the degree matrix about the structure relationships $A + I$ with self-connection. Although more and more GCN variants have been designed and achieved superior performance in different graph-related tasks, most existing GCN variants highly depend on the local neighborhood interactions and seriously limit the robustness of the learned node embeddings. To solve this problem, Gasteiger et al. [37] regarded the above spectral graph convolution as the information diffusion process between different nodes. They assumed that information distribution between different nodes can reach equilibrium after multi-iterations. Based on this assumption, the graph diffusion convolution is defined as

$$g_\theta(L') * x = \sum_{k=0}^{\infty}\theta_k(D^{-\frac{1}{2}}AD^{-\frac{1}{2}})^k x. \tag{4}$$

By setting different filter parameters θ, we can obtain different graph filters, such as low-pass filters and high-pass filters. In GDC [37] and MAGN [38], they enhanced the low-frequency information of graph structured data by $\theta_D = \beta(1-\beta)^k$, $\sum_{k=0}^{\infty}\theta_k = 1$, and $\beta \in (0, 1)$. k is the diffusion distance. Recently, graph diffusion convolution has received increasing attention in graph representation learning. However, how to generalize the existing graph diffusion convolution to UMGRL for adaptively selecting the optimal frequency information is undiscovered.

2.2 Deep canonical correlation analysis

Canonical correlation analysis (CCA) [39,40] is a classical subspace learning and multivariate data analysis method, which aims to seek its optimal mapping direction for different data variables to make different variables after mapping generate a maximized correlation coefficient in the same subspace. Specifically, for the given two random variables $M = \{m_1, m_2, \ldots, m_{N_2}\} \in \mathbb{R}^{d_M \times N_2}$ and $Q = \{q_1, q_2, \ldots, q_{N_2}\} \in \mathbb{R}^{d_Q \times N_2}$, CCA maximizes the correlation $\rho(M^{\mathrm{T}}W_M, Q^{\mathrm{T}}W_Q)$ between $W_M^{\mathrm{T}}S_{MM}W_M$ and $W_Q^{\mathrm{T}}S_{QQ}W_Q$ via the learned mapping directions $W_M \in \mathbb{R}^{d_M \times 1}$ and $W_Q \in \mathbb{R}^{d_Q \times 1}$, i.e.,

$$\max_{W_M, W_Q}\rho(M^{\mathrm{T}}W_M, Q^{\mathrm{T}}W_Q) = \frac{W_M^{\mathrm{T}}S_{MQ}W_Q}{\sqrt{W_M^{\mathrm{T}}S_{MM}W_M}\sqrt{W_Q^{\mathrm{T}}S_{QQ}W_Q}}, \tag{5}$$

where $S_{MM} = \frac{1}{N_2-1}MM^{\mathrm{T}}$, $S_{MQ} = \frac{1}{N_2-1}MQ^{\mathrm{T}}$, and $S_{QQ} = \frac{1}{N_2-1}QQ^{\mathrm{T}}$. M and Q denote the normalized node embeddings. d_M, d_Q, N_2, and T represent the feature dimensions of M, feature dimensions of Q, sample numbers, and matrix transposition, respectively. S_{MM}, S_{MQ}, and S_{QQ} denote the covariance matrix between two variables. Formally, the final objective function can be simplified as follows:

$$\max_{W_M, W_Q} W_M^{\mathrm{T}}S_{MQ}W_Q \quad \text{s.t. } W_M^{\mathrm{T}}S_{MM}W_M = W_Q^{\mathrm{T}}S_{QQ}W_Q = 1. \tag{6}$$

Owing to the superiority of the above CCA theory, a large number of researchers have proposed many classical CCA variants for multi-view dimensionality reduction [28,41]. For example, sparse CCA [42] is proposed to solve the uncertainty of model operation, which aims to make some data variables converge

to zero by introducing additional spare regularization terms. To deal with multi-view learning with a large number of non-linear relationships, kernel CCA [43] utilized the non-linear kernel function to map the original data variables to the high-dimensional feature space. Laplacian multiset CCA [44] utilized local geometry distribution among within-view and between-view to better discover the nonlinear correlations among different data variables. In recent years, deep learning has become a successful technique in many areas, such as computer vision and natural language processing. Owing to their advantage, many deep CCA networks are designed to deal with multi-view tasks. For example, Soft CCA [45] utilized the deep neural networks to conduct dimensionality reduction and replaced the exact decorrelation with soft decorrelation constraints for reducing the expensive model computational complexity. Their unconstrained objective can be reformulated as follows:

$$\min_{\varkappa_1,\varkappa_2} \frac{1}{2}\|J_{\varkappa_1}(M) - J_{\varkappa_2}(Q)\|_2^2 + \alpha(\|J_{\varkappa_1}(M) - I\|_1 + \|J_{\varkappa_2}(Q) - I\|_1), \tag{7}$$

where $\varkappa_1$ and $\varkappa_2$ denote the weight parameters of the deep neural networks $J_{\varkappa_1}()$ and $J_{\varkappa_2}()$. Apart from Soft CCA [45], CCANet [46] first utilized the widely-used CCA to learn the optimal two-view multistage filter banks. After then, the binary hashing and block-wise histogram are introduced to generate the fused sample features for multi-view learning. Different from CCANet, TCCANet [47] maximized the canonical correlation with the arbitrary number of views for considering the high-order-correlations that existed in multi-view data. Although more and more CCA variants have effectively boosted the task performance of multi-view learning, how to apply the existing CCA to UMGRL and discover more supervision information from the data itself for guiding model optimization is still a challenging problem.

3 Method

In this section, we describe how to utilize the designed multi-level canonical correlation analysis loss to discover effective supervision information from the data itself for training a robust UMGDN in detail. As shown in Figure 1, our proposed UMGDN has a total of 4 steps. Specifically, (1) send arbitrary graph-structured data with multiple interaction relationships G to the proposed UMGDN for generating the effective node embeddings of each graph. (2) The proposed adaptive graph diffusion convolution is divided into feature transform and propagation modules. During the feature propagation, the diffusion propagation mechanism with adaptive filter parameters is introduced to capture long-range dependency relationships between nodes and adaptively select the optimal graph filter for each graph. (3) A multi-level canonical correlation analysis is simultaneously introduced to the feature transform and propagation processes for supervising their training process. After multiple epochs optimization, we can learn its robust node features for each graph. (4) During the downstream tasks, the widely-used attention mechanism is introduced to fuse the node embeddings of each graph for obtaining its final embeddings.

3.1 Unsupervised multiplex graph diffusion networks

Before introducing our proposed UMGDN model, we first describe the related definitions and symbols. In this paper, $G = \{G_1, G_2, \ldots, G_R\} = \{V, \xi, X\}$ denotes the graph-structured data with multiple interaction relationships. $V = \{v_1, v_2, \ldots, v_N\} \in \mathbb{R}^N$, $\xi = \{\xi_1, \xi_2, \ldots, \xi_R\}$, $X = \{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^{N\times F_1}$, and $\mathbb{A} = \{A_1, A_2, \ldots, A_R\}$ represent the set of nodes, node edges, node features, and adjacency relationships, respectively, where v_N, ξ_R, x_N, and $A_R \in \mathbb{R}^{N\times N}$ is the N-th node, the set of all edges in G_R, N-th node's embeddings, and the adjacency relationships matrix of G_R. $G_R = \{V, \xi_R, X\}$ denotes the graph-structured data with R-th adjacency relationships A_R. R, N, and F_1 are the numbers of the interaction relationships (graph), nodes, and node embedding dimensions. The main goal of the proposed UMGDN is to learn the robust low-dimensional node embeddings for each graph G_R. With the more representative node embeddings of each graph G_R, they are easy to combine with any models for improving the performance of downstream tasks.

In the classical GCN [18] model, the graph convolution layer for multip graph representation learning G is defined as follows:

$$H_R^{(L+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}(A_R + I)\tilde{D}^{-\frac{1}{2}}H_R^{(L)}W_R^{(L)}), \tag{8}$$

where $H_R^{(L)}$ and $H_R^{(L+1)}$ denote the input and output embeddings of the L-layer GCN in G_R. $W_R^{(L)}$ is the learnable parameters matrix of the L-layer GCN in G_R. σ denotes the non-linear activation

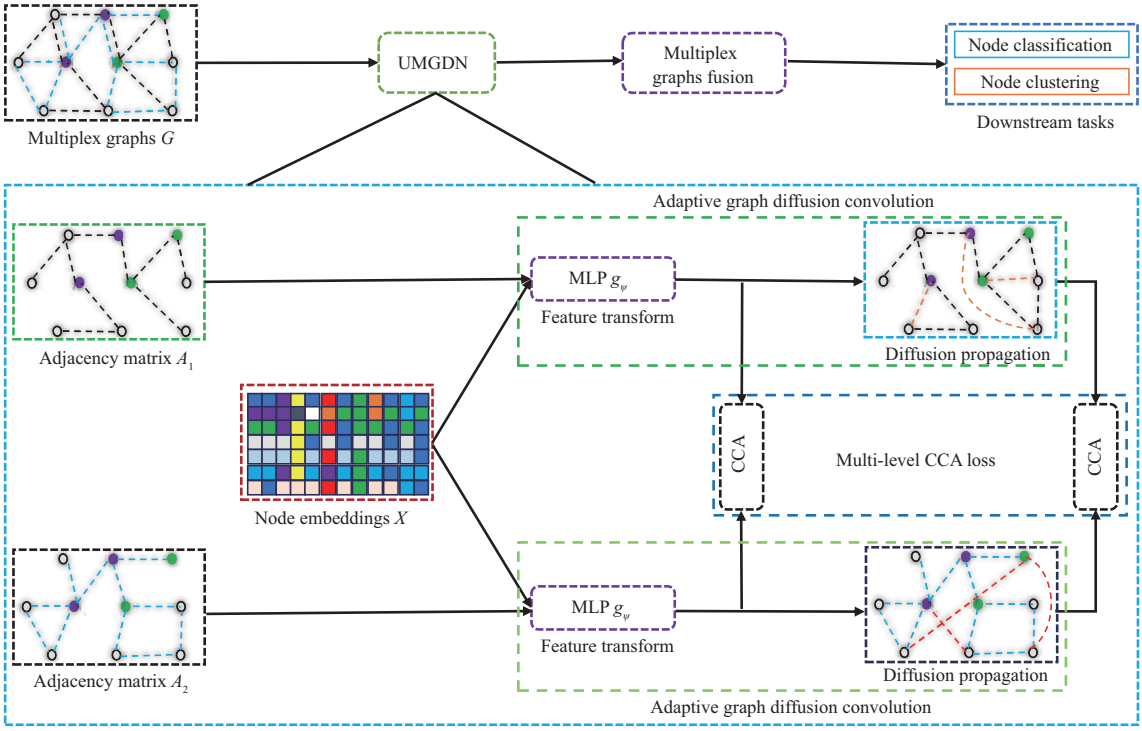

Figure 1 (Color online) Basic framework of the proposed unsupervised multiplex graph diffusion networks (UMGDN) model.

function. Although more and more GCN variants based on (8) have been proposed, it is difficult to guarantee the global optimization of all learnable parameters with the layer numbers increase, which has been demonstrated in [48]. Specifically, the original node feature X is mapped to a low-dimensional space by $H_R^{(L)}W_R^{(L)}$, and then feature propagation $\tilde{D}^{-\frac{1}{2}}(A_R+I)\tilde{D}^{-\frac{1}{2}}H_R^{(L)}$ is introduced to propagate information between nodes. As the number of graph convolution layers increases, the entanglement of a large number of learnable parameters and feature aggregation enlarges the training difficulty of GCN and further degrades the discriminant of $H_R^{(L+1)}$. In fact, feature transformation $H_R^{(L)}W_R^{(L)}$ and propagation $\tilde{D}^{-\frac{1}{2}}(A_R+I)\tilde{D}^{-\frac{1}{2}}H_R^{(L)}$ processes in (8) depend on the learnable weight matrix and provided graph structure, respectively. Thus, the above graph convolution layer does not need to intertwine the feature transformation with propagation. To solve this problem, we decouple the feature transform and propagation training processes of the graph convolution layer to further improve the generalization of the learnable parameters, i.e.,

$$Z_R = g_\psi(X), \tag{9}$$

$$H_R^{(L+1)} = \tilde{D}^{-\frac{1}{2}}(A_R+I)\tilde{D}^{-\frac{1}{2}}\cdots\sigma(\tilde{D}^{-\frac{1}{2}}(A_R+I)\tilde{D}^{-\frac{1}{2}}Z_R), \tag{10}$$

where $g_\psi()$ and ψ denote the multi-layer perceptron and its corresponding weight parameters. Z_R is the node embeddings obtained after feature transformation in G_R. Although this operation can improve the generalization of the learnable parameters, it only considers the influence of the direct neighbors for its embeddings in the feature aggregating process, i.e., restricted receptive fields. For (10), how to capture the long-range dependency relationships between nodes precisely for generating robust node embeddings is important. Inspired by the diffusion mechanism [37, 38], we make a generation for (1) to increase the receptive fields of each node and give the definition of the adaptive graph diffusion convolution, i.e.,

$$g_\theta(L')*x = g_\theta(L')x = \sum_{k=0}^{K-1}\theta_k(L')^k x = \sum_{k=0}^{K-1}\theta_k(I-D^{-\frac{1}{2}}AD^{-\frac{1}{2}})^k x$$
$$= \sum_{k=0}^{K-1}\theta_k\sum_{k_1=0}^{K_1-1}\binom{K_1}{k_1}I^{K_1-k_1}(-D^{-\frac{1}{2}}AD^{-\frac{1}{2}})^{k_1}x$$

$$\begin{aligned} &= \sum_{k_1=0}^{K-1}\sum_{k=k_1}^{K-1}\theta_k\binom{K_1}{k_1}(-1)^{k_1}(D^{-\frac{1}{2}}AD^{-\frac{1}{2}})^{k_1}x \\ &= \sum_{k_1=0}^{K-1}\mu_{k_1}(D^{-\frac{1}{2}}AD^{-\frac{1}{2}})^{k_1}x, \end{aligned} \tag{11}$$

where $\mu_{k_1} = \sum_{k=k_1}^{K-1}\theta_k\binom{K_1}{k_1}(-1)^{k_1}$ and K denote the learnable graph filter parameters and diffusion distance, respectively. Compared with the existing graph diffusion convolution, our proposed method can adaptively learn the optimal graph filter by μ_{k_1}, not the simple low-pass or high-pass graph filter. More importantly, this operation can enlarge the receptive fields of each node in a single graph convolution layer, which can better build long-range dependency relationships between nodes and directly improve the discriminant of node embeddings $H_R^{(L+1)}$. Thus, the above operation powerfully guarantees that our proposed UMGDN is theoretically more expressive than fixed filter parameters-based graph representation learning. Based on (11), the above feature propagation process can be optimized as follows:

$$Z_R'' = H_R^{(L+1)} = \sum_{k_1=0}^{K-1}\mu_{k_1}(D^{-\frac{1}{2}}A_RD^{-\frac{1}{2}})^{k_1}\cdots\sigma\left(\sum_{k_1=0}^{K-1}\mu_{k_1}(D^{-\frac{1}{2}}A_RD^{-\frac{1}{2}})^{k_1}Z_R\right), \tag{12}$$

where Z_R'' denotes the final node embeddings in G_R.

3.2 Model training

Different from the existing UMGRL methods, our proposed method does not need to introduce complex self-supervised strategies (complex data augmentation, complex pretext tasks, complex negative pairs construction), which further improves the efficiency of the model. Inspired by the great success of CCA in various areas, we design a simple pretext task to train a robust UMGDN; i.e., we maximize the correlation of the same node features from different graphs on both the transform and propagation processes for model training. The designed multi-level canonical correlation analysis loss can be written as follows:

$$\begin{aligned} \text{Loss} &= \min_{\psi,\mu_{k_1}}(\text{Loss}_{\text{FP}} + \text{Loss}_{\text{FT}}) \\ &= \min_{\psi,\mu_{k_1}}\sum_{i=1,j=i+1}^{R-1}(\|Z_i'' - Z_j''\|_2^2 + \gamma_i''(\|(Z_i'')^{\text{T}}Z_i'' - I\|_2^2 + \|(Z_j'')^{\text{T}}Z_j'' - I\|_2^2)) \\ &\quad + \min_{\psi,\mu_{k_1}}\sum_{i=1,j=i+1}^{R-1}(\|Z_i - Z_j\|_2^2 + \gamma_i(\|Z_i^{\text{T}}Z_i - I\|_2^2 + \|Z_j^{\text{T}}Z_j - I\|_2^2)), \end{aligned} \tag{13}$$

where γ_i'' and γ_i denote the parameters trading off different terms. Loss_{FP} and Loss_{FT} are the CCA loss on the features transform and propagation process, respectively. For Loss_{FP} and Loss_{FT}, the first term encourages maximizing the mutual information among the same node features from different graphs. The second term aims to minimize the mutual information among the different dimensionality from the sample graph by pushing the off-diagonal elements of the covariance matrix close to 0. Under the guidance of multi-level CCA loss, each node can learn its robust node embeddings including the common semantics between the same category and discriminate semantics between the different categories by UMGDN. More importantly, our proposed multi-level CCA loss can discover more supervision information from the data itself and further enhance the global optimization of all learnable parameters in UMGDN in comparison to simple depending on the CCA loss on the feature transform or propagation process.

3.3 Multiplex graphs fusion for downstream tasks testing

After multiple epochs optimization, we can learn its robust node features for each graph. However, how to fuse the node embeddings of each graph to acquire its final embeddings is also important. To make a fair comparison with the existing UMGRL and also consider the feature diversities from different graphs, we introduce the widely-used attention mechanism to fuse the node embeddings of multiple graphs.

$$a_R = \tanh((f_\varphi(Z_R''))^{\text{T}}W_{a_R}Z_R''), \tag{14}$$

Algorithm 1 Unsupervised multiplex graph diffusion networks with multi-level canonical correlation analysis.

Input: Node features X, adjacency matrix from multiple graphs $\mathbb{A} = \{A_1, A_2, \ldots, A_R\}$;
Parameters: Parameter dimensions ψ_1 and ψ_2, diffusion distance K, balance parameters γ_i'' and γ_i;
Output: The performance of a given downstream task;
Initialize parameters ψ and μ_{k_1};
while not done **do**
  Compute node embeddings Z_R on the feature transformation process via (9);
  Compute node embeddings Z_R'' on the feature propagation process via (12);
  Obtain the multi-level CCA loss L on both the feature transformation and propagation via (13);
  Update parameters ψ and μ_{k_1} via gradient descent;
end while
Obtain the fused node embeddings Z_{final} for a given downstream task via (14)–(16);
Return The performance of a given downstream task.

where a_R and W_{a_R} denote the attention coefficient and learnable weight parameters of the R-th graph G_R. $f_\varphi(Z_R'')$ denotes the node embeddings after multi-layer perceptron $f_\varphi()$. And then, we further introduce the normalization operation to acquire the importance of each graph for the final node embeddings, i.e.,

$$a_R' = \frac{\exp(a_R)}{\sum_{r=1}^{R} \exp(a_r)}, \tag{15}$$

where a_R' represents the normalized importance coefficient of each graph. Based on the normalized a_R', we can obtain the final embeddings Z_{final}:

$$Z_{\text{final}} = \sum_{r=1}^{R} a_R' Z_R'', \tag{16}$$

which can be applied to any downstream task testing. The overall algorithm flowchart of the proposed UMGDN method is shown in Algorithm 1.

4 Experiment

In this paper, we evaluate the efficiency of the proposed UMGDN on two citation multiple graph datasets (ACM [49] and DBLP [49]) and two movie multiple graph datasets (IMDB [49] and Freebase [23]). The detailed data descriptions are summarized as follows.

4.1 Datasets

ACM [49] is composed of 3025 papers with the bag-of-words description. These nodes are divided into 3 classes, such as database, data mining, and wireless communication. The number of node feature dimensions is 1830. It describes the multiple interaction relationships between nodes by PAP and paper-subject-paper (PSP). In the downstream node classification tasks, 600 nodes are selected for training. DBLP [49] is a citation multiple graph dataset with 3 node interaction relationships including paper-author-paper (PAP), paper-paper-paper (PPP), and paper-author-term-author-paper (PATAP). It is composed of 7907 papers with the 2000-dimensional bag-of-words description, which is sampled from 4 categories including natural language processing, artificial intelligence, data mining, and computer vision. For citation multiple graph datasets, node attributes are sampled from the bag-of-words description of the paper abstract. In DBLP [49], 80 papers are chosen for training downstream node classification tasks.

IMDB [49] contains 3550 movies collected from 3 categories including drama, action, and comedy. The feature dimensions of all movies are set to 1007, which is described by the bag-of-words of the movie plot. It exits 2 interaction relationships between movies, i.e., movie-actor-movie (MAM) and movie-director-movie (MDM). 80 movies are used for training downstream node classification tasks. Freebase [23] is made up of 3492 movies with 3 interaction relationships between movies, i.e., MAM, movie-writer-movie (MWM), and MDM. All movies of Freebase are described by one-hot encoding with a size of 3492. IMDB [49] and Freebase [23] are sampled from the same categories. It chooses 60 movies to train downstream node classification tasks.

Table 1 Performance comparison on node classification and node clustering. The best and second-best results are in bold and underlined, respectively. OOM denotes out-of-memory.

	Classification								Clustering		
	ACM		IMDB		DBLP		Freebase		ACM	IMDB	DBLP
	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	NMI		
Deep Walk [51]	73.9	74.8	53.2	55.0	53.3	53.7	49.3	52.1	31.0	11.7	34.8
node2vec [52]	74.1	74.9	55.3	55.0	54.3	54.7	66.2	66.9	30.9	12.3	38.2
GCN [18]	86.9	87.0	60.3	61.1	73.4	71.7	50.5	53.3	67.1	17.6	46.5
GAT [32]	85.0	84.9	49.7	51.3	76.9	75.4	55.1	59.7	67.1	17.6	46.5
MNE [59]	79.2	79.7	55.2	57.4	56.6	56.2	52.1	54.3	54.5	1.3	13.6
ANRL [53]	81.9	82.0	57.3	57.6	77.0	69.9	20.6	44.7	51.5	16.3	33.2
DGCN [54]	88.8	88.8	58.2	59.2	70.7	69.8	46.0	46.0	69.1	14.3	46.2
CAN [55]	59.0	63.6	57.7	58.8	70.2	69.4	44.3	53.1	50.4	7.4	32.3
CMNA [60]	78.2	78.8	54.9	56.6	56.6	56.1	–	–	49.8	15.2	42.0
mGCN [61]	85.8	86.0	62.3	63.0	72.5	71.3	54.6	57.5	66.8	18.3	46.8
DGI [56]	88.1	88.1	59.8	60.6	72.3	72.0	54.9	58.2	64.0	18.2	55.1
HAN [49]	87.8	87.9	59.9	60.7	71.6	70.8	53.2	57.2	65.8	16.4	47.2
DMGI [22]	89.8	89.8	64.8	64.8	77.1	76.6	54.9	57.6	68.7	19.6	40.9
$\text{DMGI}_{\text{attn}}$ [22]	88.7	88.7	60.2	60.6	77.8	77.0	55.8	58.3	70.2	18.5	55.4
GraphCL [57]	88.4	88.3	61.9	62.3	81.4	80.6	35.9	37.2	67.3	14.9	54.5
HDI [27]	90.1	90.0	63.4	63.8	81.4	80.0	49.6	55.3	65.0	19.4	57.0
HDMI [27]	90.1	90.1	65.0	**65.8**	82.0	81.1	56.1	59.2	69.5	19.8	58.2
HeCo [23]	88.2	88.3	64.7	65.2	81.4	80.6	59.2	61.7	67.8	19.1	56.1
MCGC [24]	90.2	90.0	64.3	64.9	81.6	80.5	56.6	59.4	69.0	21.1	51.1
MvAGC [62]	77.8	79.1	59.8	61.5	50.9	54.2	–	–	66.5	**21.9**	28.1
GCA [26]	79.8	79.7	52.3	53.3	OOM	OOM	–	–	44.3	0.7	OOM
BGRL [25]	–	–	63.1	63.4	81.9	80.7	–	–	–	–	–
RGRL [58]	90.3	90.2	65.3	**65.8**	<u>83.0</u>	81.8	<u>59.4</u>	62.1	69.4	20.9	56.8
MHNF [63]	<u>92.1</u>	<u>91.9</u>	<u>65.5</u>	<u>65.7</u>	82.9	<u>82.1</u>	**59.5**	<u>62.8</u>	<u>70.9</u>	20.4	<u>59.1</u>
UMGDN	**92.3**	**92.2**	**66**	65.6	**84.5**	**83.4**	58.1	**66.1**	**73.7**	<u>21.7</u>	**59.6**

4.2 Experimental setting

In this paper, we conduct extensive experiments on supervised node classification and unsupervised node clustering tasks to evaluate the effectiveness of the proposed UMGDN model. Specifically, we first utilize the trained UMGDN and multiplex graphs fusion operation to generate the final node embeddings for the given downstream task. For node classification, the provided label information from the downstream task is used to train a logistic regression based on the acquired node embeddings. For node clustering, we train the K-means model with the acquired node embeddings. All experiments are conducted five times and show their average results. The Macro-F1 (Ma-F1), Micro-F1 (Mi-F1), and normalized mutual information (NMI) are introduced to evaluate the task performance of all comparison models and our model on supervised node classification and unsupervised node clustering tasks, respectively. For our UMGDN, we introduce the grid search technique to search its optimal parameters. The numbers of hidden units ψ_1 and output feature dimensions ψ_2 are searched in $\{4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048\}$. Diffusion distance K is tuned from 1 to 10. Dropout is selected from 0.1 to 0.9. Balance parameters γ_i'' and γ_i are tuned from 0.09 to 0.0009. The layers of the adaptive graph diffusion convolution are set to 2. All model parameters are initialized by Glorot [50] and the Adam optimizer [50] with a learning rate of 0.0005 is introduced to update model parameters.

4.3 Node classification and clustering

In this subsection, we compare the proposed UMGDN with many existing GRL approaches including homogeneous GRL [18, 25–27, 32, 51–58] and MGRL [22–24, 27, 49, 59–63]. Table 1 shows the average results of all comparison models and our model on supervised node classification and unsupervised node clustering tasks. From Table 1, we can see the following observations.

• Our proposed UMGDN obtains a novel state-of-the-art performance in most cases. For example, compared with the classical DGI and DMGI, our UMGDN obtains a 4.2%, 4.1%, 6.2%, 5%, 12.2%,

Table 2 Ablation study on node classification and node clustering. The best results are in bold.

	ACM			IMDB			DBLP		
	Ma-F1	Mi-F1	NMI	Ma-F1	Mi-F1	NMI	Ma-F1	Mi-F1	NMI
UMGDN (w/o ADC, Loss_{FT})	72.8	72.3	59.1	58.6	59.8	2.4	74.4	72.7	31.1
UMGDN (w/o Loss_{FT})	81.3	81.7	70.3	59.8	60.6	15.3	79.2	77.7	41.2
UMGDN	**92.3**	**92.2**	**73.7**	**66.0**	**65.6**	**21.7**	**84.5**	**83.4**	**59.6**

11.4%, 3.2%, 7.9%, 9.7%, 3.5%, 4.5%; 2.5%, 2.4%, 1.2%, 0.8%, 7.4%, 6.8%, 3.2%, 8.5%, 5%, 2.1%, 18.7% improvement, in turns. The proposed UMGDN obtains improvement of 0.2%, 0.3%, 0.5%, 1.5%, 1.3%, 3.3%, 2.8%, 0.5% in comparison to the state-of-the-art model of node classification (MNHF, RGRL) and node clustering (MNHF). Owing to the limited node numbers, our UMGDN cannot acquire training enough and shows a bad task performance on individual tasks in comparison to some self-supervised technologies-based UMGRL (HeCo, MvAGC, HDMI).

• Compared with the homogeneous GRL, MGRL obtains the best performance in most cases. These results show that reasonably utilizing the multiple interaction relationships between nodes to improving the robustness of low-dimensional node embeddings is important. Secondly, UMGRL based on self-supervised technologies (such as DMGI, HDMI) obtains a large improvement in comparison to supervised/semi-supervised UMGRL (such as HAN, MNE). It indicates that the acquired supervised information from data itself by constructing complex pretext tasks can further enhance the generalization of the trained MGRL on the downstream tasks. Finally, our proposed UMGDN without complex self-supervised strategies obtains competitive results. On the one hand, these results indicate the effectiveness of the proposed adaptive graph diffusion convolution and multi-level canonical correlation analysis loss; on the other hand, they also show that the complex data augmentation, complex pretext tasks, and complex negative pairs construction are not necessary for self-supervised technologies-based UMGRL.

4.4 Ablation experiments

In Table 2, we conduct extensive experiments to analyze the influence of the proposed adaptive graph diffusion convolution (ADC) (Eq. (12)) and multi-level canonical correlation analysis loss modules for downstream tasks. UMGDN (w/o ADC, Loss_{FT}) denotes that without utilizing the ADC and the CCA loss Loss_{FT} on the features transform process during model training, which adopts the widely-used feature propagation module (Eq. (10)) and only uses Loss_{FP} to supervise model's optimization. UMGDN (w/o Loss_{FT}) represents without utilizing the CCA loss Loss_{FT} on the features transform process for UMGDN training. As shown in Table 2, we can see that our proposed modules all improve the performance of the downstream tasks. For example, UMGDN (w/o Loss_{FT}) obtains gains of 8.5%, 9.4%, 11.2%, 1.2%, 0.8%, 12.9%, 4.8%, 5%, 10.1% in comparison to UMGDN (w/o ADC, Loss_{FT}). UMGDN also outperform UMGDN (w/o Loss_{FT}) by 11%, 10.5%, 3.4%, 6.2%, 5%, 6.4%, 5.3%, 5.7%, 18.4%. Secondly, the CCA loss Loss_{FT} on the feature propagation process plays a significant role in improving the generalization of the proposed UMGDN. It also reveals that the proposed multi-level canonical correlation analysis loss Loss can discover more supervision information from data itself in comparison to simple depending on the Loss_{FP}. These results also reveal that the proposed ADC (Eq. (12)) can effectively capture long-range dependency relationships between nodes and further generate robust node embeddings by learning an optimal graph filter.

4.5 Parameters sensitivity

To evaluate the effectiveness of the proposed UMGDN on each graph, we show the average results of three classical self-supervised technologies-based UMGRL and our method in Table 3. As shown in Table 3, we can know that the proposed UMGDN obtains a large improvement in all downstream tasks and all datasets. For example, compared with the second-best methods (HDI (PSP), HDI (PAP), and GraphCL (PAP)) on the ACM dataset, our method achieves gains of 13.6%, 13.4%, 2.2%, 2.3% on the node classification, 16.2%, 1.7% on the node clustering. These results demonstrate that our proposed UMGDN can effectively improve the robustness of the extracted node embeddings aggregating single graph structure in comparison to the existing UMGRL. It also indirectly evaluates the robustness of the final node embeddings applied to the downstream task. Apart from this experiment, we also investigate

Table 3 Performance comparison of each graph on node classification and node clustering. The best and second-best results are in bold and underlined, respectively. OOM denotes out-of-memory.

	ACM						DBLP								
	PSP			PAP			PAP			PPP			PATAP		
	Ma-F1	Mi-F1	NMI	Ma-F1	Mi-F1	NMI	Ma-F1	Mi-F1	NMI	Ma-F1	Mi-F1	NMI	Ma-F1	Mi-F1	NMI
DGI [56]	66.3	66.8	52.6	85.5	85.3	65.1	80.4	79.6	54.7	72.8	71.7	40.4	24.0	27.2	5.4
GraphCL [57]	64.9	65.8	52.4	83.3	82.4	67.5	80.6	77.9	53.9	67.8	67.5	34.7	23.6	28.6	5.2
HDI [27]	74.2	74.4	52.8	88.9	88.8	66.2	81.2	80.3	56.2	75.1	74.5	40.8	24.1	28.4	5.4
GCA [26]	64.5	65.6	38.9	74.8	74.9	6.2	71.6	71.0	7.6	67.9	66.5	22.3	OOM	OOM	OOM
UMGDN	**87.8**	**87.8**	**69.0**	**91.1**	**91.1**	**69.2**	**82.8**	**81.7**	**58.3**	**79.6**	**79.0**	**46**	**73.2**	**73.1**	**36.2**

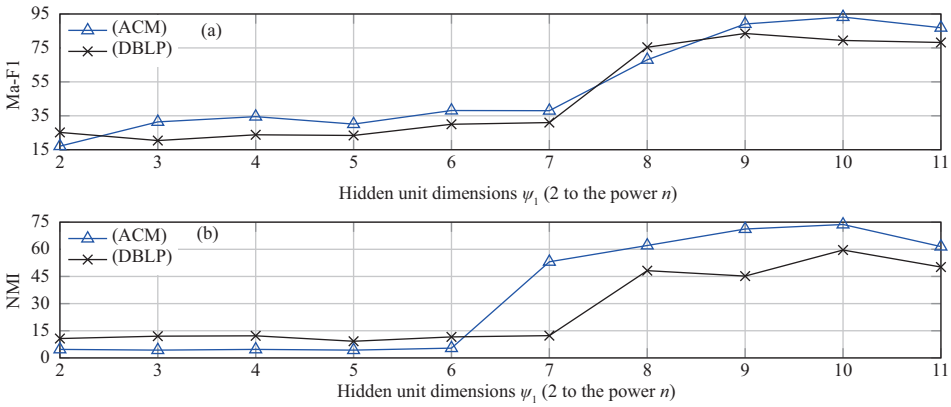

Figure 2 (Color online) Sensitivity analysis of the hidden unit dimensions ψ_1 on (a) node classification and (b) node clustering.

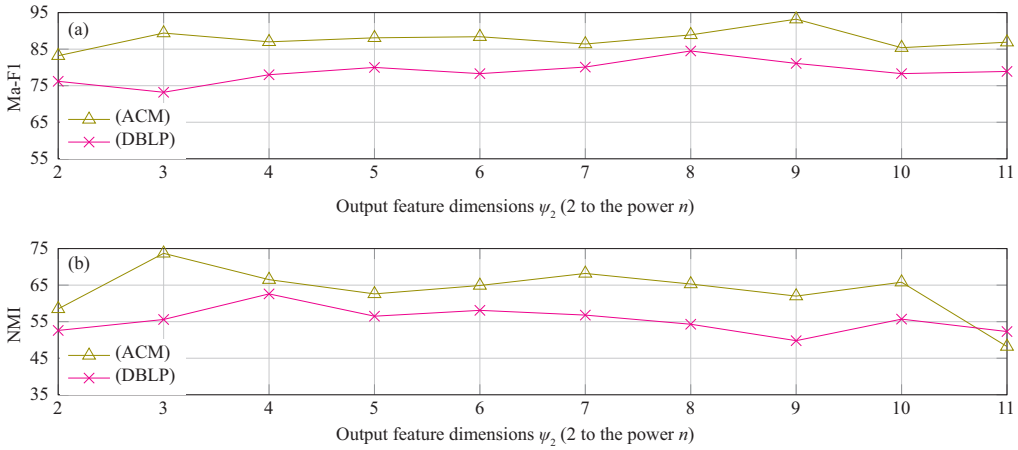

Figure 3 (Color online) Sensitivity analysis of the output feature dimensions ψ_2 on (a) node classification and (b) node clustering.

the sensitivity analysis of hidden unit dimensions ψ_1, output feature dimensions ψ_2, and diffusion distance K for the proposed UMGDN.

Effect of hidden unit dimensions ψ_1 and output feature dimensions ψ_2. Figures 2(a) and (b) report the node classification and clustering performance of the proposed UMGDN on the ACM and DBLP datasets when we change the hidden unit dimensions ψ_1 from 2^2 to 2^{11}. Figures 3(a) and (b) show the effect of output feature dimensions ψ_2 on the ACM and DBLP datasets for node classification and clustering performance. As we can observe in Figures 2 and 3, on the sensitivity analysis of the hidden unit dimensions ψ_1, too small hidden unit dimensions will hurt the performance of all downstream tasks. For example, UMGDN with $\psi_1 = 2^9$ and UMGDN with $\psi_1 = 2^{10}$ on all downstream tasks of the DBLP dataset obtain the best performance. On the sensitivity analysis of the output feature dimensions ψ_2, the node classification task needs a large output feature dimension ψ_2. Otherwise, a too large output feature dimension ψ_2 will hurt the node clustering performance of our method. For example, UMGDN with $\psi_2 = 2^9$ and UMGDN with $\psi_2 = 2^3$ obtain the best performance on the node classification and

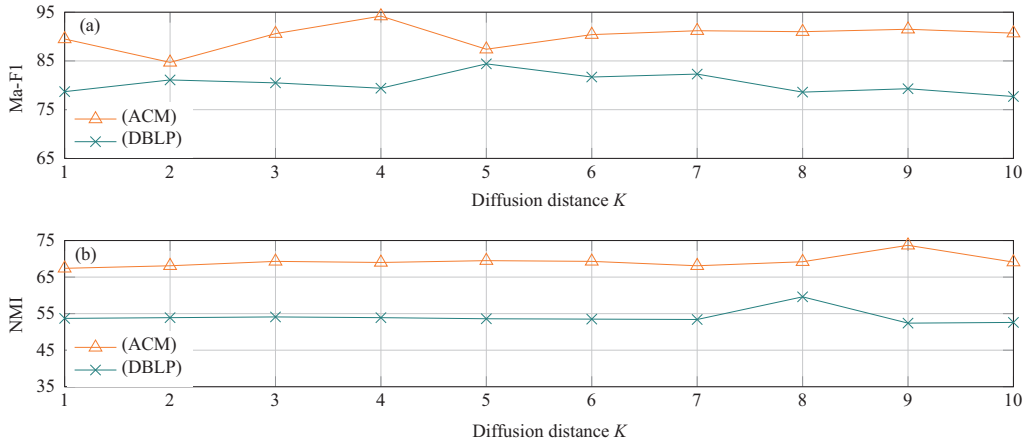

Figure 4 (Color online) Sensitivity analysis of the diffusion distance K on (a) node classification and (b) node clustering.

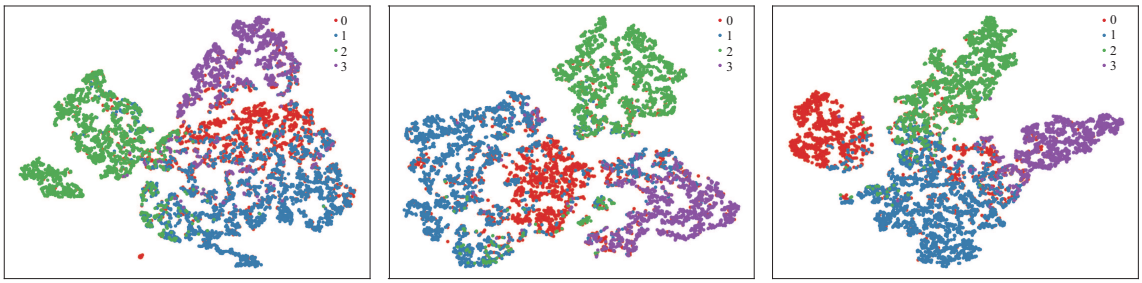

Figure 5 (Color online) t-SNE visualization of the test set feature embeddings from different UMGRL methods on node classification. Each color represents a different class. (a) $\text{DMGI}_{\text{attn}}$ (DBLP); (b) HDMI (DBLP); (c) UMGDN (DBLP).

clustering, respectively. The main reason is that the trained logistic regression on the node classification task can achieve second-dimensional reduction. The final node embeddings with small output feature dimensions ψ_2 cannot guarantee the effective discriminate ability of the trained logistic regression model.

Effect of diffusion distance K. Figures 4(a) and (b) illustrate the effect of diffusion distance K for the proposed UMGDN. From these results, we can observe that a too large diffusion distance K will damage the performance of node classification. For node clustering, it needs a too large diffusion distance K. The main reason is that each node with a too small diffusion distance K on the node clustering cannot take full advantage of long-range dependency relationships between nodes to make the updated low-dimensional node features have a robust discrimination ability. With the increase of diffusion distance K, the model's performance increases and then decreases. These results also suggest that reasonably choosing an appropriate diffusion distance K for each downstream task and dataset is important.

4.6 t-SNE visualization

To better show the superiority of the proposed method in MGRL, we simultaneously report the t-SNE visualization of the test set feature embeddings from two classical UMGRL ($\text{DMGI}_{\text{attn}}$, HDMI) and our UMGDN methods on node classification. For t-SNE, sample features belonging to the same class are expected to cluster together. From these results of Figure 5, we can see that the proposed UMGDN has a clearer inter-class boundary, which directly indicates that our method can learn more expressive node embeddings applied to the downstream task with the help of the adaptive diffusion propagation module and multi-level canonical correlation analysis loss.

5 Conclusion

Recently, self-supervised learning has received extensive exploration and has become the dominant method in multiplex graph representation learning, which is termed UMGRL. However, the superior performance

of most existing UMGRL methods highly depends on complex model architectures or designs. Moreover, these approaches only consider the influence of the direct neighbors for its embeddings in the feature aggregating process. To solve the above-mentioned problem, we propose a simple yet effective UMGDN supervised by multi-level canonical correlation analysis. Specifically, we remove the complex data augmentation, complex pretext tasks, and complex negative pairs construction operation for expanding the scope of application of the proposed UMGDN. And then, we divide the widely-used graph convolution layer into feature transform and propagation processes, and also propose the adaptive diffusion propagation to learn long-range dependency relationships between all nodes in the graph. Finally, a multi-level canonical correlation analysis loss is introduced to provide more supervision information for UMGDN by simultaneously maximizing the correlation of the learned node features from multiple graphs on the feature transform and propagation processes. To demonstrate the effectiveness of the proposed UMGDN, we conduct extensive experiments on four benchmarks and two graph-related tasks. Experiment results show that our proposed UMGDN without complex designs outperforms the existing UMGRL approaches.

Acknowledgements This work was supported in part by National Key Research and Development Program of China (Grant No. 2022YFF0712300), National Natural Science Foundation of China (Grant No. 62172177), Knowledge Innovation Program of Wuhan-Shuguang, Fundamental Research Funds for the Central Universities (HUST) (Grant No. 2022JYCXJJ034), and Open Research Fund from Shandong Provincial Key Laboratory of Computer Network (Grant No. SKLCN-2021-02).

References

1 Jin T S, Dai H Q, Cao L J, et al. Deepwalk-aware graph convolutional networks. Sci China Inf Sci, 2022, 65: 152104
2 Wu T T, Duan F Q, Chang L, et al. Human-object interaction detection via interactive visual-semantic graph learning. Sci China Inf Sci, 2022, 65: 160108
3 Feng J P, Wang X G, Liu W Y. Deep graph cut network for weakly-supervised semantic segmentation. Sci China Inf Sci, 2021, 64: 130105
4 Zhang K, Jiang H, Zhang J, et al. Semi-supervised medical report generation via graph-guided hybrid feature consistency. IEEE Trans Multimedia, 2024, 26: 904–915
5 Wu S, Sun F, Zhang W, et al. Graph neural networks in recommender systems: a survey. ACM Comput Surv, 2023, 55: 1–37
6 Jiao L, Chen J, Liu F, et al. Graph representation learning meets computer vision: a survey. IEEE Trans Artif Intell, 2023, 4: 2–22
7 Deng C, Yang X, Nie F, et al. Saliency detection via a multiple self-weighted graph-based manifold ranking. IEEE Trans Multimedia, 2019, 22: 885–896
8 Yang X, Gong Y, Liu W, et al. Semantic-preserving adversarial text attacks. IEEE Trans Sustain Comput, 2023, 8: 583–595
9 Fu S, Wang S, Liu W, et al. Adaptive graph convolutional collaboration networks for semi-supervised classification. Inf Sci, 2022, 611: 262–276
10 Fu S, Liu W, Tao D, et al. HesGCN: Hessian graph convolutional networks for semi-supervised classification. Inf Sci, 2020, 514: 484–498
11 Fu S, Liu W, Zhang K, et al. Semi-supervised classification by graph p-Laplacian convolutional networks. Inf Sci, 2021, 560: 92–106
12 Liang J Y, Liu X L, Bai L, et al. Incomplete multi-view clustering via local and global co-regularization. Sci China Inf Sci, 2022, 65: 152105
13 Liu Y, Tu W, Zhou S, et al. Deep graph clustering via dual correlation reduction. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2022. 36: 7603–7611
14 Georgantas C, Richiardi J. Multi-view omics translation with multiplex graph neural networks. In: Proceedings of Companion Proceedings of the Web Conference, 2022. 1030–1036
15 Yang C, Xiao Y, Zhang Y, et al. Heterogeneous network representation learning: a unified framework with survey and benchmark. IEEE Trans Knowl Data Eng, 2020, 34: 4854–4873
16 Maskey S, Levie R, Lee Y, et al. Generalization analysis of message passing neural networks on large random graphs. In: Proceedings of Advances in Neural Information Processing Systems, 2022
17 Schlichtkrull M, Kipf T N, Bloem P, et al. Modeling relational data with graph convolutional networks. In: Proceedings of the European Semantic Web Conference, 2018. 593–607
18 Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. In: Proceedings of the International Conference on Learning Representations, 2017
19 Yun S, Jeong M, Kim R, et al. Graph transformer networks. In: Proceedings of Advances in Neural Information Processing Systems, 2019. 32
20 Zhao J, Wang X, Shi C, et al. Heterogeneous graph structure learning for graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2021. 35: 4697–4705
21 Yu P, Fu C, Yu Y, et al. Multiplex heterogeneous graph convolutional network. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022. 2377–2387
22 Park C, Kim D, Han J, et al. Unsupervised attributed multiplex network embedding. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2020. 34: 5371–5378
23 Wang X, Liu N, Han H, et al. Self-supervised heterogeneous graph neural network with co-contrastive learning. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2021. 1726–1736
24 Pan E, Kang Z. Multi-view contrastive graph clustering. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 34: 2148–2159
25 Thakoor S, Tallec C, Azar M G, et al. Large-scale representation learning on graphs via bootstrapping. In: Proceedings of the International Conference on Learning Representations, 2022
26 Zhu Y, Xu Y, Yu F, et al. Graph contrastive learning with adaptive augmentation. In: Proceedings of the Web Conference, 2021. 2069–2080
27 Jing B, Park C, Tong H. HDMI: high-order deep multiplex infomax. In: Proceedings of the Web Conference, 2021. 2414–2424
28 Yang X, Liu W, Liu W, et al. A survey on canonical correlation analysis. IEEE Trans Knowl Data Eng, 2021, 33: 2349–2368
29 Feng Y, You H, Zhang Z, et al. Hypergraph neural networks. AAAI, 2019, 33: 3558–3565
30 Fu S, Liu W, Guan W, et al. Dynamic graph learning convolutional networks for semi-supervised classification. ACM Trans Multimedia Comput Commun Appl, 2021, 17: 1–13

31 Lei C, Fu S, Wang Y, et al. Self-supervised guided hypergraph feature propagation for semi-supervised classification with missing node features. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2023
32 Veličković P, Cucurull G, Casanova A, et al. Graph attention networks. In: Proceedings of the International Conference on Learning Representations, 2018
33 Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. In: Proceedings of Advances in Neural Information Processing Systems, 2017. 30
34 Chen J, Ma T, Xiao C. FastGCN: fast learning with graph convolutional networks via importance sampling. In: Proceedings of the International Conference on Learning Representations, 2018
35 Xu K, Hu W, Leskovec J, et al. How powerful are graph neural networks? In: Proceedings of the International Conference on Learning Representations, 2019
36 Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. In: Proceedings of Advances in Neural Information Processing Systems, 2016. 29
37 Gasteiger J, Weißenberger S, Günnemann S. Diffusion improves graph learning. In: Proceedings of Advances in Neural Information Processing Systems, 2019. 32
38 Wang G, Ying R, Huang J, et al. Direct multi-hop attention based graph neural network. 2020. ArXiv:2009.14332
39 Hotelling H. Relations between two sets of variates. In: Breakthroughs in Statistics. New York: Springer, 1992. 162–190
40 Du L, Wang H A, Zhang J, et al. Adaptive structured sparse multiview canonical correlation analysis for multimodal brain imaging association identification. Sci China Inf Sci, 2023, 66: 142106
41 Liu W, Yang X, Tao D, et al. Multiview dimension reduction via Hessian multiset canonical correlations. Inf Fusion, 2018, 41: 119–128
42 Witten D M, Tibshirani R, Hastie T. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. Biostatistics, 2009, 10: 515–534
43 Akaho S. A kernel method for canonical correlation analysis. In: Proceedings of the International Meeting of Psychometric Society, 2006
44 Yuan Y H, Li Y, Shen X B, et al. Laplacian multiset canonical correlations for multiview feature extraction and image recognition. Multimed Tools Appl, 2017, 76: 731–755
45 Chang X, Xiang T, Hospedales T M. Scalable and effective deep CCA via soft decorrelation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018. 1488–1497
46 Yang X, Liu W, Tao D, et al. Canonical correlation analysis networks for two-view image recognition. Inf Sci, 2017, 385-386: 338–352
47 Yang X, Liu W, Liu W. Tensor canonical correlation analysis networks for multi-view remote sensing scene recognition. IEEE Trans Knowl Data Eng, 2022, 34: 2948–2961
48 Liu M, Gao H, Ji S. Towards deeper graph neural networks. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data, 2020. 338–348
49 Wang X, Ji H, Shi C, et al. Heterogeneous graph attention network. In: Proceedings of the World Wide Web Conference, 2019. 2022–2032
50 Zhang Z. Improved Adam optimizer for deep neural networks. In: Proceedings of the IEEE/ACM International Symposium on Quality of Service, 2018. 1–2
51 Perozzi B, Al-Rfou R, Skiena S. DeepWalk: online learning of social representations. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014. 701–710
52 Grover A, Leskovec J. node2vec: scalable feature learning for networks. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. 855–864
53 Zhang Z, Yang H, Bu J, et al. ANRL: attributed network representation learning via deep neural networks. In: Proceedings of the International Joint Conference on Artificial Intelligence, 2018. 18: 3155–3161
54 Zhuang C, Ma Q. Dual graph convolutional networks for graph-based semi-supervised classification. In: Proceedings of the World Wide Web Conference, 2018. 499–508
55 Meng Z, Liang S, Bao H, et al. Co-embedding attributed networks. In: Proceedings of the ACM International Conference on Web Search and Data Mining, 2019. 393–401
56 Velickovic P, Fedus W, Hamilton W L, et al. Deep graph infomax. In: Proceedings of the International Conference on Learning Representations, 2019
57 You Y, Chen T, Sui Y, et al. Graph contrastive learning with augmentations. In: Proccedings of Advances in Neural Information Processing Systems, 2020. 33: 5812–5823
58 Lee N, Hyun D, Lee J, et al. Relational self-supervised learning on graphs. In: Proceedings of the ACM International Conference on Information and Knowledge Management, 2022. 1054–1063
59 Zhang H, Qiu L, Yi L, et al. Scalable multiplex network embedding. In: Proceedings of the International Joint Conference on Artificial Intelligence, 2018. 18: 3082–3088
60 Chu X, Fan X, Yao D, et al. Cross-network embedding for multi-network alignment. In: Proceedings of the World Wide Web Conference, 2019. 273–284
61 Ma Y, Wang S, Aggarwal C C, et al. Multi-dimensional graph convolutional networks. In: Proceedings of the SIAM International Conference on Data Mining, 2019. 657–665
62 Lin Z, Kang Z. Graph filter-based multi-view attributed graph clustering. In: Proceedings of the International Joint Conference on Artificial Intelligence, 2021. 2723–2729
63 Sun Y, Zhu D, Du H, et al. MHNF: multi-hop heterogeneous neighborhood information fusion graph representation learning. IEEE Trans Knowl Data Eng, 2023, 35: 7192–7205