IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 February 2024, accepted 15 March 2024, date of publication 21 March 2024, date of current version 3 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3380012

RESEARCH ARTICLE

Improving Variational Graph Autoencoders With Multi-Order Graph Convolutions

LINING YUAN[1], PING JIANG[1], ZHU WEN[1], AND JIONGHUI LI[2]
[1]School of Public Security Big Data Modern Industry, Guangxi Police College, Nanning 530028, China
[2]Suzhou Keda Technology Corporation Ltd., Suzhou 215011, China

Corresponding author: Zhu Wen (wenzhu@gxjcxy.edu.cn)

This work was supported in part by the Key Research and Development Program of Guangxi under Grant AB22035034, and in part by the Social Science Fund of Guangxi under Grant 23FTQ005.

ABSTRACT Variational Graph Autoencoders (VAGE) emerged as powerful graph representation learning methods with promising performance on graph analysis tasks. However, existing methods typically rely on Graph Convolutional Networks (GCN) to encode the attributes and topology of the original graph. This strategy makes it difficult to fully learn high-order neighborhood information, which weakens the capacity to learn higher-quality representations. To address the above issues, we propose the Multi-order Variational Graph Autoencoders (MoVGAE) with co-learning of first-order and high-order neighborhoods. GCN and Multi-order Graph Convolutional Networks (MoGCN) are utilized to generate the mean and variance for the variational autoencoders. Then, MoVGAE uses the mean and variance to calculate node representations. Specifically, this approach comprehensively encodes first-order and high-order information in the graph data. Finally, the decoder reconstructs the adjacency matrix by performing the inner product of the representations. Experiments with the proposed method were conducted on node classification, node clustering, and link prediction tasks on real-world graph datasets. The results demonstrate that MoVGAE achieves state-of-the-art performance compared to other baselines in various tasks. Furthermore, the robustness analysis verifies that MoVGAE has obvious advantages in the processes of graph data with insufficient attributes and topology.

INDEX TERMS Variational graph autoencoders, graph convolutional networks, multi-order neighborhood, high-order information, graph representation learning.

I. INTRODUCTION

Many real-world datasets can be represented as graphs, such as social networks [1], criminal networks [2], and citation networks [3]. The entities are represented as nodes, and the relationships between entities are represented as edges. Nowadays, graph data analysis is attracting increasing attention, particularly in downstream tasks such as node classification [4] and link prediction [5]. Traditional machine learning methods rely on manually designed features to extract specific topology information from adjacency matrix. Due to the high-dimensional nonlinearity of graph data, these methods usually have high computational complexity and memory requirements [6]. Recently, graph representation learning [7] has received increasing attention in light of its powerful representation ability and its favorable performance. It can effectively encode the topology and attribute information of the graph, and generate low-dimensional vector representations that are easy to process.

At present, graph representation learning methods are mainly divided into three categories: matrix factorization methods [5], random walk methods [8], and deep learning methods [7]. Methods based on matrix factorization learn the representations by decomposing the adjacency matrix, such as GraRep++ [9] and NGE [10]. Methods based on random walk learn representations by constructing local neighborhood connectivity through random walk, such as Deepwalk [11] and node2vec [12]. Methods based on deep

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa M. Fouda.

learning learn representations by combining Graph Neural Networks (GNN) [13] and autoencoders [14]. Variational Graph Autoencoders (VGAE) [15] is an important deep learning method. It introduces Graph Convolutional Networks (GCN) [16] as the encoder for Variational Autoencoders (VAE) [17]. This approach enables the learning of the probability distribution of node attributes and structures. The node representations are computed by the distribution, and then reconstruct the adjacency matrix by the inner product of these representations.

Over the past few years, VGAE and its variants emerged as powerful methods for graph representation learning, showing promising performance on challenging graph analysis tasks. Linear-VGAE [18] replaces the GCN encoder with simple linear models, which involve fewer operations and parameters. In link prediction, Linear-VGAE achieves competitive performance compared to GCN-based methods. Variational Graph Normalized Autoencoders (VGNAE) [19] modifies the GCN encoder with L2-normalization to improve the representations of isolated nodes. OSA-VGAE [20] applies one-shot aggregation of deep vision models to prevent oversmoothing [21] and gradient vanishing in multi-layer GCN encoder, enhancing the representation ability of deep models. Multi-Scale Variational Graph Autoencoders (MSVGAE) [22] generates multiple sets of low-dimensional vectors at different scales to represent the mixed probability distribution of the original graph and performs multiple samplings in each dimension. Furthermore, an attribute reconstruction strategy is introduced to enhance the learning of topology information. GC-VGAE [23] reconstructs the adjacency matrix and attribute matrix simultaneously by modifying the decoder. This approach enhances the capability of GC-VGAE to preserve the original graph information.

When learning low-dimensional node representations, these methods utilize GCN as the encoder and the adjacency matrix as the convolution kernel to extract first-order neighborhood information. For high-order neighborhood information, it can only be extracted by stacking multi-layer GCN. However, increasing the number of GCN layers can lead to over-smoothing, making the node representations indistinguishable. In addition, existing methods have greatly restricted the processes of graph data with insufficient attributes and topology.

To address the above issues, we propose an innovative graph representation learning method, termed Multi-order Variational Graph Autoencoders (MoVGAE). The purpose of MoVGAE is to generate representations by co-learning first-order and high-order neighborhood information. Firstly, we designed the Multi-order Graph Convolutional Networks (MoGCN) to extract high-order information. Secondly, GCN and MoGCN are utilized to generate the mean and variance of the variational autoencoders. MoVGAE uses the mean and variance to calculate node representations. Finally, the decoder reconstructs the adjacency matrix by performing the inner product of the representations. The main contributions of our work are summarized as follows:

- We design a new MoGCN for graph representation learning, which is capable of extracting high-order neighborhood information.
- MoVGAE uses GCN and MoGCN as encoders to generate node representations that co-learn first-order and high-order information.
- Extensive experiments on three benchmark datasets demonstrate that MoVGAE achieves state-of-the-art performance while maintaining high generalization and scalability.
- The robustness analysis confirms that MoVGAE has obvious advantages in handling graph data with insufficient attributes and topology.

II. RELATED WORK

The first-order neighborhood represents the node pairs directly connected by edges in the graph, reflecting the local structure. The high-order neighborhood represents the node pairs with r-hop relationships, reflecting the global structure. Due to the sparsity of graphs, the real-world datasets often contain many unobserved or missing links [24]. This phenomenon indicates that the first-order neighborhood insufficient for representing the topology.

Several social science studies have analyzed the theory of high-order relations. For example, the relationship of two individuals in the social network is associated with the extent of overlap in their friendships [25]. If two people have many mutual friends, they are likely to have the same interests and become friends. In addition, Figure 1 presents the statistics of first-order and second-order node pairs for the common datasets KarateClub [26] and Football [27]. The number of second-order node pairs is significantly more than that of first-order node pairs. To enhance topology preservation, it is essential to incorporate high-order information as supplements to first-order information.

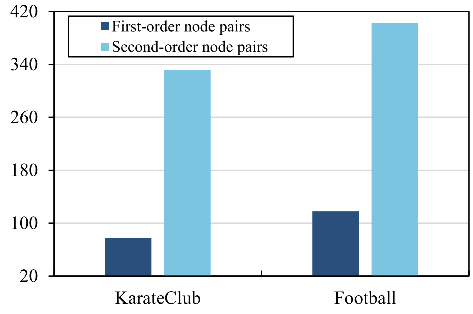


FIGURE 1. Statistics of first-order and second-order node pairs.

In recent years, many methods introduce high-order information into graph representation learning. Structural Deep Network Embedding (SDNE) [15] modifies the reconstruction loss by imposing a higher penalty on the non-zero elements of the adjacency matrix, so that the node representations with second-order proximity are closer in the feature space. Large-scale Information Network Embedding (LINE) [28] sets each node as the context for the others.

The similarity of the context between two nodes means that the neighborhood of the two nodes is similar, so that the node representations are also similar in the feature space. GraRep [9] constructs a set of transition probability representations with different orders, and then fuses them to generate node representations that preserve high-order information. k-GNN [29] uses the multi-dimensional graph isomorphism heuristic, which considers high-order graph structures at multiple scales. This approach confirms that higher-order information is beneficial for graph-level tasks. HGRN [30] updates the topology by aggregating node representations with small step sizes, adds links with multi-hop neighbors, and adaptively aggregates features from high-order neighbors through an attention mechanism. Motivated by the above methods, we modify the GCN and propose a novel MoGCN which can extract high-order information.

III. PRELIMINARIES

Based on the vanilla autoencoder, VAE utilizes neural networks to construct two probability density distribution models. The inference network encodes the original data and generates the variational probability distribution. The generative network produces an approximate distribution of the original data based on the variational probability distribution. In short, the inference network utilizes the encoder to calculate the mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}$, and the generative network utilizes the decoder to generate the reconstruction probability distribution.

VAE adds suitable noise during the vanilla autoencoder encoding process. Firstly, VAE utilizes the encoder to calculate the $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3)$ and $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \sigma_3)$. σ is used to regulate the weight of the noise $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \varepsilon_3)$ and control the degree of noise interference. To ensure that the weights allocated to the noise are positive, the $\boldsymbol{\sigma}$ is exponentiated. Then, $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ are combined to generate representation $\boldsymbol{y} = (y_1, y_2, y_3)$ of the input data. Finally, the $\boldsymbol{y}$ is fed into the decoder for reconstruction. Compared to vanilla autoencoder, VAE not only utilizes a reconstruction loss function, but also introduces constraints on the noise:

$$\sum_{i=1}^{3} (e^{\sigma_i} - (1+\sigma_i) + (\mu_i)^2) \tag{1}$$

Assuming (1) is not added, in order to ensure the quality of the generated results, the model aims to minimize the interference of noise during the encoding process by assigning smaller weights to the noise. This usually leads to good training performance but poor predictability of the generated results. After adding (1), it can constrain the variance encoding from rapidly tending towards negative infinity, thus serving as regularization term. Combining Gaussian mixture models, the reconstruction loss and noise constraint of VAE are rewritten in the form of probability distributions and Kullback-Leibler (KL) divergence:

$$\mathcal{L}_{VAE} = \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x})}[\log(\boldsymbol{x}|\boldsymbol{y})] - \mathrm{KL}[q(\boldsymbol{y}|\boldsymbol{x})||p(\boldsymbol{y})] \tag{2}$$

VGAE is an unsupervised learning framework for graph-structured data based on the VAE. It utilizes a GCN encoder and a simple inner product decoder to learn interpretable latent representations for the original graph. In order to improve performance, VGNAE proposes a novel GCN called a Graph Normalized Convolutional Networks (GNCN) [19] to generate better embeddings for isolated nodes. It modifies the encoder of VGAE by using GCN and GNCN to generate $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ separately, thereby achieving competitive results for link prediction tasks. Motivated by VGAE and VGNAE, we utilize GCN and MoGCN as encoders in MoVGAE to capture both first-order and high-order information.

IV. THE MODEL

A. OVERALL FRAMEWORK

MoVGAE takes the attribute matrix $\boldsymbol{X}$ and the adjacency matrix $\boldsymbol{A}$ as input to the encoder. It extracts features and reduces the dimensionality of the original graph to generate low-dimensional representations. The overall framework of the model is shown in Figure 2. MoVGAE adopts a VAE structure, where the encoder utilizes multi-layer GCN to calculate the mean $\boldsymbol{\mu}$ and multi-layer MoGCN to calculate the variance $\boldsymbol{\sigma}$. Then, the mean $\boldsymbol{\mu}$, variance $\boldsymbol{\sigma}$, and Gaussian noise $\boldsymbol{\varepsilon}$ are used to calculate node representations $\boldsymbol{Y}$. The decoder utilizes the inner product of $\boldsymbol{Y}$ to reconstruct the adjacency matrix. Finally, MoVGAE uses the same probabilistic reconstruction loss and noise-constrained optimization parameters as VGAE.

B. MULTI-ORDER GRAPH CONVOLUTIONAL NETWORKS

In this section, we introduce MoGCN, which is capable of extracting high-order neighborhood information. Early GCNs were commonly described as linear shift-invariant operations on the adjacency matrix [31], utilizing adjacency matrix polynomials to build neural network model $f_{\boldsymbol{\Theta}}(\boldsymbol{X}, \boldsymbol{A})$ [32]:

$$f_{\boldsymbol{\Theta}}(\boldsymbol{X}, \boldsymbol{A}) = \sum_{k=0}^{K} \boldsymbol{\theta}_k \boldsymbol{A}^k \boldsymbol{X} \tag{3}$$

where $\boldsymbol{\Theta} = (\theta_0, \ \theta_1, \cdots, \ \theta_K)^T$ is the weights of different order adjacency matrices. The effect of $f_{\boldsymbol{\Theta}}(\boldsymbol{X}, \boldsymbol{A})$ is not ideal, and the computational complexity is relatively high. To reduce complexity, the first-order neighborhood is used instead of the multi-order linear combination to construct the neural network model $f_1(\boldsymbol{X}, \boldsymbol{A})$ [16]:

$$f_1(\boldsymbol{X}, \boldsymbol{A}) = \theta(\boldsymbol{I} + \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{A} \boldsymbol{D}^{-\frac{1}{2}})\boldsymbol{X} \tag{4}$$

where $\boldsymbol{D}$ is the degree matrix for symmetric normalization of $\boldsymbol{A}$, and $\boldsymbol{I}$ is the identity matrix. The information from high-order neighbors can only be transmitted by stacking multiple layers of GCN. However, having too many layers can result in over-smoothing [33], [34], causing node representations to become indistinguishable. The first-order neighborhood alone cannot fully represent the topology of

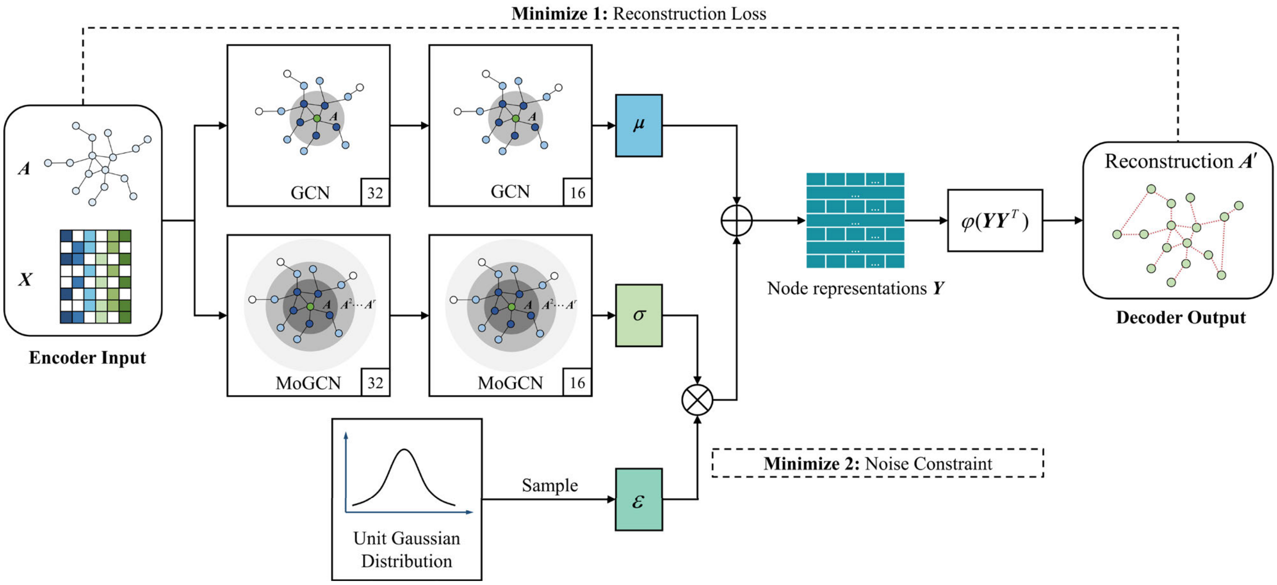


FIGURE 2. The framework of MoVGAE.

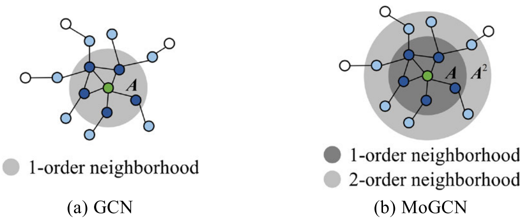
the graph, it is necessary to introduce high-order information as a supplement. Combining first-order GCN and high-order GraRep, we propose a novel neural network model $f_k(\boldsymbol{X}, \boldsymbol{A})$:

$$f_k(\boldsymbol{X}, \boldsymbol{A}) = \theta[\boldsymbol{I} + \boldsymbol{D}_k^{-\frac{1}{2}}(\boldsymbol{A} + \widehat{\boldsymbol{A}^k})\boldsymbol{D}_k^{-\frac{1}{2}}]\boldsymbol{X} \tag{5}$$

where $\widehat{\boldsymbol{A}^k}$ is the symmetric normalized k-order adjacency matrix. We modified (5) based on GCN and derived the propagation equation between MoGCN layers as follows:

$$\boldsymbol{H}^{(l+1)} = \tilde{\boldsymbol{D}}_k^{-\frac{1}{2}}\tilde{\boldsymbol{A}}_k\tilde{\boldsymbol{D}}_k^{-\frac{1}{2}}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)} \tag{6}$$

where, $\tilde{\boldsymbol{A}}_k = \boldsymbol{I} + \boldsymbol{A} + \boldsymbol{A}^k$, $\tilde{\boldsymbol{D}}_k$ is the degree matrix of $\tilde{\boldsymbol{A}}_k$, $\boldsymbol{W}^l$ is the weight matrix of the l^{th} layer, $\boldsymbol{H}^l$ is the feature matrix of the l^{th} layer, and $\boldsymbol{H}^{(0)}$ is $\boldsymbol{X}$. Figure 3 illustrates the extent of information propagation between nodes in GCN and MoGCN(2-order). The convolutional kernel of MoGCN introduces $\boldsymbol{A}^2$, enabling it to extract second-order neighborhood features even without direct connections. In addition, MoGCN utilizes high-order information to enhance the feature extraction capability of GCN for long-distance nodes.



(a) GCN (b) MoGCN

FIGURE 3. The information propagation between nodes in GCN and MoGCN.

C. MULTI-ORDER VARIATIONAL GRAPH AUTOENCODERS

In this section, we introduce the graph representation learning model MoVGAE. The encoder utilizes 2-layer GCN to compute the mean μ and 2-layer MoGCN to compute the variance σ:

$$\begin{aligned}\boldsymbol{\mu} &= \tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}[\delta(\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{W}_{\mu}^{(1)})]\boldsymbol{W}_{\mu}^{(2)} \\ \log\boldsymbol{\sigma} &= \tilde{\boldsymbol{D}}_k^{-\frac{1}{2}}\tilde{\boldsymbol{A}}_k\tilde{\boldsymbol{D}}_k^{-\frac{1}{2}}[\delta(\tilde{\boldsymbol{D}}_k^{-\frac{1}{2}}\tilde{\boldsymbol{A}}_k\tilde{\boldsymbol{D}}_k^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{W}_{\sigma}^{(1)})]\boldsymbol{W}_{\sigma}^{(2)}\end{aligned} \tag{7}$$

where, $\tilde{\boldsymbol{A}} = \boldsymbol{I} + \boldsymbol{A}$, $\tilde{\boldsymbol{D}}$ is the degree matrix of $\tilde{\boldsymbol{A}}$, δ is the ReLU activation function. Then, the mean $\boldsymbol{\mu}$, variance $\boldsymbol{\sigma}$, and Gaussian noise $\boldsymbol{\varepsilon}$ are used to calculate node representations $\boldsymbol{Y}$:

$$\boldsymbol{Y} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\varepsilon} \tag{8}$$

where, $\boldsymbol{Y} = \{\boldsymbol{y}_i\}_{i=1}^N$, N is the number of nodes, $\boldsymbol{y}_i$ is the representation of node i, $\odot$ is Hadamard product. Next, $\boldsymbol{Y}$ will be reparameterized to construct the probability distribution of nodes in the latent space [15]:

$$\begin{aligned} q(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{A}) &= \prod_{i=1}^{N} q(\boldsymbol{y}_i|\boldsymbol{X}, \boldsymbol{A}) \\ q(\boldsymbol{y}_i|\boldsymbol{X}, \boldsymbol{A}) &= \mathcal{N}(\boldsymbol{\mu}_i, \text{diag}(\boldsymbol{\sigma}_i^2)) \end{aligned} \tag{9}$$

The decoder utilizes probabilistic node representations to generate the reconstructed adjacency matrix $\boldsymbol{A'}$:

$$p(\boldsymbol{A'}|\boldsymbol{Y}) = \prod_{i=1}^{N}\prod_{j=1}^{N} p(A'_{ij}|\boldsymbol{y}_i, \boldsymbol{y}_j) \tag{10}$$

where A_{ij} is the element of $\boldsymbol{A'}$. MoVGAE is trained using probabilistic reconstruction loss and noise constraints. Finally, the optimization function for model training is:

$$\mathcal{L} = \mathbb{E}_{q(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{A})}[\log p(\boldsymbol{A'}|\boldsymbol{Y})] - \text{KL}[q(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{A})||p(\boldsymbol{Y})] + \mathcal{L}_{reg} \tag{11}$$

where, $\mathcal{L}_{reg}$ is an L2-norm regularization to prevent overfitting, which is defined as follows:

$$\mathcal{L}_{reg} = \frac{1}{2}\sum_{k=1}^{K}\left\|\boldsymbol{W}^{(k)}\right\|_F^2 \tag{12}$$

D. COMPLEXITY ANALYSIS

In the training process of MoVGAE, the calculation of high-order adjacency matrix and the normalization of attribute matrix and adjacency matrix are non-parametric processes, so the computational complexity is mainly influenced by the GCN encoder and MoGCN encoder. The complexity of GCN is usually represented as $O(|\boldsymbol{E}|)$, where $\boldsymbol{E}$ represents the set of edges. Referring to GCN, the complexity of MoGCN can be represented as $O(|\boldsymbol{E}_k|)$. Since MoGCN utilizes k-order adjacency matrix for feature extraction, the number of edges in the set will exponentially increase with the increase of k. In addition, most VGAE variants use the complete graph as input, making it difficult to perform calculations on large-scale graph data. The method of calculating high-order adjacency matrix through power exponentiation in MoVGAE is even more difficult to implement.

V. EXPERIMENTS

A. DATASETS

To verify the effectiveness of baselines and MoVGAE in graph analysis tasks, we conducted experiments using five benchmark graph datasets. The statistics of the datasets are shown in Table 1. Considering the characteristics of these datasets, we select different datasets to evaluate the performance of different tasks. The detailed descriptions are listed as follows:

TABLE 1. The statistics of graph datasets.

Dataset	#Nodes	#Edges	#Attriubutes	#Labels
Cora	2708	5429	1433	7
Citeseer	3327	4732	3703	6
Pubmed	19717	44338	500	3
Wikipedia	2405	17981	4973	19
BlogCatalog	5196	171743	8189	6

1) CITATION NETWORK

Cora, Citeseer, and Pubmed [35]. In these datasets, nodes represent papers, edges represent citation relationships between different papers, node attributes represent the content of the papers, and node labels represent the research topics.

2) WEB NETWORK

Wikipedia [36]. In the dataset, nodes represent web pages, edges represent hyperlinks between different web pages, node attributes represent the content of the web pages, and node labels represent the category.

3) SOCIAL NETWORK

BlogCatalog [37]. In the dataset, nodes represent bloggers, edges represent social relationships between bloggers, node attributes represent keywords of blogger information, and node labels represent the topic categories of blog posts.

B. BASELINES

We compare our model with VGAE and its variants:

- **VGAE** [15]**:**The model that combines the graph domain with the variational autoencoder utilizes the topology and attributes information of the graph.
- **VGNAE** [19]**:**The model uses L2 regularization to modify the GCN encoder in VGAE, enhancing the embedding representation of isolated nodes.
- **GC-VGAE** [23]**:** The model modifies the optimization method of VGAE by introducing a symmetric attribute decoder, which can simultaneously reconstruct the attribute matrix and the adjacency matrix.
- **MSVGAE** [22]**:**The model modifies the feature encoding process of VGAE in a multi-scale and equivariant manner, while also utilizing attribute information to enhance the learning of topology.
- **OSA-VGAE** [20]: The model introduces one-shot aggregation to prevent over-smoothing and gradient vanishing in the encoder of VGAE, enhancing the representation capability of deep models.

C. EVALUATION METRICS AND SETUPS

We use Micro-F1 and Macro-F1 as the evaluation metric for node classification:

$$\begin{aligned} \mathrm{Micro}-\mathrm{F1} &= \frac{2^{*}P^{*}R}{P+R} \\ \mathrm{Macro}-\mathrm{F1} &= \frac{\sum_{l\in L}\mathrm{F1}(l)}{|L|} \end{aligned} \tag{13}$$

where, P is precision, R is recall, and $\mathrm{F1}(l)$ is the F1 score for label l. Micro-F1 and Macro-F1 are commonly used metrics for multi-class classification tasks. Micro-F1 takes into account the number of nodes in each class during the calculation, making it suitable for imbalanced data distributions. Macro-F1 treats each class equally during the calculation and is more influenced by classes with high precision and recall values.

We use NMI (normalized mutual information) as the evaluation metric for node clustering:

$$\mathrm{NMI} = 1 - \frac{H(M_1|M_2)_{norm} + H(M_2|M_1)_{norm}}{2} \tag{14}$$

where $H(\cdot)$ is the information entropy. NMI is used to measure the similarity between the clustering results of M_1 and M_2. A higher NMI value indicates that the clustering result shares more information with the true result, leading to a better clustering effect of the model.

We compare the performance of different methods based on their ability to correctly classify edges and non-edges.

Therefore, we use AUC (the area under the receiver operating characteristic curve) and AP (the area under the precision-recall rate curve) as evaluation metrics for link prediction. AUC and AP are a common metrics of binary classification task. AUC sets the threshold just below each positive example and calculates the average recall rate of the negative class, which can provide a reasonable evaluation of the classifier in the case of imbalanced samples. AP sets the threshold just below each positive example and calculates the average precision rate of the positive class, which measures the classification performance of the model for each class.

To ensure the fairness of the experiment, both baselines and MoVGAE are initialized with the hyperparameters in VGAE. The hidden layer dimension of the encoder is set to 32, and the dimensions of the mean and variance vectors are set to 16. During training, the Adam optimizer [38] is used to update model parameters with learning rate of 0.01 and 200 iterations. In addition, the order of the MoGCN encoder in MoVGAE is set to 2.

D. NODE CLASSIFICATION

In this section, we evaluated the performance through node classification tasks. The node representations generated by each model were used as input for Support Vector Machines (SVM) [39]. We randomly sampled 5% to 25% of the labels at 5% intervals to create the training set, and then randomly selected 50% of the labels from the remaining nodes as the test set. The models used the same data set division, and we recorded the Micro-F1 (%) and Macro-F1 (%) scores. The results of node classification are shown in Figures 4∼8.

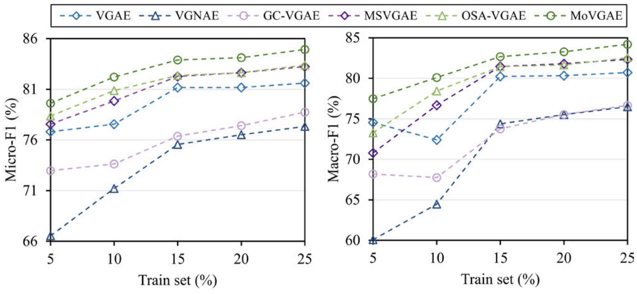


FIGURE 4. Experimental results of node classification on Cora.

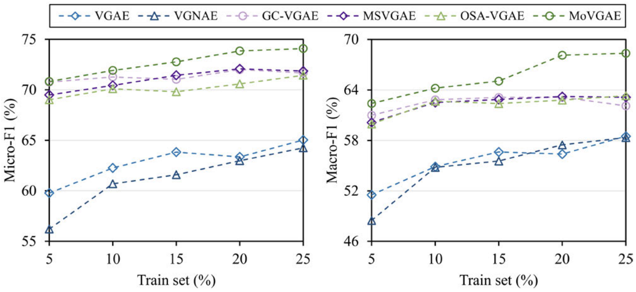


FIGURE 5. Experimental results of node classification on Citeseer.

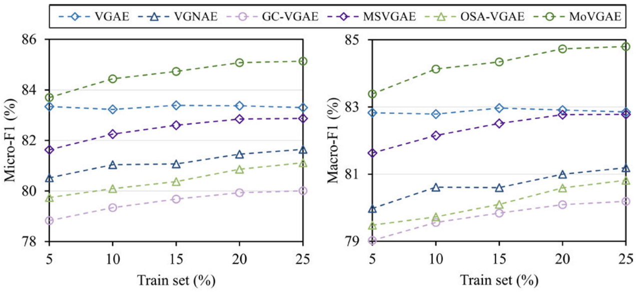


FIGURE 6. Experimental results of node classification on Pubmed.

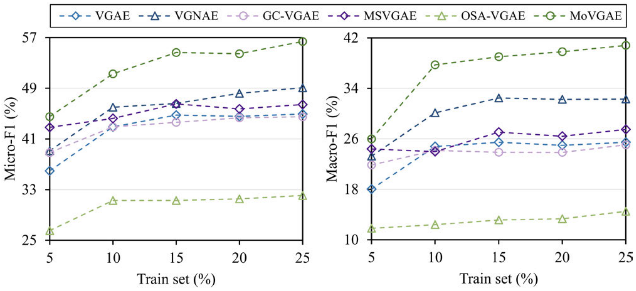


FIGURE 7. Experimental results of node classification on Wikipedia.

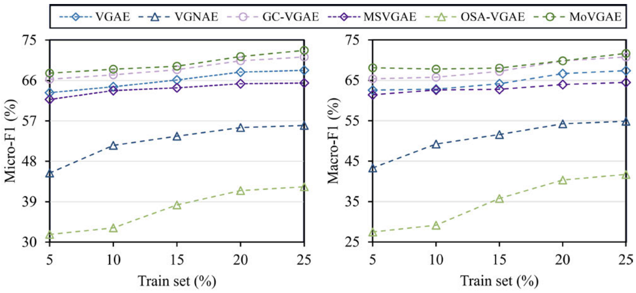


FIGURE 8. Experimental results of node classification on BlogCatalog.

The figures demonstrate that the proposed MoVGAE model outperforms other comparison baselines in node classification tasks, and show superior performance. Besides, MoVGAE shows stronger generalization on different datasets compared to the baselines. The results indicate that introducing high-order information can improve the model's representation capability, preserve rich node similarity and diversity information in the generated low-dimensional representation, and enhance the performance of node classification. As the training set labels increase, the classification results of various models show an upward trend. However, the improvement of some baselines is not significant, and the performance cannot be enhanced with the increase of supervised information. Compared to baselines, MoVGAE can effectively utilize both first-order and higher-order information to predict labels for unlabeled nodes. It can achieve better performance using only 10% of the training labels (as shown in Table 2, **bold** indicates the best results, and underline indicates the second-best results), further indicating that the

TABLE 2. Experimental results of node classification (10% of training data).

Methods	Cora		Citeseer		Pubmed		Wikipedia		BlogCatalog	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
VGAE	77.55	72.39	62.26	54.89	83.23	82.79	42.91	24.77	64.61	62.81
VGNAE	71.20	64.47	60.69	54.80	81.04	80.61	45.99	30.13	51.56	49.23
GC-VGAE	73.63	67.74	71.28	62.82	79.34	79.56	42.91	24.15	67.28	65.78
MSVGAE	79.84	76.68	70.44	62.51	82.25	82.15	44.23	23.97	63.77	62.63
OSA-VGAE	80.87	78.44	70.11	62.66	80.09	79.73	31.28	12.40	33.18	29.16
MoVGAE	**82.20**	**80.10**	**71.92**	**64.22**	**84.44**	**84.13**	**51.28**	**37.71**	**68.51**	**67.79**

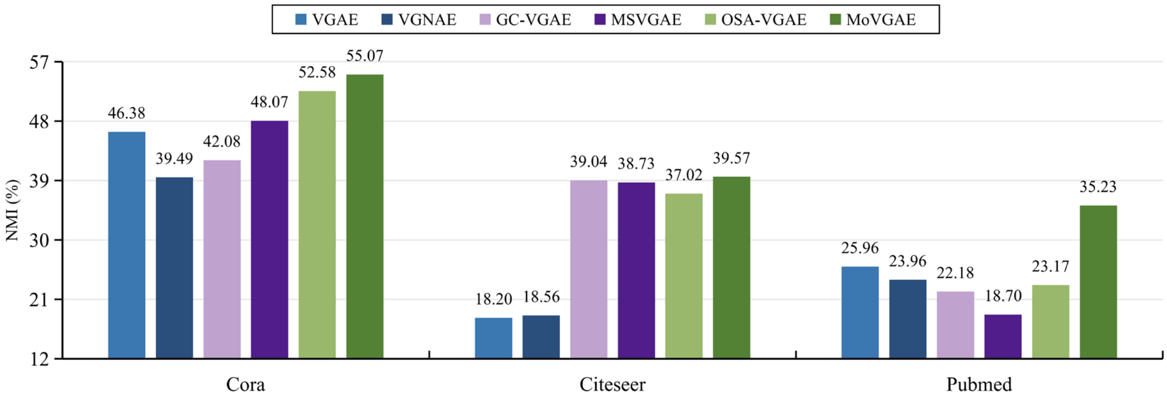


FIGURE 9. Experimental results of node clustering.

encoding method can enhance the distinguishability of node representations.

E. NODE CLUSTERING

In this section, we evaluated the performance through node clustering tasks [40]. We utilize node representations as input for the k-means++ [41] to perform unsupervised node clustering and record the NMI (%). In the k-means++, the value of k is set to the number of label classes in datasets. The results of node clustering are shown in Figure 9.

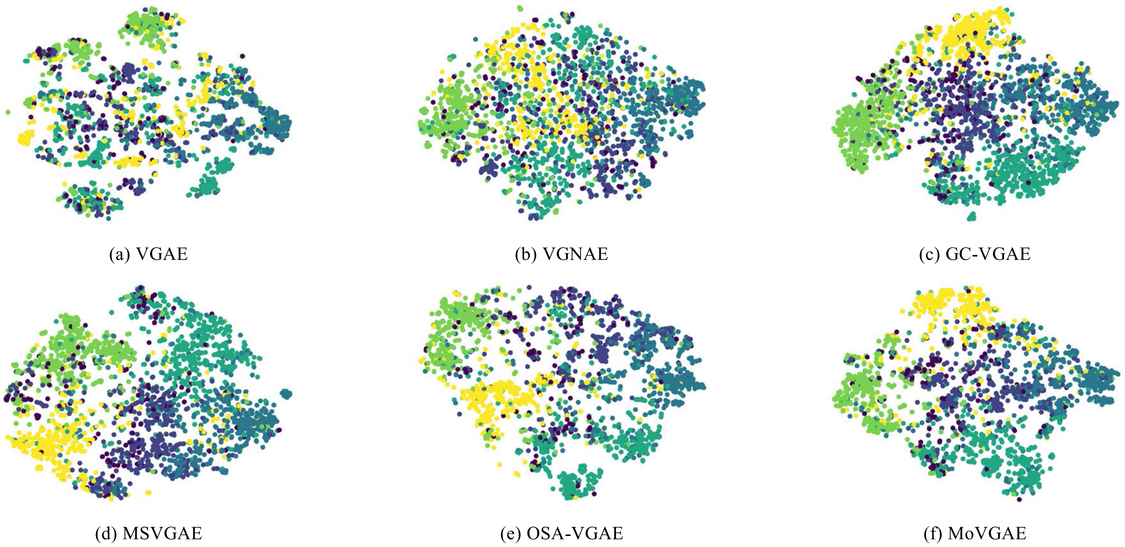
The figure demonstrates that MoVGAE outperforms other comparison baselines in node clustering tasks, and show superior performance. Besides, compared to the vanilla VGAE, MoVGAE significantly improves the performance of node clustering. The results indicate that introducing high-order information can improve the model's preservation of global structure, retain richer community information in node representations, make nodes within the same community closer in the feature space, and enhance the performance of node clustering.

F. LINK PREDICTION

In this section, we evaluated the performance through link prediction tasks. Firstly, remove 20% of the links and randomly sample 20% of the non-links (node pairs without links) to construct the test set. Then, use the remaining 80% of the links to train the model. Finally, the reconstructed adjacency matrix for link prediction is generated through the inner product of node representations. Each model uses the same data partition and records the AUC (%) and AP (%) values. The results of the link prediction are shown in Table 3 (**bold** indicates the best result and underline indicates the second result).

The table demonstrates that MoVGAE achieves state-of-the-art performance compared to other baselines in link prediction tasks. The results indicate that introducing high-order information can improve the model's capacity to maintain and infer local topological structures, preserving richer neighborhood information in low-dimensional node representations, and enhancing the performance of link prediction. Furthermore, MoVGAE achieved the best performance in different tasks on different datasets. The results indicate that introducing high-order information can simultaneously enhance the model's generalization and versatility.

MoVGAE and others are evolved from VGAE, but the difference is that they are improved in various ways. In node classification, node clustering, and link prediction, the results of MoVGAE show significant improvement compared to VGAE. Not only does it outperform current state-of-the-art baselines, but its performance is also more stable. This demonstrates that incorporating self-supervised information can improve the model's capacity to represent the original



(a) VGAE

(b) VGNAE

(c) GC-VGAE

(d) MSVGAE

(e) OSA-VGAE

(f) MoVGAE

FIGURE 10. Visualization of node representations on Citeseer.

graph information, allowing the node representations to preserve node distinctiveness, local and global structural characteristics, consequently enhancing the performance of downstream graph tasks.

TABLE 3. Experimental results of link prediction.

Methods	Cora		Citeseer		Pubmed	
	AUC	AP	AUC	AP	AUC	AP
VGAE	89.60	90.87	90.46	91.90	92.60	93.21
VGNAE	89.81	90.79	90.10	91.03	93.89	94.53
GC-VGAE	90.11	90.29	91.77	91.27	93.96	94.01
MSVGAE	91.93	91.97	93.66	93.79	92.05	92.21
OSA-VGAE	92.26	92.74	94.08	94.38	93.78	94.32
MoVGAE	**93.08**	**93.51**	**94.18**	**94.73**	**96.43**	**96.65**

G. VISUALIZATION

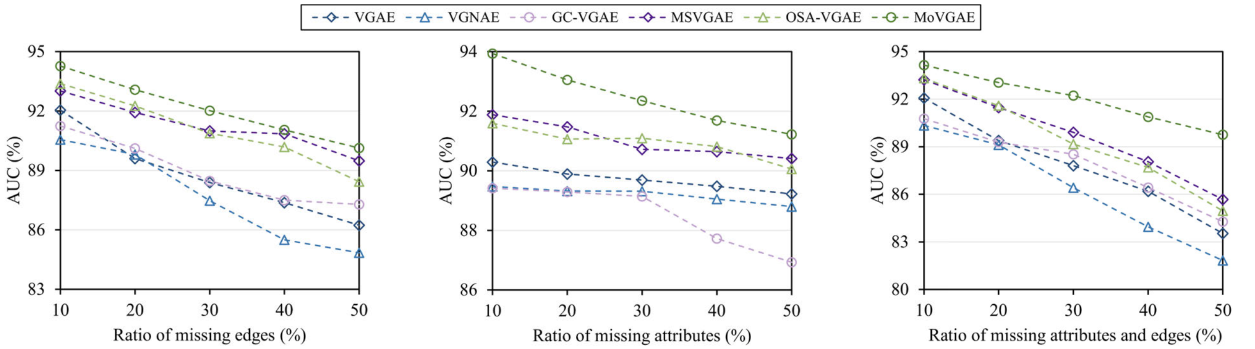
The node representation contains the attributes and topological structure information of the graph. Visualizing the node representation can intuitively reflect the model's capacity to represent the original graph information. Firstly, t-SNE [42] is used to reduce the node representations of baselines and MoVGAE on the Citeseer to 2-dimension. Then, the data points on the 2-dimensional space are labeled with six different colors based on the node labels. The results of the visualization experiment are shown in Figure 10.

The visualization results are usually characterized by nodes of the same color being close to each other, while nodes of different colors are separated from each other. Except for VGAE and VGNAE, the other models can extract feature information from the graph data to form partial community structures. However, the visualized results of different models show significant differences in intra-class similarity and inter-class boundaries. The node distribution of GC-VGAE and MSVGAE is sparse, with low intra-class similarity. Conversely, OSA-VGAE and MoVGAE have nodes distributed more closely within the same class. In addition, the visualization intuitively reflects the model's capacity to maintain similar features of nodes within the same community, and proves that introducing high-order information can enhance the representation capability of the VGAE.

H. ROBUSTNESS ANALYSIS

To ensure the timeliness of the methods, we selected Cora dataset and link prediction task for testing in the robustness experiments. Most graph representation learning methods require high-quality training data. When the training set contains complex noise, missing data, and other issues, the effectiveness of the model is often not guaranteed, which is known as the problem of robustness. We evaluate the robustness of baselines and MoVGAE by reducing the number of edges, the number of attributes, and simultaneously reducing edges and attributes. In link prediction task, we use the removed edges to construct the test set. In the experiment of randomly attacking edges, we reduce 10% of edges from the train set each time. The results of randomly attacking edges are shown in Figure 11(a). In the experiment of randomly attacking attributes, we reduce 10% of attributes from the train set each time. The results of randomly attacking attributes are shown in Figure 11(b). In the experiment of randomly attacking edges and attributes, we reduce 10% of edges and attributes from the training set each time. The results of randomly attacking edges and attributes are shown in Figure 11(c).

From the results, it is evident that the performance of each model shows a decreasing trend as the amount of graph information in the train set decreases. However, the



(a) AUC of random attack edges

(b) AUC of random attack attributes

(c) AUC of random attack edges and attributes

FIGURE 11. Experimental results of robustness analysis.

performance of MoVGAE consistently outperforms that of other baselines. In the case of attacking edges, high-order neighbors can serve as a supplement to first-order neighbors, enhancing the topological relationships between nodes. In the case of attacking attributes, introducing high-order neighborhoods allows nodes to aggregate more potential features, thereby mitigating the impact of missing attributes. In the extreme case of attacking edges and attributes, the proposed method still maintains better results, demonstrating strong robustness. The experimental results indicate that introducing high-order information can retain more comprehensive original graph information and enhance the model's representation capability in cases where attributes and topology are insufficient.

I. HYPERPARAMETERS STUDY

In MoVGAE, the order of the MoGCN encoder is an important hyperparameter. During the experiments, the other hyperparameters in MoVGAE, such as dimension, number of iterations, and initial learning rate, are the most commonly used hyperparameters. To ensure the timeliness of the methods and save space, we selected Cora dataset and node clustering task for testing. Figure 12 shows the NMI values, with different number of orders.

From the Figure 12, we notice that the clustering results improve as the order increased from 2-order to 3-order. This mean that the increased order facilitates the representation of more information about high-order neighborhoods, thus promoting good clustering. However, performance declines as the number of orders increases further. As the order increases, each node has more indirect neighbors, which weakens the topology differences between nodes and reduces the importance of critical nodes during feature aggregation. Therefore, it is necessary to set the appropriate order during the model training to achieve the better performance. From the overall observation and analysis, the performance of MoVGAE at different orders consistently outperforms VGAE and its variants. This further validates that the introduction of high-order information enhances the ability for representation learning.

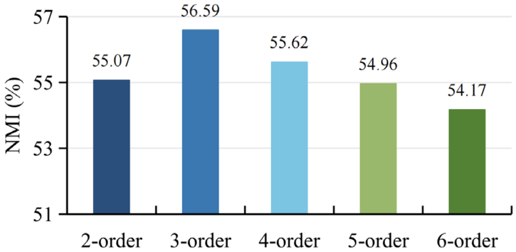


FIGURE 12. Experimental results of hyperparameters study.

J. TRAINING TIME

To ensure the timeliness of the methods, we selected Cora dataset for testing in the training time experiments. Here, we report the preparation time (data preprocessing, parameters loading, model importing) and the training time for 100 epochs (forward propagation, loss function calculation, backward propagation), measured in seconds wall-clock time. Figure 13 summarizes the results.

When comparing the preparation time of different models, it is noted that VGAE and its variants only require simple data preprocessing, so the results are relatively close. Compared to baselines, MoVGAE requires a longer preparation time because it needs to compute the k^{th} power of the adjacency matrix and perform symmetric normalization during data preprocessing. When comparing the training time of

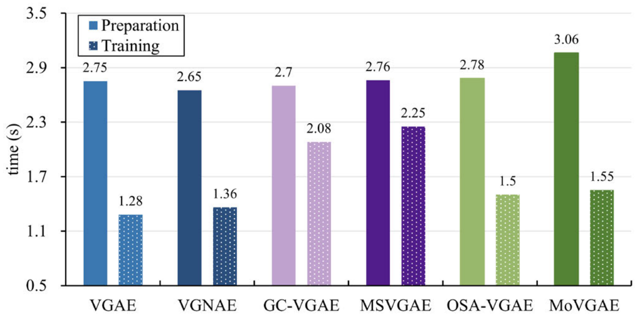


FIGURE 13. Experimental results of training time.

different models, VGNAE, OSA-VGAE, and MoVGAE, which directly modify the encoder, yield results close to the original VGAE. GC-VGAE, which adds attribute reconstruction loss, and MSVAGE, which calculates multi-scale information, require significantly longer training times compared to other models. From the overall observation and analysis, MoVGAE not only achieves high prediction accuracy but also demonstrates better time efficiency.

K. ANALYSIS OF HIGH-ORDER METHODS

In graph data, edges are more likely to form between nodes of the same type. For example, the intra-class edge rate of Cora is 81%, and the intra-class edge rate of Citeseer is 73.6%. Benefiting from this homogeneity, most methods can achieve good performance by aggregating the first-order neighborhood with GCN. However, these methods ignore the transmission of information from distant nodes and lack attention to high-order neighbors. In works such as LINE and SDNE, implicit high-order information have been proven to improve the experimental performance of downstream tasks. Inspired by the above works, we propose a MoGCN that can capture high-order information and integrate with graph representation learning methods. From the results, it was observed that by setting the appropriate order, MoVGAE using GCN and MoGCN encoders can effectively capture the information of first-order and high-order neighbors, thereby enhancing representation capability.

VI. CONCLUSION

In this paper, we propose a novel method for graph representation learning called MoVGAE, which utilizes GCN and MoGCN encoders to simultaneously capture first-order and high-order information from graph data. The results demonstrate that MoVGAE achieves state-of-the-art performance compared to other baselines in various tasks. The introduction of high-order information enhances the model's representation capacity, improving its generalization and versatility. In the case of insufficient attributes and topology information, MoVGAE still maintains better performance and demonstrates strong robustness. In future work, we will consider introducing feature fusion using attention mechanisms to allocate different weights to neighbors. Additionally, we hope that future research will explore ideas to address realistic issues, such as identifying abnormal accounts on social networks and rapidly uncovering the communities within criminal networks.

REFERENCES

[1] W. R. Kerr and M. Mandorff, "Social networks, ethnicity, and entrepreneurship," *J. Hum. Resour.*, vol. 58, no. 1, pp. 183–220, Jan. 2023, doi: 10.3368/jhr.58.3.0719-10306r2.

[2] T. Diviák, "Structural resilience and recovery of a criminal network after disruption: A simulation study," *J. Experim. Criminol.*, pp. 1–29, Mar. 2023, doi: 10.1007/s11292-023-09563-z.

[3] X. An, X. Sun, S. Xu, L. Hao, and J. Li, "Important citations identification by exploiting generative model into discriminative model," *J. Inf. Sci.*, vol. 49, no. 1, pp. 107–121, Feb. 2023, doi: 10.1177/0165551521991034.

[4] X. Lin, C. Zhou, J. Wu, H. Yang, H. Wang, Y. Cao, and B. Wang, "Exploratory adversarial attacks on graph neural networks for semi-supervised node classification," *Pattern Recognit.*, vol. 133, Jan. 2023, Art. no. 109042, doi: 10.1016/j.patcog.2022.109042.

[5] E. Nasiri, K. Berahmand, and Y. Li, "Robust graph regularization nonnegative matrix factorization for link prediction in attributed networks," *Multimedia Tools Appl.*, vol. 82, no. 3, pp. 3745–3768, Jan. 2023, doi: 10.1007/s11042-022-12943-8.

[6] M. Xu, "Understanding graph embedding methods and their applications," *SIAM Rev.*, vol. 63, no. 4, pp. 825–853, Jan. 2021, doi: 10.1137/20m1386062.

[7] R. Zou, Y. Liu, C. Li, Y. Zhang, and Y. Hu, "Graph representation learning: A review," *J. Beijing Normal Univ.*, vol. 59, no. 5, pp. 716–724, Oct. 2023, doi: 10.12202/j.0476-0301.2023172.

[8] S. Polizzi, T. Marzi, T. Matteuzzi, G. Castellani, and A. Bazzani, "Random walk approximation for stochastic processes on graphs," *Entropy*, vol. 25, no. 3, p. 394, Feb. 2023, doi: 10.3390/e25030394.

[9] M. Ouyang, Y. Zhang, X. Xia, and X. Xu, "GraRep++: Flexible learning graph representations with weighted global structural information," *IEEE Access*, vol. 11, pp. 98217–98229, 2023, doi: 10.1109/ACCESS.2023.3313411.

[10] J. Yang, S. Yang, Y. Fu, X. Li, and T. Huang, "Non-negative graph embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, Jun. 2008, pp. 1–8, doi: 10.1109/CVPR.2008.4587665.

[11] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 701–710.

[12] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 855–864.

[13] L. Wu, P. Cui, J. Pei, L. Zhao, and X. Guo, "Graph neural networks: Foundation, frontiers and applications," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Washington, DC USA, Aug. 2022, pp. 4840–4841, doi: 10.1145/3534678.3542609.

[14] A. Singh and T. Ogunfunmi, "An overview of variational autoencoders for source separation, finance, and bio-signal applications," *Entropy*, vol. 24, no. 1, p. 55, Dec. 2021, doi: 10.3390/e24010055.

[15] J. Li, H. Shao, D. Sun, R. Wang, Y. Yan, J. Li, S. Liu, H. Tong, and T. Abdelzaher, "Unsupervised belief representation learning with information-theoretic variational graph auto-encoders," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Madrid Spain, Jul. 2022, pp. 1728–1738, doi: 10.1145/3477495.3532072.

[16] I. Ullah, M. Manzo, M. Shah, and M. G. Madden, "Graph convolutional networks: Analysis, improvements and results," *Int. J. Speech Technol.*, vol. 52, no. 8, pp. 9033–9044, Jun. 2022, doi: 10.1007/s10489-021-02973-4.

[17] R. Wei, C. Garcia, A. El-Sayed, V. Peterson, and A. Mahmood, "Variations in variational autoencoders—A comparative evaluation," *IEEE Access*, vol. 8, pp. 153651–153670, 2020, doi: 10.1109/ACCESS.2020.3018151.

[18] G. Salha, R. Hennequin, and M. Vazirgiannis, "Simple and effective graph autoencoders with one-hop linear models," in *Machine Learning and Knowledge Discovery in Databases*, vol. 12457. Cham, Switzerland: Springer, 2021, pp. 319–334, doi: 10.1007/978-3-030-67658-2_19.

[19] S. J. Ahn and M. Kim, "Variational graph normalized autoencoders," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manag.* New York, NY, USA: Association for Computing Machinery, Oct. 2021, pp. 2827–2831, doi: 10.1145/3459637.3482215.

[20] L. Yuan and Z. Liu, "Graph representation learning by autoencoder with one-shot aggregation," *J. Comput. Appl.*, vol. 43, no. 1, pp. 8–14, Jan. 2023, doi: 10.11772/j.issn.1001-9081.2021101860.

[21] G. Wu, S. Lin, X. Shao, P. Zhang, and J. Qiao, "QPGCN: Graph convolutional network with a quadratic polynomial filter for overcoming over-smoothing," *Int. J. Speech Technol.*, vol. 53, no. 6, pp. 7216–7231, Mar. 2023, doi: 10.1007/s10489-022-03836-2.

[22] Z. Guo, F. Wang, K. Yao, J. Liang, and Z. Wang, "Multi-scale variational graph autoencoder for link prediction," in *Proc. 14th ACM Int. Conf. Web Search Data Mining*. New York, NY, USA: Association for Computing Machinery, Feb. 2022, pp. 334–342, doi: 10.1145/3488560.3498531.

[23] L. Guo and Q. Dai, "Graph clustering via variational graph embedding," *Pattern Recognit.*, vol. 122, Feb. 2022, Art. no. 108334, doi: 10.1016/j.patcog.2021.108334.

[24] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 1225–1234, doi: 10.1145/2939672.2939753.

[25] J. Firth, "A synopsis of linguistic theory, 1930–1955," in *Studies in Linguistic Analysis*, 1957, pp. 1–32.

[26] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropological Res.*, vol. 33, no. 4, pp. 452–473, Dec. 1977, doi: 10.1086/jar.33.4.3629752.

[27] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, Sep. 1999, doi: 10.1145/324133.324140.

[28] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, Florence Italy, May 2015, pp. 1067–1077, doi: 10.1145/2736277.2741093.

[29] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe, "Weisfeiler and Leman go neural: higher-order graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 4602–4609, doi: 10.1609/aaai.v33i01.33014602.

[30] L. He, L. Bai, X. Yang, H. Du, and J. Liang, "High-order graph attention network," *Inf. Sci.*, vol. 630, pp. 222–234, Jun. 2023, doi: 10.1016/j.ins.2023.02.054.

[31] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013, doi: 10.1109/TSP.2013.2238935.

[32] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. NIPS*, 2016, pp. 3844–3852.

[33] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

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

[35] B. T. Hung, "Link prediction in paper citation network based on deep graph convolutional neural network," in *Computer Networks, Big Data and IoT*, vol. 117. Singapore: Springer, 2022, pp. 897–907, doi: 10.1007/978-981-19-0898-9_67.

[36] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, "Network representation learning with rich text information," in *Proc. IJCAI*, 2015, pp. 2111–2117.

[37] L. Tang and H. Liu, "Relational learning via latent social dimensions," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jun. 2009, pp. 817–826.

[38] Z. Xie, X. Wang, H. Zhang, I. Sato, and M. Sugiyama, "Adaptive inertia: Disentangling the effects of adaptive learning rate and momentum," 2020, *arXiv:2006.15815*.

[39] H. Ma, F. Ding, and Y. Wang, "A novel multi-innovation gradient support vector machine regression method," *ISA Trans.*, vol. 130, pp. 343–359, Nov. 2022, doi: 10.1016/j.isatra.2022.03.006.

[40] C. Wang, S. Pan, C. P. Yu, R. Hu, G. Long, and C. Zhang, "Deep neighbor-aware embedding for node clustering in attributed graphs," *Pattern Recognit.*, vol. 122, Feb. 2022, Art. no. 108230, doi: 10.1016/j.patcog.2021.108230.

[41] H. Li and J. Wang, "Collaborative annealing power k-means++ clustering," *Knowl.-Based Syst.*, vol. 255, Nov. 2022, Art. no. 109593, doi: 10.1016/j.knosys.2022.109593.

[42] T. T. Cai and R. Ma, "Theoretical foundations of t-SNE for visualizing high-dimensional clustered data," *J. Mach. Learn. Res.*, vol. 23, pp. 1–54, Jan. 2022.

LINING YUAN received the B.E. and M.S. degrees in public security technology from the People's Public Security University of China, Beijing, China, in 2019 and 2022, respectively.

He is currently a Lecturer with the School of Public Security Big Data Modern Industry, Guangxi Police College, Nanning, China. His research interests include machine learning, graph neural networks, and graph representation learning.

PING JIANG received the B.E. degree in computer science and technology from Guangxi Normal University, Nanning, China, in 2005, and the M.S. degree in computer technology from Guangxi University, Nanning, in 2012.

She is currently a Professor with the School of Public Security Big Data Modern Industry, Guangxi Police College, Nanning. Her research interests include data science, big data technology, and natural language processing.

ZHU WEN received the B.E. degree in computer science and technology from Southwest Normal University, Chongqing, China, in 2004, and the M.S. degree in computer technology from Chongqing University, Chongqing, in 2010.

She is currently an Associate Professor with the School of Public Security Big Data Modern Industry, Guangxi Police College, Nanning, China. Her research interests include machine learning, data science, and big data technology.

JIONGHUI LI received the B.E. degree in information security from Guangxi Police College, Nanning, China, in 2022.

He is currently a Senior Engineer with Suzhou Keda Technology Corporation Ltd., China. His research interests include cyber security, information security, and big data technology.