Journal of Network Intelligence
Taiwan Ubiquitous Information

Volume **8**, Number **2**, May **2023**

Graph Neural Network for Metapath Aggregation based on Neighborhood Similarity

Dong-Yang Wang
School of Computer and Information Engineering
Xiamen University of Technology
Xiamen, 361024, China
2022031494@s.xmut.edu.cn

Kai-Biao Lin*
School of Computer and Information Engineering
Xiamen University of Technology
Xiamen, 361024, China
Engineering Research Center of Big Data Application in Private Health Medicine,
Fujian Provincial University, Putian 351100, China
2010110706@t.xmut.edu.cn

Zhi-Rong Wu
Xiamen Jingpei Software engineering Co.,LTD
Xiamen, 361008, China
001wzr@163.com

Jiang-Tao Lu
School of Computer and Information Engineering
Xiamen University of Technology
Xiamen, 361024, China
2022031417@s.xmut.edu.cn

*Corresponding author: Kai-Biao Lin

Received October 6, 2022, revised December 5, 2022, accepted January 18, 2023.

Abstract. *Most graphs or networks in the real world are inherently heterogeneous and involve different types of nodes and relationships. Typically, existing models define multiple metapaths in the heterogeneous graph to capture complex semantic relationships and guide neighbor selection. However, such models ignore node characteristics and do not consider information about intermediate nodes. Alternatively, in such models, the target node is assimilated by neighboring nodes or only information about a metapath is considered. Thus, in this paper, we proposed the neighborhood similarity-based metapath aggregation graph neural network (MANS-GNN) model. The proposed MANS-GNN first uses node mapping for the input of node features. Second, to avoid embedding and excessive assimilation between different node types, a similarity-aware neighborhood selector based on reinforcement learning is employed to select the most similar neighborhood for the target node. Then, a local aggregation module is employed to merge intermediate semantic nodes. Finally, global aggregation is employed to merge information from individual metapaths. The proposed MANS-GNN model was evaluated experimentally in node classification and node clustering tasks on two heterogeneous graph datasets. Experimental results demonstrated that the proposed MANS-GNN model obtains more accurate predictions than the baseline model.*
Keywords: Metapaths, Heterogeneous graph, Neighbor selection,Reinforcement learning

1. **Introduction.** Many real-world problems can be solved by structuring data into graphs, where objects and the relationships between objects are represented by nodes and edges, respectively [1]. In the real world, large amounts of data can be modeled in the form of heterogeneous graphs [2], e.g., smart cities [3], stock evaluation function [4], recommender systems [5], road traffic monitor systems [6] and natural language processing [7]. Heterogeneous graphs comprise different types of nodes and edges associated with attributes. For example, a network graph in a movie scenario contains five of node types, i.e., user(U), movie(M), director(D), group(G), and movie type(T). In this case, the meaning of the edge is intuitive. For example, the edge type U-G indicates that the user has joined the group, and the edge type U-M indicates that the user has seen the movie and given a score. Heterogeneous graphs involve different types of nodes and edges compared to homogeneous graphs; thus, heterogeneous graphs are rich in semantic information and heterogeneous structural information. The concept of homogeneous graph embedding is applied in heterogeneous graph embedding, which represents the nodes in the graph as low-dimensional vectors and preserves the heterogeneous structural and semantic information in the graph, thereby making heterogeneous graph embedding better for various downstream tasks, e.g., node classification and node clustering [8].

Most existing methods for embedding in heterogeneous graphs are based on the idea of metapaths, each of which is an alternating sequence of nonrepetitive nodes and edges [1]. For example, in a network of movie scenes comprising users, movies, and directors, user-movie-user (UMU) and user-movie-director-movie-user (UMDMU) are two different metapaths describing the relationship between users. Here, the UMU metapath associates two users watching the same movie, and the UMDMU metapath connects two users who watch movies by the same director. Early heterogeneous graph embedding models, e.g., Metapath2vec [9], used random wandering based on metapaths to obtain the heterogeneous neighbors of nodes, which were then input to the skip-gram [10] model to learn the representation of nodes. With the rapid development of deep learning technologies, increasing attention is being paid to extending deep learning techniques to graph structures. For example, graph neural networks (GNN) have emerged, e.g., the graph convolution network (GCN) [11] and the graph attention network (GAT) [12]. In addition, the GNN has been introduced in the heterogeneous graph model, where a specially designed neural layer is used to realize graph representation learning. Heterogeneous graph attention network (HAN) [14] uses node-level and semantic-level attention mechanisms to mine metapaths in heterogeneous graphs, learns the representations of nodes in different metapaths, and aggregates these representations with the learned weights.

Compared to traditional network embedding methods, existing metapath-based embedding methods perform better on downstream tasks, e.g., node classification and node clustering; however, several limitations must be considered [15]. (1) Existing metapath-based embedding methods may not use node features, resulting in poor performance on heterogeneous graphs with a large number of nodes. (2) Imprecise feature selection may not be able to extract the effective semantics of nodes and may easily lead to embedding and excessive assimilation between different types of nodes, which greatly reduces the accuracy of GNN feature learning representation. (3) Existing metapath-based embedding methods may only considers the start and end nodes of the meta path, and does not consider the intermediate nodes and relationships on the meta path, which results in information loss. (4) Existing metapath-based embedding methods may relies on only a single metapath to embed the heterogeneous graph, which loses the information of other meta paths, thereby resulting in poor performance.

To address these limitations, this paper proposes a metapath aggregation graph neural network (MANS-GNN) based on neighborhood similarity. The proposed MANS-GNN solves the above problems well by applying node mapping, a similarity-aware neighborhood selector based on reinforcement learning (RL), local aggregation, and global aggregation to obtain the final node embedding. The dimensionality of different node types is not equal; thus, the proposed MANS-GNN first applies node mapping to shadow heterogeneous node genus features into the same latent vector space. Then, the MANS-GNN performs local aggregation for each metapath through a layer of attention mechanisms. Prior to aggregation, to avoid over-assimilation of embeddings between different types of nodes, the proposed MANS-GNN also employs a similarity-aware neighborhood selector based on RL, which is used to select the most similar neighborhood of the target node under a certain relationship. After local aggregation, the MANS-GNN uses an attention mechanism [16] for global aggregation and merges the potential vectors obtained from multiple metapaths into the final node embedding. By integrating multiple metapaths, the model can learn comprehensive semantic information from the heterogeneous graph.

Our primary contributions are summarized as follows.

(1) The proposed MANS-GNN model considers node features, the information of all nodes on the meta-path and multiple meta-paths, and avoids the problem of over-assimilation of embeddings between different types of nodes.

(2) The RL-based similarity-aware neighborhood selector selects the most similar neighborhood of the target node under a certain relationship; thus, the proposed model avoids over-assimilation of embeddings between different types of nodes.

(3) Numerous node classification and node clustering experiments were conducted on the IMDB and DBLP datasets. The experimental results demonstrate that the node embeddings learned by the proposed MANS-GNN outperform existing models.

2. **Related Work.** In the following, we review related research, including GNNs, heterogeneous graph embeddings, and the combination of GNNs and RL.

2.1. **Graph neural networks.** Due to the rapid development of deep learning in recent years, GNNs have made great progress and attracted considerable attention in graph representation learning. The core concept on a GNN is to aggregate the feature information of neighboring nodes through neural networks and combine the independent information of the nodes with the corresponding structural information in the graph [16]. According to the differences in modeling of actual graph data, GNNs can be roughly divided into two categories, i.e., homogeneous GNNs and heterogeneous GNNs.

Homogeneous GNN models typically do not consider node data types and edge properties. Conventional methods include GCN [11], GAT [12], and Graph-SAGE [18]. GCN [11] defined the first successful graph convolution by applying convolution operations to graphs. Through the approximate analysis of the graph in the frequency domain, the local structure and node characteristics of the graph are modeled and learned. Graph-SAGE [18] is an inductive learning framework that uses node attribute information to generate unknown node feature representation on a large-scale graph. Here, the core idea is to generate the feature representation of the central node by learning a function that aggregates the neighbor nodes rather than learning the embedding of the node itself. In contrast, GAT [12] achieves adaptive matching of weights to different neighbors through a self-attentive mechanism to aggregate neighboring nodes, thereby improving the representational power of the model. Although these models have strong graph representation learning ability, their primary limitation is that they ignore the diversity of data types and relationships in real-world data [23].

In contrast, heterogeneous GNN models do consider node data types and edge properties. Conventional approaches include HNE [13], HAN [14], HGT [19], HetSANN [20], and MAGNN [15]. HNE [13] Use highly nonlinear multi-layer embedded functions to capture complex interactions between heterogeneous data in the network. HAN [14] employs hierarchical attention to describe node level and semantic level structures, and MAGNN [15] considers the intermediate node information of the metapath. HGT [19] involved parameters related to node and edge types to characterize the heterogeneous attention on each edge such that it can maintain a special representation for different types of nodes and edges. In addition, HetSANN [20] can encode the structural information of HIN directly without a metapath, thereby realizing more information representation. However, to the best of our knowledge, no previous study has investigated methods to select neighboring nodes in order to construct the most expressive, explanatory, and stable aggregations.

2.2. **Heterogeneous graph embedding.** Heterogeneous graphs comprise multiple types of nodes and edges, which makes it difficult to preserve structural information in node embeddings properly [21]. Heterogeneous graph embedding is essentially the projection of different types of nodes in a graph into the same low-dimensional vector space. Many studies have addressed this challenge. For example, Metapath2vec [9] implements a metapath-based random walker technique and utilizes the skip-gram model [10] to generate node embeddings. HERec [22] transforms a heterogeneous map into multiple homogeneous maps based on different meta paths, and it applies the DeepWalk model to learn the node embedding of the target type.

A GNN can learn the representation of graph structured data efficiently; thus, various GNN-based approaches have been extended to model heterogeneous graphs and encode the representation of attribute information [21]. For example, HAN [14] converts heterogeneous graphs into multiple homogeneous graphs based on metapaths, aggregates neighborhood information using GATs, and uses attention mechanisms to combine various metapaths.

However, the above methods either ignore node features, discard all intermediate nodes on the metapath, or utilize only a single metapath. Thus, performance improvements can be achieved by realizing more comprehensive use of the information embedded in the heterogeneous graph [15].

2.3. **Combining graph neural networks with reinforcement learning.** With the development of GNNs and RL methods, numerous studies have attempted to combine these technologies [23]. For example, Graph2Seq [24], which is an RL-based graph-to-sequence model, employs a deep alignment network to effectively integrate the answer information into the article, and a bidirectional GNN is used to process the directed channel graph. Many studies have used RL to optimize the learning of graph representations. For example, RL-HGNN [25] models the metapath design process as a Markov decision process and uses a policy network to design metapaths adaptively for each node to learn an efficient representation. In addition, DeepPath [26] employs an agent with a continuous state based on a knowledge graph embedding that extends the path by sampling the most promising relations and reasoning in the knowledge graph vector space. Policy-GNN [27] employs a meta-policy to determine the number of aggregations per node adaptively, and the meta-policy is trained using deep RL by exploiting the feedback from the model. GraphNAS [28] employs a search space that covers sampling functions, aggregation functions, and gating functions, and it employs RL to search the graph neural architecture. However, although the GraphNAS [28] and Policy-GNN [27] models are more concerned with the search of neural architectures, neither model considers heterogeneous neighborhoods in aggregation.

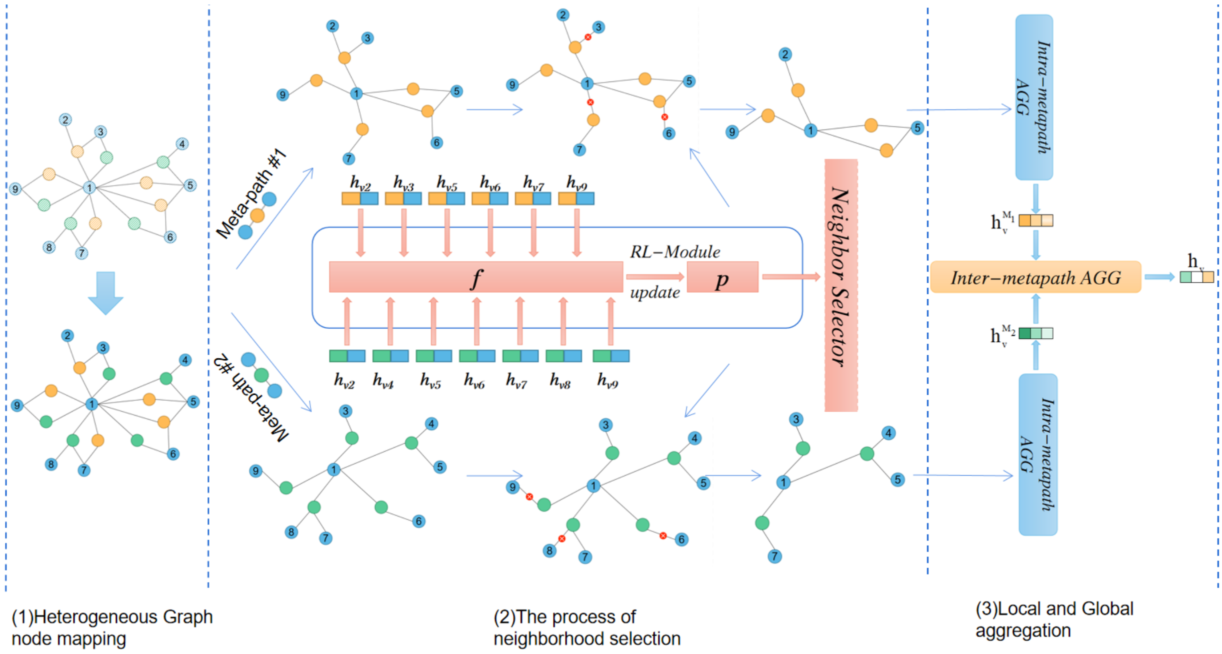


FIGURE 1. The overall architecture of MANS-GNN.

3. **Proposed Metapath Aggregation Graph Neural Network.** Here, we present the proposed MANS-GNN based on neighborhood similarity. Figure 1 illustrates the overall architecture of the MANS-GNN, which comprises the following components. (1) Heterogeneous node features are mapped into the same feature space. (2) An over-selection operation is performed on the similarity-aware neighborhood selector based on RL. (3) The final node embedding is generated through local aggregation, intra-metapath aggregation using the attention mechanism, and finally global aggregation and inter-metapath aggregation using the attention mechanism.

3.1. **Node mapping.** Nodes and edges in heterogeneous graphs have different types. Different types of node attributes have different dimensional feature vectors, and even if the nodes happen to be of the same dimension, they may belong to different feature spaces. To solve this problem, it is necessary to first project the heterogeneous node features into the same feature space.

Here, for a node of type , the mapping process can be expressed as follows:

$$h'_v = W_a * h_v \tag{1}$$

where h_v is the feature representation of node v before the mapping process, and h_v is the feature representation of node v after the mapping process. In addition, W_a is the parameter weight matrix of the nodes of type a. After this process, the features of the heterogeneous nodes in the graph are mapped into the same feature space with a certain number of dimensions.

3.2. **Neighborhood selector.** MANS-GNN learns semantic information in embedded nodes by encoding metapath examples using a relational rotation encoder to convert all node features of a metapath instance into a single vector. The relational rotation encoder, proposed by RotatE for knowledge graph embedding, is a meta-path instance encoder based on the rotation of relations in complex spaces [29]. The relational rotary encoder is expressed as follows:

$$h_{M(v,u)} = f(M(v,u)) = f(h'_v, h'_u, \{h'_g, \forall g \in \{t^{M(v,u)}\}\}) \tag{2}$$

where $M(v,u)$ is a metapath instance connecting the target node v to its metapath-based neighbor u. $t^{M(v,u)}$ denotes the intermediate node of $M(v,u)$.

After encoding the metapath instances into a vector representation, for a target node v, a metapath instance based on the target node v is considered a neighborhood of the target node v. First, we calculate the neighborhood similarity between nodes and neighbors based on the proposed neighborhood similarity metric. Here, a Multi-layer Perceptron (MLP) is used as a node predictor, and the node and neighbors prediction result scores are used for the similarity measurement. For a target node v under the relation r of edge (v,v'), the similarity measure between v and v' is defined as follows:

$$S(v,v') = ||\sigma(MLP(h_{vv'}))|| \tag{3}$$

The RL-based similarity-aware neighborhood selector performs adaptive filtering to select the best number of similar neighbors automatically, thereby avoiding the high cost of data annotation. Here, the sampling is used with an adaptive filtering threshold to select similar neighbors under each relation, and an RL algorithm is used to identify the optimal threshold during GNN training.

Specifically, in the training phase, for node v in the current batch under relation r, MANS-GNN first calculates a set of similarity metric scores using Equation (3). Then, the neighborhoods are arranged in descending of the similarity measurement score, retaining the part of the current batch with the highest similarity and discarding the rest of the neighborhoods.

To optimize the computational efficiency of the neighbor selection, the proposed model employs an RL framework to find the optimal threshold t_r. Given an initial threshold t_r, a_r is defined as a fixed small value for which the neighborhood selector chooses to increase or decrease t_r. The optimal t_r value is expected to find the most similar neighborhood of the target node under relation r. The average similarity score for cycle e under relation r is expressed as follows:

$$G(S_r)^{(e)} = \frac{\sum_{v \in V_{train}} S_r(v,v')^{(e)}}{|V_{train}|} \tag{4}$$

Then, the reward mechanism is designed based on the difference in mean similarity scores between two consecutive batches. The reward for period e is as defined as follows:

$$f(t_r,a_r)^{(e)} = \begin{cases} +1 & if \quad G(S_r)^{(e-1)} - G(S_r)^{(e)} \geq 0 \\ -1 & if \quad G(S_r)^{(e-1)} - G(S_r)^{(e)} \leq 0 \end{cases} \tag{5}$$

Note that the reward is positive when the average distance of the newly selected neighborhood of cycle e is less than the previous cycle; otherwise, the reward is negative. We designed greedy strategies that do not require search and use immediate rewards to update actions.

3.3. **Local and global aggregation.** After selecting the best neighborhood, we adopt local aggregation and use the attention mechanism to weighted sum the metapath instances $M(v,u)$ based on the target node v, as shown in Equation (6).

$$\begin{aligned} \beta_{vu}^{M} &= \frac{exp(LeakyReLU(a_M^T * [h'_v || h_{M(v,u)}]))}{\sum_{k \in N_v^M} exp(LeakyReLU(a_M^T * [h'_v || h_{M(v,k)}]))} \\ h_{vu}^{M} &= \sigma(\textstyle\sum_{u \in N_v^M} \beta_{vu}^{M} * h_{M(v,u)}) \end{aligned} \tag{6}$$

Here,a_M^T is a parametric attention vector for metapath M, $||$ denotes the vector connectivity operator, N_v^M is the set of neighbors of node v based on the metapath M, and β_{vu}^{M} is the normalized importance weight learned for each metapath instance. Finally, the output is passed through the activation function $\sigma(\bullet)$.

The learning process is stabilized using a multiheaded attention mechanism, thereby reducing the high variance associated with heterogeneous graphs. The proposed model applies independent attention mechanisms and splices their outputs as follows:

$$h_v^M = \|_{t=1}^T \sigma(\textstyle\sum_{u \in N_v^M} [\beta_{vu}^M]_t * h_{M(v,u)}) \tag{7}$$

where $[\beta_{vu}^M]_t$ is the normalized importance of the metapath instance $M(v,u)$ on the t-th attention head.

After aggregating the information of the nodes within each metapath at the local aggregation layer, the semantic information of all metapaths is combined using the global aggregation layer. Note that different metapaths have different importance in the heterogeneous graph; thus, the attention mechanism is used to assign different weights to different metapaths for aggregation.

The feature vector of node v under a particular metapath is fused using the attention mechanism as follows.

$$\begin{aligned} I_{M_i} &= g_a^T * \tfrac{1}{|V_a|} \textstyle\sum_{v \in V_a} tanh(Q_a * h_v^{M_i} + l_a) \\ \alpha_{M_i} &= \tfrac{exp(I_{M_i})}{\sum_{M \in P_a} exp(I_M)} \\ h_v^{P_a} &= \textstyle\sum_{M \in P_a} \alpha_M * h_v^M \end{aligned} \tag{8}$$

Here, $|V_a|$ is the number of potential vector sets of node type a, Q_a and I_a are learnable parameters, g_a^T is a parameterized attention vector of node type a, and α_{M_i} is the relative importance of metapath M_i to the node type a.

Finally, linear transformation is employed to embed and project the nodes into the vector space with the required output dimension.

$$h_v = \sigma(W_k * h_v^{P_a}) \tag{9}$$

Here, $\sigma(\bullet)$ is the activation function, and W_k is the weight matrix. This projec can be considered a linear classifier for node classification.

3.4. **Loss function.** After local and global aggregation, the final node representation that can be used in different downstream tasks is obtained. Under the guidance of a few labeled nodes, the model weight is optimized by minimizing the cross entropy through backpropagation and gradient descent in order to learn meaningful node embeddings in the heterogeneous graph. The cross-entropy loss formula is given as follows:

$$L = -\sum_{v \in V_L} \sum_{g=1}^{G} O_v[g] * \log h_v[g] \tag{10}$$

where V_L is the set of nodes with labels, G is the number of classes, O_v is the single-hot label vector of node v, and h_v is the predicted probability vector of node v.

Table 1. Experimental datasets

Dataset	Number of node	Number of edge	Metapath
IMDB	movie(M):4278 director(D):2081 actor(A):5257	M-D:4278 M-A:12828	MDM MAM
DBLP	author(A):4057 paper(P):14328 term(T):7723	A-P:19645 P-T:85810 P-V:14328	APA APTPA APVPA

4. **Experiments.** The effectiveness of the proposed MANS-GNN model for heterogeneous graph embedding was evaluated experimentally.

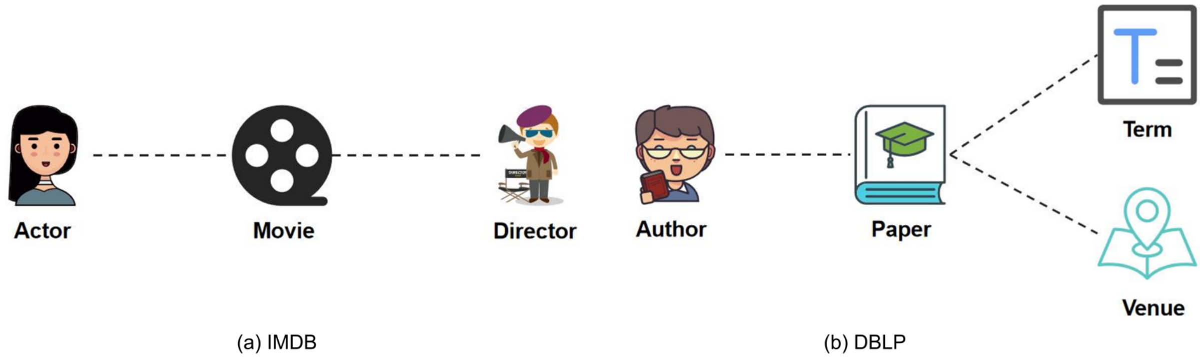


FIGURE 2. Network patterns for two heterogeneous graph datasets: (a) IMDB and (b) DBLP.

4.1. **Datasets.** The IMDB and DBLP heterogeneous graph datasets were used to perform node classification and node clustering experiments to evaluate the performance of the proposed MANS-GNN model, which was compared to existing graph embedding models. Table 1 shows the statistical information of these two datasets, and Figure 2 shows the network patterns of these two datasets. IMDB is an online database of movies and TV shows, including information about actors, production teams, and plot summaries. We used a subset of IMDB collected from the web that has been preprocessed with data and contains 4278 films, 2081 directors, and 5257 actors. In the experiments, the film nodes were divided into training, validation, and test sets of 400 (9.35%), 400 (9.35%), and 3478 (81.30%) nodes, respectively. DBLP is a computer science bibliography website. In these experiments, we used a subset of the DBLP database containing 4057 authors, 14328 papers, 7723 terms, and 20 publication sites In the experiments, the author nodes were divided into training, validation, and test sets with 400 (9.86%), 400 (9.86%), and 3257 (80.28%) nodes, respectively.

4.2. **Baseline.** The proposed MANS-GNN model was compared to different types of graph embedding models, including the traditional heterogeneous graph embedding model, the homogeneous GNN model, and the heterogeneous GNN model. The baseline models considered in our evaluations are summarized as follows.

Metapath2vec is the traditional heterogeneous graph embedding model. This model is based on a random walker of metapaths, and the skip-gram model is then used to generate node embeddings. Note that we selected the most efficient metapath by considering test performance.

The GCN is a homogeneous GNN model. It is a semi-supervised graph convolutional network. We tested GCN on metapath-based homogeneous graphs, and we report the best performance.

The GAT is a homogeneous GNN model. It is also a semi-supervised graph convolutional network. We tested GAT on metapath-based homogeneous graphs, and we report the best performance.

The HAN is a heterogeneous GNN model. Learn metapath specific node embeddings from different metapath-based isomorphic graphs. This model employs a two-layer attention mechanism to aggregate them into vector embeddings representing each node in the network.

4.3. **Parameter settings.** For the proposed MANS-GNN models, we set the dropout rate to 0.005, the training, validation, and test sets were split at the same proportions, and the ADAM optimizer was employed, where the learning rate was set to 0.001, and the weight decay (i.e., the L2 penalty) was set to 0.001. We trained these models for 100 cycles and stopped early with 20 patience. For the GAT, HAN, and MANS-GNN models, the number of attentional heads was set to eight. For the HAN and MANS-GNN models, the dimension of the attention vector in the metapath inter-aggregation was set to 128. In addition, the RL action step was set to 0.02 in the proposed MANS-GNN model.For the traditional heterogeneous graph embedding model Metapath2vec, we set the window size to five, the step size to 100, the number of steps per node to 40, and the number of negative samples to five.To facilitate a fair comparison, the embedding dimension of all models was set to 64.

TABLE 2. Experimental results (%) of node classification task on IMDB and DBLP datasets.

Dataset	Metric	Train %	Metapath2vec	GCN	GAT	HAN	MANS-GNN
IMDB	Macro-F1	20%	48.14	49.93	50.73	53.69	**54.34**
		40%	49.78	50.41	51.51	53.70	**56.14**
		60%	50.58	51.63	52.46	54.01	**56.90**
		80%	50.14	51.81	52.33	54.12	**57.27**
	Micro-F1	20%	49.15	49.78	50.64	55.21	**54.50**
		40%	50.99	50.71	51.67	55.17	**56.41**
		60%	51.81	51.29	52.23	55.37	**57.22**
		80%	51.53	51.61	52.77	55.53	**57.66**
DBLP	Macro-F1	20%	85.47	85.33	88.35	90.54	**93.10**
		40%	86.78	86.15	89.56	91.34	**93.35**
		60%	88.10	87.26	89.96	92.15	**93.70**
		80%	88.69	88.56	90.55	92.76	**93.56**
	Micro-F1	20%	86.59	86.03	89.22	91.33	**93.60**
		40%	87.15	86.54	90.15	91.89	**93.81**
		60%	88.23	87.69	90.69	92.56	**94.14**
		80%	89.56	88.57	91.29	93.15	**94.00**

4.4. **Node classification.** We first performed a node classification task to evaluate the effectiveness of the proposed model. We feed the embeddings of labeled nodes generated by each model to a linear support vector machine (SVM) classifier with varying training proportions. Similarly, the training/test split of the linear support vector machine is the same as that of the embedding model. The evaluation metrics used in these experiments were the Macro-F1 and Micro-F1. The node classification results are shown in Table 2.

The results demonstrate that the proposed MANS-GNN outperformed the baseline models under different training scales. We found that the heterogeneous GNN model obtained better results, which indicates that the GNN structure can make better use of heterogeneous node features and improve the embedding performance. The proposed MANS-GNN outperformed the best baseline (i.e., HAN), which indicates that metapath instances contain richer information than the metapath-based neighbor instances. The performance of the proposed MANS-GNN improved with increasing proportion at different training set proportion. On the DBLP dataset, the proposed MANS-GNN model outperformed the strongest baseline by 1–2%.

Table 3. Experimental results (%) on IMDB and DBLP datasets for node clustering task.

Dataset	Metric	Metapath2vec	GCN	GAT	HAN	MANS-GNN
IMDB	NMI	3.95	5.54	5.11	8.67	**12.23**
	ARI	4.33	5.66	5.53	7.65	**11.75**
DBLP	NMI	72.15	72.15	72.33	76.79	**80.78**
	ARI	76.54	73.46	72.59	82.37	**85.89**

4.5. **Node clustering.** A node clustering experiment was also conducted to evaluate the effectiveness of the proposed MANS-GNN model. Here, the embeddings of the tagged nodes generated by each learning model were input to the k-means clustering algorithm. The number of clusters in k-means was equal to the number of classes in each dataset. In addition, the normalized mutual information (NMI) and adjusted Rand index (ARI) evaluation metrics were used to evaluate the clustering results. In this experiment, k-means clustering was repeated 10 times for each run of the embedding model, and each embedding model was tested 10 times. The node clustering results are shown in Table 3.

The results demonstrate that the proposed MANS-GNN model outperformed all of the compared baseline models in the node clustering task. The experimental results show that the heterogeneous graph models MANSGNN and HAN outperform the homogeneous graph complex models GCN and GAT on both datasets. MANS-GNN outperforms HAN because MANS-GNN uses an RL-based similarity-aware neighbourhood selector to choose the most similar neighbours of the target nodes under certain relationships, which avoids over-assimilation of different types of node embeddings. It follows that the heterogeneous model has an advantage over the homogeneous model in terms of node clustering. The use of an RL-based similarity-aware neighbourhood selector can also improve model performance.

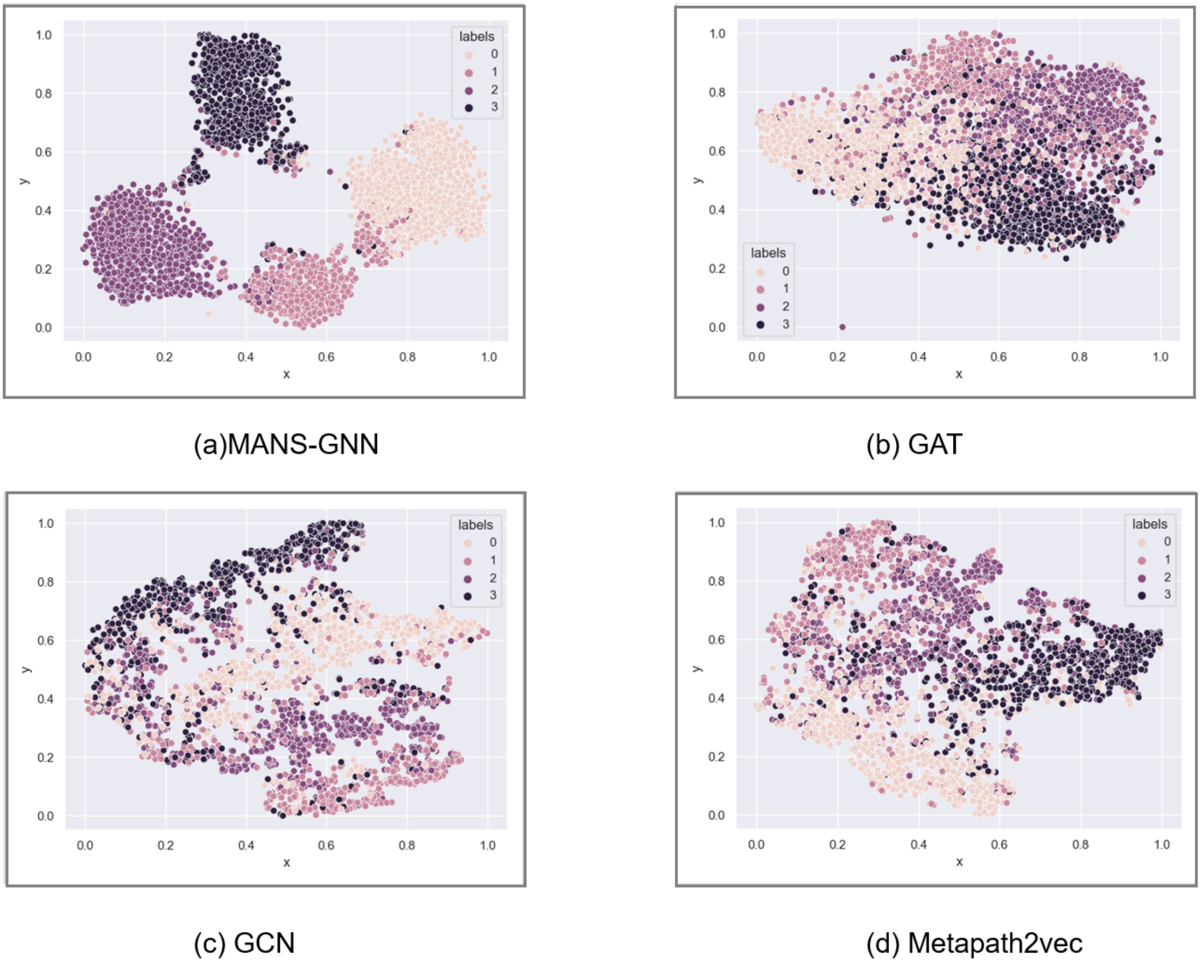
Table 4. Quantitative results of ablation studies (%)

Variant	IMDB				DBLP			
	Macro-F1	Micro-F1	NMI	ARI	Macro-F1	Micro-F1	NMI	ARI
$MANS-GNN_{nb}$	54.08	54.35	9.15	7.19	91.33	91.95	75.74	81.81
$MANS-GNN_{avg}$	55.09	55.03	12.28	10.53	92.90	93.45	79.03	84.43
$MANS-GNN_{linear}$	55.98	55.06	11.62	9.47	93.39	93.82	77.35	82.09
$MANS-GNN_{norl}$	55.88	55.75	11.43	9.54	92.85	93.38	79.62	84.75
$MANS-GNN_{rot}$	**56.16**	**56.45**	**12.23**	**11.75**	**93.43**	**93.89**	**80.78**	**85.89**

4.6. **Ablation study.** In order to verify the validity of each component of the proposed model, additional experiments were conducted using different variants of the MANS-GNN model. The ablation study results are shown in Table 4. Here, $MANS-GNN_{rot}$ represents the version of proposed model used for comparison with the baselines in Tables 2 and 3. $MANS-GNN_{nb}$ only considers the metapath-based neighbors, $MANS-GNN_{avg}$ used the mean metapath instance encoder, and $MANS-GNN_{linear}$ used the linear element path instance encoder. Finally, $MANS-GNN_{norl}$ is the model for removing neighborhood selectors. Note that all other settings were same for these MANS-GNN variants.

A comparison of the $MANS-GNN_{nb}$ model to the $MANS-GNN_{avg}$, $MANS-GNN_{linear}$, and $MANS-GNN_{rot}$ model revealed that aggregating metapath instances rather than metapath-based neighbors improved performance, which validates the efficacy of the intra-metapath aggregation process. The results obtained by the $MANS-GNN_{avg}$,

$MANS{-}GNN_{linear}$, and $MANS{-}GNN_{rot}$ models demonstrate that the relational rotary encoders help improve the performance of the proposed MANS-GNN model. We also found that the $MANS-GNN_{rot}$ model exhibited a clear advantage over the $MANS-GNN_{norl}$ model, which indicates that including the neighborhood selector helps improve performance.



(a)MANS-GNN (b) GAT

(c) GCN (d) Metapath2vec

Figure 3. Visualization of node clustering in DBLP dataset.

4.7. **Visualization.** To facilitate a more intuitive comparison, we visualized the learned embeddings in two-dimensional space. This method represents the distribution of nodes in a low-dimensional space. We learned the node embeddings for each model on the DBLP dataset, and the nodes embedded in the DBLP dataset were visualized using the t-SNE. For ease of observation, this paper take all the test sample nodes. The results are shown in Figure 3.

As shown in Figure 3, Metapath2vec is poorly expressed and has difficulty distinguishing the distribution of nodes. The baseline methods, e.g., the GAT and GCN models, can essentially distinguish between the different classes of nodes; however, there are no clear boundaries, and a significant number of different nodes are mixed together. In addition, we found that the proposed MANS-GNN model outperformed the GAT and GCN models.

Finally, the proposed MANS-GNN model clearly distinguished the different classes and exhibits clear boundaries. These results further demonstrate the effectiveness of adding the RL-based neighborhood selector for similarity perception.

5. **Conclusion.** In this paper, we have proposed the MANS-GNN model to address several limitations in existing metapath-based embedding methods, i.e., (1) node features are not utilized; (2) excessive assimilation of embedding occurs between different types of nodes; (3) intermediate nodes on the metapath are not considered; and (4) only single metapaths are considered. The proposed MANS-GNN model solves these problems by applying node mapping, an RL-based similarity-aware neighborhood selector, and local aggregation and global aggregation to generate node embeddings. The RL-based neighborhood selector can adaptively filter and automatically select the best number of similar neighbors to avoid high data annotation costs. In addition, a metapath instance encoder is employed to extract the deep-rooted structural and semantic information in the metapath instance.

The proposed MANS-GNN model was evaluated experimentally on node classification and node clustering tasks using the IMDB and DBLP datasets. We found that the proposed model outperformed the compared baseline models. In addition, ablation studies have demonstrated that the proposed MANS-GNN model's metapath instance encoder and RL-based similarity perception neighborhood selector can improve performance.

The reinforcement learning algorithms currently in use define relationships manually. In the future, we therefore aim to use multi-negotiation reinforcement learning algorithms to adaptively identify meaningful relationships on nodes, thus enabling automatic representation learning of heterogeneous data. In addition, to investigate how our model can be applied to other tasks such as health insurance fraud detection, recommender systems, and so on.

Acknowledgment. This research was funded by the Science Foundation of Fujian Province (No. 2021J011188), the Open Fund of Engineering Research Center of Big Data Application in Private Health Medicine, Fujian Provincial University (KF2020003).

REFERENCES

[1] A. Gm, P.-A. Li, and A. Sl, "Heterogeneous Graph Embedding by Aggregating Meta-path and Meta-structure through Attention Mechanism," *Neurocomputing*, vol. 468, pp. 276–285, 2022.

[2] Y. Sun, J. Han, "Mining heterogeneous information networks: a structural analysis approach," in *ACM SIGKDD Explorations Newsletter*, vol. 14, no. 2, pp. 20–28, 2013.

[3] K. Wang, C.-M. Chen, and M.-S. Hossain, "Transfer reinforcement learning-based road object detection in next generation IoT domain," *Computer Networks*, vol. 193, 108078, 2021.

[4] T. L. Luo, M.-E. Wu , and C.-M. Chen, "A framework of deep reinforcement learning for stock evaluation functions," *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 7, pp. 1–11, 2020.

[5] J. Zhang, X. Shi, S. Zhao, and I. King, "STAR-GCN: Stacked and Reconstructed Graph Convolutional Networks for Recommender Systems," *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 4264–4270, 2019.

[6] P. Xu, K. Wang, and M.-M. Hassan, , "Adversarial Robustness in Graph-Based Neural Architecture Search for Edge AI Transportation Systems," *IEEE Transactions on Intelligent Transportation Systems*, 2022. [Online]. Available: https://doi.org/10.1109/TITS.2022.3197713

[7] L. Yao, C. Mao, and Y. Luo, "Graph Convolutional Networks for Text Classification," in *Association for the Advancement of Artificial Intelligence*. pp. 7370–7377, 2019.

[8] S.-K. Wu, "Metapath-Based elational Selection Graph Neural Network," *Modern Computer*, no. 22 , pp. 30–37 ,2021.

[9] Y. Dong, N.-V. Chawla, and A. Swami, "metapath2vec: Scalable Representation Learning for Heterogeneous Networks," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 135–144, 2017.

[10] W. Ling, C. Dyer, and A. W. Black, et al, "Two/too simple adaptations of word2vec for syntax problems," *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 1299–1304, 2015.

[11] T.-N Kipf, M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *International Conference on Learning Representations (ICLR 2017)*, ICLR,2016.

[12] P. Velikovi, G. Cucurull, and A. Casanova, et al, "Graph Attention Networks," in *International Conference on Learning Representations (ICLR 2018)*, ICLR,2018.

[13] S. Chang, W. Han, and J. Tang, et al, "Heterogeneous network embedding via deep architectures," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining* , pp. 119–128, 2015.

[14] X. Wang, H. Ji, and C. Shi, et al, "Heterogeneous Graph Attention Network," in *the world wide web conference*, pp. 2022–2032, 2019.

[15] X. Wang, J. Zhang, and Z. Meng, et al, "MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding," in *proceedings of The Web Conference 2020*, pp. 2331–2341, 2021.

[16] X. Wang, H. Ji, and C. Shi, et al, "Heterogeneous Graph Attention Network," in *the world wide web conference*, pp. 2022–2032, 2019.

[17] Y. Ren, B. Liu, and C. Huang, et al, "Hdgi: An unsupervised graph neural network for representation learning in heterogeneous graph," *AAAI Workshop*, pp. 1638–1645, 2020.

[18] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," in *Advances in neural information processing systems 30(NIPS 2017)*, NIPS,2017.

[19] Z. Hu, Y. Dong, and K. Wang, et al, "Heterogeneous Graph Transformer," in *proceedings of The Web Conference 2020*, pp. 2704–2710, 2020.

[20] H. Hong, H. Guo, and Y. Lin, et al, "An Attention-based Graph Neural Network for Heterogeneous Structural Learning," in *proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 04, pp. 4132–4139, 2020.

[21] Y. Chang, C. Chen, and W. Hu, et al, "Megnn: Meta-path extracted graph neural network for heterogeneous graph representation learning," *Knowledge-Based Systems*, vol. 235, 107611, 2022.

[22] C. Shi, B. Hu, and W.-X. Zhao, et al, "Heterogeneous Information Network Embedding for Recommendation," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 2, pp. 357–370, 2018.

[23] H. Peng, R. Zhang, and Y. Dou,et al, "Reinforced Neighborhood Selection Guided Multi-Relational Graph Neural Networks," in *ACM Transactions on Information Systems (TOIS)*, vol. 40, no. 4, pp. 1–46, 2021.

[24] Y. Chen, L. Wu, and M.-J. Zaki, "Reinforcement Learning Based Graph-to-Sequence Model for Natural Question Generation," in *International Conference on Learning Representations (ICLR 2020)*, ICLR,2020.

[25] Z. Zhong, C.-T. Li, and J. Pang, "Reinforcement Learning Enhanced Heterogeneous Graph Neural Network," *arXiv preprint arXiv:2010.13735*, 2020.

[26] W. Xiong, T. Hoang, and W.-Y. Wang, "DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning," in *proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 564–573, 2017.

[27] K.-H. Lai, D. Zha, and K. Zhou, et al, "Policy-GNN: Aggregation Optimization for Graph Neural Networks," in *proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 461–471, 2020.

[28] C. Guan, X. Wang, and H. Chen, et al, "Large-scale graph neural architecture search," *International Conference on Machine Learning*, pp. 7968–7981, 2022.

[29] Z. Sun, Z.-H. Deng, and J.-Y. Nie, et al, "RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space," in *International Conference on Learning Representations (ICLR 2019)*, ICLR,2019.