Neural Computing and Applications (2023) 35:3357–3372
https://doi.org/10.1007/s00521-022-07862-6

ORIGINAL ARTICLE

Multimodal heterogeneous graph attention network

Xiangen Jia[1] · Min Jiang[2] · Yihong Dong[1] · Feng Zhu[1] · Haocai Lin[1] · Yu Xin[1] · Huahui Chen[1]

Received: 28 February 2022 / Accepted: 21 September 2022 / Published online: 10 October 2022

Abstract

The real world involves many graphs and networks that are essentially heterogeneous, in which various types of relations connect multiple types of vertices. With the development of information networks, node features can be described by data of different modalities, resulting in multimodal heterogeneous graphs. However, most existed methods can only handle unimodal heterogeneous graphs. Moreover, most existing heterogeneous graph mining methods are based on meta-paths that depend on domain experts for modeling. In this paper, we propose a novel multimodal heterogeneous graph attention network (MHGAT) to address these problems. Specifically, we exploit edge-level aggregation to capture graph heterogeneity information to achieve more informative representations adaptively. Further, we use the modality-level attention mechanism to obtain multimodal fusion information. Because plain graph convolutional networks can not capture higher-order neighborhood information, we utilize the residual connection and the dense connection access to obtain it. Extensive experimental results show that the MHGAT outperforms state-of-the-art baselines on three datasets for node classification, clustering, and visualization tasks.

Keywords Multimodal · Heterogeneous networks · Graph convolutional networks · Network representation learning · Clustering

1 Introduction

Many scenes in the real world can be described by graphs, such as social relationships, user-commodity interactions, and paper-to-paper citation relationships. The nodes in these networks are represented as dense semantic vectors with low-dimensions by mining the implicit information in graphs. These vectors preserve the original data graph's structural and attribute features as much as possible to perform node classification, clustering, and visualization.

It is not easy to directly model and analyze graph data using traditional machine learning methods because it is non-Euclidean space data. Some scholars have made some efforts, proposing that [24] uses random walks to generate a sequence of nodes and combines a skip-gram model to generate a low-dimensional vector representation. Node2-vec [9] extended Deepwalk with more sophisticated random walks and breadth-first search schema to improve the embedding effect. LINE [33] adds second-order similarity to first-order neighbor similarity to capture network structure information. With deep learning development, the convolutional neural network (CNN) has made great strides in computer vision. Inspired by the convolutional neural network, Thomas et al. [15] proposed a graph

✉ Yihong Dong
dongyihong@nbu.edu.cn

Xiangen Jia
xiangen2020@163.com

Min Jiang
j3966@163.com

Feng Zhu
m18888641057@163.com

Haocai Lin
810335314@qq.com

Yu Xin
xinyu@nbu.edu.cn

Huahui Chen
chenhuahui@nbu.edu.cn

[1] Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315040, China

[2] Ningbo Smart Urban Management Center, Ningbo 315041, China

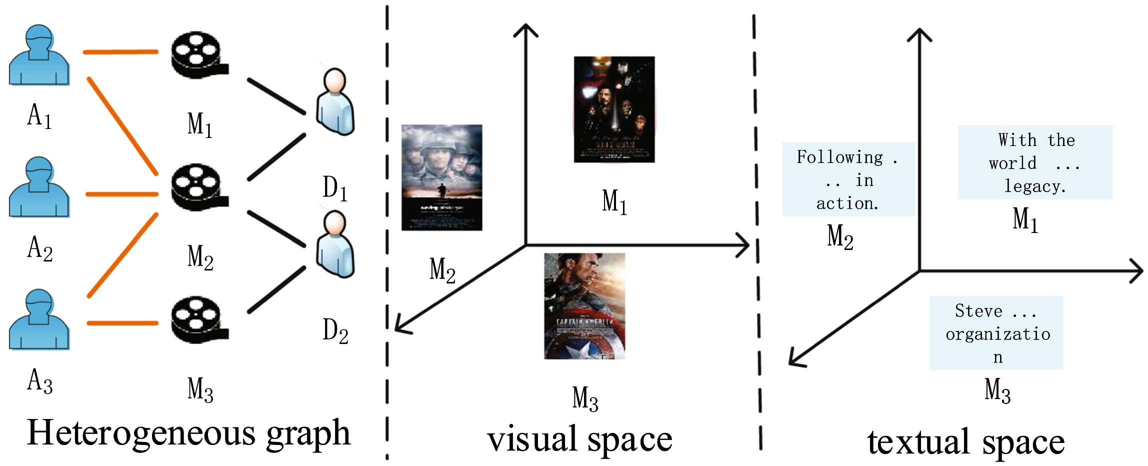


Fig. 1 A movie information graph specific modality illustration

convolutional network(GCN), which inherits the ability of CNNs in feature extraction to obtain embedded representations by message passing aggregate neighborhood node feature vectors. The attention mechanism introduced by GAT [35] based on GCN further enhances the aggregation capability of the network. However, most real networks are heterogeneous networks that contain richer semantic information instead of homogeneous ones. The most common method can be exploited to convert heterogeneous graphs into homogeneous graphs for modeling and analysis by meta-paths to capture heterogeneous information in the network. For example, as it is shown in Fig. 1, the movie information graph contains three types of vertices, including actor(A), movie(M), and director(D). The meta-path “A-M-A” indicates the actors’ co-act relationship, while “M-A-M” indicates movies acted by the same actor. Apart from using random walks based on meta-paths to extract node structure information, Metapath2vec [7] also uses skip-gram to learn node vector representations containing heterogeneity information. After transforming the heterogeneous graph into multiple metapath-based homogeneous graphs, HAN [37] uses a hierarchical graph attention network to aggregate neighbors’ information and exploits the attention mechanism to combine individual meta-paths.

Heterogeneous graph convolutional networks also have significant performance in many real-world scenarios. Community detection [32] aims to group nodes in a chart into clusters with dense internal connections. CP-GNN [20] recursively embeds higher-order relationships between nodes into node embeddings and uses attention mechanisms to distinguish the importance of different relationships. The model can learn node embeddings that are both well preserved for higher-order relationships between nodes and well applied for community detection. The convolutional graph network captures the interactions between individual elements represented by nodes in the graph. In particular, in medical applications, nodes can represent individuals in a potentially large population (patients or healthy controls) accompanied by a set of features. At the same time, the edges of the graph intuitively contain associations between subjects. Parisot et al. [23] exploit graph convolutional network and represent populations as a sparse graph, where its nodes are associated with imaging-based feature vectors. Meanwhile, phenotypic information is integrated as edge weights, to train the model using node classification.

However, there are two challenges in heterogeneous network representation learning:

- How to reduce domain experts’ dependency and information loss during heterogeneous network representation learning. Most methods are based on meta-paths, which needs to be extracted by domain experts. Moreover, during random walks based on meta-paths, they maybe lose information. Although HetSANN [12] can map neighborhood nodes into the low-dimensional space by using different node type transformation matrices, which aggregates the low-dimension neighborhood node representations through an attention mechanism to process the heterogeneous graph directly, these matrices in different types of neighbor nodes in each layer lead to too many model parameters.
- How to utilize multimodal information in heterogeneous graphs. Most existed methods mentioned are applied to unimodal graphs. With the development of information networks, network node features can be represented in many forms, such as text, images, and videos. As shown in Fig. 1, the movie can be described as a text or presented as a poster in a movie information graph. Images are more expressive than textual forms, which can show information about a scene in a movie.

The heterogeneous graph with different modal information forms a multimodal heterogeneous graph. This graph contains rich semantic features, but there is a semantic gap between different modalities. How to fuse modal information becomes a research hotspot. Moreover, the traditional graph convolutional networks cannot be stacked on multiple layers, which results in the model not being able to extract higher-order semantics.

To cope with the above two challenges, we design a Multimodal Heterogeneous Graph Attention Network (MHGAT) to address the above problems through edge-level aggregation, multimodal fusion, and high-order information mixing. To obtain heterogeneity information, MHGAT groups the connected nodes according to their edge types to aggregate to form edge-level vectors, reducing redundant information. Then, the attention mechanism is utilized to aggregate the edge-level vectors adaptively. The proposed model aggregates different modalities through a modality-level attention mechanism that adaptively perceives the modal weights for different tasks. MHGAT further combines the residual connection to prevent over smoothing for obtaining higher-order semantic information through the dense connection. The main contributions of this paper are as follows:

- We emphasize the critical importance of clearly using multimodal data. This paper proposes a novel multimodal heterogeneous graph attention network named MHGAT, which can learn multimodal heterogeneous graph representation.
- In order to handle multimodal heterogeneous graphs, we propose the modality-level attention mechanism, which assigns different modalities with different weights adaptively, to get multimodal fusion semantic information.
- To adaptively perceive heterogeneous information, MHGAT groups neighborhood nodes according to connected edge types. It performs intra-group aggregation to form edge-level feature representations, converted into edge-level feature dimensions by sharing parameters. Finally, the attention mechanism is introduced to acquire neighborhood information. It reduces the complexity of the model and alleviates the inefficient computation of softmax in the attention mechanism.
- There are few public datasets for multimodal heterogeneous graphs. Therefore, we constructed three new multimodal heterogeneous graph datasets(IMDB, AMAZON, and DOUBAN). The new datasets contain information on both image and text modal information. These new datasets can facilitate future research exploring multimodal heterogeneous graphs. We conducted extensive experiments on these three datasets, using two edge-level aggregators, mean and max. MHGAT outperforms other state-of-the-art models on node classification, clustering, and visualization tasks.

2 Related work

2.1 Graph convolutional networks

With deep learning development, graph convolutional networks have become a research hotspot in graph representation learning. Existing graph convolutional networks are divided into spectral and spatial methods, where graph convolutional networks based on spectral methods define convolutional operations on the graph via Laplace transform. ChebyNet [6] employs graph Laplace's Chebyshev polynomial expansion to define the convolutional kernel. To make spectral convolutional networks useful for semi-supervised learning, Kipf et al. simplified ChebyNet and proposed a graph convolutional network (GCN) [15]. However, the convolution in the spectral domain has many limitations, and training the network involves entering the entire graph into the network, which cannot be adapted to large graphs and is less scalable. Graph convolutional networks based on spatial methods do not have these limitations of spectral methods. It defines the convolution on the node domain and aggregates feature information from neighbors. GraphSage [10] fixes the number of random samples in the neighborhood and aggregates the neighbor node information through an aggregate function defined at the node. It is also suitable for large-scale network representation learning. MLC-GCN [41] introduces an adaptive structure coarsening module that generates a series of coarse graphs to build convolutional networks based on these graphs for graph classification. GAT [35] leverages the attention mechanism to learn the weights of neighbor nodes and then exploits weighted summation to get node embedding. Meanwhile, the introduction of the attention mechanism gives interpretability to the model. SK-GCN [46] leverages the syntactic dependency tree and commonsense knowledge via GCN for aspect-level sentiment classification. Mix-hop [1] acquires embedding representation by neighborhood information orders of neighborhood information.AM-GCN [38] proposes an adaptive multi-channel graph convolutional network for semi-supervised classification to capture structural information and semantic information about node features in graphs. The networks mentioned above are modeled based on homogeneous graphs with only one type of nodes and edges in their graphs. In image recognition, MSGCN [40] is a method based on multi-scale graph convolutional networks that can propagate label information from a small number of labeled nodes to other unlabeled nodes for a wide range of remote sensing image recognition.

The attention mechanism has become one of the more effective mechanisms in becoming a convolutional graph network. Many graph representation learning efforts have introduced the attention mechanism. Graph attention networks have been proposed to learn the importance between a node and its neighbors and to fuse the neighbors for node classification. Multi-channel graph convolutional networks can enhance graph mining's ability to obtain information about different features in the graph. However, the above graph neural networks cannot handle various nodes and edges and can only be applied to homogeneous graphs.

2.2 Heterogeneous graph representation learning

There are different types of vertices and relations in the heterogeneous graph, rich in semantic information. Most existing heterogeneous graph embedding models are built based on meta-paths. Metapath2vec [7] generates walking sequences guided by meta-paths. The skip-gram model uses walking sequences to obtain embedding representations of nodes. HIN2VEC [8] considers not only different node types but also complex and diverse relationship types. Heer [44] obtains heterogeneous graph embeddings by utilizing representations to alleviate semantic differences between nodes in heterogeneous graphs. HERec [27] exploits meta-paths to transform the heterogeneous graph into homogeneous graphs in multiple dimensional projections and then utilizes Deepwalk for representation learning. GATNE [30] extends the model to multiple heterogeneous attribute networks for analyzing various interactions between users and products, which has been successfully applied in recommendation systems of e-commerce platforms. To capture the complex structure and rich semantic information in heterogeneous graphs, HAN [37] converts heterogeneous graphs into multiple metapath-based homogeneous graphs, uses graph attention network structures to aggregate neighbor information, and leverages the attention mechanism to combine various metapaths to obtain the final embedded representation. All of the methods mentioned above require setting up meta-paths, limiting the models' performance in some way. During random walks based on meta-paths, information from intermediate nodes may be ignored. HetSANN [12] uses a type conversion matrix to convert nodes of different types into the same low-dimensional space. It aggregates neighborhoods through the attention mechanism without pre-defined meta-paths. Moreover, all of these models can only handle unimodal data, blocking the rich semantics of multimodal data. In computer vision, Jing et al [14] obtained further performance improvements by transferring the information obtained from graph convolutional networks processing different heterogeneous network structures and tasks to multi-label classification and joint segmentation classification tasks. In natural language processing, HeteGCN [25] combines predictive text embedding and TextGCN [42] using different graphs in each layer of the network for learning feature embeddings to derive document embeddings.

However, the heterogeneous graph embedding methods introduced above are based on meta-paths. Although they may have improved performance over homogeneous graph embedding methods on some heterogeneous graph datasets, they cannot directly handle heterogeneous graphs to rely on neighborhood experts to extract meta-paths, which still has room for improvement. Moreover, all of the above methods cannot handle multimodal heterogeneous graphs.

2.3 Multimodal representation

In multimodal applications, how to realize multimodal representation is one of the most critical problems. However, few existing works combine heterogeneous network representation learning with multimodal representations. Multimodal representations [2] can be grouped into two main categories: joint representations and coordinated representations. Joint representation is the mapping of information from all modalities into the same vector space. One of the simplest forms of joint representation is unimodal concatenation. The neural network has become the most popular representation method with successful applications in computer vision [11], natural language processing [5], and speech recognition [26]. In recent years, an increasing number of related methods have been applied in multimodal representations. Mroueh et al. [22] establish automatic speech recognition algorithms by obtaining unimodal representations using deep neural networks and fusing speech and visual information representations at the last hidden layer. Silberer et al. [29]use self-encoders to extract modal representations from text and images, respectively, and address the problem of distributed representation of lexical meanings. Neural networks can fuse and learn multimodal information into the same semantic space representation. Probabilistic graphical models [3] are another mainstream approach to joint representations, constructing multimodal representations by latent random variables. Srivastava et al. [31] utilize a multimodal depth Boltzmann machine to learn the representation of image and text joint spaces. The joint representation needs to ensure that every modality is present, but different nodes may have different modal information in real networks. Unlike the joint representation, the coordinated representation characterizes each modality individually and coordinates them through constraints. Kiros et al. [16] use coordinated representations of text and images using LSTM models and rank loss to coordinate semantic spaces. In Personalized recommendation, MMGCN [39] uses user-item interactions to guide each modality's representations for further personalized micro-video recommendations. HGMF [4] stacked two layers of graph attention layers to achieve the fusion of incomplete multimodal data. In high-quality content recognition, MGCN [36] converts textual content and visual content into graphs providing full guiding semantics for high-quality content recognition.

Although these methods mentioned above can handle multimodal networks, they ignore the differences between different modalities. Data of different modalities are not equally important in the learning of the embedding representation of the network, e.g., images are more expressive than text, so it is not reasonable to assign the same weight to all modalities to achieve the effect of fully mining the embedding representation of the network. In our model, the common part and specific part of features are modeled separately to address the above issues.

3 Methodology

In this section, we will elaborate on a novel multimodal heterogeneous graph attention network (MHGAT). MHGAT consists of three major components: edge-level aggregation, modality-level Aggregation, and higher-order information fusion, as shown in Fig. 2. Pre-trained neural networks extract modal features. After obtaining the modal features, MHGAT uses edge-level aggregation to acquire the unimodal embedding vector of nodes. Then, MHGAT uses the modality-level attention mechanism to obtain multimodal fusion representations. Finally, node embeddings are obtained by higher-order information fusion.

3.1 Multimodal heterogeneous graph

A multimodal heterogeneous graph is defined as $G = (V, E, A)$, where V, E, and A denote the set of nodes, edges, and multimodal attributes, e.g., text, image, and audio, respectively. The heterogeneous graph associated node type mapping function: $\phi(v) : V \rightarrow O$ and edge type mapping function: $\varphi(e) : E \rightarrow R$. O and R are types of vertices and edges, where $|O| + |R| > 2$. There are different types of edge connections between nodes in a multimodal heterogeneous graph, and the nodes have different modal information.

3.2 Edge-level aggregation

In a heterogeneous graph, nodes are connected to their neighbors by edges of different types. Edges of the same type have similar semantic information. We group neighborhood nodes according to edge types and perform intragroup aggregation to capture valid information under the

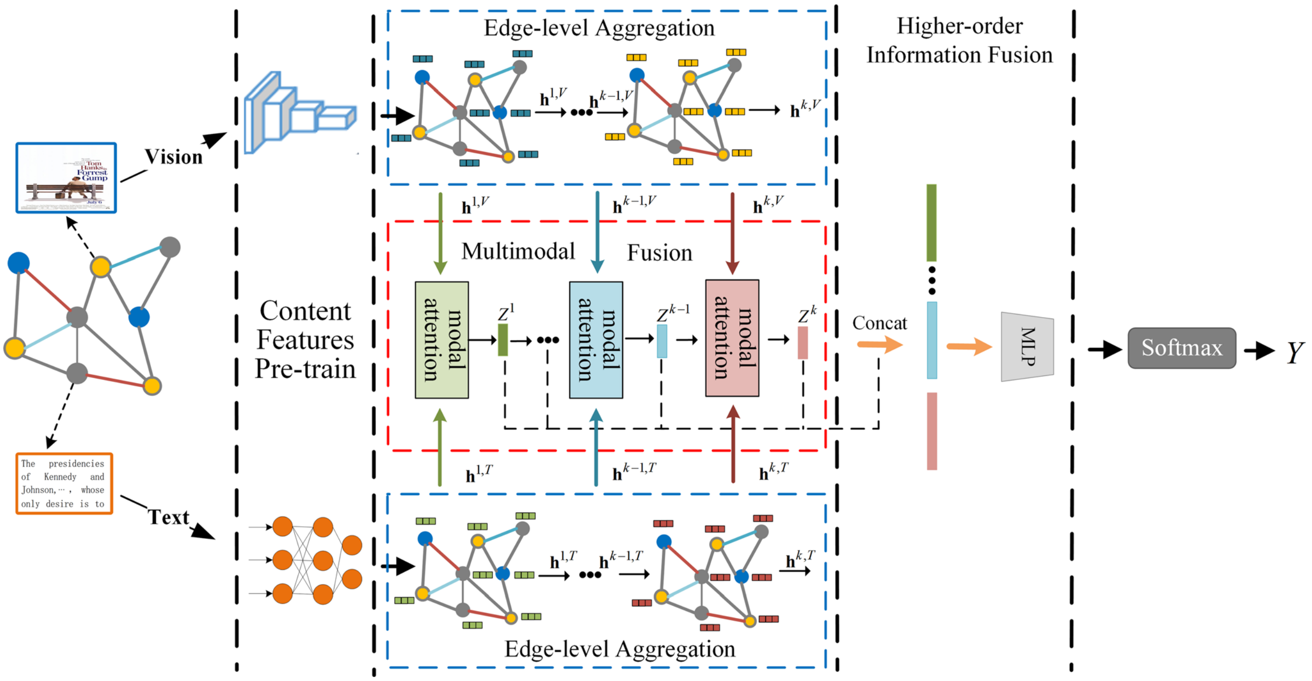


Fig. 2 The overall architecture of MHGAT. Through the pre-trained content extraction network, X-modality forms a unimodal heterogeneous graph. MHGAT consists of three modules: Edge-level aggregation, Modality-level aggregation, and High-order Information fusion

same edge-level semantics and reduce redundant information.

On the m-modality, the r-type edge embedding of the node in the k-th layer network is denoted as:

$$h_{i,r}^{k,m} = \mathrm{agg}\left(\left\{h_{j,r}^{(k-1),m}, \forall v_j \in N_{i,r}\right\}\right) \tag{1}$$

Where $\mathrm{agg}(*)$ denotes an aggregate function and $N_{i,r}$ is the neighbors of node v_i connected based on edge type r. $h_i^{(0),m}$ is denoted as a vector of initialized node v_i features on the m-modality. For example, it represents the image feature vector in the image modality or the text vector in the text modality, etc.

Inspired by Graphsage [10], the two aggregator functions used in the paper are: mean aggregator and max aggregator.

- *Mean aggregator* The mean-based aggregator is a local approximation of the spectral convolution. It is an average pooling operation in a particular pattern that does not require dimensional transformation.

$$h_{i,r}^{k,m} = \frac{1}{\left|N_{i,r}\right|} \sum_{j \in N_{i,r}} h_{j,r}^{(k-1),m} \tag{2}$$

- *Max aggregator* The aggregator uses max-pooling aggregation for feature selection and dimensional awareness. As follows:

$$h_{i,r}^{k,m} = \max\left\{h_{j,r}^{(k-1),m}, j \in N_{i,r}\right\} \tag{3}$$

In a heterogeneous graph, nodes are connected to different types of edges, containing various semantic information. Edge-level aggregation reduces information redundancy by integrating information from neighbor nodes in the same relationship. Different relations have different semantics. Since each semantic's importance to the nodes is different, we introduce the attention mechanism, which can be adaptive to perceive the importance of semantics and form more meaningful node embeddings.

Due to the different feature spaces of different types of edge embeddings, we designed a transformation matrix $M^{k,m}$ that can project the features of different edge types into the same feature space. The projection process is as follows:

$$h'^{k,m}_{i,r} = M^{k,m} \bullet h_{i,r}^{k,m} \tag{4}$$

Where $h'^{k,m}_{i,r}$ denotes the feature vector after the feature space transformation. By feature space transformation, the edge-level attention mechanism can handle embedding vectors of any type of edge.

After obtaining the feature transformation vectors on the m-modality, the self-attention mechanism is used to learn the weights of different types of edges. On the m-modality, the attention of edge type r for the node v_i in the k-th layer network is $e_{i,r}^{k,m}$. The attention of edge type r is calculated as follows:

$$e_{i,r}^{k,m} = \mathrm{LeakyReLU}\left(a^{k,m^T} \bullet \left[M^{k,m} \bullet h_i^{k,m} \parallel M^{k,m} \bullet h_{i,r}^{k,m}\right]\right) \tag{5}$$

Here $\parallel$ is the concatenation operation. $a^{k,m}$ is a trainable attentional transformation vector, and the LeakyReLU nonlinear function is applied as the activation function. Since $e_{i,r}^{k,m}$ is asymmetric, the attention on the edge of type r is entirely different from attention on its inverse relation r' for node v_i. This shows the asymmetric nature of the attention mechanism, enabling the extraction of heterogeneous information about the heterogeneous graph.

The attention mechanism allows every type of edge embedding to participate in the attention computation. Only the attention associated with the node via an edge embedding of type $r \in N_{i,R}$ needs to be calculated, where $N_{i,R}$ is the set of edge types connected to node v_i in the heterogeneous graph. To make embeddings of different edge types comparable, we normalize all attentions in the set of edge types using the softmax function.

$$s_{i,r}^{k,m} = \mathrm{softmax}\left(e_{i,r}^{k,m}\right) = \frac{\exp\left(e_{i,r}^{k,m}\right)}{\sum_{r^s \in N_{i,R}} \exp\left(e_{i,r^s}^{k,m}\right)} \tag{6}$$

Obviously, the weighting factors $s_{i,r}^{k,m}$ are different for different edge types. This is not due to the vector concatenation's inconsistency but derives from the different importance of different edge type embeddings on node v_i.

The embedding of node v_i in the k-th layer network on m-modality can be obtained by weighted aggregation of edge embedding and the corresponding coefficients. As follows:

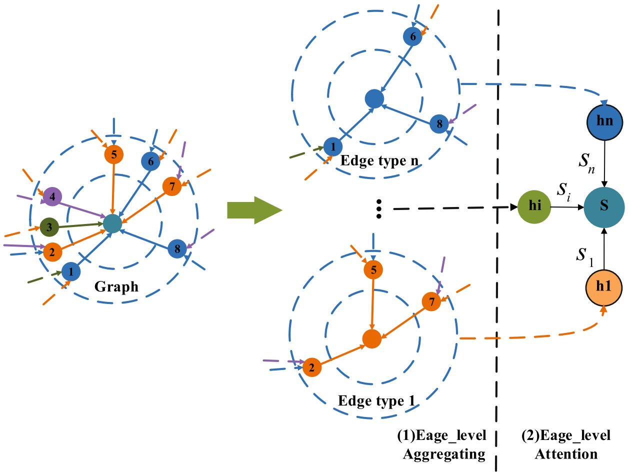


Fig. 3 Edge-level aggregation process

$$S_i^{k,m} = \sigma\left(\sum\nolimits_{r\in N_{i,R}} s_{i,r}^{k,m} \bullet h'^{k,m}_{i,r}\right) \tag{7}$$

Where $S_i^{k,m}$ is the embedding representation of node v_i in the k-th layer network on the m-modality. Figure 3 shows the edge-level attention mechanism's aggregation process. First, we aggregate groups according to edge types to form edge embedding vectors, and then we transform the feature space. Finally, we utilize the attention mechanism for edge-level aggregation to form unimodal representation vectors of nodes. The attention mechanism makes the aggregated vectors more meaningful, while the model is more adaptive and interpretable.

Li et al. [18] showed that two layers of GCNs perform best. If the network exceeds two layers, it will weaken GCN's performance and produce an over-smoothing phenomenon. To prevent the over-smoothing phenomenon, we introduce a residual connection [11]. On the m-modality, the embedding vector of node v_i in the k-th layer network is $S'^{k,m}_i$.As follows:

$$S'^{k,m}_i = S_i^{k,m} + h'^{k,m}_i \tag{8}$$

3.3 Multimodal fusion

Each node has different modal information in the multimodal heterogeneous graph, while each modal information has a specific semantic meaning. How to perform modal fusion is crucial. To fuse the multimodal information, we design the modality-level attention network. In this way, MHGAT can adaptively select important modal information.

To fuse multimodal information, it is necessary to unify the multimodal information into the common quantitative standard. We use two-layer feedforward neural networks to calculate the modalities' weight values, where the first layer network unifies the modal information into the same dimension, and the second layer network outputs the weight of the modality.

$$\omega_i^{k,m} = W_2 \bullet \tanh\left(W_1 \bullet S'^{k,m}_i + b\right) \tag{9}$$

Here, $\omega_i^{k,m}$ is the weight value of node v_i on the m-modality in the k-th layer network. W_1 and W_2 are learnable parameter matrices, and b is the bias vector. The tanh function was chosen as the activation function because it can better maintain the nonlinearity between features [19]. The modality-level attention network perceives different modalities differently, making multimodal information fusion interpretable. We use the softmax function to normalize all the modal weights and obtain the final modal attention score:

$$\delta_i^{k,m} = \text{softmax}\left(\omega_i^{k,m}\right) = \frac{\exp\left(\omega_i^{k,m}\right)}{\sum_{m'\in M}\exp\left(\omega_i^{k,m'}\right)} \tag{10}$$

Where M represents the set of all modalities; larger $\delta_i^{k,m}$ ndicates that the m-modality is more important for node v_i in the k-th layer network. The final embedding of node v_i fusing multimodal information in the k-th layer network is represented as follows:

$$Z_i^k = \sum\nolimits_{m\in M} \delta_i^{k,m} \bullet S'^{k,m}_i \tag{11}$$

The final node embedding vector is the vector after fusing multimodal semantics, and the nodes are richer in semantic information.

3.4 High-order information fusion

Some researchers have shown that there are many limitations for plain GCNs [45], which do not enable higher-order information access. To capture higher-order neighborhood information, the model needs to stack more layers. To retain more feature information, DenseNet [13] connects all network-level outputs as final output features. We refer to the similar structural design, where the embedding of node v_i is represented as F_i:

$$F_i = \mathop{\Big\|}_{k=\{1,2,\ldots,K\}} Z_i^k \tag{12}$$

Where $\|$ denotes the concatenation operation, Z_i^k is the output vector of node v_i in the k-th layer network. The modeling framework inherits the advantages of DenseNet, which alleviates the loss of gradients during network training and enhances feature transfer, making more efficient utilization of the output features of each layer. We add feature selection networks after the output feature vector to enhance the interaction between features. The final output node v_i embedding vector is denoted as F'_i:

$$F'_i = \partial\left(W_f \bullet F_i\right) \tag{13}$$

Where W_f is the matrix of trainable parameters, ∂ is the activation function. The output final vector mixes multimodal information and performs well for different tasks. To exploit this feature, the experiment uses semi-supervised node classification task training networks.

$$O_i = \text{softmax}\left(W_o \bullet F'_i\right) \tag{14}$$

Here, O_i is the final output of the classification. W_o represents the trainable dimensional adjustment matrix. With the guidance of the truth label, we minimize cross-entropy loss:

$$L = \sum_{i \in V_T} {y_i}^T \bullet \log(O_i) \tag{15}$$

Where y_i represents the true one-hot label vector for vertex v_i. V_T represents the set of vertices in the training set.

The overall MHGAT algorithm process is in Algorithm 1.

```
Algorithm 1 MHGAT
Input: Input data: Multimodal heterogeneous graph $G = (V, E, A)$
Output: Node Embedding $\{F'_v, \forall v \in V\}$
Multimodal initialization features of nodes obtained by pre-training models: H $= \{H^{0,1}, H^{0,2}, ..., H^{0,m}\}$
Edge-level Aggregation and Multimodal Fusion:
for $k = 1...K$ do
    for $i \in V$ do
        for $m = 1...M$ do
            for $r = 1...R$ do
                The neighborhood nodes are aggregated by edge type grouping into $h_{i,r}^{k,m}$.
                Semantic space transformation:${h'}_{i,r}^{k,m} \leftarrow M^{k,m} \bullet h_{i,r}^{k,m}$.
            end for
            Calculate Edge-Level Attention:$s_{i,r}^{k,m}$.
            Calculate vectors for nodes in the k-th layer network on the m-modality: $S_i^{k,m}$.
            Join the residual connection and the node vector is ${S'}_i^{k,m}$.
        end for
        Calculate modality-level attention:$\delta_i^{k,m}$.
        Obtaining multimodal fusion vectors for nodes in the k-th layer network: $Z_i^k \leftarrow \sum_{m \in M} \delta_i^{k,m} \bullet {S'}_i^{k,m}$.
    end for
    High-order information:$F_i \leftarrow \underset{k=\{1,2,...,K\}}{\|} Z_i^k$.
end for
High-order Information Fusion:
Final Node Output Embedding: $F'_i \leftarrow \partial \left(W_f \bullet F_i\right)$.
Node classification results: $O_i \leftarrow$ softmax $\left(W_o \bullet F'_i\right)$.
```

4 Experiments

We constructed three datasets. Many methods [37] were chosen for node classification, node clustering and visualization, tasks that measure the network representation learning ability of the model. Therefore, we chose these tasks as comparison experimental tasks. Finally, we performed modal-level attention mechanism analysis and parameter sensitivity analysis on MHGAT (Table 1).

Table 1 Datasets

Dataset	Nodes	Edge	Edge types	Classes
IMDB	11616	34212	4	3
AMAZON	13189	174154	3	3
DOUBAN	6627	15032	4	2

4.1 Datasets and experimental settings

All datasets are of real scenarios and have practical significance. As follows:

IMDB[1] is data from online movie sites, which includes information about actors and movie profiles. We select these movies (M), actors (A), and directors (D) to construct a heterogeneous graph of their relationships and use the movies as the classification target. The movies are labeled into three categories based on their labels: action, comedy, and drama. We used poster images and synopses of movies as information about the two modalities of the movie. To extract text and image features, we used the bag-of-words model and the pre-trained Resnet [11], respectively. There are four edge types in the constructed heterogeneous graph: AM, MA, MD, and DM.

AMAZON[2] contains product reviews and metadata from Amazon, including user (U) reviews of products (I), images of products, and descriptions of products, etc. As there are many different types of products, we selected the appliances category for dataset construction and extracted the subset of these by review time. We take product images and product descriptions as the features. We also used the bag-of-words and pre-trained Resnet to extract their modal features. There are three semantic relationships in the constructed graph: UI, IU, and II.

DOUBAN[3] is data from the Douban movie website, including movie profiles and information of actors and directors. We construct the heterogeneous graph using the relationship between movies (M), actors (A), and directors (D) to classify movies into two categories: comedy and action. We use the poster image of the movie and the movie profile as two modalities. We used the pre-trained Resnet to extract the image features. To extract the Chinese text features, we used the doc2vec [17] model. There are four types of edge types in the graph: AM, MA, MD, and DM.

We adopt the Macro-F1, Micro-F1, AUC, NMI, and ARI as the metrics for evaluation, which are defined as:

- TP - positive sample predicted to be positive by the model.
- FP - Negative samples predicted to be positive by the model.

[1] https://www.imdb.com.

[2] http://deepyeti.ucsd.edu/jianmo/amazon.

[3] https://movie.douban.com.

- TN - Negative samples predicted to be negative by the model.
- FN - positive sample predicted to be negative by the model.

$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{16}$$

Micro-f1: Calculate F1 by first calculating the global number of TP, FN, and FP.

Macro-f1: calculate F1 for each category separately, then do the average.

AUC is the area enclosed by the ROC curve and axis. The closer the AUC value is to 1, the better the performance of the model.

NMI(Normalized Mutual Information) is standardized mutual information that measures the degree of agreement between two data distributions.

$$NMI(Y,X)=\frac{2 * \mathrm{I}(Y;X)}{H(Y)+H(X)} \tag{17}$$

Where Y is the class labels; X is the cluster labels; $H(\bullet)$ is an information entropy calculation function; $\mathrm{I}(Y;X)$ is the mutual information b/w Y and X.

ARI (Adjusted Rand index) is a measure of the similarity between two clusters of data. Assume that U is the external evaluation criterion, i.e., the true label, and V is the clustering result. Set four statistics as follows:

- a - The number of pairs of data points that are of the same class in U and also of the same class in V.
- b -The number of pairs of data points that are not in the same class in U and are not in the same class in V.
- c - The number of pairs of data points that are not of the same class in U, but are of the same class in V.
- d - The number of pairs of data points that are of the same class in U, but belong to different classes in V.

$$RI = \frac{a+b}{a+b+c+d}, ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} \tag{18}$$

Our experiments have been conducted in the following environmental setting:

- Operating system: Ubuntu 18.04
- CPU: Intel(R) Xeon(R) CPU i7-9700k@ 3.60GHz
- GPU: GeForce GTX 2080 Ti
- Software versions: Python 3.7;Pytorch 1.4.0;Numpy 1.16.2;NetworkX 2.4; scikit-learn 0.22.2; dgl-cuda10.1 0.4.3; Pandas 1.1.3

4.2 Baselines

To verify the model's effectiveness, five state-of-the-art models are selected for comparison, including unsupervised and semi-supervised methods, as follows:

Deepwalk [24] utilizes random walk combined with the Skip-gram model to obtain node embedding representations.

Metapath2vec [7] acquires structural information about the graph based on metapath-guided random walks and learns to obtain representations using the Skip-gram model.

GCN [15] is a semi-supervised graph convolution network that obtains node representations by aggregating its neighbors' feature vectors.

GAT [35] exploits the attention mechanism to perform node representation learning combined with graph convolution network.

HAN [37] combines graph attention network with metapaths to propose a hierarchical heterogeneous graph attention network.

HetSANN [12] converts nodes of different types into the same low-dimensional space, aggregating neighborhoods through the attention mechanism.

Simple-HGN [21] extends the attention mechanism of GAT by adding edge type information to the attention calculation.

MHGATmax is MHGAT edge-level aggregation using the max aggregator.

MHGATmean is MHGAT edge-level aggregation using the mean aggregator.

To train the model effectively, we trained the model with a learning rate of 0.001, 4 layers of the model, a maximum number of iterations of 1000, a patience of 100 using the early stopping strategy, by using Adam to optimize the model. We applied dropout to the output of each layer with a dropout rate of 0.6. To ensure a fair comparison, we set the output dimension uniformly to 64. The best parameter settings for comparing models were chosen in the original paper.

4.3 Node classification

Node classification is an essential task for measuring network representation learning models. MHGATmean and MHGATmax use semi-supervised training. We divided the dataset into the training set (20%), the validation set (10%), and the test set (70%). The model forms embedding representations of nodes by aggregating the semantics of different modalities. We selected the embedding vectors of nodes in test set and used 20%, 40%, 60%, and 80% of the data, respectively, to train the logistic regression classifier accordingly. To ensure the experimental results' validity, the mean values of Micro-F1, Macro-F1, and AUC obtained by running the model 10 times were selected as the results, with the higher the index indicating better embedding.

Table 2 shows that the two aggregator functions used by MHGAT produced better experimental results at different

Table 2 Classification results of the evaluation

Dataset	Metrics	Train %	Unsupervised		Semi-supervised						
			Deepwalk	Metapath2vec	GCN	GAT	HAN	HetSANN	Simple-HGN	MHGATmax	MHGATmean
			(2014)	(2017)	(2017)	(2018)	(2019)	(2020)	(2021)		
IMDB	Macro-F1	20	0.4824	0.4612	0.6604	0.7026	0.7630	0.7429	0.7468	0.7948	**0.7983**
		40	0.4924	0.4828	0.6689	0.7110	0.7742	0.7494	0.7600	0.7996	**0.8020**
		60	0.5048	0.4801	0.6734	0.7087	0.7758	0.7461	0.7663	0.7978	**0.8022**
		80	0.5147	0.4900	0.6712	0.7120	0.7744	0.7430	0.7652	0.7979	**0.8006**
	Micro-F1	20	0.4985	0.4832	0.6693	0.7098	0.7662	0.7468	0.7506	0.7974	**0.8010**
		40	0.5099	0.5022	0.6769	0.7178	0.7768	0.7533	0.7635	0.8022	**0.8046**
		60	0.5213	0.5010	0.6810	0.7151	0.7782	0.7498	0.7693	0.8004	**0.8046**
		80	0.5332	0.5128	0.6806	0.7195	0.7778	0.7481	0.7694	0.8013	**0.8044**
	AUC	20	0.6239	0.6124	0.7521	0.7823	0.8246	0.8101	0.8129	0.8481	**0.8508**
		40	0.6324	0.6266	0.7577	0.7884	0.8325	0.8150	0.8227	0.8517	**0.8534**
		60	0.6410	0.6257	0.7607	0.7863	0.8337	0.8123	0.8270	0.8503	**0.8534**
		80	0.6499	0.6346	0.7605	0.7896	0.8333	0.8111	0.8270	0.8510	**0.8533**
AMAZON	Macro-F1	20	0.5223	0.6253	0.6800	0.7150	0.7517	0.7857	0.7928	**0.8301**	0.8205
		40	0.5350	0.6349	0.6899	0.7266	0.7593	0.7896	0.8029	**0.8324**	0.8224
		60	0.5369	0.6441	0.6944	0.7306	0.7617	0.7909	0.8046	**0.8318**	0.8256
		80	0.5390	0.6491	0.6961	0.7346	0.7646	0.7921	0.8087	**0.8347**	0.8340
	Micro-F1	20	0.6937	0.6468	0.7988	0.8060	0.8277	0.8478	0.8487	**0.8782**	0.8735
		40	0.7076	0.6555	0.8042	0.8099	0.8330	0.8512	0.8563	**0.8801**	0.8746
		60	0.7123	0.6640	0.8059	0.8127	0.8352	0.8529	0.8575	**0.8797**	0.8766
		80	0.7146	0.6690	0.8065	0.8135	0.8358	0.8533	0.8587	0.8802	**0.8819**
	AUC	20	0.7703	0.6468	0.8491	0.8545	0.8708	0.8859	0.8865	**0.9086**	0.9051
		40	0.7807	0.6555	0.8531	0.8574	0.8747	0.8884	0.8922	**0.9101**	0.9059
		60	0.7842	0.6640	0.8544	0.8585	0.8764	0.8897	0.8931	**0.9098**	0.9074
		80	0.7860	0.6690	0.8549	0.8601	0.8769	0.8900	0.8940	0.9101	**0.9114**
DOUBAN	Macro-F1	20	0.6141	0.6253	0.8493	0.8787	0.8623	0.8818	0.8555	**0.8956**	0.8922
		40	0.6307	0.6349	0.8484	0.8798	0.8635	0.8822	0.8764	**0.8968**	0.8932
		60	0.6449	0.6441	0.8527	0.8815	0.8629	0.8819	0.8813	**0.8977**	0.8924
		80	0.6523	0.6492	0.8550	0.8782	0.8668	0.8804	0.8863	**0.9004**	0.8946
	Micro-F1	20	0.6245	0.6468	0.8519	0.8807	0.8647	0.8839	0.8578	**0.8976**	0.8944
		40	0.6421	0.6555	0.8508	0.8816	0.8657	0.8844	0.8786	**0.8988**	0.8955
		60	0.6553	0.6640	0.8547	0.8830	0.8661	0.8837	0.8831	**0.8993**	0.8944
		80	0.6630	0.6690	0.8569	0.8799	0.8688	0.8823	0.8881	**0.9021**	0.8966
	AUC	20	0.6245	0.6468	0.8519	0.8807	0.8647	0.8839	0.8578	**0.8976**	0.8944
		40	0.6421	0.6555	0.8508	0.8816	0.8657	0.8844	0.8786	**0.8988**	0.8955
		60	0.6553	0.6640	0.8548	0.8830	0.8661	0.8837	0.8831	**0.8993**	0.8944
		80	0.6630	0.6690	0.8568	0.8799	0.8688	0.8823	0.8881	**0.9021**	0.8966

Bold indicates the best model performance in the current test task

training scales than all baselines. GCN combines structure and features for better performance than traditional network embeddings (Deepwalk, Metapath2vec). GAT is a network based on the attention mechanism. The embedding effect is better than plain GCN because the attention mechanism network can select more meaningful semantic information. HAN combines meta-paths and attention mechanisms to obtain richer heterogeneous semantic information and performs better on IMDB and AMAZON datasets than GAT, which considers only homogeneous information. HetSANN transfers neighborhood nodes to the same semantic space for aggregation, which performs better than HAN on AMAZON and DOUBAN datasets. Simple-HGN outperforms HetSANN on IMDB and

Table 3 Node clustering results

Dataset	Metrics	Unsupervised		Semi-supervised						
		Deepwalk (2014)	Metapath2vec (2017)	GCN (2017)	GAT (2018)	HAN (2019)	HetSANN (2020)	Simple-HGN (2021)	MHGATmax	MHGATmean
IMDB	NMI	0.0050	0.0057	0.1197	0.2350	0.2649	0.2585	0.2686	0.3068	**0.3085**
	ARI	−0.0022	−0.0022	0.1199	0.2823	0.2888	0.2998	0.3193	0.3586	**0.3630**
AMAZON	NMI	0.0141	0.0145	0.1781	0.2930	0.2438	0.3143	0.3509	0.4125	**0.4758**
	ARI	0.0278	−0.0302	0.1407	0.3784	0.2490	0.4034	0.3557	0.5212	**0.6007**
DOUBAN	NMI	0.0167	0.0019	0.0241	0.4205	0.3872	0.4354	0.4395	0.4924	**0.5041**
	ARI	−0.0104	−0.0020	0.0428	0.5263	0.4241	0.5414	0.5492	0.5993	**0.6138**

Bold indicates the best model performance in the current test task

AMAZON datasets due to its ability to adaptively sense different relations types. MHGAT utilizes the attention mechanism to select the appropriate neighborhood information and combines the semantic information of different modalities to make the best performance. By choosing the appropriate aggregator function, we can apply MHGAT to different scenarios. Compared to the most competitive HAN, MHGAT also has a high-performance improvement. Using the mean aggregate function on IMDB and DOUBAN datasets works better than the max aggregate function, and the max aggregate function works better on the AMAZON dataset.

The above analysis shows that MHGAT performs best. In heterogeneous graphs, it is essential to explore heterogeneous information in the network and fuse multimodal information.

4.4 Clustering

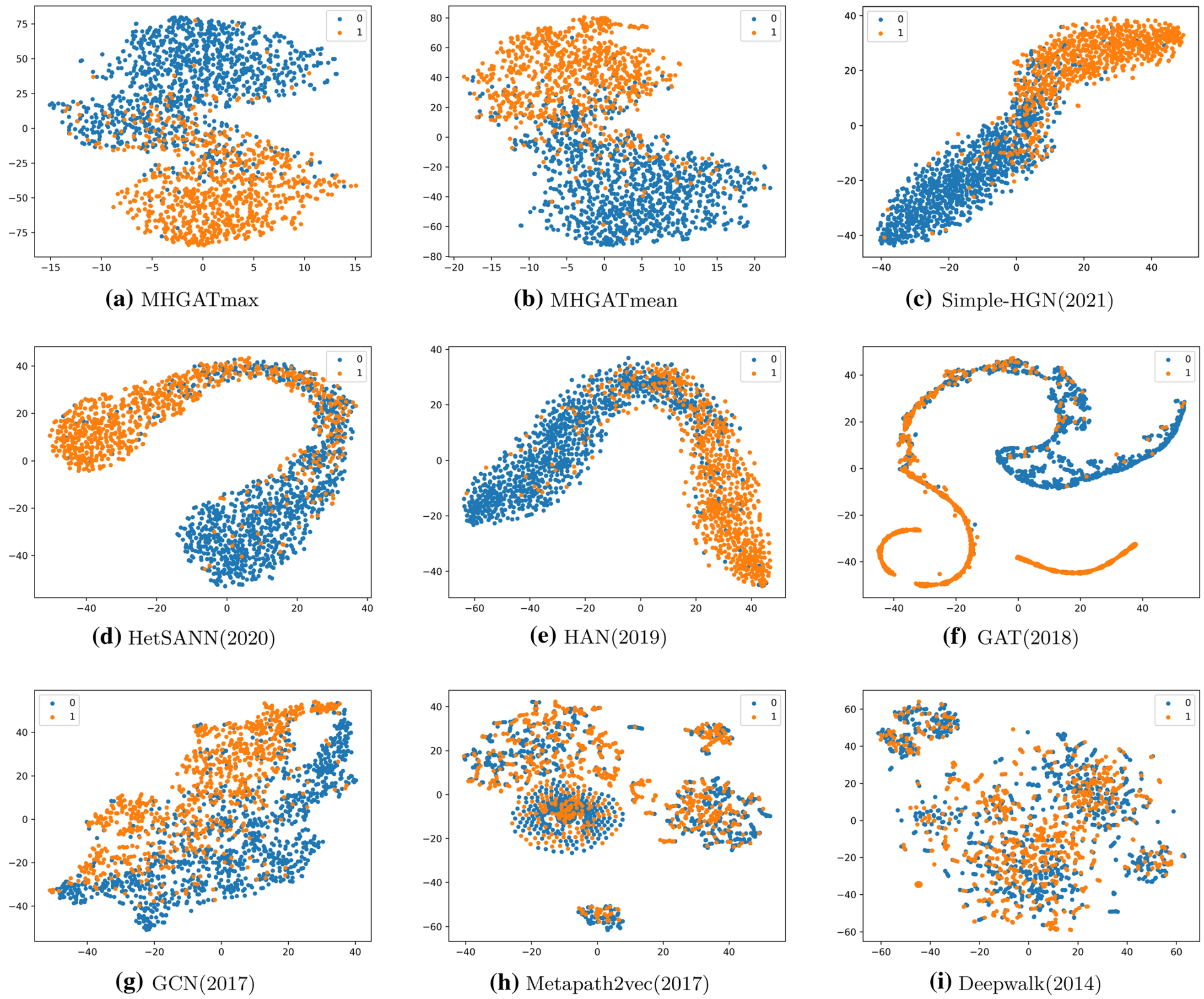
Clustering [28] is the process of dividing groups of data into clusters, where objects in the same cluster are similar to each other. We train each model using node classification tasks for obtaining test set node representations. The node representations and corresponding labels are clustered via K-means, where the number of categories clustered is the number of vertex types. NMI and ARI are used as evaluation metrics, with larger values indicating better model representation. Since the clustering result is affected by the initialization of centroids of k-means, the experiment is repeated ten times, and the average result is taken as the final result, as shown in Table 3.

As can be seen in Table 3, MHGATmean and MHGATmax model embedding performance is superior to all baselines. Shallow models, such as Deepwalk and Metapath2Vec, perform worse than the deep model because the random-walks-based approach forces nodes to be close to each other in the embedding space [43]. Thus models based on random walks can result in inadequate exploration of heterogeneous information in heterogeneous networks. GCN does not exploit heterogeneous information during aggregation of neighbor nodes, performing poorly in these depth models. GAT performs better using the attention mechanism than the plain GCN. HAN, which incorporates heterogeneous information, outperforms GAT on IMDB, but GAT outperforms HAN on AMAZON and DOUBAN. On AMAZON and DOUBAN, the meta-path-based HAN will cause information loss, while the adaptive-aware HetSANN fully explores the information about the node neighbors and learns the appropriate weights for different neighborhoods. In the unimodal model, Simple-HGN perceives different relationship types, and its performance is better. The best results model is MHGAT, which can adaptively perceive heterogeneous information about heterogeneous graphs and fuse multimodal information to provide a comprehensive representation of multimodal heterogeneous graphs.

4.5 Visualization

Visualization is an essential application in network representation learning, reflecting visually embedding nodes' community division structure. For a more intuitive comparison, we visualized the DOUBAN dataset. t-SNE [34] is used to project the embedding vectors of nodes in the test set into 2D space and color them according to movie types.

Figure 4 shows that the different types of movies in GAT and GCN are mixed, with no clear separation between them. In GAT, apart from the same category's nodes are farther apart, there is more overlap between the different categories. Deepwalk is entirely indistinguishable from the categories, and all of them are mixed up. Since metapath2vec considers only one type of meta-path, it causes information loss. The differences between the nodes are not visible at all in the metapath2vec visualization image. Compared to metapath2vec, the HAN performs better under the guidance of multiple metapaths. However,



(a) MHGATmax **(b)** MHGATmean **(c)** Simple-HGN(2021)

(d) HetSANN(2020) **(e)** HAN(2019) **(f)** GAT(2018)

(g) GCN(2017) **(h)** Metapath2vec(2017) **(i)** Deepwalk(2014)

Fig. 4 Visualization

the smaller distances between node classes and more mixing indicate that HAN is losing much heterogeneous information. HetSANN has a more prominent separation compared to HAN. Simple-HGN has less overlap in the category compared to HetSANN. Our proposed model MHGAT can adaptively acquire heterogeneous information and fuse multimodal information to obtain richer semantics. MHGAT can be divided into different communities with high intra-class similarity. The above results show that MHGAT outperforms the baselines in visualization.

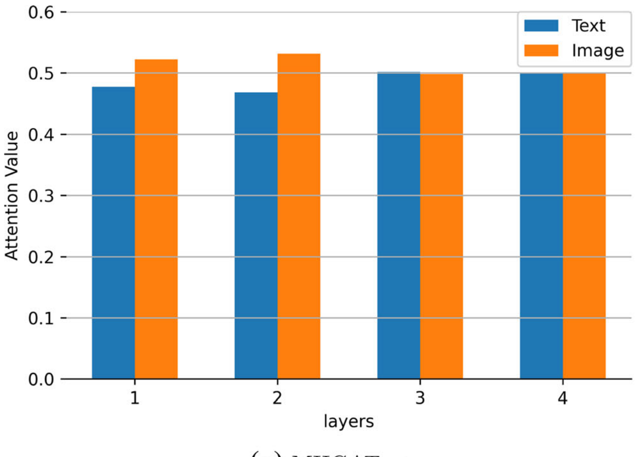
4.6 Analysis of modality-level attention mechanism

MHGAT can adaptively sense the importance of different modalities when learning feature representations using the modality-level attention mechanism. To better understand the importance of different node representation modalities, we selected M1879 nodes on the DOUBAN dataset for the analysis. We fix the network level to 4 layers and select the node classification task to train the network.

In Fig. 5, it can be seen that the attention values are higher for images than for text in the first two layers of the network due to the richer information contained in the image features. As the network levels increase, the text attention values and image attention values tend to be similar for the network to acquire more integrated semantics.

4.7 Parameter sensitivity

To investigate the effect of changing parameters on the model, this section performs a parameter sensitivity



(a) MHGATmax **(b)** MHGATmean

Fig. 5 Unimodal attention value

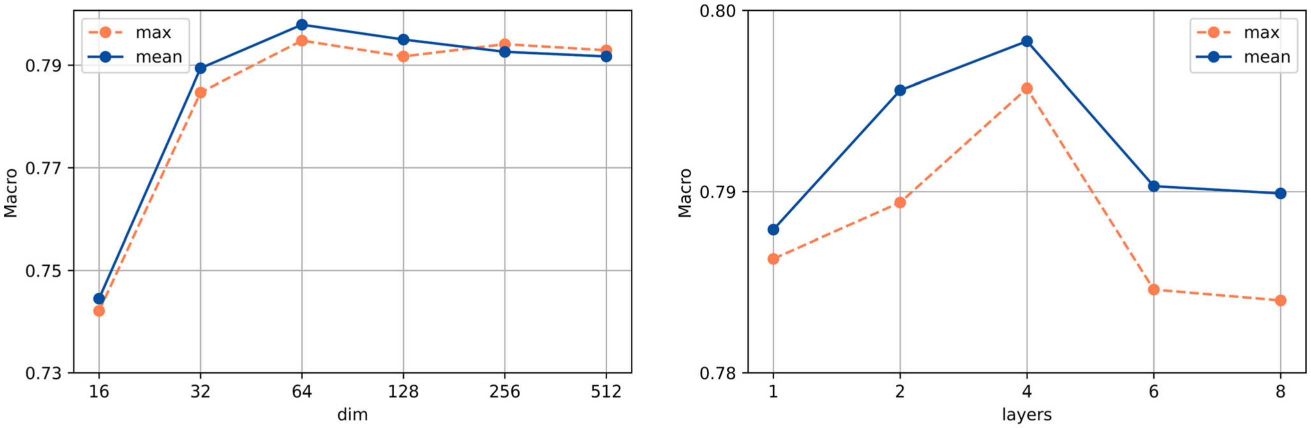
analysis on the IMDB dataset. Taking the node classification task as an example, the effects of different embedding dimensions and network layers on the model performance are analyzed.

Dimension of the embedding. We fixed the embedding network hierarchy to 4 layers. Figure 6a shows that the node embedding effect increases as the embedding dimension increases, with a peak at dimension 64. If the embedding dimensions increase, the performance decreases, indicating that the excessively high dimensions may fuse more noisy information.

Network layer In the network hierarchy experiment in Fig. 6b, we fixed the embedding dimension to 64 dimensions. GCN [45] performs optimally when the network level is two layers, beyond which the over smoothing phenomenon occurs. Since GCN continuously aggregates neighboring node features, the global node features tend to be similar after multi-layer network aggregation, while weakening the nodes' features and producing over smoothing. MHGAT has the best performance when the network level reaches four layers. This is due to the fact that MHGAT retains the features of all network layers so that the final node representation contains not only its own feature information but also a mixture of higher-order semantic information. As the number of network levels increases, the model has relatively more parameters, overfitting occurs, making the performance decrease.

4.8 Discussion

Homogeneous network modeling approaches fail to distinguish the heterogeneity of objects and relations between them in the actual interaction data, resulting in irreversible information loss. Deepwalk, GCN, and GAT are homogeneous network models, while GAT performs best among homogeneous network analysis approaches due to its ability to aggregate important neighbor node information. However, they all ignore the graph's multiple types of node



(a) Dimension of the embedding **(b)** Network layer

Fig. 6 Parameter sensitivity

and edge heterogeneity information, leading to sub-optimal results. Compared with homogeneous networks, heterogeneous networks contain different types of objects and their interactions, which contain rich structural and semantic information, thus providing interpretability for implicit mining information.

Most existing heterogeneous network approaches are based on meta-paths, which essentially extract the substructure of the heterogeneous network and reflect the rich semantic information contained in the paths. The selection of meta-paths relies heavily on domain knowledge. It is difficult for unfamiliar or complex heterogeneous networks to rely on domain knowledge to select the appropriate meta-paths. Moreover, as the meta-path length increases, the random wandering process of the paths is very time-consuming and can cause information loss. The meta-path-based modeling approach means that the information obtained from the data is limited to the defined meta-paths, and it is impossible to explore new knowledge from the data automatically, limiting the model's capability. Metapath2Vec performs random walks for heterogeneous networks based on specific meta-paths, which limits the exploration capability of the model. HAN models the heterogeneous networks based on different meta-paths and then combines the attention mechanism. Combining different meta-path semantic information enables the model to select important meta-path information but still has to select meta-paths manually. HetSANN can map neighborhood nodes to low-dimensional space by using transformation matrices of different node types, and the attention mechanism aggregates low-dimensional neighborhood node representations to directly handle heterogeneous graphs, which increases model parameters if the number of neighboring node types in each layer is too large, causing model challenging to train. Since GAT ignores the edge or node types, Simple-HGN considers the edge type embedding in the attention mechanism of GAT to directly analyze the heterogeneous network, incorporating the heterogeneity information in the network and improving the model performance. In unimodal models, HetSANN and Simple-HGN perform well in node classification, clustering, and visualization tasks.

The methods mentioned above are all applicable to unimodal graphs. With the development of information networks, network node characteristics can be represented by multiple modalities. The heterogeneous graphs with different modal information form a multimodal heterogeneous graph. This graph contains rich semantic features, but semantic gaps exist between different modalities. We designed MHGAT to reduce redundant information by grouping connected nodes according to their edge types and to aggregate them to form edge-level vectors. Then, the attention mechanism is used to aggregate the edge-level vectors adaptively. The proposed model aggregates different modalities through a modal-level attention mechanism that adaptively perceives the modal weights of different tasks. MHGAT further incorporates residual connections to prevent over-smoothing and obtain higher-order semantic information through dense connections. MHGAT can directly handle multimodal heterogeneous graphs and fuse multimodal information. However, if the modal variety is further increased, causing the model parameters to increase. This is our next research problem.

MHGAT performs best in node classification, clustering, and visualization tasks. Community detection [32] intends to find potentially relevant connections from a complex network of graph structures and to form division sets from aggregations of nodes with the same characteristics. Since MHGAT has an excellent performance in both node clustering and visualization tasks, it has good graph representation learning ability and can better distinguish different types of nodes. It can be applied to community detection tasks. The model captures the interactions between individual elements represented by the nodes. In particular, in medical applications [23], nodes can represent individuals in a potentially large group (patients or healthy controls) accompanied by a set of features, while the edges of the graph contain associations between subjects intuitively. This representation allows for the simultaneous inclusion of a large amount of imaging and non-imaging information and individual subject features in the disease classification task. Each patient is a node, and the node features are image features obtained from the CNN, with the connection relationships based on non-images. Due to the performance of MHGAT on the node classification task, MHGAT can be used to train graphs where a portion of patients(nodes) are disease labeled to predict whether an individual has a disease or not. Since MHGAT has excellent network representation learning capability, it can also be applied to many practical scenarios to improve network data mining performance.

5 Conclusion

We explore multimodal heterogeneous graph mining by proposing the multimodal heterogeneous graph attentional network, MHGAT. Instead of using meta-paths, MHGAT employs edge-level aggregation to obtain heterogeneous information adaptively. Meanwhile, MHGAT uses the attention mechanism to fuse information from different modalities. To acquire higher-order semantics, we combine the residual connection and the dense connection to obtain higher-order neighborhood information. MHGAT performs better than the baselines on node classification, clustering, and visualization tasks.

At present, there is little work on multimodal heterogeneous graph mining, and the proposal of MHGAT provides a new idea for multimodal heterogeneous network mining. MHGAT does not use meta-paths, which provides a solution for adaptive heterogeneous graph information. In the meantime, the model can acquire higher-order semantics, providing an explorable method for deepening graph convolution network.

In the future, we will extend MHGAT to multimodal recommendation tasks to establish a more accurate recommendation model. It will explore user interests at more granular levels, providing a deeper understanding of user preferences.

Acknowledgments This work was supported in part by Zhejiang NSF Grants No. LY20F020009, China NSF Grant No. 61572266 and No.61602133, Ningbo NSF Grants No.202003N4086, as well as programs sponsored by K.C. Wong Magna Fund in Ningbo University. (Corresponding author: Yihong Dong.)

Data availibility The datasets that support the findings of this study are available in https://github.com/jiaxiangen/MHGAT.

Declarations

Conflict of interest The authors have no competing interests.

References

1. Abu-El-Haija S, Perozzi B, Kapoor A, Alipourfard N, Lerman K, Harutyunyan H, Ver Steeg G, Galstyan A(2019) Mixhop: higher-order graph convolutional architectures via sparsified neighborhood mixing. In: 36th international conference on machine learning, ICML 2019, vol. 2019, pp. 32–41
2. Baltrusaitis T, Ahuja C, Morency LP (2019) Multimodal machine learning: a survey and taxonomy. IEEE Trans Pattern Anal Mach Intell 41(2):423–443
3. Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35(8):1798–1828
4. Chen J, Zhang A (2020) Hgmf: heterogeneous graph-based fusion for multimodal data with incompleteness. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 1295–1305
5. Chen Y, Yuan J, You Q, Luo J (2018) Twitter sentiment analysis via bi-sense emoji embedding and attention-based lstm. In: Proceedings of the 26th ACM international conference on multimedia, MM '18, Association for Computing Machinery, New York, pp. 117-125,
6. Defferrard M, Bresson X, Vandergheynst Pierre (2016) Convolutional neural networks on graphs with fast localized spectral filtering. Adv Neural Inf Process Syst 59:3844–3852
7. Dong Y, Chawla NV, Swami A (2017) Metapath2vec: scalable representation learning for heterogeneous networks. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, Part F 1296:135–144
8. Fu TY, Lee WC, Lei Z (2017) HIN2Vec: Explore meta-paths in heterogeneous information networks for representation learning. In: International conference on information and knowledge management, proceedings, vol. Part F1318, pp 1797–1806
9. Grover A, Leskovec J (2016) Node2vec: scalable feature learning for networks. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, vol. 13–17, pp. 855–864
10. Hamilton WL, Ying R, Leskovec J (2017) Inductive representation learning on large graphs. Adv Neural Inf Process Syst 2017:1025–1035
11. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
12. Hong H, Guo H, Lin Y, Yang X, Li Z, Ye J (2020) An attention-based graph neural network for heterogeneous structural learning. In: AAAI, pp. 4132–4139
13. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings - 30th IEEE conference on computer vision and pattern recognition, CVPR 2017, vol. 2017, pp. 2261–2269
14. Jing Y, Yang Y, Wang X, Song M, Tao D (2021) Amalgamating knowledge from heterogeneous graph neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 15709–15718
15. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th international conference on learning representations
16. Kiros R, Salakhutdinov R, Zemel RS (2014) Unifying visual-semantic embeddings with multimodal neural language models. CoRR arXiv:1411.2539
17. Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: Proceedings of the 31st international conference on international conference on machine learning - Vol. 32, pp. 1188-1196, AAAA.org
18. Li Q, Han Z, Wu XM (2018) Deeper insights into graph convolutional networks for semi-supervised learning. In: Proceedings of the 32nd AAAI conference on artificial intelligence, AAAI '18, pp. 3538–3545, AAAI Press
19. Luan S, Zhao M, Chang XW, Precup D (2019) Break the ceiling: stronger multi-scale deep graph convolutional networks. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. Curran Associates Inc, Red Hook, pp 10945–10955
20. Luo L, Fang Y, Cao X, Zhang X, Zhang W (2021) Detecting communities from heterogeneous graphs: a context path-based graph neural network model. In: Proceedings of the 30th ACM international conference on information & knowledge management, pp. 1170–1180
21. Lv Q, Ding M, Liu Q, Chen Y, Feng W, He S, Zhou C, Jiang J, Dong Y, Tang J (2021) Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining, pp. 1150–1160
22. Mroueh Y, Marcheret E, Goel V (2015) Deep multimodal learning for audio-visual speech recognition. In: 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp. 2130–2134
23. Parisot S, Ktena Sofia I, Ferrante E, Lee M, Guerrero R, Glocker B, Rueckert D (2018) Disease prediction using graph convolutional networks: application to autism spectrum disorder and alzheimer's disease. Med Image Anal 48:117–130
24. Perozzi B, Al-Rfou R, Skiena S (2014) DeepWalk: Online learning of social representations. in: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, pp. 701–710
25. Ragesh R, Sellamanickam S, Iyer A, Bairi R, Lingam V (2021) Hetegcn: heterogeneous graph convolutional networks for text classification. In: Proceedings of the 14th ACM international conference on web search and data mining, pp. 860–868

26. Sak H, Senior A, Rao K, Beaufays F (2015) Fast and accurate recurrent neural network acoustic models for speech recognition. In: Proceedings of the annual conference of the international speech communication association, INTERSPEECH, vol. 2015, pp. 1468–1472
27. Shi C, Hu B, Zhao WX, Yu PS (2019) Heterogeneous information network embedding for recommendation. IEEE Trans Knowl Data Eng 31(2):357–370
28. Shi C, Li Y, Zhang J, Sun Y, Yu PS (2017) A survey of heterogeneous information network analysis. IEEE Trans Knowl Data Eng 29(1):17–37
29. Silberer C, Lapata M (2014) Learning grounded meaning representations with autoencoders. In: 52nd annual meeting of the association for computational linguistics, ACL 2014 - proceedings of the conference, vol. 1, pp. 721–732
30. Song K, Zhang Y, Wang X, Zuo J (2019) Representation learning for heterogeneous network with multiple link attributes. In: ACM unternational conference proceeding series, pp. 1358–1368
31. Srivastava N, Salakhutdinov R (2014) Multimodal learning with deep Boltzmann machines. J Mach Learn Res 15:2949–2980
32. Su X, Xue S, Liu F, Wu J, Yang J, Zhou C, Hu W, Paris C, Nepal S, Jin D, Sheng QZ, Yu PS (2022) A comprehensive survey on community detection with deep learning. IEEE Trans Neural Netw Learn Syst pp. 1–21. https://ieeexplore.ieee.org/document/9732192
33. Tang Q, Qu J, Wang M, Zhang M, Yan M, Mei J (2015) LINE: large-scale information network embedding. In: Proceedings of the 24th international conference on World Wide Web, International World Wide Web Conferences Steering Committee, pp. 1067–1077
34. van der Maaten L, Hinton G (2008) Visualizing data using t-sne. J Mach Learn Res 9(86):2579–2605
35. Veličković P, Casanova A, Liò P, Cucurull G, Romero A, Bengio Y (2018) Graph attention networks. In: 6th international conference on learning representations, ICLR 2018 - conference track proceedings, arXiv: 1710.10903
36. Wang J, Jun H, Qian S, Fang Q, Changsheng X (2020) Multimodal graph convolutional networks for high quality content recognition. Neurocomputing 412:42–51
37. Wang X, Ji H, Cui P, Yu P, Shi C, Wang B, Ye Y (2019) Heterogeneous graph attention network. In: The web conference 2019 - proceedings of the World Wide Web conference, WWW 2019, pp. 2022–2032
38. Wang X, Zhu M, Bo D, Cui P, Shi C, Pei J (2020) AM-GCN: adaptive multi-channel graph convolutional networks. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1243–1253
39. Wei Y, He X, Wang X, Hong R, Nie L, Chua TS (2019) MMGCN: multi-modal graph convolution network for personalized recommendation of micro-video. In: MM 2019 - proceedings of the 27th ACM international conference on multimedia, pp. 1437–1445
40. Wu J, Li B, Qin Y, Ni W, Zhang H, Fu R, Sun Y (2021) A multiscale graph convolutional network for change detection in homogeneous and heterogeneous remote sensing images. Int J Appl Earth Obs Geoinf 105:102615
41. Xie Y, Yao C, Gong M, Chen C, Qin AK (2020) Graph convolutional networks with multi-level coarsening for graph classification. Knowl-Based Syst 194:105578
42. Yao L, Mao C, Luo Y (2019) Graph convolutional networks for text classification. In: Proceedings of the AAAI conference on artificial intelligence vol. 33, pp. 7370–7377
43. You J, Ying R, Leskovec J (2019) Position-aware graph neural networks. In: Kamalika C, Ruslan S (eds) Proceedings of the 36th international conference on machine learning, vol. 97 of Proceedings of machine learning research, Long Beach, California, USA, pp. 7134–7143
44. Zhang J, Lu CT, Zhou M, Xie S, Chang Y, Yu Philip S (2016) HEER: heterogeneous graph embedding for emerging relation detection from news. In: Proceedings - 2016 IEEE international conference on big data, big data 2016, IEEE, pp. 803–812
45. Zhang Z, Cui P, Zhu W (2020) Deep learning on graphs: a survey. IEEE Trans Knowl Data Eng 34:249–270
46. Zhou J, Huang JX, Hu QV, He L (2020) SK-GCN: modeling syntax and knowledge via graph convolutional network for aspect-level sentiment classification. Knowl-Based Syst 205:106292

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.