Learning Bi-Typed Multi-Relational Heterogeneous Graph Via Dual Hierarchical Attention Networks

Yu Zhao, Shaopeng Wei, Huaming Du, Xingyan Chen, Qing Li, Member, IEEE, Fuzhen Zhuang, Member, IEEE, Ji Liu, Member, IEEE, and Gang Kou

Abstract—Bi-typed multi-relational heterogeneous graph (BMHG) is one of the most common graphs in practice, for example, academic networks, e-commerce user behavior graph and enterprise knowledge graph. It is a critical and challenging problem on how to learn the numerical representation for each node to characterize subtle structures. However, most previous studies treat all node relations in BMHG as the same class of relation without distinguishing the different characteristics between the intra-type relations and inter-type relations of the bi-typed nodes, causing the loss of significant structure information. To address this issue, we propose a novel Dual Hierarchical Attention Networks (DHAN) based on the bi-typed multi-relational heterogeneous graphs to learn comprehensive node representations with the intra-type and inter-type attention-based encoder under a hierarchical mechanism. Specifically, the former encoder aggregates information from the same type of nodes, while the latter aggregates node representations from its different types of neighbors. Moreover, to sufficiently model node multi-relational information in BMHG, we adopt a newly proposed hierarchical mechanism. By doing so, the proposed dual hierarchical attention operations enable our model to fully capture the complex structures of the bi-typed multi-relational heterogeneous graphs. Experimental results on various tasks against the state-of-the-arts sufficiently confirm the capability of DHAN in learning node representations on the BMHGS.

Index Terms—Bi-typed multi-relational heterogeneous graph, graph learning, dual hierarchical attention networks, GNNs

1 INTRODUCTION

Bi-typed multi-relational heterogeneous graph (BMHG) typically consists of two different types of nodes and multiple intra-type/inter-type relations among them, which are ubiquitous in the real-world scenarios [1], such as academic social networks [2], [3], e-commerce user behavior graph [4], and enterprise knowledge graph [5], [6]. These graphs have rich and valuable heterogeneous information that is worth deep mining. For more clarity, we formally define the BMHG in Definition 1. Without loss of generality, let us take OAG dataset [3] as an example of the BMHG, which consists of two types of nodes, i.e. authors and papers, and multiple relationships, i.e. colleague, cite, is_ordinary_author_of, etc, as shown in Fig. 1.

Definition 1. Bi-typed Multi-relational Heterogeneous Graph. A bi-typed multi-relational heterogeneous graph is defined as a connected graph $BMHG = (V, L, T, R)$. $V$ denotes the node set, and $L$ denotes a link set. They are associated with two functions: (i) a node type mapping function $\phi : V \to T$, $|T| = 2$. $V = \{V_1, V_2\}$, $V_1 \cap V_2 = \emptyset$. Each node $v \in V$ belongs to one particular node type in the node type set $T$: $\phi(v) \in T$. (ii) a link class mapping function $\psi : L \to R$. $\forall l_1, l_2 \in L$, $\psi(l_1) \in R_{\text{intra}}$ and $\psi(l_2) \in R_{\text{inter}}$ denote the node intra-type relationships and the node inter-type relationships, respectively. BMHG has multiple relationships (i.e., $|R_{\text{intra}}| > |T| - 1 > 0$ and $|R_{\text{inter}}| > 1$).

In this article, we focus on how to encode the bi-typed multi-relational heterogeneous graphs, providing an effective and flexible way to use their structural knowledge. The ultimate goal is to pursue perfect low-dimension distributed representations for nodes and relations mainly according to heterogeneous information in the BMHG. The learned results are essential for the inference tasks over graph, such as link prediction [15], [16], node classification [17], [18], node clustering [1] and graph classification [19], [20].

Previous heterogeneous graph learning studies attempt to adopt the advanced Graph Neural Networks (GNNs) to
learn heterogeneous graph while preserving the heterogeneous structures [3], [11], [12]. However, most of the existing methods usually ignore the distinguished characteristics between the node intra-type relations and inter-type relations in the bi-typed multi-relational heterogeneous graphs, which inevitably leads to graph significant structural information loss.

To solve the problem, we propose a novel Dual Hierarchical Attention Networks (DHAN) utilizing the intra-type and inter-type attention-based encoder under a hierarchical mechanism. The former encoder model aggregates intra-node information (Section 3.2), while the latter encoder captures inter-node information (Section 3.3). What's more, to learn comprehensive node representations based on the BMHG, we adopt a newly proposed hierarchical mechanism. Equipped with those modules, the proposed dual hierarchical attention operations endow our model with ability to fully capture the complex structures of the bi-typed multi-relational heterogeneous graphs. The comparison between previous existing methods with our proposed DHAN in terms of nodes heterogeneity and edges heterogeneity is shown in Table 1.

To evaluate the effectiveness of our proposed model, we generate three different kinds of datasets according to the paper citation thresholds, including OAG10Y, OAG20Y and OAG10Y. We conduct extensive experiments on these datasets with author disambiguation and paper classification task against the state-of-the-art methods, which sufficiently demonstrate the better capability of our proposed DHAN in learning node representations in the bi-typed multi-relational heterogeneous graphs.

The contributions of our work are summarized as follows:

- In this article, we focus on embedding the bi-typed multi-relational heterogeneous graphs. To the best of our knowledge, no one attempts to deal with the task before. This paper is expected to further facilitate the bi-typed heterogeneous graph-involved applications, such as academic network mining [12], recommendation system [21], enterprise knowledge graph embedding [22], etc.
- To tackle the bi-typed multi-relational heterogeneous graph learning task, we propose a novel dual hierarchical attention networks (DHAN). Specifically, we equipped DHAN with the intra-type and inter-type attention networks under a newly proposed hierarchical mechanism, which enables the proposed model to sufficiently capture the complex structural knowledge in the BMHG.
- We conduct extensive experiments to evaluate the performance of the proposed model. The results demonstrate the superiority of the proposed model against the SOTA methods for learning node representations on bi-typed multi-relational heterogeneous graphs. The source code and data of this paper can be obtained from: https://github.com/superweisp/DHAN2022.

2 RELATED WORK

2.1 Graph Embedding

Recent years have witnessed a growing interest in developing graph learning algorithms [23] since most real-world data can be represented by graphs conveniently. Classical graph learning methods aim to reduce the dimension of graph data into low-dimensional representations (i.e., graph embedding), such as the linear method PCA [24] and the non-linear method LLE [25]. Inspired by the basic idea from probabilistic language models such as skip-gram [26] and bag-of-words [27], some random walk-based methods are proposed to learn node representations, such as DeepWalk [28] and its advanced extension Node2Vec [23]. Current methods pay attention to random walk on spatio-temporal graphs [29], [30] and its multiscale nature [31]. There are also some matrix factorization-based methods for graph learning tasks [32], [33]. We refer the readers to [34] for more surveys on graph learning literature.

However, the above mentioned methods only consider the structural information of graph, and could not take node attribution into consideration.

2.2 Graph Neural Networks

Graph Neural Networks (GNNs) develop a deep neural network to deal with arbitrary graphs for representation learning [12], [35], [36], [37], [38]. GNNs have been successfully applied to various tasks over graphs [8], [39], such as graph classification [19], [20], link prediction [15], and node classification [17], [18]. The Graph Convolutional Networks (GCNs), as a representative GNN model, generalize convolutional operation on the graph-structured data [9], [40]. Graph Attention Networks (GATs) learn from the underlying graph structure by incorporating attention mechanism into GCNs [40], where the hidden representation of each node is computed by recursively aggregating its local neighbors’ features, and the weighting coefficients are calculated inductively with self-attention strategy [41]. We refer the readers to [35] for more references of GNNs.

Despite the success of the above methods, they are constrained to perform only on homogeneous graphs, which thus could not handle the rich information in heterogeneous graphs.
TABLE 1

Comparison Between Several SOTA Methods and the Proposed Model in Terms of Nodes Heterogeneity and Edges Heterogeneity

| Models                | Graph Heterogeneity |
|-----------------------|---------------------|
| Name                  | Bi-typed | inter-type multi-relations | intra-type multi-relations |
| GCN[7]                | ✓✗✗ ✓     | ✓✗✗ ✓            | ✓✗✗ ✓                |
| GAT[8]                | ✓✗✗ ✓     | ✓✗✗ ✓            | ✓✗✗ ✓                |
| RGCN[9]               | ✓✗✗ ✓     | ✓✗✗ ✓            | ✓✗✗ ✓                |
| GTN[10]               | ✓✗✗ ✓     | ✓✗✗ ✓            | ✓✗✗ ✓                |
| HAN[11]               | ✓✗✗ ✓     | ✓✗✗ ✓            | ✓✗✗ ✓                |
| HetGNN[3]             | ✓✗✗ ✓     | ✓✗✗ ✓            | ✓✗✗ ✓                |
| HGT[12]               | ✓✗✗ ✓     | ✓✗✗ ✓            | ✓✗✗ ✓                |
| HGConv[13]            | ✓✗✗ ✓     | ✓✗✗ ✓            | ✓✗✗ ✓                |
| ie-HGCN[14]           | ✓✗✗ ✓     | ✓✗✗ ✓            | ✓✗✗ ✓                |
| DHAN (Ours)           | ✓✗✗ ✓     | ✓✗✗ ✓            | ✓✗✗ ✓                |

2.3 Heterogeneous Graph Neural Networks

Heterogeneous graphs contain different types of nodes and edges [3], [11], [42], which have rich and valuable heterogeneous information. Heterogeneous graph modeling methods are useful for various tasks, such as short text classification [42], spam review detection [43], conversation generation [44], sentiment analysis [45]. To deal with heterogeneous graphs, Wang et al. [11] proposed heterogeneous graph attention networks (HAN), which mainly concentrate on the different meta-paths. Zhang et al. [3] proposed HetGNN that uses specialized Bi-LSTM to integrate the heterogeneous node attributes and neighbors. Busbridge et al. [46] proposed RGAT by extending non-relational GATs to incorporate relational information, but with poor performance. Hu et al. [12] proposed heterogeneous graph transformer (HGT) to model web-scale heterogeneous graphs, which considers graph heterogeneity, dynamic nature and efficient training for large-scale graph. Jin et al. [47] proposed GIAM to distinguish one-hop and multi-hop meta-paths in the propagation process. Some works also concentrate on special network structure, such as text-rich networks [48] and bipartite graphs [49]. Specifically, previous works utilized matrix-based methods to apply bipartite graphs on graph clustering[49], [50], graph partitioning [51] and graph matching [52]. Nowadays, the researchers model bipartite graphs as low-dimension representations and apply them on more tasks, such as graph generation [53] and recommender system [54].

Despite their success, to the best of our knowledge, no one focuses on bi-typed multi-relational heterogeneous graph learning. Previous methods usually ignore the heterogeneous characteristics of inter-type and inter-type relationships of bi-typed nodes in BMHG. Different from the conventional heterogeneous GNNs, this paper concentrates on the bi-typed heterogeneous graph learning task and attempts to design dual hierarchical graph attention networks to learn comprehensive node representations. Table 1 summarizes the key advantages of our model in terms of modeling graph heterogeneity, compared with a variety of state-of-the-art heterogeneous GNNs models.

3 METHODOLOGY

This section introduces the framework of the overall architecture, as shown in Fig. 2. (1) Node Representation Initialization. We firstly initialize paper node representations through a pre-trained XLNet with their titles. Then we calculate author node representations by averaging their corresponding paper nodes’ representations. (2) Dual Hierarchical Attention Networks (DHAN). The proposed DHAN consists of two sub-modules: intra-type attention-based encoder and inter-type attention-based encoder, which aim to fully capture the structural knowledge of BMHG. To model node multi-relational information in BMHG, we will introduce a newly proposed hierarchical mechanism, as shown in Fig. 3. Next, we gives the analysis of BMHG, and the details of DHAN.

3.1 Analysis of the Properties of BMHG

The properties of BMHG include two aspects: (i) Bi-typed property. Different from previous conventional heterogeneous graph, BMHG contains two types of relationships (i.e., intra-type relationships and inter-type relationships), which describe completely distinct connections between nodes. For example, in academic network, an author can be with several intra-type relationships (i.e., colleague, APA1 and APA2 relations), which describe the social connections of the author, while the inter-type relationships (i.e., is_important_author_of and is_ordinary_author_of) describe the contributions of the author to papers. (ii) Multi-relational property. On the one hand, the importance of each type of relationships is locally heterogeneous [55] with respect to different target nodes. That is to say, different nodes assign unequal weights to same relationships. On the other hand, the importance of different relationships has similarity (i.e., general pattern), which can only be captured from a global view [11]. Thus, considering global pattern avoids local optimal and noisy links.

The motivation of the proposed model is twofold accordingly: (i) To model the bi-typed property of BMHG (i.e., the
distinctions of intra-type relationships and inter-type relationships), we thus utilize Intra-type Attention-based Encoder (see Section 3.2) and Inter-type Attention-based Encoder (see Section 3.3) to model these two distinct types of relationships respectively. (ii) To model the multi-relational property of BMHG, we attempt to take both global weights and local weights into consideration when aggregating relationship semantic information with respect to different target nodes.

3.2 Intra-Type Attention-Based Encoder

The intra-type attention networks aim to learn the node embeddings by aggregating node information from their same type of neighbors, as shown in Fig. 2a. Given a set of nodes with the same type \( V_a \), and a node pair \((v_i, v_j) \) that are connected via node intra-type relationship \( F_k^{\text{intra}} \), we firstly perform transformation based on node type to project original node representation into \( \mathbb{R}^d \) latent space as follow:

\[
H_i^{(a)} = W^{(a)} H_i^{(a)},
\]

where \( W^{(a)} \in \mathbb{R}^{d \times d} \) is a trainable weight matrix related to a corresponding node type. \( H_i^{(a)} \in \mathbb{R}^{|V_a| \times d} \) and \( H^{(a)} \in \mathbb{R}^{|V_a| \times d} \) are the original and transformed node representations, respectively.

For node \( v_i \), different types of intra-type relationships contribute different semantics to its embeddings, and so do different nodes with the same relationship. Hence, we then employ attention mechanism here in node-level and relation-level to hierarchically aggregate signals from the same types of neighbors to target node \( v_i \). We first perform self-attention on the nodes to formulate the importance \( e_{F_k}^{ij} \) of a specific-relation based node pair \((v_i, v_j)\) as follows:

\[
L = -\sum_{i,j,v_i} y_i \log(z_i)
\]
where $\hat{h}_i^\Phi = \text{att}_\text{local}(h_i^\Phi, h_j^\Phi; \Phi_k) = \text{LeakyReLU}(a^\Phi_{ij} \cdot [h_i^\Phi \| h_j^\Phi])$, 
\begin{equation}
\phi_{ij} = \text{att}_\text{node}(h_i^\Phi, h_j^\Phi; \Phi_k) = \text{LeakyReLU}(a^\Phi_{ij} \cdot [h_i^\Phi \| h_j^\Phi]) ,
\end{equation}
where $h_i^\Phi, h_j^\Phi \in \mathbb{R}^{d'}$ are transformed hidden features of the node $v_i$ and $v_j$, respectively. $\| \|$ denotes the concatenate operation. $a^\Phi_{ij} \in \mathbb{R}^{2d'^2 \times 1}$ is the shared node-level attention weight vector under relation $\Phi_k$. LeakyReLU is a nonlinear activation function.

Based on Eq. (2), we calculate the $\phi_{ij}$ for all nodes $v_j \in \mathcal{N}_\text{intra}^\Phi(v_i)$, where $\mathcal{N}_\text{intra}^\Phi(v_i)$ denotes specific relation-based neighbors of $v_i$. To make importance easily comparable across different nodes, we normalize them across all choices of $v_j$ using the softmax function:
\begin{equation}
\phi_{ij} = \text{softmax}(\phi_{ij}) = \frac{\exp(\phi_{ij})}{\sum_{j \in \mathcal{N}_\text{intra}^\Phi(v_i)} \exp(\phi_{ij})},
\end{equation}
where $\mathcal{N}_\text{intra}^\Phi(v_i)$ denotes relation-specific aggregated information for node $v_i$.

We apply the the softmax function to make relation importance comparable within inter-type relations. The representation of node $v_i$ which contains global and local information.

Then, the embedding $h_i^\Phi$ of node $v_i$ under given relation $\Phi_k$ is calculated by aggregating its intra-type neighbors’ projected representations with the corresponding coefficients as follows:
\begin{equation}
h_i^\Phi = \text{LeakyReLU} \left( \text{Norm}_\Phi \left( \sum_{v_j \in \mathcal{N}_\text{intra}^\Phi(v_i)} a^\Phi_{ij} \cdot h_j^\Phi \right) \right),
\end{equation}
where $\text{Norm}_\Phi$ denotes relation-specific layer normalization operation. Since the attention coefficient $a^\Phi_{ij}$ is computed for a particular relationship, $h_i^\Phi$ is semantic-specific and capable of capturing one kind of semantic information.

To learn more comprehensive node representations, we fuse different relation-specific aggregated information of nodes. Different from previous methods that either consider global weights [11] or local weights [13] of relationships, we take advantage of both of the two factors in relation-level attention, considering both the heterogeneity with regard to different nodes and the common information that a type of relation has among all nodes. First, we calculate the local importance $\gamma_{ij}^\Phi$ of relation $\Phi_k$ with respect to node $v_i$ as follows:
\begin{equation}
\gamma_{ij}^\Phi = q^\top (h_i^\Phi \| h_j^\Phi),
\end{equation}
where $q \in \mathbb{R}^{2d'}$ is a trainable parameter. Then, we implement the softmax function to normalize the node-relation specific local importance across different relations.
\begin{equation}
\beta_{ij}^\Phi = \text{softmax}(\gamma_{ij}^\Phi) = \frac{\exp(\gamma_{ij}^\Phi)}{\sum_{i \in \mathcal{N}_\text{intra}^\Phi(v_i)} \exp(\gamma_{ij}^\Phi)},
\end{equation}
where $\beta_{ij}^\Phi$ indicates how important relation $\Phi_k$ is for node $v_i$, which measures local importance of intra-relation $\Phi_k$.

Second, to prevent model from local optimum and alleviate effects of noisy links, we design a relation global importance $\beta_{v_i}^\Phi$, which denotes how important intra-type $\Phi_k$ is for all nodes $v_i \in \mathcal{V}_v$. Finally, as shown in 3, we fuse different relation-specific aggregated information of nodes in both local and global view, as follow:
\begin{equation}
\begin{aligned}
\mathbf{z}_i &= \sum_{\Phi_k \in \mathcal{R}_\text{intra}} \left( t \beta_{0i}^\Phi + (1 - t) \beta_{v_i}^\Phi \right) \cdot h_i^\Phi,
\end{aligned}
\end{equation}
where $\mathbf{z}_i \in \mathbb{R}^{d'}$ is the learned representation of node $v_i$, which contains global and local information. $h_i^\Phi$ denotes aggregated information for node $v_i$ under intra-type relation $\Phi_k$. $\beta_{0i}^\Phi$ and $t$ can be learned from training.

### 3.3 Inter-Type Attention-Based Encoder

Different from the above intra-type attention networks, the inter-type attention-based encoder aims to deal with the interaction between different types of nodes. We set $v_i^{(1)} \in \mathcal{V}_1$ and $v_i^{(2)} \in \mathcal{V}_2$. $\mathbf{z}_i^{(1)}$, $\mathbf{z}_i^{(2)}$ are the learned representations of the node $v_i^{(1)}$ and $v_i^{(2)}$ by intra-type attention networks, respectively.

We calculate the node-level importance $c_{ij}^\Phi$ for all nodes $v_i \in \mathcal{N}_\text{inter}^\Phi(v_i)$, where $\mathcal{N}_\text{inter}^\Phi(v_i)$ denotes the neighbors of node $v_i$ under specific inter-relation $\Phi_m$. We normalize them across all choices of $v_j$ using the softmax function:
\begin{equation}
\begin{aligned}
c_{ij}^\Phi &= \text{att}_\text{node}(\mathbf{z}_i, \mathbf{z}_j; \Phi_m) \\
&= \text{LeakyReLU}(a_{ij\Phi} \cdot \langle W^{(1)} \mathbf{z}_i \| W^{(2)} \mathbf{z}_j \rangle),
\end{aligned}
\end{equation}
\begin{equation}
\gamma_{ij}^\Phi = \text{softmax}(c_{ij}^\Phi) = \frac{\exp(c_{ij}^\Phi)}{\sum_{i \in \mathcal{N}_\text{inter}^\Phi(v_i)} \exp(c_{ik}^\Phi)},
\end{equation}
where $W^{(1)}, W^{(2)} \in \mathbb{R}^{d \times d'}$ are two type-specific matrices to map their features $\mathbf{z}_i, \mathbf{z}_j$ into a common space. $a_{ij\Phi} \in \mathbb{R}^{d'^2}$ is a trainable weight vector. Then, as shown in Fig. 2, the relation representation of node $v_i^{(1)}$ can be aggregated by its different types of neighbors’ representations with the corresponding coefficients as follows:
\begin{equation}
\mathbf{z}_i^\Phi = \text{LeakyReLU} \left( \text{Norm}_\Phi \left( \sum_{v_j \in \mathcal{N}_\text{inter}^\Phi(v_i)} \gamma_{ij}^\Phi \cdot W^{(2)} \mathbf{z}_j \right) \right),
\end{equation}
where $\text{Norm}_\Phi$ indicates layer normalization operation related to the inter-type relation.

Similar to the above hierarchical attention, all relation representations are fused to get the final representations:
\begin{equation}
\begin{aligned}
f_i^\Phi &= q^\top (\mathbf{z}_i \| \mathbf{z}_i^\Phi),
\end{aligned}
\end{equation}
\begin{equation}
\begin{aligned}
e_i^\Phi &= \text{softmax}(f_i^\Phi) = \frac{\exp(f_i^\Phi)}{\sum_{i \in \mathcal{V}} \exp(f_i^\Phi)},
\end{aligned}
\end{equation}
where $q \in \mathbb{R}^{2d'}$ is a projection vector. $f_i^\Phi$ denotes the importance of relation embedding $\mathbf{z}_i^\Phi$ related to node $v_i^{(1)}$. We apply the the softmax function to make relation importance comparable within inter-type relations. The representation $u_i$ of node $v_i$ is obtained by fusing these relation-specific representations.




\[
\mathbf{u}_i = \sum_{\mathbf{w}_m \in \mathcal{R}_{\text{inter}}} c_{\mathbf{w}_m} \cdot \mathbf{z}_i^\mathbf{w}_m,
\]

where \(\mathcal{R}_{\text{inter}}\) indicates the set of relations among different types of nodes (i.e., node inter-type links).

In inter-type hierarchical attention, the aggregation of different nodes' embedding is seamlessly integrated, and they are mingled and interactively affected each other, as shown in Fig. 2b.

### 3.4 Weighted Residual Connection

For both intra-type encoder and inter-type encoder, we use weighted residual connection and layer normalization to alleviate over-smooth in practice.

\[
\mathbf{z}_i = \text{Norm}(\lambda \sigma(\mathbf{z}_i) + (1 - \lambda) \mathbf{h}_i),
\]

\[
\mathbf{u}_i = \text{Norm}(\tilde{\lambda} \sigma(\mathbf{u}_i) + (1 - \tilde{\lambda}) \mathbf{z}_i),
\]

where \(\lambda\) and \(\tilde{\lambda}\) are hyperparameters.

### 3.5 Optimization

For node classification tasks, such as paper-venue classification and paper-field classification in OAG dataset, we predict labels based on nodes' final representations. For link prediction task (i.e., author disambiguation), we predict whether connections exist based on node pairs' similarities by element-wise product of representations.

We train our model by minimizing the cross-entropy loss. Inspired by [56], we promote the training efficiency by adding Temperature \(T\) in the learning.

\[
\mathcal{L} = - \sum_{i \in \mathcal{Y}_L} y_i \log \left( \frac{\hat{y}_i}{T} \right),
\]

where \(\mathcal{Y}_L\) is the set of labeled nodes. \(y_i\) and \(\hat{y}_i\) are the ground truth and the predicted label for node \(i\), respectively.

The time complexity of DHAN can be determined as: \(O((|\mathcal{R}| \cdot |\mathcal{V}| + |\mathcal{L}|)D^2)\), where \(|\mathcal{R}|\) denotes the total number of intra-type and inter-type relationships, \(|\mathcal{V}|\) denotes the total number of the two types of nodes, \(|\mathcal{L}|\) denotes the total edge number and the \(D\) denotes the dimension of the representation. The linear complexity with respect to node number ensures the scalability of the model that it can be applied on larger scale datasets.

### 4 Experiments

#### 4.1 Experimental Settings

##### 4.1.1 Datasets

We generate three different kinds of datasets by extracting different sub-graphs from the popular Open Academic Graph (OAG) dataset [3] with various paper citation thresholds, including \(OAG1Y, OAG2Y\) and \(OAG10Y\). In \(OAG1Y\), we only retain the papers which are cited more than once a year. In \(OAG2Y\) and \(OAG10Y\), we loose the time constraints to 2 years and 10 years, respectively. They contain two types of nodes (i.e., authors and papers), and several preliminary links including (author, colleague, author), (author, is_important_author_of, paper), (author, is_ordinary_author_of, paper), (paper, cite, paper). Note that the “important” authorship indicates an author is the first or second author of a paper, and the “ordinary” authorship indicates an author is not the important author of a paper. The basic statistics of all datasets are included in Table 2. The intra-type relations of authors include: colleague, APA1 and APA2. APA1 and APA2 indicate the co-authorship of important authors and ordinary authors, respectively. The intra-type relations of papers include: cite, rev_cite, is_same_venue_of, is_same_field_of. The inter-type relation between author and paper includes: is_important_author_of and is_ordinary_author_of.

##### 4.1.2 Baselines

To demonstrate the effectiveness of our proposed model DHAN, we compare it with three types of SOTA baselines: (1) the homogeneous graph neural networks which do not consider multi-relationships between nodes, such as GCN, GAT; (2) the heterogeneous graph neural networks which
take different relationships into consideration, such as RGCN, HGT; (3) the heterogeneous networks which implement a hierarchical mechanism to aggregate different kinds of relations in graphs, such as HAN, HGConv.

**Homogeneous models:**

- Graph Convolutional Networks (GCN) [7], [57]: a popular model which simply averages neighboring nodes’ representations in aggregation.
- Graph Attention Networks (GAT) [8]: a recent model which takes attention mechanism to align different weights to neighbors during the information aggregating process.

**Heterogeneous models:**

- Relational Graph Convolutional Networks (RGCN) [9]: an advanced extension of GCN, which takes relation information into consideration by giving different weights for difference relationships.
- Heterogeneous graph neural network (HetGNN) [3]: a multi-modal heterogeneous graph model which utilizes Bi-LSTM to process multi-moding information, then applies attention mechanism in heterogeneous information fusing.
- Graph Transformer Networks (GTN) [10]: a novel heterogeneous graph neural network based on GCN which updates adjacent matrix of different relations during training process.
- Heterogeneous Graph Transformer (HGT) [12]: a state-of-the-art model which implements on heterogeneous graph with different types of nodes and multiple relations.

**Hierarchical models:**

- Heterogeneous Graph Attention Network (HAN) [11]: one of the earliest model which implements hierarchical attention on graph neural network based on meta-path.
- Heterogeneous Graph Convolution (HGConv) [13]: an efficient model which utilizes hierarchical mechanism based on different node types and different relations.
- Interpretable and efficient Heterogeneous Graph Convolutional Network (ie-HGCN) [14]: a SOTA model which firstly implements object-level aggregation and then aggregates type-level information based on different meta-paths.

### 4.1.3 Model Setting and Training Details

We implement DHAN with PyTorch and PyTorch Geometric (PyG). We use a pre-trained XLNet [58] to initialize the paper nodes’ representations. Then the author nodes’ initial representations are aggregated by averaging their published papers’ embeddings. We set the dropout rate of DHAN among [0.1, 0.2, 0.3, 0.4, 0.5] and the temperature T from [0.01, 0.05, 0.1, 1, 1.5, 10]. The $\ell_2$ regularization weight is set from [1e-4, 1e-3, 1e-2, 1e-1]. For the paper field L1 task (PF_L1), we add one more weighted residual connection in inter-type aggregation process without adding any new parameters. All models are trained with AdamW optimizer with the Cosine Annealing Learning Rate Scheduler. For all the baseline models and DHAN, we use 128 hidden dimension. For each model, we run 200 epochs and choose the best which has higher NDCG and lower loss compared with former training processes on validation datasets in order to alleviate the overfitting problem. To obtain the experimental results of all baselines, we run official codes provided by the original papers. Finally, we report the results of each model on the testing datasets.

### 4.1.4 Task Target

Based on the properties of BMHG, we conduct the following experiments to analyze the proposed model’s ability in capturing complex structure information among them. Specifically, we perform node classification, link prediction and node clustering to verify the model’s general representing ability with regard to both supervised and unsupervised aspects. To illustrate that the learned node embedding can capture the subtle difference among different intra-nodes, we conduct node embedding visualization. Besides, we conduct the interpretability experiment to analyze the proposed model’s ability to capture structure information among inter-nodes. Moreover, we perform variants analysis and ablation study to demonstrate the sub-modules’ efficiency in learning both intra- and inter-structure information among nodes. Finally, we conduct parameter analysis to study the proposed model’s performance under different hyper-parameters.

### 4.2 Classification and Link Prediction

#### 4.2.1 Evaluation Protocol

We evaluate our model on three tasks, including author disambiguation (AD), paper-venue (PV), paper-field in L1 level (PF_L1) classification and paper-field in L2 level (PF_L2) classification. In the datasets, the fields of papers are divided into several hierarchical levels (such as Operating system/file system), and lower level means more detailed categories. In other words, L2 (such as ‘file system’) has much more categories than L1 (such as operating system). The author disambiguation task could be treated as a link prediction task which aims to predict the possible link between the same name and their associated papers. Both of the paper-venue and paper-field classifications are multi-classification problem. In paper-venue classification, each paper belongs to only one venue, while each paper may belong to several fields of L1 level and L2 level in paper-field classification tasks. We adopt accuracy (ACC), Normalized Discounted Cumulative Gain (NDCG) and Mean Reciprocal Rank (MRR) as evaluation metrics.

#### 4.2.2 Results and Analysis

The experimental results of the proposed model and SOTA baselines are reported in Table 3. We can observe from Table 3 that our proposed DHAN outperforms all the baselines on all tasks across most of metrics on all datasets. For instance, our model improves the ACC, NDCG and MRR of author disambiguation on OAG1Y from 0.6477 to 0.8343, 0.5394 to 0.7828, and 0.3479 to 0.6799 respectively comparing to the state-of-the-art model ie-HGCN, which confirms
the capability of DHAN in learning bi-typed multi-relational heterogeneous graph.

**Analysis.** (1) Compared with homogeneous GNNs, i.e. GCN and GAT, DHAN achieves significant and consistent performance, which indicates that our proposed model can sufficiently capture the heterogeneous information from the data. (2) Compared with heterogeneous GNNs (i.e., RGCN, HetGNN, GTN and HGT), the proposed model DHAN outperforms all baselines in link prediction tasks on all datasets and indicators. This is mainly because our model is specially designed for bi-typed multi-relational graphs. Hence, it can sufficiently utilize interactions between two types of nodes, which can not be well captured by general heterogeneous graph neural networks. Besides, the proposed model also achieves comparable results in classification tasks on most of datasets and indicators. The observation confirms that our model is able to distinguish different relations delicately by utilizing the hierarchical mechanism. (3) Compared with the conventional hierarchical attention model HGConv and ie-HGConv, our model performs better on all tasks in all datasets. Our model takes advantage of the two typical hierarchical models by fusing relation global information and local information. To be more specific, HAN proposed to aggregate different types of relation information with same global information. To be more specific, HAN proposed to aggregate different types of relation information with same global information. To be more specific, HAN proposed to aggregate different types of relation information with same global information. To be more specific, HAN proposed to aggregate different types of relation information with same global information.

**4.3 Node Clustering**

We conduct node clustering based on the paper-venue task on three datasets. Here, we first get node representations via feed forward of each GNN. We them apply K-Means to implement node clustering and evaluate the performance using NMI and ARI based on their ground truth and predicted categories. Since the results tend to be affected by initial centroids, to make performance more stable, we repeat the former process 10 times and report average results in Table 4. Experiments results show that our model outperforms all baselines, e.g. on OAG1Y, DHAN outperforms the

| Datasets | Tasks | Metrics | GCN [7] | GAT [8] | RGCN [9] | HAN [11] | HetGNN [3] | GTN [10] | HGT [12] | HGConv [13] | ie-HGConv [14] | DHAN |
|----------|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| OAG1Y    | PV    | NDCG    | 0.2611  | 0.2750  | 0.2693  | 0.2880  | 0.2375  | 0.2680  | 0.2970  | 0.2885  | 0.2465  | 0.2995  |
|          |       | MRR     | 0.1295  | 0.1391  | 0.1335  | 0.1508  | 0.1031  | 0.1300  | 0.1623  | 0.1502  | 0.1069  | 0.1643  |
|          | AD    | NDCG    | 0.7180  | 0.7271  | 0.7492  | 0.7227  | 0.6587  | 0.7408  | 0.7515  | 0.7476  | 0.7304  | 0.7532  |
|          |       | MRR     | 0.6892  | 0.6905  | 0.7220  | 0.6916  | 0.6189  | 0.7088  | 0.7169  | 0.7179  | 0.6996  | 0.7213  |
|          | AD    | NDCG    | 0.3598  | 0.3678  | 0.4191  | 0.3817  | 0.3059  | 0.3910  | 0.4502  | 0.4209  | 0.3297  | 0.4512  |
|          |       | MRR     | 0.3156  | 0.3300  | 0.4311  | 0.3593  | 0.2183  | 0.3725  | 0.4958  | 0.4403  | 0.2528  | 0.4960  |
| OAG2Y    | PV    | NDCG    | 0.2604  | 0.2780  | 0.2739  | 0.2899  | 0.2465  | 0.2569  | 0.2947  | 0.2862  | 0.1828  | 0.2969  |
|          |       | MRR     | 0.1282  | 0.1445  | 0.1376  | 0.1553  | 0.1137  | 0.1200  | 0.1616  | 0.1496  | 0.0502  | 0.1629  |
|          | AD    | NDCG    | 0.7076  | 0.7271  | 0.7410  | 0.7384  | 0.6614  | 0.7284  | 0.7455  | 0.7438  | 0.7195  | 0.7520  |
|          |       | MRR     | 0.6838  | 0.6985  | 0.7131  | 0.7069  | 0.6282  | 0.6905  | 0.7075  | 0.7139  | 0.6861  | 0.7177  |
|          | AD    | NDCG    | 0.3651  | 0.3737  | 0.4275  | 0.3882  | 0.3075  | 0.4000  | 0.4544  | 0.4265  | 0.3383  | 0.4558  |
|          |       | MRR     | 0.3226  | 0.3427  | 0.4429  | 0.3629  | 0.2179  | 0.3955  | 0.4916  | 0.4391  | 0.2694  | 0.4925  |
| OAG10Y   | PV    | NDCG    | 0.2604  | 0.2718  | 0.2739  | 0.2598  | 0.2515  | 0.2317  | 0.2801  | 0.2655  | 0.2405  | 0.2816  |
|          |       | MRR     | 0.1282  | 0.1399  | 0.1376  | 0.1225  | 0.1196  | 0.0971  | 0.1445  | 0.1287  | 0.1047  | 0.1476  |
|          | AD    | NDCG    | 0.7219  | 0.7300  | 0.7520  | 0.7169  | 0.6837  | 0.7339  | 0.7550  | 0.7489  | 0.7222  | 0.7530  |
|          |       | MRR     | 0.6902  | 0.6950  | 0.7266  | 0.6834  | 0.6554  | 0.6953  | 0.7196  | 0.7188  | 0.6899  | 0.7197  |
|          | AD    | NDCG    | 0.3595  | 0.3641  | 0.4205  | 0.3768  | 0.3125  | 0.3892  | 0.3877  | 0.4189  | 0.3342  | 0.4556  |
|          |       | MRR     | 0.3081  | 0.3184  | 0.4196  | 0.3385  | 0.2274  | 0.3679  | 0.3735  | 0.4214  | 0.2559  | 0.4868  |

**Evaluation of different methods on three datasets.**

The Classification and Link Prediction Results Table 3.
SOTA model ie-HGCN with a margin as large as 0.0277 on ARI. The results demonstrate the superiority of the learned node representations.

### 4.4 Ablation Study

To evaluate the contribution of different model components of DHAN, we conduct an ablation study. We generate variants of DHAN by adjusting the use of its model components and comparing their performance on three tasks on OAG1Y. The three ablated variants are as follows: (1) DHAN w/o dual operation, which does not distinguish the node intra-type and inter-type relation, and only takes one hierarchical attention. (2) DHAN w/o hierarchical architecture, which deletes hierarchical architecture in both intra-type and inter-type encoders. (3) DHAN w/o global attention, which deletes the relation global attention.

Fig. 4 shows the results of the variants on all three datasets, from which we can observe that removing either dual operation or hierarchical architecture will lead to performance decreasing. Specifically, the proposed model DHAN significantly outperforms DHAN w/o dual operation, which confirms the benefits of the dual mechanism. Thus, we highlight the importance of designing a specific model architecture on the bi-typed graphs rather than a general heterogeneous graph model. Compared with DHAN w/o global attention and DHAN w/o hierarchical architecture, we can find that fusing both global information and local information makes a great contribution to the performance of DHAN. Moreover, we could also observe that DHAN w/o global attention always performs better than DHAN w/o hierarchical architecture, which is in line with the fact that DHAN w/o hierarchical architecture is also a simplified version of DHAN w/o global attention removing local attention mechanism.

### 4.5 Visualization

To make a more intuitive comparison, we project the representations of paper nodes into two-dimensional space by t-SNE. The node representations are learned on OAG1Y based on PF_L1 tasks. We randomly choose two fields that no papers belongs to both. The color indicates the publishing field of the papers in Fig. 5. The less mixed areas the better. We can observe that our model DHAN performs best in visualization as there are more distinct boundaries and fewer mixed nodes. Besides, we also find that those hierarchical heterogeneous models (i.e., HAN and ie-HGCN), perform better than general heterogeneous graph models (i.e., HetGNN and GTN).

### 4.6 Variant Analysis

We conduct variant analysis of DHAN on OAG1Y with four tasks to show the effectiveness of its architecture. (1) DAHN-RGCN substitutes the proposed hierarchical attention mechanism with RGCN and keeps model structure unchanged. (2) Inverted Architecture firstly implements inter-type hierarchical aggregation and then applies intra-type hierarchical aggregation. (3) Parallel Architecture conducts intra-type and inter-type hierarchical aggregation simultaneously and concatenates the updated representation of two types of nodes respectively. The results are shown in Fig. 6, from which we can observe that all the variants perform worse than DHAN. DHAN-RGCN utilizes RGCN rather than our hierarchical module to aggregate different types of relation information, which thus leads to a performance decrease. The proposed DHAN performs better than both Inverted Architecture and Parallel Architecture, which demonstrates our model structure is a more efficient architecture (i.e., first conducting intra-type relation aggregation then implementing inter-type relation aggregation).

### 4.7 Interpretability of the Hierarchical Attention

To demonstrate the interpretability of DHAN, we show the learned attention scores in Fig. 7. The global attention is the learned weight for different relations, and the average attention is calculated as the average of the sum of global attention score and heterogeneous attention score of all nodes. Here, we show the results of PF_L2 task and AD task on OAG10Y.
Specifically, we can observe from Fig. 7a that the learned global attention score of relation cite and rev_cite gain more weight than other relations in PF_L2 task. This is in line with the fact that those papers which are either cited by or cite target paper contribute much more than other related papers to the target paper while performing paper field tasks. Besides, the “is_important_author_of” and “is_ordinary_author_of” relationships obtain more significant weight than the “is_same_venue_of” and “is_same_field_of” relationships, which is also in line with intuition. Moreover, the “is_important_author_of” relationship acquires a bit more considerable weight than “is_ordinary_author_of”, which confirms the interpretability of our model again. A similar conclusion on AD task is shown in 7(b). However, different from Fig. 7a, the global attention weight of “is_important_author_of” is the largest one among all relations, which denotes that papers with same important author have much more influence than other related papers in the author disambiguation task. This is mainly because that the author disambiguation task cares more about relations between authors and papers, which is also in line with our intuition. Above all, we can find that the average attention score of each relation is significantly different from global attention weight. Actually, in the PF_L2 task, the average attention score of the “is_important_author_of” relation and the corresponding standard variance is 0.5682 and 0.0960. The former two facts demonstrate the necessity of combining both global information and local information for information aggregation.

4.8 Parameter Analysis

The hyper-parameters play an important role in model performance, and one of the most essential hyper-parameters is the dimension of representations. We conduct parameter analysis in the PF_L2 and AD task on the OAG1Y dataset. For PF_L2 task, the average attention score of the “cite” relation and corresponding standard variance are 0.3293 and 0.0124. In AD task, the average attention score of the “is_important_author_of” relation and the corresponding standard variance is 0.5682 and 0.0960. The former two facts demonstrate the necessity of combining both global information and local information for information aggregation.
Fig. 8. Parameter sensitivity of DHAN on PF_L2 and AD task with different dimensions in OAG1Y.

The results are shown in Fig. 8, from which we can observe that the proposed model reaches its best performance when the dimension of output representation is set as 128. Specifically, the performance first rises with the dimension increasing and then reaches its optimal state since the model needs larger dimension to embody rich information. After that, the performance decreases as a result of overfitting.

Moreover, we conduct analysis of the layer number in the AD task on the OAG1Y dataset. We find that the proposed model performs best with 3 layers of intra-type attention-based encoder and 4 layers of inter-type attention-based encoder. The result demonstrates that the proposed model need several layers to aggregate high-order neighbors’ information, while too many layers lead to performance degeneration, which is mainly because of over-smoothing.

5 CONCLUSION AND FUTURE WORK

In this article, we focus on how to learn node efficient representations on bi-typed multi-relational heterogeneous graph. To this end, we propose a novel Dual Hierarchical Attention Networks (DHAN). To the best of our knowledge, we are the first attempt to deal with this task. Specifically, DHAN contains intra-type and inter-type attention-based encoders which enables DHAN to sufficiently leverage not only the node intra-type neighboring information but also the inter-type neighboring information in BMHG. Moreover, to sufficiently model node multi-relational information in BMHG, we adopt a newly proposed hierarchical mechanism, which takes both global and local importance of relationships into consideration. By doing so, the proposed dual hierarchical attention operations enable our model to fully capture the complex structures of the BMHGs. We conduct extensive experiments on various tasks against the state-of-the-arts, which sufficiently confirms the capability of DHAN in learning node comprehensive representations in BMHGs. Interesting future work directions include generalizing DHAN to other BMHG-based applications.

REFERENCES

[1] X. Wang, D. Bo, C. Shi, S. Fan, Y. Ye, and P. S. Yu, “A survey on heterogeneous graph embedding: Methods, techniques, applications and sources,” 2020, arXiv:2011.14867.

[2] G. Wang, Q. Hu, and P. S. Yu, “Influence and similarity on heterogeneous networks,” in Proc. Conf. Inf. Knowl. Manage., vol. 12, pp. 1462–1466.

[3] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, “Heterogeneous graph neural network,” in Proc. Int. Conf. Knowl. Discov. Data Mining, 2019, pp. 793–799.

[4] X. Niu et al., “A dual heterogeneous graph attention network to improve long-tail performance for shop search in e-commerce,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2020, pp. 3405–3415.

[5] V. Y. Guleva, M. V. Skvorcova, and A. V. Boukhannovsky, “Using multiplex networks for banking systems dynamics modelling,” Procedia Comput. Sci., vol. 66, pp. 257–266, 2015.

[6] S. Li, Y. Liu, and C. Wu, “Systemic risk in bank-firm multiplex networks,” Finance Res. Lett., vol. 33, 2020, Art. no. 101232.

[7] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in Proc. Int. Conf. Learn. Representations, 2017.

[8] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” in Proc. Int. Conf. Learn. Representations, 2018.

[9] M. Schlüchtermann, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, “Modeling relational data with graph convolutional networks,” in Proc. Eur. Semantic Web Conf., 2018, pp. 593–607.

[10] Y. Ziyu, L. Jianxin, Z. Wei, C. Jiangtao, and W. Qun, “Interpretable and efficient heterogeneous graph convolutional network,” IEEE Trans. Knowl. Data Eng., early access, Aug. 06, 2021, doi: 10.1109/TKDE.2021.3101356.

[11] X. Wang et al., “Heterogeneous graph attention network,” in Proc. Int. World Wide Web Conf., 2019, pp. 2022–2032.

[12] Z. Hu, Y. Dong, K. Wang, and Y. Sun, “Heterogeneous graph transformer,” in Proc. Int. World Wide Web Conf., 2020, pp. 2704–2710.

[13] Y. Le, S. Lelei, D. Bowen, L. Chuanren, L. Weifeng, and X. Hui, “Hybrid micro/macro level convolution for heterogeneous graph learning,” 2020, arXiv:2012.14722.

[14] J. Feng, M. Huang, Y. Yang, and X. Zhu, “Gake: Graph aware knowledge embedding,” 2016, arXiv:1604.06511.

[15] J. B. Lee, R. Rossi, X. Kong, S. Kim, E. Koh, and A. Rao, “Graph convolutional networks with motif-based attention,” in Proc. Conf. Inf. Knowl. Manage., 2019, pp. 499–508.

[16] K. Zhang, Y. Zhi, J. Wang, and J. Zhang, “Adaptive structural fingerprints for graph attention networks,” in Proc. Int. Conf. Learn. Representations, 2020.

[17] J. Wu, J. He, and J. Xu, “Demo-Net: Degree-specific graph neural networks for node and graph classification,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2019, pp. 406–415.

[18] J. B. Lee, R. Rossi, and X. Kong, “Graph classification using structural attention,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2018, pp. 1–9.

[19] W. Guo et al., “Dual graph enhanced embedding neural network for network prediction,” 2021, arXiv:2106.00314.

[20] S. Gualdi, G. Cimini, K. Primicerio, D. Clemente, and D. Challet, “Statistically validated network of portfolio overlaps and systemic risk,” Sci. Rep., vol. 6, no. 1, pp. 1–14, 2016.

[21] A. Grover and J. Leskovec, “Node2vec: Scalable feature learning for networks,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2016, pp. 855–864.

[22] I. T. Jolliffe and I. Cadima, “Principal component analysis: A review and recent developments,” Philos. Trans. Roy. Soc. A Math., Phys. Eng. Sci., vol. 374, no. 2065, 2016, Art. no. 20150202.

[23] T. S. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” Science, vol. 290, no. 5500, pp. 2323–2326, 2000.

[24] D. Guthrie, B. Allison, W. Liu, L. Guthrie, and Y. Wilks, “A closer look at skip-gram modelling,” in Proc. Int. Conf. Lang. Res. Evol., vol. 6, pp. 1222–1225, 2006.

[25] Y. Zhang, R. Jin, and Z.-H. Zhou, “Understanding bag-of-words model: A statistical framework,” Int. J. Mach. Learn. Cybern., vol. 1, no. 1/4, pp. 43–52, 2010.

[26] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2014, pp. 701–710.

[27] S. Servedio, M. Fox, P. Biswal, and H. Zha, “Dyrep: Learning representations over dynamic graphs,” in Proc. Int. Conf. Learn. Representations, 2019.

[28] A. Jabri, A. Owens, and A. Efros, “Space-time correspondence as a contrastive random walk,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 19 545–19 560.
Z. Huang, A. Silva, and A. Singh, “A broader picture of random-walk based graph embedding,” in Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Mining, 2021, pp. 685–695.

M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, “Asymmetric transitivity preserving graph embedding,” in Proc. 22nd ACM SIGKDD Conf. Knowl. Discov. Data Mining, 2016, pp. 1105–1114.

A. Tsitsulin, M. Munkhoeva, D. Motkin, P. Karras, I. Oseledets, and E. Müller, “Frede: Anytime graph embeddings,” Proc. VLDB Endowment, vol. 14, no. 6, pp. 1102–1110, 2021.

F. Chen, Y.-C. Wang, B. Wang, and C.-C. J. Kuo, “Graph representation learning: A survey,” APSIPA Trans. Signal Inf. Process., vol. 9, 2020.

J. Zhou, G. Cui, Z. Zhang, C. Y. Z. L. L. C. Li, and M. Sun, “Graph neural networks: A review of methods and applications,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020.

M. Zhang and Y. Chen, “Link prediction based on graph neural networks,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2018, pp. 5165–5175.

Y. Hou et al., “Measuring and improving the use of graph information in graph neural networks,” in Proc. Int. Conf. Learn. Representations, 2020.

D. Wang, J. Liu, Y. Zheng, X. Qiu, and X. Huang, “Heterogeneous graph neural networks for extractive document summarization,” in Proc. Assoc. Comput. Linguistics, 2020.

J. B. Lee, R. A. Rossi, S. Kim, N. K. Ahmed, and E. Koh, “Attention models in graphs: A survey,” ACM Trans. Knowl. Discov. Data, vol. 13, no. 6, pp. 1–25, 2018.

S. Vashishth, S. Sanval, V. Nitin, and P. Talukdar, “Composition-based multi-relational graph convolutional networks,” in Proc. Int. Conf. Learn. Representations, 2020.

K. K. Thekamparampil, C. Wang, S. Oh, and L.-J. Li, “Attention-based graph neural network for semi-supervised learning,” 2018, arXiv:1803.03735v1.

L. Hu, T. Yang, C. Shi, H. Ji, and X. Li, “Heterogeneous graph attention networks for semi-supervised short text classification,” in Proc. Proc. Conf. Empir. Methods Natural Lang. Process., 2019, pp. 4821–4830.

A. Li, Z. Qiu, R. Liu, Y. Yang, and D. Li, “Spam review detection with graph convolutional networks,” in Proc. Conf. Inf. Knowl. Manage., 2019, pp. 2703–2711.

H. Zhou, T. Yang, M. Huang, H. Zhao, J. Xu, and X. Zhu, “Commonsense knowledge aware conversation generation with graph attention,” in Proc. Int. Conf. Comput. Intel., 2018, pp. 1–7.

K. Wang, W. Shen, Y. Yang, X. Quan, and R. Wang, “Relational graph convolutional network for aspect-based sentiment analysis,” in Proc. Assoc. Comput. Linguistics, 2020, pp. 3229–3238.

D. Busbridge, D. Sherburn, P. Cavallo, and N. Y. Hammerla, “Relational graph attention networks,” in 2019, arXiv:1904.05811.

D. Jin, Z. Yu, D. He, C. Yang, P. Yu, and J. Han, “GCN for HIN via implicit utilization of attention and meta-paths,” IEEE Trans. Knowl. Data Eng., early access, Nov. 25, 2021, doi: 10.1109/TKDE.2021.3130712.

Z. Yu et al., “AS-GCN: Adaptive semantic architecture of graph convolutional networks for text-rich networks,” in Proc. IEEE Int. Conf. Des. Mater., 2021, pp. 837–846.

Y. Li, F. Nie, H. Huang, and J. Huang, “Large-scale multi-view spectral clustering via bipartite graph,” in Proc. Conf. Assoc. Advance. Intell., 2015, pp. 2750–2756.

F. Nie, X. Wang, C. Deng, and H. Huang, “Learning a structured optimal bipartite graph for co-clustering,” in Proc. Conf. Int. Conf. Neural Inf. Process. Syst., 2017, pp. 4132–4141.

Z. Li, X.-M. Wu, and S.-F. Chang, “Segmentation using superpixels: A bipartite graph partitioning approach,” in Proc. IEEE Conf. Comput.Vis. Pattern Recognit., 2012, pp. 789–796.

K. Riesen and H. Bunke, “Approximate graph edit distance computation by means of bipartite graph matching,” Image Vis. Comput., vol. 27, no. 7, pp. 950–959, 2009.

R. Li, S. Zhang, B. Wan, and X. He, “Bipartite graph network with adaptive message passing for unbiased scene graph generation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 110–11119.

C. Li, K. Jia, D. Shen, C.-J. R. Shi, and H. Yang, “Hierarchical representation learning for bipartite graphs,” in Proc. Int. Joint Conf. Intell. Artif., 2019, pp. 2873–2879.

L. Yu, L. Sun, B. Du, C. Liu, W. Lv, and H. Xiong, “Heterogeneous graph representation learning with relation awareness,” IEEE Trans. Knowl. Data Eng., early access, Mar. 17, 2022, doi: 10.1109/TKDE.2022.3160268.

W. Jiancan et al., “Self-supervised graph learning for recommendation,” in Proc. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval, 2021, pp. 726–735.

M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, “Simple and deep graph convolutional networks,” in Proc. Int. Conf. Mach. Lear., 2020, pp. 1725–1734.

Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, “XLNet: Generalized autoregressive pretraining for language understanding,” 2019, arXiv:1906.08237.

L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE,” J. Mach. Learn. Res., vol. 9, no. 11, pp. 2579–2605, 2008.
Fuzhen Zhuang (Member, IEEE) received the PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences. He is currently a full professor with the Institute of Artificial Intelligence, Beihang University, Beijing 100191, China. His research interests include machine learning and data mining, including transfer learning, multi-task learning, multi-view learning and recommendation systems. He has published more than 100 papers in the prestigious refereed conferences and journals, such as KDD, WWW, SIGIR, ICDE, IJCAI, AAAI, EMNLP, Nature Communications, IEEE Transactions on Knowledge and Data Engineering, ACM Transactions on Knowledge Discovery from Data, IEEE Transactions on Cybernetics, IEEE Transactions on Neural Networks and Learning Systems, ACM Transactions on Intelligent Systems and Technology, etc.

Ji Liu (Member, IEEE) received the BS degree from the University of Science and Technology of China, Hefei, China, in 2005, the master’s degree from Arizona State University, Tempe, AZ, USA, in 2010, and the PhD degree from the University of Wisconsin–Madison, Madison, WI, USA, in 2014. He is currently an assistant professor of computer science, electrical and computer engineering with the Goergen Institute for Data Science, University of Rochester (UR), Rochester, NY, USA, where he created the Machine Learning and Optimization Group. He has authored more than 70 papers in top journals and conferences including Journal of Machine Learning Research, IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Neural Networks and Learning Systems, Transactions on Knowledge Discovery from Data, NIPS, ICML, SIGKDD, ICCV, and CVPR. He was a recipient of the Award of Best Paper Honorable Mention at SIGKDD 2010, the Award of Best Student Paper Award at UAI 2015, and the IBM Faculty Award. He is named one MIT technology review’s 35 innovators under 35 in China.

Gang Kou received the BS degree from the Department of Physics, Tsinghua University, China, the master’s degree from the Department of Computer Science, University of Nebraska at Omaha, and the PhD degree in information technology from the College of Information Science & Technology, Univ. of Nebraska at Omaha. He is a distinguished professor of Chang Jiang Scholars Program with the Southwestern University of Finance and Economics, managing editor of International Journal of Information Technology & Decision Making (SCI) and managing editor-in-chief of Financial Innovation (SSCI). He is also editors for other journals, such as: Decision Support Systems, and European Journal of Operational Research. Previously, he was a professor with the School of Management and Economics, University of Electronic Science and Technology of China, and a research scientist with Thomson Co., R & D. He has published more than 100 papers in various peer-reviewed journals. Gang Kou’s h-index is 57 and his papers have been cited for more than 10000 times. He is listed as the highly cited researcher by Clarivate Analytics (Web of Science).