This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221099

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*, Gang Kou *Member, IEEE*

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 challenge 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 **D**ual **H**ierarchical **A**ttention **N**etworks (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 Figure 1.

Definition 1. ***Bi-typed Multi-relational Heterogeneous Graph***. *A bi-typed multi-relational heterogeneous graph is defined as a connected graph* $\mathcal{BMHG} = (\mathcal{V}, \mathcal{L}, \mathcal{T}, \mathcal{R})$. *$\mathcal{V}$ denotes the node set, and $\mathcal{L}$ denotes a link set. They are associated with two functions: (i) a node type mapping function* $\varphi : \mathcal{V} \rightarrow \mathcal{T}$, $|\mathcal{T}| = 2$. $\mathcal{V} = \{\mathcal{V}_1, \mathcal{V}_2\}$, $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$. *Each node* $v \in \mathcal{V}$ *belongs to one particular node type in the node type set* $\mathcal{T} : \varphi(v) \in \mathcal{T}$.

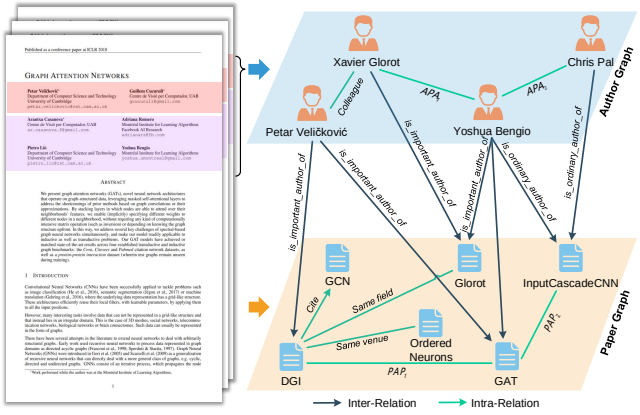


Fig. 1. A toy example of bi-typed multi-relational heterogeneous graph (BMHG) in the academic networks. This graph consists of two types of nodes, i.e. *authors* and *papers*. Their links could be divided into two classes: (1) node intra-type relations, such as "*colleague*" between authors, "*cite*" and "*same venue*" between papers; (2) node inter-type relations, such as "*is_important_author_of*", "*is_ordinary_author_of*". Table 2 reports a detailed statistics of the graph.

(ii) a link class mapping function $\psi : \mathcal{L} \rightarrow \mathcal{R}$. $\forall l_1, l_2 \in \mathcal{L}$, $\psi(l_1) \in \mathcal{R}_{intra}$ *and* $\psi(l_2) \in \mathcal{R}_{inter}$ *denote the node intra-type relationships and the node inter-type relationships, respectively.* $\mathcal{BMHG}$ *has multiple relationships (i.e.,* $|\mathcal{R}_{inter}| > |\mathcal{T}| - 1 > 0$ *and* $|\mathcal{R}_{intra}| > 1$*).*

In this paper, 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

- *Y. Zhao, Q. Li and X. Chen are with Fintech Innovation Center, Financial Intelligence and Financial Engineering Key Laboratory of Sichuan Province, Institute of Digital Economy and Interdisciplinary Science Innovation, Southwestern University of Finance and Economics, China. E-mail: zhaoyu@swufe.edu.cn*
- *S. Wei, H. Du and G. Kou are with School of Business Administration, Southwestern University of Finance and Economics, China.*
- *F. Zhuang is with Institute of Artificial Intelligence, Beihang University, Beijing, China, and with Zhongguancun Laboratory, Beijing, China. Email:zhuangfuzhen@buaa.edu.cn*
- *J. Liu is with Kuaishou Technology, USA. E-mail:ji.liu.uwisc@gmail.com*
- *G. Kou (E-mail: kougang@swufe.edu.cn) is the corresponding author.*

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221099

TABLE 1
Comparison between several SOTA methods and the proposed model in terms of nodes heterogeneity and edges heterogeneity.

Models		Graph Heterogeneity		
Name	**Main ideas**	**Bi-typed** $\lvert\mathcal{T}\rvert = 2$	**inter-type multi-relations** $\lvert\mathcal{R}_{\text{inter}}\rvert > 1$	**intra-type multi-relations** $\lvert\mathcal{R}_{\text{intra}}\rvert > 1$
GCN [7]	• Average message passing	✗	✗	✗
GAT [8]	• Attention based message passing	✗	✗	✗
RGCN [9]	• Multi-relations • Hierarchical weighting message passing	✗	✗	✓
GTN [10]	• Multi-relations • Self-adaption weighting message passing	✗	✓	✗
HAN [11]	• Meta-path relations • Hierarchical attention based message passing	✗	✓	✗
HetGNN [3]	• Heterogeneous features of nodes • Attention based message passing	✓	✗	✗
HGT [12]	• Self-attention based message passing • Node-balanced graph sampling • Time encoding	✗	✓	✓
HGConv [13]	• Multi-relations • Hierarchical attention based message passing	✗	✓	✗
ie-HGCN [14]	• Meta-path based relations • Hierarchical attention based message passing	✗	✓	✗
DHAN (Ours)	• Distinguish inter-type relationship and intra-type relationship • A newly proposed global-local hierarchical mechanism	✓	✓	✓

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 **D**ual **H**ierarchical **A**ttention **N**etworks (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 popular Open Academic Graph (OAG) [3] with various paper citation thresholds, including *OAG1Y*, *OAG2Y* 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 paper, 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.

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 task, 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 Figure 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 submodules: 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 Figure 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* relationships), 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

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221099

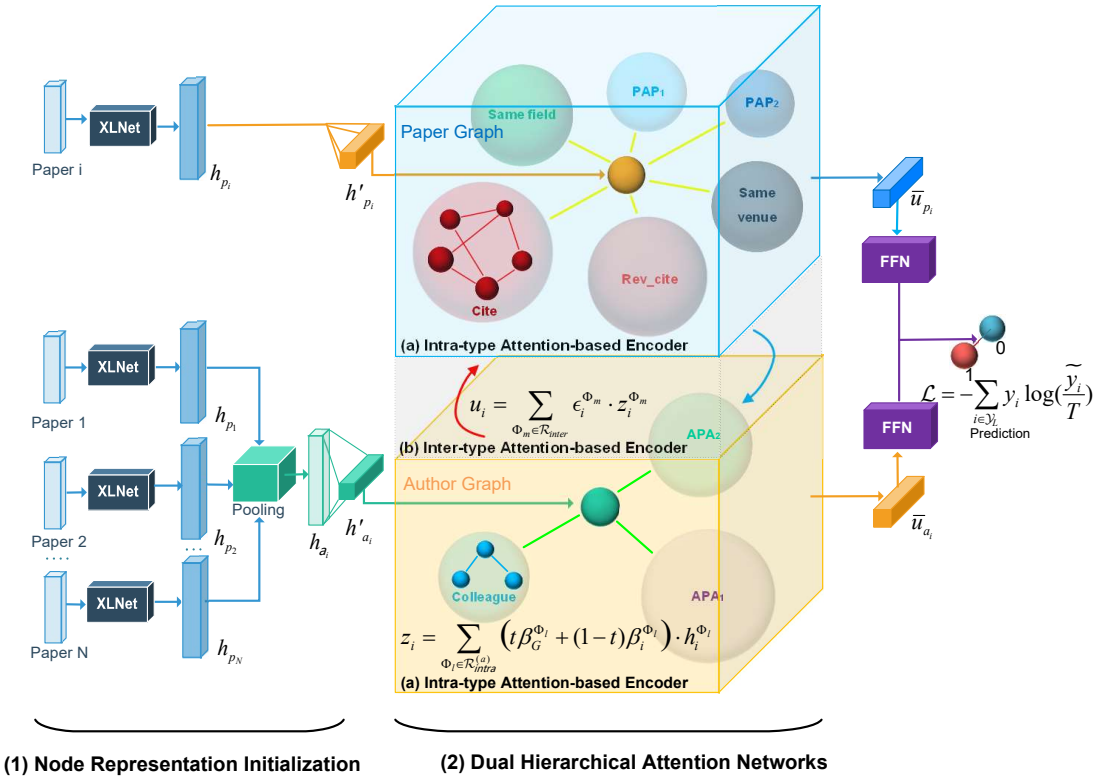


Fig. 2. The overall architecture of the proposed method. (1) **Node Representation Initialization** We utilize a pretrained XLNet to retrieval papers' representations from their titles. Authors' representations are calculated by averaging their published papers' information. (2) **Dual Hierarchical Attention Networks**: (a) **Intra-type Attention-based Encoder** aims to aggregate different types of intra-type relationships with our novel hierarchical mechanism. For each type of node, the intra-type hierarchical attention module shares the same structure but with different parameters; (b) **Inter-type Attention-based Encoder** is designed for updating information between two types of nodes. Each type of node incorporates their inter-type neighbors' information with different weights according to different relationships and node pairs.

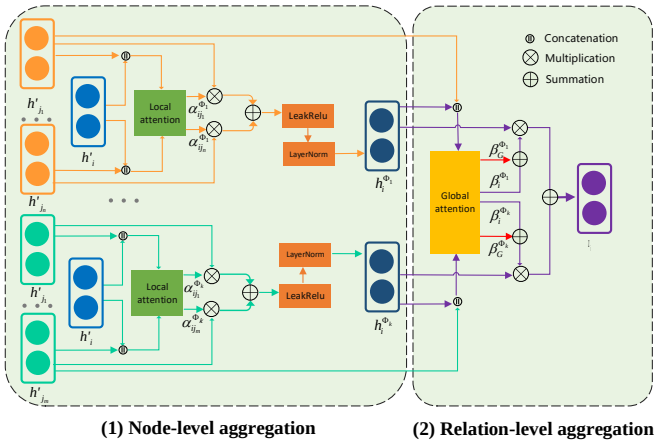


Fig. 3. The proposed hierarchical attention mechanism. (1) Node-level aggregation aims to capture neighbor nodes' importance $\alpha_{ij}^{\Phi_k}$ under a specific type of relation Φ_k. Then relation representations $\mathbf{h}_i^{\Phi_k}$ are aggregated by weighting and summing the target node's Φ_k based neighbors' information. (2) Relation-level aggregation firstly assigns relation importance utilizing attention mechanism based on target node embedding and relation representations. Then relation representations are aggregated to get comprehensive neighbor information $\mathbf{z}_i$ with final importance, which consists of global relation weight $\beta_G^{\Phi_l}$ and local relation weight $\beta_i^{\Phi_k}$.

neighbors, as shown in Figure 2 (a). Given a set of nodes with the same type $\mathcal{V}_a \in \{\mathcal{V}_1, \mathcal{V}_2\}$, and a node pair $(v_i, v_j)(\in \mathcal{V}_a)$ that are connected via node intra-type relationship $\Phi_k \in \mathcal{R}_{\text{intra}}^{(a)}$, we firstly perform transformation based on node type to project original node representation into $\mathbb{R}^d$ latent space as follow:

$$\mathbf{H}'^{(a)} = \mathbf{W}^{(a)}\mathbf{H}^{(a)}, \tag{1}$$

where $\mathbf{W}^{(a)} \in \mathbb{R}^{d \times d'}$ is a trainable weight matrix related to a corresponding node type. $\mathbf{H}^{(a)} \in \mathbb{R}^{|\mathcal{V}_a| \times d}$ and $\mathbf{H}'^{(a)} \in \mathbb{R}^{|\mathcal{V}_a| \times d'}$ 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 $e_{ij}^{\Phi_k}$ of a specific-relation based node pair (v_i, v_j) as follows:

$$e_{ij}^{\Phi_k} = \text{att}_{\text{local}}(\mathbf{h}'_i, \mathbf{h}'_j; \Phi_k) = \text{LeakyRelu}(\mathbf{a}_{\Phi_k}^{\top} \cdot [\mathbf{h}'_i \| \mathbf{h}'_j]), \tag{2}$$

where $\mathbf{h}'_i \in \mathbb{R}^{d'}, \mathbf{h}'_j \in \mathbb{R}^{d'}$ are transformed hidden features of the node v_i and v_j, respectively. $\|$ denotes the concatenate operation. $\mathbf{a}_{\Phi_k}^{\top} \in \mathbb{R}^{2d' \times 1}$ is the shared node-level attention weight vector under relation Φ_k. LeakyReLU is a nonlinearity activation function.

Based on Eq. (2), we calculate the $e_{ij}^{\Phi_k}$ for all nodes $v_j \in \mathcal{N}_{\text{intra}}^{\Phi_k}(v_i)$, where $\mathcal{N}_{\text{intra}}^{\Phi_k}(v_i)$ 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:

$$\alpha_{ij}^{\Phi_k} = \text{softmax}_j(e_{ij}^{\Phi_k}) = \frac{\exp\left(e_{ij}^{\Phi_k}\right)}{\sum\limits_{v_p \in \mathcal{N}_{\text{intra}}^{\Phi_k}(v_i)} \exp\left(e_{ip}^{\Phi_k}\right)}, \tag{3}$$

Then, the embedding $\mathbf{h}_i^{\Phi_k}$ of node v_i under given relation Φ_k is calculated by aggregating its intra-type neighbors' projected representations with the corresponding coefficients as follows:

$$\mathbf{h}_i^{\Phi_k} = \text{LeakyRelu}\Big(\text{Norm}_{\Phi_k}\Big(\sum_{v_j \in \mathcal{N}_{\text{intra}}^{\Phi_k}(v_i)} \alpha_{ij}^{\Phi_k} \cdot \mathbf{h}_j'\Big)\Big), \tag{4}$$

where Norm_{Φ_k} denotes relation-specific layer normalization operation. Since the attention coefficient $\alpha_{ij}^{\Phi_k}$ is computed for a particular relationship, $\mathbf{h}_i^{\Phi_k}$ 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. Firstly, we calculate the local importance $g_i^{\phi_k}$ of relation Φ_k with respect to node v_i as follows:

$$g_i^{\Phi_k} = \mathbf{q}^\top \Big(\mathbf{h}_i' \| \mathbf{h}_i^{\Phi_k}\Big), \tag{5}$$

where $\mathbf{q} \in \mathbb{R}^{2d' \times 1}$ is a trainable parameter. Then, we implement the softmax function to normalize the node-relation specific local importance across different relations.

$$\beta_i^{\Phi_k} = \text{softmax}_k(g_i^{\Phi_k}) = \frac{\exp\left(g_i^{\Phi_k}\right)}{\sum\limits_{\Phi_l \in \mathcal{R}_{\text{intra}}^{(a)}} \exp\left(g_i^{\Phi_l}\right)}, \tag{6}$$

where $\beta_i^{\Phi_k}$ indicates how important relation Φ_k is for node v_i, which measures local importance of intra-relation Φ_k. Secondly, to prevent model from local optimum and alleviate effects of noisy links, we design a relation global importance $\beta_G^{\Phi_l}$, which denotes how important intra-type Φ_l is for all nodes $v_i \in \mathcal{V}_a$. Finally, as shown in 3, we fuse different relation-specific aggregated information of nodes in both local and global view, as follow:

$$\mathbf{z}_i = \sum_{\Phi_l \in \mathcal{R}_{\text{intra}}^{(a)}} \Big(t\beta_G^{\Phi_l} + (1-t)\beta_i^{\Phi_l}\Big) \cdot \mathbf{h}_i^{\Phi_l}, \tag{7}$$

where $\mathbf{z}_i \in \mathbb{R}^{d'}$ is the learned representation of node v_i, which contains global and local information. $\mathbf{h}_i^{\Phi_l}$ denotes aggregated information for node v_i under intra-type relation Φ_l. t is a smooth parameter to balance the global and local importance of intra-type relation Φ_l. $\beta_G^{\Phi_l}$ 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 $v_j^{(2)} \in \mathcal{V}_2$. $\mathbf{z}_i^{(1)}, \mathbf{z}_j^{(2)}$ are the learned representations of the node $v_i^{(1)}$ and $v_j^{(2)}$ by intra-type attention networks, respectively.

We calculate the node-level importance $c_{ij}^{\Phi_m}$ for all nodes $v_j \in \mathcal{N}_{\text{inter}}^{\Phi_m}(v_i)$, where $\mathcal{N}_{\text{inter}}^{\Phi_m}(v_i)$ denotes the neighbors of node v_i under specific inter-relation Φ_m. We normalize them across all choices of v_j using the softmax function:

$$\begin{aligned} c_{ij}^{\Phi_m} &= att_{node}(\mathbf{z}_i, \mathbf{z}_j; \Phi_m) \\ &= \text{LeakyRelu}(\mathbf{a}_{\Phi_m}^\top \cdot [\mathbf{W}^{(1)}\mathbf{z}_i \| \mathbf{W}^{(2)}\mathbf{z}_j]), \end{aligned} \tag{8}$$

$$\gamma_{ij}^{\Phi_m} = \text{softmax}_j(c_{ij}^{\Phi_m}) = \frac{\exp\left(c_{ij}^{\Phi_m}\right)}{\sum\limits_{v_k \in \mathcal{N}_{\text{inter}}^{\Phi_m}(v_i)} \exp\left(c_{ik}^{\Phi_m}\right)}, \tag{9}$$

where $\mathbf{W}^{(1)}, \mathbf{W}^{(2)} \in \mathbb{R}^{d' \times d'}$ are two type-specific matrices to map their features $\mathbf{z}_i, \mathbf{z}_j$ into a common space. $\mathbf{a}_{\Phi_m} \in \mathbb{R}^{2d'}$ is a trainable weight vector. Then, as shown in Figure 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:

$$\mathbf{z}_i^{\Phi_m} = \text{LeakyRelu}\Big(\text{Norm}_{\Phi_m}\Big(\sum_{v_j \in \mathcal{N}_{\text{inter}}^{\Phi_m}(v_i)} \gamma_{ij}^{\Phi_m} \mathbf{W}^{(2)} \mathbf{z}_j\Big)\Big), \tag{10}$$

where Norm_{Φ_m} 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:

$$f_i^{\Phi_m} = \tilde{\mathbf{q}}^\top \Big(\mathbf{z}_i \| \mathbf{z}_i^{\Phi_m}\Big), \tag{11}$$

$$\epsilon_i^{\Phi_m} = \text{softmax}_m(f_i^{\Phi_m}) = \frac{\exp\left(f_i^{\Phi_m}\right)}{\sum\limits_{\Phi_n \in \mathcal{R}_{\text{inter}}} \exp\left(f_i^{\Phi_n}\right)}, \tag{12}$$

where $\tilde{\mathbf{q}} \in \mathbb{R}^{2d'}$ is a projection vector. $f_i^{\Phi_m}$ denotes the importance of relation embedding $\mathbf{z}_i^{\Phi_m}$ related to node $v_i^{(1)}$. We apply the the softmax function to make relation importance comparable within inter-type relations. The representation $\mathbf{u}_i$ of node v_i us obtained by fusing these relation-specific representations.

$$\mathbf{u}_i = \sum_{\Phi_m \in \mathcal{R}_{\text{inter}}} \epsilon_i^{\Phi_m} \cdot \mathbf{z}_i^{\Phi_m}, \tag{13}$$

where $\mathcal{R}_{inter}$ 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 Figure 2 (b).

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.

$$\bar{\mathbf{z}}_i = \text{Norm}\Big(\lambda\sigma(\mathbf{z}_i) + (1-\lambda)\mathbf{h}_i\Big), \tag{14}$$

$$\bar{\mathbf{u}}_i = \text{Norm}\Big(\tilde{\lambda}\sigma(\mathbf{u}_i) + (1-\tilde{\lambda})\mathbf{z}_i\Big), \tag{15}$$

where λ 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.

$$\mathcal{L} = -\sum_{i \in \mathcal{Y}_L} y_i \log(\frac{\tilde{y_i}}{T}) \,, \tag{16}$$

where $\mathcal{Y}_L$ is the set of labeled nodes. y_i and $\tilde{y_i}$ are the ground truth and the predicted label for node i, respectively.

The time complexity of DHAN can be determined as: $\mathcal{O}\big((|\mathcal{R}|\cdot|\mathcal{V}| + |\mathcal{L}|)D^2\big)$, 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 lager 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*.

TABLE 2
Statistics of the datasets *OAG1Y*, *OAG2Y* and *OAG10Y*, which are extracted from the popular Open Academic Graph [3] with various citation thresholds.

Datasets		*OAG1Y*	*OAG2Y*	*OAG10Y*
Bi-typed nodes	***#Papers***	494,051	825,234	1,564,109
	#Authors	480,575	734,451	1,266,569
Author intra-relations	***#Colleague***	285,393,669	562,821,414	1,400,301,929
	#APA1	369,973	600,344	1,074,851
	#APA2	1,015,964	1,413,447	2,059,826
Paper intra-relations	***#Cite/Rev_Cite***	4,847,142	7,367,512	22,407,910
	#Same Field	160,283,374,629	440,183,678,370	1,548,687,874,807
	#Same Venue	273,272,355	619,484,732	1,929,963,113
	#PAP1	3,022,137	5,966,848	13,450,631
	#PAP2	4,973,945	9,042,142	17,648,847
Author-Paper inter-relations	***#Important author***	800,061	1,306,953	2,372,890
	#Ordinary author	661,250	1,019,506	1,687,184
Training data Period		2000 - 2015		
Validation data Period		2015 - 2016		
Testing data Period		2016 - 2019		

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 ℓ_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-HGCN, 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 importance, which overlooks heterogeneity of different nodes. HGConv aggregates relation information with heterogeneous weight related to different nodes, which neglects common information that a type of relation has among all nodes. In contrast, our model overcomes their limits by incorporating both the merits of relation global information as well as local information. (4) In sum, we believe the better performance is due to the better design of our model. First, DHAN can gain improvements via taking both the node intra-type and inter-type attention into consideration. Second, our model also uses an efficient hierarchical attention mechanism to encode the bi-typed multi-relational heterogeneous graph.

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 out model outperforms all baselines, e.g. on *OAG1Y*, DHAN outperforms the 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.

Figure 4 shows the results of the variants on all three datasets, from which we can observe that removing either dual operation

TABLE 3
Classification and link prediction results.

Evaluation of different methods on three datasets.

Datasets	Tasks	Metrics	GCN [7]	GAT [8]	RGCN [9]	HAN [11]	HetGNN [3]	GTN [10]	HGT [12]	HGConv [13]	ie-HGCN [14]	DHAN
OAG1Y	PV	NDCG	0.2661	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**
	PF_L1	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**
	PF_L2	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**
	AD	NDCG	0.7297	0.7915	0.7820	0.7497	0.6430	0.7403	0.8037	0.7715	0.7539	**0.8222**
		MRR	0.6436	0.7241	0.7120	0.6693	0.5309	0.6567	0.7403	0.6982	0.6749	**0.7651**
		ACC	0.4627	0.5678	0.5610	0.5002	0.3240	0.4800	0.6040	0.5406	0.5106	**0.6394**
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**
	PF_L1	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**
	PF_L2	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**
	AD	NDCG	0.6726	0.7783	0.7841	0.7509	0.6258	0.7167	0.8040	0.7797	0.6718	**0.8332**
		MRR	0.5698	0.7073	0.7147	0.6716	0.5099	0.6260	0.7410	0.7093	0.5688	**0.7796**
		ACC	0.3769	0.5539	0.5584	0.5073	0.2959	0.4330	0.5959	0.5513	0.3599	**0.6554**
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**
	PF_L1	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
	PF_L2	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**
	AD	NDCG	0.6042	0.7201	0.7685	0.7169	0.5712	0.6804	0.7979	0.7659	0.6477	**0.8343**
		MRR	0.4841	0.6326	0.6955	0.6284	0.4430	0.5807	0.7338	0.6923	0.5394	**0.7828**
		ACC	0.2862	0.4645	0.5426	0.4615	0.2476	0.3920	0.5973	0.5471	0.3479	**0.6799**

TABLE 4
Node clustering results.

Datasets	Metrics	GCN [7]	GAT [8]	RGCN [9]	HAN [11]	HetGNN [3]	GTN [10]	HGT [12]	HGConv [13]	ie-HGCN [14]	DHAN
OAG1Y	ARI	0.0340	0.0286	0.0308	0.0337	0.0141	0.0350	0.0636	0.0276	0.0451	**0.0728**
	NMI	0.6566	0.6477	0.6469	0.6541	0.6231	0.6652	0.6746	0.6489	0.6479	**0.6764**
OAG2Y	ARI	0.0156	0.0194	0.0109	0.0225	0.0120	0.0176	0.0134	0.0313	0.0090	**0.0474**
	NMI	0.6577	0.6646	0.6571	0.6666	0.6302	0.6773	0.6740	0.6678	0.4758	**0.7012**
OAG10Y	ARI	0.0152	0.0286	0.0124	0.0079	0.0135	0.0369	0.0318	0.0337	0.0018	**0.0370**
	NMI	0.6422	0.6477	0.6510	0.6449	0.6207	0.6706	0.6727	0.6744	0.6491	**0.6818**

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 [59]. 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 Figure 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

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221099

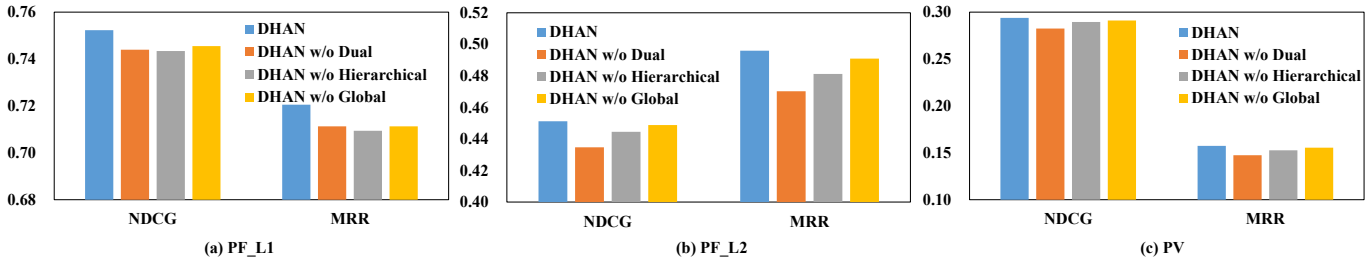


Fig. 4. The ablation study of our model on *OAG1Y*.

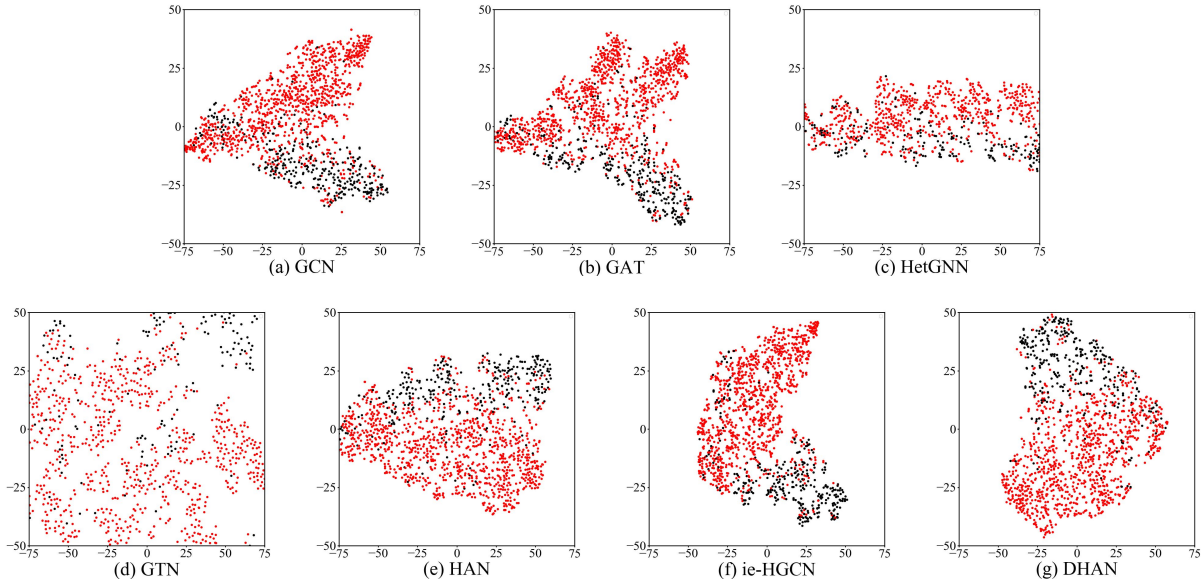


Fig. 5. Visualization of node representation on OAG1Y. Each point indicates a paper and its color indicates its publication field. Less mixed areas mean better performance. We can observe that the proposed DHAN outperforms other models.

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 Figure 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).

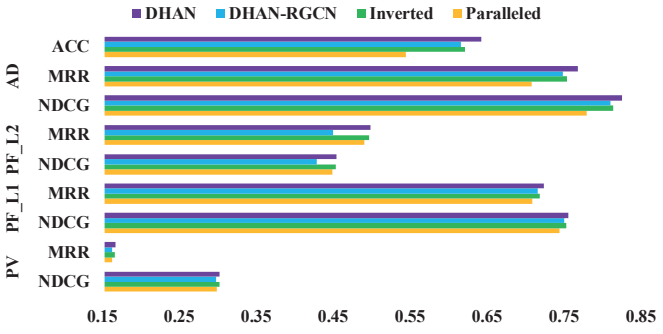


Fig. 6. Variant analsis of DHAN on *OAG1Y*.

4.7 Interpretability of the Hierarchical Attention

To demonstrate the interpretability of DHAN, we show the learned attention scores in Figure 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 Figure 7 (a) 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 Figure 7 (a), 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 *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.

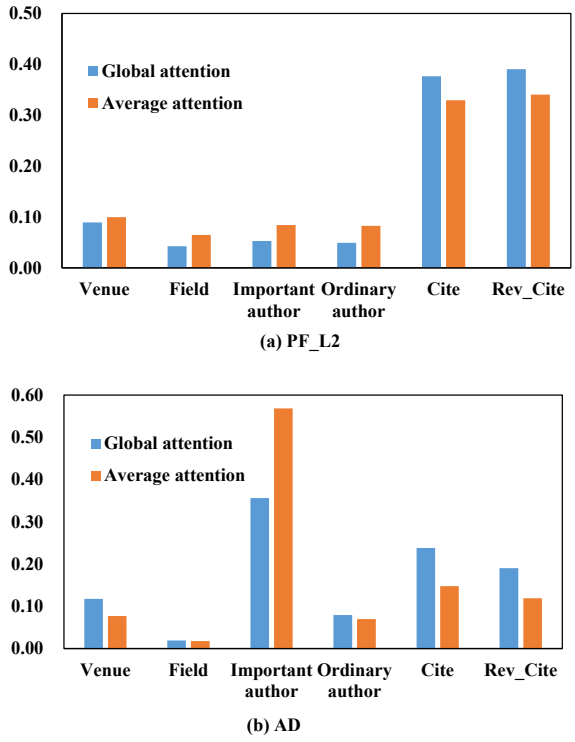


Fig. 7. The presentation of the learned attention scores of DHAN.

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. The results are shown in Figure 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.

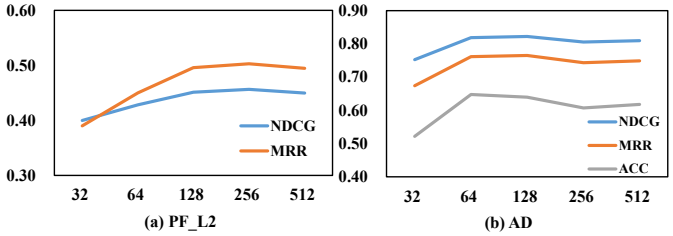


Fig. 8. Parameter sensitivity of DHAN on PF_L2 and AD task with different dimensions in *OAG1Y*.

5 Conclusion and Future Work

In this paper, 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.

Acknowledgments

The research is supported by the National Key Research and Development Program of China under Grant No. 2021ZD0113602, and the National Natural Science Foundation of China under Grant Nos. 61906159, 62176014, 71873108, 62072379, U1811462, 71725001, and 71910107002, and Guanghua Talent Project of Southwestern University of Finance and Economics, Financial Innovation Center, SWUFE (Project NO.FIC2022C0008) and "Double-First Class" International Innovation Project (SYL22GJCX07).

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," *arXiv preprint arXiv:2011.14867*, 2020.

[2] G. Wang, Q. Hu, and P. S. Yu, "Influence and similarity on heterogeneous networks," in *Proceedings of CIKM*, 2012, pp. 1462–1466.

[3] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proceedings of KDD*, 2019, pp. 793–803.

[4] X. Niu, B. Li, C. Li, R. Xiao, H. Sun, H. Deng, and Z. Chen, "A dual heterogeneous graph attention network to improve long-tail performance for shop search in e-commerce," in *Proceedings of SIGKDD*, 2020, pp. 3405–3415.

[5] V. Y. Guleva, M. V. Skvorcova, and A. V. Boukhanovsky, "Using multiplex networks for banking systems dynamics modelling," *Procedia Computer Science*, vol. 66, pp. 257–266, 2015.

[6] S. Li, Y. Liu, and C. Wu, "Systemic risk in bank-firm multiplex networks," *Finance Research Letters*, vol. 33, p. 101232, 2020.

[7] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of ICLR*, 2017.

[8] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proceddings of ICLR*, 2018.

[9] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *Proceedings of ESWC*, 2018, pp. 593–607.

[10] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, "Graph transformer networks," in *Proceedings of NeurIPS*, 2019.

[11] X. Wang, H. Ji, C. Shi, B. Wang, P. Cui, P. Yu, and Y. Ye, "Heterogeneous graph attention network," in *Proceddings of WWW*, 2019, pp. 2022–2032.

[12] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proceedings of WWW*, 2020.

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

[14] Y. Yaming, G. Ziyu, L. Jianxin, Z. Wei, C. Jiangtao, and W. Quan, "Interpretable and efficient heterogeneous graph convolutional network," *IEEE TKDE*, 2021.

[15] S. Abu-El-Haija, B. Perozzi, R. Al-Rfou, and A. Alemi, "Watch your step: Learning node embeddings via graph attention," in *Proceedings of NeurIPS*, 2018, pp. 9180–9190.

[16] J. Feng, M. Huang, Y. Yang, and X. Zhu, "Gake: Graph aware knowledge embedding," in *Proceedings of COLING*, 2016, pp. 641–651.

[17] J. B. Lee, R. A. Rossi, X. Kong, S. Kim, E. Koh, and A. Rao, "Graph convolutional networks with motif-based attention," in *Proceedings of CIKM*, 2019, pp. 499–508.

[18] K. Zhang, Y. Zhu, J. Wang, and J. Zhang, "Adaptive structural fingerprints for graph attention networks," in *Proceddings of ICLR*, 2020.

[19] J. Wu, J. He, and J. Xu, "Demo-net: Degree-specific graph neural networks for node and graph classification," in *Proceedings of SIGKDD*, 2019, pp. 406–415.

[20] J. B. Lee, R. Rossi, and X. Kong, "Graph classification using structural attention," in *Proceedings of SIGKDD*, 2018, pp. 1–9.

[21] W. Guo, R. Su, R. Tan, H. Guo, Y. Zhang, Z. Liu, R. Tang, and X. He, "Dual graph enhanced embedding neural network for ctrprediction," *arXiv preprint arXiv:2106.00314*, 2021.

[22] S. Gualdi, G. Cimini, K. Primicerio, R. Di Clemente, and D. Challet, "Statistically validated network of portfolio overlaps and systemic risk," *Scientific reports*, vol. 6, no. 1, pp. 1–14, 2016.

[23] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proceedings of SIGKDD*, 2016, pp. 855–864.

[24] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, 2016.

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

[26] D. Guthrie, B. Allison, W. Liu, L. Guthrie, and Y. Wilks, "A closer look at skip-gram modelling," *LREC*, vol. 6, pp. 1222–1225, 2006.

[27] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, no. 1-4, pp. 43–52, 2010.

[28] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of SIGKDD*, 2014.

[29] R. Trivedi, M. Farajtabar, P. Biswal, and H. Zha, "Dyrep: Learning representations over dynamic graphs," in *International conference on learning representations*, 2019.

[30] A. Jabri, A. Owens, and A. Efros, "Space-time correspondence as a contrastive random walk," *Advances in neural information processing systems*, vol. 33, pp. 19 545–19 560, 2020.

[31] Z. Huang, A. Silva, and A. Singh, "A broader picture of random-walk based graph embedding," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 685–695.

[32] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, "Asymmetric transitivity preserving graph embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 1105–1114.

[33] A. Tsitsulin, M. Munkhoeva, D. Mottin, P. Karras, I. Oseledets, and E. Müller, "Frede: anytime graph embeddings," *Proceedings of the VLDB Endowment*, vol. 14, no. 6, pp. 1102–1110, 2021.

[34] F. Chen, Y.-C. Wang, B. Wang, and C.-C. J. Kuo, "Graph representation learning: a survey," *APSIPA Transactions on Signal and Information Processing*, vol. 9, 2020.

[35] J. Zhou, G. Cui, Z. Zhang, C. Y. Z. L. L. W. C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," in *arXiv preprint arXiv:1812.08434*, 2019.

[36] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in *Proceddings of NeurIPS*, 2018, pp. 5165–5175.

[37] Y. Hou, J. Zhang, J. Cheng, K. Ma, R. T. B. Ma, H. Chen, and M.-C. Yang, "Measuring and improving the use of graph information in graph neural networks," in *Proceddings of ICLR*, 2020.

[38] D. Wang, P. Liu, Y. Zheng, X. Qiu, and X. Huang, "Heterogeneous graph neural networks for extractive document summarization," in *Proceedings of ACL*, 2020.

[39] J. B. Lee, R. A. Rossi, S. Kim, N. K. Ahmed, and E. Koh, "Attention models in graphs: A survey," *ACM Transactions on Knowledge Discovery from Data*, vol. 13, no. 6, pp. 1–25, 2018.

[40] S. Vashishth, S. Sanyal, V. Nitin, and P. Talukdar, "Composition-based multi-relational graph convolutional networks," in *Proceedings of ICLR*, 2020.

[41] K. K. Thekumparampil, C. Wang, S. Oh, and L.-J. Li, "Attention-based graph neural network for semi-supervised learning," in *arXiv:1803.03735v1*, 2018.

[42] L. Hu, T. Yang, C. Shi, H. Ji, and X. Li, "Heterogeneous graph attention networks for semi-supervised short text classification," in *Proceedings of EMNLP*, 2019, pp. 4821–4830.

[43] A. Li, Z. Qin, R. Liu, Y. Yang, and D. Li, "Spam review detection with graph convolutional networks," in *Proceedings of CIKM*, 2019, pp. 2703–2711.

[44] H. Zhou, T. Yang, M. Huang, H. Zhao, J. Xu, and X. Zhu, "Commonsense knowledge aware conversation generation with graph attention," in *Proceedings of IJCAI*, 2018, p. 1–7.

[45] K. Wang, W. Shen, Y. Yang, X. Quan, and R. Wang, "Relational graph attention network for aspect-based sentiment analysis," in *Proceedings of ACL*, 2020.

[46] D. Busbridge, D. Sherburn, P. Cavallo, and N. Y. Hammerla, "Relational graph attention networks," in *arXiv preprint arXiv:1904.05811*, 2019.

[47] 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 Transactions on Knowledge and Data Engineering*, 2021.

[48] Z. Yu, D. Jin, Z. Liu, D. He, X. Wang, H. Tong, and J. Han, "As-gcn: Adaptive semantic architecture of graph convolutional networks for text-rich networks," in *Proceedings of ICDM*. IEEE, 2021, pp. 837–846.

[49] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *Proceedings of AAAI*, 2015.

[50] F. Nie, X. Wang, C. Deng, and H. Huang, "Learning a structured optimal bipartite graph for co-clustering," in *Proceedings of NeurIPS*, vol. 30, 2017.

[51] Z. Li, X.-M. Wu, and S.-F. Chang, "Segmentation using superpixels: A bipartite graph partitioning approach," in *Proceedings of CVPR*. IEEE, 2012, pp. 789–796.

[52] K. Riesen and H. Bunke, "Approximate graph edit distance computation by means of bipartite graph matching," *Image and Vision computing*, vol. 27, no. 7, pp. 950–959, 2009.

[53] R. Li, S. Zhang, B. Wan, and X. He, "Bipartite graph network with adaptive message passing for unbiased scene graph generation," in *Proceedings of CVPR*, 2021, pp. 11 109–11 119.

[54] C. Li, K. Jia, D. Shen, C.-J. R. Shi, and H. Yang, "Hierarchical representation learning for bipartite graphs." in *Proceedings of IJCAI*, vol. 19, 2019, pp. 2873–2879.

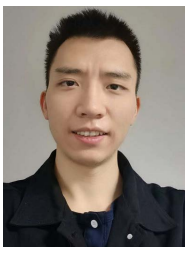
[55] L. Yu, L. Sun, B. Du, C. Liu, W. Lv, and H. Xiong, "Heterogeneous graph representation learning with relation awareness," *IEEE TKDE*, 2022.

[56] W. Jiancan, W. Xiang, F. Fuli, H. Xiangnan, C. Liang, L. Jianxun, and X. Xing, "Self-supervised graph learning for recommendation," in *Proceedings of SIGIR*, 2021, pp. "726–735".
[57] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proceedings of ICML*. PMLR, 2020, pp. 1725–1735.
[58] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *arXiv preprint arXiv:1906.08237*, 2019.
[59] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

Yu Zhao received the B.S. degree from Southwest Jiaotong University in 2006, and the M.S. and Ph.D. degrees from the Beijing University of Posts and Telecommunications in 2011 and 2017, respectively. He is currently an Associate Professor at Southwestern University of Finance and Economics. His current research interests include machine learning, NLP, knowledge graph, Fintech. He has authored more than 30 papers in top journals and conferences including IEEE TKDE, IEEE TNNLS, IEEE TMC, ACL, ICME, etc.

Shaopeng Wei received the B.S. degree from Huazhong Agricultural University in 2019, and now is a Ph.D student in Southwestern University of Finance and Economics. His research interests include graph learning and relevant applications in recommendation system and Fintech.

Huaming Du received his M.S. degree from China University of Petroleum, Beijing, China, in 2018. He is currently pursuing the Ph.D. degree in Southwestern University of Finance and Economics, Chengdu, China. His research interests include Fintech, reinforcement learning, and graph representation learning.



Xingyan Chen received the Ph. D degree in computer technology from Beijing University of Posts and Telecommunications (BUPT), in 2021. He is currently a lecturer with Southwestern University of Finance and Economics. He has published papers the IEEE TMC, IEEE TCSVT, IEEE TII, and IEEE INFOCOM etc. His research interests include Multimedia Communications, Multi-agent Reinforcement Learning and Stochastic Optimization.

Qing Li received his PhD degree from Kumoh National Institute of Technology in February of 2005, Korea, and his M.S. and B.S. degrees from Harbin Engineering University, China. He is a postdoctoral researcher at Arizona State University and the Information & Communications University of Korea. He is a professor at Southwestern University of Finance and Economics, China. His research interests include natural language processing, FinTech. He has published more than 70 papers in the prestigious refereed conferences and journals, such as IEEE TKDE, ACM TOIS, AAAI, SIGIR, ACL, WWW, etc.

Fuzhen Zhuang received the PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences. He is currently a full Professor in 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 TKDE, ACM TKDD, IEEE T-CYB, IEEE TNNLS, ACM TIST, etc.

Ji Liu received the B.S. 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 Ph.D. 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 JMLR, TPAMI, TNNLS, TKDD, NIPS, ICML, SIGKDD, ICCV, and CVPR. Dr. Liu 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 is a Distinguished Professor of Chang Jiang Scholars Program in 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 of School of Management and Economics, University of Electronic Science and Technology of China, and a research scientist in Thomson Co., R & D. He received his Ph.D. in Information Technology from the College of Information Science & Technology, Univ. of Nebraska at Omaha; Master degree in Dept of Computer Science, Univ. of Nebraska at Omaha; and B.S. degree in Department of Physics, Tsinghua University, China. 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).