Exploring Multi-dimension User-Item Interactions with Attentional Knowledge Graph Neural Networks for Recommendation

Zhu Wang, *Member, IEEE,* Zilong Wang, Xiaona Li, Zhiwen Yu, *Senior Member, IEEE,* Bin Guo, *Senior Member, IEEE,* Liming Chen, *Senior Member, IEEE,* Xingshe Zhou, *Senior Member, IEEE,*

Abstract—It is commonly agreed that a recommender system should use not only explicit information (i.e., historical user-item interactions) but also implicit information (i.e., incidental information) to deal with the problem of data sparsity and cold start. The knowledge graph (KG), due to its expressive structural and semantic representation capabilities, has been increasingly used for capturing auxiliary information for recommender systems, such as the recent development of graph neural network (GNN) based models for KG-aware recommendation. Nevertheless, these models have the shortcoming of insufficient node interactions or improper node weights during information propagation, which limits the performance of recommender systems. To address this issue, we propose a Multi-dimension Interaction based attentional Knowledge Graph Neural Network (MI-KGNN) for enhanced KG-aware recommendation. MI-KGNN characterizes similarities between users and items through information propagation and aggregation in knowledge graphs. As such, it can optimize the updating direction of node representation by fully exploring multi-dimension interactions among nodes during information propagation. In addition, MI-KGNN introduces a dual attention mechanism, which allows users and items to jointly determine the weight of neighbor nodes. As a result, MI-KGNN can effectively capture and represent both structural (i.e., the topology of interactions) and semantic information (i.e., the weight of interactions) in the knowledge graph. Experimental results show that the proposed model significantly outperforms baseline methods for top-K recommendation. Specifically, the recall rate is increased by 5.78%, 6.66%, and 3.22% on three public datasets, compared with the best performance of existing methods.

Index Terms—Recommender system, knowledge graph, graph neural networks, information propagation, dual attention mechanism.

1 Introduction

Recommender systems play an important role for alleviating information overload, which are ubiquitous in Internet applications, such as e-commerce platforms (Amazon, Taobao), social networks (Facebook, Twitter), etc. A typical recommender system usually analyzes a user's historical behavior data to identify user preferences, so as to provide personalized recommendation services. One of the most classical recommendation algorithms is collaborative filtering, which achieves great success in many scenarios [1], [2], [3]. Collaborative filtering predicts possible future behavior of users based on historical interactions between users and items. Although collaborative filtering is efficient, it is not effective in scenarios where user and item interactions are sparse [4], [5], [6], [7], [8], [9], [10].

In order to solve this problem, a common method is to use side information to describe the characteristics of users and items in more details [11], [12], [13], [14]. Recently, a growing number of researchers [2], [15], [16], [17], [18], [19], [20], [21], [22] choose to use knowledge graphs (KG) as side information, because a KG contains rich semantic information which contributes to the diversity and interpretability of recommendation results. Existing KG-aware recommender systems can be divided into three categories: embedding-based methods [2], [15], [20], [23], [24], path-based methods [25], [26], [27], [28], [29], [30] and GNN-based methods [16], [31], [32], [33], [34], [35]. Embedding-based methods usually lack explicit modeling, leading to poor interpretability of the recommendation result. The performance of path-based methods mainly depends on manually selected meta paths.

Due to the limitations of the first two kinds of methods, more and more studies are devoted to GNN-based methods. Most existing recommendation algorithms are composed of two parts. One part is responsible for obtaining user/item representations by mining historical user-item interactions as well as possible side information. The other part uses the obtained representations to predict the possibility of future interactions. Many recent studies [16], [31], [32], [35] have proved that GNN-based methods are efficient in learning user and item representations, which address the drawbacks of previous approaches by exploring the semantic and structural information in a knowledge graph. GNN-based methods usually leverage information propagation mechanisms to improve the representation of nodes. Specifically, a common node propagates its information along the path of the graph, and a central node aggregates the information propagated from its neighbors. Through multiple iterations of information propagation, GNN-based methods can aggregate information from higher-order neighbors. For example, Knowledge Graph Attention Network (KGAT) [31] uses an attention mechanism to adjust the collaborative

- *Zhu Wang, Zilong Wang, Xiaona Li, Zhiwen Yu, Bin Guo and Xingshe Zhou are with Northwestern Polytechnical University, Xi'an, 710072, China.*
  E-mail: wangzhu@nwpu.edu.cn
- *L. Chen is with the School of Computing, Ulster University, Newtownabbey, U.K.*

Manuscript received July 12, 2021.

knowledge graph that includes a user-item bipartite graph and a knowledge graph. To accurately capture user preferences, KGAT explores the long-range connectivity in the graph. KGCN [16] adds high-dimensional user representations to the information propagation process and uses the knowledge graph to learn the preferences of different users. The KGCN-LS [32] model was proposed to address the overfitting problem of KGCN, by introducing a label smoothing regularization module. Recently, an adaptive graph convolutional network (AGCN) was designed to adaptively adjust the graph embedding learning parameters by incorporating both the given attributes and the estimated attribute values, which iteratively performs graph representation and attribute inference. Moreover, a knowledge graph-based intent network (KGIN) [35] was proposed to model user-item interactions at a finer granularity, which represents the interaction intent as an attentive combination of KG relations.

While existing GNN-based methods (e.g., KGAT, KGCN, KGCN-LS, AGCN and KGIN) can capture user preferences more accurately, they still have not fully utilized all the possible information for node embedding. Thereby, we propose to explore possible information in a more systematic way, mainly aiming to address the following two challenges.

On one hand, there are rich and various interactions among the nodes of a KG, how to fully extract and utilize such interactions is the first challenge. The classical collaborative filtering algorithm is based on two basic assumptions: 1) users with same behaviors are similar; 2) items related to the same user are similar. These assumptions allow information to propagate in the user-item bipartite graph. However, these two assumptions have only considered users and items. If we consider item attributes and user interests as entities in a KG, these two assumptions can be generalized to four basic assumptions: 1) users with same behaviors are similar (A1); 2) items related to the same user are similar (A2); 3) items with same attributes are similar (A3); and 4) users with same interests are similar (A4). However, existing GNN-based methods have weakened at least one of these assumptions, e.g., KGCN satisfies A1, A2, and A3, and K-GAT satisfies A1, A2, and A4, as illustrated in Section 4.2. To fully address these assumptions, we add the representations of user nodes to the information propagation process of the knowledge graph, and increase information interactions between central nodes[1] and neighbor nodes. In such a way, a user node will be able to impact the propagation of information on each path, and thus a structurally personalized KG is obtained for each user.

On the other hand, different nodes usually have different degrees of importance in a KG, how to quantitatively characterize and update such differences during the information propagation process is the second challenge. To address this challenge, we introduce a dual attention mechanism to the information propagation process. Specifically, we consider both the importance of a common node to a user and the importance of a common node to a central node (i.e., an item in the knowledge graph), which is the most significant difference between MI-KGNN and the attention mechanism currently used in GNN. As a result, we can obtain a semantically (i.e., quantitatively) personalized KG for each user.

1. In this paper, central nodes refer to items in the knowledge graph that need to be represented.

The contribution of this paper is summarized as follows:

- To better represent the node and characterize user interests, we propose a GNN-based KG-aware recommendation model by exploring multi-dimension interactions during the information propagation and aggregation process. Specifically, the representations of user nodes are added to the information propagation process of the knowledge graph, and information interactions are increased between central nodes and neighbor nodes.
- To optimize the information propagation process, a dual attention mechanism is proposed by considering both user interests and central node representations. In particular, a user attention mechanism is designed to assign each neighbor node with a weight based on its importance to the user, and a neighborhood attention mechanism is introduced to allocate weights based on neighbor nodes' significance to the central node. Meanwhile, to reduce the computational cost of information propagation and aggregation, we design a compound optimization approach by systematically considering receptive field adjustment, activation function pruning and aggregation process reduction.
- We conducted extensive experiments on three public datasets. Results show that the proposed MI-KGNN model significantly outperforms state-of-the-art baselines. Particularly, in top-K recommendation, the recall rate is increased by 5.78%, 6.66%, and 3.22% for the three datasets respectively, compared with the best performance of existing methods. Meanwhile, the model's computational cost can be reduced significantly by around 20$\sim$50%.

2 Related Work

2.1 Graph Convolutional Neural Networks

Graph Convolutional Neural Networks (GCN) introduce convolutional neural networks to graphs for robust feature learning. Existing studies on GCN can be divided into two categories, which are spectral-based approaches and spatial-based approaches.

Early studies on GCN mainly follow the spectral-based approach. For example, an representative GCN-related model was proposed by Bruna et al. in 2014 [36], which introduced the eigen-decomposition of Laplacian matrix to perform graph convolution. On this basis, ChebNet [37] and 1stChebNet [38] further designed a set of convolutional filters. These spectral-based approaches define graph convolutions by introducing filters from the perspective of graph signal processing and have achieved impressive results in lots of graph related analytic tasks. However, such approaches need to handle the entire graph at the same time, which makes it difficult to scale to large graphs.

To deal with this issue, lots of studies have been devoted to spatial-based approaches, which define graph convolutions via aggregating feature information from neighbors. For example, GraphSage [39] introduces an information propagation mechanism and defines aggregation functions to assemble information of neighbor nodes. Thereafter, information propagation mechanisms start to be widely adopted by many studies [16], [31], [32].

2.2 KG-aware Recommender Systems

Since the KG contains rich semantic information and structural information, many studies start to use KGs as side information in recommender systems to alleviate the problem of data sparsity and cold start. KG-aware recommender systems can be divided into three categories: embedding-based [2], [15], [20], [23], [24], path-based [25], [26], [27], [28], [29], [30] and GNN-based [16], [31], [32].

Embedding-based methods usually use the structural information in KGs to learn the implicit representations of nodes and perform well on a number of tasks, such as the completion of a KG. For example, as a classic embedding-based method, collaborative knowledge base embedding (CKE) [40] extracts the node's textual representation and visual representation and integrates semantic information in a KG. However, this kind of methods lack an end-to-end ways of training and are poorly explainable.

Path-based methods extracts features based on meta path and inputs them into a prediction model. For example, personalized entity recommendation (PER) [28] indicates the connectivity between nodes by exploring the meta path between nodes, which provides additional guidance for the recommendation results. However, the performance of path-based methods mainly depends on the choice of the meta-path which requires manual designs.

To address the shortcomings of the first two categories of studies, lots of GNN-based methods have been proposed. Most works in this category are based on representation learning, which focus on learning user preferences and then predict the possibility of interaction according to node representations. For example, Wang et al. proposed KGAT [31], which introduced attention mechanisms to GCN and taken into account remote connection information. KGCN [16] allowed user nodes to influence the weight of information flow during the information propagation process. To solve the over-fitting problem in KGCN, KGCN-LS further introduced a label smoothing mechanism. MMGCN [41] considered user preferences for different modalities on the model-specific user-item bipartite graph. There are also some studies dedicated to simplifying the model to reduce the computation complexity. For example, LightGCN [42] simplified the GCN model by including only the most essential components for recommendations. In addition to representation learning, another line of GNN-based methods focuses on extracting rules from a KG and then utilize them to assist with recommendations [15].

None of the above GNN-based methods has fully leveraged interactions between items and their neighborhood during the embedding propagation process. In our previous work [33], we tried to address this issue by exploring multi-dimension interactions systematically. However, the performance improvement is limited due to the lack of an effective attention mechanism to characterize the different importance of a node in a knowledge graph, which is one of the challenges to be addressed in this work.

2.3 Graph Attention Networks

Graph attention networks employ attention mechanisms which assign larger weights to important nodes. An attention mechanism is usually introduced in the aggregation process. For example, Graph Attention Network (GAT) [43] leveraged multi-head attention to control the weight of the neighbor's information. Similarly, Gated Attention Network (GAAN) [44] introduced a multi-head attention mechanism, in which a self-attention mechanism is added to control the information weight of each head. MGAT [45] proposed a gated attention mechanism that utilizes the advantage of the gate and attention mechanism to control the information propagation process.

Existing studies on attention mechanisms in GCN mainly focused on the importance of neighborhood nodes to the central node. The same idea is applied to GCN-based methods in KG-aware recommender systems, ignoring the possibility of directly quantifying user preferences during the information propagation process. In this work, we propose a dual attention mechanism which considers both user interests and central node representations to determine the weight of neighbor nodes.

3 Problem Definition

In this section, we give a formal definition of the KG-aware recommendation problem.

Classical recommendation systems recommend items to users by exploring the user-item interaction matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, in which $y_{uv} = 1$ if user u and item v have historical interactions, otherwise $y_{uv} = 0$. By contrast, a KG-aware recommendation system achieves recommendations by further adding a knowledge graph. A knowledge graph $\mathcal{G}$ usually is composed of entities and relationships, which form triples (h, r, t) where h is the head entity e_1, t is the tail entity e_2, and r is the relationship between e_1 and e_2. A triple represents a fact related to the entities, e.g., the triple (Spider-Man: Homecoming, film.film.director, Jon Watts) means that the film "Spider-Man: Homecoming" is directed by Jon Watts.

The task of KG-aware recommendation is to learn the characteristics of users and items by exploring both the interaction matrix $\mathbf{A}$ and the knowledge graph $\mathcal{G}$, so as to predict the likelihood of future interactions. Specifically, we need to build a model that learns the user representation $\mathbf{u} \in \mathbb{R}^d$ as well as the item representation $\mathbf{v} \in \mathbb{R}^d$, and then develop a prediction function $F(\mathbf{u}, \mathbf{v} | \mathbf{W}, \mathbf{A}, \mathcal{G})$ to calculate the possibility of a new interaction between u and v, in which $\mathbf{W}$ stands for the parameter of function F.

4 MI-KGNN

In this section, we first introduce the basic idea of the proposed model, and then analyze some of the existing models. Finally, we present the detailed design of the MI-KGNN model and perform mathematical verification.

4.1 Basic Idea of MI-KGNN

One of the most representative recommendation algorithms is collaborative filtering (CF), which is based on two basic

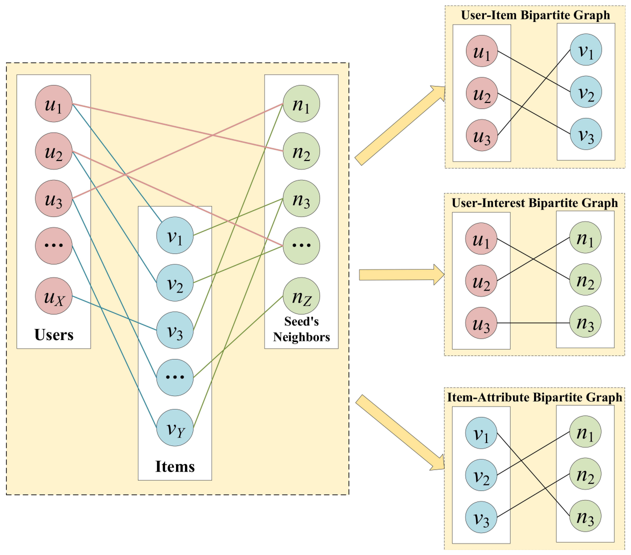
User-Item Bipartite Graph
User-Interest Bipartite Graph
Item-Attribute Bipartite Graph
u_1 u_2 u_3 … u_X
Users
v_1 v_2 v_3 … v_Y
Items
n_1 n_2 n_3 … n_Z
Seed's Neighbors

Fig. 1. Illustration of Multi-dimension Interactions.

assumptions: 1) users with the same behavior are similar (A1); and 2) items related to the same user are similar (A2). These two assumptions promote information flows in the user-item bipartite graph, as shown in the top right of Fig. 1.

In a KG-aware recommendation system, each item in the user-item bipartite graph corresponds to an entity in the KG. We refer to the entities that have corresponding items in the user-item bipartite graph as seeds. The seeds and their neighbors can form another bipartite graph (the bottom right of Fig. 1), and neighbors of a seed are regarded as its attributes. Thereby, based on the idea of collaborative filtering, a straightforward assumption is that items with same attributes are similar (A3). Specifically, in this paper we define such seed items in the knowledge graph as central nodes.

Furthermore, neighbors of a seed may also contribute to the representation of the user who interacted with the corresponding item. In other words, item attributes are also facts that influence the user's choice when selecting items. Thus, such attributes represent the user's possible preferences (i.e., interests) that might have resulted in certain historical interactions. Users and neighbors of the seeds form the 3rd bipartite graph, as shown in middle right of Fig. 1. Consequently, another straightforward assumption is that users with same interests are similar (A4).

Thereby, we propose to develop a GNN-based KG-aware recommendation model by utilizing all the above assumptions, i.e., exploring all the possible interactions, aiming to optimize the updating direction of node representation.

Meanwhile, considering that different nodes usually have different degrees of importance in a KG, we further refine the model by introducing attention mechanisms, aiming to optimize the weight of information propagation.

To sum up, the basic idea of MI-KGNN is to learn more accurate node representations by exploring both structural information (i.e., the topology of interactions) and semantic information (i.e., the weight of interactions) based on knowledge graph neural networks.

4.2 Analysis of Existing Models

While there are some existing KG-aware recommendation models that are designed based on the above assumptions, their updating direction ignores or weakens at least one of the assumptions. In the rest of this section, we will analyze two representative models KGAT [31] and KGCN [16].

KGAT fuses the user-item bipartite graph and the knowledge graph into a collaborative knowledge graph in which users do not interact directly with the neighbors of seeds. KGAT mainly consists of three layers: an embedding layer, a propagation layer and a prediction layer. The loss function of KGAT is defined as follows:

$$\begin{aligned} L^{\mathrm{KGAT}} =& L^{\mathrm{EM}} + \sum_{(u,v^+,v^-)\in O} -\ln\sigma\left(y(u,v^+)-y(u,v^-)\right) \\ &+ \lambda\|\mathbf{W}^{\mathrm{KGAT}}\|_2^2, \end{aligned} \tag{1}$$

where L^{EM} is the loss function of the embedding layer, $\sigma(\cdot)$ is the sigmoid function, $O=\{(u,v^+,v^-)|(u,v^+)\in\mathbf{R}^+,(u,v^-)\in\mathbf{R}^-\}$ denotes the training set, in which $(u,i)\in\mathbf{R}^+$ while $A_{ui}=1$ and $(u,i)\in\mathbf{R}^-$ while $A_{ui}=0$. $\mathbf{W}^{\mathrm{KGAT}}$ is the parameter of the KGAT model, and λ is the parameter of the L2 regularization.

We can find that besides the L2 regularization, the loss function of KGAT mainly consists of an embedding layer and a propagation layer. Specifically, the embedding layer is designed in the same way as TransR [24], and as KGAT uses the gradient descent method to update the model parameters, the updating direction in a single embedding layer is as follows:

$$\Delta\mathbf{u}^{\mathrm{EM}} = g_1(\mathbf{v},\mathbf{W}^{\mathrm{KGAT}}), \tag{2}$$

$$\Delta\mathbf{v}^{\mathrm{EM}} = g_2(\mathbf{n}_{\mathrm{v}},\mathbf{u},\mathbf{W}^{\mathrm{KGAT}}), \tag{3}$$

where $\mathbf{u}$ and $\mathbf{v}$ represent the embedding of user u and item v. $\mathbf{n}_{\mathrm{v}}$ is the embedding of v's neighbors, $\Delta\mathbf{u}$ and $\Delta\mathbf{v}$ are the increments of $\mathbf{u}$ and $\mathbf{v}$, and g_1 and g_2 represent functions related to $\mathbf{v}$.

Furthermore, $y(\cdot)$ in Eq. (1) is the prediction function, which is defined as the inner product as follows:

$$y(u,v) = \mathbf{u}^T\cdot\mathbf{v}. \tag{4}$$

Thereby, the updating direction in a single propagation layer is as follows:

$$\Delta\mathbf{u}^{\mathrm{ppg}} = g_3(\mathbf{v}), \tag{5}$$

$$\Delta\mathbf{v}^{\mathrm{ppg}} = g_4(\mathbf{u}). \tag{6}$$

Based on Eq. (2)~(6), the updating direction of user embedding and item embedding in KGAT can be represented as follows:

$$\Delta\mathbf{u}^{\mathrm{total}} = g_1(\mathbf{v},\mathbf{W}^{\mathrm{KGAT}}) + g_3(\mathbf{v}), \tag{7}$$

$$\Delta\mathbf{v}^{\mathrm{total}} = g_2(\mathbf{n}_{\mathrm{v}},\mathbf{u},\mathbf{W}^{\mathrm{KGAT}}) + g_4(\mathbf{u}). \tag{8}$$

Based on Eq. (7), by repeating the embedding layer for multiple times, information of the two-hop neighbors will be propagated to the user node, i.e., the user's preference information (i.e., the attribute of items) is transmitted to the user node during embedding propagation. In particular, the experimental results of KGAT demonstrate that the best performance is obtained when the embedding layer is repeated for 4 times. However, since a user's preference information needs to pass through the seed node first, the information will be weakened when the model parameters are updated. Thereby, we can conclude that in KGAT the updating direction of a user's embedded value is not close enough to the user's actual interest.

The second representative model is KGCN, which considers user embedding during message propagation, i.e., allows users to interact directly with the seed's neighbors. For different users, the embedding of the same item is different. The embedding formula of the seed node in a single KGCN layer is as follows:

$$\mathbf{v}^u = \sigma\left(\mathbf{W}(\mathbf{v} + \sum_{n_v^j \in \mathbf{N}(v)} \frac{\exp(\mathbf{u}\cdot\mathbf{r}_{v,n_v^j})}{\sum_{n_v^j \in \mathbf{N}(v)} \exp(\mathbf{u}\cdot\mathbf{r}_{v,n_v^j})} \cdot \mathbf{n}_v^j) + \mathbf{b}\right), \quad (9)$$

where $\mathbf{v}^u$ is the embedding of item v from the perspective of user u. $\mathbf{N}(v)$ represents the set of v's neighbors, n_v^j is the j^{th} neighbor, and $\mathbf{r}_{v,n_v^j}$ stands for the relationship between v and n_v^j. $\mathbf{W}$ and $\mathbf{b}$ are trainable parameters, and $\sigma(\cdot)$ is the activation function. The prediction function is the inner product of the embedding of u and the embedding of v, defined as:

$$\hat{y}_{uv} = \mathbf{u}^T \cdot \mathbf{v}^u. \quad (10)$$

During the model training process, KGCN also uses the gradient descent method to update the model parameters. The loss function of KGCN is as follows:

$$L^{\mathrm{KGCN}} = \sum_{u\in\mathbf{U}}\left(\sum_{v:y_{uv}=1} L_c(y_{uv}, \hat{y}_{uv}) - \sum_{i=1}^{T^u} E_{v_i\sim P(v_i)} L_c(y_{uv_i}, \hat{y}_{uv_i})\right) + \lambda||\mathbf{W}^{\mathrm{KGCN}}||_2^2, \quad (11)$$

where $\mathbf{U}$ is the user set, L_c is the cross-entropy loss, P is a negative sampling distribution, and T^u is the number of negative samples for user u. In KGCN, P follows a uniform distribution and $T^u = |\{v : y_{uv} = 1\}|$. The last term is the L2-regularizer. According to Eq. (9)∼(11), we can derive the updating direction of user embedding and item embedding as follows:

$$\Delta\mathbf{u} = f_1(\mathbf{v}^u) + f_2(\mathbf{r}, \mathbf{n}_{\mathrm{v}}, \mathbf{W}^{\mathrm{KGCN}}), \quad (12)$$

$$\Delta\mathbf{v} = f_3(\mathbf{u}, \mathbf{W}^{\mathrm{KGCN}}). \quad (13)$$

According to Eq. (12) and (13), we can see v's neighbors will not directly impact the updating direction of v's embedded value, although multi-layer KGCN will propagate the influence of v's neighbors and $\mathbf{n}_{\mathrm{v}}$ will eventually affect $\mathbf{v}^u$. In other words, $\mathbf{n}_{\mathrm{v}}$ has a weak effect on v during the model's updating phase. Thereby, we conclude that in the KGCN model the updating direction of the item's embedded value is not close enough to its neighbors.

4.3 Information Propagation and Aggregation in MI-KGNN

In this section, we will present the detailed design of the information propagation and aggregation mechanism in the MI-KGNN model.

A straightforward way to restrict the updating direction of the model is to design a new loss function. However, it is not practical if we simply modify the loss function without optimizing the model. For example, if we add $\mathbf{u}^T \cdot \mathbf{n}_{\mathrm{v}}$ directly to the loss function of the KGAT model, the updating direction of user embedding will be as follows:

$$\Delta\mathbf{u}^{\mathrm{total}} = g_1(\mathbf{v}, \mathbf{W}^{\mathrm{KGAT}}) + g_3(\mathbf{v}) + \mathbf{n}_{\mathrm{v}}. \quad (14)$$

While the new updating direction seems to fit assumption A4, a direct modification to the loss function is equivalent to forcing the user to interact with the seed's neighbors in the graph, and the weights of these interactions aren't trainable. Therefore, we choose to design a new model by modifying the way information is exchanged during embedding propagation, rather than directly modifying the loss function.

As presented in Section 4.1, a collaborative knowledge graph can be viewed as three bipartite graphs. However, it is not wise to perform node embedding on all these bipartite graphs at the same time. We choose to perform node embedding only in the KG, and increase the interaction between users and seed neighbors as well as the interaction between seeds and their neighbors during embedding propagation. Accordingly, we design the MI-KGNN layer, which is used to aggregate useful information in the KG to enhance the recommendation performance, as shown in Fig. 2.

4.3.1 *Information Propagation*

Inspired by the information propagation mechanism in GC-N, MI-KGNN allows the node to propagate its embedding through paths in the graph. Specifically, our basic idea is that users should be able to control the weight of information propagation. In such a way, user preferences will be reflected in the information propagation process and the generated KG is personalized for each user. Generally, the information propagated by neighbor nodes are gathered by a central node, so it can control the weight of the propagation path. Meanwhile, as there are different kinds of relationships between entities in the KG, information propagation is also related to the relationship of the corresponding path.

According to the definition of KG-aware recommendation, given a candidate pair of user u and item v, there is an entity corresponding to v in the KG. To address the 4th assumption, we allow u to interact directly with v's neighbors in MI-KGNN. As a result, u's preference can be characterized based on v's neighbors. Specifically, we define the information propagated by v's neighbors from the perspective of user u as follows:

$$\mathbf{v}_N^u = \sum_{n_v^j \in \mathbf{N}(v)} f(\mathbf{u}, \mathbf{v}, \mathbf{r}_{v,n_v^j}, \mathbf{n}_v^j) \cdot \mathbf{n}_v^j, \quad (15)$$

where $\mathbf{u}$ and $\mathbf{v}$ denote the embedding of user u and item v, $\mathbf{N}(v)$ represents the set of neighbors connected to v in the KG, $\mathbf{r}_{v,n_v^j}$ is the representation of the relationship between item v and its neighbor n_v^j, $\mathbf{n}_v^j$ is the embedding of the j^{th} neighbor of v, and $\mathbf{v}_N^u$ denotes the information propagated by v's neighbors from the perspective of user u. Function $f(\cdot)$ corresponds to the proposed dual attention mechanism which controls the weight of information propagation, and we will elaborate its details in the following sections.

4.3.2 *Information Aggregation*

To obtain the representation of a central node in a knowledge graph, we need to aggregate the information propagated from its neighbors. To address the 3rd assumption,

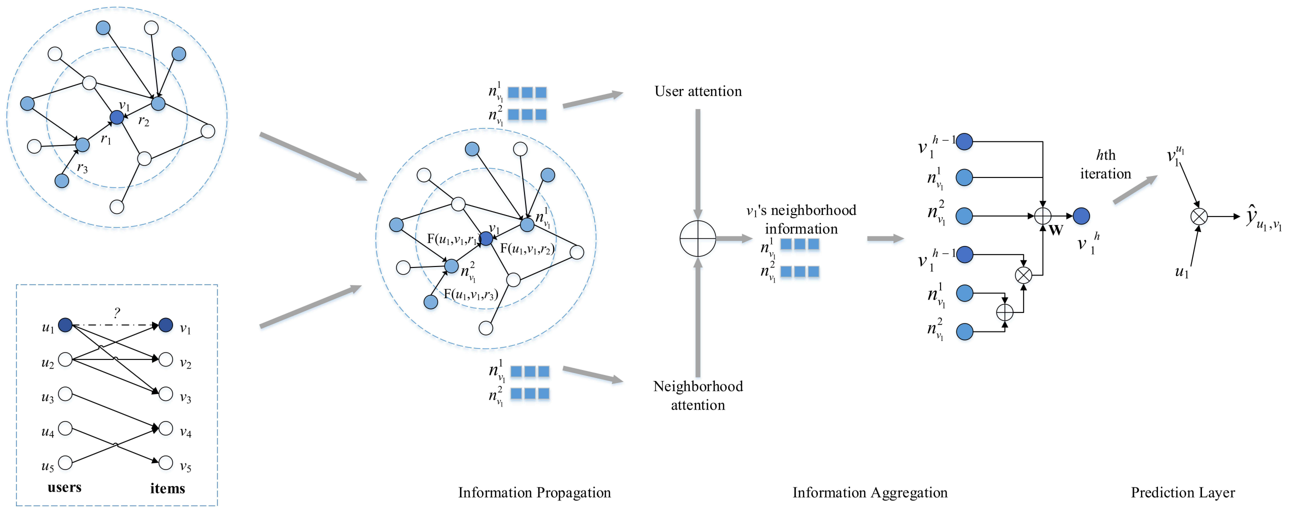

Fig. 2. The proposed MI-KGNN model.

interactions between a central node and its neighbor nodes need to be increased during the process of information aggregation. The aggregator agt: $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ in MI-KGNN is defined as:

$$\begin{aligned}\mathbf{v}^u &= agt(\mathbf{v}, \mathbf{v}^u_N) \\ &= \sigma\left(\mathbf{W}_1\left(\mathbf{v} + \mathbf{v}^u_N\right) + \mathbf{W}_2 Q(\mathbf{v}, \mathbf{v}^u_N) + \mathbf{b}\right).\end{aligned} \tag{16}$$

Here the aggregation result $\mathbf{v}^u$ is a personalized representation of node v from the perspective of user u, where $\mathbf{W}_1$,$\mathbf{W}_2$ and $\mathbf{b}$ are trainable parameters, and $Q(\mathbf{v}, \mathbf{v}^u_N)$: $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is the element-wise product function which strengthens the connection between the neighbor node and the central node. As a result, in the process of parameter optimization, the parameter updating direction of the central node will be directly connected with its neighbors. The mathematical analysis will be given in Section 4.6.

To sum up, to fully address the four assumptions, we add user interests (i.e., $\mathbf{v}^u_N$) to the information propagation process, and increase information interactions between central nodes and neighbor nodes (i.e., $Q(\mathbf{v}, \mathbf{v}^u_N)$) during information aggregation. In such a way, a user node is able to impact the propagation of information on each path, and thus a structurally personalized KG is obtained for each user.

4.4 Dual Attention Mechanism in MI-KGNN

To further optimize the representation of nodes, we put forward a dual attention mechanism, including a user attention mechanism and a neighborhood attention mechanism. In this section, we will describe the design of the dual attention mechanism in details.

4.4.1 User Attention

To predict the probability of a new interaction between user u and item v, the model needs to aggregate the information propagated by nodes in the neighborhood of item v. However, for a given item v, the importance of its neighbors is different in the perspective of different users. For example, different users may choose to watch the same movie, possibly because she likes the director, the subject matter, or the genre of the movie.

Based on this insight, we propose a user attention mechanism to enable personalized information propagation, i.e., assigning different weights to the neighbor nodes based on their importance to a target user. Specifically, the importance of a given neighbor node can be quantified as the inner product of the user vector and the information after propagation as follows:

$$d_{u,n^j_v} = \mathbf{u}^T \cdot Softmax(\mathbf{r}_{v,n^j_v} \odot \mathbf{n}^j_v), \tag{17}$$

where $\odot$ is the Hadamard product, and d_{u,n^j_v} is u's preference for the information of neighbor node n^j_v. Based on the user's preference for each neighbor node, the normalized weight can be defined as:

$$\alpha_{u,n^j_v} = \frac{\exp(\mathbf{u}^T \cdot Softmax(\mathbf{r}_{v,n^j_v} \odot \mathbf{n}^j_v))}{\sum_{n^j_v \in \mathbf{N}(v)} \exp(\mathbf{u}^T \cdot Softmax(\mathbf{r}_{v,n^j_v} \odot \mathbf{n}^j_v))}. \tag{18}$$

4.4.2 Neighborhood Attention

According to the assumptions mentioned above, the information of neighbor nodes includes not only user preferences but also attributes of the central node. Therefore, for a central node, the importance of its neighbors is also different.

Based on this insight, we introduce the second attention mechanism, i.e., the neighborhood attention, which assigns attention weights based on a neighbor node's importance to the central node. Specifically, the degree of contribution can be quantified as the inner product of the transmitted information and the central node vector, which can be formally defined as:

$$d_{v,n^j_v} = \mathbf{v}^T \cdot Softmax(\mathbf{r}_{v,n^j_v} \odot \mathbf{n}^j_v), \tag{19}$$

where d_{v,n^j_v} is the contribution of neighbor node n^j_v's information to central node v. Similarly, the normalization neighborhood attention weight can be formally defined as follows:

$$\alpha_{v,n^j_v} = \frac{\exp(\mathbf{v}^T \cdot Softmax(\mathbf{r}_{v,n^j_v} \odot \mathbf{n}^j_v))}{\sum_{n^j_v \in \mathbf{N}(v)} \exp(\mathbf{v}^T \cdot Softmax(\mathbf{r}_{v,n^j_v} \odot \mathbf{n}^j_v))}. \tag{20}$$

4.4.3 Dual Attention

To achieve effective dual attention based information propagation, we need to fuse the user attention mechanism and the neighborhood attention mechanism.

Considering that the two attention mechanisms are used to control the same information vector, the user attention and neighborhood attention can be combined in the form of a weighted sum. Thereby, we define the dual attention mechanism as:

$$\alpha_{u,v,v_N^j} = \theta \cdot \alpha_{u,v_N^j} + (1-\theta) \cdot \alpha_{v,v_N^j}, \tag{21}$$

where θ is a hyperparameter that controls the degree of influence of the two attention mechanisms. Meanwhile, it also represents the extent to which user behaviors are affected by personal preferences.

Specifically, based on the dual attention mechanism as well as Eq. (18) and (20), the function $f(\cdot)$ in Eq. (15) can be formally defined as:

$$\begin{aligned} f(\cdot) &= \theta \cdot \alpha_{u,v_N^j} + (1-\theta) \cdot \alpha_{v,v_N^j} \\ &= \theta \cdot \frac{\exp(\mathbf{u}^T \cdot Softmax(\mathbf{r}_{v,n_v^j} \odot \mathbf{n}_v^j))}{\sum_{n_v^j \in \mathbf{N}(v)} \exp(\mathbf{u}^T \cdot Softmax(\mathbf{r}_{v,n_v^j} \odot \mathbf{n}_v^j))} + \\ &\quad (1-\theta) \cdot \frac{\exp(\mathbf{v}^T \cdot Softmax(\mathbf{r}_{v,n_v^j} \odot \mathbf{n}_v^j))}{\sum_{n_v^j \in \mathbf{N}(v)} \exp(\mathbf{v}^T \cdot Softmax(\mathbf{r}_{v,n_v^j} \odot \mathbf{n}_v^j))}. \end{aligned} \tag{22}$$

The dual attention mechanism allows the MI-KGNN model to select more important neighbor nodes when aggregating neighborhood information, i.e., neighbors that are more important to a user or a central node can have a higher information propagation weight. Furthermore, by adjusting the hyperparameter θ, the two attention mechanisms' influences on the information propagation process can be fine-tuned, which improves the model's adaptability to different applications or datasets. More details will be elaborated in the experiment section.

4.5 Interaction Prediction and Scalability Optimization

4.5.1 Interaction Prediction

In sections 4.3 and 4.4, we mainly present the 1-layer MI-KGNN model, which only aggregates one-hop neighbor information. If we need to consider the information of h-hop neighbors, MI-KGNN can be extended to a h-layer model by iterating the 1-layer calculation process for h times. After h times of aggregation, the information of h-hop neighbors will be propagated to the central node. In this way, the information of a central node is the representation of the corresponding item from the user's perspective.

MI-KGNN calculates the inner product of the user vector $\mathbf{u}$ and the item information $\mathbf{v}^u$ to predict whether user u and item v will interact with each other. Specifically, it is defined as follows (the same as Eq. (10)):

$$\hat{y}_{uv} = \mathbf{u}^T \cdot \mathbf{v}^u, \tag{23}$$

where $\mathbf{v}^u$ is the representation of item v from user u's perspective, as defined in Eq. (16).

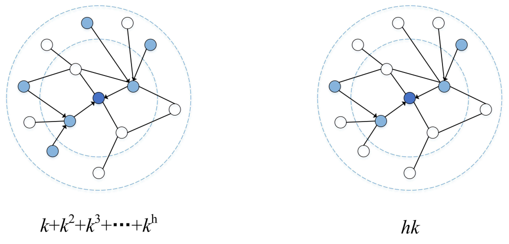

Fig. 3. Receptive field adjustment.

Like most existing models, MI-KGNN uses the gradient descent algorithm to update model parameters. Specifically, the loss function is defined as follows:

$$\begin{aligned} L = \sum_{u \in \mathbf{U}} \left(\sum_{v: y_{uv}=1} L_c(y_{uv}, \hat{y}_{uv}) - \sum_{i=1}^{T^u} E_{v_i \sim P(v_i)} L_c(y_{uv_i}, \hat{y}_{uv_i}) \right) \\ + \lambda \|\mathbf{W}_1 + \mathbf{W}_2\|_2^2. \end{aligned} \tag{24}$$

4.5.2 Scalability Optimization

According to the proposed MI-KGNN model, even if only k nodes are selected from the neighbors of each node at the first layer (i.e., the size of the receptive field is k), there will be k^2 nodes at the second layers. Consequently, the number of nodes will be k^h at the h^{th} receptive field layer. In other words, the complexity of the graph neural networks will increase exponentially with the depth of the receptive field. Therefore, we need to find out a way to reduce the computational cost of the proposed model.

In a knowledge graph, the one-hop neighbors of an item often represent some attribute characteristics of the item, and it is also the potential interest of a user. Therefore, we believe that the semantic information contained in the one-hop neighbors of a node is very profitable for node representation, and we still choose a fixed number of neighbors for each node as the receptive field in the first-level neighborhood. Nevertheless, when considering the second-level receptive field of a node, we choose to select only one adjacent node (non-central node) as the receptive field node for each one-hop neighbor, then the second-level receptive field node will be the same as the first-level. As such, each subsequent layer is the same as the second layer, as shown in Figure 3, which reduces the cost of information aggregation.

It can be seen from Eq.(17) and Eq.(19) that the information is first fed into a non-linear activation function every time the importance of the transmitted information is calculated. In neural networks, nonlinear activation functions are usually introduced to improve the nonlinear expression ability of the model. For the proposed dual attention mechanism, a node's relative importance to the user and the central node is calculated based on the type of transmitting path and the information being transmitted. This process is carried out at a layer of the network, and there is already a nonlinear activation function in the information aggregation process at the same layer. Thereby, a reasonable assumption is that the benefit of adding a nonlinear activation function to the dual attention mechanism is trivial, and the mechanism can be optimized by cutting the activation function. The optimized dual attention mechanism is defined as follows:

$$\begin{aligned} f(\cdot) &= \theta \cdot \alpha_{u,v_N^j} + (1-\theta)\cdot \alpha_{v,v_N^j} \\ &= \theta \cdot \frac{\exp(\mathbf{u}^T \cdot \mathbf{r}_{v,n_v^j} \odot \mathbf{n}_v^j)}{\sum_{n_v^j \in \mathbf{N}(v)} \exp(\mathbf{u}^T \cdot \mathbf{r}_{v,n_v^j} \odot \mathbf{n}_v^j)} + \\ &(1-\theta)\cdot \frac{\exp(\mathbf{v}^T \cdot \mathbf{r}_{v,n_v^j} \odot \mathbf{n}_v^j)}{\sum_{n_v^j \in \mathbf{N}(v)} \exp(\mathbf{v}^T \cdot \mathbf{r}_{v,n_v^j} \odot \mathbf{n}_v^j)}. \end{aligned} \tag{25}$$

To further reduce the computational cost, we introduce a layer-by-layer information aggregation strategy. Specifically, given a receptive field of h layers, we let the model perform information aggregation for h times. At the first time, the information of the h-layer's receptive field is propagated to the $h-1$ layer, i.e., only the nodes of the $h-1$ layer are used as the center for information aggregation. At the h^{th} time, the information of the first layer's receptive field is propagated to the central node. This ensures that the information disseminated at the last time contains information of the entire receptive field, and all the information is only disseminated for once without any redundancy.

4.6 Mathematical Analysis

Based on Eq. (15)-(24), the updating direction of the user vector and item vector in the MI-KGNN model can be derived as follows:

$$\mathbf{\Delta u} = h_1(\mathbf{v}^u) + h_2(\mathbf{r}, \mathbf{n}_\mathrm{v}, \mathbf{W}_1, \mathbf{W}_2), \tag{26}$$

$$\mathbf{\Delta v} = h_3(\mathbf{u}, \mathbf{W}_1) + h_4(\mathbf{r}, \mathbf{n}_\mathrm{v}, \mathbf{u}, \mathbf{W}_1, \mathbf{W}_2). \tag{27}$$

Accordingly, we can see that the user vector is related to both of the item and the user's potential preferences (i.e., the item's neighbor nodes), and the item vector is related to both of the user and the item's attributes (i.e., the item's neighbor nodes). In other words, the proposed model increases the interaction between items and their neighbors during the information aggregation process, and the updating direction of its parameters is already in line with all the four assumptions by considering only a single MI-KGNN layer.

On the contrary, according to Eq. (7)-(8) and Eq. (12)-(13), we can see that neither the KGAT model nor the KGCN model have fully addressed the four assumptions. Specifically, while these models will eventually satisfy all the assumptions to certain extent after multiple iterations, there will always be one assumption that is weakened, resulting in inaccurate node representation.

Assume that a total of m samples are extracted from the user-item bipartite graph, the dimension represented by the model node is d, the number of nodes in the first layer of receptive field of each node is n, and the model considers the information of the H layer of receptive field. MI-KGNN adds a dual attention mechanism in the process of information dissemination, based on which a neighbor's relative importance is calculated and normalized. Therefore, the information propagation process of MI-KGNN is more complicated, which is $O(mHn^{H+2}d)$. Moreover, the complexity of obtaining the node's receptive field and parameter initialization are $O(mn^H)$ and $O(mn^Hd)$, respectively. Therefore, the overall complexity is as follows:

$$F = O(mn^H) + O(mn^Hd) + O(mHn^{H+2}d) = O(mHn^{H+2}d). \tag{28}$$

Originally, given that the size of the receptive field is n, the number of nodes in the H-layer receptive field will be n^H. The proposed receptive field adjustment strategy reduces the amounts of nodes from n^H to nH. As a result, the complexity of the following three process can be lowered. Specifically, the node acquisition cost of the receptive field becomes $O(mnH)$, and the parameter initialization complexity is $O(mnHd)$. Moreover, in the information aggregation process, each iteration needs to aggregate information form nH nodes in the receptive field, each node needs to collect the propagated information from n neighbor nodes to finish information aggregation, and the importance of the propagated information by each neighbor is calculated through a dual attention mechanism, thus the complexity of the H^{th} information updating process is $O(mH^2n^3d)$. To sum up, the model's computational complexity is as follows when the receptive field adjustment strategy is adopted.

$$F = O(mnH) + O(mnHd) + O(mH^2n^3d) = O(mH^2n^3d). \tag{29}$$

In the process of information propagation, the deletion of activation function reduces part of the model's multiplication and addition operations, but the complexity formula remains unchanged if only this optimization strategy is adopted.

With the proposed layer-by-layer aggregation strategy, nodes in the receptive field only need to transmit information once during the H times of iterations, where all nodes except the outermost ones aggregate neighborhood information only for one time. As a result, the cost of the information updating process is $O(mn^{H+1}d)$, and the model's overall computational complexity is as follows:

$$F = O(mn^H) + O(mn^Hd) + O(mn^{H+1}d) = O(mn^{H+1}d). \tag{30}$$

According to Eq. (28)-(30), if all of the three proposed optimization strategies are applied, the computational complexity of the model is as follows:

$$F = O(mnH) + O(mnHd) + O(mHn^3d) = O(mHn^3d). \tag{31}$$

5 Experiments

5.1 Dataset Description

To evaluate the performance of MI-KGNN, we utilize the following three public datasets:

- MovieLens-20M: a stable benchmark dataset, which consists of approximately 20 million ratings and 465,000 tag applications applied to 27,000 movies by 138,000 users.
- Last.FM: a music listening dataset, which contains artist listening records from 1892 users from Last.FM online music system.
- Dianping-Food: a restaurant recommendation dataset provided by Dianping.com, which consists of 23 million interactions between nearly 2.3 million users and 1.3 thousand restaurants.

In the pre-processing stage, a user's positive evaluation of an item is marked as 1. At the same time, we randomly select the same number of negative samples for each user as the unwatched set. Specifically, in the movie dataset, the threshold for positive scoring is set to 4.

Due to the fact that the differences in knowledge graphs will largely affect the results, we use the same knowledge graph as KGCN-LS. More detailed information about the used dataset is summarized in Table 1, where H is the depth of the receptive field, d is the dimension of embeddings, n is the neighbor sampling size, λ is the L2 regularizer weight, α denotes the learning rate, β represents the label smoothness regularizer weight.

TABLE 1
Statistics and hyperparameter settings for the datasets.

	movie	music	restaurant
#users	138,159	1,872	2,298,698
#items	16,954	3,846	1,362
#interactions	13,501,622	42,346	23,416,418
#entities	102,569	9,366	28,115
#relations	32	60	7
#triples	499,474	15,518	160,519
h	1	1	1
d	32	16	16
n	16	8	8
a	0.9	0.9	-0.35
λ	1×10^{-7}	9.1×10^{-5}	9.8×10^{-8}
α	2×10^{-3}	5×10^{-4}	2×10^{-3}
batch size	1,024	128	1,024
β	1	0.1	0.5

5.2 Baselines

The baseline methods we selected in the experiment are as follows:

- SVD [46]: a CF-based model which measures node similarity through interaction
- LibFM [47]: a feature-based model trained with the user ID and the item ID as features.
- LibFM+TransE [48]: unlike LibFM, the feature used in this method is the representation obtained with the TransE method.
- PER [28]: a path-based method, which extracts qualified meta-paths as connectivity between a user and an item.
- CKE [40]: an embedding-based method that combines various item embeddings from different sources including TransR [24] on KG.
- RippleNet [49]: a state-of-the-art algorithm that spread user preferences on the knowledge graph to learn the potential interests of users.
- KGCN [16]: a GCN-based method which transforms a heterogeneous KG into a personalized weighted graph.
- KGCN-LS [32]: a GCN-based method which optimizes the over-fitting problem of KGCN.
- KGAT [31]: a GCN-based model that adds user nodes to the knowledge graph and uses the attention mechanism to control the weight of information propagation.
- CKAN [34]: a GCN-based method similar to KGCN-LS, which utilizes different neighborhood aggregation schemes on the user-item graph and KG respectively, to obtain user and item embeddings.
- AGCN [22]: a GCN-based model which iteratively learns graph embedding parameters with previously learned attribute values, by sending the updated attribute values back to the attributed graph for better graph embedding learning.
- KGIN [35]: a knowledge graph-based intent network, which model each intent as an attentive combination of KG relations, encouraging the independence of different intents for better model capability and interpretability.

5.3 Experimental Settings

We divide each dataset into training set, validation set and test set with a ratio of 3:1:1. As the performance of GCN-based recommendation models were not stable, we carried out three comparative experiments and evaluate different models by averaging the results.

We conducted both top-K recommendation and click-through rate (CTR) prediction experiments. In particular, the Recall@k and NDCG@k metric are used to evaluate the model's performance of the top-K recommendation, and the AUC metric is adopted to validate the applicability of CTR prediction.

The experiment was implemented using Python3.6, TensorFlow 1.12.0 and NumPy 1.14.3. The GPU was GTX 1080Tip. The hyperparameter settings are shown in Table 1.

5.4 Results and Discussions

5.4.1 Performance Comparison

To compare the performance of different models, we conducted both top-K recommendation and CTR prediction experiments.

Top-K recommendation. Results of the top-K recommendation are presented in Table 2. By adjusting the hyperparameters, the performance of some baseline models are slightly improved, compared with the results reported in the original research. Specifically, due to hardware constraints, when running KGAT on the restaurant dataset, we encounter the 'insufficient video memory' problem, indicating that KGAT has the shortcoming of large model size. Therefore, there is no result of KGAT on the restaurant dataset.

According to Table 2, we can find that the proposed MI-KGNN model achieves the best performance, which is about 5.78%, 6.66%, and 3.22% higher than the best performance (underlined) of state-of-the-art methods in the three datasets, respectively. Among all the baselines, the performance of AGCN and KGIN are most close to MI-KGNN. While KGIN considers user-item interactions at a finer granularity of intents and explores long-range semantics of interaction paths under the GNN paradigm, AGCN adaptively adjusts the graph learning process by incorporating both the given attributes and the estimated attributes. Therefore, these two model outperforms the other GNN-based models (e.g., KGCN-LS and KGAT), even though they have addressed three out of the four assumptions. Nevertheless, the performance of KGIN and AGCN are still not as good as MI-KGNN, which verifies the importance of exploring all the proposed assumptions.

To further investigate the working mechanism of GNN-based models, we focus on the performance of the two

TABLE 2
The results of Recall@K in top-K recommendation.

Model	movie					music					restaurant				
	R@2	R@10	R@20	R@50	R@100	R@2	R@10	R@20	R@50	R@100	R@2	R@10	R@20	R@50	R@100
MI-KGNN	**0.070**	**0.205**	**0.293**	**0.438**	**0.576**	**0.050**	**0.160**	**0.224**	**0.340**	**0.465**	**0.050**	0.168	**0.245**	**0.364**	**0.522**
KGIN	0.064	0.192	0.274	0.415	0.562	0.048	0.142	0.208	0.325	0.438	0.048	**0.172**	0.231	0.345	0.501
AGCN	0.060	0.190	0.272	0.418	0.566	0.048	0.141	0.210	0.327	0.440	0.047	0.170	0.230	0.350	0.502
CKAN	0.058	0.185	0.270	0.402	0.560	0.038	0.114	0.182	0.305	0.412	0.048	0.168	0.229	0.344	0.498
KGAT	0.050	0.180	0.266	0.400	0.560	0.030	0.118	0.202	0.320	0.430	/	/	/	/	/
KGCN-LS	0.052	0.186	0.268	0.334	0.468	0.045	0.119	0.185	0.275	0.360	0.047	0.170	0.224	0.340	0.487
KGCN-avg	0.040	0.152	0.232	0.325	0.448	0.032	0.112	0.175	0.265	0.364	0.039	0.157	0.208	0.324	0.475
RippleNet	0.045	0.130	0.205	0.278	0.447	0.032	0.101	0.201	0.242	0.336	0.040	0.155	0.195	0.328	0.440
CKE	0.034	0.107	0.166	0.244	0.322	0.023	0.070	0.128	0.180	0.296	0.034	0.138	0.196	0.305	0.437
PER	0.022	0.077	0.112	0.16	0.243	0.014	0.052	0.089	0.116	0.176	0.023	0.102	0.172	0.256	0.354
LibFM+TransE	0.041	0.125	0.194	0.28	0.396	0.032	0.102	0.198	0.259	0.326	0.044	0.161	0.228	0.343	0.455
LibFM	0.039	0.121	0.185	0.271	0.388	0.03	0.103	0.201	0.263	0.330	0.043	0.156	0.220	0.332	0.448
SVD	0.036	0.124	0.192	0.277	0.401	0.029	0.098	0.195	0.240	0.332	0.039	0.152	0.215	0.329	0.451

TABLE 3
The results of NDCG@K in top-K recommendation.

Model	movie					music					restaurant				
	N@2	N@10	N@20	N@50	N@100	N@2	N@10	N@20	N@50	N@100	N@2	N@10	N@20	N@50	N@100
MI-KGNN	**0.189**	0.191	**0.224**	**0.262**	**0.303**	**0.064**	**0.095**	**0.116**	0.152	0.184	**0.059**	**0.093**	**0.112**	**0.142**	**0.179**
KGIN	0.186	**0.194**	0.221	0.256	0.283	0.063	0.093	0.112	**0.158**	**0.192**	0.055	**0.093**	0.110	0.140	0.165
AGCN	0.185	**0.194**	0.222	0.255	0.282	0.062	0.092	0.113	0.155	0.190	0.056	0.092	**0.112**	0.140	0.164
CKAN	0.181	0.184	0.216	0.229	0.247	0.052	0.088	0.105	0.132	0.149	0.054	0.092	0.108	0.139	0.162
KGAT	0.185	0.190	0.220	0.255	0.277	0.042	0.090	0.111	0.154	0.189	/	/	/	/	/
KGCN-LS	0.180	0.183	0.215	0.225	0.242	0.053	0.090	0.108	0.140	0.142	0.052	0.091	0.106	0.136	0.152

classical models, i.e., KGCN-LS and KGAT. In top-K recommendation, KGAT performs better when the value of k is larger than 20, and the performance of KGCN-LS is better when k is less than 20. While both KGAT and KGCN-LS are GCN-based models, they handle data differently. In particular, KGAT adds user nodes to the knowledge graph, forming a collaborative knowledge graph. Afterwards, it processes the collaborative knowledge graph based on GCN, and achieves the best performance at the 4^{th} layer. The reason is that in the collaborative knowledge graph, the distance between users who have interacted with the same item is 4. Thereby, when the depth of the perception domain is 4, the semantic information in the knowledge graph is fully utilized. However, the 4-layer receptive field often contains many useless nodes and the model suffers from the over-smoothing problem. As a result, lots of nodes in the KGAT model have similar scores, making it achieve better performance when the value of k is relatively larger.

Different from KGAT, KGCN-LS doesn't directly add user nodes into the knowledge graph but allows users to control the information propagation weight. According to the experimental results, KGCN-LS achieves the best performance when the receptive field has 1-2 layers, just like most GCN-based models. This is not only because such models are easy to be over-smoothed, but also because the KGCN-LS model has indeed captured the user's preference within 1-2 layers. Thereby, KGCN-LS performs better when the value of k is less than 20. However, interactions between the central node and neighbor nodes of the KGCN-LS model are not sufficient, resulting in less accurate representation of some nodes in the knowledge graph.

We use the NDCG@k metric to further evaluate the performance of GCN-based models, and the results are summarized in Table 3. Accordingly, we can see that the trend of NDCG@k is basically the same as that of Recall@k, indicating that the proposed MI-KGNN model consistently outperforms existing GCN-based models.

CTR Prediction. To validate the performance of CTR prediction, we mainly adopt the AUC metric and the corresponding results are summarized in Table 4. Accordingly, we can find that MI-KGNN, KGAT, KGCN-LS, CKAN, AGCN and KGIN have similar performance on CTR prediction, and outperform the other models. The next better models are KGCN, RippleNet and LibFM+TransE, all of which have leveraged the information in a KG to assist the recommendation. As a representative path-based method, PER performed very poorly on CTR prediction. The reason should be that the performance of path-based methods largely depends on the used meta-path, and we don't have enough expert knowledge to adjust the choice of meta-path for the dataset. Moreover, we also find that models with good performance on CTR prediction usually can't achieve satisfactory results in top-K recommendations. For example, the performance of the LibFM+TransE model is not as good as that of RippleNet in top-K recommendation. The reason might be that RippleNet can better capture user preferences,

TABLE 4
The results of AUC in CTR prediction.

Model	movie	music	restaurant
MI-KGNN	0.980	**0.803**	0.849
KGIN	**0.981**	0.802	**0.850**
AGCN	0.980	**0.803**	0.849
CKAN	0.980	0.802	**0.850**
KGAT	0.980	**0.803**	/
KGCN-LS	0.979	**0.803**	**0.850**
KGCN-avg	0.975	0.774	0.844
RippleNet	0.960	0.770	0.833
CKE	0.924	0.744	0.802
PER	0.832	0.633	0.746
LibFM+TransE	0.966	0.777	0.839
LibFM	0.959	0.778	0.837
SVD	0.963	0.769	0.838

and the recommended items are more comprehensive than LibFM+TransE.

5.4.2 Result of ablation experiments

To verify the effectiveness of the proposed dual attention mechanism, we conducted a set of ablation experiments with different variants of the MI-KGNN model, and the experimental results are demonstrated in Table 5.

Specifically, the experiment includes four different models, which are based on 1) neither of the two proposed attention mechanisms (i.e., only based on the proposed information propagation and aggregation mechanism, which is the same as the model proposed in our previous work [33]); 2) only the user attention mechanism; 3) only the neighborhood attention mechanism; and 4) the dual attention mechanism, respectively.

Compared with the model without attention mechanisms, the average recall rate of the dual-attention-based MI-KGNN model increased by 27.05%, 21.5% and 3.5% in the movie, music and restaurant datasets, which proves the effectiveness of the proposed dual attention mechanism. It is worth mentioning that the performance improvement achieved by the attention mechanism varies in terms of the used dataset, which is much more significant on the movie and music datasets. This is because the proposed attention mechanism has captured a specific characteristic of the movie and music datasets, which does not exist in the restaurant dataset. To be specific, in these two datasets, about 90% of the item pairs that share a common user in the user-item bipartite graph have a distance less than or equal to 2 in the knowledge graph. In other words, most users in these two datasets have clear preferences when choosing items (movie or music), and such preferences usually have significant influence on user behaviors. In contrast, the restaurant dataset does not show such a characteristic. To sum up, the proposed dual attention mechanism has depicted the above characteristic, making MI-KGNN achieve much significant performance improvements on the movie and music datasets, while the improvement on the restaurant dataset is relatively limited.

Moreover, we observed that the model based on user attention mechanism (i.e., MI-KGNN w/o neig. att.) obtains better performance than the one based on neighborhood attention mechanism (i.e., MI-KGNN w/o user att.), especially on the movie and music datasets. According to the characteristic of these two datasets, as discussed in the above paragraph, we can conclude that the user attention mechanism is more useful to datasets where user behaviors are significantly affected by preferences.

We also observed that a limited performance improvement is achieved by adding the neighborhood attention mechanism to the model, i.e., compared with the user attention mechanism, the neighborhood attention mechanism is less effective. Therefore, we should assign a relatively large value (i.e., $\theta > 0.5$) to the hyperparameter θ in Eq. (21) if the user preference plays an important role in the target dataset. On the contrary, its value should be small (i.e., $\theta < 0.5$) and even be negative (i.e., $-1 < \theta < 0$).

5.4.3 Effect of Scalability Optimization

In order to verify the effectiveness of the proposed scalability optimization method, we conducted experiments on different variants of Fast-MI-KGNN (abbreviated as FMK in following sections):

- w-r-FMK: the MI-KGNN model optimized by the receptive field selection strategy
- w-a-FMK: the MI-KGNN model with only the reduction activation function strategy
- w-l-FMK: the MI-KGNN model with only the layer-by-layer information aggregation strategy
- w/o-r-FMK: the Fast-MI-KGNN model without the receptive field selection strategy
- w/o-a-FMK: the Fast-MI-KGNN model without the activation function reduction strategy
- w/o-l-FMK: the Fast-MI-KGNN model without the layer-by-layer information aggregation strategy
- FMK: the Fast-MI-KGNN model with all of the three optimization strategies

We first demonstrate how the proposed scalability optimization strategies impact the model's recommendation accuracy. Since these optimization strategies have little effect on AUC in the click-through rate estimation experiment, we mainly analyze their impacts on the performance of top-K recommendation. The experimental results are shown in Table 6.

As can be seen in Table 6, the model that achieves the best performance is the one which only uses the receptive field selection strategy (i.e., w-r-FMK). This is because the receptive field selection strategy will only affect the model when considering multi-layer information aggregation. Specifically, this strategy allows the model to obtain information while maintaining the differences between nodes during multi-layer information aggregation. Therefore, compared with other methods, it is more likely to improve the recommendation performance.

The w-l-FMK model reduces the amount of information processing by introducing a layer-by-layer aggregation approach. However, it does not reduce the information within a node's receptive field, but makes the updating of the node's weight less frequently. In other words, it simplifies the updating process of the aggregation weights without adding any extra-information, which weakens the model's representation ability during information aggregation. Therefore, given the same number of model layers, the adoption of the layer-by-layer aggregation strategy causes a slight decline in the recommendation accuracy.

Among the three proposed scalability optimization strategies, the only one that does not change the way of information propagation or aggregation is the pruning of activation functions. Theoretically, elimination of the activation function would reduce the nonlinear expression ability of the model. However, considering that there is still a nonlinear activation function at the same layer model, such an impact should be quite light. According to the experimental results, the adoption of this strategy has little impact on the model, which is even smaller than the impact incurred by the random initialization of parameters.

To validate how the proposed scalability optimization strategies impact the model's computational cost, we use

TABLE 5
The results of ablation experiment.

Model	movie					music					restaurant				
	N@2	N@10	N@20	N@50	N@100	N@2	N@10	N@20	N@50	N@100	N@2	N@10	N@20	N@50	N@100
MI-KGNN	**0.070**	**0.205**	**0.274**	**0.438**	**0.576**	**0.050**	**0.160**	**0.216**	**0.340**	**0.465**	**0.050**	**0.168**	**0.252**	0.364	**0.522**
w/o neig. att.	0.058	0.168	0.249	0.425	0.556	0.049	0.153	0.202	0.324	0.453	0.031	0.112	0.204	0.332	0.483
w/o user att.	0.040	0.133	0.205	0.357	0.486	0.049	0.158	0.210	0.275	0.370	**0.050**	0.140	0.226	0.345	0.433
w/o att.	0.050	0.160	0.216	0.358	0.483	0.045	0.125	0.178	0.286	0.374	0.047	**0.168**	0.244	**0.366**	0.485

TABLE 6
The results of Recall@K in top-K recommendation (Scalability Study).

Model	movie					music					restaurant				
	R@2	R@10	R@20	R@50	R@100	R@2	R@10	R@20	R@50	R@100	R@2	R@10	R@20	R@50	R@100
FMK	**0.065**	0.186	0.324	0.480	**0.618**	0.047	0.134	0.218	0.331	**0.463**	**0.048**	0.159	0.239	0.357	0.479
w-r-FMK	0.063	**0.192**	**0.326**	**0.481**	0.609	**0.051**	**0.141**	**0.221**	0.334	0.461	**0.048**	**0.164**	**0.244**	**0.363**	**0.504**
w-a-FMK	0.042	0.154	0.288	0.437	0.512	0.040	0.132	0.195	0.313	0.433	0.042	0.145	0.217	0.335	0.456
w-l-FMK	0.044	0.149	0.305	0.447	0.499	0.041	0.127	0.186	0.337	0.453	0.040	0.137	0.221	0.337	0.453
w/o-r-FMK	0.044	0.145	0.298	0.446	0.503	0.041	0.118	0.182	0.338	0.436	0.042	0.146	0.215	0.329	0.425
w/o-a-FMK	0.054	0.160	0.257	0.367	0.496	0.040	0.133	0.209	0.341	0.444	0.043	0.153	0.230	0.341	0.470
w/o-l-FMK	0.059	0.188	0.321	0.470	0.556	0.048	0.135	0.216	**0.354**	0.452	0.046	0.160	0.238	0.357	0.469
MI-KGNN	**0.070**	**0.205**	**0.293**	**0.438**	**0.576**	**0.050**	**0.160**	**0.224**	**0.340**	**0.465**	**0.050**	0.168	**0.245**	**0.364**	**0.522**

TABLE 7
The running time of different variants in top-K recommendation (s).

Model	movie				music				restaurant			
	$h=1$	$h=2$	$h=3$	$h=4$	$h=1$	$h=2$	$h=3$	$h=4$	$h=1$	$h=2$	$h=3$	$h=4$
FMK	**428**	**615**	**6,430**	**88,754**	**24**	**30**	**45**	**354**	**254**	**762**	**4832**	**33,937**
w-r-FMK	498	716	7,442	101,238	29	41	59	422	307	978	6470	45,794
w-a-FMK	483	854	8,461	117,382	29	46	69	457	302	1069	7052	50,163
w-l-FMK	447	773	7,614	104,951	25	39	60	439	260	939	6127	43,887
w/o-r-FMK	**428**	758	7,462	102,327	**24**	38	57	435	**254**	934	5967	42,312
w/o-a-FMK	447	633	6,676	92,318	25	32	48	386	260	791	4953	34,770
w/o-l-FMK	483	699	7,261	100,725	29	38	58	419	302	925	6382	45,375
MI-KGNN	498	872	8,581	118,417	29	47	71	464	307	1118	7121	50,869

the model's average running time of one training round as the performance metric. In the experiment, the model was trained for at least 10 rounds using each dataset. Experimental results, as shown in Table 7, indicate that all the three optimization strategies can reduce the training cost.

Specifically, among the three proposed strategies, the layer-by-layer aggregation strategy has the most significant effect on the reduction of training time, while the activation function pruning strategy has the lowest influence. It is worth mentioning that according to the complexity analysis in Section 4.6, the receptive field selection strategy should be the one that has the greatest influence on the model's computational complexity, which is not supported by the result. There are two reasons for this phenomenon. First, the complexity analysis was conducted separately for each pair of links, while in actual experiment we train the model in batches. As a result, parameter initialization and extraction of the receptive field are completed in advance, and the adoption of this strategy will not affect the calculation of these two steps. Second, the model is implemented based on TensorFlow, where the random selection operation of tensors is quite complicated rather than the expected complexity ($O(\mathrm{C})$). During experiments the number of neighbors is relatively small due to hardware limitations, making the decline of computational cost not as significant as expected.

Specifically, if the model needs to aggregate information from multi-hop neighbors, it is a good option to adopt the receptive field adjustment strategy so as to reduce the risk of excessive smoothing. It can help balance the gain due to information aggregation and the loss caused by excessive smoothing, making the model obtain useful information from multi-hop neighbors and achieve better recommendation performance. Meanwhile, according to experimental results, the layer-by-layer aggregation strategy is the most effective way to reduce the training time. However, it will inevitably cause a slight decline to the recommendation accuracy, which is a key issue that we need to consider when designing a recommendation system.

In summary, by introducing a compound optimization approach that systematically considers receptive field adjustment, activation function pruning and aggregation process reduction, the model's computational complexity can be reduced significantly by around 20$\sim$50%, while achieving almost the same recommendation accuracy and even higher Recall rates.

5.4.4 *Effect of hyperparameters*

In the experiment, we made some adjustments to the hyperparameters, which are used to determine the number of neighbors and the number of propagation layers. The specific results are displayed in Table 8 and Table 9.

TABLE 8
Recall@10 of MI-KGNN w.r.t. neighbor sampling size n.

n	2	4	8	16	32
movie	0.140	0.167	0.184	**0.205**	0.176
music	0.124	0.153	**0.160**	0.143	0.134
restaurant	0.160	0.165	**0.167**	0.164	0.148

As shown in Table 8, although more neighbors can provide more information, there usually exist noises in the

TABLE 9
Recall@10 of MI-KGNN w.r.t. depth of support set h.

h	1	2	3	4
movie	**0.205**	0.153	0.102	0.056
music	**0.160**	0.132	0.115	0.073
restaurant	**0.167**	0.141	0.124	0.076

information which might interfere with the final result. Even though the attention mechanism can reduce the weight of such noises, it is still necessary to choose an appropriate number of neighbors.

As MI-KGNN imitates the information propagation process in GCN, it also suffers from the problem of over-smoothing. Thereby, like most GCN-based methods, the proposed MI-KGNN model achieves the best performance when the receptive field has 1-2 layers, as shown in Table 9.

6 Conclusion

In this work, we have proposed a GCN-based KG-aware recommendation model, i.e., MI-KGNN, which characterizes the similarity between users and items based on representation learning in the knowledge graph. To obtain more accurate node representation, on one hand, we increased multi-dimension interactions among the node in the process of information propagation and aggregation, by fully addressing the four basic assumptions proposed based on the idea of collaborative filtering. On the other hand, we optimized the information propagation weights by developing a dual attention mechanism. Moreover, to reduce the computational complexity of the model, we have also put forward a compound optimization strategy by systematically considering receptive field adjustment, activation function pruning and aggregation process reduction.

Experiments on three widely used public datasets demonstrated that the proposed model significantly outperformed existing state-of-the-art methods on top-K recommendation. First, compared with the best performance of existing methods, the average recall rate of MI-KGNN was increased by 5.78%, 6.66%, and 3.22% for the three used datasets. Second, ablation experiments demonstrated that the proposed dual attention mechanism is able to improve the model's recall rate by 27.05%, 21.5% and 3.5%, respectively. Third, the model's computational cost can be reduced significantly by around 20$\sim$50% by incorporating the compound optimization strategy.

Our future work will focus on the following areas. First, we plan to design a better solution to address the overfitting problem of the proposed model. Although the information propagation weight is controlled in MI-KGNN, we have not introduced new regularization except for the L2 regularizer. Second, when the number of layers in the receptive field increases, MI-KGNN still suffers from the smooth transition problem as most GCN-based methods. Thereby, we will investigate how to improve the performance of deep MI-KGNN.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (No. 61960206008, 62032018, 62072375), and the Fundamental Research Funds for the Central Universities (No. 3102019AX10).

References

[1] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web*, ser. WWW '01, New York, NY, USA, 2001, pp. 285–295.
[2] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR'19, New York, NY, USA, 2019, pp. 165–174.
[3] Z. Yu, M. Tian, Z. Wang, B. Guo, and T. Mei, "Shop-type recommendation leveraging the data from social media and location-based services," *ACM Trans. Knowl. Discov. Data*, vol. 11, no. 1, 2016.
[4] Z. Cheng, Y. Ding, L. Zhu, and M. Kankanhalli, "Aspect-aware latent factor model: Rating prediction with ratings and reviews," in *Proceedings of the World Wide Web Conference*, 2018, pp. 639–648.
[5] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir *et al.*, "Wide & deep learning for recommender systems," in *Proceedings of the 1st workshop on deep learning for recommender systems*, 2016, pp. 7–10.
[6] Z. Wang, D. Zhang, X. Zhou, D. Yang, Z. Yu, and Z. Yu, "Discovering and profiling overlapping communities in location-based social networks," *IEEE Transactions on Systems Man and Cybernetics: Systems*, vol. 44, no. 4, pp. 499–509, 2014.
[7] B. Guo, J. Li, V. W. Zheng, Z. Wang, and Z. Yu, "Citytransfer: Transferring inter- and intra-city knowledge for chain store site recommendation based on multi-source urban data," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 4, Jan. 2018.
[8] J. Li, B. Guo, Z. Wang, M. Li, and Z. Yu, "Where to place the next outlet? harnessing cross-space urban data for multi-scale chain store recommendation," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, ser. UbiComp '16, New York, NY, USA, 2016, p. 149C152.
[9] D. Yang, D. Zhang, Z. Yu, and Z. Wang, "A sentiment-enhanced personalized location recommendation system," in *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, ser. HT '13, New York, NY, USA, 2013, p. 119C128.
[10] Z. Yu, M. Tian, Z. Wang, B. Guo, and T. Mei, "Shop-type recommendation leveraging the data from social media and location-based services," *ACM Trans. Knowl. Discov. Data*, vol. 11, no. 1, 2016.
[11] Y. Zhang, H. Yin, Z. Huang, X. Du, G. Yang, and D. Lian, "Discrete deep learning for fast content-aware recommendation," in *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. ACM, 2018, pp. 717–726.
[12] L. Hu, S. Jian, L. Cao, and Q. Chen, "Interpretable recommendation via attraction modeling: Learning multilevel attractiveness over multimodal movie contents," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI'18*, 7 2018, pp. 3400–3406.
[13] J. McInerney, B. Lacker, S. Hansen, K. Higley, H. Bouchard, A. Gruson, and R. Mehrotra, "Explore, exploit, and explain: personalizing explainable recommendations with bandits," in *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 2018, pp. 31–39.
[14] S. Nandanwar, A. Moroney, and M. N. Murty, "Fusing diversity in recommendations in heterogeneous information networks," in *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. ACM, 2018, pp. 414–422.
[15] W. Ma, M. Zhang, Y. Cao, W. Jin, C. Wang, Y. Liu, S. Ma, and X. Ren, "Jointly learning explainable rules for recommendation with knowledge graph," in *Proceedings of the World Wide Web Conference*. ACM, 2019, pp. 1210–1221.
[16] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *Proceedings of the World Wide Web Conference*. ACM, 2019, pp. 3307–3313.
[17] Y. Cao, X. Wang, X. He, Z. Hu, and T.-S. Chua, "Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences," in *Proceedings of the World Wide Web Conference*. ACM, 2019, pp. 151–161.

[18] F. Monti, M. M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, 2017, pp. 3700–3710.

[19] Y. Wu, H. Liu, and Y. Yang, "Graph convolutional matrix completion for bipartite edge prediction." in *Proceedings of the 10th International Conference on Knowledge Discovery and Information Retrieval*, ser. KDIR'18, 2018, pp. 49–58.

[20] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 974–983.

[21] Y. Wang, S. Tang, Y. Lei, W. Song, S. Wang, and M. Zhang, "Disenhan: Disentangled heterogeneous graph attention network for recommendation," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, ser. CIKM '20, New York, NY, USA, 2020, pp. 1605–1614.

[22] L. Wu, Y. Yang, K. Zhang, R. Hong, Y. Fu, and M. Wang, "Joint item recommendation and attribute inference: An adaptive graph convolutional network approach," in *Proceedings of the 43rd ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '20, New York, NY, USA, 2020, pp. 679–688.

[23] E. Palumbo, G. Rizzo, and R. Troncy, "Entity2rec: Learning user-item relatedness from knowledge graphs for top-n item recommendation," in *Proceedings of the 11th ACM Conference on Recommender Systems*. ACM, 2017, pp. 32–36.

[24] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of the 29th AAAI conference on artificial intelligence*, 2015.

[25] B. Hu, C. Shi, W. X. Zhao, and P. S. Yu, "Leveraging meta-path based context for top-n recommendation with a neural co-attention model," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 1531–1540.

[26] H. Zhao, Q. Yao, J. Li, Y. Song, and D. L. Lee, "Meta-graph based recommendation fusion over heterogeneous information networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 635–644.

[27] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[28] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han, "Personalized entity recommendation: A heterogeneous information network approach," in *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 2014, pp. 283–292.

[29] R. Catherine and W. Cohen, "Personalized recommendations using knowledge graphs: A probabilistic logic programming approach," in *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 2016, pp. 325–332.

[30] C. Shi, Z. Zhang, P. Luo, P. S. Yu, Y. Yue, and B. Wu, "Semantic path based personalized recommendation on weighted heterogeneous information networks," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 2015, pp. 453–462.

[31] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 950–958.

[32] H. Wang, F. Zhang, M. Zhang, J. Leskovec, M. Zhao, W. Li, and Z. Wang, "Knowledge-aware graph neural networks with label smoothness regularization for recommender systems," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19, New York, NY, USA, 2019, pp. 968–977.

[33] Z. Wang, Z. Wang, Z. Yu, B. Guo, and X. Zhou, "Mi-kgnn: Exploring multi-dimension interactions for recommendation based on knowledge graph neural networks," in *Proceedings of the 15th International Conference on Green, Pervasive, and Cloud Computing*, 2020, pp. 155–170.

[34] Z. Wang, G. Lin, H. Tan, Q. Chen, and X. Liu, "Ckan: Collaborative knowledge-aware attentive network for recommender systems," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '20, New York, NY, USA, 2020, pp. 219–228.

[35] X. Wang, T. Huang, D. Wang, Y. Yuan, Z. Liu, X. He, and T.-S. Chua, "Learning intents behind interactions with knowledge graph for recommendation," in *Proceedings of the Web Conference*, ser. WWW '21, New York, NY, USA, 2021, pp. 878–887.

[36] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and deep locally connected networks on graphs," in *Proceedings of the 2nd International Conference on Learning Representations*, ser. ICLR '14, 2014.

[37] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16, 2016, pp. 3844–3852.

[38] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the International Conference on Learning Representations*, ser. ICLR'17, 2017.

[39] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, Red Hook, NY, USA, 2017, pp. 1025–1035.

[40] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 2016, pp. 353–362.

[41] Y. Wei, X. Wang, L. Nie, X. He, R. Hong, and T.-S. Chua, "Mmgcn: Multi-modal graph convolution network for personalized recommendation of micro-video," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1437–1445.

[42] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '20, New York, NY, USA, 2020, pp. 639–648.

[43] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proceedings of the 6th International Conference on Learning Representations*, ser. ICLR'18, 2018.

[44] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D.-Y. Yeung, "Gaan: Gated attention networks for learning on large and spatiotemporal graphs," in *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence*, ser. UAI'18, 2018.

[45] Z. Tao, Y. Wei, X. Wang, X. He, X. Huang, and T.-S. Chua, "Mgat: Multimodal graph attention network for recommendation," *Information Processing & Management*, vol. 57, no. 5, p. 102277, 2020.

[46] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 426–434.

[47] S. Rendle, "Factorization machines with libfm," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 3, p. 57, 2012.

[48] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'13, Red Hook, NY, USA, 2013, pp. 2787–2795.

[49] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo, "Ripplenet: Propagating user preferences on the knowledge graph for recommender systems," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2018, pp. 417–426.

Zhu Wang received the PhD degree in computer science and technology from Northwestern Polytechnical University, Xi'an, China. He is currently an associate professor with Northwestern Polytechnical University. He has worked as a visiting student in the Institut TELECOM SudParis, France, from November 2010 to April 2012. His research interests include pervasive computing, machine learning and human-computer interaction. He is a member of the IEEE.

Zilong Wang received the bachelor's degree from Northwestern Polytechnical University, Xi'an, China. He is a postgraduate student in the School of Computer Science, Northwestern Polytechnical University. His research interests include machine learning and recommender system.

Xiaona Li received the bachelor's degree from Taiyuan University of Science and Technology, Taiyuan, China. She is currently a postgraduate student in the School of Computer Science, Northwestern Polytechnical University. Her research interests include machine learning and recommender system.

Zhiwen Yu (SM'13) received the PhD degree in computer science and technology from Northwestern Polytechnical University, Xi'an, China. He is currently a professor with Northwestern Polytechnical University. He has worked as an Alexander Von Humboldt Fellow with Mannheim University, Germany, from November 2009 to October 2010. His research interests include pervasive computing and human-computer interaction. He is a senior member of the IEEE.

Bin Guo (SM'14) received the PhD degree in computer science from Keio University, Minato, Japan, in 2009. He is currently a professor with Northwestern Polytechnical University, Xi'an, China. He was a postdoctoral researcher in the Institut TELECOM SudParis, France. His research interests include ubiquitous computing, mobile crowd sensing, and human-computer interaction. He is a senior member of the IEEE.

Liming Chen is a professor in the School of Computing at University of Ulster, Newtownabbey, United Kingdom. He received his B.Eng and M.Eng from Beijing Institute of Technology (BIT), Beijing, China, and his Ph.D in Artificial Intelligence from De Montfort University, UK. His research interests include data analysis, ubiquitous computing, and human-computer interaction. He is a senior member of the IEEE.

Xingshe Zhou is a professor with Northwestern Polytechnical University, Xi'an, China. His research interests include cloud computing and human-computer interaction. He is a senior member of the IEEE.