Applied Intelligence (2023) 53:10734–10750
https://doi.org/10.1007/s10489-022-04048-4

Entity-driven user intent inference for knowledge graph-based recommendation

Shaosong Li[1] · **Bo Yang**[1] · **Dongsheng Li**[2]

Accepted: 27 July 2022 / Published online: 24 August 2022

Abstract

It has been proven that a knowledge graph (KG) has the ability to improve the accuracy of recommendations, owing to its capability of storing the auxiliary information of items in a heterogeneous structure. Recently, intent inference methods have been developed to explore the user preference information from a KG and user-item interactions and to help improve the recommendation accuracy. The inferred user intent can also be regarded as a part of the reason why the recommendation model recommends a certain item to a user. It is known that there are two types of information in a KG: entities and relations. However, existing recommendation models infer user intents from only the information contained in the relations in a KG. In this paper, we propose a new recommendation model, the entity-driven knowledge intent network (EKIN) to infer user intents using information from both entities and relations and make recommendations for users. For EKIN, we propose to construct an entity-driven user intent graph (EUIG) for each user. The EUIG exploits two types of information in a KG to infer user intents. A graph neural network is constructed with multi-hop propagation in the KG and EUIG to learn the representation of entities, relations and user intents. Moreover, we distill information on users' interactions based on their inferred intents and aggregate the interactions to encode the user characteristics. The experimental results on three real-world datasets demonstrate that the proposed EKIN outperforms the state-of-the-art KG-based recommendation models.

Keywords Recommender system · Knowledge graph · Graph neural network · User intent inference

1 Introduction

Knowledge graph-based recommendation (KGR) has become a research trend in recommender systems. A knowledge graph (KG) preserves real data in a heterogeneous structure as triples that are represented as <head entity, relation, tail entity>, thus providing comprehensive entity and relation knowledge to enrich feature embeddings of items and users in recommender systems [1, 2]. Moreover, factual information in KGs can be utilized to improve the explainability of recommendations by linking extra attributes to items [3, 4].

In the research on KGR, graph neural networks (GNNs) have been widely adopted, owing to the impressive ability of GNNs to process graphs with complex relationships and interdependency between objects [5–7]. Various GNNs, e.g., graph attention networks (GATs) [8, 9], graph convolutional networks (GCNs) [10, 11], have been proposed and studied in KGR. Some researchers also combined the principle of GNN with traditional methods of recommendation for KGR. For example, in [12, 13], KGs with collaborative filtering (CF) algorithms were fused and GNN was used to explore the similarities of items and users. In [14, 15], negative sampling methods were integrated into the KG, and the items surrounding entities were aggregated to learn a more accurate item representation. It has been shown that the state-of-the-art performance in KGR can be achieved using GNN-based models [8, 16–18].

In recent research on KGR, *intent inference* methods [18, 19] have been developedIn these methods, users' potential

✉ Bo Yang
yangbo@uestc.edu.cn

Shaosong Li
shaosongli97@gmail.com

Dongsheng Li
dongshengli@fudan.edu.cn

1 School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China

2 School of Computer Science, Fudan University, Shanghai, 200441, China

intents, which can be utilized by the KGR model to improve the accuracy of recommendations. The inferred intents can also serve as an explanation of the recommendation results. Cao et al. [19] used a knowledge graph competition model to encode the relations and entities of the KG, and a translation-based user preference (TUP) model for recommendations. User intents were inferred from the relations in the KG and two models were jointly trained. Wang et al. [18] utilized a GNN to capture structural information in a KG and aggregate user embeddings with user-item interactions. User intents were modelled as an weighted combination of all KG relations.

It can be noted that existing intent inference methods [18, 19] are based on the assumption that user intents are only associated with relations, rather than entities, in a KG. However, since there are a large number of triples in the form of <head entity, relation, tail entity> and two different types of information in a KG, we argue that user intents may be associated with both relations and entitiesthe triple <movie A, genre, romance> in the KG. In existing methods [18, 19], user intents are referred from the relation, i.e., "genre". However, we argue that the user's preference for "movie A" is not only associated with the "genre", but in fact is more associated with *what* specific genre "movie A" is, i.e., the user's preference for "movie A" can be inferred more accurately from "romance" (or from both "romance" and "genre") than from only "genre".

In this paper, we propose a KGR model called the *entity-driven knowledge intent network* (EKIN) and a new intent inference method, which infers user intents from the information of both entities and relations. As a result, more accurate recommendations can be made. More specifically, we propose a new user intent inference method, which integrates entities and relations with user intents in a graph, which is referred to as an *entity-driven user intent graph (EUIG)*. As illustrated in Fig. 1, the proposed model consists of three components: 1) **KG representation learning**. We implement graph attention network (GAT) in the KG to capture entity and relation embeddings, and the attention mechanism [23, 24] is adopted to calculate the impact of neighbour nodes on the centre nodes. Through multi-hop propagation, entity and relation embeddings take full advantage of the structural knowledge in the KG. 2) **User intent inference**. We propose a user intent inference method by constructing an EUIG for each userprevious research [18, 19], in which user intents were modelled as a weighted combination of all KG relations, we express a user intent by a graphmanner. In this component, we propose a frequency-based attention computation method to calculate the weights of entities and dynamically integrate entity and relation embeddings into user intents in the EUIG. 3) **User characteristic aggregation**. Different items may have different impacts on the user. Therefore, after modelling the user intents embeddings as described above, in modelling the user embeddings, we propose utilizing GAT to calculate the weights for different items. According to the user intents inferred by the EUIG, we distill user-item interactions for the item weights, where important items are aligned with

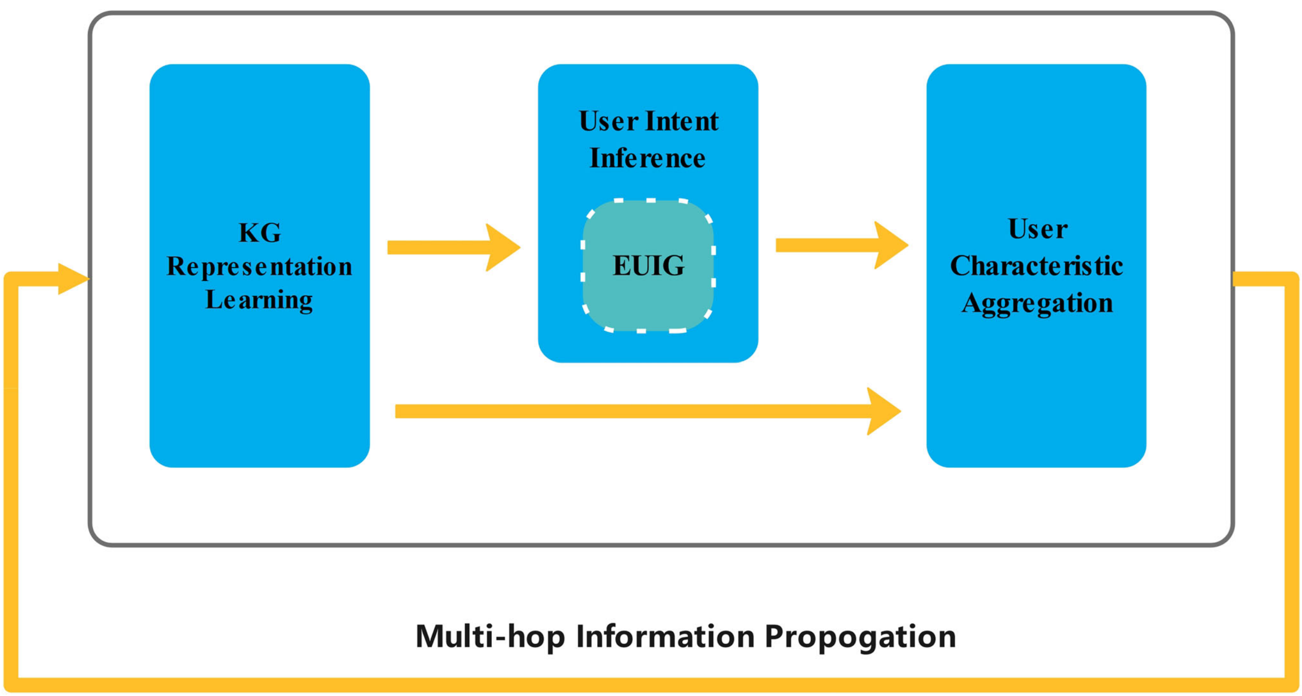


Fig. 1 An illustration of the overall structure of proposed model

larger attribution weights. Based on the user's interactions with the weighted itemscharacteristics and items in the multi-hop layers, and recommend for users.

The main contributions of this paper are summarized as follows:

- We propose a model for KGR to infer user intents based on not only entities but also relations in a KG. Specifically, we propose an entity-driven user intent inference method by constructing an entity-driven user intent graph (EUIG) for each userthis is the first research that promotes the inference of user intents based on both entities and relations. The EUIG can also represent the user intents in an explicit manner.
- We propose calculating the impacts of different items on a user's characteristic according to the user intents inferred by the EUIG. We aggregate users' weighted interactions to capture these characteristics.
- We propose a new model named EKIN, which implements a GNN to learn the information in the KG and EUIG. The experimental results on three real-world datasets demonstrate that the proposed EKIN outperforms the state-of-the-art KG-based recommendation models.

2 Problem formulation

Before introducing the proposed model, we first provide a detailed introduction of the data involved in our study of KGR: user-item interactions, the knowledge graph and entity-driven user intent graph.

User-item interactions In a general scenario of a recommender system, we have a set of M users $\mathcal{U} = \{u_1, u_2, ...u_M\}$ and a set of N items $\mathcal{I} = \{i_1, i_2, ...i_N\}$. In the interaction process between users and items, we focus on the implicit feedback of users, which involves clicking, browsing, purchasing, collecting, etc. Therefore, user-item interactions can be represented by a matrix $\mathbf{Y} \in \mathbf{R}^{M \times N}$, where $y_{ui} = 1$ denotes that there is a recorded interaction between user u and item i. Nevertheless, $y_{ui} = 0$ does not indicate that user u dislikes item i as user u may not have taken notice of this item.

Knowledge graph The KG preserves factual information in the form of heterogeneous graph [1]. The structural data involved in the KG are triples that are represented as <head entity, relation, tail entity>. For example, the triple <Harry Potter, written by, J.K. Rowling> indicates the fact that "Harry Potter was written by J.K. Rowling". In knowledge graph-based recommendation (KGR), KG is mainly used as auxiliary messages in the recommendation process. Therefore, KG mainly contains all kinds of information about the items in the recommender system. We have a set of KG triples $\mathcal{G} = \{(h, r, t) \mid h, t \in \mathcal{E}, r \in \mathcal{R}\}$ for KGR, where $\mathcal{E}$ and $\mathcal{R}$ are collections of entities and relations respectively. Furthermore, each item in the recommender system has the only one entity in the KGa representative item in the recommender system, and there is also an entity representing this movie in the KG. Therefore, we can obtain a corresponding relationship ($\mathcal{I} \in \mathcal{E}$) between items in recommender system and entities in the KG.

Entity-driven user intent graph In this paper, we propose a novel approach to infer user intentsAn example of an EUIG is illustrated in Fig. 2. The left part of Fig. 2 shows the historical interactions of user u and the KG associated with user u's interactions. The right part of Fig. 2 is an EUIG constructed for user u. In EUIG, user intent p is the core, the attributes of the edges are relations in the KG and the attributes of the nodes are entities in the KG. Let $\mathcal{IG}_u = \{(r, v, j) \mid r \in \mathcal{R}, v \in \mathcal{E}, j \in \mathbb{R}\}$ be the EUIG for user u, and j be the frequency of the entity v connecting with user u's interactions. The method of constructing EUIGs is described in detail in Section 3.2. An EUIG is constructed for each user, and we can can infer the user intents through their EUIGs.

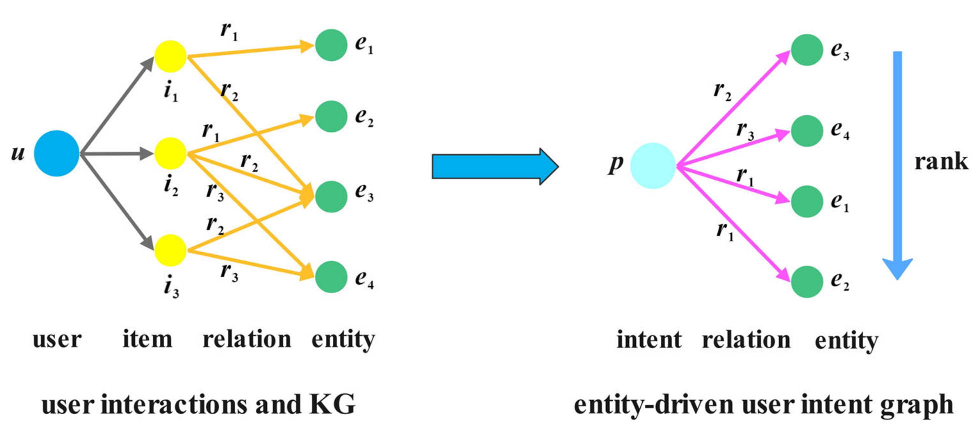


Fig. 2 An example of the proposed EUIG

3 Methodology

In this section, we describe the proposed model EKIN in detail. Figure 3 shows an illustration of the detailed structure of EKIN. Our proposed model consists of three key components: 1) **KG representation learning**. We use a graph attention network (GAT) and represent the centre nodes using the weighted neighbouring nodes in the KG. 2) **User intent inference**. We construct an EUIG for each user, and propose a frequency-based attention computation method to infer user intents from the EUIGs. 3) **User characteristic aggregation**. Based on the above-mentioned user intents, we dynamically aggregate users' interactions to study their characteristics. Then we implement multi-hop propagation in these components to make full use of the structural knowledge in the KG and EUIG to precisely provide recommendation results.

3.1 KG representation learning

In KGR, the KG preserves auxiliary item information to help recommender systems provide accurate recommendation results. On the contrary, in the scenario of recommender systems without KGs, such as some traditional recommendation models [25, 26], the models only use user-item interactions to obtain item representations. Compared with these recommender systems, KGR can be used to obtain the feature information of items from a deeper level. For example, user u_1 has interacted with the item "The Avengers". Based on this user-item interaction, the KG provides some additional information about the item, including "the director of The Avengers is Joss Whedon"; "the genre of The Avengers is action, science fiction, fantasy and adventure", "the actors of The Avengers include Robert Downey Jr and Chris Evanscapture more comprehensive representations of items and improve the accuracy of the recommendation results. Therefore, how to use heterogeneous information in a KG is a key challenge of KGR.

Recently, GNN-based recommendation models [16–18] have shown great potential in processing graph structure data, such as KG. These models capture the features of each node by integrating their neighbours into representations. Furthermore, KG has a large number of complex paths that has not been fully mined. For example, there is a path between entity e_2 and e_3 $: e_2 \overset{r_3}{\leftarrow} e_6 \overset{r_1}{\leftarrow} e_1 \overset{r_2}{\leftarrow} e_3$. To acquire such a relationship between entities, GNN-based recommendation models explore the KG by the multi-hop propagation. In the first step, e_2 aggregates representations from e_6, and e_6 aggregates representations from e_1 using the same approach. In the next step, the representation of e_6 includes the information of e_1. When capturing neighbouring messages from e_6, e_2 includes the information of e_1. neighbouring relations and entities set of entity i. We implement the one-hop propagation in KG as follows:

$$\mathbf{e}_i^{(1)} = f_{KG}\left(\left\{\left(\mathbf{e}_i^{(0)}, \mathbf{e}_r, \mathbf{e}_v^{(0)}\right) \mid (r, v) \in \mathcal{N}_i\right\}\right) \quad (1)$$

where f_{KG} () is the one-hop propagation function in KG to integrate nodes' neighbours into representations, $\mathbf{e}_v^{(0)} \in \mathbb{R}^d$ is the initialized representation of entity v, and $\mathbf{e}_i^{(1)} \in \mathbb{R}^d$ is the representation that aggregates the messages from the one-hop connectivity.

Moreover, we propose that under the form of KG triples <head entity, relation, tail entity>, different tail entities should have different impacts on the same head entity. For instance, a movie has multiple actors, each of whom forms

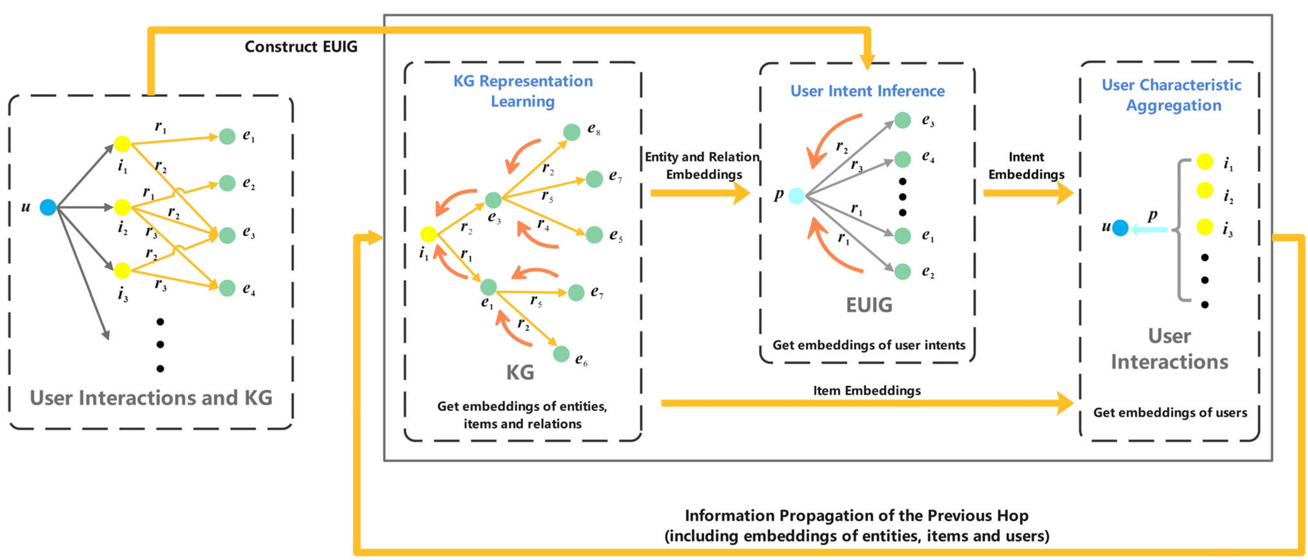


Fig. 3 The detailed structure of the proposed EKIN

a triple with the movie. We consider that the lead roles as the tail entity should have a higher impact on the movie than the supporting roles in these triples. Based on this idea, we utilize GAT to calculate the impact of neighbouring nodes on the centre nodes and learn the representations of entities and relations in the KG. Therefore, our one-hop propagation function $f_{KG}()$ is formulated as follows:

$$\mathbf{e}_i^{(1)} = \sum_{(r,v)\in\mathcal{N}_i} \alpha(r, v)^{(0)} \mathbf{e}_r \odot \mathbf{e}_v^{(0)} \tag{2}$$

where $\alpha(r, v)^{(0)}$ is a KG aggregation attention score that is used to assess the impacts of the neighbours in layer 0. In the different layers, we consider that the impacts on the centre nodes based on their neighbours are different. Therefore, the distribution of one node's KG aggregation attention score is distinguished in different layers. $\odot$ is a dot product between the embeddings of relation r and entity v. Here, the KG aggregation attention score $\alpha(r, v)^{(0)}$ is formulated as follows:

$$\alpha(r, v)^{(0)} = \frac{exp\left(d{1}_{kg}^{(0)}(r, v)\right)}{\sum_{(r',v')\in\mathcal{N}_i} exp\left(d{1}_{kg}^{(0)}(r', v')\right)} \tag{3}$$

$$d{1}_{kg}^{(0)}(r, v) = rrelu\left(d{2}_{kg}^{(0)}(r, v)\, U_{kg}^{(0)}\right) \tag{4}$$

$$d{2}_{kg}^{(0)}(r, v) = (\mathbf{e}_r \odot \mathbf{e}_v^{(0)} + \mathbf{e}_v^{(0)}) W{1}_{kg}^{(0)} + \mathbf{e}_i^{(0)} W{2}_{kg}^{(0)} \tag{5}$$

where entity v is the neighbour of entity i. $W{1}_{kg}^{(0)}$, $W{2}_{kg}^{(0)}$ and $U_{kg}^{(0)}$ are weighted matrices of layer 0 in the KG. Since the representations of nodes are distinguishable in different layers, we derive different weighted matrices to filter out the information that we do not need. $rrelu$ is a nonlinear activation function. Therefore, according to the formulas mentioned above, we can dynamically propagate the information of different layers in the KG.

3.2 User intent inference

In this paper, we propose an entity-driven user intent inference approach to infer user intents from both information of entities and relations. In previous works [18, 19], researchers transferred user-item interactions in recommender systems from <user, item> pairs to <user, intent, item> triples. Since a KG consists of a large number of triples in the form of <head entity, relation, tail entity>, the triples are aligned in the recommender systems as they are in the KG, and user intents are considered to be associated with the relations in the KG. Both of them infer user intents by a weighted combination of all relations in KG and distribute different weighted parameters to different user intents. However, we argue that there are two issues with these modelsfor different relations. (2) The procedure of user intent inference is implicit. The models complete user intent inference by learning the weighted parameters.

To alleviate such problems, we construct an entity-driven user intent graph (EUIG) for each user before training the model, and different user intents are explicitly expressed by different EUIGs. Figure 2 shows an example of how to construct an EUIG. As shown in left part of Fig. 2, we determine the connection frequency between each entity and the items that user u interacts with. Then we rank the entities connected with items in order of the frequency, from which k entities with the highest frequency and their corresponding relations are taken to construct the EUIG, where k is a hyperparameter. For example, in Fig. 2, there are three items that are connected with entity e_3. The frequency of e_3 is 3, which is the highest frequency among all entities. Therefore, entity e_3 ranks first in the EUIG of user u. We consider that the most frequent entities contain the implicit intent information of users, and our EUIG explicitly represents these intentswhich is the EUIG of user u, where j is the frequency of the entity v. We infer user intents as follows:

$$\mathbf{p}_u^{(0)} = f_{IG}\left(\left\{\left(\mathbf{e}_u^{(0)}, \mathbf{e}_r, \mathbf{e}_v^{(0)}\right) \mid (r, v, *) \in \mathcal{IG}_u\right\}\right) \tag{6}$$

where $f_{IG}()$ is the user intent inference function, and $\mathbf{p}_u^{(0)}$ is the inferred intent of user u. $f_{IG}()$ is formulated as follows:

$$\mathbf{p}_u^{(0)} = \sum_{(r,v,j)\in\mathcal{IG}_u} \beta(r, v, j)^{(0)} \mathbf{e}_r \odot \mathbf{e}_v^{(0)} \tag{7}$$

where $\beta(r, v)^{(0)}$ is an attention score to judge the impacts of the entities on the user intent. From the EUIG, we can utilize the information from entities and relations in the KG to infer user intents. Moreover, in the process of inferring user intents, we not only consider the impacts of embeddings of entities and relations on the user intents, but also focus on the impacts of the connection frequency between entities and items in the KG of the user intents. Intuitively, the entities that have a larger frequency should be a better representation of the user intent:

$$\beta(r, v, j)^{(0)} = \frac{exp\left(d{1}_{ig}^{(0)}(r, v, j)\right)}{\sum_{(r',v',j')\in\mathcal{IG}_u} exp\left(d{1}_{ig}^{(0)}(r', v', j')\right)} \tag{8}$$

$$d{1}_{ig}^{(0)}(r, v, j) = rrelu\left((d{2}_{ig}^{(0)}(r, v, j) \,\|\, j) \mid U{1}_{ig}^{(0)} \mid\right) \tag{9}$$

$$d{2}_{ig}^{(0)}(r, v, j) = ((\mathbf{e}_r \odot \mathbf{e}_v^{(0)}) W{1}_{ig}^{(0)} + \mathbf{e}_u^{(0)} W{2}_{ig}^{(0)}) U{2}_{ig}^{(0)} \tag{10}$$

where j is a frequency parameter obtained by the EUIG. It can be seen from Formula (9) that if the frequency parameter j is large, $d_{1_{ig}}^{(0)}(r, v, j)$ is large, and the entity related to j will occupy a more important position with respect to the user intent. $\|$ is the concatenation operation between embeddings, and $|\cdot|$ is the absolute operation. $U1_{ig}^{(0)}, U2_{ig}^{(0)}, W1_{ig}^{(0)}$ and $W2_{ig}^{(0)}$ are weighted matrices in layer 0 of the EUIG. In this component, we can infer user intents differently based on the embeddings of entities in each layer. Then we aggregate the user-item interactions according to these user intents in different layers.

3.3 User characteristic aggregation

In this component, we propose distilling users' interactions based on their intents, since we consider the fact that users may interact with some items that cannot indicate their true intents. For example, a woman who likes romantic movies may watch the movie "Train to Busan" since this movie was in vogue during that time. Users may follow this trend to interact some items outside of their normal intents, and these items should not substantially influence user feature embeddingshistorical interactions based on their intents, where important items are aligned with larger attribution weights. Let $\mathcal{H}_u = \{i \mid y_{ui} = 1, y_{ui} \in \mathbf{Y}\}$ be the interactions of user u. To ensure the balance of the whole model, we adopt a similar strategy in the **KG representation learning** component to aggregate users' interactions:

$$\mathbf{e}_u^{(1)} = f_R\left(\left\{\left(\mathbf{e}_u^{(0)}, \mathbf{p}_u^{(0)}, \mathbf{e}_i^{(0)}\right) \mid i \in \mathcal{H}_u\right\}\right) \tag{11}$$

where $\mathbf{e}_u^{(0)}$ is the initialized representation of user u, and $\mathbf{e}_p^{(0)}$ is the intent of user u as inferred by the EUIG. We use the weighted aggregation of users' interactions to construct the representation of users:

$$\mathbf{e}_u^{(0)} = \sum_{i \in \mathcal{H}_u} \gamma(i)^{(0)} \mathbf{e}_i^{(0)} \tag{12}$$

where $\gamma(i)^{(0)}$ is an attention score to assess the importance of the interactions and aggregate them. The attention score $\gamma(i)^{(0)}$ depends on the user intent, the user embedding of the previous layer and the corresponding item embedding:

$$\gamma(i)^{(0)} = \frac{exp\left(rrelu(d_r^{(0)}(i)U_r^{(0)}))\right)}{\sum_{i' \in \mathcal{H}_u} exp\left(rrelu(d_r^{(0)}(i')U_r^{(0)}))\right)} \tag{13}$$

$$d_r^{(0)}(i) = \left(\mathbf{e}_i^{(0)} + \mathbf{p}_u^{(0)}\right) W1_r^{(0)} + \mathbf{e}_u^{(0)} W2_r^{(0)} \tag{14}$$

where item i is one of the interactions of user u. $U_{r^0}, W1_r^{(0)}$ and $W2_r^{(0)}$ are weighted matrices of layer 0 in the process of user characteristic aggregation. According to the above formulas, we rely on both the user intent and the user representation obtained in the previous layer to filter the users' interactions, and precisely capture their characteristics.

3.4 Multi-hop propagation and model prediction

One-hop propagation functions are included in Formula (1), Formula (5) and Formula (10); as a result, we can implement multi-hop propagation by these functions. Our EKIN first uses the current layer embeddings of entities and items to capture the embeddings in the next layer; second, we utilize the embeddings of entities and relations to infer user intents; and finally, we use the current layer embeddings of users and items to capture the user embeddings in the next layer:

$$\begin{aligned}
\mathbf{e}_i^{(l)} &= f_{KG}\left(\left\{\left(\mathbf{e}_i^{(l-1)}, \mathbf{e}_r, \mathbf{e}_v^{(l-1)}\right) \mid (r, v) \in \mathcal{N}_i\right\}\right) \\
\mathbf{p}_u^{(l)} &= f_{IG}\left(\left\{\left(\mathbf{e}_u^{(l-1)}, \mathbf{e}_r, \mathbf{e}_v^{(l-1)}\right) \mid (r, v, j) \in \mathcal{IG}_u\right\}\right) \\
\mathbf{e}_u^{(l)} &= f_R\left(\left\{\left(\mathbf{e}_u^{(l-1)}, \mathbf{p}_u^{(l-1)}, \mathbf{e}_i^{(l-1)}\right) \mid i \in \mathcal{H}_u\right\}\right)
\end{aligned} \tag{15}$$

After L-layer propagation, we have embeddings of items and users from all L layers; then, we represent the final embeddings of items and users by aggregating all L-layer embeddings:

$$\begin{aligned}
\mathbf{e}_u &= \mathbf{e}_u^{(0)} + \mathbf{e}_u^{(1)} + \cdots + \mathbf{e}_u^{(L)} \\
\mathbf{e}_i &= \mathbf{e}_i^{(0)} + \mathbf{e}_i^{(1)} + \cdots + \mathbf{e}_i^{(L)}
\end{aligned} \tag{16}$$

Finally, we use the inner product between the item embeddings and user embeddings to calculate the prediction score:

$$\hat{y}_{ui} = \mathbf{e}_u^T \mathbf{e}_i \tag{17}$$

3.5 Loss function

In the training procedure of our EKIN, we use the Bayesian personalized ranking (BPR) algorithm [21] to design our loss function. The main idea of this algorithm is to randomly find some samples which the user hasn't interacted with as the user's negative samples in each epoch and complete the optimization of the model by maximizing the scores difference between the positive samples and the negative samples:

$$\mathcal{L}_{BPR} = \sum_{(u, i^+, i^-) \in \mathcal{Z}} -ln\, \sigma\left(\hat{y}_{ui^+} - \hat{y}_{ui^-}\right) \tag{18}$$

where $\mathcal{Z} = \{(u, i^+, i^-) \mid y_{ui^+} = 1, y_{ui^-} = 0, y_{ui^+}, y_{ui^-} \in \mathbf{Y}\}$ is the training set of our EKIN. The positive sample $y_{ui^+} = 1$ indicates that user u has interacted with item i^+, and the negative sample , $y_{ui^-} = 0$ indicates that user u has not previously interacted with item $i^- \sigma(\cdot)$ is the sigmoid function. Then we derive our loss function by combining the BPR loss and the regularization loss:

$$\mathcal{L} = \mathcal{L}_{BPR} + \lambda \|\Theta\|_2^2 \tag{19}$$

where $\Theta = \left\{e_u^{(0)}, e_v^{(0)}, e_u, e_v \mid u \in \mathcal{U}, v \in \mathcal{E}\right\}$ are the parameters of our EKIN, and λ is a hyperparameter to maintain the balance between the BPR loss and the $L2$-regularization loss. We obtain the model parameters by minimizing this loss function.

4 Experiments

In this section, we evaluate the proposed EKIN, and provide experimental results on three scenarios: shopping, book and music recommendations.

4.1 Datasets

To present the performance of the proposed EKIN, we perform experiments with the following datasets in different scenarios: shopping, book and music recommendations. These datasets have been widely used in many previous studies [9, 16–18, 20, 27–31].

- **Alibaba-iFashion** [20] is a publicly available dataset of fashion items from Alibaba online shopping systems. It contains a variety of real-life outfits, e.g., shoes, coats, t-shirts, hats and rings. These outfits are regarded as items in the recommender system. There are more than 1.7 million interactions and 30 thousand items in this dataset.
- **Book-Crossing** is a book dataset of the book-crossing community that was released by [17]. It includes more than 139 thousand interactions between readers and books.
- **Last.FM** is a public dataset of the Last.FM online music system released by [17]. It mainly contains the interactions between users and artists. Artists are considered as items in the recommender system.

The three datasets for the recommender system are briefly introduced above. Since we make recommendations based on both recommendation datasets and KGs, we need to construct the KGs related to the items in recommendation datasets. We follow previous works [8, 17, 18] to construct the KGs for each dataset and ensure that each item in recommendation datasets is aligned with only one entity in the KGs. Since we focus on the user implicit feedback on items and the feedback in Book-Crossing and Last.FM is ratings that are explicit, we utilize the method in [17] to process these datasets. If the rating between the user and the item is larger than 0, we regard the item as the positive sample in the dataset. Table 1 indicates the summarized statistics of the three datasets: Alibaba-iFashion, Book-Crossing and Last.FM.

4.2 Baselines

To demonstrate the effectiveness of our proposed EKIN, we compare with the following baselines: the CF-based model (MF), embedding-based model (CKE) and GNN-based models (KGAT, KGNN-LS, CKAN and KGIN).

- **MF** [21] (matrix factorization) is a typical CF-based model for recommender systems. It only utilizes user-item interactions to learn the representations of users and items without using inforamtion in the KG.
- **CKE** [22] is a classical embedding-based KGR modelthat adopts a heterogeneous graph embedding method TransR [35] to capture the representations of items and uses the principle of CF (collaborative filtering) to make recommendations for users. KG is used to enrich the representations of items.
- **KGAT** [8] is a GNN-based KGR model. It fuses the user-item interactions into the KG to construct a whole graph named CKG and utilizes GNN to propagate information in CKG to obtain the user and item embeddings. KGAT utilizes an attention mechanism to judge the impacts of the neighbours when propagating the information in CKG.
- **KGNN-LS** [16] is also a GNN-based KGR model. It the transforms KG into a user-specific weighted graph, applies the label smoothness assumption to regularize the edge weights in this graph and uses GNN to obtain the personalized item embeddings.
- **CKAN** [17] is a GNN-based KGR model, which uses distinguishing strategies to propagate information in user-item interactions and KG respectively. CKAN

Table 1 Statistics of the datasets

		Alibaba-iFashion	Book-Crossing	Last.FM
User-Item	# users	114,737	17,860	1,872
Interaction	# items	30,040	14,967	3,846
	# interactions	1,781,093	139,746	42,346
Knowledge	# entities	59,156	77,903	9,366
Graph	# relations	51	25	60
	# triples	279,155	151,500	15,518

designs a collaborative signal to combine the messages from both sides, and generates user and item embeddings.

- **KGIN** [18] is a GNN-based KGR modelthat they have previously interacted with.

4.3 Experimental settings

We use recall@K and studies to evaluate the performance of the top-k recommendations [8, 10, 16–19], and K is a parameter set that is as 20 by default. In the training process of our model, since there are only positive samples in the datasets, is adopted to optimize all parameters in the models.

We implement our EKIN in PyTorch [32], and conduct a grid search, which has been widely used in previous research [15–19], for hyperparameters in all models. We set the user, item, entity and relation embeddings d as 64, and the batch size of training and testing procedure as 1024 for all models. The learning rate is searched among $\{10^{-4}, 10^{-3}, 10^{-2}\}$, the $L2$ normalization hyperparameter λ is tuned in $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$, and the number of multi-hop propagation layers L is searcher in $\{1, 2, 3\}$ for all GNN-based models. Moreover, we adopt a uniform distribution to initialize all embeddings in the models.

Specifically, for our EKIN, we set the learning rate as 10^{-4}, the number of propagation layers L as 3, and the $L2$ hyperparameter λ as 10^{-5}. Since different KGs related to the datasets have different average links between entities and relations, we distribute different numbers of entities linked with the user intents to different datasets when constructing the EUIGs. We set the number of linked entities k as 20 in Alibaba-iFashion and 8 in both Book-Crossing and Last.FM.

4.4 Performance comparison

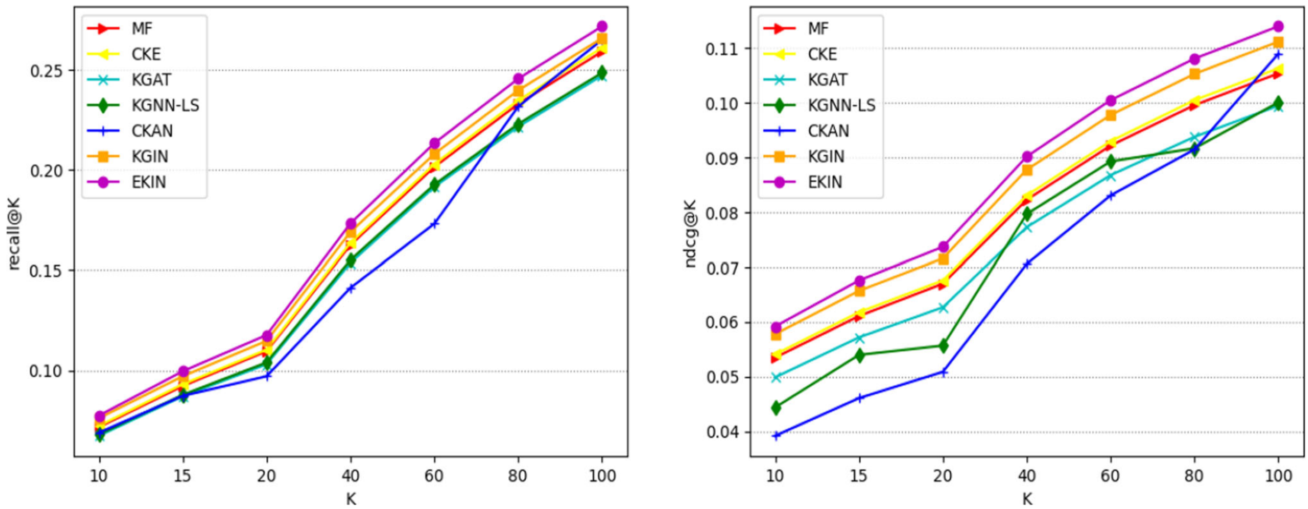
We determine the top-K recommendations on the three datasets for all baselines and our EKIN. The results of top-K recommendations and statistical significance tests are presented in Table 2, where recall@20 and NDCG@20 are compared for all models. We also conduct the experiments on the three datasets using different settings of k for the top-K recommendations (e.g., recall@10, recall@15 et al.), and the results are presented in Figs. 4, 5 and 6. The observations as follows:

- In comparison with all baselines, our proposed EKIN achieves the best performance among the three datasets. Specifically, when using EKIN, improvements of 2.5%, 4.7% and 7.2% for recall@20 and 2.9%, 5.8% and 7.2% for NDCG@20 are achieved over the state-of-the-art baselines on Alibaba-iFashion, Book-Crossing and Last.FM respectively. These improvements represent the effectiveness of our EKIN. We attribute these improvements to the followingfrom the entities, relations and the heterogeneous structure in the KG are contained in the EUIG to infer user intents. To distinguish between user intents, we construct an EUIG for each user based on their interactions. Compared to KGIN, in which the user intents are also explored, the experimental results demonstrate that entity-relation pairs are more suitable for user intents than relations. weighted aggregation in the KG, EKIN can lead entity embeddings to selectively aggregate neighbour information and distill users' interactions based on the user intents inferred by the EUIG. This allows our EKIN to capture the characteristics of entities and users better than the baselines; (3) The impressive ability of GNN.

Table 2 The result of Recall and NDCG for the top-K recommendations

Model	Alibaba-iFashion		Book-Crossing		Last.FM	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
MF	0.1095	0.0670	0.1235	0.0753	0.3102	0.2052
CKE	0.1103	0.0676	0.1418	0.0879	0.3254	0.2104
KGAT	0.1030	0.0627	0.1518	0.0932	0.3302	0.2109
KGNN-LS	0.1039	0.0557	0.1489	0.0979	0.3256	0.2085
CKAN	0.0970	0.0509	0.1560	0.0908	0.3300	0.2067
KGIN	0.1147	0.0716	0.1563	0.0998	0.3335	0.2111
EKIN	**0.1176**	**0.0737**	**0.1637**	**0.1056**	**0.3576**	**0.2263**
p-value	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001

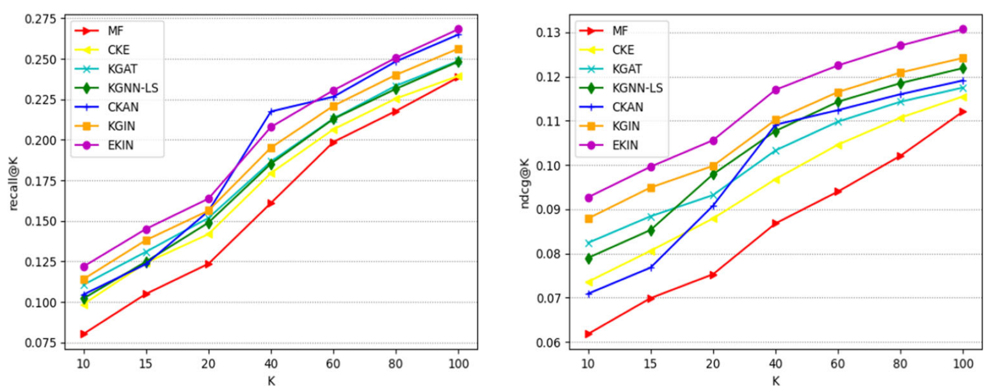
Overall performance comparison
The bold entries indicate the best experimental results obtained in all models



(a) Recall@K (b) NDCG@K

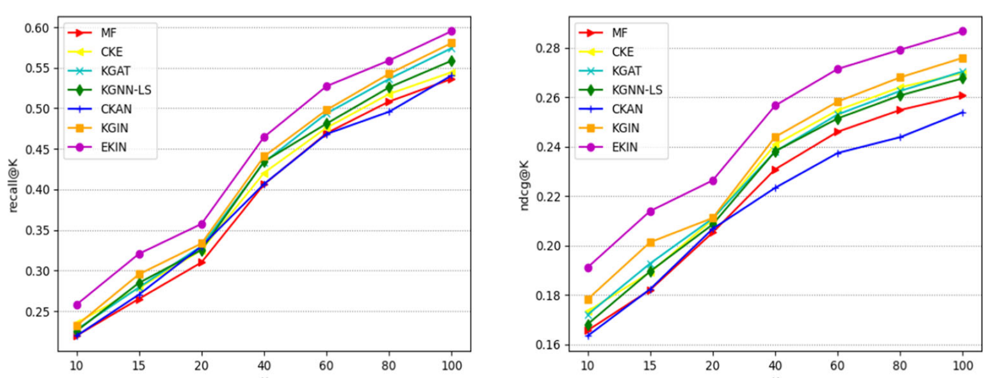
Fig. 4 Recall@K and NDCF@K for the top-K recommendations on Alibaba-iFashion

Fig. 5 Recall@K and NDCG@K for the top-K recommendations on Book-Crossing



(a) Recall@K (b) NDCG@K

Fig. 6 Recall@K and NDCG@K for the top-K recommendations on Last.FM



(a) Recall@K (b) NDCG@K

Since GNN can process graphs with complex relationships well, GNN-based models are better at dealing with heterogeneous information in the KG than the baselines without GNN.

- initial items, which guides the model to precisely obtain user embeddings and improves the recommendation accuracy.
- The information in a KG can improve the performance of recommendation. beneficial to integrate auxiliary information into recommender systems.
- This indicates that the GNN has an impressive ability to capture the information in a KG. The GNN adopts the multi-hop propagation strategy and utilizes the neighbour information to learn the representation of each node. However, CKE outperforms GNN-based models on Alibaba-iFashion. The reasons could be as follows: (1) The multi-hop information in a KG is very large, and training to obtain a better performance is difficult; (2) Many informative messages on Alibaba-iFashion are contained in the one-hop connections. CKE utilizes TransR to learn the information in the KG, therefore, better performance on Alibaba-iFashion may be achieved by CKE when with these connections.

4.4.1 Impact of the entity information and relation information on user intents

In this section, we demonstrate the influence of entity and relation information on the user intent inference. We separately remove the entity embeddings and relation embeddings from EUIGrepresent the EKINs that do not use entity and relation information to infer user intents, respectively. Table 3 lists the experimental results on three datasets using EKIN, EKIN$_{w/o\ E}$ and EKIN$_{w/o\ R}$.

It can be observed from Table 3 that removing the information of entities or relations from the user intents reduces the accuracy of recommendations. EKIN$_{w/o\ E}$ and EKIN$_{w/o\ R}$ only consider the user intent as a weighted combination of relation embeddings and entity embeddings, respectively. It can be noted that EKIN$_{w/o\ R}$ outperforms EKIN$_{w/o\ E}$, which indicates that entities contain more information about user intents than relations in the KG. EKIN utilizes the entity-relation pairs to devise user intents. Since EKIN exploits not only the information of entities and relations, but also the heterogeneous information of the KG, better performance is achieved using EKIN compared with both EKIN$_{w/o\ E}$ and EKIN$_{w/o\ R}$.

4.4.2 Impact of the propagation layers

Next, we consider the influence of the number of multi-hop propagation layers on the recommendation accuracy. The more layers of propagated information there are, the wider range of information in the KG that can be obtained in the embeddings, but the complexity of the model will also be improved. Therefore, we search the propagation layer L in $\{1, 2, 3\}$, and present the results in Table 4. EKIN-L indicates that the model has L propagation layersWe observe the following:

- The propagation layer L influences the amount of information contained in the embeddings of entities in KG. For example, there is a path between entity e_1 and e_4: $e_1 \overset{r_2}{\leftarrow} e_2 \overset{r_1}{\leftarrow} e_3 \overset{r_3}{\leftarrow} e_4$. If $L = 1$, the embedding of entity e_1 will only contain the information of entity e_2. If $L = 2$, the embedding of entity e_1 will contain the information of both entities e_2 and e_3. Therefore, the larger propagation layers make the recommendation intuitively more accurate.
- Specifically, EKIN-2 has the better recommendation performances than EKIN-1 on all datasets. We attribute this performance to the following points: (1) The information of the KG in a 2-hop propagation contains more information than that in a 1-hop propagation, which is useful for recommendations. It can be seen that the 2-hop propagation can make embeddings of entities capture larger ranges of neighbours' information, which is beneficial for the model to learn the characteristic of items. (2) A larger range of information propagation can assist the model in better learning user preferences.
- We observe that the NDCG performance on Last.FM shows the phenomenon of first decreasing and then increasing. We believe that increasing the number of propagation layers also has the disadvantage that the

Table 3 Impact of the entity information and relation information

		EKIN$_{w/o\ E}$	EKIN$_{w/o\ R}$	EKIN
Alibaba-iFashion	Recall	0.1081	0.1169	0.1176
	NDCG	0.0672	0.0729	0.0737
Book-Crossing	Recall	0.1559	0.1582	0.1637
	NDCG	0.0998	0.1006	0.1056
Last.FM	Recall	0.3398	0.3494	0.3576
	NDCG	0.2136	0.2207	0.2263

Table 4 Impact of the propagation layers L

		EKIN-1	EKIN-2	EKIN-3
Alibaba-iFashion	Recall	0.1064	0.1167	0.1176
	NDCG	0.0655	0.0728	0.0737
Book-Crossing	Recall	0.1420	0.1621	0.1637
	NDCG	0.0891	0.1023	0.1056
Last.FM	Recall	0.3464	0.3517	0.3576
	NDCG	0.2201	0.2178	0.2263

information contained in the nodes will become smooth and more similar, especially when the amount of the data in the dataset is small or the propagation layers are large. It can be seen from Table 1 that the size of the Last.FM dataset is the smallest among all datasets, and there are only 15518 triples in the Last.FM KG. With an increase in the propagation layers, the improvement of the recall performance on Last.FM dataset is not as significant as the other two datasets. The gap in the recall performance between EKIN-2 and EKIN-1 is very small. Therefore, even though the NDCG performance seems to first decrease, we believe that this phenomenon is within the normal range of data fluctuation.

4.4.3 Impact of the number of entity-relation pairs in EUIG

In the process of constructing the EUIG, we choose different numbers of entity-relation pairs to infer user intents based on the characteristics of different KGs. Here, we search the number of entity-relation pairs k in the range of $\{4, 8, 12, 16, 20\}$, and conduct experiments on the Book-Crossing and Last.FM datasets. We summarize the results in Table 5. We observe that the number of entity-relation pairs in the EUIG has the following features:

- The number of entity-relation pairs in the EUIG affects the factor numbers that we consider when making recommendations for users. For example, we connect the entity-relation pair, i.e., <romance, genre>, with user u_3. When making recommendations for user u_3, we will focus on whether the item has the genre of romance. Therefore, it is important to choose the proper number of entity-relation pairs in the EUIG according to the features of the datasets.
- On Last.FM, the best performance among all hyperparameters is achieved when the number of entity-relation pairs is set to 8. When analyzing the Last.FM dataset, there are an average of 22 items in each user's interaction, and each user's interaction is linked with 16 entity-relation pairs on average. Therefore, it is necessary to find the pairs that can represent the user preferences from these entity-relation pairs. By ranking the entity-relation pairs in order of frequency, the experimental results show that the best choice is to select the largest 8 pairs to connect with user intents.
- Specifically, we find that the differences in performances on Book-Crossing among different hyperparameters is small. the number of items the user actually interacts with, we add zero embeddings to the EUIG to ensure the consistency of the structure. The experimental results show that the best performance on Book-Crossing is achieved when the number of entity-relation pairs is set to 8.

4.5 Time complexity analysis

The time consumption of EKIN is mainly due to the aggregation of information and the user intent inference. In the component of information aggregation, the time complexity can be divided into two parts, user characteristics and entity representations. The calculation complexity of user characteristics is $O(L|\mathbf{Y}|d^2)$, where L, $|\mathbf{Y}|$, d represent the number of propagation layers, the number of interactions in the recommender system and the number of embedding

Table 5 Impact of the number of entity-relation pairs in EUIG

	Book-Crossing		Last.FM	
	Recall	NDCG	Recall	NDCG
4	0.1610	0.1029	0.3466	0.2180
8	0.1637	0.1056	0.3576	0.2263
12	0.1624	0.1042	0.3467	0.2189
16	0.1587	0.1010	0.3461	0.2167
20	0.1590	0.1025	0.3353	0.2100

size, respectively. The calculation of entity representations is $O(L|\mathcal{G}|d^2)$, where $|\mathcal{G}|$ is the number of triplets in the KG. In the component of user intent inference, the calculation complexity is $O(L|\mathcal{U}|kd^2)$, where $|\mathcal{U}|$ is the number of users in the recommender system, and k is the number of entity-relation pairs in the EUIG, which is a hyperparameter set as 8 on the Book-Crossing and Last.FM datasets and 20 on the Alibaba-iFashion dataset. Therefore, the time complexity of the whole model is $O(L|\mathbf{Y}|d^2 + L|\mathcal{G}|d^2 + L|\mathcal{U}|kd^2)$. Figure 7 shows the time consumption of different models running on the three datasets.

4.6 A case study of the proposed EKIN

In this section, we present a case study of the proposed EKIN. The case study consists of two parts: user intent inference by the EUIG and user characteristic aggregation by the GAT. We take user u_{25} from the Book-Crossing dataset as an example to provide an in-depth illustration of the model.

4.6.1 User Intent Inference by the EUIG

In the Book-Crossing dataset, user u_{25} has interacted with 16 items. Since these items also appear in the KG of Book-Crossing, we determine the connection frequencies between the entities and these items. Then, we rank the entities connected with these items in order of their frequencies and take the k entities with the highest frequencies and their corresponding relations to construct the EUIG of user u_{25}, where k is a hyperparameter and is set to 8 in the Book-Crossing dataset. Figure 8 shows an illustration the EUIG for user u_{25}. In the EUIG, the central node p_{25} represents the intent of the user, and other nodes represent the selected entities with their corresponding relations and frequencies. For example, $(e_{19951}, r_2, 4)$ means that entity e_{19951} and relation r_2 are connected four times with the items interacted by user u_{25} in the KG. The weights on the lines are calculated by the frequency parameters and the embeddings of entities and relations. The frequency parameters determine the trends of the weights, and the embeddings fine-tune the values of the weights. Finally, we combine the weighted embeddings of entities and relations to obtain the embedding of the user intent in the EUIG.

4.6.2 User characteristic aggregation by the GAT

We utilize the graph attention network (GAT) to aggregate the information of items that the user has interacted with. Figure 9 shows the structure of GAT related to user u_{25}. In this figure, the intent of this user and 16 items that this user has interacted with are shown. We calculate the attention weights of these items based on the user intent, p_{25}, which is obtained by the EUIG of the user. Figure 10 shows the attention weights of GAT related to user u_{25}. We integrate the weighted embeddings of the items, which is used for recommendations, to obtain the embedding of user u_{25}.

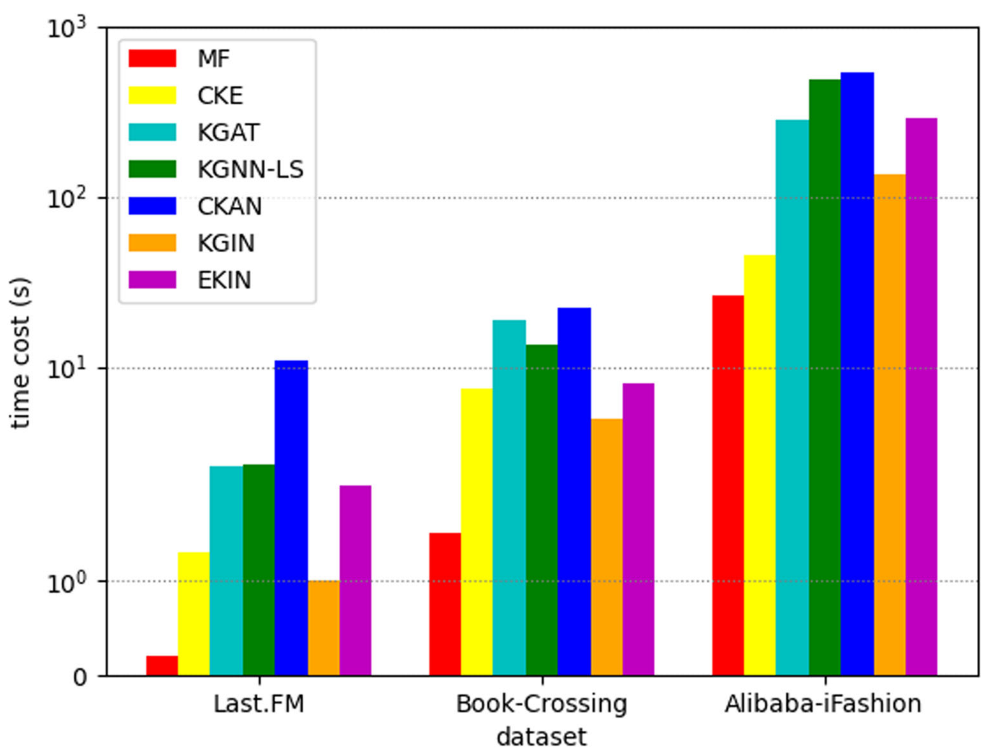


Fig. 7 Time consumption of different models

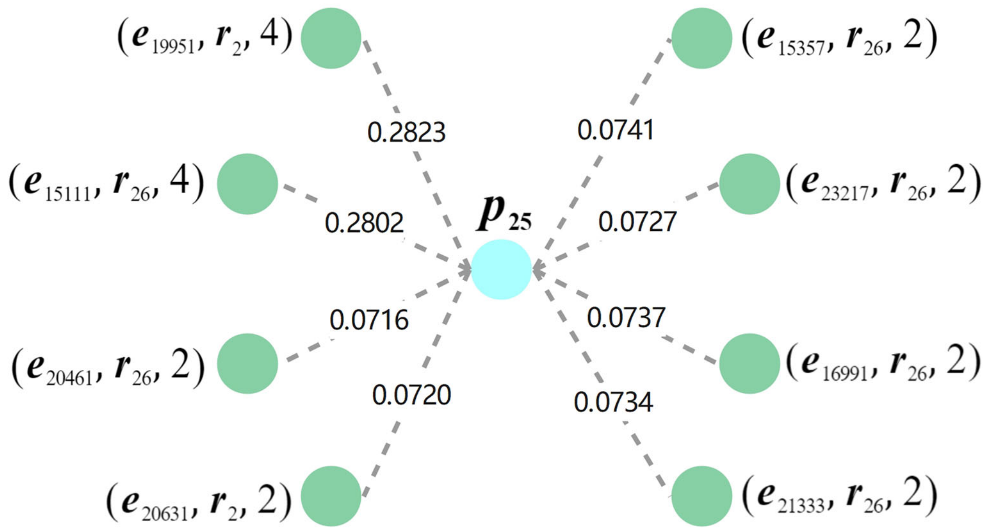


Fig. 8 The EUIG of user $\boldsymbol{u}_{25}$

5 Related work

The KG has been widely used in various scenarios, including recommender systems, question answering systems [33] etc. The KG stores real data from many fields in a heterogeneous structure, and can provide various additional item information for recommender systems. According to how to use the heterogeneous information in KG, the KG-based

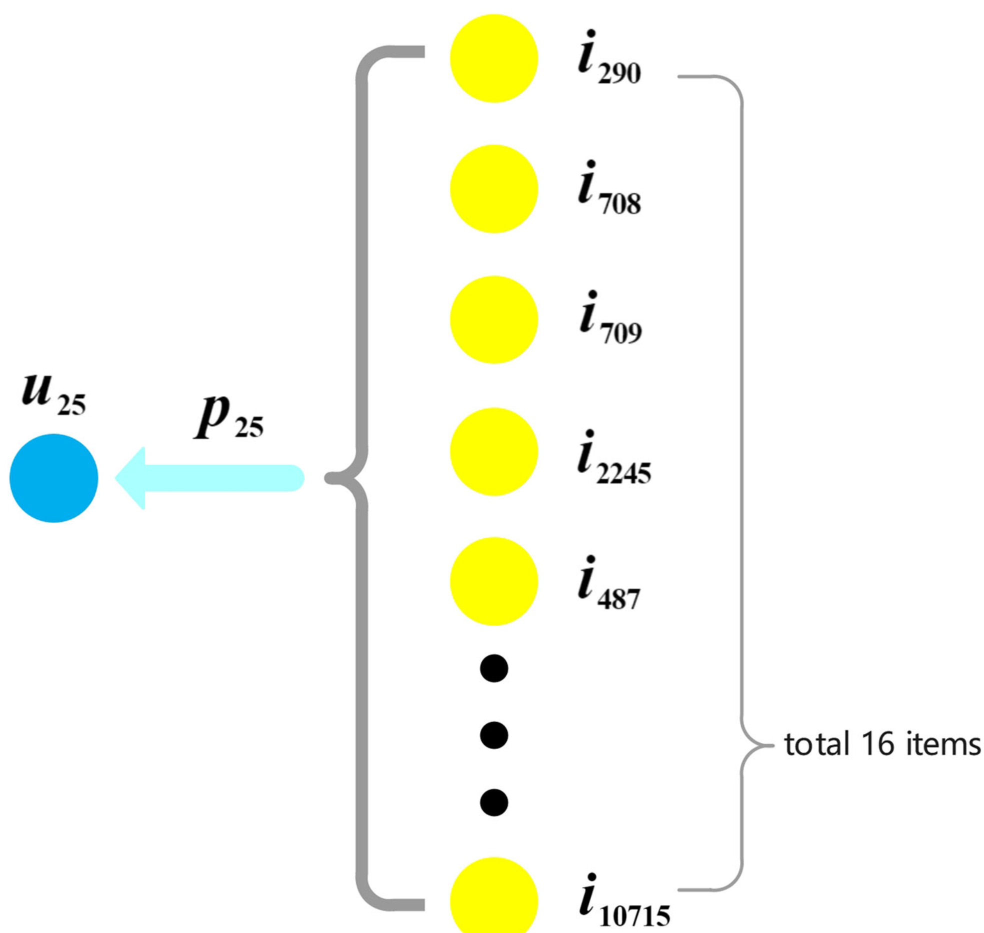


Fig. 9 The structure of GAT as related to user $\boldsymbol{u}_{25}$

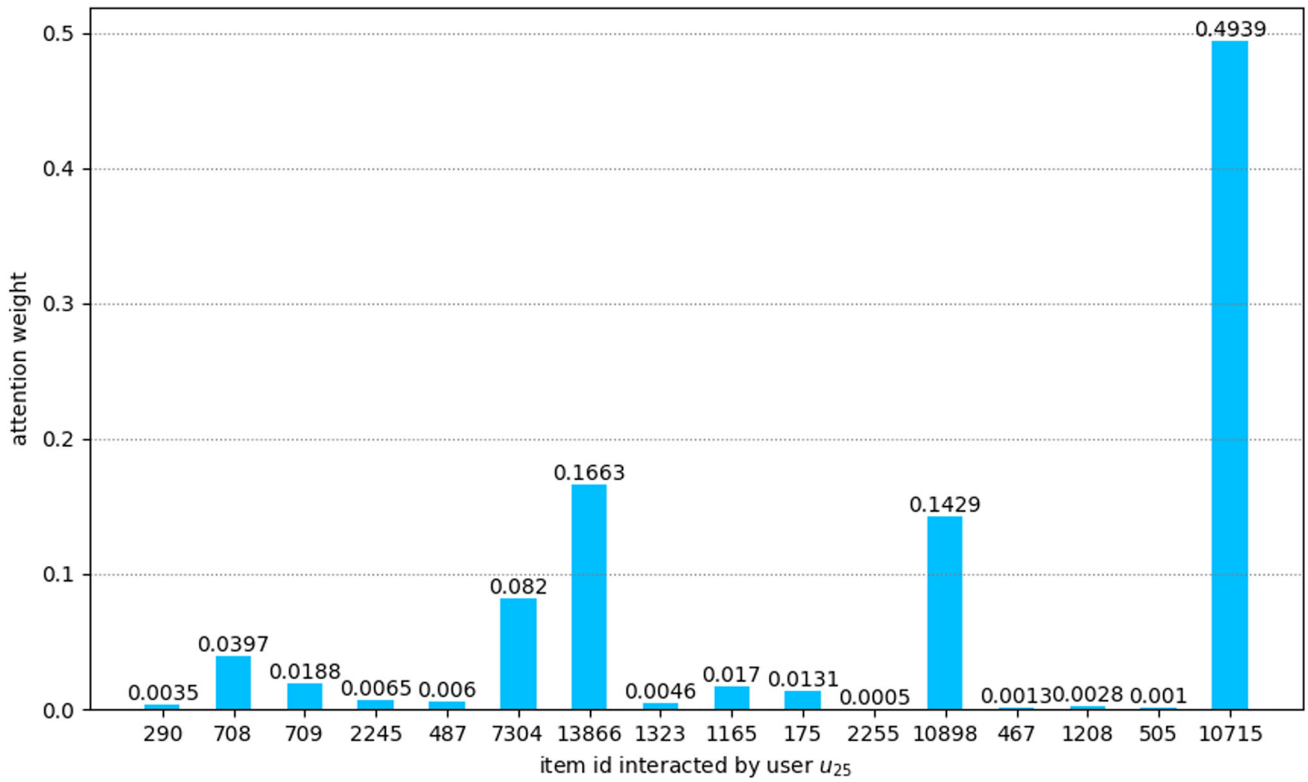


Fig. 10 The attention weights of GAT

recommendation (KGR) can be roughly segmented into three categories [3]: embedding-based KGR, connection-based KGR and propagation-based KGR.

5.1 Embedding-based KGR

The embedding-based KGR leverages the factual information in the KG to enhance the information contained in the user or item embeddings. This type of KGR model first uses some knowledge graph completion methods (such as TransE [35]) to encode entity and relation information into the embeddings. Then, these embeddings are leveraged to enrich the representation of users or items in recommender systems.Finally, recommendations are provided to users.

Wang et al. [34] proposed a KGR model named DKN for news recommendations. DKN first captures the entities in news by CNN and then utilizes the knowledge graph completion method TransD [35] to obtain the entities' embeddings in the KG. It integrates all entities that occur in the news to obtain the news embeddings. Zhang et al. [22] proposed an embedding-based KGR model named CKE. CKE uses TransR to encode entity information into embeddings and combines the embeddings with items to recommend for users. Cao et al. [19] proposed a model named KTUP , which includes the recommendation module and the knowledge graph completion module. KTUP utilizes TransH [35] in the knowledge graph completion module to encode entities and relations into embeddings, and constructs user preferences by relations in the KG. In the recommendation module, KTUP calculates the prediction score according to the user preferences. Then KTUP jointly trains two modulesextra embeddings to restrain the recommendation, it is diffcult to leverage the high-order relations between entities in a KG.

5.2 Connection-based KGR

The connection-based KGR mainly uses the connection relationships in KG to assist the recommender systems. This type of KGR model explores the relationships between entities in a KG. Some of the connection-based KGR models [36, 37] divide the KG into several subgraphs. According to these subgraphs, the relationship rules between entities are summarized to assist recommending for users. Other connection-based KGR models [38–40] mostly encode the relationships between items in the KG into embeddings, which can be fused into recommender systems.

Zhao et al. [37] proposed a connection-based KGR model named FMG. FMG generates the subgraphs from the KG, and extracts L different paths to capture the complex relationships between items. Then, FMG determines L

different user-item relationship matrices to obtain user and item embeddings. Ma et al. [39] proposed a rule-guided recommendation model named RuleRec to extract the explainable rules from the KG. RuleRec utilizes a rule learning module to exploit the rules between items and entities from the KG, and provides recommendation results guided by the rules. Although these connection-based KGR models can utilize the connection relationships in the KG to guide recommendations, some information is inevitably lost when the user-item or item-item rules are artificially designed.

5.3 Propagation-based KGR

To adequately explore the information in the KG, the propagation-based KGR has been proposed to aggregate the information of entities, relations and higher-order connections to obtain more comprehensive embeddings of users and items. In recent research on the propagation-based KGR, graph-based neural networks (GNNs) has been widely used. GNNs improve the representation of entities by aggregating the embeddings of multi-hop neighbours in the KG, which can exploit the heterogeneous structure of KG fully. Then the GNN-based models utilize the rich representation of items to recommend for users.

In recent years, GNN-based models [8, 16–18] have been widely studied and adopted and achieve the state-of-the-art performance. Wang et al. [27] proposed the propagation-based KGR model RippleNet, which propagates users' preference information in a KG to obtain rich representations of users. RippleNet uses relation matrices to distribute different weights to different neighbours in the propagation process. Through the propagation strategy in the KG, RippleNet can obtain information of triples in different propagation layers, and capture the comprehensive representation of users for recommendations. Wang et al. [10] introduced GNN into the KGR model and proposed KGCN to aggregate information in KG. In KGCN, the weights of neighbours not only depend on the relation information in triples but are also influenced by the user embeddings. In order to solve the over-fitting problem in KGCN, Wang et al. [16] proposed KGNN-LS, which utilizes a label smoothness algorithm to alleviate the over-fitting problem of KGCN. Wang et al. [8] proposed a GNN-based model KGAT. KGAT first uses TransR [35] to obtain the initialized embeddings of the entities and then aggregates the information of KG, which is similar to KGCN. After L-layer propagation, KGAT obtains the embeddins of users and items in each layer, and concatenates these embeddings to get the final representation of users and items. In order to explore information of user intents, Wang et al. [18] proposed a GNN-based model named KGIN. The user intents can be considered as the reason for the interaction between the users and the items. KGIN connects the information of user intents with the relations in the KG, and utilizes the relations information to model user intents. The mutual information maximization method is adopted to enrich the information contained in the embeddings of user intents. KGIN recommends for users according to the user intents and improves the accuracy of recommendation.

GNN has an impressive ability to process data with complex relations, which is suitable for dealing with the heterogeneous data contained in the KG. GNN improves the representation of entities by aggregating the embeddings of multi-hop neighbours in KG, and utilizes the rich representation of users and items to make recommendations for users. Recently, user intent inference methods [18, 19] have gradually become one of the research directions of GNN-based KGR. By capturing the information contained in KG to infer the user intents, the recommendation score of each item is calculated based on the information of the inferred intents, which can be used to make accurate recommendations for users.

6 Conclusion

In this paper, we propose a knowledge graph-based recommendation model EKIN. For the EKIN, we propose a novel user intent inference method, which can construct an *entity-driven user intent graph* (EUIG) for each userfrom both entities and relations. Then we adopt a graph neural network (GNN) with multi-hop propagation in the KG and EUIG to learn the comprehensive representation of entities, items and user intents. By selectively aggregating the users' interactionsdemonstrate the effectiveness of the proposed EKIN.

In future work, we will conduct an in-depth study on the imbalance issue between positive and negative samples, which is commonly encountered in KGR. Since there are only positive samples in the datasets, we will adopt some negative sampling strategies to acquire some informative negative samples, which can assist the recommendation model in capturing the features of users and items. It can improve the accuracy of recommendation.

Acknowledgements This work is supported by the National Natural Science Foundation of China under Project No. 61977013.

References

1. Ji S, Pan S et al (2022) A survey on knowledge graphs: Representation, acquisition, and applications[J]. IEEE Trans Neural Netw Learn Syst (TNNLS) 33(2):494–514. https://doi.org/10.1109/TNNLS.2021.3070843

2. Wang Q, Mao Z et al (2017) Knowledge graph embedding: A survey of approaches and applications[J]. IEEE Trans Knowl Data Eng (TKDE) 29(12):2724–2743. https://doi.org/10.1109/TKDE.2017.2754499
3. Guo Q, Zhuang F et al (2020) A survey on knowledge graph-based recommender systems[J]. IEEE Trans Knowl Data Eng (TKDE). https://doi.org/10.1109/TKDE.2020.3028705
4. Hsu PY, Chen CT et al (2022) Explainable mutual fund recommendation system developed based on knowledge graph embeddings[J], Applied Intelligence. pp 1–26. https://doi.org/10.1007/s10489-021-03136-1
5. Wu Z, Pan S et al (2021) A comprehensive survey on graph neural networks[J]. IEEE Trans Neural Netw Learn Syst (TNNLS) 32(1):4–24. https://doi.org/10.1109/TNNLS.2020.2978386
6. Zhang Z, Cui P et al (2022) Deep learning on graphs: a survey[J]. IEEE Trans Knowl Data Eng (TKDE) 34(1):249–270. https://doi.org/10.1109/TKDE.2020.2981333
7. Hamilton W, Ying R et al (2017) Inductive representation learning on large graphs[C]. In: Proceedings of the 31st conference on neural information processing systems (NIPS), pp 1025–1035
8. Wang X, He X et al (2019) KGAT: Knowledge graph attention network for recommendation[C]. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining (KDD), pp 950–958. https://doi.org/10.1145/3292500.3330989
9. Zhang Z, Zhang L et al (2022) KRAN: Knowledge refining attention network for recommendation[J]. ACM Trans Knowl Discov Data Mining (TKDD) 16(2):39. https://doi.org/10.1145/3470783
10. Wang H, Zhao M et al (2019) Knowledge graph convolutional networks for recommender systems[C]. In: Proceedings of the world wide web conference (WWW), pp 3307–3313. https://doi.org/10.1145/3308558.3313417
11. Tian Y, Yang Y et al (2021) Joint knowledge pruning and recurrent graph convolution for news recommendation[C]. In: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval (SIGIR), pp 51–60. https://doi.org/10.1145/3404835.3462912
12. Sang L, Xu M et al (2021) Knowledge graph enhanced neural collaborative recommendation[J]. Expert Syst Appl 164:113992. https://doi.org/10.1016/j.eswa.2020.113992
13. Hui B, Zhang L et al (2022) Personalized recommendation system based on knowledge embedding and historical behavior[J]. Appl Intell 52(1):954–966. https://doi.org/10.1007/s10489-021-02363-w
14. Wang X, Xu Y et al (2020) Reinforced negative sampling over knowledge graph for recommendation[C]. In: Proceedings of the world wide web conference (WWW), pp 99–109. https://doi.org/10.1145/3366423.3380098
15. Chen C, Zhang M et al (2020) Jointly non-sampling learning for knowledge graph enhanced recommendation[C]. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval (SIGIR), pp 189–198. https://doi.org/10.1145/3397271.3401040
16. Wang H, Zhang F et al (2019) Knowledge-aware graph neural networks with label smoothness regularization for recommender systems[C]. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining (KDD), pp 968–977. https://doi.org/10.1145/3292500.3330836
17. Wang Z, Lin G et al (2020) CKAN: Collaborative knowledge-aware attentive network for recommender systems[C]. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval (SIGIR), pp 219–228. https://doi.org/10.1145/3397271.3401141
18. Wang X, Huang T et al (2021) Learning intents behind interactions with knowledge graph for recommendation[C]. In: Proceedings of the world wide web conference (WWW), pp 878–887. https://doi.org/10.1145/3442381.3450133
19. Cao Y, Wang X et al (2019) Unifying knowledge graph learning and recommendation: towards a better understanding of user preferences[C]. In: Proceedings of the world wide web conference (WWW), pp 151–161. https://doi.org/10.1145/3308558.3313705
20. Chen W, Huang P et al (2019) POG: personalized outfit generation for fashion recommendation at Alibaba iFashion[C]. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining (KDD), pp 2662–2670. https://doi.org/10.1145/3292500.3330652
21. Rendle S, Freudenthaler C et al (2009) BPR: Bayesian personalized ranking from implicit feedback[C]. In: Proceedings of the 25th conference on uncertainty in artificial intelligence (UAI), pp 452–461
22. Zhang F, Yuan N et al (2016) Collaborative knowledge base embedding for recommender systems[C]. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (KDD), pp 353–362. https://doi.org/10.1145/2939672.2939673
23. Vaswani A, Shazeer N et al (2017) Attention is all you need[C]. In: Proceedings of the 31st annual conference on neural information processing systems (NIPS), pp 6000–6010
24. Brauwers G, Frasincar F et al (2021) A general survey on attention mechanisms in deep learning[J], IEEE Transactions on Knowledge and Data Engineering (TKDE). https://doi.org/10.1109/TKDE.2021.3126456
25. He X, Liao L et al (2017) Neural collaborative filtering[C]. In: Proceedings of the world wide web conference (WWW), pp 173–182. https://doi.org/10.1145/3038912.3052569
26. Covington P, Adams J et al (2016) Deep neural networks for YouTube recommendations[C]. In: Proceedings of the 10th ACM Conference on Recommender Systems (RecSys), pp 191–198. https://doi.org/10.1145/2959100.2959190
27. Wang H, Zhang F et al (2018) RippleNet: Propagating user preferences on the knowledge graph for recommender systems[C]. In: Proceedings of the 27th ACM international conference on information and knowledge management (CIKM), pp 417–426. https://doi.org/10.1145/3269206.3271739
28. Wang S, Qiu J et al (2021) A deep neural network model for fashion collocation recommendation using side information in e-commerce[J]. Appl Soft Comput 110:107753. https://doi.org/10.1016/j.asoc.2021.107753
29. Ding Y, Ma Y et al (2021) Leveraging two types of global graph for sequential fashion recommendation[C]. In: Proceedings of the 2021 international conference on multimedia retrieval (ICMR), pp 73–81. https://doi.org/10.1145/3460426.3463638
30. Wang H, Zhang F et al (2019) Multi-task feature learning for knowledge graph enhanced recommendation[C]. In: Proceedings of the World Wide Web Conference (WWW), pp 2000–2010. https://doi.org/10.1145/3308558.3313411
31. Huang W, Wu J et al (2022) Cross attention fusion for knowledge graph optimized recommendation[J], Applied Intelligence. https://doi.org/10.1007/s10489-021-02930-1
32. Paszke A, Gross S et al (2019) PyTorch: An imperative style, high-performance deep learning library[C]. In: Proceedings of the 33rd international conference on neural information processing systems (NIPS), pp 8026–8037
33. Huang X, Zhang J et al (2019) Knowledge graph embedding based question answering[C]. In: Proceedings of the 12th ACM International Conference on Web Search and Data Mining (WSDM), pp 105–113. https://doi.org/10.1145/3289600.3290956
34. Wang H, Zhang F et al (2018) DKN: Deep knowledge-aware network for news recommendation[C]. In: Proceedings of the World Wide Web Conference (WWW), pp 1835–1844. https://doi.org/10.1145/3178876.3186175

35. Ji G, He S, Xu L et al (2015) Knowledge graph embedding via dynamic mapping matrix[C]. In: Proceedings of the 53rd annual meeting of the association for computational linguistics (ACL) and the 7th international joint conference on natural language processing, pp 687–696
36. Shi C, Zhang Z et al (2015) Semantic path based personalized recommendation on weighted heterogeneous information networks[C]. In: Proceedings of the 24th ACM international on conference on information and knowledge management (CIKM), pp 453–462. https://doi.org/10.1145/2806416.2806528
37. Zhao H, Yao Q et al (2017) Meta-graph based recommendation fusion over heterogeneous information networks[C]. In: Proceedings of the 23rd ACM SIGKDD International conference on knowledge discovery and data mining (KDD), pp 635–644. https://doi.org/10.1145/3097983.3098063
38. Wang X, Wang D et al (2019) Explainable reasoning over knowledge graphs for recommendation[C]. In: Proceedings of the 33rd AAAI Conference on artificial intelligence and 31st Innovative applications of artificial intelligence conference and 9th AAAI Symposium on educational advances in artificial intelligence (AAAI/IAAI/EAAI), pp 5329–5336. https://doi.org/10.1609/aaai.v33i01.33015329
39. Ma W, Zhang M et al (2019) Jointly learning explainable rules for recommendation with knowledge graph[C]. In: Proceedings of the world wide web conference (WWW), pp 1210–1221. https://doi.org/10.1145/3308558.3313607
40. Xian Y, Fu Z et al (2019) Reinforcement knowledge graph reasoning for explainable recommendation[C]. In: Proceedings of the 42nd International ACM SIGIR conference on research and development in information retrieval (SIGIR), pp 285–294. https://doi.org/10.1145/3331184.3331203

Shaosong Li received the B.Eng. degree in computer science from University of Electronic Science and Technology of China, Chengdu, China, in 2020. He is currently pursuing M.Eng. degree with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include recommender systems, data mining, and information retrieval.

Bo Yang received the B.Eng. and M.Eng. degrees in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 1995 and 1998, respectively, and the Ph.D. degree in software quality and reliability engineering from the National University of Singapore, Singapore, in 2002. He has been a Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China, since 2008. He has published over 90 quality research papers in international journals and conferences. Based on Google Scholar, his H-index is 24 and i10-index is 38. His current research interests include machine learning, data mining, and cloud computing. Prof. Yang served as the General Chairs of IEEE 20th International Conference on Dependable, Autonomic, and Secure Computing (DASC) in 2022, IEEE 6th International Conference on Computer and Communication Systems (ICCCS) in 2021 and International Workshop on Knowledge and Data Engineering in Web-Based Learning in 2010; the Program Chairs of IEEE 8th, 18th, and 19th DASC in 2009, 2020, and 2021, IEEE 4th and 5th ICCCS in 2019 and 2020, respectively, and International Conference on Cloud and Service Computing in 2011; the Program Vice-Chair of the IEEE 12th International Conference on High Performance and Communications in 2010; the Publicity Chair of DASC 2019; and the program committee member/technical program committee member for more than 40 other conferences/workshops. He has been on the Editorial Board of *Journal of Cloud Computing-Advances, Systems and Applications* since 2012. He has been a Senior Member of IEEE since 2013. He has been a member of the China Computer Federation Technical Committee on Collaborative Computing since 2011.

Dongsheng Li obtained Ph.D. from School of Computer Science of Fudan University, China, in 2012. His research interests include recommender systems and general machine learning applications. He has been a principal research manager with Microsoft Research Asia (MSRA) since February 2020. Before joining MSRA, he was a research staff member with IBM Research – China since April 2015. He is also an adjunct professor with School of Computer Science, Fudan University, Shanghai, China. In April 2018, he won one of the highest technical awards in IBM – the IBM Corporate Award.