The Computer Journal, 2023, 1–12
https://doi.org/10.1093/comjnl/bxad085
Original Article

Knowledge-Aware Dual-Channel Graph Neural Networks For Denoising Recommendation

Hanwen Zhang[1], Li-e Wang[1,2,3,*], Zhigang Sun[1,2,3] and Xianxian Li[1,2,3]

[1]College of Computer Science and Engineering, Guangxi Normal University, No. 15, Yucai Road, Guilin 541004, China
[2]Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, No. 15, Yucai Road, Guilin 541004, China
[3]Guangxi Key Lab of Multi-source Information Mining and Security, Guangxi Normal University, No. 15, Yucai Road, Guilin 541004, China
*Corresponding author: wanglie@gxnu.edu.cn

Abstract

Knowledge graph (KG) is introduced as side information into recommender systems, which can alleviate the sparsity and cold start problems in collaborative filtering. Existing studies mainly focus on modeling users' historical behavior data and KG-based propagation. However, they have the limitation of ignoring noise information during recommendation. We consider that noise exists in two parts (i.e. KG and user-item interaction data). In this paper, we propose Knowledge-aware Dual-Channel Graph Neural Networks (KDGNN) to improve the recommendation performance by reducing the noise in the recommendation process. Specifically, (1) for the noise in KG, we design a personalized gating mechanism, namely dual-channel balancing mechanism, to block the propagation of redundant information in KG. (2) For the noise in user-item interaction data, we integrate personalized and knowledge-aware signals to capture user preferences fully and use personalized knowledge-aware attention to denoise user-item interaction data. Compared with existing KG-based methods, we aim to propose a knowledge-aware recommendation method from a new perspective of denoising. We perform performance analysis on three real-world datasets, and experiment results demonstrate that KDGNN achieves strongly competitive performance compared with several compelling state-of-the-art baselines.

Keywords: knowledge graph, recommender systems, noise information, dual-channel graph neural networks

1. INTRODUCTION

In recent years, recommender systems (RS) have been widely used in various fields, such as news [1], movies [2] and e-commerce [3]. By capturing users' historical interaction data and mining their interest preferences, it can actively provide users with the content of interest. Among them, the methods of collaborative filtering (CF) [4–6] are the most widely used. However, CF-based methods usually suffer from data sparsity and cold-start problems. A widely used solution is to integrate various side information, such as knowledge graph (KG), which contains rich facts and relations. Researchers favor KG-based approaches as they can well alleviate the sparsity and cold start problems and improve the performance of RS.

There have been many studies on KG-based recommendation, aiming to learn high-quality user and item representations from the rich facts and semantic information in KG. Most early studies [7–9] dealt with triples in KG independently, ignoring potentially valuable information about the local neighborhood around the entity. Several subsequent studies [10–12] treat KG as a heterogeneous information network, exploiting various connectivity patterns between items to enrich user-item interactions. However, they heavily rely on manually designed meta-paths, which are difficult to implement. With the rise of graph neural network (GNN) [13, 14] research, GNN-based methods [15–21] can be a good remedy for the shortcomings of previous studies. The key idea is to recursively propagate information among multi-hop nodes to enrich the representation of nodes. These solutions achieve good recommendation performance because they can effectively integrate multi-hop neighbors into the representation.

Some recent work [20, 21] briefly mentions the problem of noise. CG-KGR [20] mentioned that the construction of KGs is independent of the collection of historical user-item interactions and the information in these KGs is not necessarily helpful for all user recommendations. KGIC [21] mentioned that the combination of sparse interactions and redundant KG facts further leads to an imbalance in information utilization. However, they did not make a specific study on the problem of noise in recommendations. We describe the proposed noise in this paper. Specifically, noise exists in user-item interaction data and KG. From a coarse-grained perspective, items that users click on by mistake or dislike after browsing are called noise. From a fine-grained perspective, users often have different preferences for interactive items, but they are treated equally in the user modeling process, which makes the user modelings of features inaccurate. Thus, those interaction items that are not liked by users are also regarded as noise. As shown in Fig. 1, the user interacts with four movies, three of which belong to the science fiction genre, and we can speculate that the user's interest is more inclined to the science fiction genre. Compared with Titanic, which belongs to the romance genre, the user prefers movies in the science fiction genre, such as Star Wars and The Terminator. Similarly, user's interest in movies under the same genre (e.g. science fiction) is not identical. So, treating each item equally will introduce noise to the modeling of the user. In KG, GNN aggregates the neighborhood knowledge of items, and the neighborhood knowledge information of noise

Received: December 30, 2022. **Revised:** June 20, 2023. **Accepted:** July 15, 2023

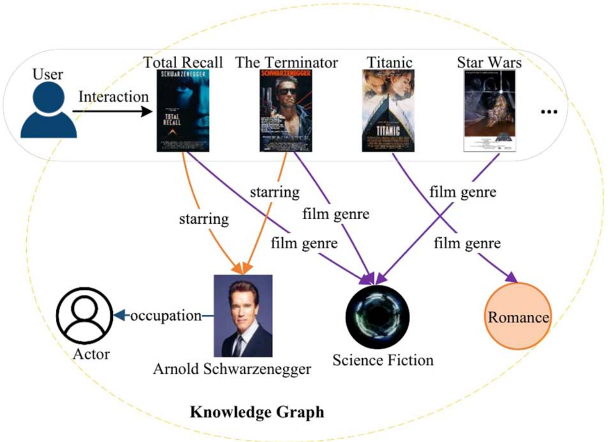


Figure 1. An illustrative example showing that user-related items reflect different levels of user preferences. Assuming each item is viewed to equivalently reflect user preferences, this could introduce noise to the user representation modeling, as the user shows a clear preference for science fiction genre movies.

items (i.e. items that users click on by mistake or dislike after browsing) is called noise. In dual-channel GNN, the aggregated unbalanced redundant information is called noise. Unlike other studies that focus only on node aggregation, we consider the fact that semantic relations can reflect user preferences to some extent. However, simple aggregation of node embeddings and edge embeddings (e.g. node embedding plus edge embedding) faces the problem of information imbalance because it is impossible to determine which nodes are more important than edges or which edges are more important than nodes.

To solve the above problems, we aim to design a knowledge-aware denoising recommendation method. Our design goal is to effectively reduce the noise in the recommendation process and achieve fine-grained modeling of the user to improve recommendation performance. Therefore, this paper proposes Knowledge-aware Dual-Channel Graph Neural Networks (KDGNN) for Denoising Recommendation. Specifically, unlike most methods that only propagate node information, we design a Dual-Channel GNN that propagates node and edge information separately, and uses a novel dual-channel balancing mechanism to adaptively adjust the weight between entities(nodes) and relations(edges) to control the impact of redundant noise. The purpose is to maximize the capture of users' potential interest and block the propagation of redundant information. In addition, since an item's contextual knowledge information can reflect the user's fine-grained preferences for the item, we construct a knowledge entity set for each user by dual-channel GNN, where the knowledge entity fuse item(entity) and contextual knowledge information. The knowledge entities and learnable embedding of the user are considered two signals representing the user and are fused to capture user preferences fully. Finally, we use personalized knowledge-aware attention to assign weights to items to realize the denoising of the user-item interaction data and fine-grained modeling of user representation. We conduct extensive experiments on three datasets, and the experimental results show that our method outperforms state-of-the-art methods.

In summary, the contributions of this paper are as follows:

- We design a novel dual-channel GNN, which can propagate both node and edge information and adaptively balance the contributions of both to prevent the propagation of redundant noise in KG.
- From a new perspective of denoising, we propose a KG-based model, KDGNN, which can comprehensively capture the users' potential preferences while effectively reducing the propagation of redundant information in KG and employs personalized knowledge-aware attention to reduce the noise of user-item interaction data, thus enhancing the representations of user and item to improve recommendation performance.
- Finally, we design the corresponding algorithm and conduct extensive experiments on three real datasets. The experimental results show that our method outperforms the convincing baseline.

2. RELATED WORK

Embedding-based methods [1, 7–9] usually use knowledge graph embedding to preprocess KG, map entities and relations into a low-dimensional vector space to retain the structural information of the original KG and then use the learned entity embedding and relational embedding in recommendation tasks. For example, CKE [9] applied TransR [22] on KG triplets and fed the knowledge-aware embeddings of items into matrix factorization. KTUP [8] employed TransH [23] to learn relational embedding and entity embedding and transferred them to recommendation tasks while training KG completion and recommendation tasks to achieve mutual enhancement. However, most of these methods deal independently with triples in KG. They do not utilize valuable higher order information from around the entity, which limits the extent of user preference mining for items.

Path-based methods [10–12, 24] explore various connectivity patterns between items in KG by constructing meta-paths of information propagation to provide additional guidance and interpretability for recommendations. Personalized Entity Recommendation (PER) [24] treated KG as a heterogeneous information network and extracted potential features based on meta-paths to represent the connections between users and items along different relational paths. Policy-guided path reasoning (PGPR) [12] formulated the recommendation problem as a Markov decision process, using reinforcement learning to search for reasonable paths between user-item pairs. These methods capture the higher order knowledge of items in KG and provide relatively superior performance. However, they rely heavily on domain knowledge and manually designed meta-paths, which are often labor-intensive, especially in complex KGs.

GNN-based methods [15–21] use the information aggregation mechanism of GNN to enrich entity representation by aggregating multi-hop neighbor embeddings in KG to capture entity features and graph structure to model remote connections. KGCN [15] and KGNN-LS [16] used graph convolution algorithm to aggregate neighbor entities to obtain richer entity representation and higher order personalized interests of users. KGAT [17] combined a bipartite graph of user-item interactions and KG into a collaborative knowledge graph (CKG). It explicitly models higher-order connections in the KG in an end-to-end manner, which recursively propagates embeddings from node neighbors and employs an attention mechanism to distinguish the importance of neighbors. CKAN [18] provided a new method of combining collaborative information with knowledge information together. KGIN [19] considered the attention composition of relations in KG as the intention behind user interaction items and proposed relational path-aware aggregation to preserve the overall semantics of relational paths.

CG-KGR [20] explicitly propagated collaborative information in user-item interactions to profile their latent representations. Based on this latent summarization, it seamlessly fuses this collaborative encoding as guidance to customize the knowledge extraction from external KG. KGIC [21] incorporated contrastive learning into KG-aware recommendation, making sufficient and coherent use of CF and KG in a self-supervised manner. However, most aggregation schemes for knowledge-aware recommendation rarely consider the problem of redundant information in the propagation process, let alone how to reasonably aggregate node and relational features. In addition, the above methods do not consider that users' behavioral data inevitably contain noise, which negatively affects the modeling of user preferences. Our work differs from the above in that we propose a dual-channel GNN that can propagate both node and edge information, and design a dual-channel balancing mechanism to reduce the propagation of redundant noise. Moreover, we introduce personalized and knowledge-aware signals to distinguish the noise in user-item interaction data. Overall, we enhance the representations of user and item based on the rich semantic information in KG, and denoise the recommendation process to improve the recommendation performance.

3. PROBLEM FORMULATION

In this section, we formulate the KG-based recommendation problem as follows.

Interaction Data. In a typical recommendation scenario, let $\mathcal{U} = \{u_1, u_2, \dots, u_M\}$ denote the set of all users, and $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ denote the set of all items. The user-item interaction matrix $\mathbf{Y} \in \mathbb{R}^{M \times N}$ is defined according to users' implicit feedback, where $y_{uv} = 1$ indicates that user u engages with item v, such as clicking, browsing or purchasing; otherwise $y_{uv} = 0$.

Knowledge Graph. Besides historical interaction records of users, a knowledge graph $\mathcal{G}$ is introduced for providing item side information. KG describes real-world facts in the form of triples, denoted as $\{(h, r, t) \mid h, t \in \mathcal{E}, r \in \mathcal{R}\}$, where $\mathcal{E}$ is the set of entities and $\mathcal{R}$ is the set of relations. Here h, r and t denote the head entity, relation and tail entity in the triplet, respectively. For example, the triplet (***RobertZemeckik***, **isDirectorOf**, ***ForrestGump***) indicates the fact that ***Robert Zemeckis*** directed the film ***ForrestGump***. Furthermore, an item $v \in \mathcal{V}$ corresponds to an entity $e \in \mathcal{E}$. For example, in the movie recommendation, the item ***BeautifulLife*** is associated with the entity of the same name in the KG. Therefore, we build a set of item-entity alignments $\mathcal{A} = \{(v, e) \mid v \in \mathcal{V}, e \in \mathcal{E}\}$, where (v,e) denotes the alignment of item v with entity e.

Problem Statement. Given a user-item interaction matrix $\mathbf{Y}$ and a knowledge graph $\mathcal{G}$, our KG-based recommendation task is to predict the probability that a user u will interact with an item v in which he/she has not previously participated. Specifically, our goal is to learn a prediction function $\hat{y}_{uv} = \mathcal{F}(u, v \mid \Theta, \mathbf{Y}, \mathcal{G})$, where $\hat{y}_{uv}$ denotes the probability that user u will engage with item v, and Θ denotes the model parameters of function $\mathcal{F}$.

4. METHODOLOGY

In this section, we describe in detail KDGNN. As shown in Fig. 2, the model framework consists of two main components: (1) Dual-Channel GNN (DCGNN), which propagates node(entity) and edge(relation) information separately, and adaptively adjusts the balance of entities and relations to achieve the purpose of controlling noise propagation in KG. (2) Personalized Knowledge-aware Attention Network, which comprehensively models user preferences from both the learnable embedding aspect of the user (i.e. user's personalized signal) and the fine-grained aspect of KG (i.e. knowledge-aware signal) and exploits personalized knowledge-aware attention to assign weights to items, so as to achieve the purpose of denoising user-item interaction data. KDGNN obtains the final high-quality representations of users and items and then outputs the predicted click probability.

4.1. Dual-channel GNN

As shown in Fig. 2, the dual-channel GNN consists of entity channel, relational channel and dual-channel balancing mechanism. The entity channel propagates contextual entity features of items. The relational channel propagates contextual relation features of items, and the dual-channel balancing mechanism balances the contributions of both. Next, we will discuss each of these components in detail.

4.1.1. *Entity channel and relational channel*

Inspired by RippleNet [25], DKN [1] and CKAN [18], items that the user has interacted with historically can directly represent user preferences. By aligning the relevant items with the entities in KG, the contextual neighborhood information of the entities (i.e. items) can reflect the fine-grained preferences of users for their interaction items. Therefore, to enhance the ability to model user interests, we extract contextual entities encoded in KG in the entity channel to enrich the representation of all historical interaction items.

Traditional GNN-based approaches usually consider information associated with nodes, which does not apply to KG where edge features (types of relations) are more important. For example, suppose a user watches multiple movies of the same genre. In that case, we can infer that the semantic relation of 'movie genre' plays a key role in the user's decision-making, which helps to understand the user preferences more finely. Similar to the entity channel, we extract contextual relations encoded in KG in the relational channel to enrich the representation of all historical interaction items.

Inspired by GNN, we capture the set of neighboring entities and relations of item(entity) v in KG. As the number of neighbors of a node can vary from one to a thousand or even more, it is inefficient to take the full size of a node's neighborhood [26]. GraphSage [13] adopted sampling to obtain a fixed number of neighbors for each node. Following KGCN[15], KGNN-LS[16] and CKAN[18], to keep the computational pattern of each batch fixed and more efficient, we sample a uniform set of fixed-size neighbors (entities and relations) for each entity instead of using its full neighbors, e.g. $|N_v^o| = K$. Note that the symbol o is a uniform placeholder for symbol e or r, since the calculation and formulation for entity and relation in this article are similar in many scenarios. As neighboring entities and relations have different importance to the items, we use knowledge-aware attention to calculate the attention score between item and neighboring information.

$$s(v, o) = \frac{\exp\left(\mathbf{x}_v^\top \mathbf{x}_o\right)}{\sum_{o' \in N_v^o} \exp\left(\mathbf{x}_v^\top \mathbf{x}_{o'}\right)}, \tag{1}$$

where N_v^o is the set of neighboring entities or relations directly connected to item v, $\mathbf{x}_v$ is the representation of the item v, $\mathbf{x}_o$ is the representation of neighboring entity e or neighboring relation r. To capture the neighboring topology of item v, we refine a neighborhood representation vector $\mathbf{c}_o$ according to the following

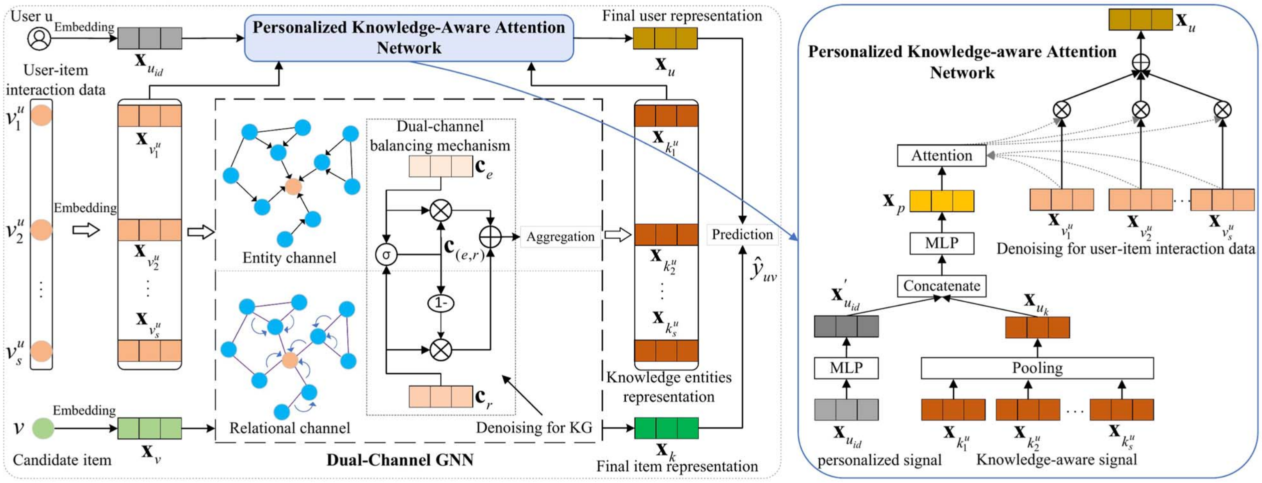


Figure 2. The overall architecture of the proposed KDGNN model.

equation.

$$\mathbf{c}_o = \sum_{o \in N_v^o} s(v, o)\mathbf{x}_o, \tag{2}$$

where $\mathbf{c}_o$ can be a neighborhood entity representation $\mathbf{c}_e$ for the entity channel output or a neighborhood relation representation $\mathbf{c}_r$ for the relational channel output.

4.1.2. *Dual-channel balancing mechanism*

It is a challenge to reasonably aggregate the information in the entity channel and the relational channel. Because simply treating the contributions of the entity channel and the relational channel as fixed and not distinguishing the contributions of both may bring redundant noise. Inspired by GRU[27], we design a dual-channel balancing mechanism to adaptively balance the contributions of both. Given neighborhood entity representation $\mathbf{c}_e$ and neighborhood relation representation $\mathbf{c}_r$, we combine them to generate a representation $\mathbf{c}_{(e,r)}$ of contextual knowledge information.

$$g = \sigma\left(\mathbf{W}_e\mathbf{c}_e + \mathbf{W}_r\mathbf{c}_r\right) \tag{3}$$

$$\mathbf{c}_{(e,r)} = g \cdot \mathbf{c}_e + (1 - g) \cdot \mathbf{c}_r, \tag{4}$$

where $\mathbf{W}_e$ and $\mathbf{W}_r$ are the trainable weight matrices, σ is the nonlinear function sigmoid and g is the learnable gating signal.

Finally, we aggregate the representation of the current item with its representation of contextual knowledge information as the representation of the knowledge entity.

$$\mathbf{x}_v^{(l)} = agg\left(\mathbf{x}_v^{(l-1)}, \mathbf{c}_{(e,r)}^{(l-1)}\right), \tag{5}$$

$$\mathbf{x}_k = \mathbf{x}_v^{(l)}, \tag{6}$$

where l denotes the depth of layers, $agg(.)$ is aggregation function and $\mathbf{x}_k$ is the representation of knowledge entity.

Following KGCN[15], we achieve this through the following three aggregation approaches.

- Sum aggregation

$$agg_{\text{sum}} = \sigma\left(\mathbf{W}\left(\mathbf{x}_v + \mathbf{c}_{(e,r)}\right) + \mathbf{b}\right), \tag{7}$$

where $\mathbf{W}$ and $\mathbf{b}$ are trainable weight and bias, σ is the nonlinear function sigmoid.

- Concat aggregation

$$agg_{\text{concat}} = \sigma\left(\mathbf{W}\left(\mathbf{x}_v \| \mathbf{c}_{(e,r)}\right) + \mathbf{b}\right), \tag{8}$$

where $\|$ is the concatenation operation.

- Neighbor aggregator

$$agg_{\text{neighbor}} = \sigma\left(\mathbf{W} \cdot \mathbf{c}_{(e,r)} + \mathbf{b}\right). \tag{9}$$

We will evaluate our model by the three aggregators mentioned above in the experimental section.

4.2. Personalized knowledge-aware attention network

4.2.1. *Knowledge entity set*

Although items that the user has interacted with in the history can directly represent the user preferences, existing methods ignore the noise of user behavior data, which directly affects the recommendation quality. In addition, this approach only focuses on the user's explicit interest in interactive items without considering the contextual knowledge information of interactive items because the contextual knowledge in KG can reflect the user's fine-grained interest in the item, which helps understand the user preferences in a more detailed way. Therefore, to capture user preferences comprehensively and accurately, we use KG to construct a knowledge entity set K^u for each user. Knowledge entity fuses the entity and its contextual knowledge information, which can be generated by a dual-channel GNN. A user's knowledge entity set K^u consists of the knowledge entities that he/she has interacted with, i.e. $K^u = \left\{k_1^u, k_2^u, \ldots, k_i^u, \ldots, k_s^u\right\}$, and k_i^u denotes the i-th knowledge entity in the user's knowledge entity set. Since each user interacts with a different number of items, we set the size of the knowledge entity set according to the average number s of user interaction items, i.e. $|K^u| = s$.

4.2.2. *Personalized knowledge-aware attention*

Unlike KGCN [15] we use independent potential vectors to represent the user, as shown in Fig. 2, and view the learnable embedding of the user and the user's associated knowledge entities as two kinds of signals representing the user. On the one hand, the learnable embedding of the user is directly generated through the Embedding Layer, which can provide personalized signal to the user. On the other hand, each user can be represented by the embedding of its corresponding knowledge entities, which is generated by the mean pooling of the knowledge entities in the knowledge entity set.

$$\mathbf{x}_{u_k} = pool_{mean}\left(\mathbf{x}_{k_1^u}, \mathbf{x}_{k_2^u}, \dots \mathbf{x}_{k_i^u}, \dots, \mathbf{x}_{k_s^u}\right), \tag{10}$$

$$\mathbf{x}_{k_i^u} = \text{DCGNN}\left(\mathbf{x}_{v_i^u}\right), \tag{11}$$

where $pool_{mean}$ is a mean pooling operation, $\mathbf{x}_{k_i^u}$ is the representation of the knowledge entity k_1^u, $\mathbf{x}_{v_i^u}$ is the representation of the item v_i that user u has interacted with and DCGNN(.) is a dual-channel GNN.

Then, we integrate the learnable embedding of the user with the embedding of knowledge entities to model a high-quality representation of user preferences in both coarse-grained and fine-grained ways. Specifically, we input the learnable embedding of the user (i.e. $\mathbf{x}_{u_{id}}$) into the MLP and concatenate with the embedding of knowledge entities (i.e. $\mathbf{x}_{u_k}$) in the knowledge entity set to generate a vector. Finally, this vector is fed into the MLP to generate the final representation $\mathbf{x}_p$ of the user preference, which integrates personalized and knowledge-aware signals representing the user. The formula is as follows.

$$\mathbf{x}'_{u_{id}} = \sigma\left(\mathbf{W}_{id} \cdot \mathbf{x}_{u_{id}} + \mathbf{b}_{id}\right), \tag{12}$$

$$\mathbf{x}_p = \sigma\left(\mathbf{W}_c \cdot concat\left(\mathbf{x}'_{u_{id}}, \mathbf{x}_{u_k}\right) + \mathbf{b}_c\right), \tag{13}$$

where $\mathbf{W}_{id}, \mathbf{W}_c, \mathbf{b}_{id}$ and $\mathbf{b}_c$ are learnable weight parameters and $\mathbf{x}_{u_{id}}$ is learnable embedding of user.

To eliminate the impact of noise on user-item interaction data, we use personalized knowledge-aware attention to assign weights to each item for denoising. In other words, we measure the credibility of items for user preferences by calculating the similarity score between user preferences representation $\mathbf{x}_p$ and items. The denoising method not only attenuates the impact of noise from the items but also enhances the user-item interaction.

We calculate the similarity $s(p, v^u)$

$$s(p, v^u) = \frac{\exp\left(\mathbf{x}_p^\top \mathbf{x}_{v^u}\right)}{\sum_{v'^u \in \mathcal{V}^u} \exp\left(\mathbf{x}_p^\top \mathbf{x}_{v'^u}\right)}, \tag{14}$$

where $\mathcal{V}^u$ is the set of items of the user u history interaction, i.e. $\mathcal{V}^u = \{v_1^u, v_2^u, \dots, v_s^u\}$.

Finally, we assign weights based on user preference to the items with which the user has interacted with and generate a weighted representation of the items, thus reducing the effect of noise while modeling the user representation x_u in a fine-grained manner.

$$\mathbf{x}_u = \sum_{v^u \in \mathcal{V}^u} S(p, v^u) \cdot \mathbf{x}_{v^u}. \tag{15}$$

4.3. Learning algorithm

To predict user interaction with the candidate item, we generate final representations of the user and the candidate item and calculate the probability of the user interaction with the item. We obtain the user's final representation $\mathbf{x}_u$, generated in Section 4.2.2. Given a candidate item v, we enhance the representation of the candidate item by extracting the contextual knowledge encoded in KG.

$$\mathbf{x}_k = \text{DCGNN}(\mathbf{x}_v), \tag{16}$$

Finally, we conduct the inner product to calculate the probability that the user interacts with the candidate item.

$$\hat{y}_{uv} = \sigma(\mathbf{x}_u^\top \mathbf{x}_k), \tag{17}$$

where σ is sigmoid function.

In order to optimize the embedding and model parameters, balance the number of positive and negative samples and ensure the effect of model training, we extract the same number of negative samples as positive samples for each user, and we adopt the following loss function:

$$\begin{aligned}\mathcal{L} = \sum_{u \in \mathcal{U}} \Bigg(&\sum_{v \in \{v | (u,v) \in \mathcal{P}^+\}} \mathcal{J}(y_{uv}, \hat{y}_{uv}) \\ &- \sum_{v \in \{v | (u,v) \in \mathcal{P}^-\}} \mathcal{J}(y_{uv}, \hat{y}_{uv}) \Bigg) + \lambda ||\Theta||_2^2,\end{aligned} \tag{18}$$

where $\mathcal{J}$ is the cross-entropy loss, $\mathcal{P}^+$ means positive user-item pair set, while $\mathcal{P}^-$ is the opposite. Θ is the set of model parameters, the last term is the $L2$-regularizer.

Algorithm 1: KDGNN algorithm

Input : Interaction matrix $\mathbf{Y}$; knowledge graph $G(\mathcal{E}, \mathcal{R})$; Hyper-parameters: L, s, $\mathcal{V}^u$, $agg(.)$

Output: prediction function $\mathcal{F}(u, v \mid \Theta, \mathrm{Y}, \mathcal{G})$

1 **while** *KDGNN not converge* **do**
2   **for** $(u,\ v)$ *in* Y **do**
3   Sample s item v^u from the set $\mathcal{V}^u$;
4     $\mathbf{x}_{k^u} \leftarrow \text{DCGNN}(\mathbf{x}_{v^u}), \forall v^u \in \mathcal{V}^u$;
5   Obtain $\mathbf{x}_u$ according to Eq. (10) and Eq. (12)-(15);
6     $\mathbf{x}_k \leftarrow \text{DCGNN}(\mathbf{x}_v)$;
7   Calculate predicted probability $\hat{y}_{uv} = \sigma(\mathbf{x}_u^\top \mathbf{x}_k)$;
8   Update parameters Θ of $\mathcal{F}$ by gradient descent;
9   **end**
10 **end**
11 **Function** DCGNN$(\mathbf{x}_v)$
12   **for** $l = 1, \dots L$ **do**
13   Obtain $\mathbf{c}_e$ and $\mathbf{c}_r$ according to Eq. (1)-(2);
14   Obtain $\mathbf{c}_{e,r}$ according to Eq. (3)-(4);
15     $\mathbf{x}_v^{(l)} \leftarrow agg\left(\mathbf{x}_v^{(l-1)}, \mathbf{c}_{(e,r)}^{(l-1)}\right)$;
16   **end**
17   $\mathbf{x}_k \leftarrow \mathbf{x}_v^{(L)}$;
18   return $\mathbf{x}_k$
19 **end**

The learning algorithm of KDGNN is presented in Algorithm 1. L denotes the maximum number of layers of DCGNN, s denotes the average number of user interaction items and $\mathcal{V}^u$ is the set of items of the user u history interaction. For a given user-item pair (u, v) (line 2) and Interaction matrix $\mathbf{Y}$, we first randomly select s items from $\mathcal{V}^u$ and then input them into DCGNN to obtain s knowledge entities of the user (lines 3–4). DCGNN is able to propagate the rich contextual entity and relational features encoded in KG (line 13), and the dual-channel balancing mechanism prevents the propagation of redundant noise in KG (line 14). Then the representation of contextual knowledge is aggregated with itself as the input of the next layer (line 15). The final L layer is the representation of the knowledge entity (line 17). We integrate the learnable embedding of the user (i.e. user's personalized signal) with the embedding of knowledge entities (i.e. knowledge-aware signal) to comprehensively capture high-quality representation of user preferences and achieve denoising by measuring the confidence of items to user preferences, then generate a weighted representation of the items to model the final user representation (line 5). The representation of the candidate item is enhanced by a dual-channel GNN (line 6). Finally, the probability of the user interaction the item is calculated (line 7).

In terms of comprehensiveness, the method in this paper may have some limitations: (i) the proposed method does not guarantee that all relevant neighbors are sampled. This is because traversing all neighbors and using the attention mechanism to select the most relevant neighbors requires significant computational overhead. In addition, our denoising method can effectively reduce the impact of irrelevant and weakly relevant neighbors. Therefore, we designed a compromise method that combines random sampling and denoising to ensure recommendation performance while improving computational efficiency. (ii) The proposed method does not ensure that the s knowledge entities selected are the most indicative of user preferences. Similar to the previous neighbor sampling, selecting the most relevant items requires a large computational overhead. Therefore, we adopted a compromise method and minimized the impact of sampled irrelevant items on recommendation performance. We also verified that sampling s knowledge entities can achieve relatively good performance in experimental section.

5. EXPERIMENTS

In this section, we evaluate KDGNN on three real-world datasets. We first introduce datasets, baselines and experiments setup. Then we discuss the experimental results in detai.

5.1. Datasets

We use three benchmark datasets to verify the effectiveness of our model in different domains, such as music, movie and book. The three publicly available datasets vary in sparsity and size, making our model more convincing.

- **Last.FM**[1] is a music dataset containing ∼2000 users provided by the Last.fm online music system.
- **MovieLens-1M**[2] from the MovieLens website contains 6036 users and 2445 items, with about 1 million explicit ratings (on a scale of 1 to 5), and is widely used in movie recommendation.
- **Book-Crossing**[3] is collected from the book-crossing community and includes ratings (from 0 to 10) of books from various readers.

Since the interactions in Last.FM, Book-Crossing and MovieLens-1M are explicit feedback, we follow the approach of KGCN [15] and convert them to implicit feedback, where 1 denotes a positive sample, and draw an unmatched set marked as 0 for each user. The threshold of the rating to be viewed as positive is 4 for MovieLens-1M, while no threshold is set for Last.FM and Book-Crossing due to their sparsity.

Table 1. Basic statistics of the three datasets.

	Last.FM	Movielens-1M	Book-Crossing
# users	1872	6036	17 860
# items	3846	2445	14 967
# interactions	42 346	753 772	139 746
# avg-user-inter	23	125	8
# entities	9366	182 011	77 903
# relations	60	12	25
# KG triples	15 518	1241 995	151 500

As for the sub-KG construction, we follow the work in KGCN [15] and use Microsoft Satori[4] to construct for the Last.FM, Book-Crossing and MovieLens-1M datasets. Each triplet has a confidence level greater than 0.9. For simplicity, we exclude items with multiple matched or no mismatched entities. We then match the item IDs with the head of all triples and select all well-matched triples from the sub-KG. The detailed statistics for three datasets are shown in Table 1. avg-user-inter is the average number of interactions per user.

5.2. Baseline

To evaluate the performance of KDGNN, we compared it with the following state-of-the-art methods.

- **BPRMF** [28] is a classical CF method that optimizes the pairwise ranking between the positive and negative samples.
- **CKE** [9] is a classical embedding-based method that combines CF with structural, textual and visual knowledge in a unified recommendation framework.
- **RippleNet** [25] is a memory-network-like approach which represents the user by his or her related items and propagates user preferences on KG for recommendations.
- **PER** [24] is a typical path-based approach that treats KG as a heterogeneous information network and extracts meta-path-based features to represent the connectivity between users and items.
- **KGCN** [15] is a state-of-the-art GNN-based model that iteratively integrates neighbor information to enrich item embedding.
- **KGNN-LS** [16] is a state-of-the-art GNN-based model that enriches item embeddings with GNN and label smoothness regularization.
- **KGAT** [17] is another state-of-the-art GNN-based model, which combines KG with the user-item graph as CKG, and uses attention mechanism to distinguish the importance of neighbors in CKG.
- **CKAN** [18] is a state-of-the-art GNN-based model that explicitly encodes the collaborative signals that are latent in user-item interactions and naturally combines them with knowledge associations in an end-to-end manner.
- **KGIN** [19] is also a state-of-the-art GNN-based approach that identifies user-item interactions at the granularity of user intents and exploits relational dependencies to preserve the semantics of remote connections.
- **CG-KGR** [20] is a novel knowledge-aware recommendation model that proposes a collaborative guidance mechanism

1 https://grouplens.org/datasets/hetrec-2011/
2 https://grouplens.org/datasets/movielens/1m/
3 http://www2.informatik.uni-freiburg.de/cziegler/BX/
4 https://searchengineland.com/library/bing/bing-satori

Table 2. The result of *AUC* and *F*1 in CTR prediction. The best results are in bold and the second-best results are underlined.

Model	Last.FM		MovieLens-1M		Book-Crossing	
	AUC	F1	*AUC*	F1	*AUC*	F1
BPRMF	0.7563 (-14.53%)	0.7010 (-11.78%)	0.8920 (-4.72%)	0.7921 (-8.58%)	0.6583 (-15.42%)	0.6117 (-10.17%)
CKE	0.7471 (-15.94%)	0.6740 (-16.26%)	0.9065 (-3.04%)	0.8024 (-7.19%)	0.6759 (-12.41%)	0.6235 (-8.08%)
RippleNet	0.7762 (-11.59%)	0.7025 (-11.54%)	0.9190 (-1.64%)	0.8422 (-2.13%)	0.7211 (-5.37%)	0.6472 (-4.13%)
PER	0.6414 (-35.05%)	0.6033 (-29.89%)	0.7124 (-31.12%)	0.6670 (-28.95%)	0.6048 (-25.63%)	0.5726 (-17.69%)
KGCN	0.8027 (-7.91%)	0.7086 (-10.58%)	0.9090 (-2.76%)	0.8366 (-2.81%)	0.6841 (-11.07%)	0.6313 (-6.75%)
KGNN-LS	0.8052 (-7.58%)	0.7224 (-8.47%)	0.9140 (-2.20%)	0.8410 (-2.27%)	0.6762 (-12.36%)	0.6314 (-6.73%)
KGAT	0.8293 (-4.45%)	0.7424 (-5.55%)	0.9140 (-2.20%)	0.8440 (-1.91%)	0.7314 (-3.88%)	0.6544 (-2.98%)
CKAN	0.8441 (-2.62%)	0.7692 (-1.87%)	0.9099 (-2.66%)	0.8425 (-2.09%)	0.7522 (-1.01%)	0.6711 (-0.42%)
KGIN	0.8486 (-2.07%)	0.7602 (-3.08%)	0.9190 (-1.64%)	0.8441 (-1.90%)	0.7273 (-4.47%)	0.6614 (-2.52%)
CG-KGR	0.8300 (-4.36%)	0.7344 (-6.70%)	0.9230 (-1.20%)	0.8521 (-0.94%)	0.7578 (-0.26%)	0.6714 (-1.89%)
KGIC	<u>0.8572</u> (-1.05%)	<u>0.7797</u> (-0.50%)	<u>0.9252</u> (-0.96%)	<u>0.8559</u> (-0.49%)	**0.7645** (+0.62%)	**0.6768** (+0.43%)
KDGNN	**0.8662**	**0.7836**	**0.9341**	**0.8601**	<u>0.7598</u>	<u>0.6739</u>

that enables ample and coherent learning of KGs and user-item interactions.

- **KGIC** [21] is the latest state-of-the-art GNN-based approach that focuses on exploring contrast learning in KG-based recommendation and proposes a new multi-level interactive contrast learning mechanism.

5.3. Experiments setup

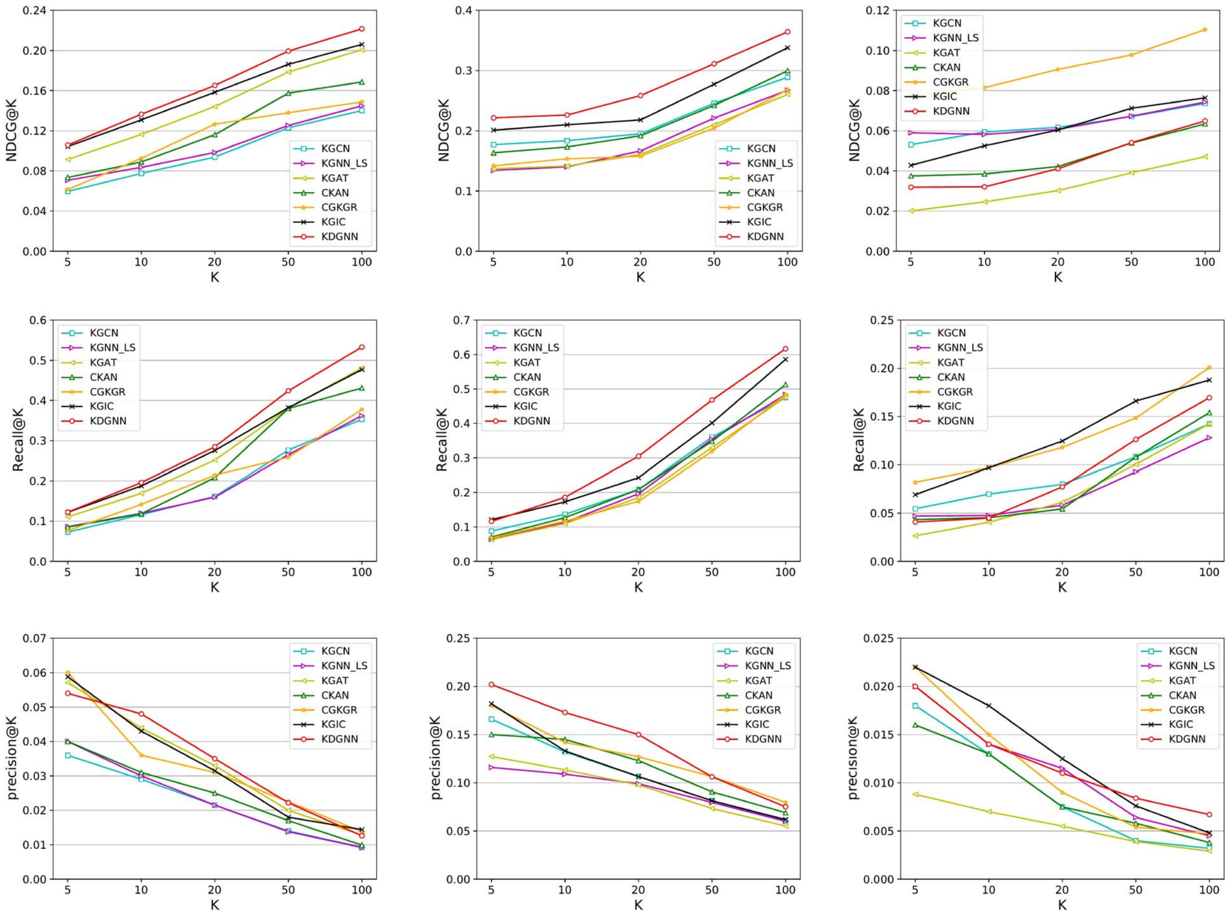
For each dataset, we divide it into a training set, a validation set and a test set in the ratio of 6:2:2. In our experiments, we consider two recommendation scenarios. (1) In click-through rate (CTR) prediction, we use the model learned from the training set to predict the probability of each interaction in the test set. (2) In top-*K* recommendation, we apply the training model to select the *K* items with the highest prediction probability for each user in the test set. For the evaluation of all methods, we select *AUC* and *F*1 to evaluate CTR prediction. For most GNN-based methods, we additionally selected *NDCG@K*, *Recall@K* and *Precision@K* to evaluate top-*K* recommendation.

This study implements the proposed model using the programming language python 3.7 and runs the codes on a Linux Server with Tesla V100-FHHL (16GB memory). Meanwhile, we implemented our model in PyTorch and used Adam to optimize the model parameters. For a fair comparison, we fix the batch size to 1024 and set the embedding size d to 64 for the three datasets. We set the size of the knowledge entity set according to the average number of user interaction items in different datasets, where Last.FM, MovieLens-1M and Book-Crossing are set to 30, 125 and 10. We perform a grid search to confirm the optimal settings based on the *AUC* on the validation set. We tune the learning rate among $\{10^{-2}, 10^{-3}, 2 \times 10^{-2}, 2 \times 10^{-3}\}$, the $L2$ normalization coefficient among $\{10^{-4}, 10^{-5}, 2 \times 10^{-4}, 2 \times 10^{-5}\}$, the number of neighbor samples among $\{4, 8, 16, 32, 64\}$ and the number of GNN layers among $\{1, 2, 3\}$. The best settings for hyper-parameters in all comparison methods are researched by either empirical study or following the original papers.

5.4. Results

The results of the CTR prediction and Top-*K* recommendation for the three datasets are given in Table 2 and Fig. 3. We have the following observations:

- KDGNN achieves strongly competitive performance on three datasets compared with the above baseline. In the three datasets of music, movie and book domains, the evaluation metric *AUC* and *F*1 achieves significant improvement on the above baseline, which reflects the effectiveness of KDGNN. This indicates that considering the semantic relations around entities can better capture the potential preferences of users, and it is effective to denoise the user's historical behavior data by utilizing the rich knowledge information in KG.
- In top-*K* recommendation, the performance of KDGNN is excellent especially in Last.FM and MovieLens-1M datasets. The performance of KDGNN demonstrates the effectiveness of dual-channel GNN and the integration of user and contextual knowledge information to fully capture user preferences. Notice that KDGNN was overtaken by CG-KGR and KGIC on the Book-Crossing dataset, despite having more prominent results on Last.FM and MovieLens-1M. Because in sparse and small-scale datasets, dual-channel GNN can capture less information, and the contextual knowledge information is not rich. KGIC utilizes the advantage of self-supervised learning to alleviate the problem of sparse supervised signals and obtains better performance. However, as datasets become denser and larger, more noise will be introduced into CG-KGR and KGIC, thus being inferior to KDGNN in Last.FM and MovliLens-1M datasets. Certainly, we can also get better results compared with other baselines. This is because KDGNN integrate personalized and knowledge-aware signals to capture user preferences fully, thus alleviating the sparsity problem.
- The better performance of CKE compared with BPRMF is due to the introduction of KG in CKE, which indicates the importance of KG as side information.
- PER performs even lower in all baselines than BPRMF because user-defined meta-paths are hardly optimal. The poor performance of CKE may be due to its inability to utilize KG by TransR-like regularization fully. Compared with BPRMF, CKE and PER, RippleNet has a more powerful performance because it uses a multi-hop neighbor structure. Overall, GNN-based baselines (such as CKAN, CG-KGR, KGIC, etc.) achieved the best results, illustrating the importance of exploiting structural information of entity contexts in KG.
- Compared with KGCN, KGNN-LS and KGAT, KGIN achieves better performance because KGIN exploits relations to model user's intents and uses relational dependencies to preserve the semantics of remote connections. Our advantage is more obvious. On the one hand, our model not only collects contextual relations about entities but also adaptively balances the



(a) Last.FM (b) MovieLens-1M (c) Book-Crossing

Figure 3. The result of *NDCG@K*, *Recall@K* and *Precision@K* in top-*K* recommendation.

contributions between entities and relations through a dual-channel balancing mechanism, thus attenuating the effect of irrelevant noise. On the other hand, our model integrates the rich contextual information in KG with the learnable embedding of the user to attenuate the noise of users' historical interactions.

5.4.1. *Impact of dual-channel GNN and personalized knowledge-aware attention network*

To verify the impact of the dual-channel GNN and the personalized knowledge-aware attention network on the model performance, we investigate three variants of the model: (1) changing the dual-channel GNN module to the traditional approach of aggregating neighbor entities (abbreviated as w/o DC); (2) removing personalized knowledge-aware attention network (abbreviated as w/o att); (3) removing dual-channel GNN and the personalized knowledge-aware attention network (abbreviated as w/o DC&att). We present the results in Table 3 and Fig. 4.

In general, the w/o DC&att variant causes the most performance degradation, which is consistent with our idea. This is because it lacks rich contextual knowledge and denoising mechanisms. In addition, w/o DC and w/o att also cause performance degradation. Overall, w/o att performs worse than w/o DC because the personalized knowledge-aware attention network contains not only knowledge entities generated by the dual-channel GNN, but also integrates signals from the user.

5.4.2. *Impact of dual-channel balancing mechanism and different user signals*

In order to observe the denoising performance based on personalized knowledge-aware attention network and dual-channel balancing mechanism, we configure three variants that model user preferences to verify the denoising ability: (1) removing only the dual-channel balancing mechanism in the dual-channel GNN (abbreviated as w/o blc). (2) remove the user-related knowledge entities and only consider the personalized signal (abbreviated as w/o k); (3) Only the part of user-related knowledge entities is considered (abbreviated as w/o p). We present the results in Table 4 and Fig. 5. First, removing the dual-channel balancing mechanism causes performance degradation. Because simply treating the contributions of entity information and relational information as fixed and not distinguishing the contribution of both will introduce irrelevant and redundant information. In addition, removing either the learnable embedding of the user or removing the user-related knowledge entities can lead to degraded model performance. It is shown that considering both the personalized and knowledge-aware signals can help capture the user's interest in a comprehensive way. The results confirm the previous analysis that integrating personalized and knowledge-aware signals is effective for denoising.

5.4.3. *Impact of neighbor sampling size*

To investigate the impact of the size of neighbor samples on performance, we sample a different number of neighbors.

Table 3. Impact of dual-channel GNN and personalized knowledge-aware attention network

Variants	Last.FM		MovieLens-1M		Book-Crossing	
	AUC	F1	*AUC*	F1	*AUC*	F1
w/o DC&att	0.8167	0.7403	0.9240	0.8517	0.7074	0.6393
w/o DC	0.8578	0.7745	0.9332	0.8592	0.7492	0.6539
w/o att	0.8201	0.7451	0.9250	0.8528	0.7090	0.6429
KDGNN	**0.8662**	**0.7836**	**0.9341**	**0.8601**	**0.7598**	**0.6739**

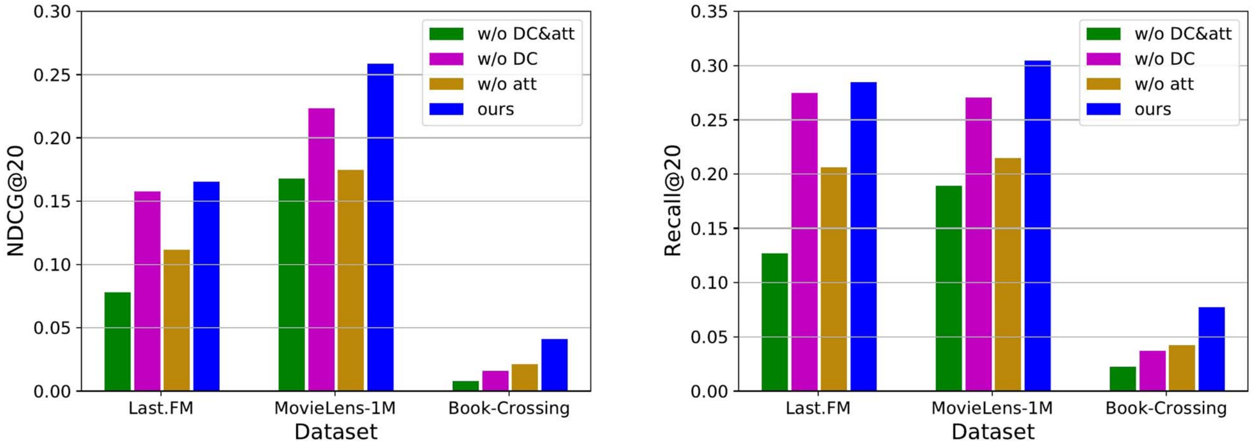


Figure 4. The result of *NDCG@20* and *Recall@20* w.r.t. impact of dual-channel GNN and personalized knowledge-aware attention.

Table 4. Impact of dual-channel balancing mechanism and different user signals

Variants	Last.FM		MovieLens-1M		Book-Crossing	
	AUC	F1	*AUC*	F1	*AUC*	F1
w/o blc	0.8641	0.7821	0.9336	0.8589	0.7499	0.6477
w/o k	0.8643	0.7800	0.9341	0.8587	0.7577	0.6685
w/o p	0.8655	0.7824	0.9340	0.8587	0.7570	0.6686
KDGNN	**0.8662**	**0.7836**	**0.9341**	**0.8601**	**0.7598**	**0.6739**

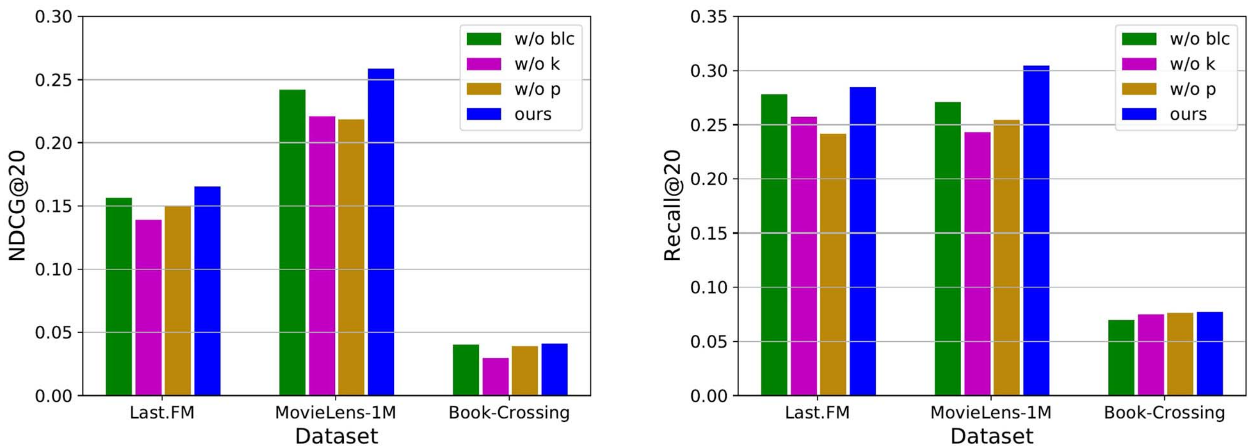


Figure 5. The result of *NDCG@20* and *Recall@20* w.r.t. impact of dual-channel balancing mechanism and different user signals.

As shown in Table 5, the model achieves the best performance when K=8. The reason is that when the number of sampled neighbors is too small, the information is relatively not rich, while the number of sampled neighbors is too large, too much neighbor information will weaken the characteristics of user and caused performance degradation. Therefore, without special instructions, the number of neighbors sampled is set to 8 in this experiment.

5.4.4. *Impact of knowledge entity set size*

Table 6 shows the impact of different knowledge entity set sizes (i.e. the number of knowledge entities considered per user) on the

Table 5. Impact of different neighbor sampling sizes

K	Last.FM		MovieLens-1M		Book-Crossing	
	AUC	F1	AUC	F1	AUC	F1
4	0.8659	**0.7858**	0.9338	0.8583	0.7565	0.6645
8	**0.8662**	0.7836	**0.9341**	**0.8601**	**0.7598**	**0.6739**
16	0.8642	0.7853	0.9339	0.8592	0.7569	0.6702
32	0.8622	0.7762	0.9338	0.8585	0.7569	0.6694
64	0.8618	0.7759	0.9333	0.8587	0.7572	0.6703

Table 6. Impact of different knowledge entity set sizes

S	Last.FM		MovieLens-1M		Book-Crossing	
	AUC	F1	AUC	F1	AUC	F1
10	0.8615	0.7787	0.9215	0.8458	0.7598	0.6739
30	**0.8662**	0.7836	0.9296	0.8539	0.7591	0.6710
50	0.8661	0.7820	0.9319	0.8570	0.7584	0.6725
70	0.8630	0.7801	0.9330	0.8588	**0.7614**	**0.6769**
90	0.8645	0.7827	0.9337	0.8593	0.7575	0.6725
110	0.8658	**0.7856**	0.9339	**0.8596**	0.7606	0.6720
130	0.8651	0.7852	**0.9341**	**0.8596**	0.7573	0.6704

model performance. The model performs worst when S is smaller than s (for the music, movie and book datasets, s is set to 30, 125 and 10, respectively.). As S gradually increases, the model performance also gradually increases. Moreover, relatively good results can be achieved when S reaches the average number s of user interaction items. When S exceeds s, performance is improved only in a few cases and the improvement is not significant, which also requires a larger computational overhead as a cost. Therefore, we select the average number of user interaction items (i.e. s) as the size of the knowledge entity set.

5.4.5. *Impact of aggregator*

We investigate the impact of different aggregator implementations on model performance, and Fig. 6 shows the results of three aggregations for each dataset. The results show that the sum aggregator and concat aggregator perform best. One possible reason is that when sum and concat perform aggregation, the characteristics of central entity and context information interact with each other, which can retain more information hidden in the embedding. In addition, the neighbor aggregator has the worst performance because it only considers the contextual structure information and ignores the importance of the central entity.

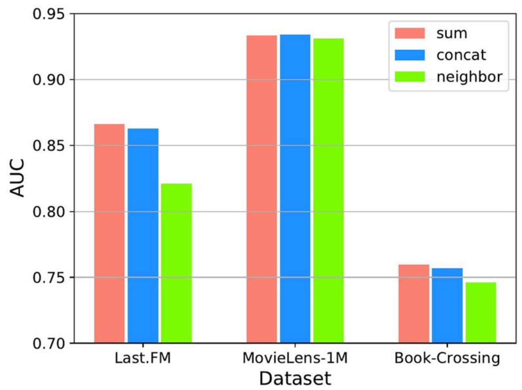


Figure 6. *AUC result for different aggregators.*

5.4.6. *Impact of embedding dimension*

To analyze the impact of embedding dimension on the model performance, Fig. 7 shows the intuitive results. Specifically, the performance of the model gradually increases as the dimension increases, reaching the best performance when the dimension is 64 or 128. Because larger dimension means more information can be encoded, while dimension beyond a certain range lead to overfitting.

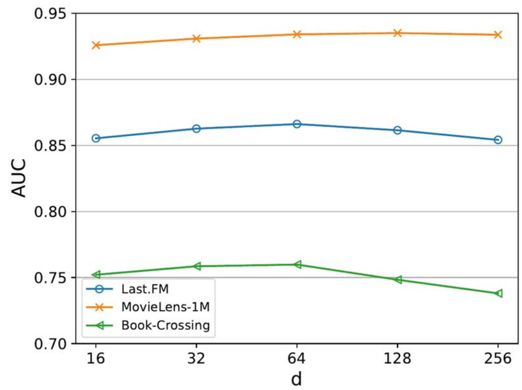


Figure 7. *AUC result for different dimensions of embedding.*

5.4.7. *Impact of depth of layer*

Finally, we explore the impact of the depth of the aggregation layer from 1 to 3 on KDGNN. Since the number of nodes in the l-hop neighborhood of a node increases exponentially with the increase of l, which introduces more neighbors and requires more memory support. Therefore, to ensure the computational efficiency and reduce memory dependency, we set the neighbor sampling size of the three datasets to 4 and the batch size of MovieLens-1M to 256 in this section. Table 7 shows the results for different depths of layer, which is similar to the conclusion in KGCN that good results are achieved when the number of layers is equal to 1, 2, and too high a number of layers is easy to introduce irrelevant information. However, unlike other methods such as KGAT and

Table 7. Impact of different depths of layer

L	Last.FM		MovieLens-1M		Book-Crossing	
	AUC	F1	*AUC*	F1	*AUC*	F1
1	**0.8659**	**0.7858**	**0.9337**	**0.8594**	**0.7565**	**0.6645**
2	0.8563	0.7750	0.9327	0.8586	0.7485	0.6510
3	0.8564	0.7748	0.9309	0.8567	0.7349	0.6328

KGIN, our model achieves the best performance at the first layer without stacking more layers. This is because the dual-channel GNN prevents the propagation of redundant noise and propagates both contextual entity and relational information, which makes our model more efficient.

6. CONCLUSION AND FUTURE WORK

In this paper, we focus on the negative impact of noise in the recommendation process. From a new perspective of denoising, a new KDGNN is proposed, which considers the importance of edge propagation and adaptively balances the contributions of entities and relations through a dual-channel balancing mechanism, thus blocking the propagation of redundant noise in KG. In addition, we integrate the learnable embedding of the user (i.e. personalized signal) with the embedding of knowledge entities (i.e. knowledge-aware signal) to comprehensively capture high-quality user preference representations and achieve denoising of user-item interaction data by personalized knowledge-aware attention. Finally, we compare with the existing SOTA method in three real datasets to demonstrate the superiority of KDGNN.

This work has generally studied personalized user representations and neglected the learning of personalized entity representations. The entity representation generated by the model is often dependent on neighbor information rather than directly on the user. In the future work, we will mine the potential semantically related entities of entities using the rich semantic relations in KG, and then update the representation of the entity by the potential semantically related entities. In addition, the internal associations between semantically related entities and different users will be explored to generate personalized entity representations.

ACKNOWLEDGMENTS

This work is supported in part by the National Natural Science Foundation of China (Nos.62262003 and U21A20474), the Guangxi Science and technology project (Guike AA22067070), the Guangxi Natural Science Foundation (No.2020GXNSFAA297075), the Center for Applied Mathematics of Guangxi (Guangxi Normal University), the Guangxi 'Bagui Scholar' Teams for Innovation and Research Project, the Guangxi Collaborative Innovation Center of Multi-source Information Integration and intelligent Processing, the Guangxi Talent Highland project of Big Data Intelligence and Application.

DATA AVAILABILITY STATEMENT

The data underlying this article will be shared on reasonable request to the corresponding author.

REFERENCES

1. Wang, H., Zhang, F., Xie, X. and Guo, M. (2018) DKN: deep knowledge-aware network for news recommendation. In Champin, P-A., Gandon, F., Lalmas M. and Ipeirotis P.G. (eds.), *Proceedings of the 2018 World Wide Web Conference*, Lyon, France, April 23–27, pp. 1835–1844. ACM, New York.
2. Diao, Q., Qiu, M., Wu, C., Smola, A. J., Jiang, J. and Wang, C. (2014) Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In Macskassy, S.A., Perlich, C., Leskovec J., Wang W. and Ghani, R. (eds.), *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, USA, August 24–27, pp. 193–202. ACM, New York.
3. Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H. and Gai, K. (2018) Deep interest network for click-through rate prediction. In Guo, Y. and Farooq, F. (eds.), *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, UK, August 19–23, pp. 1059–1068. ACM, New York.
4. Sarwar, B., Karypis, G., Konstan, J. and Riedl, J. (2001) Item- based collaborative filtering recommendation algorithms. In Shen, V.Y., Saito, N., Lyu, M.R. and Zurko, M.E. (eds.), *Proceedings of the Tenth International World Wide Web Conference*, Hong Kong, China, May 1–5, pp. 285–295. ACM, New York.
5. He, X., Liao, L., Zhang, H., Nie, L., Hu, X. and Chua, T. (2017) Neural collaborative filtering. In Barrett, V.Y., Cummings, C., Agichtein, E. and Gabrilovich, E. (eds.), *Proceedings of the 26th International Conference on World Wide Web*, Perth, Australia, April 3–7, pp. 173–182. ACM, New York.
6. Zhang, H., Shen, F., Liu, W., He, X., Luan, H. and Chua, T. (2016) Discrete collaborative filtering. In Perego, R., Sebastiani, F., Aslam, J.A., Ruthven, I. and Zobel, J. (eds.), *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, Pisa, Italy, July 17–21, pp. 325–334. ACM, New York.
7. Ai, Q., Azizi, V., Chen, X. and Zhang, Y. (2018) Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms*, **11**, 137.
8. Cao, Y., Wang, X., He, X., Hu, Z. and Chua, T. (2019) Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In Liu, L., White, R.W., Mantrach, A., Silvestri, F., McAuley J.J., Baeza-Yates, R. and Zia, L. (eds.), *The World Wide Web Conference*, San Francisco, CA, USA, May 13–17, pp. 151–161. ACM, New York.
9. Zhang, F., Yuan, N. J., Lian, D., Xie, X. and Ma, W. (2016) Collaborative knowledge base embedding for recommender systems. In Krishnapuram, B., Shah, M., Smola, A.J., Aggarwal, C.C., Shen, D. and Rastogi, R. (eds.), *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, August 13–17, pp. 353–362. ACM, New York.
10. Hu, B., Shi, C., Zhao, W.X. and Yu, P. S. (2018) Leveraging meta-path based context for top- N recommendation with A neural co-attention model. In Guo, Y. and Farooq, F. (eds.), *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, UK, August 19–23, pp. 1531–1540. ACM, New York.

11. Wang, X., Wang, D., Xu, C., He, X., Cao, Y. and Chua, T. (2019) Explainable reasoning over knowledge graphs for recommendation. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii, USA, January 27–February 1, pp. 5329–5336. AAAI Press, California.
12. Xian, Y., Fu, Z., Muthukrishnan, S., de Melo, G. and Zhang, Y. (2019) Reinforcement knowledge graph reasoning for explainable recommendation. In Piwowarski, B., Chevalier, M., Gaussier, E., Maarek, Y., Nie, J-Y. and Scholer, F. (eds.), *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Paris, France, July 21–25, pp. 285–294. ACM, New York.
13. Hamilton, W.L., Ying, Z. and Leskovec, J. (2017) Inductive representation learning on large graphs. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N. and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, Long Beach, CA, USA, December 4–9, 2017, pp. 1024–1034.
14. Kipf, T.N. and Welling, M. (2017) Semi-Supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations*, Toulon, France, April 24–26. OpenReview.net.
15. Wang, H., Zhao, M., Xie, X., Li, W. and Guo, M. (2019) Knowledge graph convolutional networks for recommender systems. In Liu, L., White, R.W., Mantrach, A., Silvestri, F., McAuley, J.J., Baeza-Yates, R. and Zia, L. (eds.), *The World Wide Web Conference*, San Francisco, CA, USA, May 13–17, pp. 3307–3313. ACM, New York.
16. Wang, H., Zhang, F., Zhang, M., Leskovec, J., Zhao, M., Li, W. and Wang, Z. (2019) Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In Teredesai, A., Kumar, V., Li, Y., Rosales, R., Terzi, E. and Karypis, G. (eds.), *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, AK, USA, August 4–8, pp. 968–977. ACM, New York.
17. Wang, X., He, X., Cao, Y., Liu, M. and Chua, T. (2019) KGAT: knowledge graph attention network for recommendation. In Teredesai, A., Kumar, V., Li, Y., Rosales, R., Terzi, E. and Karypis, G. (eds.), *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, AK, USA, August 4–8, pp. 950–958. ACM, New York.
18. Wang, Z., Lin, G., Tan, H., Chen, Q. and Liu, X. (2020) CKAN: collaborative knowledge-aware attentive network for recommender systems. In Huang, J.X., Chang, Y., Cheng, X., Kamps, J., Murdock, V., Wen, J. and Liu, Y. (eds.), *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, Virtual Event, China, July 25–30, pp. 219–228. ACM, New York.
19. Wang, X., Huang, T., Wang, D., Yuan, Y., Liu, Z., He, X. and Chua, T. (2021) Learning intents behind interactions with knowledge graph for recommendation. In Leskovec, J., Grobelnik, M., Najork, M., Tang, J. and Zia, L. (eds.), *The Web Conference 2021*, Ljubljana, Slovenia, April 19–23, pp. 878–887. ACM, New York.
20. Chen, Y., Yang, Y., Wang, Y., Bai, J., Song, X. and King, I. (2022) Attentive knowledge-aware graph convolutional networks with collaborative guidance for personalized recommendation. In *38th IEEE International Conference on Data Engineering*, Kuala Lumpur, Malaysia, May 9–12, pp. 299–311. IEEE, New York.
21. Zou, D., Wei, W., Wang, Z., Mao, X., Zhu, F., Fang, R. and Chen, D. (2022) Improving knowledge-aware recommendation with multi-level interactive contrastive learning. In Hasan, M.A. and Xiong, L. (eds.), *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, Atlanta, GA, USA, October 17–21, pp. 2817–2826. ACM, New York.
22. Lin, Y., Liu, Z., Sun, M., Liu, Y. and Zhu, X. (2015) Learning entity and relation embeddings for knowledge graph completion. In Bonet, B. and Koenig, S. (eds.), *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, Austin, Texas, USA, January 25–30, pp. 2181–2187. AAAI Press, California.
23. Wang, Z., Zhang, J., Feng, J. and Chen, Z. (2014) Knowledge graph embedding by translating on hyperplanes. In Brodley, C.E. and Stone, P. (eds.), *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, Québec City, Québec, Canada, July 27–31, pp. 1112–1119. AAAI Press, California.
24. Yu, X., Ren, X., Sun, Y., Gu, Q., Sturt, B., Khandelwal, U., Norick, B. and Han, J. (2014) Personalized entity recommendation: a heterogeneous information network approach. In Carterette, B., Diaz, F., Castillo, C. and Metzler, D. (eds.), *Seventh ACM International Conference on Web Search and Data Mining*, New York, NY, USA, February 24–28, pp. 283–292. ACM, New York.
25. Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X. and Guo, M. (2018) RippleNet: Propagating user preferences on the knowledge graph for recommender systems. In Cuzzocrea, A., Allan, J., Paton, N.W., Srivastava, D., Agrawal, R., Broder, A.Z., Zaki, M.J., Candan, K.S., Labrinidis, A., Schuster, A. and Wang, H. (eds.), *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, Torino, Italy, October 22–26, pp. 417–426. ACM, New York.
26. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C. and Yu, P.S. (2021) A comprehensive survey on graph neural networks. *IEEE Trans. Neural Networks Learn. Syst.*, **32**, 4–24.
27. Chung, J., Gülçehre, Ç., Cho, K. and Bengio, Y. (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. In *CoRR*, ***abs/1412.3555***.
28. Rendle, S., Freudenthaler, C., Gantner, Z. and Schmidt-Thieme, L. (2012) BPR: bayesian personalized ranking from implicit feedback. In *CoRR*, ***abs/1205.2618***.