Enhancing Sequential Recommendation via Decoupled Knowledge Graphs

Bingchao Wu[1,3], Chenglong Deng[1,3], Bei Guan[1,3], Yongji Wang[1,2,3](✉), and Yuxuan Kangyang[2,3]

[1] Collaborative Innovation Center, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
{bingchao2017,chenglong2018,guanbei}@iscas.ac.cn
ywang@itechs.iscas.ac.cn

[2] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
kyyx@ios.ac.cn

[3] University of Chinese Academy of Sciences, Beijing 100049, China

Abstract. Sequential recommendation can capture dynamic interest patterns of users based on user interaction sequences. Recently, there has been interest in integrating the knowledge graph (KG) into sequential recommendation. Existing works suffer from two main challenges: a) representing each entity in the KG as a single vector can confound heterogeneous information about the entity; b) triple-based facts are modeled independently, lacking the exploration of high-order connectivity between entities. To solve the above challenges, we decouple the KG into two subgraphs, namely CRoss-user Behavior-based graph and Intrinsic Attribute-based graph (Crbia), depending on the type of relation between entities. We further propose a CrbiaNet based on the two subgraphs. First, CrbiaNet obtains behavior-level and attribute-level semantic features from these two subgraphs independently by different graph neural networks, respectively. Then, CrbiaNet applies a sequential model incorporating these semantic features to capture dynamic preference of the users. Extensive experiments on three real-world datasets show that our proposed CrbiaNet outperforms previous state-of-the-art knowledge-enhanced sequential recommendation models by a large margin consistently.

Keywords: Sequential recommendation · Knowledge graph · Heterogeneous information · Graph neural network

1 Introduction

The recommendation system aims to suggest related items to users from a massive collection of items, thereby alleviating the problem of information overload. Sequential recommendation has been receiving increasing attention from researchers in the recommendation field. It is necessary to model dynamic user preference over time to provide accurate and high-quality recommendations. With the popularity and effectiveness of deep learning technologies in the fields

P. Groth et al. (Eds.): ESWC 2022, LNCS 13261, pp. 3–20, 2022.
https://doi.org/10.1007/978-3-031-06981-9_1

of computer vision and natural language processing, much of the literature on sequential recommendation has focused specifically on capturing sequential patterns from the historical interaction sequences sorted by time to predict future items for users via neural network models, such as GRU4Rec [6], Caser [22], SASRec [9], and BERT4Rec [21].

Although sequential recommendation has achieved great success in capturing dynamic user preference, it is limited by the fact that the vector of user preference is learned independently through each user's interaction sequence, and a large portion of the user interaction sequence is sparse [4]. Recently, many previous studies have focused on injecting the KG into sequential recommendation models through path-based methods (e.g., MASR [8] and KSRN [41]) and embedding-based methods (e.g., Chorus [27] and KERL [29]) to solve the aforementioned problems. The path-based approaches extract meta-paths that are relevant to user behavior sequences from the KG. However, these approaches rely heavily on expert knowledge to design reasonable meta-paths, and it is difficult to enumerate all potentially useful meta-paths [4]. The research in this paper is concerned with embedding-based approaches, which use the KG embedding methods to acquire the embedding of each entity in the KG. The existing embedding-based methods integrated into sequential recommendation models are divided into two categories, i.e., traditional distance-based models (e.g., TransE [1] and TransR [13]) and traditional semantic matching models (e.g., DistMult [35] and ComplEx [24]).

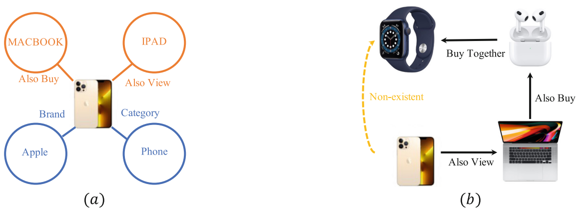


Fig. 1. (a) The heterogeneous information in the KG. (b) The high-order connectivity between items in the KG where the yellow dashed line indicates no directly connected edges between items. (Color figure online)

In the recommendation domain, there are two challenges in applying these two categories of embedding-based approaches to encode semantic features in the KG.

– Heterogeneous semantic information of items: the KG in the recommendation domain includes intrinsic attribute-level semantic information of items and behavior-level semantic information of items extracted from user logs [14]. A case is shown in Fig. 1-(a), the bottom two triples (*iPhone*, *brand*, *Apple*) and (*iPhone*, *category*, *Phone*) construct the attribute-level semantic information of *iPhone*, and the top two triples generate the behavior-level semantic information of *iPhone*. Existing embedding-based methods applied to sequential recommendation confound two types of heterogeneous information in a single vector.

– High-order connectivity between items: the embedding-based approaches mentioned above only model each fact consisting of a triplet individually, and ignore the high-order connectivity between items [28]. The high-order connectivityis a multi-hop relation path between items [30], which allows exploring deeper semantic information about items. A case is shown in Fig. 1-(b). Even though there are no directly connected edges between *iPhone* and *Apple Watch*, we can still capture the potential semantic relation through a multi-hop connection ($iPhone \longrightarrow MacBook \longrightarrow AirPods \longrightarrow Apple\ Watch$).

While the existing works (e.g., KSR [7], KERL [29] and GFE-SASRec [36]) utilize graph neural networks to model high-order connectivity, they only consider one type of KG or conflate heterogeneous information of items into a single vector. To overcome these challenges, we propose a sequential recommendation model CbiaNet[1] via merging decoupled knowledge graphs. First, we decouple the KG into two complementary subgraphs, named the cross-user behavior-based graph and the intrinsic attribute-based graph. Then, two knowledge sub-extractors encode the two subgraphs independently by graph neural networks to solve the problem of confounding heterogeneous semantics and to capture the higher-order connections between items. Next, a hierarchical knowledge aggregator combines the heterogeneous semantic information to generate high-level semantic features. Finally, a sequential model incorporating the high-level semantic features is developed to capture the dynamic preference of the users. We conduct experiments on three real-world datasets, and the experimental results show that our proposed CrbiaNet outperforms the existing state-of-the-art recommendation models. In addition, we extend the high-level semantic features to several sequential recommendation models, which also improves their performance.

2 Related Work

2.1 Sequential Recommendation

In order to model the dynamic interests of users, sequential recommendation methods utilize the user's historical interaction data. Markov chains are applied in traditional sequential recommendation methods by estimating the transition probability between items within the previous action sequence [19,20]. With the great success of deep learning methods in various fields, many efforts have been made to model users' historical interaction sequences by utilizing neural networks [6,9,12,21,22]. GRU4Rec [6] applies Gated Recurrent Units (GRU) to the session-based recommendation. NARM [12] further introduces attention-based GRU by assigning different weights to items of historical interaction sequences. Besides, Caser [22] and NextItNet [37] introduce Convolution Neural Network

[1] The codes are released at https://github.com/paulpig/sequentialRec.git..

(CNN) to learn sequential patterns as local features by using convolutional filters. Recently, various studies have validated that self-attention mechanisms effectively model dependencies between items [9,21]. SASRec [9] utilizes left-to-right Transformer models [25] to predict the next item. BERT4Rec [21] uses bidirectional Transformer models (BERT [3]) to encode user interest vectors by optimizing a Cloze task [23]. Sequential recommendations focus only on the user's own interaction sequence, ignoring the similar co-occurrence across users between items and relationships between items at the attribute level.

2.2 Knowledge-Enhanced Recommendation

KGs have been applied in various recommendation models to improve the performance of the recommendation where KGs use triples to describe realistic facts, such as the user-item KG [41], the item-item KG [34], and the item-attribute KG [41]. Several graph-based recommendation models jointly encode behavior-level user-item relations and knowledge-level item-item relations to introduce semantic knowledge from KG into the recommender system, such as KHGT [32], UGRec [39], and SMIN [14]. However, the above graph-based models cannot capture the dynamic user preference, so more research is focused on how to utilize knowledge graphs to enhance sequential recommendation models. Existing studies on injecting knowledge graphs into sequential models are mainly divided into two categories: path-based and embedding-based methods. For path-based methods, MASR [8] introduces meta-paths from the knowledge graph to capture global contextual information and applies the sequential model to capture the local contextual information. KARN [41] combines users' historical behavior sequences and the path between the user and the target item for recommendation. For embedding-based methods, KERL [29] uses TransR to obtain semantic features from KG that are fused into the sequential models. Chorus [27], RCF [34], and KDA [26] use DistMult to extract semantic features of items from KG by bilinear objectives and use the semantic features as input to the sequential model. Despite these recent advancements, the above knowledge graph embeddings cannot capture the higher-order connections between items in KG. DHIMN [33] applies a GCN-based message-passing layer to capture the high-level semantic knowledge in the KG, but ignores heterogeneous information of item relations in KG.

3 Problem Definition and Notation

3.1 Cross-User Behavior-Based Graph (CRBGraph)

In the recommendation domain, item relations extracted from user logs naturally exist in the datasets [15,26]. For example, the relation *also_buy* (*also_view*) between *iPhone* and *MacBook* means that users bought an *iPhone* and also bought (viewed) a *MacBook* afterwards. Here we represent these item relations with a cross-user behavior-based graph $\mathcal{G}_1$, defined as $\{(h,r,t)|h,t \in \mathcal{I}, r \in \mathcal{R}^b\}$

where $\mathcal{I}$ and $\mathcal{R}^b$ denote sets of item instances and item relations, respectively. The relations between item-item pairs are all positively correlated, so all types of item relations are reduced to a positive relation. This means that $r \in \{0, 1\}$ where $r = 1$ represents that there is a behavior-level link between the item-item pair.

3.2 Intrinsic Attribute-Based Graph (IAGraph)

In addition to behavior-level links between items, there are various types of item attributes, such as category and brand. Here we utilize the item-attribute pairs to generate an intrinsic attribute-based graph $\mathcal{G}_2$, defined as $\{(h, r', a)|h \in \mathcal{I}, a \in \mathcal{A}, r' \in \mathcal{R}^a\}$, where $\mathcal{I}$ and $\mathcal{A}$ denote sets of item instances and attribute values, and R^a is the set of attribute-level relations. For example, the triple (*iPhone*, *brand*, *Apple*) represents that the *brand* of the *iPhone* is *Apple*.

3.3 Task Description

Assume that there are M users and N items in the recommender system. Given the graphs $\mathcal{G}_1, \mathcal{G}_2$ and the interaction sequence $S^u = [i_1^u, i_2^u, \cdots, i_T^u]$ of user u where $i_1^u \in \mathcal{I}$ and T is the length of the interaction sequence, the knowledge-enhanced sequential recommendation task is denoted as follows:

$$i_u^* = argmax_{i_k \in \mathcal{I}} P(i_{T+1}^u = i_k | S^u, \mathcal{G}_1, \mathcal{G}_2)$$

where i_{T+1}^u is the predicted item at $T + 1$ time step, and P is the probability distribution over $\mathcal{I}$.

4 Method

The overview of CrbiaNet is shown in Fig. 2. The knowledge extractor is firstly employed to obtain heterogeneous item features from two distinct KGs, comprising a Behavior-level Knowledge Sub-extractor (BKS) and an Attribute-level Knowledge Sub-extractor (AKS). Then, the knowledge aggregator applies a hierarchical integration strategy to generate high-level semantic features by merging heterogeneous item features. Finally, a sequential interactions modeling layer merging high-level semantic features is employed to capture the dynamic user intention from the user's historical interaction sequence.

4.1 Knowledge Extractor

In this section, we design two types of graph neural networks to encode the behavior-level and attribute-level higher-order semantic features from the CRBGraph and the IAGraph, respectively. To model the CRBGraph, we design a behavior-level knowledge sub-extractor that aggregates semantic features of neighbors based on the flow direction of message passing in the graph neural network. For IAGraph, we aggregate the neighborhood information to the central node through the relationship-aware attention mechanism of the attribute-level knowledge sub-extractor.

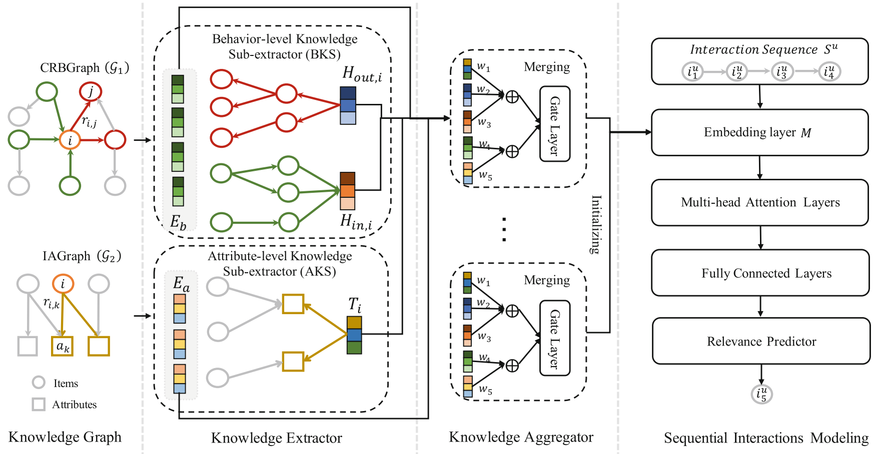


Fig. 2. The overall framework of our proposed model.

Behavior-Level Knowledge Sub-extractor (BKS). CRBGraph is a directed homogeneous graph in which each triple contains the time-series relation between the head and the tail entity. For example, the triple (*phone*, *also_buy*, *phone case*) means that users bought a *phone* and then also bought a *phone case*. Each node in the CRBGraph appears as a head entity in some related triples and as a tail entity in the rest of the related triples. This indicates that each node in the CRBGraph contains two types of time-series relations. To capture these time-series relations, we construct two-sided semantic features for each node, $\mathbf{H}_{in}$ and $\mathbf{H}_{out}$.

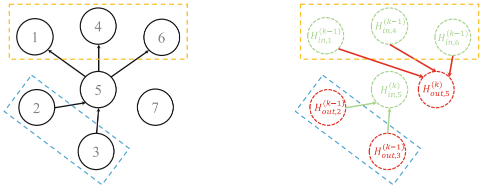
Specially, given the item i, one-hop neighbors of i are divided into out-degree neighbors $\mathcal{N}_i^{out}$ and in-degree neighbors $\mathcal{N}_i^{in}$. For example, $\mathcal{N}_5^{out}$ of the item v_5 is $\{v_1, v_4, v_6\}$ and $\mathcal{N}_5^{in}$ is $\{v_2, v_3\}$ in Fig. 3-(a). One-side semantic feature of item i at k^{th} layer, in-degree feature $\mathbf{H}_{in,i}^{(k)}$, aggregates the features $\mathbf{H}_{out,i}^{(k-1)}$ at $(k-1)^{th}$ layer in neighbors $\mathcal{N}_i^{in}$. The other-side feature $\mathbf{H}_{out,i}^{(k)}$ aggregates $\mathbf{H}_{in,i}^{(k-1)}$ in $\mathcal{N}_i^{out}$. The formula for the aggregation operation is:

$$\mathbf{H}_{in,i}^{(k)} = \sum\nolimits_{j\in\mathcal{N}_i^{in}} \frac{1}{\sqrt{|\mathcal{N}_j^{in}|}\sqrt{|\mathcal{N}_i^{out}|}}\mathbf{H}_{out,j}^{(k-1)}, \tag{1}$$

$$\mathbf{H}_{out,i}^{(k)} = \sum\nolimits_{j\in\mathcal{N}_i^{out}} \frac{1}{\sqrt{|\mathcal{N}_j^{out}|}\sqrt{|\mathcal{N}_i^{in}|}}\mathbf{H}_{in,j}^{(k-1)} \tag{2}$$

where $|\mathcal{N}_i^{out}|$ and $|\mathcal{N}_i^{in}|$ are the number of items in $\mathcal{N}_i^{out}$ and $\mathcal{N}_i^{in}$, respectively. A case is shown in Fig. 3-(b), the in-degree features $\{\mathbf{H}_{in,1}^{(k-1)}, \mathbf{H}_{in,4}^{(k-1)}, \mathbf{H}_{in,6}^{(k-1)}\}$ at $(k-1)^{th}$ layer of $\{v_1, v_4, v_6\}$ are propagated to the out-degree feature $\mathbf{H}_{out,5}^{(k)}$ at k^{th} layer of the item v_5 by Eq. 1. Note that $\mathbf{H}_{out}^{(0)} = \mathbf{H}_{in}^{(0)} = \mathbf{E}_b$, which means that $\mathbf{H}_{out}^{(0)}$ and $\mathbf{H}_{in}^{(0)}$ are from a shared embedding layer $\mathbf{E}_b$ to avoid overfitting.

Next, we stack more layers to capture higher-order item relations by Eq. 1 subject to k > 1 and obtain the final in-degree representation $\mathbf{H}_{in}$ by averaging



(a) Original Graph (b) Message Passing

Fig. 3. (a) The items in the orange and blue dashed boxes are the out-degree and in-degree neighbors; (b) The red and green dashed circles indicate the out-degree and in-degree features, respectively. (Color figure online)

the in-degree item features at each layer. The final out-degree representation $\mathbf{H}_{out}$ is derived using the similar operation. Finally, we optimize the behavior-level knowledge sub-extractor using the BPR loss [18]:

$$L_{CB} = -\sum_{(i,j,j')\in\mathcal{D}_R} \ln\sigma(\hat{y}_{ij} - \hat{y}_{ij'}); \quad \hat{y}_{ij} = \mathbf{H}_{out,i}\mathbf{H}_{in,j}^T \tag{3}$$

where $\mathcal{D}_R$ is $\{(i,j,j')|(i,r,j)\in\mathcal{G}_1 \wedge r=1, (i,r',j')\in\mathcal{G}_1 \wedge r'=0\}$.

Attribute-Level Knowledge Sub-extractor (AKS). Another knowledge sub-extractor is applied to encode potential attribute-level knowledge of items via a graph neural network, which can explore the user's preference at the attribute level. The high correlation of the attribute information and the preference behavior has been verified in [11,31].

First, the translation-based method TransR [13] is applied to model the first-order connectivity of entities in the IAGraph. However, it lacks the encoding of high-order connectivity between entities. We further introduce a graph attention network consisting of message propagation layers and message aggregation layers. For the k^{th} message propagation layer, we use the relation-aware attention mechanism to integrate neighbors of the central item i:

$$\mathbf{T}_{\mathcal{F}_i}^{(k)} = \sum_{(i,r,a)\in\mathcal{F}_i} \pi^{(k)}(i,r,a)\mathbf{T}_a^{(k)} \tag{4}$$

where $\mathcal{F}_i$ is the set of triples with the item i as the head entity in $\mathcal{G}_2$, and $\mathbf{T}_a^{(k)}$ is the feature of the entity a at k^{th} layer; $\pi^{(k)}(i,r,a)$ indicates the decay factor of the triple (i,r,a) in the message propagation [30]:

$$\begin{aligned} \pi^{(k)}(i,r,a) &= \frac{\exp(f^{(k)}(i,r,a))}{\sum_{(i,r',a')\in\mathcal{F}_i}\exp(f^{(k)}(i,r',a'))} \\ f^{(k)}(i,r,a) &= (\mathbf{W}_r\mathbf{T}_a^{(k)})^{\mathsf{T}}\tanh\Big((\mathbf{W}_r\mathbf{T}_i^{(k)} + \mathbf{T}_r^{(k)})\Big) \end{aligned} \tag{5}$$

where $\mathbf{W}_r$ is the relation-aware trainable parameter, and $\mathbf{T}_i^{(k)}$ and $\mathbf{T}_r^{(k)}$ are the features of the entity i and the relation r. For the k^{th} message aggregation layer,

$\mathbf{T}_{\mathcal{F}_i}^{(k)}$ and $\mathbf{T}_i^{(k)}$ are aggregated by two types of feature interactions and a nonlinear transformation and then passed to the $(k+1)^{th}$ layer:

$$\mathbf{T}_i^{(k+1)} = \sigma\Big(\mathbf{W}_1(\mathbf{T}_i^{(k)} + \mathbf{T}_{\mathcal{F}_i}^{(k)})\Big) + \sigma\Big(\mathbf{W}_2(\mathbf{T}_i^{(k)} \odot \mathbf{T}_{\mathcal{F}_i}^{(k)})\Big) \tag{6}$$

where σ is a LeakyReLU activation layer; $\mathbf{W}_1$ and $\mathbf{W}_2$ are the trainable parameters; Note that $\mathbf{T}^{(0)} = \mathbf{E}_a$ where $\mathbf{E}_a$ is an embedding table.

To model the higher-order connectivity in the IAGraph, we stack more layers and average the features of entities at each layer to generate $\mathbf{T}$. To optimize this sub-extractor, we introduce the BPR-based loss L_{AT}:

$$L_{AT} = -\sum_{(i,r,a,a')\in\mathcal{D}_A} \ln\sigma(\bar{y}_{i,r,a} - \bar{y}_{i,r,a'}); \quad \bar{y}_{i,r,a} = \mathbf{T}_i\mathbf{W}_r^a\mathbf{T}_a^T \tag{7}$$

where σ is a sigmoid activation layer, and $\mathcal{D}_A$ is $\{(i,r,a,a')|(i,r,a) \in \mathcal{G}_2, (i,r,a') \notin \mathcal{G}_2\}$; $\mathbf{W}_r^a$ is the trainable parameter.

4.2 Knowledge Aggregator

To merge the heterogeneous content information of each item into a fixed size embedding, we design a knowledge aggregator by integrating the semantic features of items extracted from the BKS and the AKS in a hierarchical manner. These item features consist of two parts: 1) high-order semantic features of items, including high-order out-degree features $\mathbf{H}_{out}$, high-order in-degree features $\mathbf{H}_{in}$ and high-order attribute-based features $\mathbf{T}$; 2) item embeddings, containing embeddings $\mathbf{E}_b$ of input to the AKS and embeddings $\mathbf{E}_a$ of input to the BKS. The fused high-order semantic features $\mathbf{M}_k^h$ are integrated by an attention mechanism that dynamically assigns attention weights to the three high-order semantic features mentioned above:

$$\begin{aligned}\mathbf{M}_k^h &= \sum_{V\in\{\mathbf{H}_{out},\mathbf{H}_{in},\mathbf{T}\}} w_v^k * \mathbf{V}_k,\\ w_v^k &= \frac{exp(\mathbf{W}_f^1 tanh(\mathbf{W}_f^2 {\mathbf{V}_k}^T))}{\sum_{Q\in\{\mathbf{H}_{out},\mathbf{H}_{in},\mathbf{T}\}} exp(\mathbf{W}_f^1 tanh(\mathbf{W}_f^2 {\mathbf{Q}_k}^T))}\end{aligned} \tag{8}$$

where $\mathbf{W}_f^1$ and $\mathbf{W}_f^2$ are the parameters of the attention mechanism. The fused item embeddings $\mathbf{M}_k^l$ are merged using the same attention mechanism. Next, a learnable gate is introduced to balance the contributions of the fused high-order item features $\mathbf{M}_k^h$ and the fused item embeddings $\mathbf{M}_k^l$:

$$\begin{aligned}\mathbf{G}_k &= \sigma(\mathbf{W}_g^1\mathbf{M}_k^h + \mathbf{W}_g^2\mathbf{M}_k^l)\\ \mathbf{M}_k &= \mathbf{G}_k \cdot \mathbf{M}_k^h + (1-\mathbf{G}_k)\cdot\mathbf{M}_k^l\end{aligned} \tag{9}$$

where $\mathbf{W}_g^1$ and $\mathbf{W}_g^2$ are the learnable parameters and σ is a sigmoid function; $\mathbf{M}$ is the high-level semantic knowledge.

4.3 Sequential Interactions Modeling (SIM)

In sequential interactions modeling, sequential models (e.g., GRU4Rec [6], SASRec [9], and BERT4Rec [21]) are widely used to capture the dynamic user preference based on historical interaction sequences. In this paper, we apply SASRec to encode the user interest representation, which consists of an embedding layer and self-attention blocks [25]. To inject the rich semantic knowledge extracted from the two KGs into SASRec, the embedding layer of SASRec is initialized by the high-level semantic knowledge $\mathbf{M}$ extracted from the knowledge aggregator. Specifically, given $\mathbf{M}$ and a user's interaction sequence $\mathcal{S}^u = [i_1, i_2, \cdots, i_T]$, the input embedding is:

$$\mathbf{E}_{\mathcal{S}^u} = [\mathbf{M}_0 + \mathbf{P}_0, \mathbf{M}_1 + \mathbf{P}_1, \cdots, \mathbf{M}_T + \mathbf{P}_T] \tag{10}$$

where $\mathbf{P}$ is a position embedding table. Then, we apply self-attentive blocks to establish dependencies between interactive items and capture the dynamic preference of the user through multi-head attention layers (MH) and fully connected feed-forward layers (FNN):

$$\mathbf{H}_{\mathcal{S}^u} = FFN(MH(\mathbf{E}_{\mathcal{S}^u})) \tag{11}$$

where $\mathbf{H}_{\mathcal{S}^u}$ is the hidden representation of the user interaction sequence $\mathcal{S}^u$. For MH and FFN, [25] has a detailed definition. To optimize the SIM, we adopt a binary cross entropy loss as the objective function:

$$L_{SQ} = -\sum_{S^u \in \mathcal{S}} \sum_{t \in [1,2,\cdots,T]} (\ln \sigma(\widetilde{y}_{tj}) + \sum_{k \notin S^u} \ln(1 - \sigma(\widetilde{y}_{tk})); \quad \widetilde{y}_{tj} = \mathbf{H}_{\mathcal{S}^u,t} {\mathbf{M}_j^h}^T \tag{12}$$

Note that the fused high-order item features $\mathbf{M}^h$ are used as semantic features of the target items to avoid overfitting.

4.4 Model Learning and Prediction

We use the pre-training and fine-tuning paradigm to better incorporate the semantic information extracted from KGs into the sequential recommendation model. Specifically, the BKS and the AKS are first pre-trained according to the optimization objectives in Eq. 3 and Eq. 7, and then fine-tuned together with the knowledge aggregator and the SIM using the optimization objective in Eq. 12. The final objective function of CrbiaNet is:

$$L_{CrbiaNet} = L_{SQ} + \alpha L_{CB} + \beta L_{AT} + \gamma(||\theta||_2^2) \tag{13}$$

where L_2 regularization on θ with the weight γ is designed to prevent overfitting, and α and β are the weights of the loss functions for different knowledge sub-extractors. In the inference phase, we only use the SIM as an online service to ensure the efficiency of the service.

5 Experiment

5.1 Experimental Settings

Datasets. We conduct experiments on the Amazon dataset [5], which includes the interactions between users and items and metadata of items with natural item relations [27] (e.g., *also_view*, *also_buy*) and attributes of items (e.g., price, brand and category). CRBGraph ($\mathcal{G}_1$) and IAGraph ($\mathcal{G}_2$) are constructed from the natural item relations and the attributes of items, respectively. We use three representative sub-datasets in the Amazon dataset: *Beauty*(Beauty), *Sports and Outdoors*(Sports), and *Toys and Games*(Toys). The detailed statistics of Amazon datasets are consistent with [40]. To construct user interaction sequences, we group user interaction records, sort them according to the timestamps ascendingly. We filter out users and items with less than five interaction records following previous studies [9,21].

Parameter Settings and Evaluation Metrics. CrbiaNet is trained by the Adam optimizer [10] with a learning rate of 0.001, where the batch size of the knowledge sub-extractors (BKS and AKS) and SIM are set as 2048 and 256, respectively. Gradients are clipped when the gradient norm is greater than five. The number of layers and the embedding dimensions are set to 2 and 64 for BKS, AKS, and SIM. Following previous sequential recommendation models [9,21], the maximum length of the user interaction sequence is set as 50. The weights α, β and γ are set to 1.0. Besides, the leave-one-out strategy is used for training and evaluation, and top-k HIT Ratio(HR@k) and top-k Normalised Discounted Cumulative Gain (NDCG@k) are considered to be ranking metrics. Following previous studies [6,9], we evaluate the performance of the models by combining the ground-truth item and 99 randomly sampled non-interactive negative items.

Baseline Methods. To validate the effectiveness of our proposed CrbiaNet model, we select nine previous representative models as baseline methods.

- **BPR** [18] is a classical Bayesian personalized ranking algorithm with implicit feedback based on stochastic gradient descent.
- **FM** [17] considers the combined features based on linear regression.
- **GRU4Rec** [6] applies GRU [2] to model user interaction sequences for session-based recommendations with a ranking loss function.
- **SASRec** [9] is a sequential recommendation model based on deep unidirectional transformers that capture dynamic user interests.
- **BERT4Rec** [21] uses BERT [3] to encode user interaction sequences by deep bidirectional transformers.
- **FDSA** [38] captures the dynamic user preference by simultaneously modeling both item-level and feature-level(attribute-level) sequences.
- **S^3-Rec** [40] adopts the paradigm of pre-training and fine-tuning, where attributes are employed in the pre-training phase.

Table 1. The performance of our proposed model and previous existing recommendation models on three datasets, where the best results and the second best results are marked in bold and underlined, respectively.

Datasets	Metric	BPR	FM	GRU4Rec	SASRec	BERT4Rec	FDSA	S^3-Rec	Chorus	KDA\T	CrbiaNet
Beauty	HR@5	0.3602	0.1461	0.3487	0.3754	0.4034	0.4010	0.4502	0.4575	<u>0.4846</u>	**0.5123***
	NDCG@5	0.2601	0.0934	0.2580	0.2832	0.3080	0.2974	0.3407	0.3402	<u>0.3654</u>	**0.3875***
	HR@10	0.4659	0.2311	0.4460	0.4795	0.5052	0.5096	0.5506	0.5694	<u>0.6008</u>	**0.6204***
	NDCG@10	0.2944	0.1207	0.2893	0.3168	0.3408	0.3324	0.3732	0.3766	<u>0.4031</u>	**0.4225***
Sports	HR@5	0.3629	0.1603	0.3208	0.3538	0.3922	0.3855	0.4267	<u>0.4540</u>	0.4504	**0.4860***
	NDCG@5	0.2624	0.1048	0.2257	0.2493	0.2852	0.2756	0.3104	<u>0.3346</u>	0.3273	**0.3554***
	HR@10	0.4851	0.2491	0.4389	0.4805	0.5203	0.5136	0.5614	0.5823	<u>0.5831</u>	**0.6200***
	NDCG@10	0.3018	0.1334	0.2638	0.2900	0.3264	0.3170	0.3538	<u>0.3761</u>	0.3701	**0.3988***
Toys	HR@5	0.3140	0.0978	0.3284	0.3684	0.3926	0.3994	0.4420	0.4290	<u>0.4961</u>	**0.5149***
	NDCG@5	0.2286	0.0614	0.2422	0.2712	0.2979	0.2903	0.3270	0.3306	<u>0.3806</u>	**0.3974***
	HR@10	0.4138	0.1715	0.4293	0.4751	0.4959	0.5129	0.5530	0.5291	<u>0.6015</u>	**0.6217***
	NDCG@10	0.2607	0.0850	0.2746	0.3057	0.3313	0.3271	0.3629	0.3631	<u>0.4147</u>	**0.4320***

- **Chorus** [27] is a sequential recommendation model with natural item relations and corresponding temporal dynamics.
- **KDA** [26] injects natural item relations between items, attributes of items, and temporal evolution information as additional knowledge into the sequence recommendation. For the sake of fairness, the temporal evolution information is removed in this paper and named **KDA\T**.

5.2 Performance Comparison

Table 1 shows the results of all baselines and our proposed CrbiaNet model on all datasets. First, sequential recommendation methods (e.g., GRU4Rec, SASRec, and BERT4Rec) outperform collaborative filtering methods (e.g., BPR and FM) because the dynamic user preference can be captured by encoding the history of the user's interaction with the recommender system. The performance of sequential recommendation models can be further improved by merging the attributes of items (e.g., FDSA and S^3-Rec), which indicates the attribute-based side information is helpful for recommender systems. Chorus obtains better performance due to incorporating behavior-based (natural) item relations. In addition, KDA\T achieves the previous state-of-the-art performance on the three datasets by integrating both attributed-based and behavior-based KGs. One possible reason for this is that the complex relations between the target items and the items in the user's historical interaction sequence are explicitly captured by the KGs.

Then, CrbiaNet consistently outperforms the pure and attribute-enhanced sequential recommendation models in the three datasets, thanks to the rich heterogeneous semantic features injected into the sequential interaction model. Compared with pure sequential recommendation methods (e.g., GRU4Rec, SASRec, and BERT4Rec), CrbiaNet achieves better recommendation performance, demonstrating that the underlying semantic knowledge embedded in the KGs is helpful for capturing the dynamic user preference. CrbiaNet is superior to

Table 2. The effectiveness of each component of our proposed CrbiaNet on the three datasets.

Model	Metric	Beauty	Sports	Toys
CrbiaNet	HR@10	0.6204	0.6200	0.6217
	NDCG@10	0.4225	0.3988	0.4320
CrbiaNet-BKS	HR@10	0.6033	0.5964	0.5924
	NDCG@10	0.4076	0.3801	0.4086
CrbiaNet-AKS	HR@10	0.5134	0.5189	0.5203
	NDCG@10	0.3396	0.3178	0.3422
CrbiaNet-ADD	HR@10	0.6066	0.6145	0.6192
	NDCG@10	0.4156	0.3930	0.4274
CrbiaNet-RANDOM	HR@10	0.4795	0.4805	0.4751
	NDCG@10	0.3168	0.2900	0.3057

FDSA and S^3-Rec incorporating only attribute-based knowledge, suggesting that behavior-based(natural) item relations are helpful for the recommendation. This shows that co-occurrence patterns from item-item pairs of historical interaction sequences of similar users mitigate the disadvantage of sparse user interaction behaviors.

Finally, our proposed CrbiaNet achieves the state-of-the-art performance in three datasets compared with previous knowledge-enhanced sequential recommendation models (Chorus and KDA\T). The following facts can illustrate these results: 1) independent modeling of CRBGraph and IAGraph allows encoding the heterogeneous semantic information of items more efficiently (see the Subsect. 5.4 for more discussion); 2) high-order connections between items in CRBGraph and IAGraph can be captured by message passing mechanism in the knowledge extractor; 3) the knowledge aggregator effectively aggregates the heterogeneous semantic information of items, which helps to dynamically assign attention weights to different semantic features based on user interest.

5.3 Ablation Study

To investigate the impact of components in CrbiaNet, we compare CrbiaNet with its four variants:

- CrbiaNet-BKS: This model incorporates only the semantic features extracted from CRBGraph by the behavior-level knowledge sub-extractor (BKS) into the sequential interactions modeling (SIM) to demonstrate the impact of cross-user item relations on recommendation performance.
- CrbiaNet-AKS: This model uses only the semantic features extracted from the IAGraph via attribute-level knowledge sub-extractor (AKS) to inject into the SIM.
- CrbiaNet-ADD: This model replaces the complex knowledge aggregator (KA) with the simple addition operation to fuse heterogeneous semantic features to validate the effectiveness of the integration strategy.

– CrbiaNet-RANDOM: This model replaces the high-level semantic knowledge **M** extracted from KGs with an embedding layer with 0 mean and 0.01 standard deviation.

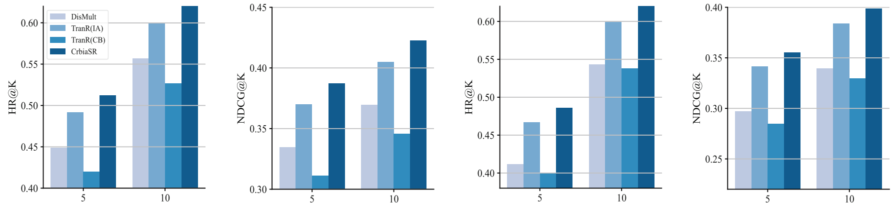
Table 2 shows HR@10 and NDCG@10 for these models in all three datasets. We summarize the following findings. First, the BKS of modeling the cross-user item relations in the CRBGraph is the most critical component of CrbiaNet. CrbiaNet-BKS offers significant performance gains over the other three variants in all three datasets, indicating that co-occurrence patterns from item-item pairs can guide the extraction of more accurate user interests. Second, CrbiaNet-AKS outperforms CrbiaNet-RANDOM by utilizing attribute-based semantic features extracted from the IAGraph, demonstrating the need to incorporate the attributes of items. In addition, CrbiaNet-AKS outperforms FDSA [38] on both NDCG@10 and HR@10, which validates that our proposed AKS can effectively extract attribute-aware high-level semantic knowledge. Last, the difference in performance between CrbiaNet and CrbiaNet-ADD suggests that the hierarchical knowledge integration strategy can better integrate heterogeneous semantic features from the KGs by dynamically assigning attention weights to features.

5.4 Effectiveness of Knowledge Extractor

To validate the effectiveness of our proposed knowledge extractor, we compare CrbiaNet with three variants in terms of graph construction and graph encoding:

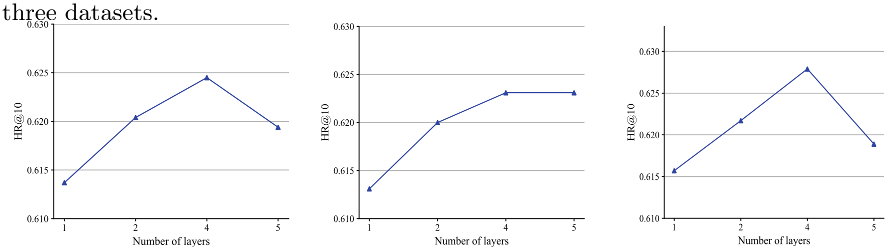
– DisMult: To explore the effectiveness of extracting heterogeneous semantic features from CRBGraph and IAGraph independently, this model first constructs a unified knowledge graph by merging CRBGraph and IAGraph, and then uses DisMult [35] instead of the knowledge extractor in this paper (for more details see [34]).
– TransR(IA): This model replaces AKS with TransR [13] to validate the necessity of potential attribute-aware high-order semantic features for recommendations.
– TransR(CB): This model uses TransR [13] instead of BKS to encode CRBGraph to validate the effectiveness of behavior-level high-order item relations.

The results of these variants and CrbiaNet are shown in Fig. 4. CrbiaNet achieves better performance than DisMult. Two reasons may cause this phenomenon: 1) CrbiaNet encodes different types of KGs independently to avoid confusion of heterogeneous semantic features; 2) the bilinear diagonal model (DisMult) cannot map attribute-level and behavior-level semantic features to the identical semantic space. Compared to TransR(IA) and TransR(CB), CrbiaNet achieves the best performance on all three datasets. This shows that high-level semantic features are practical for sequential recommendations. In addition, the most significant performance gap is observed between CrbiaNet and TransR(CB), indicating that behavior-level high-order item relations play a crucial role in encoding the dynamic user preference.



(a) HR@K on Beauty (b) N@K on Beauty (c) HR@K on Sports (d) N@K on Sports

Fig. 4. Performance of the knowledge extractor in CrbiaNet and other extractors on three datasets.



(a) Amazon Beauty (b) Amazon Sports (c) Amazon Toys

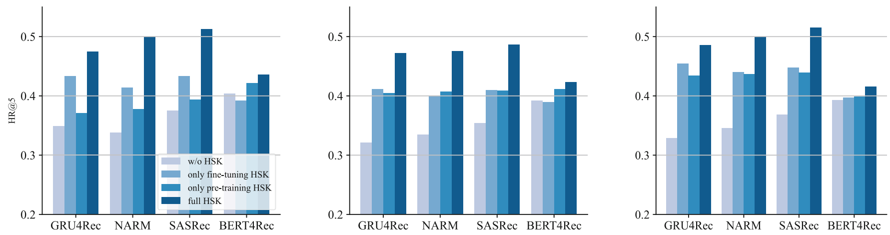
Fig. 5. Impact of knowledge extractor depth.

5.5 Impact of Knowledge Extractor Depth

This subsection considers the impact of the number of layers in the knowledge extractor to validate the necessity of high-order connections between items in the KGs. The results are summarized in Fig. 5. First, we can observe that recommendation performance is improved by stacking a certain number of layers in the knowledge extractor, indicating that stacking more layers can explore higher-order item relations in the KG and mine the potential preference of users. However, the recommendation performance of CribaNet on Amazon Beauty and Amazon Toys datasets decreases when more layers are stacked in the knowledge extractor. This shows that stacking too many layers in the knowledge extractor may lead to the problem of over-smoothing. This problem is prevalent in the graph neural networks [16], and we leave the exploration of solving this problem as future work. In addition, the over-smoothing problem does not affect CribiaNet on Amazon Sports dataset when the number of layers is stacked to five. The reason might be that there are more triples in the KG on the Sports dataset than the other two datasets, and longer-distance item relations are required to encode the heterogeneous semantic knowledge of items.

5.6 Compatibility of High-level Semantic Knowledge

To explore the validity and compatibility of High-level Semantic Knowledge **M** (HSK) mentioned in the Subsect. 4.2, we conduct an experiment employing the HSK and its three variants on four sequential models (GRU4Rec [6], NARM [12],



(a) Amazon Beauty (b) Amazon Sports (c) Amazon Toys

Fig. 6. The performance of CrbiaNet and its variants under different sequential models on three datasets.

SASRec [9], and BERT [21]): 1) **w/o HSK**: This method uses randomly initialized embeddings to replace the HSK; 2) **only fine-tuning HSK**: This method only uses the optimization objective in Eq. 12 to obtain the HSK through fine-tuning CrbiaNet; 3) **only pre-training HSK**: This approach only uses the optimization objectives in Eq. 3 and Eq. 7 to obtain the HSK through pre-training the knowledge extractor and keeps the HSK constant in the fine-tuning stage. 4) **full HSK**: This method first pre-trains the knowledge extractor to obtain the HSK, and then fine-tunes CrbiaNet to adapt the HSK to the sequential recommendation task.

The experimental results are shown in Fig. 6. First, we can observe that all sequential models achieve better performance than 'w/o HSK' when merging HSK, indicating that our proposed HSK is compatible and effective with the sequential recommendation models. In addition, the sequential models' performance decreases on both 'only fine-tuning HSK' and 'only pre-training HSK' compared to 'full HSK', which suggests that our proposed HSK can fully exploit the deeper underlying semantic features in the heterogeneous KGs.

6 Conclusion

In this paper, we propose a CrbiaNet for sequential recommendation by merging heterogeneous semantic features of entities extracted from decoupled KGs. In our approach, we decouple the original KG in the recommendation domain into two subgraphs, named the cross-user behavior-based graph and the intrinsic attribute-based graph. Then, we propose two knowledge sub-extractors to acquire higher-order features of entities with different semantics independently by graph neural networks. Finally, the high-order semantic features are combined and fed into the sequential recommendation model to enhance the representation of the user preference. We construct experiments on Amazon datasets, and the experimental results show that CrbiaNet outperforms the previous state-of-the-art recommendation models.

Acknowledgement. This research project was supported by the Foundation of Science and Technology Project of Hebei Education Department (Grants No. ZD2021063).

References

1. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, vol. 26 (2013)
2. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
3. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
4. Guo, W., et al.: Dual graph enhanced embedding neural network for ctrprediction. arXiv preprint arXiv:2106.00314 (2021)
5. He, R., McAuley, J.: Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering. In: Proceedings of the 25th International Conference on World Wide Web, pp. 507–517 (2016)
6. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939 (2015)
7. Huang, J., Zhao, W.X., Dou, H., Wen, J.-R., Chang, E.Y.: Improving sequential recommendation with knowledge-enhanced memory networks. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 505–514 (2018)
8. Huang, X., Qian, S., Fang, Q., Sang, J., Xu, C.: Meta-path augmented sequential recommendation with contextual co-attention network. ACM Trans. Multimed. Comput. Commun. Appl. (TOMM) **16**(2), 1–24 (2020)
9. Kang, W.-C., McAuley, J.: Self-attentive sequential recommendation. In: 2018 IEEE International Conference on Data Mining (ICDM), pp. 197–206. IEEE (2018)
10. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Kosinski, M., Stillwell, D., Graepel, T.: Private traits and attributes are predictable from digital records of human behavior. Proc. Natl. Acad. Sci. **110**(15), 5802–5805 (2013)
12. Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., Ma, J.: Neural attentive session-based recommendation. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 1419–1428 (2017)
13. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
14. Long, X., et al.: Social recommendation with self-supervised metagraph informax network. arXiv preprint arXiv:2110.03958 (2021)
15. Ma, W., et al.: Jointly learning explainable rules for recommendation with knowledge graph. In: The World Wide Web Conference, pp. 1210–1221 (2019)
16. Oono, K., Suzuki, T.: Graph neural networks exponentially lose expressive power for node classification. arXiv preprint arXiv:1905.10947 (2019)
17. Rendle, S.: Factorization machines. In: IEEE International Conference on Data Mining, pp. 995–1000. IEEE (2010)
18. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012)
19. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized Markov chains for next-basket recommendation. In: Proceedings of the 19th International Conference on World Wide Web, pp. 811–820 (2010)

20. Shani, G., Heckerman, D., Brafman, R.I., Boutilier, C.: An MDP-based recommender system. J. Mach. Learn. Res. **6**(9) (2005)
21. Sun, F., et al.: BERT4Rec: sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 1441–1450 (2019)
22. Tang, J., Wang, K.: Personalized top-n sequential recommendation via convolutional sequence embedding. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 565–573 (2018)
23. Taylor, W.L.: "Cloze procedure": a new tool for measuring readability. J. Q. **30**(4), 415–433 (1953)
24. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: International Conference on Machine Learning, pp. 2071–2080. PMLR (2016)
25. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
26. Wang, C., Ma, W., Zhang, M., Chen, C., Liu, Y., Ma, S.: Toward dynamic user intention: temporal evolutionary effects of item relations in sequential recommendation. ACM Trans. Inf. Syst. (TOIS) **39**(2), 1–33 (2020)
27. Wang, C., Zhang, M., Ma, W., Liu, Y., Ma, S.: Make it a chorus: knowledge- and time-aware item modeling for sequential recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 109–118 (2020)
28. Wang, M., Qiu, L., Wang, X.: A survey on knowledge graph embeddings for link prediction. Symmetry **13**(3), 485 (2021)
29. Wang, P., Fan, Y., Xia, L., Zhao, W.X., Niu, S., Huang, J.: Kerl: a knowledge-guided reinforcement learning model for sequential recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 209–218 (2020)
30. Wang, X., He, X., Cao, Y., Liu, M., Chua, T.-S.: KGAT: knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 950–958 (2019)
31. Wu, L., Yang, Y., Zhang, K., Hong, R., Fu, Y., Wang, M.: Joint item recommendation and attribute inference: an adaptive graph convolutional network approach. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 679–688 (2020)
32. Xia, L., et al.: Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 5, pp. 4486–4493 (2021)
33. Xie, T., Xu, Y., Chen, L., Liu, Y., Zheng, Z.: Sequential recommendation on dynamic heterogeneous information network. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE), pp. 2105–2110. IEEE (2021)
34. Xin, X., He, X., Zhang, Y., Zhang, Y., Jose, J.: Relational collaborative filtering: modeling multiple item relations for recommendation. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 125–134 (2019)
35. Yang, B., Yih, W.-T., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 (2014)
36. Yang, Z., Dong, S., Hu, J.: GFE: general knowledge enhanced framework for explainable sequential recommendation. Knowl.-Based Syst. **230**, 107375 (2021)

37. Yuan, F., Karatzoglou, A., Arapakis, I., Jose, J.M., He, X.: A simple convolutional generative network for next item recommendation. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pp. 582–590 (2019)
38. Zhang, T., et al.: Feature-level deeper self-attention network for sequential recommendation. In: IJCAI, pp. 4320–4326 (2019)
39. Zhao, X., Cheng, Z., Zhu, L., Zheng, J., Li, X.: UGRec: modeling directed and undirected relations for recommendation. arXiv preprint arXiv:2105.04183 (2021)
40. Zhou, K., et al.: S3-Rec: self-supervised learning for sequential recommendation with mutual information maximization. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 1893–1902 (2020)
41. Zhu, Q., Zhou, X., Wu, J., Tan, J., Guo, L.: A knowledge-aware attentional reasoning network for recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 04, pp. 6999–7006 (2020)