

Research Article

An Implicit Preference-Aware Sequential Recommendation Method Based on Knowledge Graph

Haiyan Wang ^{1,2}, Kaiming Yao ^{1,2}, Jian Luo ¹ and Yi Lin ^{1,2}

¹School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

²Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing 210023, China

Correspondence should be addressed to Haiyan Wang; wanghy@njupt.edu.cn

Received 27 May 2021; Accepted 2 August 2021; Published 16 August 2021

Academic Editor: Honghao Gao

Copyright © 2021 Haiyan Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Sequential recommendation system has received widespread attention due to its good performance in solving data overload. However, most of the sequential recommendation methods assume that user's preferences only depend on specific items in the current sequence and do not consider user's implicit interests. In addition, most of the previous works mainly focus on exploiting relationships between items in the sequence and seldom consider quantifying the degree of preferences for items implied by user's different behaviors. In order to address these above two problems, we propose an implicit preference-aware sequential recommendation method based on knowledge graph (IPAKG). Firstly, this method introduces knowledge graph to exploit user's implicit preference representations. Secondly, we integrate recurrent neural network and attention mechanism to capture user's evolving interests and relationships between different items in the sequence. Thirdly, we introduce the concept of behavior intensity and design a behavior activation unit to exploit the degree of preferences for items implied by a user's different behaviors. Through the activation unit, the user's preferences on different items are further quantified. Finally, we conduct experiments on an Amazon electronics dataset and Tmall dataset to evaluate the performance of our method. Experimental results demonstrate that our proposed method has better performance than those baseline methods.

1. Introduction

With the rapid development of online platforms in recent years, the problem of information overload has become more and more serious. Recommendation systems, aimed at mining user's preferences to recommend personalized goods or services for the user in the massive data, have achieved widespread attention and great success due to its practicality and effectiveness. So far, recommendation systems have been widely used in various fields, including e-commerce, social media, short videos, and searching engines. Traditional recommendation systems mainly focus on static preferences of users. In order to better model the dynamic and evolving interests of users, sequential recommendation is proposed. Sequential recommendation, as a type of recommendation system, recommends personalized items for users based on their historical interaction behaviors [1, 2].

Traditional recommendation methods (such as collaborative filtering and matrix factorization) mainly focus on

the static interaction of user-items, so it is difficult to meet the requirements of sequential recommendation. In order to model the sequential task, a series of models are proposed. The FPMC [3] model is based on Markov chain and matrix factorization. The main problem of this model is that this method is still modeling the static preferences of users. With the rapid development of neural networks, neural network-based models have been widely used in sequential recommendation tasks, such as recurrent neural network-based models [4–6] and convolutional neural network-based models [7]. The recurrent neural network-based model expresses the user's interest as a hidden state vector by encoding the user's historical interaction records. This method can capture the user's dynamic and evolving interest preferences, but due to the characteristics of the recurrent neural network, this method is difficult to capture the long-term dependence of user preferences. The models based on the convolutional neural network express sequential features and capture the complex relevance of items by directly modeling paired item

relationships. The attention mechanism has received widespread attention due to its high efficiency. The sequential recommendation method based on self-attention [8–12] has also been widely proposed. This method uses the attention mechanism to capture the relationship between items in the sequence, solving the problem of long dependence of the recurrent neural network models. The above-mentioned methods have achieved good performance in the sequential recommendation task. However, we emphasize that there are two main problems with the above methods. Firstly, these methods often assume that the user’s interest only depended on the certain item in the current sequence and do not consider the user’s implicit interest. For example, in movie recommendation scenarios, although the user clicks on the movie *Wonder Woman*, this does not mean that the user likes the movie itself, and it is possible that the user is only interested in the actors in the movie. Therefore, it is of great importance to exploit the implicit preferences of users. Secondly, in the sequential recommendation, most of the past work mainly focus on the relationship between different items in current recommendation sequence and seldom consider the information implied by the user’s behavior, let alone quantify this behavioral information. We argue that the user’s behavior plays an important role in the sequential recommendation. Different behaviors imply the degree of the user’s preferences for specific items. For example, if the user bookmarks or purchases an item, it can be considered that the user has a strong interest on the current item; if the user just clicks on a certain item, it can be considered that the user is just interested in this item; and if the user skips an item, it can be considered that the user has no preference for the item at all. The interaction process is shown in Figure 1. Therefore, it is very significant to quantify the user’s different behavior information. We emphasize the integration of users’ implicit preferences and behavioral intensity to improve the performance of the model.

In view of the above analysis, we propose an implicit preference-aware sequential recommendation method based on knowledge graph. Firstly, we introduce knowledge graph-embedding technology. The knowledge graph contains comprehensive auxiliary information about the item, which helps to further expand the item representation space and exploit the user’s deeper implicit preferences, so that we can obtain high-quality entity representation. Secondly, we introduce the concept of behavior intensity and design a behavioral activation unit to exploit the degree of preferences implied by the user’s different behaviors. By introducing the behavioral activation unit, the user’s microbehavior information is further mined and utilized. Finally, we integrate recurrent neural network and attention mechanism to capture the user’s dynamically evolving interests and the relationship between different items in the sequence. In order to evaluate the performance of our proposed IPAKG model, we conduct experiments on two real datasets, and the experimental results proved the efficiency of our proposed model.

The main contributions of this paper are as follows:

- (i) We proposed the IPAKG model, which introduces a knowledge graph to exploit a user’s implicit prefer-

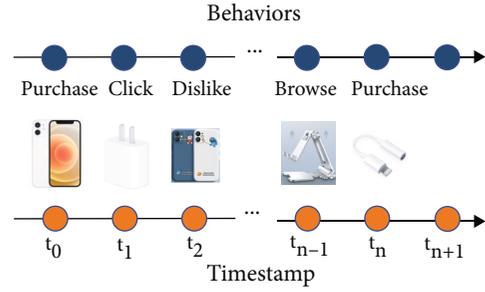


FIGURE 1: An example of user’s interaction process.

ence representations. In addition, we integrate a recurrent neural network and attention mechanism to capture the user’s evolving interest and the relationship between different items in the sequence

- (ii) In order to exploit the degree of preferences implied by the user’s different microbehaviors, we design a behavior activation unit, through which the degree of preferences implied by the user’s different microbehaviors is further quantified
- (iii) We conduct extensive experiments on two real datasets, and the experimental results demonstrate that the IPAKG model has a great improvement over the baseline models

The remainder of this paper is organized as follows. Previous research on sequential recommendation methods and related technologies is discussed in Section 2. In Section 3, the proposed IPAKG approach is explained. Then, in Section 4, the experiments are described and the results are analyzed. Finally, the conclusion and future work are discussed in Section 5.

2. Related Work

In this section, we briefly review three aspects related to our work, namely, general recommendation, sequential recommendation, and knowledge graph-based recommendation methods.

2.1. General Recommendation. Early recommendation systems usually use collaborative filtering models [13] and their variants for recommendation. These models predict a user’s ratings for items by mining the user’s long-term stable preference information, but there are cold-start problems in collaborative filtering-based models. In order to solve the problem of data sparsity, a model based on matrix factorization [14] is proposed. In addition, Yang et al. [15] proposed a new sparsity alleviation algorithm for the recommendation model to solve the sparsity problem by addressing the zero values. With the development of deep learning, the models based on deep neural networks are widely used in recommendation systems. Wu et al. [16] proposed a collaborative noise reduction autoencoder model called CDAE, which combines the traditional matrix factorization model with a deep neural network and predicts the user’s rating information through an autoencoder. Lian et al. [17] further

improved the recommendation performance by combining the traditional decomposition machine model with the deep learning model. Togashi et al. [18] proposed a knowledge graph- (KG-) aware recommender based on graph neural networks, which augments labelled samples through pseudo-labelling to tackle the cold-start problems for new users/items. The above-mentioned models show that deep learning-based methods have greater flexibility in learning user and item representations.

2.2. Sequential Recommendation. Given the user's historical interaction records, the purpose of sequential recommendation methods is to predict the items that the user may take interest in in the future based on the historical interaction data. The early sequential recommendation task is mainly based on the Markov chain. Rendle et al. [3] proposed a model called FPMC, which includes both a general Markov chain and a conventional matrix factorization model and recommends the next item to the user through the item information of the user's interaction in the past period of time. He and McAuley [19] combined Markov chains with similarity-based methods and proposed the fossil model, which can significantly improve recommendation performance, especially on sparse datasets. With the development of deep learning, sequential recommendation methods based on deep learning are widely used. Donkers et al. [4] used a recurrent neural network for sequential recommendation and achieved a better performance boost. Quadrana et al. [20] proposed a hierarchical RNN model for cross-session transmission, which can relay the potential hidden state of the terminal RNN between user sessions. Tang and Wang [21] applied the convolutional neural network to sequential recommendation and regarded the embedding representation of the past L items in the user sequence as a photo, which was used as the input of CNN. Thanks to the excellent performance of the attention mechanism in natural language processing, many people apply the attention mechanism to the recommendation system. Kang and McAuley [22] proposed the SASRec model, which mainly uses a two-layer transformer decoder to model the user's sequential behavior information. Sun et al. [23] applied the BERT model to recommendations to encode user preferences. Zhou et al. [24] proposed the RIB model, which models inherently the sequence of microbehaviors and their effects, but their method did not fully explore the user's implicit preference information. Gu et al. [25] adopted LSTM to model the user's microbehaviors. However, they ignored the different transition pattern between items and behaviors. In addition, their method does not take into account the implicit preference information of users. Meng et al. [26] incorporated user microbehaviors and item knowledge into multitask learning for session-based recommendation, which achieved good performance.

2.3. Knowledge Graph-Based Recommendation. With the development of knowledge graph technology, many people consider combining the knowledge graph with the recommendation system and improve the performance of the recommendation system by mining multiple association

relationships between items. Wang et al. [27] proposed a model called the Knowledge Path Recursive Network (KPRN), which uses knowledge graphs for recommendation. KPRN can generate path representations by combining the semantics of entities and relationships, using the order dependency in the path. The relationship can be effectively inferred on the path to infer the basic principles of the interaction between the user and the item. Lei et al. [28] proposed a conversational path reasoning (CPR) model, which can model conversational recommendation as a graphical interactive path reasoning problem. Through user feedback, this method traverses the attribute vertices in an explicit way. At the same time, by using the graph structure, CPR can delete many irrelevant candidate attributes, so that it can hit the attributes that users prefer. Wang et al. [29] capture the rich relationship between user and item by extracting paths from the knowledge graph. The above work combines the knowledge graph and the recommendation system to improve the performance of recommendation and has good interpretability. However, due to the problem of model selection, it is difficult to determine whether these methods effectively capture high-order associations.

3. Proposed Method

In this section, we introduce our proposed IPAKG model in detail. Our purpose is to provide users with a number of recommended items based on their historical interaction records. We first state the notations and task definition, then describe our methods in detail, and finally state the optimization methods.

3.1. Notations and Task Definition

3.1.1. Notations. We first define the notations that need to be used in this paper. In a sequential recommendation system, given user set $U = \{u_1, u_2, \dots, u_{|U|}\}$, item set $I = \{i_1, i_2, \dots, i_{|I|}\}$, where $|U|$ and $|I|$ represent the number of elements in the user collection and the item collection, respectively. The historical item interaction sequence of user u can be $\{i_1^{(u)}, \dots, i_t^{(u)}, \dots, i_T^{(u)}\}$, where $i_t^{(u)}$ indicates that user u interacted with item i at time t , and T indicates the interaction length of the current user. The time here is relative time. In addition, since we consider the influence of user behavior on preferences, we define the behavior set $B = \{b_1, \dots, b_i, \dots, b_{|B|}\}$, where b_i represents the behavior imposed by the user on the item, and $|B|$ represents the total number of behaviors. Behavior set B has only a few finite elements, such as purchases, views, clicks, carts, and comments. Different behaviors represent the user's preferences for the current item. Based on the above definition, the behavior interaction sequence of user u can be represented by $\{b_1^{(u)}, \dots, b_t^{(u)}, \dots, b_T^{(u)}\}$. In addition to the user's item interaction sequence and behavior interaction sequence, the knowledge graph G is also required in our sequential recommendation task. The knowledge graph G is defined on the head entity set V and the relation entity set R and is composed of a large number of triples, namely, $G = \{(h, r, t) \mid h, t \in V, r \in R\}$, where h

TABLE 1: Notations.

Symbol	Description
U	User set
I	Item set
B	Behavior set
$ U $	Number of user set
$ I $	Number of item set
$ B $	Number of behavior set
i_t^u	Item i that user u interacted with at time t
b_t^u	Behavior b that user u imposed at time t
G	Knowledge graph
V	Entity set in G
R	Relationship set in G
h	Head entity in knowledge graph
t	Tail entity in knowledge graph
r	Relationship between head and tail entity
W^r	Transformation matrix of relationship in G
e^h	Embedding of head entity in G
e^t	Embedding of tail entity in G
e^r	Embedding of relation entity in G
W^Q, W^K, W^V	Linear mapping matrices
W_z, W_r, W_h	Parameter matrices
b_z, b_r, b_h	Deviation parameters
α^k	Attention weight
e_X'	Final preference representation
L_G	Loss function related to knowledge graph
L_P	Loss function related to model
L	Final loss function
λ	Regularization coefficient

and t represent the head entity and the tail entity, respectively, and r represents the relationship between the head entity and the tail entity; a knowledge graph triplet states a factual relationship. For example, the triple (Tenet, isDirectedBy, ChristopherNolan) indicates that the movie Tenet is directed by Christopher Nolan. Since our item set I needs to be associated with the knowledge graph, we can think of I as a subset of the knowledge graph G , that is, $I \subset G$. Through the knowledge graph, we can expand the representation space of the item and mine the implicit preferences of users. Relevant symbol definitions are summarized in Table 1.

3.1.2. Task Definition. Based on the above definition, given the user's historical item interaction sequence $\{i_1^u, \dots, i_t^u, \dots, i_T^u\}$, the history behavior interaction sequence corresponding to the item $\{b_1^u, \dots, b_t^u, \dots, b_T^u\}$, the user's portrait, and the knowledge graph G , the task of sequential recommendation is to predict the item i_{T+1}^u that the user u may interact at time-step $T + 1$.

3.2. Implicit Preference Awareness Sequential Recommendation Model Based on Knowledge Graph. Our model architecture is shown in Figure 2. Our model integrates the knowledge graph information and the user's microbehavior characteristics, which can efficiently expand the user's preference representation space. In addition, through the behavioral activation unit, it can capture the degree of the user's preferences for the certain item. Our model mainly consists of five parts, namely, the knowledge graph-embedding representation layer, preference dependence-aware layer, preference-evolving layer, behavior preference activation layer, and preference attention layer. In the following part, we will elaborate on each component separately.

3.2.1. Knowledge Graph-Embedding Representation Layer. In order to fully exploit the user's implicit preferences, we introduce the knowledge graph-embedding technology. The knowledge graph-embedding technology can map the entities and relationships in the graph to a vector representation, while retaining the information of the graph structure, which is an efficient representation method. In order to map the entities and relationships in the knowledge graph into vector representations, we use the widely used knowledge graph-embedding technology TransR method [30], for the knowledge graph triples $(h, r, t) \in G$, $e_h, e_t \in R^k$ represent the embedding representation of the head entity h and the tail entity t in the knowledge graph, and $e_r \in R^m$ represents the embedding representation of the relation r in the knowledge graph. TransR learns the embedding representation of entities and relationships by optimizing $e_h' + e_r \approx e_t'$, where e_h' and e_t' are the mapping representations of e_h and e_t in the relational space r , respectively. Therefore, given the knowledge graph triples (h, r, t) , the score function can be defined as follows:

$$s(h, r, t) = \|W^r e_h + e_r - W^r e_t\|_2^2. \quad (1)$$

Among them, $W^r \in R^{(k \times d)}$ is the transformation matrix of relation r . The purpose is to map the entity from the d -dimensional space to the k -dimensional space of the relation r . The smaller the above-mentioned score function, the more likely is the triplet (h, r, t) to appear in the knowledge graph G . Through the knowledge graph embedding, we obtain the embedding representation vector e_i of the items in the user interaction sequence.

3.2.2. Preference Dependence-Aware Layer. In order to learn the dependencies between different items in the sequence, we introduce the multihead self-attention mechanism [31]. The self-attention mechanism captures the relationship between items in the sequence to improve the presentation ability of items and, at the same time, eliminates the sequence noise data. The scaled dot-product attention is defined as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V, \quad (2)$$

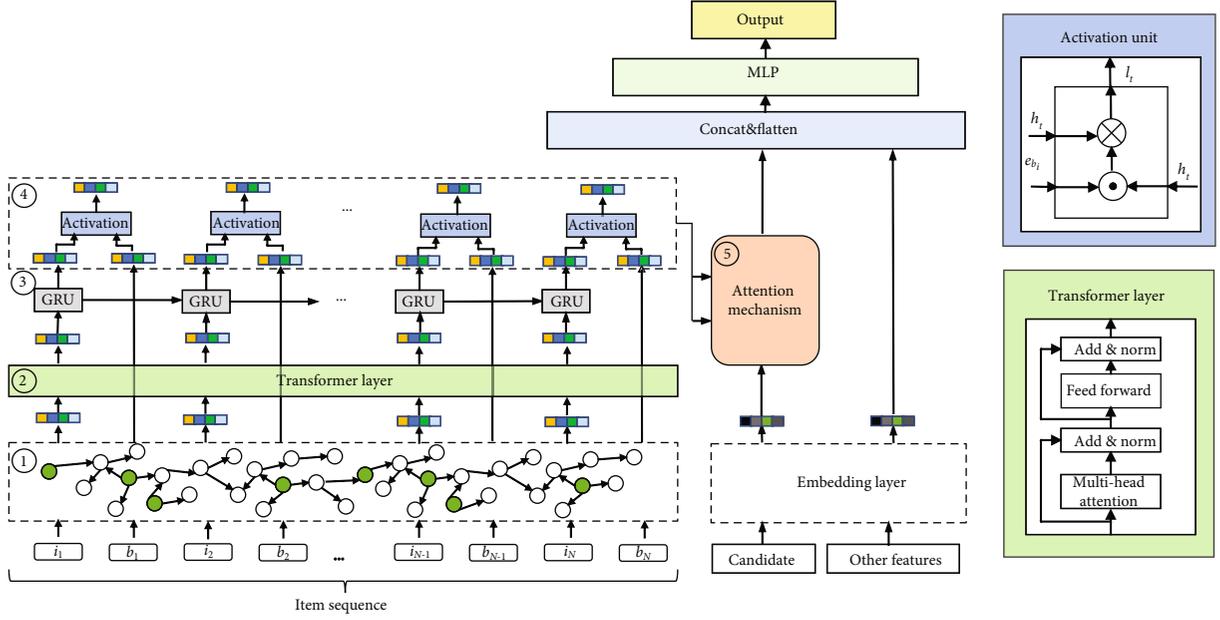


FIGURE 2: Framework of IPAKG. There are mainly five components in IPAKG, including knowledge graph-embedding representation layer, preference dependence-aware layer, preference-evolving layer, behavior preference activation layer, and preference attention layer. The prediction score of each item is obtained through the multilayer perceptron.

where $Q = EW^Q$ means queries; $K = EW^K$ means keys; $V = EW^V$ means values; W^Q , W^K , and W^V are linear mapping matrices; and $E = [e_{i_1}; \dots; e_{i_r}]$ is the stacked matrix of the embedded vector of the knowledge graph. The self-attention of multiple heads can be expressed in the following form:

$$\text{MultiHead}(E) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W^H, \quad (3)$$

$$\text{head}_i = \text{Attention}\left(EW_i^Q, EW_i^Q, EW_i^Q\right), \quad (4)$$

where h is the number of heads. Then, the above results are passed through pointwise Feed-Forward Networks (FFN) to further increase the nonlinearity of the model. In order to avoid over-fitting and to learn feature representation hierarchically, we use dropout and LeakyReLU, so the output of self-attention and FNN is as follows:

$$M = \text{LayerNorm}(E + \text{Dropout}(\text{MultiHead}(E))), \quad (5)$$

$$F = \text{LayerNorm}(M + \text{Dropout}(\text{LeakyReLU}(MW^1 + b^1)W^2 + b^2))). \quad (6)$$

Among them, W^1 , b^1 , W^2 , and b^2 are learning parameters. Through the preference dependence-aware layer, the item representation in the sequence contains the relationship information between different items.

3.2.3. Preference-Evolving Layer. The user's sequential interaction behavior includes the dynamic changes of user preferences and the preference dependencies. Therefore, in the preference-evolving layer, we model the dynamic evolution of preferences and the dependencies between preferences

through recurrent neural networks. We use GRU to model the dynamic evolution of the user's preferences and the dependence between preferences. Compared with LSTM, GRU has fewer parameters, so the convergence speed is faster and the iteration speed can be accelerated. The formula for GRU is as follows:

$$z_t = \sigma(W_z \cdot [h_{t-1}, e_{i_t}] + b_z), \quad (7)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, e_{i_t}] + b_r), \quad (8)$$

$$\tilde{h}_t = \tanh(W_h[r_t * h_{t-1}, e_{i_t}] + b_h), \quad (9)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t, \quad (10)$$

where σ is the sigmoid activation function; $*$ is the element-wise multiplication of the vector; W_z , W_r , and $W_h \in R^{d_H \times d_I}$ are the parameter matrix; b_z , b_r , and $b_h \in R^{d_H}$ are the deviation parameters; e_{i_t} is the input of the GRU; and h_t is the t^{th} hidden state vector. Through the recurrent neural network, the user's dynamic evolving preference and dependence between different preferences can be obtained.

3.2.4. Behavior Preference Activation Layer. In sequential recommendation, the behaviors that users impose on different items are different, and such different behaviors imply the degree of preferences for the certain item. For example, the user's purchase behavior implies that the user has a strong preferences or demand for the current item, while browsing behavior merely indicates that the user has certain preferences for the current item. Therefore, it is of great significance to quantify user's different behavior information. In order to model the user's preference intensity implied by the user's different behavior, we introduce the behavioral

preference activation unit. The formula of the behavioral activation unit is as follows:

$$I_i = (h_t \odot e_{b_i}) \cdot h_t. \quad (11)$$

Among them, \odot represents the inner product of the vector, and e_{b_i} represents the embedded representation of the user behavior. Through the behavior activation unit, we quantify the user's different behaviors into the user's degree of preferences for the current item.

3.2.5. Preference Attention Layer. The similarity between the target item and the items in the sequence significantly affects whether the user will interact with the target item. In order to obtain the influence weight of different items in the user's historical interaction sequence on the target item, we introduce the attention mechanism [32]. The preferences after adjustment by the attention mechanism is expressed as follows:

$$\alpha_k = \frac{\exp(I_k W_{e_x})}{\sum_k^T \exp(I_k W_{e_x})}, \quad (12)$$

$$e'_x = \sum_k^T \alpha_k I_k. \quad (13)$$

Among them, W is the parameter matrix, e_x is the embedded representation of the target item, and e'_x is the final preference representation of the user. Finally, the feature-embedding vector, preference representation vector and target item-embedding vector undergo Concat and flatten operations, and then the final prediction result is obtained through the multilayer perceptron.

3.3. Optimization. Our objective function includes the knowledge graph-embedding loss part and the prediction loss part. In the knowledge graph-embedding part, we use the TransR method, which considers the relative order relationship between valid triples and invalid triples and uses the pairwise ranking loss function to make the difference between the scores of these two triples as large as possible. The loss function can be expressed as

$$L_G = \sum_{h,r,t,t'} -\ln \sigma(s(h,r,t') - s(h,r,t)). \quad (14)$$

Among them, $T = \{(h,r,t,t') \mid (h,r,t) \in G, (h,r,t') \notin G\}$, triples (h,r,t') are obtained by randomly replacing the tail entity of the triple (h,r,t) in the knowledge graph G , and $\sigma(\dots)$ is the sigmoid activation function. In order to predict whether a certain user will interact with the item, we treat it as a two-classification problem. In order to train the model, we use the cross-entropy loss function,

$$L_p = -\frac{1}{N} \sum_{(x,y) \in D} (y \log p(x) + (1-y) \log (1-p(x))). \quad (15)$$

TABLE 2: Statistics of Amazon (electronics) and Tmall datasets.

	Amazon (electronics)	Tmall	
User-item interactions	#users	11032	33252
	#items	8865	40865
	#ratings	223654	863252
	#sparsity	99.79%	99.93%
Knowledge graph	#entity	43261	145806
	#relation	22	26
	#triple	924352	2863546

Among them, D is the training set; the number of samples is N ; y is the real label, which represents whether the user has interacted with the item; and $p(x)$ is the output value of the sigmoid function, which represents the current item predicted by the model to be interacted by the user probability. The final objective function can be expressed as

$$L = L_G + L_p + \lambda \|\theta\|_2^2. \quad (16)$$

Among them, λ is the regularization coefficient, and θ is the model parameter set. In order to prevent over-fitting, we adopt the L_2 regularization. In training, we use minibatch Adam [33] to optimize the embedding loss and prediction loss. Adam is a widely used optimizer that can adaptively control the learning rate.

4. Experiment

In this section, we first introduce the experimental dataset, the baseline model, and the evaluation metrics of the experiment, and then, we analyze the experimental results and compare them with the baseline models. In addition, we also conduct an ablation test.

4.1. Datasets and Metrics. We use two commonly used public datasets to verify the effect of my proposed IPAKG model. The statistical information of the dataset is shown in Table 2. The first dataset is the Amazon dataset which comes from the Amazon e-commerce platform; this dataset contains millions of raw data and review information about products. It is a widely used benchmark dataset [34, 35]. We conducted experiments on the Amazon electronics dataset, a subset of the Amazon dataset. In order to adapt to our scenario, we processed the data to filter out those users with less than 30 item interactions and those items with less than 20 occurrences. The second dataset is Tmall that comes from the Tmall e-commerce platform and is a competition dataset provided by IJCAI-15. This dataset contains user behavior information, such as clicks, browses, and purchases. We filter out those users with less than 40 item interactions and those items with less than 20 occurrences. For each user, we sorted the items according to the timestamp.

In order to construct the knowledge base, we refer to the past work [36] and adopt Freebase data dumps. In order to collect relevant factual information from Freebase, we keep those triples associated with the entities mapped by the items

in the dataset. Note that not all knowledge in the knowledge base is useful, so we filter out the relationships that appear with low frequency. In order to evaluate our method, we refer to past work [5, 37] and adopt a series of widely used evaluation metrics, including the hit ratio (HR), normalized discounted cumulative gain (NDCG), and mean reciprocal rank (MRR).

- (i) *HR*: hit ratio gives the percentage of users that can receive at least one correct recommendation, which has been widely used in previous work.

$$\text{HR@K} = \frac{1}{M} \sum_u I\left(\left|R_u \cap T_u\right|\right), \quad (17)$$

where $I(x)$ is an indicator function whose value is 1 when $x > 0$ and 0 otherwise. R_u is the generated recommendation list for user u ; T_u is the item set that user u interacted with in the test set. M is the number of samples.

- (ii) *NDCG*: normalized discounted cumulative gain evaluates ranking performance by taking the positions of correct items into consideration.

$$\text{NDCG@K} = \frac{\text{DCG}}{\text{IDCG}}, \quad (18)$$

$$\text{DCG@K} = \sum_{i=1}^K \frac{\text{rel}_i}{\log_2(i+1)}, \quad (19)$$

where rel_i represents the relevance of the i^{th} recommendation items. If the i^{th} item is selected by the user, then $\text{rel}_i = 1$; otherwise, $\text{rel}_i = 0$. IDCG is a normalized DCG.

- (iii) *MRR*: mean reciprocal rank is a popular ranking metric to measure recommendation quality by finding out how far from the top of the recommendation list the first successfully predicted location is.

$$\text{MRR@K} = \frac{1}{M} \sum_u \frac{1}{\text{ran } k_i}, \quad (20)$$

where rank_i refers to the rank position of the first relevant item for target user u in the recommended list.

4.2. Baseline and Parameter Settings. In order to prove the effectiveness of our proposed method, we compared the proposed IPAKG method with the following methods.

- (i) *FM* [38]: this is a typical similarity-based method that takes into account the second-order interactive information of the input features. In addition, this method is the basis of other baselines and has excellent performance on many benchmark datasets.

- (ii) *BPR* [14]: BPR is a classic method that uses matrix factorization to learn personalized ranking from implicit feedback.
- (iii) *GRU4Rec* [5]: this method improves the GRU network for session-based recommendation, which uses a session-parallel minibatch training process and also uses ranking-based loss functions to train the model.
- (iv) *GRU4Rec+* [39]: this is an improved version of GRU4Rec by implementing a new loss function and sampling approach
- (v) *SASRec* [22]: this method uses a self-attention mechanism with a left-to-right transformer to capture useful patterns in the user's sequences.
- (vi) *BERT4Rec* [23]: this method uses the bidirectional transformer model to learn the temporal behavior of users, which is a state-of-the-art sequential recommendation method.
- (vii) *KTUP* [36]: this is a translation-based user preference model, which transfers the entity embedding and relation embedding learned from knowledge graph to the user preference model and simultaneously training two different tasks.

Among them, BPR and FM are traditional recommendation methods that only consider user feedback without considering sequential information, while GRU4Rec, GRU4Rec+, SASRec, and BERT4Rec are sequential recommendation methods based on neural networks, and KTUP is a recommendation method based on neural networks and knowledge graph.

The experimental environment is CPU Intel (R) Core (TM) i9-9980XE @3.00 GHz, with 128 GB memory and two Titan XP graphics cards. We use the scientific computing library NumPy and SciPy for data processing and use TensorFlow to implement our algorithm. We randomly divide the training set, validation set, and test set with the ratio of 7:2:1 and ensure that each user has at least one item in the test set. For hyperparameters, we apply a grid search to find the best settings for each task. Specifically, the learning rate is searched in $\{0.0001, 0.001, 0.005, 0.01, 0.05\}$, the coefficient of L_2 regularization is searched in $\{10^{-5}, 10^{-4}, 10^{-3}, \dots, 10^1, 10^2\}$, the embedding size is searched in $\{32, 64, 96, 128, 256\}$, and the optimization methods include Adam and SGD. Finally, the learning rate is set to 0.001, L_2 coefficient is set to 10^{-3} , embedding size is set to 96, and the optimization method is set to Adam. For the rest of those parameters, we follow the original way for the setup of the parameters which have been proposed by their authors.

4.3. Performance Evaluation. We compare the performance of our proposed IPAKG method with all other baselines. The results on two datasets are reported on Table 3. From the results, we have the following observations.

In general, our proposed model obtains the best results in all five evaluation metrics on two datasets, which proves the

TABLE 3: Performance comparison of difference recommendation methods.

Models	Amazon (electronics)					Tmall				
	MRR	HR@10	HR@20	NDCG@10	NDCG@20	MRR	HR@10	HR@20	NDCG@10	NDCG@20
FM	0.0307	0.0473	0.0501	0.0387	0.0465	0.0865	0.1108	0.1193	0.0921	0.0976
BPR	0.0421	0.0787	0.0812	0.0589	0.0682	0.1123	0.1425	0.1634	0.1236	0.1398
GRU4Rec	0.0611	0.0923	0.1011	0.0724	0.0854	0.1522	0.1822	0.1903	0.1622	0.1701
GRU4Rec+	0.0786	0.1087	0.1103	0.0822	0.0943	0.1632	0.1944	0.2011	0.1776	0.1804
SASRec	0.0903	0.1203	0.1298	0.0921	0.0987	0.1782	0.2101	0.2187	0.1856	0.1911
BERT4Rec	0.0976	0.1358	0.1398	0.1013	0.1127	0.1956	0.2217	0.2302	0.2014	0.2176
KTUP	0.1067	0.1396	0.1423	0.1201	0.1288	0.2189	0.2298	0.2354	0.2133	0.2199
IPAKG	0.1123	0.1487	0.1533	0.1286	0.1301	0.2234	0.2407	0.2501	0.2276	0.2315

TABLE 4: Ablation study on two datasets.

Models	Amazon (electronics)					Tmall				
	MRR	HR@10	HR@20	NDCG@10	NDCG@20	MRR	HR@10	HR@20	NDCG@10	NDCG@20
IPAKG-B-KG	0.0776	0.0989	0.1043	0.0824	0.0973	0.1612	0.1844	0.1911	0.1736	0.1802
IPAKG-KG	0.0813	0.1058	0.1298	0.0913	0.1027	0.1812	0.2001	0.2187	0.1826	0.1896
IPAKG-B	0.0967	0.1226	0.1401	0.1001	0.1138	0.2011	0.2298	0.2354	0.2119	0.2107
IPAKG	0.1123	0.1487	0.1533	0.1286	0.1301	0.2234	0.2407	0.2501	0.2276	0.2315

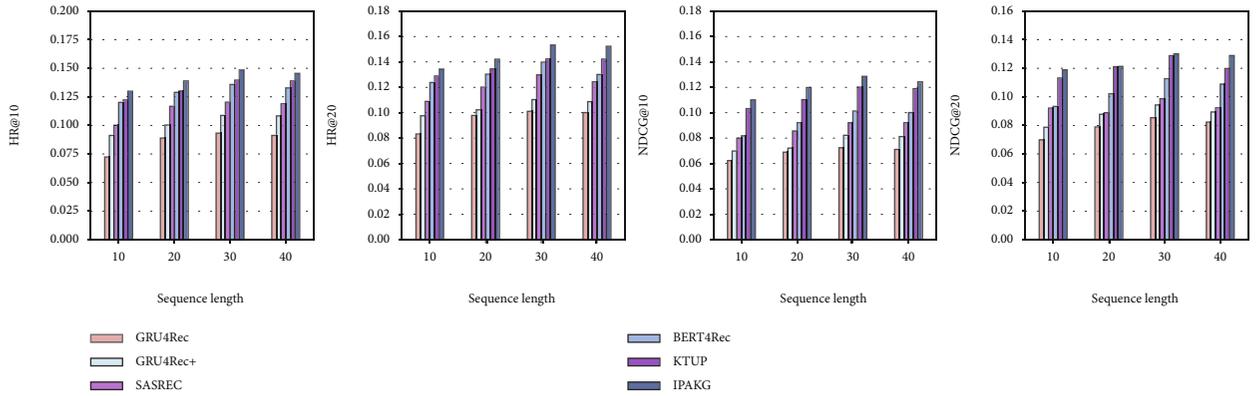


FIGURE 3: The effect of sequence length on results (Amazon electronics).

effectiveness of our model. Specifically, firstly, the IPAKG model and KTUP model both use the knowledge graph, but our model achieves better performance. This is mainly because our model uses the attention mechanism to model the interactive relationship and dynamic evolution of the user's preferences, which makes the user's preference representation more accurate. In addition, we also add user behavior information to model the user's preference degree on the certain item. Therefore, although our model and the KTUP model are both based on knowledge base technology, the model we proposed obtains better recommendation performance. Secondly, IPAKG has better performance compared to GRU4Rec and GRU4Rec+. One possible reason is that these two methods are based on conversational models and do not explicitly model the general preference information of users. Thirdly, our proposed method achieves better performance than SASRec and BERT4Rec which are based on the attention mechanism. One of the main reasons is that

our model integrates the information of the knowledge graph, through which the user's representation space are further expanded and the user's implicit preferences are fully exploited. Fourth, the above models obtain better experimental results on the Tmall dataset compared to the Amazon electronics dataset. This is mainly because the user interaction records of the Amazon electronics dataset are relatively sparse, which affects the representation of the data.

In addition, the performance of GRU4Rec+ on the above two datasets is significantly better than GRU4Rec. This is because GRU4Rec+ not only captures the sequence relationship in the user's interaction sequence but also has a better superior objective function. All the methods perform better than the traditional BPR and FM methods. This is because these two methods only model the user's general preferences and cannot model the user's sequential and dynamic preferences, which makes the preference representation insufficient. Furthermore, the main reason that the BERT4Rec

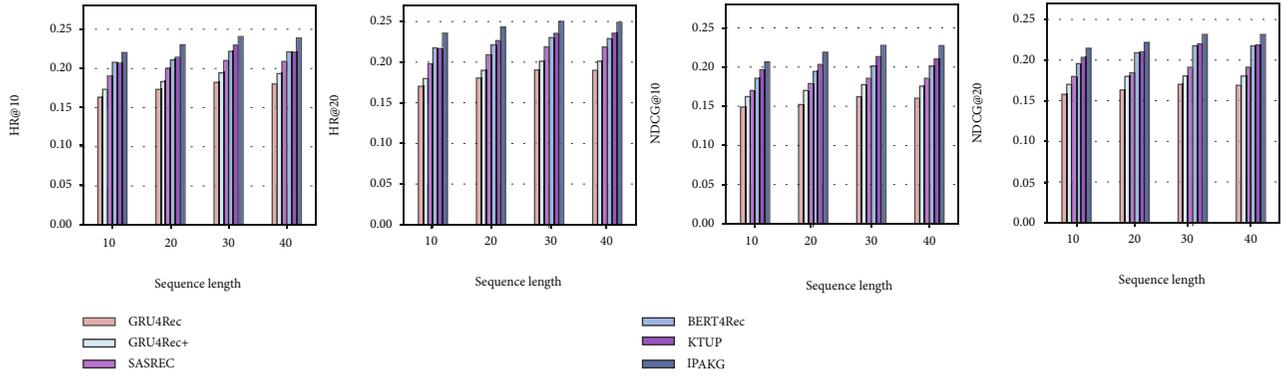
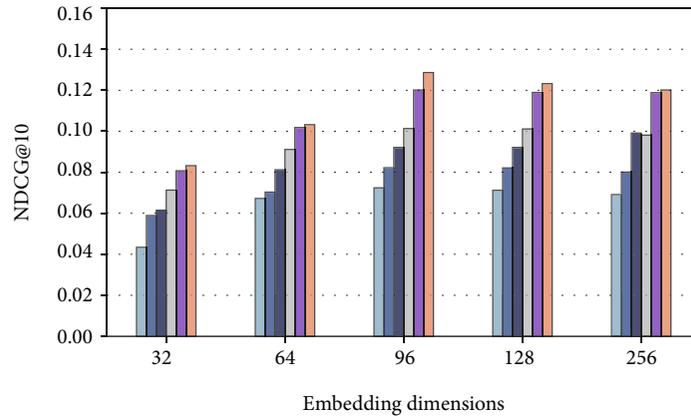
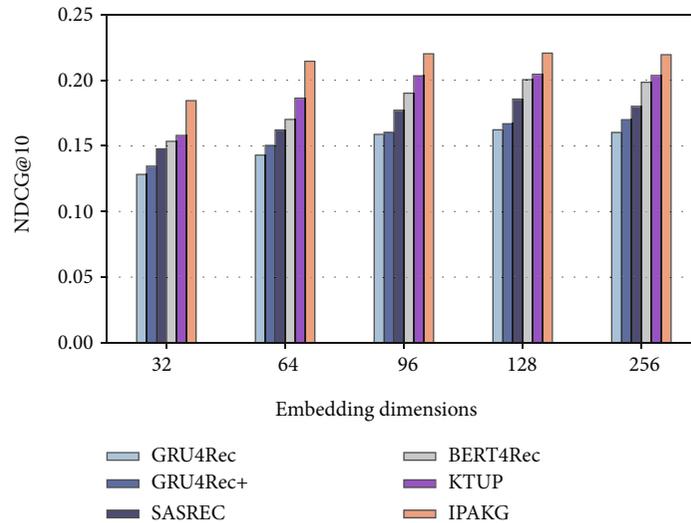


FIGURE 4: The effect of sequence length on results (Tmall).



(a) Amazon electronics



(b) Tmall

FIGURE 5: NDCG@10 values on different embedding dimensions.

model performs better than SASRec is that BERT4Rec can jointly predict the target item based on the context information on the left and right sides, so it can achieve better performance.

4.4. Ablation Study. In order to verify the contribution of each main module of our model to the overall performance,

we conduct an ablation study. We first remove the behavioral activation module separately from the original model, which means that we no longer consider the degree of preferences on items implied by the user’s different behaviors, and we call this modified model IPAKG-B. In order to verify the contribution of the knowledge graph-embedding technology to the model, we removed the knowledge graph-embedding model

and replaced it with conventional embedding technology. In this case, we did not consider the expansion information of the user's preferences and implicit preferences; we call this modified model IPAKG-KG. Finally, we remove the knowledge graph-embedding module and behavioral activation module at the same time, which means that the remaining model becomes a conventional sequential recommendation model based on recurrent neural network and attention mechanism. We call this modified model IPAKG-B-KG. The experimental results of the ablation study are shown in Table 4.

Based on the above experimental results, we have the following findings. First of all, when we removed the knowledge graph-embedding module and behavior activation module, the performance of the model is obvious decline, which means that both the knowledge graph-embedding module and the behavioral activation module play a significant role on the performance of the model. Secondly, the user's behavior activation module has a smaller impact on the model performance than the knowledge graph-embedding module. This means that the user's implicit preferences and expanded preferences have a more important influence on the recommendation. Thirdly, after removing the knowledge graph-embedding module and behavior activation module, the overall performance of the model is between GRU4Rec and GRU4Rec+. This is because that these models have a relatively similar structure, and thus, their performance is relatively similar.

4.5. Analysis of Main Parameters

4.5.1. The Effect of Sequence Length on Experimental Results.

In order to study the impact of different interaction lengths on the recommendation performance, we conduct experiments on two datasets. Figures 3 and 4 report the results of different sequence lengths on the performance of each model. It can be seen from the experimental results that for the Amazon electronics dataset, each model has better performance when the sequence length is about 40, and for the Tmall dataset, each model performs better when the sequence length is about 60. This means that the optimal sequence length depends on the characteristics of the dataset. When the length of the sequence is too short, the model only capture the user's recent preferences, and when the length of the sequence is too long, certain noise data may be introduced, both of which are not conducive to the performance of the model.

4.5.2. The Impact of Embedding Dimensions on Results.

Figure 5 shows the impact of different embedding dimensions on the recommendation performance. The hidden dimension d varies from 32 to 256, while the other optimal hyperparameters remain unchanged. The most obvious observation from these subgraphs is that the performance of each model does not increase as the dimension increases. Larger hidden dimensions do not necessarily lead to better model performance, especially for these datasets with sparse data. This may be caused by overfitting. As can be seen from the figure, the knowledge graph-based methods (that is,

KTUP and IPAKG) have relatively better performance than other methods on the two datasets. For our proposed IPAKG model, even if the hidden dimension is relatively small, our model performance is relatively better than other models.

5. Conclusion

In this paper, we propose a novel sequential recommendation method called IPAKG based on knowledge graph for next-item recommendation task. This method expands the representation space of the item by introducing the knowledge graph-embedding technology and fully exploits user's implicit preferences. Furthermore, the user's sequential behavior information is modeled through the behavioral activation unit, by which the degree of the user's preferences on certain items is fully quantified, which makes the user's preference representation more accurate. The empirical results demonstrate that our model can significantly outperform the baseline models on two real-world datasets. In addition, we conduct a detailed analysis on the IPAKG model to illustrate the effectiveness of our method. At present, our method mainly considers the extended information of user interaction items but does not model the information of similar users. In the next step, we will consider using the knowledge graph to further model the information of similar users to further improve the performance of recommendation.

Data Availability

The original dataset used in this work is available from the corresponding author on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported partly by the National Natural Science Foundation of China under Grant No. 61772285 and the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing.

References

- [1] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, "Parallel recurrent neural network architectures for feature-rich session-based recommendations," in *Proceedings of the 10th ACM conference on recommender systems*, pp. 241–248, New York, NY, USA, 2016.
- [2] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, "Learning hierarchical representation model for nextbasket recommendation," in *Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 403–412, New York, NY, USA, 2015.
- [3] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proceedings of the 19th international conference on World wide web*, pp. 811–820, New York, NY, USA, 2010.

- [4] T. Donkers, B. Loepp, and J. Ziegler, "Sequential user-based recurrent neural network recommendations," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pp. 152–160, New York, NY, USA, 2017.
- [5] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session based recommendations with recurrent neural networks," 2015, <https://arxiv.org/abs/1511.06939>.
- [6] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, "Recurrent recommender networks," in *Proceedings of the tenth ACM international conference on web search and data mining*, pp. 495–503, New York, NY, USA, 2017.
- [7] A. Yan, S. Cheng, W.-C. Kang, M. Wan, and J. McAuley, "Cos-Rec: 2D convolutional neural networks for sequential recommendation," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 2173–2176, New York, NY, USA, 2019.
- [8] X. Huang, S. Qian, Q. Fang, J. Sang, and C. Xu, "CSAN: contextual self-attention network for user sequential recommendation," in *Proceedings of the 26th ACM international conference on Multimedia*, pp. 447–455, New York, NY, USA, 2018.
- [9] W. Ye, S. Wang, X. Chen, X. Wang, Z. Qin, and D. Yin, "Time matters sequential recommendation with complex temporal information," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1459–1468, New York, NY, USA, 2020.
- [10] C. Zhou, J. Bai, J. Song et al., "ATRank: an attention-based user behavior modeling framework for recommendation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [11] R. Ren, Z. Liu, Y. Li et al., "Sequential recommendation with self-attentive multi-adversarial network," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 89–98, New York, NY, USA, 2020.
- [12] J. Li, Y. Wang, and J. McAuley, "Time interval aware self-attention for sequential recommendation," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 322–330, New York, NY, USA, 2020.
- [13] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426–434, New York, NY, USA, 2008.
- [14] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," 2012, <https://arxiv.org/abs/1205.2618>.
- [15] X. Yang, S. Zhou, and M. Cao, "An approach to alleviate the sparsity problem of hybrid collaborative filtering based recommendations: the product-attribute perspective from user reviews," *Mobile Networks & Applications*, vol. 25, no. 2, pp. 376–390, 2020.
- [16] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-n recommender systems," in *Proceedings of the ninth ACM international conference on web search and data mining*, pp. 153–162, New York, NY, USA, 2016.
- [17] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "xDeepFM: combining explicit and implicit feature interactions for recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1754–1763, New York, NY, USA, 2018.
- [18] R. Togashi, M. Otani, and S. Satoh, "Alleviating cold-start problems in recommendation through pseudo-labelling over knowledge graph," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 931–939, New York, NY, USA, 2021.
- [19] R. He and J. McAuley, "Fusing similarity models with Markov chains for sparse sequential recommendation," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 191–200, Barcelona, Spain, 2016.
- [20] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pp. 130–137, New York, NY, USA, 2017.
- [21] J. Tang and K. Wang, "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, New York, NY, USA, 2018.
- [22] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 197–206, Singapore, 2018.
- [23] F. Sun, J. Liu, J. Wu 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, New York, NY, USA, 2019.
- [24] M. Zhou, Z. Ding, J. Tang, and D. Yin, "Micro behaviors: A new perspective in e-commerce recommender systems," in *Proceedings of the eleventh ACM international conference on web search and data mining*, pp. 727–735, New York, NY, USA, 2018.
- [25] Y. Gu, Z. Ding, S. Wang, and D. Yin, "Hierarchical user profiling for e-commerce recommender systems," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 223–231, New York, NY, USA, 2020.
- [26] W. Meng, D. Yang, and Y. Xiao, "Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1091–1100, New York, NY, USA, 2020.
- [27] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, "Explainable reasoning over knowledge graphs for recommendation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 5329–5336, 2019.
- [28] W. Lei, G. Zhang, X. He et al., "Interactive path reasoning on graph for conversational recommendation," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2073–2083, New York, NY, USA, 2020.
- [29] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *Proceedings of the 795 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 950–958, New York, NY, USA, 2019.
- [30] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, 2015.
- [31] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," 2017, <https://arxiv.org/abs/1706.03762>.

- [32] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, <https://arxiv.org/abs/1409.0473>.
- [33] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, <https://arxiv.org/abs/1412.6980>.
- [34] R. He and J. McAuley, "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, New York, NY, USA, 2016.
- [35] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pp. 43–52, New York, NY, USA, 2015.
- [36] Y. Cao, X. Wang, X. He, Z. Hu, and T.-S. Chua, "Unifying knowledge graph learning and recommendation: towards a better understanding of user preferences," in *The world wide web conference*, pp. 151–161, New York, NY, USA, 2019.
- [37] X. Chen, H. Xu, Y. Zhang et al., "Sequential recommendation with user memory networks," in *Proceedings of the eleventh ACM international conference on web search and data mining*, pp. 108–116, New York, NY, USA, 2018.
- [38] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme, "Fast context-aware recommendations with factorization machines," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 635–644, New York, NY, USA, 2011.
- [39] B. Hidasi and A. Karatzoglou, "Recurrent neural networks with top-k gains for session-based recommendations," in *Proceedings of the 27th ACM international conference on information and knowledge management*, pp. 843–852, New York, NY, USA, 2018.