

## APPLIED RESEARCH

# A Recommendation Strategy Integrating Higher-Order Feature Interactions With Knowledge Graphs

YANRU LIN<sup>1</sup>, SHIYU DU<sup>2</sup>, YIMING ZHANG<sup>2</sup>, KAI DUAN<sup>1</sup>, QING HUANG<sup>2</sup>, AND PENG AN<sup>3</sup><sup>1</sup>Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China<sup>2</sup>Engineering Laboratory of Advanced Energy Materials, Ningbo Institute of Materials Technology and Engineering, Chinese Academy of Sciences, Ningbo 315201, China<sup>3</sup>College of Electronics and Information Engineering, Ningbo University of Technology, Ningbo, Zhejiang 315211, China

Corresponding author: Shiyu Du (dushiyu@nimte.ac.cn)

This work was supported in part by the Key Research and Development Projects of Zhejiang Province under Grant 2022C01236 and Grant 2019C01060; in part by the National Natural Science Foundations of China under Grant 52250005, Grant 21875271, Grant U20B2021, Grant 21707147, Grant 51372046, Grant 51479037, Grant 91226202, and Grant 91426304; in part by the Entrepreneurship Program of Foshan National Hi-Tech Industrial Development Zone; in part by the Major Project of the Ministry of Science and Technology of China under Grant 2015ZX06004-001; and in part by the Ningbo Natural Science Foundations under Grant 2014A610006, Grant 2016A610273, and Grant 2019A610106.

**ABSTRACT** Knowledge Graphs (KG) are efficient auxiliary information in recommender systems. However, in knowledge graph feature learning, a major objective is improvement for recommendation performance. One of the problems in the existing methods is that they cannot uncover the deep interaction information of users in a simple way, and this motivates effective learning of the potential embedded information through the knowledge graph. The Graph Convolutional Network (GCN) can be useful for learning information about graph structured data. This paper proposes a method that fuses higher order feature interactions and knowledge graphs and uses them for recommendation. For users, they use Gated Recurrent Units (GRU) to focus on their preferences so that the ability of convolutional neural networks in processing user preference features is enhanced; for items, the cross-learning module is adopted to learn higher order features between items and entities; for users and entities, KG and user-item interaction information are combined followed by feature extraction of graph structured data by Light GCN, allowing the model to learn potential user-entity associations from the graph structured data. Current experiments on two real datasets show that the proposed model performs better than some recently developed models.

**INDEX TERMS** Recommendation systems, knowledge graphs, convolutional neural networks.

## I. INTRODUCTION

Online information browsing has permeated people's daily lives as the Internet era has progressed [1]. The information overload is becoming an issue due to the growth of the Internet, making it difficult for users to sift through the mass of information to find their truly interests. Recommendation systems have been developed as a result of the urgent requirement for an effective information filtering mechanism [2]. The aim of recommendation system is to extract the user's

needs and interests from their previous interactions with the items and then provide the findings to the user. Content-based recommendation [24], [25], [26], [37], collaborative filtering-based recommendation [24], [38], and their combinations [27], [28], [40], [41] are three primary types of recommendation algorithms. The recommendation accuracy keeps increasing from the initial periods of machine learning to the present day of deep learning [8]. Aiming to address the sufferings from data sparsity [43] and cold-start issues, a common solution is to incorporate different auxiliary information [40] into recommendation models, including user/item attributes, social networks or contextual information.

The associate editor coordinating the review of this manuscript and approving it for publication was Alireza Sadeghian.

Knowledge graphs could offer logical justifications for the recommendation outcomes since they have structured semantic knowledge of auxiliary information. For the cold start problem, most of the existing solutions are achieved by enhancing knowledge embedding, adopting heterogeneous graphs to enrich data representation or using Graph Neural Networks (GCN). In this paper, we focus on alleviating the cold start problem by introducing graph convolutional neural networks to explore users' potential interests.

Major types of knowledge graph-based recommendation systems include the embedding-based method and the path based approach. In embedding-based techniques, Knowledge graph embedding is carried out to preprocess KG, and the item embedding representation is subsequently used in recommendations. In order to obtain an embedding representation, Collaborative Knowledge Base Embedding (CKE) [6] analyzes the structural data of KG using TransR [3], while Deep Knowledge-aware Network (DKN) [11] builds a knowledge graph by extracting text data and fusing the word vectors of news with the entities of the knowledge graph. The path-based approach involves constructing a user-item graph and using the connectivity of entities in the graph to make recommendations. The recommendation results are improved by the similarity of connections between users and/or items, but this approach heavily depends on manually created meta-paths. The "matrix factorization (MF) + factorization machine (FM)" method is used by Meta-Graph Based Recommendation Fusion [12] to integrate meta-graph into the Heterogeneous Information Network (HIN) based recommendation system and address the information fusion issues. The MF technique is used to produce potential features of users and possible elements of items for each meta-graph similarity. A group lasso regularization-based FM approach is presented to automatically choose the appropriate meta-graph method from the observed data for various meta-graph attributes. When coding a given entity representation, Wang et al. [29] proposes a model Knowledge Graph Convolutional Networks (KGCN) that fuses KG features and graph convolutional neural networks to integrate neighbour information and deviations to acquire neighbourhood structure more effectively. This model enables users to receive more personalized interest recommendations.

In recent years, deep learning has been increasingly applied to recommendation algorithms owing to the widespread use of these techniques in image processing, machine translation, and intelligent questioning and answering systems. The enhanced efficiency of recommendations originates from neural networks' effective access to deep semantic information among data. Yang et al. [14] introduces a novel Knowledge-enhanced Deep recommendation framework incorporating Generative and Adversarial Network (KTGAN), which is a GAN-based model with two stages. The first stage is to learn KG embedding in combination with Metapath2Vec [36] and tag embedding in combination with Word2Vec [35], while the second stage uses a generator G and a discriminator D to refine the user's initial

representation. The knowledge-enhanced Sequential Recommender (KSR) [15] framework combines Gated Recurrent Units (GRU) network with a Key-Value memory network, where the GRU network is used to capture the serialized preferences of the user, and the Key-Value memory network models the attribute-level preferences of the user through Trans E [34]. In this work, we introduce knowledge graphs as auxiliary information into the recommendation model and apply deep learning techniques to learn user-item interaction information from the supplementary data. The main contributions of this paper are as follow:

- (1) The cross-learning module is proposed for mining explicit and implicit higher-order interactions between items, which provides auxiliary information for recommendations.
- (2) To enrich the relationship between users and entities, a user-item-entity graph is constructed, and a graph feature learning module is used to explore the semantic relationships in the graph structure data to better improve the recommendation performance.
- (3) We conduct extensive experiments on two benchmark datasets, and the results show that the proposed model outperforms the other benchmark models.

## II. RELATED WORK

### A. RECOMMENDER SYSTEM

With the development and application of deep learning technology, neural networks have been applied to all aspects of recommendation systems, and the recommendation algorithms have experienced from content-based recommendations to collaborative filtering-based recommendations, which are further developed from user- and item-based collaborative filtering to model-based Collaborative Filtering (CF). Since deep learning can effectively mine data features, researchers have introduced large amount of auxiliary information, such as text, tags, social relationships, etc., to extract favorable features for recommendations. ConvMF [18] model embeds convolutional neural networks into maximum likelihood estimation to generate recommendations based on review text analysis. Zhou et al. proposes Deep Interest Network (DIN) [19], presenting a deep learning method based on mini-batch regularization and data-adaptive activation function, which adds an attention mechanism after the embedding layer, aiming to generate user features from the recommended items. The Wide&Deep [20] model merged linear models and Deep Neural Network (DNN) models, which utilize the memory capability of the shallow model and the generalization capability of the deep model in a comprehensive way, enabling the proposed model to combine the features of accuracy and scalability, and thus achieving improved feature extraction capabilities. Knowledge-aware Path Recurrent Network (KPRN) [21], which uses knowledge graphs as auxiliary information for recommendation, employs Recurrent Neural Networks (RNN) to learn users' preferences, and finally uses path information to enhance the interpretability of recommendations. MKR [22] is

a multi-task learning recommendation framework that combines the knowledge graph learning task with the recommendation task well for training, resulting in improved performance of the final recommendation.

### B. KNOWLEDGE GRAPH-BASED RECOMMENDATION

Knowledge graph has powerful semantic expression capability and is essentially a knowledge base of semantic network. Google proposed the knowledge graph concept in 2012, aiming to use the knowledge base to improve the performance of search engines and enhance the user's search efficiency [44]. With the development and application of artificial intelligence, knowledge graph has been widely used in intelligent search, intelligent Q&A and personalized recommendation [44]. Presently, the application of knowledge graph feature learning to recommender systems is mainly through three methods, i.e. *sequential learning*, *joint learning*, and *multi-task learning*, with typical DKN, CKE, and MKR models, respectively. In this study, we will conduct modeling by using a multi-task learning strategy.

The embedding-based approaches [6], [7], [9], [10], [11], [13], [15] characterize entities and relationships by knowledge graph embedding (KGE) [3], [4], [34], which in turn obtains additional semantic information. Relations provide additional information for knowledge graph-based recommendations that can use inter-node reasoning to discover new connections [45]. DKFM [7] is proposed for points of interest (POI) recommendation, using TransE [34] to train urban data for enriching the representation of destinations as well as for improving the relevant recommendation performance. BEM [10] uses two types of graph data, i.e. knowledge-related graphs and behavioral graphs, for firstly learning the initial embedding representation from the graph using TransE and Graph Neural Networks (GNN), respectively. After that, a Bayesian framework is used to extract the two types of embeddings. KTUP [9] uses TransH [4] to jointly learn the enhanced item-preference embeddings and the entity-relationship embeddings. Path-based approaches [5], [8], [17], [18] aim to uncover multiple connections between users and items based on graphs to explore potential information in knowledge graphs. McRec [5] designs a mutual attention mechanism based on meta-paths for top-K recommendations, extracts potential features of meta-paths, and exploits connectivity between users and different types of relational paths or items on graphs. The KPRN [17] exploits the sequential dependencies in paths, making it possible to reason effectively over the paths. Although the path-based approach makes more natural use of the knowledge graph, but it relies heavily on the design of meta-paths, which require specialized domain knowledge for effective meta-paths.

### III. PROBLEM FORMULATION

In the study of knowledge graph-enhanced recommendations for a particular recommendation scenario, we have a collection of users  $U = \{u_1, u_2, \dots, u_m\}$  and a group of items  $V = \{v_1, v_2, \dots, v_n\}$ ; where  $m$  and  $n$  denote the

TABLE 1. Notations and explanations.

Notations	Description
$U = \{u_1, u_2, \dots\}$	the set of users
$V = \{v_1, v_2, \dots\}$	the set of items
$\mathcal{G} = (\mathcal{E}, \mathcal{R})$	Knowledge graph
$(h, r, t)$	the knowledge graph triple
$\mathcal{E} = \{e_1, e_2, \dots\}$	the set of entities
$\mathcal{R} = \{r_1, r_2, \dots\}$	the set of relations
$y_{ui} \in Y$	Users' implicit feedback
$\hat{y}_{ui}$	Predicted engaging probability
$\hat{t}$	the predicted tail of $t$
$e_u$	the embedding of user
$e_i$	the embedding of item
$F$	the CIFL module

corresponding numbers. Typically, the user-item interaction data is represented as a bipartite graph, and the user-item interaction matrix  $Y \in R_{m \times n}$  is established based on implicit user feedback.

$$y = \begin{cases} 1, & \text{if } (u, i) \text{ is interact} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The presence of the  $y_{ui} = 1$  here denotes user interaction with the item, which may take actions like browsing, clicking, purchasing, rating, etc; and  $y_{ui} = 0$  indicates that the user did not engage with the item, yet it does not imply that the user disapproved of the item. It is also possible that the user did not navigate to the item since no interaction took place. The knowledge graph  $\mathcal{G}$  is made up of entity-relationship-entity triples  $(h, r, t)$ , with  $h$ ,  $r$ , and  $t$  standing for the head entity, relationship, and tail entity, respectively. For example, the triplet (*Paul Kalanithi*, *book. written\_work. author*, *When Breath Becomes Air*) identifies *Paul Kalanithi* as the writer of "*When Breath Becomes Air*". Knowledge graphs provide factual knowledge and rich semantic information about items. In many recommendation scenarios, an item  $v \in V$  corresponding to one or more entities, so that we could establish item-entity alignments. Through the alignment of items and knowledge graph entities, the knowledge graph can provide complementary information for interaction information.

The challenge of our knowledge-aware recommendation is to forecast the potential interest of a user  $u$  in an unobserved item  $v$  given a user-item interaction matrix  $Y$  and a knowledge graph  $\mathcal{G}$ . The aim is to train a prediction function  $\hat{y}_{uv} = F(u, v | \theta, Y, \mathcal{G})$ , where  $\theta$  is the model parameter of the function  $F$ , to calculate the chance of user  $u$  selecting item  $v$ . Table 1 lists a description of each symbol used in this paper.

### IV. METHODOLOGY

In this section, we provide an overview of the general framework before detailing our proposed approach and presenting the implementation details of the specific modules individually.

#### A. AN OVERVIEW

The general framework of our model is shown in Figure 1, which consists of three main tasks; that is, knowledge graph

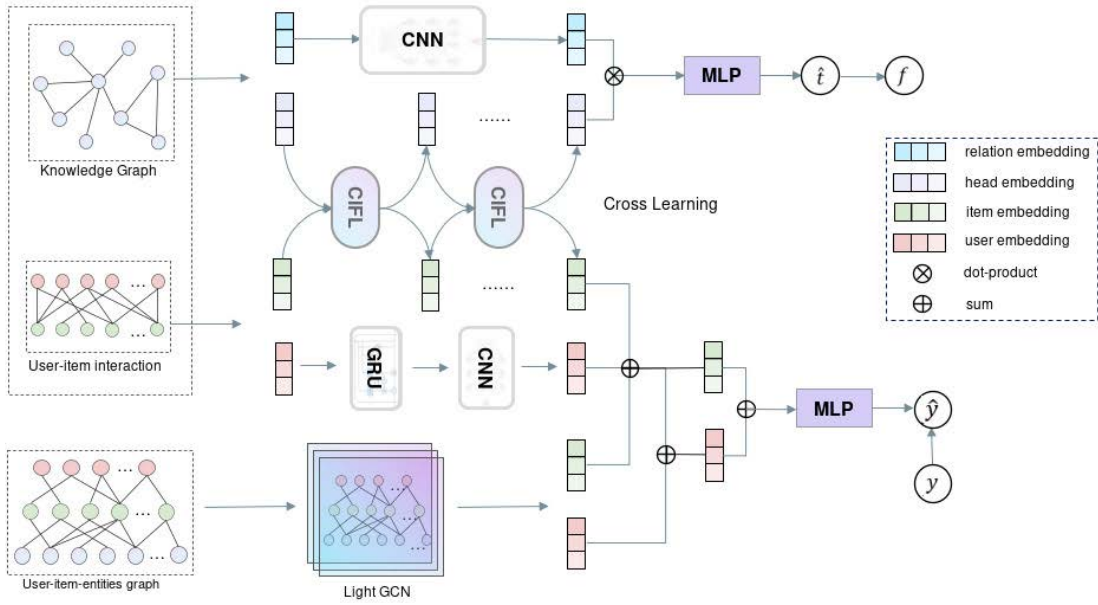


FIGURE 1. Overall framework of the proposed model.

embedding learning, feature learning of graph data, and recommendation module. (a) Knowledge graph embedding learning is a task that mainly focuses on learning the knowledge graph tripple data and linking the knowledge graph feature learning with the recommendation task through a Cross Item-Feature Learning (CIFL) module for data sharing and parameter transfer to achieve better recommendations. For this task, the head  $h$  and relation  $r$  of the knowledge graph are used as input, and  $h$  and  $r$  are obtained through processing the CIFL module and Convolutional Neural Networks (CNN) respectively. A Multilayer Perceptron (MLP) is implemented to predict the tail, with the use of a similarity function as the evaluation of the final result. (b) Graph data feature learning module, through combining the knowledge graph data and user-item interaction data into a user-item-entity graph, to process the user-item-entity graph by a Light GCN network for extracting the fine-grained information of users and items. (c) Recommendation module is linked with the knowledge graph embedding learning module through the CIFL module. The user and item embedding vectors obtained through the graph data feature learning module are stitched together with the current user and item embedding vectors, and finally the final results are predicted using a multilayer perceptron.

## B. GRAPH DATA FEATURE LEARNING

The items might be represented by how the users interact; however, in the past, only the user-item graph was taken into account. The representation of how information is interacted with is too broad, so we wish to know the fine-grained characteristics more thoroughly. In order to build the user-item-entity graph, which preserves more comprehensive structural information, we integrate the user-item interaction graph with

the knowledge graph. Therefore, in order to automatically encode the route information within the nodes and maintain the information of nearby nodes and the user-item-entity path information, we would treat the structural information using Light GCN, a graph encoder with path-awareness. In general, graph data feature learning is to learn the relationship between users and entities in the graph, so as to explore the deeper features of users. The light GCN is a light yet efficient GCN network that forgoes nonlinear activation and typical feature transformation in favor of a simple message transmission mechanism and the most fundamental GCN aggregation formulation at the  $k$  layer, is illustrated in Equation 2.

$$e_u^{(k+1)} = \sum_{i \in N_u} \frac{1}{\sqrt{|N_u|} |N_v|} e_v^{(k)} \quad (2)$$

$$e_v^{(k+1)} = \sum_{u \in N_v} \frac{1}{\sqrt{|N_u|} |N_v|} e_u^{(k)} \quad (3)$$

where  $N_u$  and  $N_v$  represent the set of items that user  $u$  and item  $v$  have interacted with, respectively, and  $e_v^{(k)}$  and  $e_u^{(k)}$  denote the embeddings of user  $u$  and item  $v$  at the  $k^{th}$  level. The symmetric regularization term  $\frac{1}{\sqrt{|N_u|} |N_v|}$  follows the standard GCN structure with the purpose of avoiding the size of the embedding to increase with the graph convolution operation. We then sum over them to obtain the most embedding representations of the users and the items.

$$e_u = \sum_{k=0}^K \alpha_k e_u^{(k)} \quad (4)$$

$$e_v = \sum_{k=0}^K \alpha_k e_v^{(k)} \quad (5)$$



### C. CROSS ITEM-FEATURE LEARNING

Based on the user-item interaction information, we can infer the user's item preferences. To better learn the interaction information between items and entities, we design a CIFI module, which models the higher-order interactions between items and their corresponding entity features. We utilize the main module of XDeepFM [23], a Compressed Interaction Network (CIN), for item preference extraction with explicit higher-order features. The idea of CIN is to achieve automatic explicative learning of higher-order feature interactions so that the feature interactions occur between individual feature vectors, i.e., feature interactions at the vector level. The CIN network includes CNN and RNN, which can adjust the feature order by controlling the number of layers of the neural network so that feature interactions occur vector-wise, effectively avoiding the defect that the number of model parameters increases exponentially with the number of layers, thus achieving the purpose of efficient extraction. In the CIN module, the output of each layer is used as the input of the next layer, and the neurons of each layer are calculated based on the hidden layer of the previous layer and the original feature vector is calculated as shown below.

$$X_{h,*}^k = \sum_{i=1}^{H_{k-1}} \sum_{j=1}^m W_{ij}^{k,h} (X_{i,*}^{k-1} \circ X_{j,*}^0) \quad (6)$$

The original features of the input and hidden layers of the neural network are converted into matrices denoted as  $X^0$  and  $X^k$ , respectively, where the  $i$ -layer embedded feature vector is presented by  $X_{i,*}^0 = e^i$ ,  $H_k$  is the  $i$ -row of  $X^0$ ,  $H_k$  is the number of feature vectors in the  $k$  layer,  $x_{i,*}^{k-1}$  is the  $i$  layer row of the output matrix of the  $k-1$  layer in the CIN,  $X_{j,*}^0$  is the  $j$ -layer row feature vector of the original feature, and  $w^{k,h} \in R^{H_{k-1} \times m}$  denotes the parameter matrix of the  $h^{th}$  feature vector. Equation 6 allows us to divide the CIN model computation into three stages. The output state  $X^k$  of the preceding hidden layer and the output matrix  $X^0$  of the embedding layer is used to compute  $X_{i,*}^{k-1} \circ X_{j,*}^0$  so as to provide an intermediate result of  $Z_{k+1}$ , which is the first stage. The output of the next hidden layer is produced in the second stage by performing a layer-by-layer feature mapping operation on the intermediate result of the first stage and using a convolution kernel of  $mH$  vectors at  $H_{k+1}$  positions. Each feature mapping matrix created in the second stage is added together and merged in the third stage to produce  $P_i^k = \sum_{j=1}^D X_{ij}^k$ . A DNN is used to facilitate feature interactions created in the bit-wise level, allowing each feature to interact implicitly with other features and thereby satisfying the higher-order feature interactions between various components. This allows collecting of more data regarding the implicit interaction of items. The following equations are used to calculate implicit interaction information.

$$\mathbf{x}^l = \sigma(\mathbf{W}^{(l)} \mathbf{e} + \mathbf{b}^l) \quad (7)$$

$$\mathbf{x}^k = \sigma(\mathbf{W}^{(k)} \mathbf{x}^{(k-1)} + \mathbf{b}^k) \quad (8)$$

where  $W^k$  stands for the weight between the  $k$ -layer and  $(k+1)$ -layer, and  $b^k$  represents for the bias of the  $(k+1)$ -layer.  $x^k$  is the output of the  $k$ -layer,  $\sigma$  is the activation function. That is, the higher-order features are implicit higher-order interactions at the element level.

### D. KNOWLEDGE GRAPH EMBEDDING MODULE

The knowledge graph, as a powerful knowledge network structure, mainly consists of the triad  $\{(h, r, t) \mid h, t \in E, r \in R\}$ , where  $E$  is the set of entities in the knowledge graph and  $R$  is the set of relations. The knowledge graph representation learning refers to embedding the entities and relations composing knowledge into a low-dimensional vector space while preserving the structural and semantic information of KG. In this work, a semantic matching-based knowledge graph learning model is used to mine the attribute features of items so as to find out the potential connections among items, connecting the knowledge graph feature learning module and the recommendation module through the CIFI module, and then obtaining the vector representation of items using the training method of multi-task learning. Firstly, the head  $h$  in the knowledge graph triple is processed using the CIFI module; secondly, the relationship  $r$  is processed using the convolution module; thirdly,  $h$  and  $r$  are stitched together; its dimensionality is reduced by an  $l$ -layer MLP, to obtain a prediction ( $\hat{t}$ ). The similarity function is used to evaluate the prediction results.

$$h = F(V, h)[h] \quad (9)$$

$$r = C(r) \quad (10)$$

$$t = MLP\left(\begin{bmatrix} h \\ r \end{bmatrix}\right) \quad (11)$$

where  $C$  stands for the convolutional neural network, and  $F$  for the CIFI module. Equation 12 illustrates how a similarity function may be used to get the highest similarity of the triplet  $(h, r, t)$ .  $\sigma$  is the sigmoid activation function and  $\hat{t}$  is the prediction vector for tail  $t$ .

$$\text{similarity}(t, \hat{t}) = f_{\text{kg}}(t, \hat{t}) = -\sigma(t^\top \hat{t}) \quad (12)$$

### E. RECOMMENDATION MODULE

The inputs for the model are the original feature vectors of users and items, which are fed into the embedding layer of users and items to obtain the low-dimensional dense vector representations of users and items. The user embedding set is  $U \in R$  and the item embedding set is  $V \in R$ . The embedding vectors of users and items are represented as  $u \in R, v \in R$ . For different users, the items in the current sequence have different attention levels. Therefore, we use the gated cyclic unit based on the attention mechanism to obtain the item preferences for the current user in the sequence. We use CNN to obtain its preference feature distribution for the refined current preferences. For a given user's original feature vector  $u$ , the  $u_{gru}$  and  $u_{cnn}$  after GRU and CNN processing are

as follows:

$$u_{gru} = GRU(u) \quad (13)$$

$$u_{cnn} = CNN(u_{gru}) \quad (14)$$

For item  $v$ , we processed it using the CIFL module to obtain:

$$v_l = F(V, h)[h] \quad (15)$$

The  $e_u$  and  $e_v$  obtained by the graph data feature learning process are stitched together with the  $u_{cnn}$  and  $v_l$  received by the recommendation module to obtain the final  $u$  and  $v$ . Finally, the  $u$  and  $v$  are stitched together by an  $l$ -layer MLP for user  $u$  click-through rate prediction.

## F. PARAMETER LEARNING

The loss function of our model is as follows:

$$\begin{aligned} L &= L_{RS} + L_{KG} \\ &= \sum_{u \in U, v \in V} \mathcal{J}(\hat{y}_{uv} - y_{uv}) + \sum_{(h, r, t) \in G} \text{Similarity}(t, \hat{t}) \\ &\quad + \lambda \|W\|_2^2 \end{aligned} \quad (16)$$

where  $L_{RS}$  is a measure of the loss value of the recommended model,  $L_{KG}$  is the loss in the computed knowledge embedding, with  $\mathcal{J}$  being the cross-entropy loss function, while the last term is a regularization term to prevent overfitting, with  $\lambda$  being the regularization term.

The learning parameters of this model are composed of two parts, i.e., the recommendation module and the KG module, and the training algorithm of our model is given in Algorithm 1. The parameters are shared and influence each other between these two parts. In our experiments, the model parameters are trained in a joint learning manner, i.e., joint training. The joint training optimizes the parameters of both the KG embedding part and the recommendation part during the training process, in such a way that the two parts can interact with each other to learn a more reliable representation of the final prediction. We also use alternate calls to the optimizer for multi-task learning, where in each iteration we first learn the features in the user-item-entity graph, then learn the knowledge embedding, and finally connect the two parts by a cross-learning module that performs gradient descent for parameter updating.

## V. EXPERIMENTS

In this work, several experiments are carried out to assess the efficacy of our proposed models. In this section, we present the experimental data collection, assessment metrics, baseline approach, and implementation details. The experiments and results are then analyzed.

### A. EXPERIMENT SETTINGS

For the purpose of evaluating the recommendation performance of our proposed model, we performed experiments on two benchmark datasets to answer the following questions.

### Algorithm 1 The Training Algorithm for Our Model

**Input:** Interaction Matrix  $Y$ , Knowledge Graph  $\mathcal{G}$

**Output:**  $F(u, v | \theta, Y, \mathcal{G})$

```

1: Initialize model parameters
2: Build a user-item-entity graph
3: for Number of training iterations do
4:   // Recommendation task
5:   while i+batchSize < len(Y) do
6:     Pass the miniBatch from Y into the recommenda-
       tion module
7:     Using the Adam optimizer, update the F parame-
       ter according to Eq.(2)-(5), (9)-(12)
8:     i+ = batchSize
9:   while i+batchSize < len(G) do
10:    Pass the miniBatch from  $\mathcal{G}$  into the recommenda-
        tion module
11:    Using the SGD optimizer, update the F parameter
        according to Eq.(6)-(8), (13)-(15)
12: return result

```

RQ1: How well does our suggested model do compared to contemporary knowledge-aware recommendation techniques?

RQ2: Do the key components really have an impact on improving the performance of the model?

RQ3: How does our suggested method perform when the hyperparameters are changed?

### 1) DATASETS DESCRIPTION

We evaluate this model using two benchmark datasets, Book-Crossing and Last.FM. These two datasets are from open source and differ in size and sparsity, making our experimental results more convincing using this dataset from two different domains.

Book-Crossing<sup>1</sup>: Book-Crossing is a dataset containing more than 1 million displayed ratings of books, ranging from 1 to 10.

Last.FM<sup>2</sup>: This collection of user preferences data is from the online music service Last.FM, with each user's top artists and play counts included in the dataset.

Because of the interactions in Book-Crossing and Last.FM are explicit feedback, we transform them into implicit feedback according to RippleNet [8]; where 1 denotes a positive sample, and we do not set a positive scoring threshold for Book-Crossing and Last.FM data due to their sparsity. For the construction of sub-KG, we used Microsoft Satori where each sub-KG follows a triple. Each sub-KG follows a triple format and is a subset of the whole KG with a confidence level greater than 0.9. For a given sub-KG, we filter the triple with a good match from the sub-KG by comparing the name of the dataset with the head entity of the triple. The basic statistics of these two datasets are shown in Table 2.

<sup>1</sup><http://www2.informatik.uni-freiburg.de/cziegler/BX/>

<sup>2</sup><https://grouplens.org/datasets/hetrec-2011/>

**TABLE 2.** Statistics of the dataset.

		Book-crossing	Last.FM
User-Item	#Users	17,390	1,870
Interaction	#Items	143,610	3,843
	#Interactions	138,612	42,346
Knowledge Graph	#Entities	24,040	9,367
	#Relations	10	60
	#Triplets	19,680	16,710

## 2) EVALUATION METRICS

We assess our method in various experimental scenarios: (1) The threshold for a favorable rating in click-through-rate (CTR) prediction is set at 0.5. We would recommend item  $v$  to user  $u$  if  $y_{uv} > 0.5$ . We choose AUC and ACC (area under the ROC curve) as methods to evaluate CTR as we apply the training model to each interaction in the test sets. (2) In top-K recommendations, we pick Recall@K and Precision@K to assess the recommendation list and apply the trained model to choose the  $k$  items with the highest predicted click-through rate for each user in the test set. Equation 17 shows Recall is evaluated by the proportion of appropriate items recommended for the target user to the total number of relevant items. According to Equation 18, precision is the proportion of accurate item predictions to all item guesses.

$$\text{Recall} = \frac{\text{Number of correct prediction}}{\text{Number of total relevant Items}} \quad (17)$$

$$\text{Precision} = \frac{\text{Number of Correct Predictions}}{\text{Number of Total Predictions}} \quad (18)$$

## 3) BASELINES

In this work, we employ CKE, Neural Factorization Machine (NFM), Bayesian Personalized Ranking (BPR), Knowledge Graph Attention Network (KGAT), Explainable CF over Knowledge Graph (ECFKG), MKR, Collaborative Knowledge-aware Attentive Network (CKAN) and Knowledge Graph-based Intent Network (KGIN) as baselines; and the details are discussed as below.

CKE [6] is a typical embedding-based recommendation method that uses structured, textual and visual contents to make recommendations.

NFM [30] is a DNN structure introduced on the basis of FM model to learn more data information using nonlinear structure. NFM uses bi-linear interaction structure for second-order cross information, which makes DNN structure able to learn cross feature information better and reduces the difficulty of DNN to learn higher-order cross feature information.

BPR [31] is a Bayesian personalized ranking matrix decomposition framework which proposes a general optimization criterion for personalized ranking, BPR-Opt. In order to maximize BPR-Opt, a general learning algorithm based on stochastic gradient descent, Learn BPR, is utilized.

KGAT [32] is a knowledge graph-based model which applies attention mechanism to knowledge graph convolution to construct higher-order relationships.

**TABLE 3.** Performance comparison of different recommender systems.

model	Book-Crossing		Last.FM	
	AUC	ACC	AUC	ACC
CKE	0.627	0.617	0.742	0.631
NFM	0.696	0.687	0.777	0.724
ECFKG	0.696	0.697	0.700	0.702
KGAT	0.732	0.671	0.785	0.667
BPRFM	0.675	0.681	0.728	0.702
MKR	0.731	0.691	0.786	0.755
KGNN-LS	0.686	0.641	0.782	0.722
CKAN	0.739	0.652	0.796	0.753
KGIN	0.743	0.691	0.804	0.728
OURS	0.749	0.709	0.798	0.754

ECFKG [33] constructs a user-item knowledge graph. In this KG, users, items and their related attributes are treated as entities, and users' historical behaviors such as purchase and mention are regarded as a type of relationship between entities.

MKR [22] is a recommendation algorithm based on multi-task learning, which utilizes the recommendation module and the knowledge graph characterization module for alternate learning as a way to combine the user-item matching and knowledge graph embedding tasks.

CKAN [39] is built on the basis of KGNN-LS with independent collaboration propagation and knowledge signals on CF and KG, respectively, using a heterogeneous propagation strategy to encode both types of information and applying a knowledge-aware attention mechanism to distinguish the contributions of different knowledge neighbors.

KGIN [42] is to reveal user-item interactions at the intent granularity by aiding item knowledge, which models each intent as an attentional combination of KG relationships, enhances different intent independence, considering more fine-grained user-item relationships and remote semantics of relationship paths.

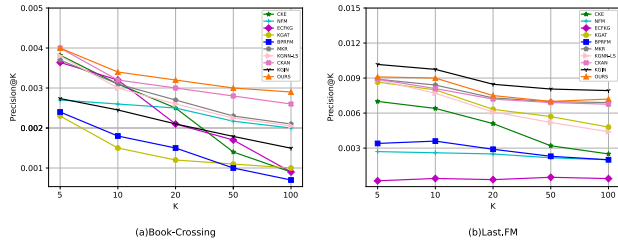
## 4) IMPLEMENTATION DETAILS

We train all our models on an NVIDIA GeForce RTX 2080 Ti with 11Gb of video memory; and for each dataset we divide it into training set, validation set, and testing set in a 6:2:2 ratio. We repeat each experiment 10 times and report the average results for tuning the key parameters. For comparisons, we have fixed the batch size of data for all of models as 128, initialized the embedding parameters using the Xavier method, and optimized our approach using the Adam optimizer. The optimal parameter settings were determined by grid search, adjusting the learning rate in  $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$  and the aggregation depth of the user-item-entity graph in  $\{1, 2, 3, 4\}$ . The optimal hyper-parameter settings for the comparison experiments are either from the actual study or follow the original paper.

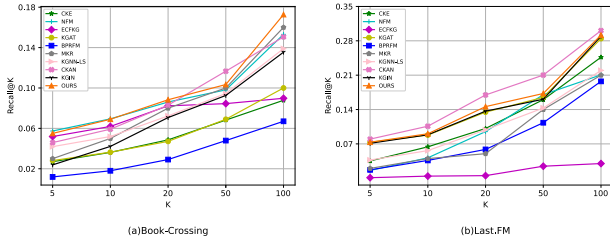
## B. EXPERIMENTAL RESULTS

### 1) PERFORMANCE COMPARISON (RQ1)

We show the results of our proposed model and some state-of-the-art methods for CTR prediction experiments and



**FIGURE 2.** Result of Precision@K in top-K recommendation. The left graph shows the precision results of Book-Crossing dataset with different K values, and the right graph is from Last.FM.

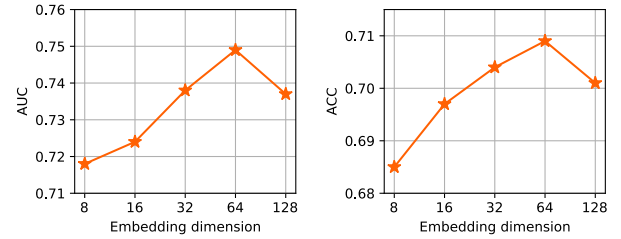


**FIGURE 3.** Result of Recall@K in top-K recommendation. The left graph shows the recall results of Book-Crossing dataset with different K values, and the right graph is from Last.FM.

top-K experiments on two real datasets in Table 3, Figure 2 and Figure 3. From the table, we can see that our proposed model outperforms all baselines on top-K and shows excellent performance on the CTR task. Knowledge-aware based models can obtain better performance, and we can observe from the experimental results that most of the knowledge graph based models outperform the traditional models, which indicates the introduction of knowledge graph can effectively enrich the relational information and effectively improve the recommendation. However, the simple integration of KG does not ensure the performance improvement, and the CF-based BPRMF approach is better than the embedding-based CKE approach, which indicates that the imbalance of the knowledge graph may lead to the degradation of the model performance. From the experimental results, we can demonstrate that our approach obtains better performance than the method only based on collaborative knowledge graphs and proves the effectiveness of extracting higher-order interaction feature information using deep learning. The hybrid methods KGAT and MKR based on knowledge graphs obtain better performance compared to other methods. MKR outperforms the matrix decomposition-based model, showing the power of neural network models in recommendation systems. Compared with KGAT, our model makes a new attempt to capture fine-grained information in graph data using GCN and to obtain the higher-order user-item interaction information using DNN. While KGNN-LS shows better performance, indicating the power of GNN in graph data processing, further suggesting that richer KG factual information can improve model performance. For Last.FM and Book-Crossing, we consider positive implicit feedback when the user's score for the item is greater than 0.5.

**TABLE 4.** Comparison of the recommended performance of the ablation models for the two datasets. The two methods lack graph structure data and DNN structure, respectively.

model	Book-Crossing		Last.FM	
	AUC	ACC	AUC	ACC
w/o DNN structure	0.736	0.699	0.787	0.742
w/o graph-structure data	0.733	0.696	0.782	0.739
ours	0.749	0.709	0.798	0.754



**FIGURE 4.** Trend of model performance with embedding dimension.

## 2) ABLATION ANALYSIS (RQ2)

To investigate whether both the graph feature learning module and the cross-learning module proposed in this method contribute to the performance of the model, we propose two simplified methods for comparison and set up the following comparison scenarios: 1) not using the user-item-entity graph structure in the recommendation module and using only the interaction graph of user-item information. 2) using only CIN for item preference extraction in the cross-learning module, and not using DNN. As can be seen from the table 4, removing the graph feature learning and cross-learning modules degrades the performance of the model. Specifically, in the top-K task, using the graph feature learning module outperforms the case when it is not used, indicating that a higher-order connected graph structure is effective in improving model performance and providing more feature descriptions. In the case without DNN, the model performance decreases compared to when it is used, as seen in the figure and table, indicating the importance of implicit higher-order features for inter-item connectivity.

## 3) HYPERPARAMETER SENSITIVITY (RQ3)

**Embedding Size:** in the neural network recommendation model, the embedding dimension as an important hyperparameter affects the model performance. Therefore, we verify the impact of different embedding dimensions on the performance of this model while ensuring that other parameters consistent with those mentioned in the previous section. The effects of varying embedding dimensions on the models AUC and ACC under the Book-Crossing dataset are shown in Figure 4. It can be seen from the figure that the performance of the model improves with the increase of embedding dimensions, and the best embedding dimension is 64. However, the performance of the model start to decrease at 64, which might be attributed to the overfitting and other problems brought by the increase of embedding dimensions, resulting in the



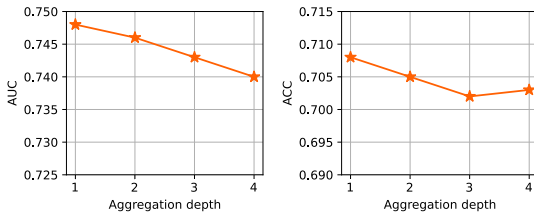


FIGURE 5. Trend of model performance with aggregation depth.

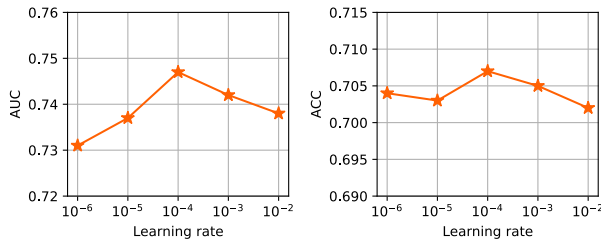


FIGURE 6. Trend of model performance with learning rate.

decrease of the generalization ability of the model. Therefore, we choose 64 as the embedding dimension of the model.

Aggregation depth: to investigate the effect of different aggregation depths on the user-item-entity graph, we studied the impact of depths on the model performance on the Book-Crossing and Last.FM datasets, taking values in the range  $\{1, 2, 3, 4\}$ . The empirical results show that the model works best on both datasets at  $L = 1$ , indicating that the aggregated information of the user-item-entity graph is expressed by first-order neighbors.

Learning rate: the learning rate as an important parameter affects the recommended performance, shown in Figure 6. From the figure, it is shown that the AUC and ACC performance of the model improves when the learning rate increases from  $10^{-4}$  to  $10^{-3}$ , and the performance decreases when it increases further. Therefore, a setting of  $10^{-3}$  is considered optimal.

## VI. CONCLUSION AND FUTURE WORK

In this work, we propose a novel recommendation framework that incorporates higher-order feature interactions and knowledge graph feature learning, which jointly acts on the knowledge graph structure and user-item interaction information in a unified neural network model to achieve recommendations. Specifically, this paper proposes a knowledge graph feature learning module based on a combination of convolutional networks and cross-learning, which exploits the higher-order structural similarity between entities and items in the knowledge graph to enhance the embedding learning of items. We design a cross-learning module that uses CIN and DNN to learn both explicit and implicit higher-order feature interactions, which is used to link the knowledge graph feature learning and recommendation modules to enhance the recommendation performance. The parameters of the framework are all optimized by means of union, and the experimental results show that the framework can capture

rich semantic information as well as the complex hidden relationships between users and recommended items well. The graph convolution method is also used to mine the higher-order connectivity of graph data to this recommendation framework. Finally, by conducting extensive experiments on two real datasets, the results demonstrate the effectiveness of our proposed model. In future work, we will work on top-K prediction, and deal with sparse data more effectively, using knowledge graph data, heterogeneous graphs and graph convolution to solve the cold start problem, and extending the framework proposed in this paper for article recommendation.

## REFERENCES

- [1] D. Das, L. Sahoo, and S. Datta, "A survey on recommendation system," *Int. J. Comput. Appl.*, vol. 160, no. 7, pp. 6–10, Feb. 2017.
- [2] P. Maes, "Agents that reduce work and information overload," *Commun. ACM*, vol. 37, no. 7, pp. 30–40, Jul. 1994.
- [3] C. Moon, P. Jones, and N. F. Samatova, "Learning entity type embeddings for knowledge graph completion," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 2215–2218.
- [4] W. Zhen, J. Zhang, J. Feng, and C. Zheng, "Knowledge graph embedding by translating on hyperplanes," in *Proc. Nat. Conf. Artif. Intell.*, 2014, pp. 1112–1119.
- [5] B. Hu, C. Shi, and W. Zhao, "Leveraging meta-path based context for top-N recommendation with a neural co-attention model," in *Proc. 24th ACM SIGKDD Int. Conf.*, 2018, pp. 1531–1540.
- [6] F. Zhang, N. J. Yuan, D. Lian, X. Xing, and W. Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proc. 22nd ACM SIGKDD Int. Conf.*, 2016, pp. 353–362.
- [7] A. Dadoun, R. Troncy, O. Ratier, and R. Petitti, "Location embeddings for next trip recommendation," in *Proc. Companion World Wide Web Conf. (WWW)*. New York, NY, USA: ACM, 2019, pp. 896–903.
- [8] H. Wang, F. Zhang, J. Wang, Z. Miao, and M. Guo, "RippleNet: Propagating user preferences on the knowledge graph for recommender systems," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 417–426.
- [9] 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 *Proc. World Wide Web Conf. (WWW)*. New York, NY, USA: ACM, 2019, pp. 151–161.
- [10] Y. Ye, X. Wang, J. Yao, K. Jia, J. Zhou, Y. Xiao, and H. Yang, "Bayes embedding (BEM): Refining representation by integrating knowledge graphs and behavior-specific networks," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manag. (CIKM)*. New York, NY, USA: ACM, 2019, pp. 679–688.
- [11] H. Wang, F. Zhang, X. Xie, and M. Guo, "DKN: Deep knowledge-aware network for news recommendation," in *Proc. Int. World Wide Web Conf. Steering Committee (WWW)*, Republic Canton Geneva, CHE, 2018, pp. 1835–1844.
- [12] H. Zhao, Q. Yao, J. Li, Y. Song, and D. L. Lee, "Meta-graph based recommendation fusion over heterogeneous information networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*. New York, NY, USA: ACM, 2017, pp. 635–644.
- [13] F. Dai, X. Gu, B. Li, J. Zhang, M. Qian, and W. Wang, "Meta-graph based attention-aware recommendation over heterogeneous information networks," in *Proc. Int. Conf. Comput. Sci.*, 2019, pp. 580–594.
- [14] D. Yang, Z. Guo, Z. Wang, J. Jiang, and W. Wei, "A knowledge-enhanced deep recommendation framework incorporating gan-based models," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*. IEEE, 2018, pp. 1368–1373.
- [15] H. Jin, W. X. Zhao, H. Dou, J. R. Wen, and E. Y. Chang, "Improving sequential recommendation with knowledge-enhanced memory networks," in *Proc. 41st Int. ACM SIGIR Conf.*, 2018, pp. 505–514.
- [16] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han, "Personalized entity recommendation: A heterogeneous information network approach," in *Proc. 7th ACM Int. Conf. Web Search Data Mining*, 2014, pp. 283–292.
- [17] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, "Explainable reasoning over knowledge graphs for recommendation," in *Proc. AAAI*, Jul. 2019, vol. 33, no. 1, pp. 5329–5336.

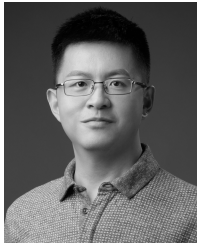
- [18] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proc. ACM Conf.*, 2016, pp. 233–240.
- [19] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery; Data Mining (KDD)*. New York, NY, USA: ACM, 2018, pp. 1059–1068.
- [20] H. T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, and M. Ispir, "Wide deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, 2016, pp. 7–10.
- [21] T. Bansal, D. Belanger, and A. McCallum, "Ask the GRU: Multi-task learning for deep text recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, 2016, pp. 107–114.
- [22] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, and M. Guo, "Multi-task feature learning for knowledge graph enhanced recommendation," in *Proc. World Wide Web Conf. (WWW)*. New York, NY, USA: ACM, 2019, pp. 2000–2010.
- [23] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "XDeepFM: Combining explicit and implicit feature interactions for recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1754–1763.
- [24] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2011, pp. 73–105.
- [25] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in *Proc. ACM Conf. Comput. Supported Cooperat. Work (CSCW)*. New York, NY, USA: ACM, 1994, pp. 175–186.
- [26] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*. Cham, Switzerland: Springer, 2007, pp. 325–341.
- [27] A. Vall, M. Dorfer, H. Eghbal-zadeh, M. Schedl, K. Burjorjee, and G. Widmer, "Feature-combination hybrid recommender systems for automated music playlist continuation," *User Model. User-Adapted Interact.*, vol. 29, no. 2, pp. 527–572, Apr. 2019.
- [28] P. Kouki, J. Schaffer, J. Pujara, J. O'Donovan, and L. Getoor, "Personalized explanations for hybrid recommender systems," in *Proc. 24th Int. Conf. Intell. User Interfaces*, Mar. 2019, pp. 370–390.
- [29] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 3307–3313.
- [30] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 355–364.
- [31] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, Montreal, QC, Canada, Jun. 2009.
- [32] X. Wang, X. He, Y. Cao, M. Liu, and T. S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *Proc. 25th ACM SIGKDD Int. Conf.*, 2019, pp. 950–958.
- [33] Q. Ai, V. Azizi, X. Chen, and Y. Zhang, "Learning heterogeneous knowledge base embeddings for explainable recommendation," *Algorithms*, vol. 11, no. 9, p. 137, Sep. 2018.
- [34] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Neural Inf. Process.*, 2013, pp. 1–9.
- [35] J. Bhatta, D. Shrestha, S. Nepal, S. Pandey, and S. Koirala, "Efficient estimation of Nepali word representations in vector space," *J. Innov. Eng. Educ.*, vol. 3, no. 1, pp. 71–77, Mar. 2020.
- [36] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 135–144.
- [37] J. Son and S. B. Kim, "Content-based filtering for recommendation systems using multiattribute networks," *Expert Syst. Appl.*, vol. 89, pp. 404–412, Dec. 2017.
- [38] M. Nilashi, O. Ibrahim, and K. Bagherifard, "A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques," *Expert Syst. Appl.*, vol. 92, pp. 507–520, Feb. 2018.
- [39] Z. Wang, G. Lin, H. Tan, Q. Chen, and X. Liu, "CKAN: Collaborative knowledge-aware attentive network for recommender systems," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 219–228.
- [40] H. Zhao, Q. Yao, Y. Song, J. Kwok, and D. L. Lee, "Side information fusion for recommender systems over heterogeneous information network," *ACM Trans. Knowl. Discovery Data*, vol. 15, no. 4, pp. 1–32, 2021.
- [41] X. Zhang, H. Liu, X. Chen, J. Zhong, and D. Wang, "A novel hybrid deep recommendation system to differentiate user's preference and item's attractiveness," *Inf. Sci.*, vol. 519, pp. 306–316, May 2020.
- [42] X. Wang, T. Huang, D. Wang, Y. Yuan, Z. Liu, X. He, and T.-S. Chua, "Learning intents behind interactions with knowledge graph for recommendation," in *Proc. Web Conf. (WWW)*. New York, NY, USA: ACM, 2021, pp. 878–887.
- [43] X. Wang, X. He, and T.-S. Chua, "Learning and reasoning on graph for recommendation," in *Proc. 13th Int. Conf. Web Search Data Mining*, Jan. 2020, pp. 890–893.
- [44] S. M. Kazemi, R. Goel, K. Jain, I. Kobyzev, A. Sethi, P. Forsyth, and P. Poupart, "Representation learning for dynamic graphs: A survey," *J. Mach. Learn. Res.*, vol. 21, no. 70, pp. 1–73, 2020.
- [45] I. Gasmi, F. Anguel, H. Seridi-Bouchelaghem, and N. Azizi, "Context-aware based evolutionary collaborative filtering algorithm," in *Modelling and Implementation of Complex Systems*, S. Chikhi, A. Amine, A. Chaoui, D. E. Saidouni, and M. K. Kholadi, Eds. Cham, Switzerland: Springer, 2021, pp. 217–232.
- [46] Z. Y. Khan, Z. Niu, A. S. Nyamawe, and I. U. Haq, "A deep hybrid model for recommendation by jointly leveraging ratings, reviews and metadata information," *Eng. Appl. Artif. Intell.*, vol. 97, Jan. 2021, Art. no. 104066.



**YANRU LIN** received the bachelor's degree in network engineering from the Shangluo College, in 2020. She is currently pursuing the master's degree in computer technology with Ningbo University. Her current research interests include knowledge graph and recommendation systems.



**SHIYU DU** received the Ph.D. degree from the Department of Chemistry, Purdue University, USA, in 2009. From July 2009 to December 2013, he served as a Postdoctoral Researcher and a Visiting Scientist at the Los Alamos National Laboratory, USA. During his stay in Los Alamos National Laboratory, he mainly undertook the theoretical calculation of key physical properties of nuclear fuel and nuclear material. He returned to China working full-time, in January 2014, and established a research team of "theoretical design and performance simulation of energy materials" at the Ningbo Institute of Materials Technology and Engineering, Chinese Academy of Sciences. He is currently the Doctoral Supervisor and a Professor with the Ningbo Institute of Materials Technology and Engineering, Chinese Academy of Sciences. He is also the Chief Scientist of the National Key Research and Development Program. He is currently working in the structural design and characterization of various energy and structural materials such as nuclear materials by the strategy of materials genome initiative and the multiscale computer simulation method. He has published more than 200 peer-reviewed research articles in *Nature Communications*, *Proceedings of the National Academy of Sciences*, *Journal of the American Chemical Society*, *Angewandte Chemie International Edition*, *ACS Nano*, *Nanoscale*, *The Journal of Physical Chemistry Letters*, and other international scientific research journals.



**YIMING ZHANG** received the master's degree (Hons.) in engineering and the Ph.D. degree from the Queen Mary University of London, in 2006 and 2010, respectively. From 2001 to 2003, he studied mechanical and electronic engineering at the Zhejiang University of Technology. He is currently an Associate Researcher. He is also mainly engaging in the application of material informatics, intelligent optimization algorithms, and system design methods in the research and devel-

opment of new ternary layered materials and extreme environmental energy materials. At present, he has published more than 40 papers in *Acta Materialia*, *Materialia*, *Philosophical Magazine*, *Journal of Power Sources*, and other magazines, with more than 400 citations, six patents, and five software copyrights. He is a member of the British Society of Minerals, Mining and Materials (IOM3).



**KAI DUAN** received the bachelor's degree in computer science and technology from Zhejiang Ocean University, in 2020. He is currently pursuing the master's degree in computer technology with Ningbo University. He is currently working on knowledge question answering systems and knowledge graph.



**QING HUANG** received the bachelor's degree in inorganic non-metallics and the master's degree in materials science from Tianjin University, in 1995 and 2002, respectively, and the Ph.D. degree in materials science from the Shanghai Institute of Ceramics, Chinese Academy of Sciences (CAS). From 2005 to 2010, he carried out postdoctoral research at the National Institute for Materials Science, Japan, and the University of California Davis, USA. In 2010, he joined the Ningbo Institute of Materials Technology and Engineering, and focused on the develop-

ments and applications of energy materials under extreme environments including the creation of non-van der Waals layered materials (MAX phases and MXenes) and silicon carbide composites for high-safety energy systems. He is currently a Professor with the Ningbo Institute of Materials Technology and Engineering, CAS.



**PENG AN** received the bachelor's and Ph.D. degrees from the Department of Engineering Physics, Tsinghua University, China. From August 2015 to March 2016, he served as a Visiting Scientist at Heidelberg University, Germany. He is currently a Professor with the Ningbo University of Technology and the Supervisor of Postgraduate. He has published more than 20 papers in *Information Processing and Management*, *Complexity*, and other magazines, with more than 180 citations, and eight patents and two software copyrights. His current research inter-

ests include pattern recognition and machine learning and wireless sensor networks.

...