Course Recommendation Method Based on Dual-End Collaborative Information of Knowledge Graph

Yizhi Zhang[1,*], Xinyuan Ji[1], Danni Yan[1], Liming Xu[1], Bello Ahmad Muhammad[2]
[1]School of Artificial Intelligence and Computer Science, Shaanxi Normal University, Xi'an, Shaanxi, 710062, China
[2]School of Computer Science and Technology, Henan Institute of Science and Technology, Xinxiang, Henan 453003, China
[*]Corresponding author

Knowledge graphs contain rich semantic information, and their integration into recommendation systems has significantly alleviated challenges such as data sparsity and the cold-start problem. However, existing knowledge graph-based recommendation methods primarily focus on optimizing the item side of the graph, often neglecting the user side. This limitation leads to insufficient utilization of explicit collaborative information derived from user–item interactions, resulting in embedding representations that fail to effectively capture the latent semantics of both users and items. To address this issue, we propose a novel approach, Course Recommendation Method Based on Dual-End Collaborative Information of Knowledge Graph (DCIKG-Rec). The proposed method simultaneously models both users and items, enabling the integration of collaborative information and knowledge associations through heterogeneous propagation techniques to enhance representation learning. Furthermore, DCIKG-Rec employs a knowledge-aware attention mechanism to evaluate the importance of neighbors at each layer for different entities, and a bias-based attention mechanism to preserve collaborative information during multi-layer propagation. Finally, the learned representations of users and items are utilized to predict the probability of user–item interactions. Extensive experiments conducted on a real-world dataset demonstrate that DCIKG-Rec achieves an AUC of 0.8964 and an F1 score of 0.7952 in click-through rate prediction. In addition, its Top-K recommendation performance shows superior recall compared with several state-of-the-art baseline models.

Index Terms—**Recommendation System, Collaborative Filtering, Heterogeneous Propagation, Knowledge Graph.**

I. INTRODUCTION

In the era of booming digital learning, online learning resources are becoming increasingly abundant [1]. How to precisely recommend courses that meet learners' personalized needs from the vast amount of learning resources has already become a hot issue that attracts common attention in both the educational technology field and the computer science field [2]. With the rise and development of Knowledge Graph (KG) technology, integrating it into recommendation systems has provided new ideas and approaches for solving the problem of learning resource recommendation.

Knowledge graphs integrate a large number of entities and the relationships between them in a structured manner [3]. They can unearth the deep semantic connections among learning resources (such as courses), learners, and related attributes, thereby helping recommendation systems better understand user preferences and resource characteristics.

In the early days, most of the recommendation methods based on knowledge graphs enhanced item representations through knowledge graph embedding [4]. Examples include TransE [5], TransH [6], TransR [7], TransD [8], and TranSparse [9], etc. This reduced the high dimensionality and heterogeneity of knowledge graphs, but ignored the connectivity among entities [10]. Subsequently, many scholars used meta-paths to enrich the interactions between users and items, improving the interpretability of recommendation results. However, meta-paths need to be manually designed [11]. Although these methods have improved the accuracy of recommendations and the interpretability of recommendation results [12], they still have problems such as poor generalization ability and unstable performance [13].

Graph convolutional network (GCN) has excellent information extraction and representation capabilities on graph-structured data and has gradually become a research hotspot in recommendation systems [14]. Its powerful information propagation ability can mine high-order semantic relationships between entities and enhance the expression ability of the model, such as KGCN [15]. However, through in-depth research and practice, it has been found that there are some limitations in existing knowledge graph-based recommendation systems. On the one hand, most of such systems focus on optimizing the item (learning resource, such as a course) representation side, devoting their main efforts to mining the characteristics of the course itself and optimizing the course representation based on knowledge graph relationships, while relatively ignoring the refined modeling of the user side. This unbalanced modeling approach makes it difficult for the system to comprehensively capture the collaborative relationship between users and learning resources, resulting in ignoring many explicit collaborative information that should be emphasized during the recommendation process, and further affecting the degree of conformity of the recommendation results to the real needs of users. On the other hand, some improved models such as Knowledge Graph Attention Network (KGAT) [16] recognized the necessity of dual-side modeling for both the user and item domains, and their work marked an important step forward. However, a key assumption in the modeling process of KGAT is that the interaction items in the User-Item-Knowledge Graph (UIG) and the associated entities in the Knowledge Graph (KG) are treated as isomorphic nodes. In real learning scenarios, however, there are fundamental differences between

Manuscript received December 26, 2024; revised November 28, 2025.
Corresponding author: Yizhi Zhang (yzzhang@snnu.edu.cn).

them in terms of semantic levels, functional attributes, and the roles they play within the overall learning ecosystem. Furthermore, as the propagation layers in the recommendation system deepen, the user and item representations extracted from different layers encapsulate information at varying levels of abstraction. These representations reflect user interests and resource characteristics in diverse forms. If such differences are not properly leveraged, the recommendation system will inevitably lose a significant amount of valuable information, thereby undermining both the accuracy and the comprehensiveness of the recommendations.

To more effectively leverage knowledge graphs to enhance the quality of course recommendations, this study proposes a course recommendation method based on collaborative knowledge dual-end embedding. This method focuses on modeling both the user side and the item side, fully considering the characteristics of each side and the complex and subtle collaborative relationships between them. This ensures that no information is lost due to an overemphasis on one side during the recommendation process. Additionally, an innovative bias attention mechanism is introduced, which effectively integrates information based on its importance at different levels and its relevance to the recommendation goal. This allows the user and project representations used for the final recommendation decision to reflect key information as fully as possible. As a result, the accuracy of course recommendations is significantly improved, providing learners with more personalized course suggestions and driving the continuous development of online learning resource recommendation systems. The contributions of the paper are as follows:

- We propose a novel course recommendation method, DCIKG-Rec, which differs from existing knowledge graph-based recommendation approaches. The model jointly considers both the user side and the item side, effectively integrating collaborative information with knowledge associations.
- We design a knowledge-aware attention mechanism to evaluate the importance of neighbors at each propagation layer for different entities, and further introduce an innovative bias-based attention mechanism to preserve collaborative information during multi-layer propagation.
- The effectiveness of the proposed DCIKG-Rec is validated through extensive experiments on real-world datasets, demonstrating its superior performance compared with state-of-the-art baselines.

The remainder of this paper is organized as follows. Section II reviews the current research progress in related fields. Section III defines the key concepts involved in this study and discusses the new challenges that need to be addressed. Section IV provides a detailed description of the proposed DCIKG-Rec model. Section V presents experimental evaluations of the proposed method against several baseline approaches on real-world datasets, followed by an in-depth analysis and comparison of the results. Finally, Section VI concludes the paper and outlines directions for future work.

II. Related Works

In this section, we introduce recommendation methods based on knowledge graphs. With the deep integration of recommendation systems and knowledge graph technology, such methods have continuously evolved and gradually formed several mainstream categories, including embedding-based methods, path-based methods, and neural network-based methods.

A. Embedding Based Methods

Represented by the Trans series of algorithms [4] [5], these methods map the entities and relations of knowledge graphs into a low-dimensional vector space and learn embedding representations through specific models and loss functions. For instance, TransE optimizes by treating relations as translation vectors from the head entity to the tail entity. While such approaches demonstrate clear advantages in knowledge graph completion and link prediction, they perform poorly in recommendation tasks. This is because recommendation tasks are inherently complex and highly sensitive to user personalization, making it difficult to accurately capture the associations between users and items solely through embedding representations. When confronted with diverse user interests and intricate item relationships, the information obtained is insufficient to support high-quality recommendations.

B. Path Based Methods

These methods focus on the connections among entities in the knowledge graph and make recommendations by utilizing the entity connection patterns. For example, Hu et al. [17] combined the convolutional neural network with meta-paths to obtain the vector representations of users and items, and mined the connection path information by manually designing meta-paths. However, it relies on manually designed meta-paths. When the application scenarios change, it is necessary to redesign them, which is costly and lacks scalability, making it difficult to be widely adapted to different recommendation scenarios.

C. Graph Neural Network Based Methods

An increasing number of studies have demonstrated that graph neural network (GNN) methods are highly effective in addressing the cold-start problem in recommendation, with the application of Graph Convolutional Networks (GCN) being particularly extensive. Wang et al. [15] proposed Knowledge Graph Convolutional Networks (KGCN), which employed GCN to aggregate entity representations in a biased manner and leveraged high-order links in the Knowledge Graph (KG) to explore users' latent interests, thereby alleviating the issue of sparse user interaction data. However, KGCN overlooked the modeling of user-side information and explicit collaborative signals. Lei et al. [18] introduced a knowledge graph-enhanced neural collaborative recommendation framework. Building on GCN, this framework mined high-order semantics from the KG and modeled entity context information by stacking multiple convolutional layers. An attention network was then designed to learn the weight distribution of user interaction items and candidate items. Through weighted aggregation, user representations were obtained. Finally, the representations of items and users were fed into the Neural Collaborative Filtering (NCF) model to capture their interaction features and predict users' latent preferences for items. Wang et al. [16] proposed the Knowledge Graph Attention Network (KGAT), which recursively propagated high-order neighborhood information of nodes by stacking propagation layers. KGAT employed Graph Attention Networks (GAT) to learn neighbor weights during propagation, thereby exploring the importance of different high-order connections. However, KGAT failed to distinguish between user nodes and other entity nodes, which was unreasonable and limited its ability to handle new users effectively.

Based on the limitations of existing methods, this research proposes a course recommendation method based on dual-end collaborative information of the knowledge graph (DCIKG-Rec). This method conducts dual-end modeling for the user end and the item end, aiming to fully consider the respective characteristics of users and items as well as the collaborative relationship between them. Meanwhile, it effectively integrates information at different levels through the biased attention mechanism, thereby improving the accuracy of course recommendations.

III. Problem Statement

In this section, we first formulate the course-recommended problem and mentioning some key definitions and notations used.

Definition 1 Interactive Matrix Y: We have a set of M students $U = \{u_1, u_2, ..., u_M\}$ and a set of N courses $C = \{c_1, c_2, ..., c_N\}$. Based on students' viewing history records, we can obtain the interaction matrix $Y \in R^{|M|*|N|}$, where $y_{uc} = 1$ indicates that student u has interacted with course c, and $y_{uc} = 0$ otherwise. Please note that $y_{uc} = 0$ does not mean that the student dislikes or does not need course c, maybe the student likes course c but just accidentally overlooked it or failed to find it.

Definition 2 Knowledge Graph G: The knowledge graph is denoted as $G = \{(h, r, t)|h, t \in E, r \in R\}$, where h, r, and t are the head entity, relation, and tail entity, respectively. Moreover E and R are the set of entities and the set of relations in G, respectively. For example, the triple (*Data Structures*, *Discipline*, *Computer Science*) indicates that the course Data Structures belongs to the discipline of Computer Science.

We define the set $A = \{(c, e)|c \in C, e \in E$, where each pair ($c$, e) indicates that course c can be aligned with entity e in the knowledge graph, thereby clarifying the correspondence between courses and entities.

Definition 3 Course Recommendation: Given the user-item interaction matrix Y and the knowledge graph G, the current task is to predict the probability that a student u will interact with the courses c that he has not previously engaged with. Specifically, we aim to learn the prediction function $\hat{y}_{uc} = \mathcal{F}(u, c|Y, G, \Theta)$, where $\hat{y}_{uc}$ represents the probability that student u clicks on course c, and Θ denotes the training parameters of the prediction function $\mathcal{F}$.

For better readability and ease of reference, the key notations and their definitions used throughout this paper are summarized in Table I.

IV. The Proposed Method

The DCIKG-Rec is designed as an end-to-end model that performs joint modeling from both the user side and the item side. The overall framework of the model is illustrated in Fig. 1. It consists of three main components:1) The heterogeneous propagation layer: This layer effectively propagates collaborative signals by leveraging students' interaction histories with courses as well as knowledge associations within the knowledge graph. 2) Knowledge-aware Attention Embedding Layer: To differentiate the contributions of various neighbors, the weights of entity neighbors are learned through a knowledge-aware attention mechanism. The final representation of each entity is then composed of its weighted neighbors. 3) The prediction layer: The embedding representations obtained from different propagation layers are aggregated using a biased attention mechanism to generate the final representations of students and courses. The prediction probability is subsequently computed by taking the dot product of these two representations.

A. Heterogenous Propagation Layer

The heterogeneous propagation layer consists of two components: collaborative propagation and knowledge graph propagation. Collaborative propagation utilizes the embedding representations of students and courses to transmit essential collaborative signals, whereas knowledge graph propagation disseminates knowledge associations along the edges of the knowledge graph, thereby enhancing the representations of both courses and students.

1) Collaborative Propagation

The courses that students have studied in the past can reflect their preference tendencies to some extent. Therefore, the courses that students have studied before are used to represent the students. After the alignment and transformation between the set of courses that student u has studied in history and the entities in the knowledge graph, it will be taken as the initial seed set for student u to carry out information propagation in the knowledge graph.

The definition of the set of courses that student u has studied in history is as follows:

$$U_c^0 = \{c \mid y_{uc} = 1\} \tag{1}$$

Align the set of courses that student u has studied in history with the entities in the knowledge graph to obtain the initial entity set of student u. The initial entity set of student u is defined as follows:

$$\varepsilon_u^0 = \{e \mid (c, e) \in A \text{ and } c \in U_c^0\} \tag{2}$$

Similarly, following the principle of collaborative filtering, students who have taken the same course are likely to share similar learning preferences. This information can be utilized

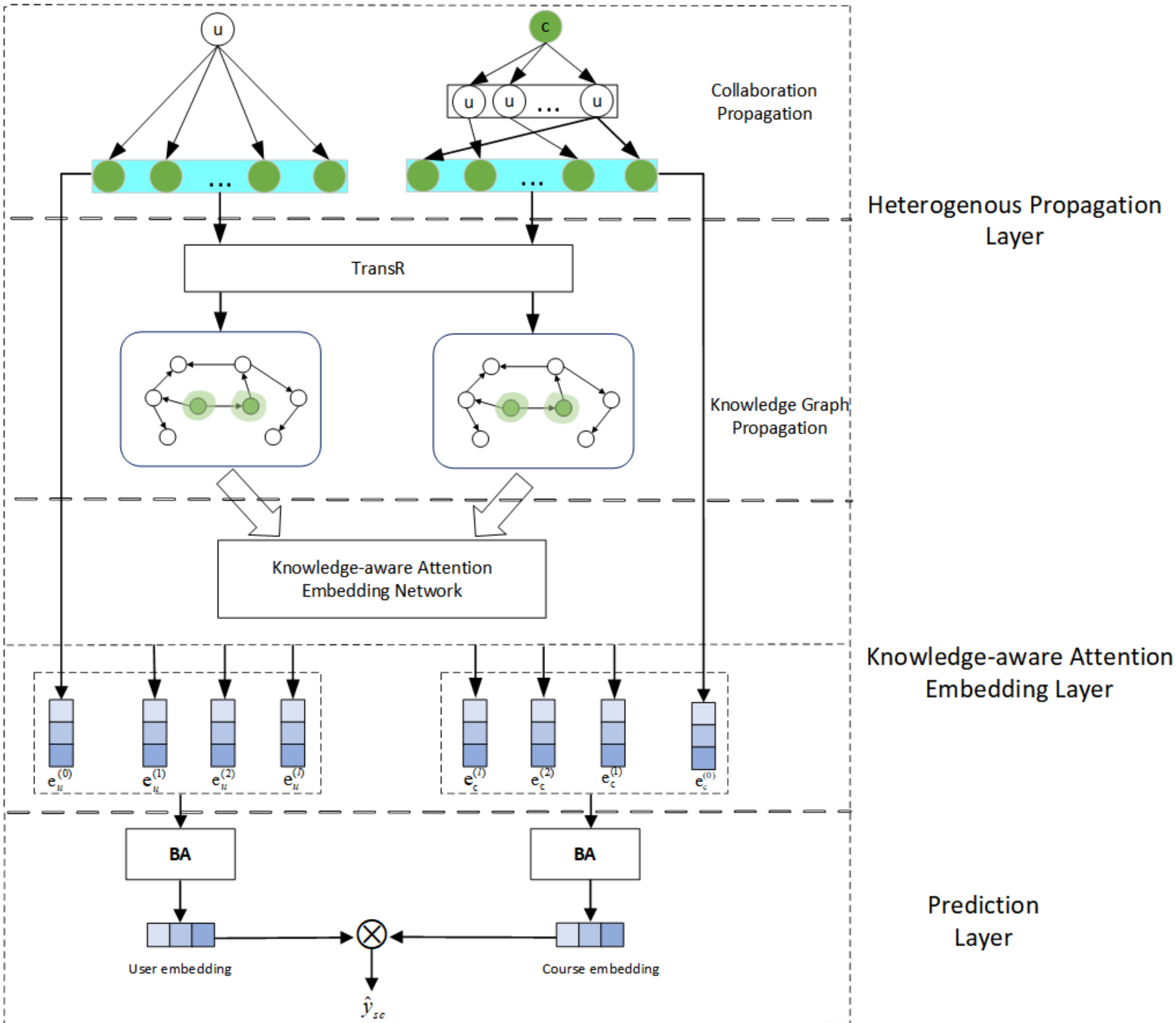


Fig. 1. The DCIKG-Rec Method

to refine the representation of a course. Specifically, other courses that these students have previously studied are incorporated to represent the target course. The courses studied by the same student are referred to as collaborative neighbors. Accordingly, the collaborative neighbor set of course C is defined as follows:

$$C_c = \{c_s \mid u \in \{u \mid y_{uc} = 1\} \text{ and } y_{uc_u} = 1\} \tag{3}$$

Here, c_u represents the set of courses, excluding c, that student u has previously studied. Align the collaborative neighbors with the entities in the knowledge graph to obtain the initial entity set of course c. The initial entity set of course c is defined as follows:

$$\varepsilon_c^0 = \{e \mid (c_u, e) \in A \text{ and } c_u \in C_c\} \tag{4}$$

After collaborative propagation, the embedding sequences of students and courses are processed into combinations of entity sequences in the knowledge graph. To distinguish the importance of different relationships of the same entity in the knowledge graph, we use the TransR method to handle the complex relationships between entities. More specifically, if there is a triple (h, r, t) in the graph, optimize the translation to learn to embed each entity and relationship. The optimization translation principle is as follows:

$$e_r^h + e_r \approx e_r^t \tag{5}$$

where, e^h, $e^t \in R^d$ and $e_r \in R^k$ denote the embedding representations of h, r, and t, respectively. Furthermore, e_r^h and e_r^t represent the projection vectors of h and t in the relation space r. Therefore, for a given triple (h, r, t), its scalability score (or energy score) is defined as:

$$g(h, r, t) = \|W_r e_h + e_r - W_r e_t\|_2^2 \tag{6}$$

where, $W_r \in R^{k \times d}$ is the transformation matrix of relation r, which projects entities in the d-dimensional entity space into the k-dimensional relation space. The lower the score, the more similar.

The training of TransR takes into account the relative order between valid triples and corrupted triples and measures their discrimination through pairwise ranking loss:

$$L_{KG} = \sum_{(h,r,t,t') \in \Gamma} -\ln \sigma(g(h, r, t') - g(h, r, t)) \tag{7}$$

where $\Gamma = \{(h, r, t, t') \mid (h, r, t) \in G,\ (h, r, t') \notin G\}$ is a

TABLE I
SUMMARY OF NOTATIONS

Symbol	Definition
U	Set of users (students)
C	Set of courses
$Y \in R^{\|M\| \times \|N\|}$	Student–course interaction matrix
G	Knowledge graph
A	Alignment set between courses and KG entities
E	Entity set in the knowledge graph
$\hat{y}_{sc}$	Predicted interaction probability
R	Relation set in the knowledge graph
(h, r, t)	Knowledge graph triple
e_h	Embedding of head entity h
e_t	Embedding of tail entity t
e_r	Embedding of relation r
W_r	Relation-specific projection matrix
e_r^h	Projection of entity h in relation r space
s_c^0	Courses previously taken by student s
ε_s^0	Initial entity set of student s
ε_c^0	Initial entity set of course c
ε_o^l	Entity set at layer l $(o \in \{s, c\})$
U_o^l	Triple set at layer l
e_o^l	Layer-wise embedding
e_c^{origin}	Original entity-based embedding of course
s_i	Unnormalized attention score
a_i	Normalized attention weight
W_1, W_2	Trainable weight matrices
b_1, b_2	Bias vectors
$\sigma(\cdot)$	Sigmoid activation function
ReLU$(\cdot)$	Rectified Linear Unit
$\|\|$	Concatenation operator
$BA(\cdot)$	Bias-based attention mechanism
L_{KG}	KG embedding loss
L_{CF}	Collaborative filtering loss
λ	Regularization coefficient
Θ	Trainable parameter set

corrupted triple constructed by randomly replacing one entity in a valid triple; $\sigma(\cdot)$ is the sigmoid function.

2) Knowledge Graph Propagation

Based on the above circumstances, under the premise of ensuring that it will not have an adverse impact on the propagation of the entity sequences of students and courses along the knowledge graph to expand their potential vector representations, student u and course c can be defined in the following recursive manner:

$$\varepsilon_o^l = \{W_r t \mid (h, r, t) \in G, h \in \varepsilon_o^{l-1}\}, \text{and } l = 1, 2, \ldots, L \quad (8)$$

Here, l represents the number of times the initial entity set is propagated. The symbol o is used to denote the placeholder for student u or course c. The definitions of student u and course c contained in the set of the l-th layer are as follows:

$$U_o^l = \{(W_r h, r, W_r t) \mid (h, r, t) \in G, h \in \varepsilon_o^{l-1}\}, \\ l = 1, 2, \ldots, L \quad (9)$$

The initial entity set obtained through collaborative propagation spreads outward in the knowledge graph like ripples in water, gradually expanding layer by layer from near to far. Through the knowledge-aware deep propagation process, high-order interaction information between students and courses can be effectively captured at the knowledge level, thereby enhancing the model's ability to represent users and items with latent vectors.

B. Knowledge-aware Attention Embedding Layer

When entities propagate through the knowledge graph, different relationships within the formed triples exert varying influence weights on the tail entities. For instance, although the courses "*Computer Network*" and "*Appreciation of Crosstalk Art*" are both taught by the same instructor and thus share the same teacher attribute, they differ completely in the discipline attribute. Motivated by this observation, we propose a knowledge-aware attention embedding method that distinguishes the attention weights of different head entities and relationships for tail entities during the propagation process in the knowledge graph. This approach is similar to the KGAT model; however, to more effectively preserve the initial collaborative information, the aggregation strategy adopted here first aggregates relevant tail entities according to the attention mechanism within the ripples of the l-th layer, and then aggregates the tail entities across layers to obtain the representation of the current layer.

Fig. 2 illustrates the workflow of the Knowledge-aware Attention Embedding Layer. First, the head entity embedding and relation embedding are concatenated and fed into a neural network to compute the attention weights of the head entity in the relation space with respect to candidate tail entities. These attention weights are then applied to the tail entity embeddings through element-wise multiplication to obtain weighted tail vectors. Finally, all weighted tail vectors are aggregated by summation to generate the output embedding of layer e_o^l.

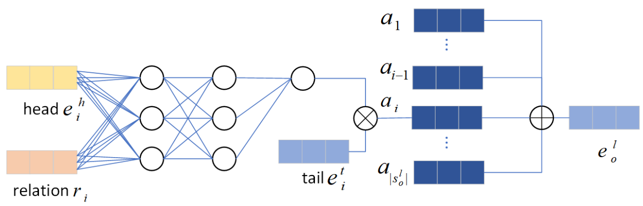


Fig. 2. Knowledge-aware Attention Embedding Network

Assume that the i-th triple in the l-th layer is (h, r, t), then the weighted embedding representation a_i of the tail entity of this triple can be expressed as:

$$a_i = \pi(e_i^{W_r h} \cdot r_i) \cdot e_i^{W_r t} \tag{10}$$

In the formula, $e_i^{W_r h}$ is the embedding representation of the head entity in the relationship r space, $e_i^{W_r t}$ is the embedding representation of the tail entity in the relationship r space. $\pi(e_i^{W_r h}, r_i)$ represents the weight of the head entity on the tail entity in the relationship r space. Different relationship spaces correspond to different embedding representations of the head and tail entities. The specific process is described as follows:

$$\begin{aligned} Z_0 &= \mathrm{ReLU}(W_0(e_i^{W_r h} \| r_i) + b_0) \\ \pi(e_i^{W_r h}, r_i) &= \sigma(W_2 \mathrm{ReLU}(W_1 Z_0 + b_1) + b_2) \end{aligned} \tag{11}$$

During this process, we choose $ReLU$ [19] to serve as the non-linear activation function, and the final activation step is achieved by using the $Sigmoid$ [20] function. The symbol "$\|$" here represents the concatenation operation, whose function is to sequentially concatenate the relevant elements or vectors. Both the matrix and the vector are parameters that can be trained and optimized. Among them, W is the weight matrix, and b is the bias vector. The different subscripts they carry clearly indicate the different layers to which they belong.

After the weight scores corresponding to different neighbors are successfully obtained, the set of triples of students or courses in the l-th layer can be represented as:

$$e_o^l = \sum_{i=1}^{|S_o^l|} a_i^{(o)}, \quad l = 1, 2, \ldots, L \tag{12}$$

Among them, the subscript and superscript o are unified placeholder identifiers for the symbols u (student) or c (course). And $|s_o^l|$ represents the number of triples in the set s_o^l. It should be noted that, since the entities in the initial entity set are just like the seeds in the process of knowledge-based propagation and are very close to the original representations, there is an extremely close correlation between the initial entity set and the original students as well as courses. Based on this, we define the initial representation $e_o^{(0)}$ for both students and courses. When the object is a student, it is denoted as $e_s^{(0)}$, when the object is a course, it is denoted as $e_c^{(0)}$. The term $|\varepsilon_o^0|$ represents the size of the set of entities related to the object.

$$e_o^l = \frac{\sum_{e \in \varepsilon_o^0} e}{|\varepsilon_o^0|} \tag{13}$$

Particularly, since course nodes usually have explicit static semantic neighbors in the knowledge graph (such as textbooks and subject labels), the course c has entities related to its original representation, while the student u does not. The original related entities are composed of the entities corresponding to it in the set of course entities:

$$e_c^{\text{origin}} = \frac{\sum_{e \in \{e | (c,e) \in A\}} e}{|\{e \mid (c, e) \in A\}|} \tag{14}$$

After completing the knowledge-aware attention embedding operation, we use formulas to present the sets of knowledge-based attention-weighted representations for students u and courses c. The specific forms are shown as follows.

$$\begin{aligned} T_u &= \{e_u^0, e_u^1, \ldots, e_u^L\}, \\ T_c &= \{e_c^{\text{origin}}, e_c^0, e_c^1, \ldots, e_c^L\} \end{aligned} \tag{15}$$

C. Prediction Layer

After going through the knowledge-based attention embedding layer, the final representations of student s and course c are as follows:

$$e_u = \sum_{l=0}^{L} a_l \, e_s^l, \quad e_c = \sum_{l=0}^{L} a_l \, e_c^l \tag{16}$$

where a_l is the attention weight of the l-th layer embedding, which measures the importance of this layer during the aggregation phase. The weights of different layers are obtained by computing and normalizing them through the biased attention mechanism, thereby reducing the dilution of collaborative information caused by multi-layer propagation and improving the accuracy of recommendation results. Among them, the calculation formula of the Bias-based Attention mechanism is as follows:

$$BA(h, r, t) = W_1^T \cdot [W_2(h_r + e_r) \otimes t_r] \tag{17}$$

where W_1^T is a trainable attention parameter vector, and $W_2 \in R^{n \times n}$ is a trainable student-course interaction matrix. h_r denotes the contextual embedding of a student under relation r, e_r denotes the embedding of the recommended course set, and t_r denotes the embedding of the student's true preferred courses. The term $h_r + e_r$ represents the set of recommended courses for the student, while t_r corresponds to the courses in which the student is originally interested. An inner product operation is then performed. Through these parameters and operations, an attention score is obtained to measure the deviation between the recommended courses and the courses that the student truly prefers, thereby adjusting the weights of embeddings across different layers.

The deviation-based attention mechanism function is normalized through the softmax function, The attention weights are obtained through softmax normalization:

$$\alpha_l = \frac{\exp(BA_l)}{\sum_{l=0}^{L} \exp(BA_{l'})} \tag{18}$$

where a_l denotes the normalized weight of the l-th layer. A higher score indicates a smaller deviation between the recommended courses and the student's true interests, meaning that the embedding of this layer contributes more to the final aggregation.

After obtaining the final representations of students and courses, the probability of a student selecting a course is predicted through an inner product:

$$\hat{y}_{uc} = e_u^T \cdot e_c \tag{19}$$

This score reflects the similarity between the two in the latent semantic space. A larger value indicates a higher semantic matching degree between the student and the course, as well as a stronger interest of the student in that course.

D. Loss Function

The loss function of the DCIKG-Rec model is composed of two parts. The first part is the loss function of the prediction result, and it is expressed as:

$$L_{CF} = \sum_{s \in U} \Big(\sum_{(s,c) \in P^+} F(y_{sc}, \hat{y}_{sc}) - \sum_{(s,c) \in P^-} F(y_{sc}, \hat{y}_{sc}) \Big) \tag{20}$$

where F is the cross-entropy loss, P^+ represents the set of positive student-course pairs, while P^- is the opposite. The second part of the loss function is the pairwise ranking loss between entities in the knowledge graph. The final loss function is:

$$L_{DCIKG-Rec} = L_{KG} + L_{CF} + \lambda \parallel \theta \parallel_2^2 \tag{21}$$

where $\theta = \{E, R, W_r, W_i, b_i, \forall i \in (0, 1, 2, a)\}$ is the set of model parameters. E and R are embedding tables used for all entities and relations, and $\parallel \theta \parallel_2^2$ is the L_2-regularization term with parameter.

V. Experiments

To prove the effectiveness and accuracy of our proposed DCIKG-Rec model, we conducted the following experiments. This section will focus on the baseline methods used for comparison, the evaluation metrics for performance assessment, the performance of the proposed DCIKG-Rec on real datasets, and the analysis of important parameters in the model.

A. Datasets

The datasets used in this study are from the XuetangX [21] MOOC platform, as detailed in Table II. We focused on the course interaction data from June 1, 2018, to December 10, 2018.

TABLE II
MOOCCUBE DATASET DESCRIPTION

	name	**number**
Student Course Interaction Statistics	users	6461
	courses	329
	interactions	51124
	entities	40428
	relations	6
	KG triple	34061

B. Evaluation Metrics

To comprehensively evaluate the performance of the trained model, the experiment considers two recommendation scenarios.

Click-Through Rate (CTR) Prediction: This task estimates the probability of each interaction in the test set. The evaluation employs the AUC and F1 metrics. Specifically, AUC measures the likelihood that a user's preferred items are ranked ahead of non-preferred items, thereby reflecting performance across recommendation lists of varying lengths. However, when the area under the ROC curve is identical, AUC alone is insufficient to fully assess the quality of the algorithm. Therefore, the experiment complements it with the F1 score. The corresponding evaluation metric formulas are as follows:

$$\begin{aligned} Recall &= \frac{TP}{TP + FN} \\ Precision &= \frac{TP}{TP + FP} \\ F1 &= 2 \times \frac{Precision \times Recall}{Precision + Recall} \end{aligned} \tag{22}$$

Top-K Recommendation: The trained model is utilized to select the top K items with the highest predicted probabilities for each user in the test set. The items that a user has actually interacted with in the test set are regarded as the ground-truth set. The model ranks the preference probabilities of the un-interacted items, and the top K items with the largest probability values are selected as the candidate set to recommend to the user. The evaluation metric Recall@K is then employed to measure the proportion of relevant items successfully retrieved within the top K recommendations. In addition, we will also analyze the effect of changing the maximum depth of the knowledge-aware propagation layer L on the performance of DCIKG-Rec.

C. Parameter Analysis

We partitioned the dataset into training, validation, and testing sets with a ratio of 6:2:2. In terms of optimization, the Adam optimizer [22] was employed to optimize all models, with the batch size fixed at 2048. For parameter initialization, we adopted the default Xavier initializer [23]. We implemented the proposed DCIKG-Rec model using the PyTorch framework and applied a grid search strategy to determine the optimal hyperparameters. Specifically, the learning rate was tuned within $\{10^{-3}, 5\times10^{-4}, 10^{-4}, 5\times10^{-5}\}$, the L_2 regularization coefficient was adjusted among $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$, and the embedding size was searched within $\{8, 16, 32, 64, 128, 256\}$. Since the optimal size of the triple sets for users and items might vary, we searched for the set sizes within $\{4, 8, 32, 64\}$. The experimental equipment configuration is shown in Table III. Additionally, our experiments identified the optimal configuration of the attention network: a two-layer structure with the hidden layer dimension set equal to the embedding size. The hyperparameters of all comparison methods were tuned either through empirical exploration or by adopting the optimal settings reported in the original literature.

TABLE III
EXPERIMENTAL ENVIRONMENT CONFIGURATION

Component	Specification
CPU	AMD Ryzen 5 4600H with Radeon Graphics
GPU	NVIDIA GeForce GTX 1650
RAM	24GB DDR4
Operating System	Windows 10 Professional 64-bit
Development Environment	Anaconda3
Programming Language	Python 3.11.4
Deep Learning Framework	PyTorch 2.5.1+cu121

D. Baseline Methods

The baseline methods that we want to compare is as follows:

1) *CKE* [24] is a typical embedding-based model that combines the CF module with the structural, textual, and visual knowledge embeddings of items within a unified Bayesian framework.

2) *RippleNet* [25] is a state-of-the-art propagation-based model that employs a memory-like network and propagates users' latent preferences within the KG to enrich user representations.

3) *KGCN* [15] is another state-of-the-art propagation-based model that extends the non-spectral GCN method to the knowledge graph by selectively and biasedly aggregating neighborhood information, enabling it to learn the structural and semantic information of the KG as well as users' personalization and latent interests.

4) *KGNN-LS* [26] is yet another state-of-the-art propagation-based model that converts the heterogeneous KG into a user-specific weighted graph and computes personalized item embeddings in the graph neural network using label smoothing regularization.

5) *KGAT* [16] is also a typical model based on propagation. It combines the user-item graph and the knowledge graph to enhance the integration of collaborative information and knowledge, and uses the attention mechanism to distinguish the importance of different neighbors of entities.

E. Experimental Results and Comparative Analysis

The specific experimental results and analysis of the comparison between the DCIKG-Rec model and other baseline models on the Mooccube dataset are as follows:

1) Top-K Recommendation

The experimental results of the proposed algorithm, together with those of the comparison algorithms, in the Top-20 and Top-50 recommendation scenarios are presented in Table IV. Here, R@20 and R@50 denote Recall@20 and Recall@50, respectively.

TABLE IV
EXPERIMENTAL RESULTS OF THE TOP-20 AND TOP-50 RECOMMENDATIONS.

Models	**R@20**	**R@50**
CKE	0.2282	0.3334
RippleNet	0.1867	0.2898
KGCN	0.2271	0.3034
KGCN-LS	0.2461	0.3127
KGAT	*0.2336	*0.3304
DCIKG-Rec	**0.2595**	**0.3518**

According to the comparison results in Table IV, the recall rate of DCIKG-Rec on the Mooccube dataset is superior to that of the baseline models. In general, the Recall value increases as the value of K grows. However, because DCIKG-Rec effectively integrates additional auxiliary information and more accurately captures knowledge representations as K increases, its advantage becomes more pronounced when K is relatively large.

2) Click-Through Rate Prediction

The AUC and F1 results on each dataset in the CTR prediction are shown in Table V.

TABLE V
THE AUC AND F1 RESULTS ON EACH DATASET IN THE CTR PREDICTION.

Models	**AUC**	**F1**
CKE	0.7543	0.7012
RippleNet	0.7934	0.7246
KGCN	0.8797	0.7640
KGCN-LS	0.8724	0.7632
KGAT	*0.8846	*0.7767
DCIKG-Rec	**0.8964**	**0.7952**

In the Click-Through Rate (CTR) prediction scenario, according to the comparison results in Table III, both the AUC and F1 scores of DCIKG-Rec on the Mooccube dataset are higher than those of several state-of-the-art baselines. Compared with KGCN and KGCN-LS, this demonstrates the necessity of encoding information on both the user side and the item side. In comparison with KGAT, it highlights that leveraging explicit collaborative information between users and courses helps to improve recommendation performance. Moreover, the knowledge graph-based methods, namely RippleNet and KGCN, outperform traditional collaborative filtering approaches, which further indicates the importance of incorporating the knowledge graph as auxiliary information in recommendation algorithms.

3) Effect of Depths of Layer.

Under different maximum depths of the knowledge-aware propagation layer L, the performance of DCIKG-Rec is shown in Figure 3.

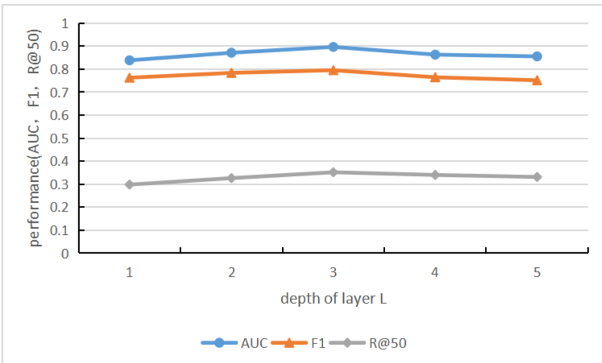


Fig. 3. The performance of DCIKG-Rec at different maximum depths of the knowledge-aware propagation layer L

As we can see from Figure 3, when the value of L is 3, all the indicators of the model achieve the best performance. One possible reason for this phenomenon is that as the number of propagation layers increases, the model acquires more knowledge information but is also subject to more noise interference, especially when the data volume is large. Maintaining a reasonable depth of the propagation layer can maximize the utilization of knowledge information in different scenarios.

VI. Conclusion

In light of the limitations of existing knowledge graph-based recommendation methods in user-end modeling and the utilization of interactive collaborative information, the proposed DCIKG-Rec model holds significant value. By jointly modeling both the user side and the course side, integrating collaborative information with knowledge associations, employing a heterogeneous propagation strategy to enrich user and course representations, and leveraging knowledge-aware attention embedding together with biased attention mechanisms to strengthen semantic information from entity neighbors while mitigating the loss of collaborative information in multi-layer propagation, the model substantially enhances the effectiveness of course recommendations. Rigorous experiments on real-world datasets demonstrate that DCIKG-Rec consistently outperforms several state-of-the-art baselines. Looking ahead, future work will further evaluate the feasibility of the model on additional datasets and incorporate students' historical learning information as auxiliary data, thereby improving both the accuracy and interpretability of the recommendation framework.

Acknowledgment

This work was supported by the Fundamental Research Funds for the Central Universities (GK202405003), Digital Education Research and Practice Project of Shaanxi Normal University (JYSZH201301) and the Special Programon AI-Driven Undergraduate Instructional Reform at Shaanxi Normal University under Grant number (25ZDZB17, 25ZDZB31).

Declarations of Competing Interest The authors report there are no competing interests to declare.

References

[1] H. K. Ahmad, B. Liu, B. A. Muhammad, and M. Umar, "Prebige: Course recommendation using course prerequisite relation embedding and bipartite graph embedding," *Journal of Networking and Network Applications*, vol. 2, no. 4, pp. 161–171, 2022.

[2] P. Tian, "Context-aware music recommendation based on word2vec,"

[3] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE transactions on knowledge and data engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[4] Q. Ai, V. Azizi, X. Chen, and Y. Zhang, "Learning heterogeneous knowledge base embeddings for explainable recommendation," *Algorithms*, vol. 11, no. 9, p. 137, 2018.

[5] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.

[6] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 28, 2014.

[7] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, 2015.

[8] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, pp. 687–696, 2015.

[9] G. Ji, K. Liu, S. He, and J. Zhao, "Knowledge graph completion with adaptive sparse transfer matrix," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, 2016.

[10] L. Xu, Z. Zhao, and D. Liang, "Prekgat: Using course prerequisite relation and knowledge graph attention network for course recommendation," in *2024 International Conference on Networking and Network Applications (NaNA)*, pp. 351–356, IEEE, 2024.

[11] T. Xuan, D. Qi, L. Zihui, and S. Guodong, "Survey on deep learning based news recommendation algorithm," *Journal of Frontiers of Computer Science & Technology*, vol. 15, no. 6, p. 971, 2021.

[12] K. Zhu, Y. Xiao, W. Zheng, X. Jiao, and C.-H. Hsu, "A novel context-aware mobile application recommendation approach based on users behavior trajectories," *IEEE Access*, vol. 9, pp. 1362–1375, 2020.

[13] Y. Du, X. Zhu, L. Chen, B. Zheng, and Y. Gao, "Hakg: Hierarchy-aware knowledge gated network for recommendation," in *Proceedings of the 45th international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 1390–1400, 2022.

[14] F. Zhai and B. Li, "Point of interest recommendation based on graph convolutional neural network," in *Journal of Physics: Conference Series*, vol. 1883, p. 012132, IOP Publishing, 2021.

[15] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *The world wide web conference*, pp. 3307–3313, 2019.

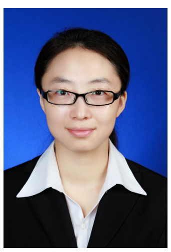
[16] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 950–958, 2019.

[17] B. Hu, C. Shi, W. X. Zhao, and P. S. Yu, "Leveraging meta-path based context for top-n recommendation with a neural co-attention model," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1531–1540, 2018.

[18] L. Sang, M. Xu, S. Qian, and X. Wu, "Knowledge graph enhanced neural collaborative recommendation," *Expert systems with applications*, vol. 164, p. 113992, 2021.

[19] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *nature*, vol. 405, no. 6789, pp. 947–951, 2000.

[20] J. Han and C. Moraga, “The influence of the sigmoid function parameters on the speed of backpropagation learning,” in *International workshop on artificial neural networks*, pp. 195–201, Springer, 1995.
[21] J. Yu, G. Luo, T. Xiao, Q. Zhong, Y. Wang, W. Feng, J. Luo, C. Wang, L. Hou, J. Li, *et al.*, “Mooccube: A large-scale data repository for nlp applications in moocs,” in *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 3135–3142, 2020.
[22] D. P. Kingma, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
[23] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, JMLR Workshop and Conference Proceedings, 2010.
[24] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, “Collaborative knowledge base embedding for recommender systems,” in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 353–362, 2016.
[25] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo, “Ripplenet: Propagating user preferences on the knowledge graph for recommender systems,” in *Proceedings of the 27th ACM international conference on information and knowledge management*, pp. 417–426, 2018.
[26] H. Wang, F. Zhang, M. Zhang, J. Leskovec, M. Zhao, W. Li, and Z. Wang, “Knowledge-aware graph neural networks with label smoothness regularization for recommender systems,” in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 968–977, 2019.



Yizhi Zhang is a Senior Engineer at the Office of Information Technology Development and Management, and a PhD candidate at the School of Artificial Intelligence and Computer Science, Shaanxi Normal University, China. Her research focuses on smart education.

Xinyuan Ji received her B.Sc degree from Luoyang Normal University, Luoyang, China, in 2024. She is currently pursuing the Master degree in school of Computer Science with Shaanxi Normal University, Xian, China. Her research focus on scene computing in Education.

Danni Yan received her B.Eng. degree in Computer Science and Technology from Northeast Forestry University, China, in 2023. She is currently a graduate student specializing in Artificial Intelligence at Shaanxi Normal University, Xi’an, China. Her current research focuses on the application of AI in education, with specific interests in smart education systems and automated code repair techniques.

Liming Xu received his Master’s degree in Computer Science and Technology from the School of Artificial Intelligence and Computer Science, Shaanxi Normal University, China, in 2025. His research interests include knowledge graph-based recommendation systems and machine learning.

Bello Ahmad Muhammad received his B.Sc. and M.Sc. degrees in Computer Science from Bayero University, Kano, Nigeria, in 2010 and 2015, respectively, and his Ph.D. in Computer Software Theory from Shaanxi Normal University, China. He is currently a postdoctoral researcher at the School of Computer Science and Technology, Henan Institute of Science and Technology, China. His research interests include learning analytics, knowledge graph-based recommendation systems, and artificial intelligence in education.