

# Construction and Application of Personalized Learning Resources Based on Knowledge Graph in Higher Education Self-study Examination Environment

Anxue Zhao and Xiaoting Jia \*

School of Information Science and Technology, Qiongtai Normal University, Haikou, Hainan, 571127, China;  
zhao20190327@163.com

**Abstract:** This paper first outlines the overall construction process of a course resource knowledge graph, analyzes the acquired dataset, and performs data cleaning and preprocessing. Relationships and entities are extracted from the dataset, and the course knowledge graph is constructed as triples and stored in the Neo4j graph database. Subsequently, semantic and difficulty information is extracted from exercise texts and learning records to obtain a comprehensive representation of students' responses at a given time. During the knowledge state update phase, unidirectional and bidirectional propagation modes are adopted for different relationships. Finally, against the backdrop of personalized learning in online education, a bipartite graph-based collaborative filtering course recommendation algorithm (CFBGR) is proposed, effectively mining the deep interaction information between learners and courses. The results show that the AUC and ACC values of the proposed model outperform those of some existing models, indicating that the research approach of considering knowledge point embedding in the knowledge tracking model has certain research significance. When recommending questions to students, the proposed method can tailor recommendations to each student's personalized learning level, ensuring that the recommended questions align with their individual needs. Additionally, the question recommendation results exhibit a high degree of accuracy and interpretability.

**Keywords:** knowledge graph; knowledge state update; collaborative filtering recommendation algorithm; knowledge tracking model

## 1. Introduction

With the development of educational informatization and intelligence, personalized learning has become a key approach to improving learning efficiency and experience [1]. Traditional teaching models struggle to adapt to individual differences and learning needs, leading to inconsistent learning outcomes and difficulty in sustaining student interest [2]. Knowledge graphs, as tools capable of representing relationships between knowledge points, offer new technical means for achieving personalized learning [3]. By constructing knowledge graphs, knowledge points from different disciplines can be organized in an interconnected manner, enabling learners to intuitively understand the relationships between knowledge points. Based on their current knowledge state, learners can receive dynamic recommendations for suitable learning resources [4-5]. Related research not only provides technical support for personalized education but also creates new possibilities for the application of knowledge graphs in the field of education [6].

As a structured form of knowledge representation, knowledge graph has a wide range of applications [7]. For example, the interpretable and personalized cognitive reasoning model of the knowledge graph was applied to the patient's electronic medical record and knowledge graph to provide personalized diagnosis and general practice decision-making, and the study [8] showed that the proposed model outperformed all baselines, with a precision@1 of 0.7873, a recall@10 of 0.9020, and a hits@10 of



0.9340. Reference [9] proposes a knowledge graph construction method for a clothing customization production system based on bidirectional fusion. Taking the suit production process of a certain custom clothing factory as an example, the proposed knowledge fusion method is validated for its effectiveness in clothing customization based on the corresponding knowledge graph. There has been extensive research on personalized learning based on knowledge graphs. However, since knowledge graph methods alone still have limitations in terms of accuracy, real-time performance, and scalability, most studies have improved the performance of personalized learning through integration or optimization [10]. Literature [11] designed an efficient personalized recommendation algorithm based on interest similarity and knowledge relevance, and developed a corresponding recommendation system. Using discrete mathematics as an example, experiments demonstrated the critical role of knowledge graphs in personalized recommendations. Literature [12] proposes a knowledge graph higher-order relevance modeling method based on graph convolutional networks, which can more accurately capture learner preferences. By weighting two factors—learner preferences and the importance of learning resources—it provides learners with the optimal learning path. Literature [13] explores the application value of integrating knowledge graphs (KGs) and causal inference (CI) in optimizing students' personalized learning paths. The research results indicate that this method helps to balance knowledge modeling, adaptability, and empirical rigor to advance personalized education.

In terms of personalized learning resource recommendation, Reference [14] developed a knowledge graph-based personalized learning resource recommendation model. Experimental results showed that the model achieved AUC and ACC values of 0.9186 and 0.8836, respectively, on the SCHOLAR dataset, and AUC and ACC values of 0.7898 and 0.6541, respectively, on the IMOOC dataset, thereby meeting the personalized learning resource recommendation needs of students. Reference [15] proposed an improved knowledge graph model to match the uniqueness of knowledge graphs in the education field. The results showed that the model improved the recall rate of course resources to 0.95-0.96, achieved a personalized satisfaction rate of 0.96, and reached a recommendation resource difference of 0.9, verifying the robustness of the algorithm in the diversity of course resources. Literature [16] employs deep knowledge tracking technology to automatically construct course knowledge graphs, combining recommendation algorithms to provide students with recommended practice resources from a practice repository. The experimental results of this algorithm are consistent with mainstream algorithms, and some metrics outperform existing algorithms. Literature [17] proposes an online learner knowledge graph system to provide effective learning resources for different types of learners. The system utilizes teaching data to provide personalized learning content. The results show that the system can accurately provide personalized learning resources for learners with different learning abilities, thereby improving their learning speed. Literature [18] recommends appropriate educational content to learners using a knowledge graph and quantifies the similarity between learner preferences and educational entity attributes in the knowledge graph using cosine similarity technology. Through simulation experiments and algorithm evaluation, the superior performance of the proposed method in providing personalized learning resources for students is demonstrated.

The application of knowledge graphs to personalized learning focuses on how to scientifically and systematically construct domain-specific knowledge graphs that fully cover the target domain knowledge and form high-quality personalized learning resources that meet users' diverse learning needs.

This paper deconstructs the data acquisition and preprocessing steps for course resources, focusing on the knowledge extraction process of course resources, introducing the storage methods for course resource knowledge graphs, and discussing knowledge graph embedding techniques. It then introduces a feature and relationship-enhanced knowledge tracking model (FRKT), which extracts semantic information and difficulty information from exercise texts and learning records, respectively. By combining the knowledge point representations contained in the exercises with the correctness of answers, it obtains a comprehensive representation of the response at a given moment. During the knowledge state update phase, FRKT employs unidirectional and bidirectional propagation modes, applying different propagation methods to different relationships to track changes in students' knowledge states. A graph structure is then used to model the historical interactions between learners and courses, constructing a learner-course bipartite graph. Graph convolutional neural networks are employed to extract local structural features from the bipartite graph, enhancing the feature representations of learners and courses to improve course recommendation accuracy. Finally, the effectiveness of personalized resource recommendation in the context of higher education self-study examinations is analyzed.

## 2. Personalized learning resource recommendation model based on knowledge graphs

### 2.1. Course Resource Knowledge Graph Construction

#### 2.1.1. Course Resource Knowledge Graph Construction Process

Based on the scope of knowledge coverage, knowledge graphs can be categorized into two types: general and domain-specific. General knowledge graphs focus on breadth, encompassing a wide range of concepts and entities, and are primarily used for popular science education and answering common knowledge questions, thereby enhancing search engine performance. Domain-specific knowledge graphs, on the other hand, focus on industry depth, require customization, and address specialized problems. The course recommendation graph discussed in this paper is an example of such a graph, which deeply explores course-related knowledge. The process of constructing a knowledge graph involves several steps: data acquisition, data cleaning and preprocessing, entity alignment, and knowledge storage. The process of constructing the course knowledge graph in this paper is shown in Figure 1. The following is a detailed description of the steps involved in constructing the course knowledge graph:

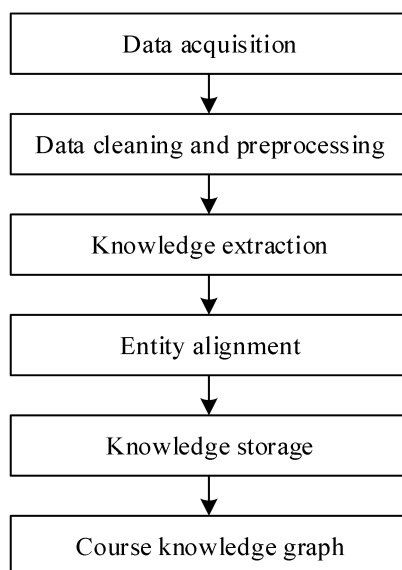
(1) Course dataset acquisition: The Mooper dataset was downloaded from a public dataset website, which primarily contains 23,412 practical training data points.

(2) Data cleaning and preprocessing: Removing obviously invalid data from the course dataset and checking for duplicate course data.

(3) Knowledge extraction: Since the Mooper dataset has already structured and presented course information and user-course interaction information, no additional extraction is required.

(4) Entity alignment: Entities in the data sources are identified and normalized to standardize entity representations across different data sources, converting them to simplified Chinese characters.

(5) Knowledge storage: The extracted entities and relationships are stored in a graph database in the form of triples.



**Figure 1.** The course knowledge map builds the flowchart.

#### 2.1.2. Course resource data acquisition and preprocessing

The MOOPER dataset was downloaded from a public dataset website and consists of two parts: interaction data and knowledge graph data. Taking course data as an example, course data includes course name, course description, number of course visitors, publication date, etc. Before using the data, data cleaning must be performed to remove invalid data. Data validation and correction are necessary to ensure data accuracy. To ensure the accuracy of course chapter information, chapter information must be checked. To ensure the accuracy of course knowledge points, it is necessary to verify whether the knowledge points are related to the course title and description, ensuring that the knowledge points are indeed covered in the course.

### 2.1.3. Course Resource Knowledge Extraction

Entities and relationships in the course knowledge graph are extracted from the obtained dataset. Since the obtained dataset is structured data, the knowledge extraction process is simplified. Specifically, entities and relationships are defined manually, and then automated programs are used for knowledge extraction, thereby avoiding the tedious process of extracting knowledge from text.

#### (1) Course resource entity extraction

Entity-related datasets can be found in the acquired data. Analyze the dataset to obtain relevant information and define entities based on this information. This paper defines 11 types of entities, with 11 entity names: course number, chapter number, practical training number, level number, knowledge point number, course access number range, teacher number, subject area, subject, school or institution, and course release time.

#### (2) Course resource relationship extraction

A knowledge graph is a graph data structure used to represent relationships between entities, with its core focus on revealing the hierarchical relationships and complex connections between entities. When constructing a knowledge graph in the course domain, these hierarchical relationships and associations become clearly evident. Courses serve as the core and concrete entities in the knowledge graph. Practical training includes multiple levels, and knowledge points are the most basic teaching units in a course, forming the foundation of course content. Each level involves one or more knowledge points. Treating entity categories as attributes of a course allows for a more concise representation of relationships between entities, making the knowledge graph easier to understand and use. The extracted relationships include 10 types. The final relationship data information is obtained.

### 2.1.4. Course Resource Knowledge Graph Storage

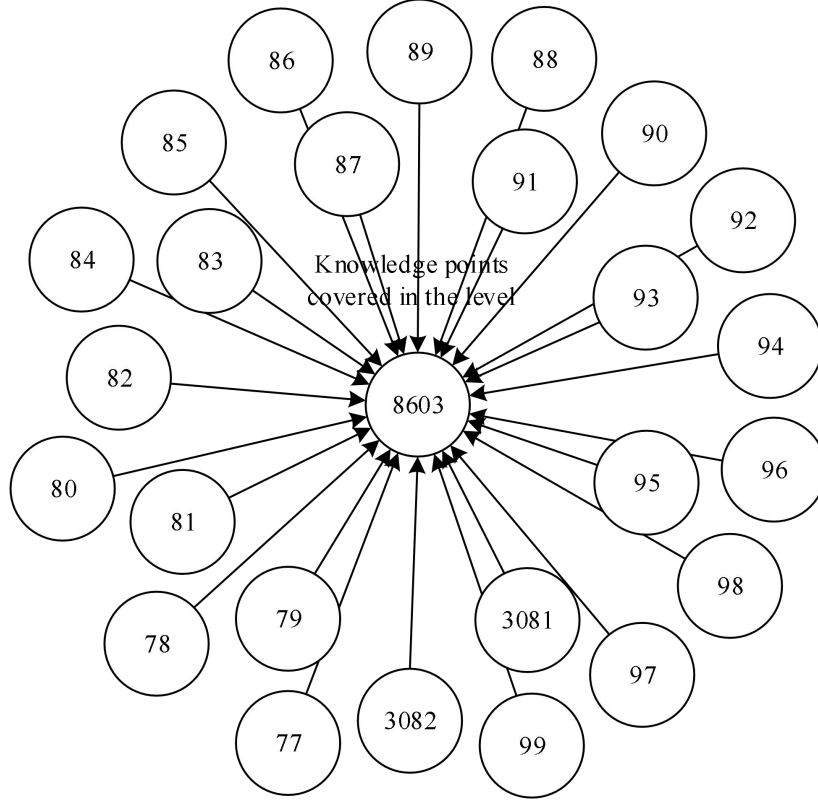
Currently, the main storage methods include Resource Description Framework (RDF) and graph databases. Compared to RDF, graph databases are non-relational databases that store and manage data using a graph data structure. In this data structure, entities are represented as nodes in the graph, while relationships between entities are represented by edges, allowing for an intuitive display of the relationships between entities.

This paper stores well-defined triples of entities and relationships in a graph database, providing more possibilities for subsequent knowledge reasoning and mining.

In the course resource knowledge graph, the triplet  $(h1, r, h2)$  is used as the basic representation unit. The entire course knowledge graph can be represented as  $G$ , which is composed of a series of such triplets, collectively depicting the knowledge system and the relationships between entities within the course domain. Course knowledge graph  $G$  expression:

$$G = \{(h1, r, h2) \mid h1, h2 \in H, r \in R\} \quad (1)$$

Taking the MOOPer dataset as an example, where  $h1, h2 \in H$ ,  $H$  is all courses in the MOOPer dataset,  $r \in R$ , and  $R$  constructs 10 relationships to describe the complex connections between these courses and their internal elements. The visualization results of the knowledge points contained in the levels are shown in Figure 2.



**Figure 2.** The level contains the knowledge point visualization diagram.

### 2.1.5. Knowledge Graph Embedding

After constructing the knowledge graph for the course domain, entities and relationships are still presented in text form and need to be further converted into vector form for computation. After training the model using the training dataset, the threshold  $T$  for the triplet prediction score is adjusted using the validation dataset to make a binary decision. The specific model evaluation metrics are:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

Among them,  $TP$  predicts positive triples as positive;  $TN$  predicts negative triples as negative;  $FP$  predicts positive triples as negative;  $FN$  predicts negative triples as positive. During the model training phase, the gradient descent method is used to optimize the knowledge embedding model.

## 2.2. Knowledge tracking model based on feature and relationship enhancement

### 2.2.1. Problem Description and Definition

As an important part of educational data mining, knowledge tracking aims to predict students' future learning performance and learning needs by establishing models of changes in students' knowledge status over time, thereby providing personalized learning support and recommendations.

Given a student's historical learning interaction sequence  $X = \{x_1, x_2, \dots, x_t\}$ , where  $x_t = (q_t, r_t)$  indicates that the student answered question  $q_t$  at time  $t$ , and  $r_t$  indicates the answer status (0 indicates an incorrect answer, and 1 indicates a correct answer).

This paper also introduces a graph structure, which can be symbolically represented as follows: the graph structure  $G(V, E)$  represents the complex relationship between exercises and knowledge points, where  $V = \{Q, K\}$  contains two types of nodes, and  $E = \{E^r, r = 1, \dots, R\}$  contains  $R$  types of relationships.

### 2.2.2. Overall Model Architecture

This section provides an overview of the proposed knowledge tracking model [19-20] FRKT. The knowledge state is represented by a vector of length  $d_h$ . There are  $N$  vectors for  $N$  knowledge points, forming an  $N \times d_h$  matrix. The values of the elements in the matrix range from  $[0,1]$ , with values closer to 1 indicating a higher level of mastery. Each student has an initial knowledge state  $H_0$ . After each question, the knowledge state  $H_{t-1}$  passed from the previous moment is updated based on the current question and the correctness of the answer. After the update unit performs the update, a new knowledge state  $H_t$  is obtained.

This process is conducted from both temporal and spatial perspectives. Temporal information updates employ the method from the classical deep knowledge tracking model, where the knowledge state  $h_{t-1}^j$  of knowledge point  $j$  is updated to  $h_{t-1,T}^j$  through a gate function  $f_{temporal}$ .

When faced with a new exercise, the model needs to use the knowledge state  $H_t$  and the new exercise  $q_{t+1}$ , combined with the knowledge points contained in the exercise, to obtain the predicted probability of answering the question correctly  $\tilde{r}_{t+1}$  using a mapping function  $f_{out}$ .

### 2.2.3. Feature-enhanced knowledge point embedding representation

#### (1) Text semantic information

Smooth inverse frequency (SIF) technology reduces the weight of unimportant words while retaining the information that contributes most to the semantics of the exercise, namely:

$$q_{i,text} = \frac{1}{|S_i|} \sum_{w \in S_i} \frac{a}{a + p(w)} f_1(w) \quad (3)$$

Where  $a$  is the training parameter,  $S_i$  represents all words in the  $i$  th exercise, and  $p(w)$  represents the probability of each word  $w$ .

#### (2) Difficulty information

Use statistical methods to obtain the exercise difficulty index:

$$q_{i,difficulty} = Q_d[level] \quad (4)$$

$$level = \sum_m \frac{|S_i| \cdot |\{r_{im} == 1\}|}{|S_i|} \cdot C_q$$

Among them,  $C_q$  is the predefined question difficulty level,  $S_i$  is the set of students who answered question  $i$ ,  $\{r_{im} == 1\}$ .

#### (3) Feature-enhanced embedding representation

For each knowledge point, combine the feature vector of the exercise with the knowledge point vector, and finally merge it with the correctness of the answer in the interaction record to obtain the answer-response vector for this knowledge point at this moment:

$$x_{t,kj} = \begin{cases} [q_{ij} \oplus 0], & \text{if } r_t = 1 \\ [0 \oplus q_{ij}], & \text{if } r_t = 0 \end{cases} \quad (5)$$

$$q_{ij} = W_1[k_j \oplus q_i] + b_1$$

$$q_i = q_{i,text} \oplus q_{i,difficulty}$$

Among them,  $W_1$  and  $b_1$  are learning parameters,  $k_j$  is the  $j$  th knowledge point included in exercise  $i$ ,  $q_{ij}$  is the representation of the  $j$  th knowledge point in exercise  $i$  after feature enhancement,

and the dimension of the vector  $\mathbf{0} = (0, 0, \dots, 0)$  is consistent with that of  $q_{ij}$ .

#### 2.2.4. Updating knowledge status

##### (1) Time Information Update

Based on educational research on knowledge point learning and previous work on knowledge tracking tasks, when a student solves a problem, it has an effect and influence on his or her knowledge state. At time  $t$ , a temporal learning effect  $\varepsilon_t^t$  is generated for knowledge point  $k$ , which updates the knowledge state of knowledge point  $j$  from  $h_{t-1}^j$  to  $h_{t-1,T}^j$ . Since a single exercise may involve multiple knowledge points, the knowledge states of all knowledge points included in the exercise must be updated:

$$\xi_\tau^t = x_{t,kj} E_r \quad (6)$$

where  $E_r \in \mathbb{R}^{2N \times d_e}$  is the embedding matrix of  $x_{t,kj}$ . Then, the temporal effect vector  $\xi_\tau^t$  is input into a gate function  $f_{temporal}$  to update the knowledge state:

$$h_{t-1,T}^j = f_{temporal}(\xi_\tau^t, h_{t-1}^j) \quad (7)$$

Here, the  $f_{temporal}$  function is an LSTM unit.

##### (2) Spatial information update

###### 1) Unidirectional propagation

After the knowledge state of knowledge point  $j$  changes from  $h_{t-1}^j$  to  $h_{t-1,T}^j$ , the change in this knowledge point will propagate along the directed relationship to its successor:

$$\begin{aligned} oneway_{jm}^r &= f_{spatial}(h_{t-1,T}^j, h_{t-1}^j, E_c(j)), \forall m \in S^r(j) \\ f_{spatial}(h_{t-1,T}^j, h_{t-1}^j, E_c(m)) \& = \text{relu}(W_p^r P_{jm}^r + b_p^r) \\ P_{jm}^r &= (h_{t-1,T}^j - h_{t-1}^j) \oplus E_c(m) \end{aligned} \quad (8)$$

$S^r(j)$  returns all successor knowledge points of knowledge point  $j$  in relation  $r$ , where  $W_p^r$  and  $b_p^r$  are learning parameters.  $\oplus$  is an operation that concatenates two vectors into a long vector.

###### 2) Bidirectional propagation

This paper uses two formulas to model the influence of knowledge point  $j$  on its neighboring nodes and the influence of neighboring nodes on knowledge point  $j$ . The following formula is used to model the influence of knowledge point  $j$  on its neighboring nodes:

$$\begin{aligned} toneighbor_{jm}^r &= f_{spatial}(h_{t-1,T}^j, h_{t-1}^j, E_c(m)), \forall m \in N^r(j) \\ f_{spatial}(h_{t-1,T}^j, h_{t-1}^j, E_c(m)) &= \text{relu}(W_s^r S_{jm}^r + b_s^r) \\ S_{jm}^r &= h_{t-1,T}^j \oplus h_{t-1}^j \oplus E_c(m) \end{aligned} \quad (9)$$

$N^r(j)$  returns all neighbor nodes of knowledge point  $j$  in relation  $r$ .  $W_s^r$  and  $b_s^r$  are learning parameters.

The following formula is used to model the influence of neighbor nodes on knowledge point  $j$ :

$$\begin{aligned} fromneighbor_j^r &= \text{relu}(W_{ss}^r R_j^r + b_{ss}^r) \\ R_j^r &= \left( h_{t-1,T}^j + \sum_{m \in N^r(j)} h_{t-1}^m \right) \oplus E_c(j) \end{aligned} \quad (10)$$

Among them,  $W_{ss}^r$  and  $b_{ss}^r$  are learning parameters.

For each affected knowledge point  $m$ , first calculate its aggregate impact  $I_m$  by determining whether the knowledge point has been affected by unidirectional and bidirectional multiple impacts, and finally update its knowledge state through a gate function. The specific calculation formula is as follows:

$$\begin{aligned} h_t^m &= f_2(I_m, h_{t-1}^m) \\ I_m &= \text{relu}(W_I A_m + b_I) \\ A_m &= \begin{cases} \sum_r \text{fromneighbor}_m^r & m = j \\ \gamma \cdot \sum_r \text{oneway}_{jm}^r + (1-\gamma) \cdot \sum_r \text{toneighbor}_{jm}^r & m \neq j \end{cases} \end{aligned} \quad (11)$$

Among them,  $W_I$  and  $b_I$  are the weight matrix and bias term,  $\gamma$  is the hyperparameter, and  $f_2$  is the gating function updated using the LSTM unit.

### 2.2.5 Prediction and Output

To evaluate the model's performance, for a new exercise  $q_{t+1}$ , the model outputs the probability that the student will answer the question correctly based on the knowledge state of the knowledge points contained in the exercise:  $P(r_{t+1} = 1 | q_{t+1}, X_t, H_t, G) = p_{t+1}$ . The specific calculation process is as follows:

$$\begin{aligned} p_{t+1} &= f_{out}(q_{t+1}) \\ f_{out}(q_{t+1}) &= \sigma \left( W_o \sum_{j \in E^k(q_{t+1})} h_t^j + b_o \right) \end{aligned} \quad (12)$$

Among them,  $W_o$  is the weight matrix learned,  $b_o$  is the bias term, and  $E^k(q_{t+1})$  is the set of knowledge points contained in exercise  $q_{t+1}$ .

During the training phase, the parameters in the model are learned by minimizing the standard cross-entropy loss between  $p_{t+1}$  and the true label  $r_{t+1}$ :

$$L = - \sum_t (p_{t+1} \log r_{t+1} + (1 - p_{t+1}) \log(1 - r_{t+1})) \quad (13)$$

## 2.3. Online course recommendation algorithm based on bipartite graphs

### 2.3.1. Model Framework

In the collaborative filtering course recommendation algorithm based on bipartite graphs [21], the model first inputs historical interaction data between learners and courses, then generates an initial graph structure based on the historical interaction data. This includes a learner-course bipartite graph and the initial vector representations of the nodes in the graph. A graph convolutional neural network is then used to learn the vector embeddings of learners and courses. Based on the feature information of learners and courses obtained from the embeddings, the inner product of the vectors is finally calculated for prediction and recommendation. The model consists of three main modules: the embedding layer, the propagation layer, and the prediction layer. The embedding layer initializes the vector representations of learners and online courses, modeling learners and courses using their interaction information. In the propagation layer, the interaction modeling expressions of learners and online courses obtained from the embedding layer are fed into the GCN network to extract higher-order information about learners and online courses, thereby optimizing the embedding representations. Finally, the prediction layer integrates the embedding representations from the propagation layer and the parameters learned within the model to predict the recommendation results. The following sections provide a detailed introduction to the components of the model.

### 2.3.2. Feature Vector Initialization

The embedding layer is responsible for inputting learner vectors and course vectors into the model and modeling learner-course interaction information. Assuming that the number of learners input into the model is  $M$  and the number of courses is  $N$ , this paper uses  $e_{u_i}^{(0)} \in R^d$  to represent the embedding vector of the  $i$  th learner in the embedding layer, and  $e_{v_j}^{(0)} \in R^d$  denotes the  $j$  th course embedding vector, where  $d$  represents the embedding dimension, which is an adjustable hyperparameter. The set of all learner embedding vectors is denoted as  $e_u^{(0)} \in R^{M \times d}$ , and the set of all course embedding vectors is denoted as  $e_v^{(0)} \in R^{N \times d}$  as follows:

$$e_u^{(0)} = [e_{u_1}^{(0)}, \dots, e_{u_i}^{(0)}, \dots, e_{u_m}^{(0)}] \quad (14)$$

$$e_v^{(0)} = [e_{v_1}^{(0)}, \dots, e_{v_j}^{(0)}, \dots, e_{v_n}^{(0)}] \quad (15)$$

In the model presented in this paper, the learner-course bipartite graph is input into a graph neural network, which further represents the interaction information between learners and courses in embedded vectors to obtain feature vectors with better expressive power.

### 2.3.3. Message Passing Mechanism

First, define the message propagation between nodes in the learner-course bipartite graph. For two adjacent nodes  $(u, i)$  in the bipartite graph, let  $e_u^{(0)}$  denote the embedding of learner  $u$  at layer 0, and let  $e_i^{(0)}$  denote the embedding of course  $i$  at layer 0. The information provided by the courses with which learner  $u$  interacts and the information provided by the learners with whom course  $i$  interacts can be represented as:

$$c_{u,i} = \sum_{i \in N_u} \frac{1}{\sqrt{|N_i|} \sqrt{|N_u|}} e_i^{(0)} \quad (16)$$

$$c_{i,u} = \sum_{u \in N_i} \frac{1}{\sqrt{|N_u|} \sqrt{|N_i|}} e_u^{(0)} \quad (17)$$

Where  $e_u^{(1)}$  is the vector representation of learner  $u$  in the first hop of the graph convolutional network,  $N_u$  denotes the number of first-order neighbors of learner  $u$ , and  $N_i$  denotes the number of first-order neighbors of course  $i$ .

This paper aggregates the neighbors connected by nodes and adds the self-connected part to obtain the complete first-order vector representation of learner  $u$  and course  $i$ :

$$e_u^{(1)} = \text{Leaky ReLU} \left( W e_u^{(0)} + \sum_{i \in N_u} \frac{1}{\sqrt{|N_i|} \sqrt{|N_u|}} e_i^{(0)} \right) \quad (18)$$

$$e_i^{(1)} = \text{Leaky ReLU} \left( W e_i^{(0)} + \sum_{u \in N_i} \frac{1}{\sqrt{|N_u|} \sqrt{|N_i|}} e_u^{(0)} \right) \quad (19)$$

The first-order propagation rules of learner nodes and course nodes in graph neural networks constitute the first-order relationship between learners and courses. In a bipartite graph, after  $k$  layers of propagation, the embedded vectors of learners and courses can be obtained as low-dimensional vector representations of nodes:

$$e_u^{(k+1)} = \text{Leaky ReLU} \left( W^{(k+1)} e_u^{(k)} + \sum_{i \in N_u} \frac{1}{\sqrt{|N_i|} \sqrt{|N_u|}} e_i^{(k)} \right) \quad (20)$$

$$e_i^{(k+1)} = \text{Leaky ReLU} \left( W^{(k+1)} e_i^{(k)} + \sum_{u \in N_i} \frac{1}{\sqrt{|N_u|} \sqrt{|N_i|}} e_u^{(k)} \right) \quad (21)$$

Among them,  $e_u^{(k+1)}$  represents the embedded representation of learner  $u$  at the  $k+1$  jump, and  $e_i^{(k+1)}$  represents the embedded representation of course  $i$  at the  $k+1$  jump. In the formula,  $e_u^{(k)}$  is the vector representation of the course generated from the previous message propagation process, which stores the messages of the  $k$  jump neighbors.

#### 2.3.4. Prediction Functions and Optimization

Let  $E^{(0)}$  denote the embedding matrix of the initial node,  $E^{(0)} = [e_1^{(0)}, e_2^{(0)}, \dots, e_n^{(0)}]^T$ , and let  $E^{(k)}$  denote the embedding matrix of the node after  $k$  layers of iteration. The embedding propagation of the first layer can be expressed in matrix form as:

$$E^{(1)} = \sigma \left( E^{(0)} W + L E^{(0)} \right) \quad (22)$$

In the equation,  $\sigma$  is the activation function, which is *Leaky ReLU* in this case.  $L$  is the Laplacian matrix of the learner-course bipartite graph, which is expressed as:

$$L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \quad (23)$$

Where  $A$  is the adjacency matrix and  $D$  is the degree matrix of the adjacency matrix. In high-order graph convolutional networks, node embedding propagation can be represented as:

$$E^{(k+1)} = \sigma \left( E^{(k)} W^{(k+1)} + L E^{(k)} \right) \quad (24)$$

After the forward propagation layer operation, the learned learner node vector is represented by  $e_u$ , and the course node vector is represented by  $e_i$ . For a given learner and course, the learner's preference for the course can be calculated using the inner product. Finally, the final rating prediction for learner  $u$  for item  $i$  is:

$$\hat{p}_{ui} = e_u^T e_i \quad (25)$$

The above calculations can be used to obtain the scores for learners and courses.

This paper uses a logarithmic loss function to convert the recommendation problem into a classification problem, which is optimized using paired learning methods during training. Specifically, a triplet  $(u, i^+, i^-)$  is constructed, where  $(u, i^+)$  represents the observed interaction relationship, denoted by  $R^+$ , and  $(u, i^-)$  represents the unobserved interaction relationship, denoted by  $R^-$ . The importance of observed interaction items is prioritized over unobserved interaction items, and the difference in scores between positive and negative samples is maximized as much as possible. The specific formula is as follows:

$$\text{Loss} = \sum_{(u, i^+, i^-) \in O} -\ln \sigma \left( \hat{p}_{ui^+} - \hat{p}_{ui^-} \right) + \lambda \|\Theta\|_2^2 \quad (26)$$

In the equation,  $O = \{(u, i^+, i^-) \mid (u, i^+) \in R^+, (u, i^-) \in R^-\}$  represents paired training data,  $\sigma$  represents the *sig mod*( $\cdot$ ) function, and  $\lambda \|\Theta\|_2^2$  is the regularization term. used to improve the model's

generalization ability and prevent overfitting, where  $\lambda$  is the regularization coefficient, and  $\Theta$  represents the set of trainable parameters in the model, including learners, course feature vectors, and the weight matrix  $W$ . In this paper, the Adam optimization algorithm [22] is used in the experiments to optimize the model parameters and minimize the loss function.

### 3. Knowledge recommendation effectiveness of learning resources in higher education self-study examinations

#### 3.1. Effectiveness of personalized recommendation of learning resources based on knowledge tracking

##### 3.1.1. Experimental setup

To evaluate the effectiveness of the proposed method, this chapter selects two publicly available datasets, ASSISTments2021 and ASSISTments2024, which are widely recognized and frequently used in knowledge tracking research, as experimental data. To ensure more accurate experimental results, we filtered out invalid information from the datasets and removed records with fewer than two interactions. The specific information of the datasets is shown in Table 1. In classification prediction tasks, cross-validation (CV) is a commonly used model evaluation method, which divides all data into training, validation, and test sets, with data proportions of 80%, 20%, and 20%, respectively. Since this paper primarily focuses on the extension of CFBGR, DKT is a classic algorithm in the field of knowledge tracking, and the DKT-F algorithm also considers the influence of knowledge point embedding behavior during the knowledge tracking process, making it comparable. Therefore, this chapter selects CFBGR, DKT, and DKT-F as experimental comparison methods.

**Table 1.** Specific information of the data set.

	ASSISTments2021	ASSISTments2024
Student population	4194	39351
Problem number	16885	58771
Interactive recording	325627	4193630

##### 3.1.2. Evaluation Indicators

This paper uses the area under the receiver operating characteristic curve (AUC) and accuracy (ACC), commonly used in the field of knowledge tracking, as evaluation metrics for knowledge tracking models.

The true positive rate (TPR) represents the proportion of samples predicted as positive and actually being positive out of all positive samples; the false positive rate (FPR) represents the proportion of samples predicted as positive but actually being negative out of all negative samples:

$$TPR = \frac{TP}{TP + FN} \quad (27)$$

$$FPR = \frac{FP}{TN + FP} \quad (28)$$

##### 3.1.3. Related parameter settings

The Adam optimizer used in the model corrects gradients from a statistical perspective, but the initial learning rate setting still affects the algorithm's operation. If the learning rate is set too high, the model may fail to converge. Therefore, we set different initial learning rates to obtain the validation set loss trend of the model running on two datasets.

In the ASSISTments2021 dataset, we set the initial learning rate to 0.001. In the ASSISTments2024 dataset, we selected 0.0011 as the initial learning rate.

##### 3.1.4. Analysis of experimental results

To test the impact of different combinations of features on the model's predictive performance, this section conducts six experimental comparisons. We first integrated features combined in different ways

into the CFBGR model and obtained experimental results for different models on two datasets. The training results of models with different feature combinations are shown in Table 2. It can be concluded that in the ASSISTments2021 dataset, the average test AUC and ACC of the model integrating all three features reached 89.76% and 80.46%, respectively, which is higher than that of other models. Compared to the CFBGR+attempt+difficulty, CFBGR+attempt+hint, and CFBGR+difficulty+hint models, the CFBGR+attempt+difficulty+hint model saw AUC improvements of 3.24%, 1.04%, and 3.32%, respectively. ACC improved to varying degrees. Additionally, in ASSISTments2024, the model that combined the three features outperformed other combinations in terms of AUC and ACC, with test AUC values increasing by 3.51%, 0.2%, and 2.94% compared to the CFBGR+attempt+difficulty, CFBGR+attempt+hint, and CFBGR+difficulty+hint models, respectively.

Through the above comparative analysis, we have demonstrated that the model achieves optimal predictive performance when considering the three features of attempt count, difficulty, and request for help. Furthermore, the AUC and ACC results of the model after integrating several different combinations of features are mostly superior to the baseline model, which validates the effectiveness of integrating behavioral features and exercise features in optimizing the performance of knowledge tracking models.

**Table 2.** Model training results of different combination characteristics.

Different combination characteristics	ASSISTments2021		ASSISTments2024	
	AUC (%)	ACC (%)	AUC (%)	ACC (%)
CFBGR	84.63	75.16	78.46	78.18
CFBGR+attempt+difficulty	86.52	79.99	81.32	79.41
CFBGR+attempt+hint	88.72	78.85	84.63	81.12
CFBGR+difficulty+hint	86.44	77.80	81.89	78.56
CFBGR+attempt+type	88.24	78.38	84.26	80.56
CFBGR+attempt+difficulty+hint	89.76	80.46	84.83	80.18
CFBGR+attempt+difficulty+hint+type	86.34	79.13	84.73	80.56

Additionally, through experimentation, we found that the predictive performance of the CFBGR model varies depending on the number of features fused. As shown in Table 3, the test results change with the number of features. By comparing the test results of different models, it can be concluded that the number of features fused in the CFBGR model has a certain impact on predictive performance. When the CFBGR model is fused with a single feature, such as the number of attempts, its test AUC values on both datasets outperform the original model. However, when two additional features are added to the CFBGR model, such as the number of attempts and exercise difficulty, its performance on the test set is somewhat better than that of the model with one feature; the CFBGR model with three features has the best prediction performance, with a test AUC of 89.85% on ASSISTments2021 and an AUC of 84.63% on ASSISTments2024, both higher than the other three models. However, when the model is fused with four features, we find that both the AUC and ACC values decrease. This is because incorporating too many features into the neural network introduces noise, interfering with the neural network's ability to calculate the relationships between exercises and knowledge points, thereby leading to a decline in final prediction performance. Therefore, considering multiple features in knowledge tracking models has certain significance for optimizing model performance, but it is necessary to select an appropriate number of more important features.

**Table 3.** As the number of characteristics changes its test results.

Different combination characteristics	ASSISTments2021		ASSISTments2024	
	AUC (%)	ACC (%)	AUC (%)	ACC (%)

CFBGR	84.73	75.16	78.18	78.10
CFBGR+attempt	85.77	78.00	81.22	78.38
CFBGR+attempt+difficulty	89.10	79.04	84.73	81.22
CFBGR+attempt+difficulty+hint	89.85	80.74	84.63	80.18
CFBGR+attempt+difficulty+hint+type	86.44	79.71	84.35	80.66

During the experiment to test the effectiveness of feature fusion, we found that each feature contributes differently to improving model performance. Therefore, we compared the experimental results of the model with feature weights and the original model. The test results of the model with feature weights are shown in Table 4. It can be concluded that the CFBGR model, which simultaneously fuses three types of features, performs better in terms of test AUC and ACC values after feature weighting. In the ASSISTments2021 dataset, the AUC value increased by 0.94%, and the ACC value increased by 3.32%; in the ASSISTments2024 dataset, the AUC and ACC values increased by 3.33% and 1.88%, respectively. This demonstrates that considering the importance of each feature through feature weighting can improve the model's predictive performance to a certain extent.

**Table 4.** The test results of the feature weighted post-model were compared.

Model	ASSISTments2021		ASSISTments2024	
	AUC (%)	ACC (%)	AUC (%)	ACC (%)
CFBGR	87.39	81.26	76.42	75.91
CFBGR + weight	88.33	84.58	79.75	77.79

The training and testing results of the model on the ASSISTments2024 dataset, as well as the testing results of the model incorporating knowledge point embedding behavior, are shown in Table 5. It can be seen that the feature-weighted CFBGR model achieved a 1.87% improvement in the testing AUC value after incorporating the knowledge point embedding mechanism, which demonstrates that considering knowledge point embedding behavior in knowledge tracking models facilitates more accurate modeling of students' knowledge states.

**Table 5.** Model test results of integrated knowledge point embedding behavior.

Model	ASSISTments2024	
	AUC (%)	ACC (%)
CFBGR + weight	79.75	77.79
CFBGR + weight + Knowledge point embedding	81.62	78.52

Finally, we compare the method proposed in this paper with existing knowledge tracking algorithms. To provide a more comprehensive and detailed comparison of the predictive performance between CFBGR and the original model, we set different hidden layer state dimensions (std\_dim) and memory sizes (memory\_size) for them. The results of the model proposed in this paper and those of different models are shown in Table 6. It can be concluded that when std\_dim=50 and memory\_size=25, CFBGR-FMF and CFBGR achieve the best test AUC and ACC, and CFBGR-FMF outperforms the other models. Compared to the baseline model CFBGR, our model achieves a significant improvement in test AUC, with test AUC and ACC increasing by 10.8% and 9.6%, respectively. Compared to the currently state-of-the-art DKT-F model, both AUC and ACC show substantial improvements.

**Table 6.** The results of this model and different models.

Model	std_dim	memory_size	AUC (%)	ACC (%)
This method	50	25	71.51	69.22

	100	15	70.55	68.78
	150	25	70.08	68.21
DKVMN	50	25	82.12	78.82
	100	15	81.58	78.59
	150	25	81.33	80.06
DKT-F	-	-	72.74	69.13
DKT	-	-	71.35	68.67

### 3.2. Analysis of the Effectiveness of Personalized Learning Resource Recommendations

#### 3.2.1. Cognitive Diagnosis DINA Model Diagnosis

This study took 150 students from the higher education self-study examination program at MG University in Province M as the research subjects, diagnosed their learning process, and recommended suitable test questions for them.

##### (1) Diagnostic Process

Based on the test questions, student response data, and knowledge points assessed by the test questions obtained in the previous stage, this experiment calculated the knowledge mastery levels of students in seven classes. The test question-student item response R matrix is shown in Table 7. Question 1 assessed knowledge points A6 and A8, Question 2 assessed knowledge points A5, A9, and A10, and Question 10 assessed knowledge points A1, A2, A3, A4, and A7. Students 1, 2, and 4 answered all questions correctly, while Student 250 answered Questions 1, 2, 3, and 7 correctly.

**Table 7.** Test question - student project reaction r matrix.

Student	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	0
4	1	1	1	1	1	1	1	0	1	1
5	1	1	1	1	1	1	1	1	1	0
6	1	1	1	1	1	0	1	1	1	0
7	1	1	1	1	1	1	1	1	1	0
8	1	1	1	1	1	0	1	1	1	0
9	1	1	1	1	1	1	1	1	0	1
...	...	...	...	...	...	...	...	...	...	...
250	1	1	0	1	0	0	1	0	0	0

##### (2) Experimental Results

Based on the diagnostic results of the DINA cognitive diagnostic model, this experiment primarily analyzed three aspects: the overall mastery of knowledge points by students, the mastery of knowledge points by students in different class groups, and the mastery of knowledge points by individual students.

##### ① Analysis of Overall Knowledge Mastery by Students

By diagnosing the test answer records of students from seven classes, this experiment obtained the specific mastery levels of 10 knowledge points by the overall student population. The overall mastery of knowledge points by students is shown in Table 8. It can be seen that, except for A3, the mastery probability of all other knowledge points by the overall student population is higher than 0.75, indicating that students have achieved mastery of these knowledge points; however, the mastery probability of knowledge point A3 is 38.06%, indicating that students have not achieved mastery of these two knowledge points. Additionally, this paper found that the overall mastery level of knowledge point 3 is poor; the overall mastery levels of knowledge points 1, 6, and 8 are moderate; and the overall mastery

levels of knowledge points 2, 4, 5, 7, 9, and 10 are good, with high mastery probabilities, all approaching 1.

**Table 8.** The overall student knowledge is mastered.

Knowledge point	Master probability
A1 Rational number concept	0.7663
A2 number shaft	0.9688
A3 inverse number	0.3806
A4 absolute value	0.9825
A5 has the right number addition and subtraction	0.9466
A6 has the rational number of multiplication and devoting	0.7986
A7 Rational multiply	0.9321
A8 binomial	0.7926
A9 polynomial	0.9144
A10 reduction	0.9305

② Analysis of Knowledge Mastery Among Different Student Groups

A total of seven classes were evaluated, and the mastery of knowledge points in different classes is shown in Table 9. The probability of students mastering knowledge points varies across classes. Knowledge points A1 and A3 are only tested by the same question, so the probability of students mastering these two knowledge points is the same within the same class. Except for Class 7, all other classes have achieved a mastery level or above for these two knowledge points, with mastery rates ranging from 50.03% to 67.44%. For knowledge points 2, 4, 5, 6, 7, and 9, all classes have reached or exceeded the mastery level, with mastery levels exceeding 70%. For knowledge point 8, Class 1 and Class 2 did not reach the mastery level, with rates of 15.85% and 6.61%, respectively. The class with the highest mastery level was Class 4, at 94.53%. For knowledge point 10, all classes except Class 3 (17.75%) reached a very high mastery level, with Class 4 having the highest mastery level at 99.08%.

**Table 9.** Different class knowledge points master the situation.

Class	Topic and its mastery(%)				
	A1	A2	A3	A4	A5
1	0.5018	0.9808	0.5018	0.8664	0.9742
2	0.6412	0.9305	0.6412	0.9706	0.7413
3	0.5878	0.5305	0.5878	0.9515	0.9328
4	0.5716	0.9815	0.5716	0.9563	0.9783
5	0.5003	0.9211	0.5003	0.9375	0.8671
6	0.6744	0.9871	0.6744	0.9464	0.9172
7	0.3662	0.9883	0.3662	0.9469	0.9165
Class	Topic and its mastery(%)				
	A6	A7	A8	A9	A10
1	0.9418	0.8083	0.1585	0.9264	0.9716
2	0.7629	0.9461	0.0661	0.9527	0.9747
3	0.5873	0.7221	0.5989	0.9265	0.1775
4	0.4946	0.8164	0.9453	0.8673	0.9908
5	0.8472	0.7264	0.7877	0.7298	0.8073

6	0.8099	0.7378	0.8176	0.8953	0.8689
7	0.7798	0.7826	0.6012	0.7592	0.8098

③ Analysis of individual students' mastery of knowledge

When students take academic tests, they often receive the same scores and class rankings, but in reality, many students' answers are different. For example, in this experiment, students No. 8, No. 23, and No. 54 all received a final score of 30 points on the 10 multiple-choice questions, but the questions they answered incorrectly were different. Since each question assesses different knowledge points and the number of knowledge points varies, the three students' mastery of knowledge points also differs.

Individual student knowledge mastery is shown in Figure 3. It can be seen that Student No. 8 has poor mastery of Knowledge Points A1 and A3, failing to meet the mastery level, but has good mastery of other knowledge points. Student No. 23 has good mastery of all knowledge points except A8. Student 54 has a good mastery of all knowledge points except for knowledge point A7. If the probability of mastering a knowledge point is between 0.5 and 1, its mastery status is recorded as 1; otherwise, it is recorded as 0. Ultimately, students and teachers can use the knowledge point mastery status calculated by cognitive diagnosis to target their learning of knowledge points, thereby mastering all knowledge points.

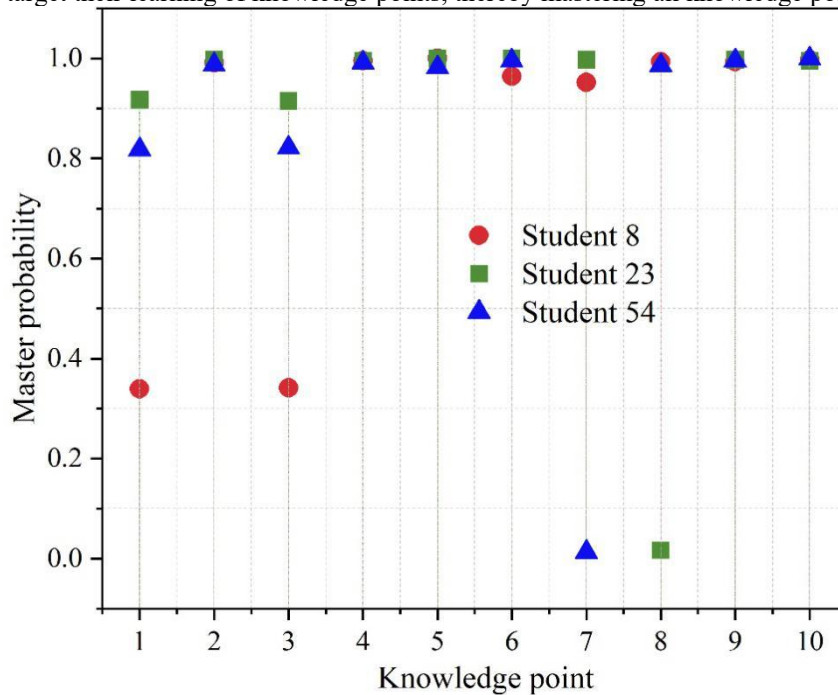


Figure 3. Students' knowledge of individual knowledge.

### 3.2.2. Personalized Test Question Recommendations and Case Analysis

Through the aforementioned experimental research, we obtained the predicted values for the next correct answers of students in seven classes and the specific probability values for their mastery of knowledge points. To achieve the ultimate goal of personalized question recommendations for students, this experiment combines the predicted values for the next correct answers with the specific values for knowledge point mastery, taking the union of the two. The specific question recommendation process and case analysis are as follows:

(1) Question Recommendation Process

In this experiment, the predicted values for a student's correct answers to questions were first calculated using a deep knowledge tracking model based on knowledge point embedding. Questions with predicted values between 0.5 and 1 were recommended to students as questions requiring reinforcement. Questions with predicted values between 0 and 0.5 are identified as questions requiring focused remediation. Finally, the questions recommended for remediation and reinforcement by the deep knowledge tracking model based on knowledge point embedding and the cognitive diagnostic DINA model are respectively taken as the union, and questions suitable for each student are recommended.

(2) Case Analysis

This section demonstrates the question recommendation results for two students using the knowledge point embedding-based learning diagnosis and personalized question recommendation method through actual case analysis. Table 10 shows the predicted values for Student 23. The predicted values for Questions 5 and 8, which Student 23 answered correctly, are relatively low, below 0.5, and these two questions are also recommended for the student.

**Table 10.** The prediction of student 23.

Question sequence	Predictive value
1	0.9089
1	0.9914
1	0.8974
1	0.9933
0	0.0035
1	0.7487
1	0.9783
0	0.3838
1	0.9517
1	0.8324

Based on the question recommendation system using a deep knowledge tracking model with knowledge point embedding, the cognitive diagnostic DINA model was further applied to provide question recommendations for Student 23. Table 11 shows Student 23's mastery of knowledge points. As shown in the table, Student 23 has not mastered Knowledge Point 10, but the mastery probabilities for all other knowledge points are very high, all exceeding 0.5, indicating that they have reached the mastery level.

By taking the union of the question recommendation results from both models, the final question recommendation list for Student No. 7 is obtained. Specifically, in the next phase, questions similar to Questions 1-4, 6, 7, and 9 should be recommended to Student No. 23 as consolidation questions, while questions similar to Questions 5, 8, and 10 should be recommended as targeted remedial questions. Additionally, teachers should also focus on strengthening their explanation of knowledge point 8 for Student 7 in future teaching processes.

**Table 11.** Student 23 knowledge point is in the situation.

Question sequence	Predictive value
A1	0.9086
A2	0.9976
A3	0.9112
A4	0.9317
A5	1.0000
A6	0.9995
A7	1.0000
A8	0.9431
A9	0.9967
A10	0.0084

#### 4. Conclusion

After establishing the overall process for constructing a course resource knowledge graph, this paper

proposes a knowledge tracking model based on feature and relationship enhancement. Using a learner-course bipartite graph as the foundation, a collaborative filtering course recommendation algorithm suitable for higher education self-study examinations is constructed.

The results show that the AUC and ACC values of the proposed model are 10.8% and 9.6% higher than those of the baseline model, respectively, and also significantly higher than those of the currently superior DKT-F model. The experimental results fully demonstrate that the proposed model outperforms existing models in predicting students' knowledge status. Additionally, by utilizing knowledge point embedding-based learning diagnosis and personalized question recommendation methods, we recommended questions tailored to learners' individual needs. This also helped educators better understand learners' knowledge levels, enabling them to adjust teaching strategies, while also helping learners understand their own knowledge levels, thereby facilitating targeted self-directed learning.

### Funding

This research was supported by the Education and Teaching Reform Project of Hainan Province in 2024: "Study on Influencing Factors and Countermeasures of College Students' Satisfaction with Online Course Resources" (No.: Hnjg2024-165); 2023 Hainan Higher Education Science Planning Research Project "Research and Practice of Online Course Quality Evaluation Method Based on Learner's Perspective" (No.: QJY20231068); The 2025 Hainan Province Higher Education Teaching Reform Research Project: "Research on the Influencing Factors of Teaching Interaction on College Students' Learning Engagement in the Online Learning Environment" (Hnjg2025-161).

### References

1. Shemshack, A., & Spector, J. M. (2020). A systematic literature review of personalized learning terms. *Smart Learning Environments*, 7(1), 33.
2. Du, L., Zhao, L., Xu, T., Wang, Y., Zu, W., Huang, X., ... & Wang, L. (2022). Blended learning vs traditional teaching: The potential of a novel teaching strategy in nursing education-a systematic review and meta-analysis. *Nurse education in practice*, 63, 103354.
3. Abu-Salih, B., & Alotaibi, S. (2024). A systematic literature review of knowledge graph construction and application in education. *Heliyon*, 10(3).
4. Cui, Y., Yu, H., Guo, X., Cao, H., & Wang, L. (2024). RAKCR: Reviews sentiment-aware based knowledge graph convolutional networks for Personalized Recommendation. *Expert Systems with Applications*, 248, 123403.
5. Yang, X., Huan, Z., Zhai, Y., & Lin, T. (2021). Research of personalized recommendation technology based on knowledge graphs. *Applied Sciences*, 11(15), 7104.
6. Qu, K., Li, K. C., Wong, B. T., Wu, M. M., & Liu, M. (2024). A survey of knowledge graph approaches and applications in education. *Electronics*, 13(13), 2537.
7. Hao, X., Ji, Z., Li, X., Yin, L., Liu, L., Sun, M., ... & Yang, R. (2021). Construction and application of a knowledge graph. *Remote Sensing*, 13(13), 2511
8. Liu, Q., Tian, Y., Zhou, T., Lyu, K., Wang, Z., Zheng, Y., ... & Li, J. (2023). An explainable and personalized cognitive reasoning model based on knowledge graph: Toward decision making for general practice. *IEEE journal of biomedical and health informatics*, 28(2), 707-718.
9. Shen, X., Li, X., Zhou, B., Jiang, Y., & Bao, J. (2023). Dynamic knowledge modeling and fusion method for custom apparel production process based on knowledge graph. *Advanced Engineering Informatics*, 55, 101880.
10. Ji, S., Pan, S., Cambria, E., Marttinen, P., & Yu, P. S. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2), 494-514.
11. Wei, Q., & Yao, X. (2022, January). Personalized recommendation of learning resources based on knowledge graph. In *2022 11th International Conference on Educational and Information Technology (ICEIT)* (pp. 46-50). IEEE.
12. Zhang, X., Liu, S., & Wang, H. (2023). Personalized learning path recommendation for e-learning based on knowledge graph and graph convolutional network. *International journal of software engineering and knowledge engineering*, 33(01), 109-131.
13. Liangkeyi, S. U. N. (2025). Integrating Knowledge Graphs and Causal Inference for AI-Driven Personalized Learning in Education. *Artificial Intelligence Education Studies*, 1(1), 64-81.
14. Zhang, J., Jing, S., & Liang, S. (2023, December). Research on personalized learning resource recommendation based on knowledge graph. In *2023 3rd International Conference on Electronic Information Engineering and Computer Communication (EIECC)* (pp. 111-118). IEEE.
15. Ni, Z., & Ni, F. (2020, December). Research on knowledge graph model of diversified online resources and personalized recommendation. In *Journal of Physics: Conference Series* (Vol. 1693, No. 1, p. 012178). IOP Publishing.
16. Yan, Z., Du, H., Lin, Z., & Jianhua, Z. (2023). Personalization exercise recommendation framework based on knowledge concept graph. *Computer Science and Information Systems*, 20(2), 857-878.
17. Martin, A. J., & Dominic, M. M. (2021). Personalization of learning objects according to the skill set of the learner using knowledge graph. *Turkish Journal of Computer and Mathematics Education*, 12(6), 3974-3987.

18. Troussas, C., Krouska, A., Tselenti, P., Kardaras, D. K., & Barbounaki, S. (2023). Enhancing personalized educational content recommendation through cosine similarity-based knowledge graphs and contextual signals. *Information*, 14(9), 505.
19. Xinjie Sun, Qi Liu, Kai Zhang, Shuanghong Shen, Lina Yang & Hui Li. (2025). Harnessing code domain insights: Enhancing programming Knowledge Tracing with Large Language Models. *Knowledge-Based Systems*, 317, 113396-113396.
20. Huazheng Luo, Zhichang Zhang, Lingyun Cui, Ziqin Zhang & Yali Liang. (2024). An efficient state-aware Coarse-Fine-Grained model for Knowledge Tracing. *Knowledge-Based Systems*, 302, 112375-112375.
21. Tianyu Wang & Dong Ge. (2025). Research on Recommendation System of Online Chinese Learning Resources Based on Multiple Collaborative Filtering Algorithms (RSOCLR). *International Journal of Human-Computer Interaction*, 41(3), 1771-1781.
22. Kohli Himani, Agarwal Jyoti & Kumar Manoj. (2022). An improved method for text detection using Adam optimization algorithm. *Global Transitions Proceedings*, 3(1), 230-234.