

Research Article



AN QRHKT : An Enhanced Question Representation and Historical Performance Fusion Model for Knowledge Tracing

Zhao Liu¹, Zhonghua Cheng^{2*}, Bin Liu^{3*}, Enzhi Dong⁴

¹PLA Army Engineering University, China

²PLA Army Engineering University, China

³PLA Army Engineering University, China

⁴PLA Army Engineering University, China

*Corresponding Author: Zhonghua Cheng, Bin Liu

Abstract:

Knowledge tracing models student exercise records to dynamically assess their knowledge levels, providing targeted guidance for education and learning. Existing knowledge tracing models lack the utilization of global information in question representation and have not overcome the issue of long-term dependency on the sequence of student responses. To address these issues, we propose an Enhanced Question Representation and Historical Performance Fusion Knowledge Tracing model (QRHKT). First, we construct a heterogeneous graph structure by combining the inclusion relations between questions and concepts, transition relations between questions, and co-occurrence relations between concepts, and enhance question representations through graph neural network algorithms. To address the long-term dependency issue, we categorize the knowledge states into current knowledge states and historically important knowledge states from the perspective of the importance of a student's historical knowledge state. The current knowledge state is obtained through a long short-term memory network, and historically important knowledge states are generated by weighted aggregation based on importance coefficients. Finally, by integrating these two types of knowledge states with question representation, we predict the student's answering performance. Experimental results show that our proposed QRHKT achieves better knowledge tracing results on three public datasets.

Keywords: Knowledge tracing; Enhanced question representation; Historical performance fusion; Heterogeneous graph structure; Graph neural networks; Long short-term memory network; Educational data mining

Introduction

The integration of Intelligent Tutoring Systems (ITS) in online education has become increasingly important. Knowledge Tracing (KT), as a key task within these systems, has emerged as an effective application of artificial intelligence in the field of educational technology, attracting widespread

attention across various domains. Essentially, KT models use students' historical learning data to construct models of their knowledge states and predict their future learning performance. Figure 1 illustrates the basic task of knowledge tracing models.

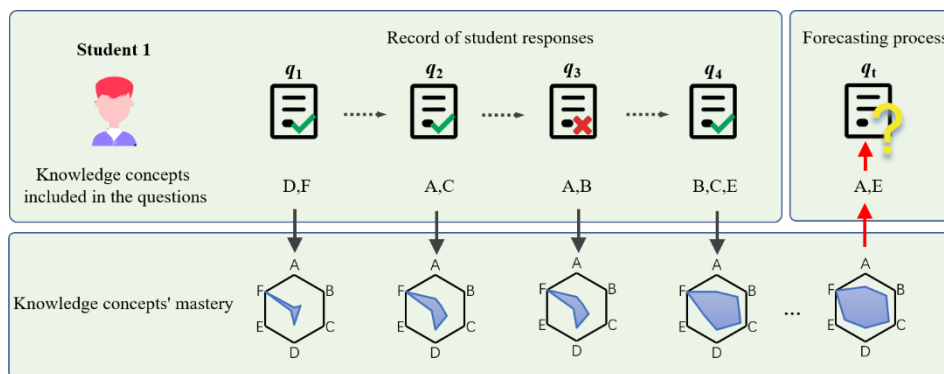


Figure 1 Basic tasks of the knowledge tracking model

Although existing KT models provide accuracy in diagnosing students' learning states, they still face several limitations. One major challenge lies in the representation of questions and concepts. Current datasets mainly contain inclusion relations between questions and concepts, lacking other useful information such as hierarchical relationships among concepts. Given the large number of questions, students may interact with each question relatively infrequently, making it challenging to learn effective question representations. Manually annotating hierarchical information of concepts by experts is time-consuming and labor-intensive. Previous studies have focused on the potential implicit relationships between questions and between concepts, but most research has concentrated on individual student response data, failing to fully utilize global information from all students' responses.

Moreover, learning is a gradual process where students improve their mastery of concepts through continuous practice. There are sequential dependencies in question sequences, where past response records influence current learning to varying degrees. Some records may have a significant impact, while others may have a lesser effect. Modeling these sequential dependencies is crucial for improving model performance. As the history of response records increases, the difficulty of modeling dependencies also increases. Some studies have attempted to address long-term dependencies in question sequences from the perspective of question similarity but have not differentiated the importance of response sequences.

Based on these insights, we propose the QRHKT model, an innovative knowledge tracing model

that improves knowledge tracing by enhancing question feature representation and historical state fusion. To enhance the learning of question representations, we designed a knowledge representation learning module based on a heterogeneous global graph neural network. This module utilizes global information from students' response sequences, calculates transition relations between questions and co-occurrence relations between concepts based on all students' sequence information, and constructs a heterogeneous global graph incorporating inclusion relations between questions and concepts. Through graph neural network algorithms, message passing, and aggregation, we learn the features of nodes in the graph. Additionally, to overcome the long-term dependency issue, we designed an adaptive knowledge state fusion module. We divide the student's knowledge states into current knowledge states and historically important knowledge states. The current knowledge state is obtained through a long short-term memory (LSTM) network, while historically important knowledge states are generated by weighted aggregation of the student's knowledge states at various historical time steps based on importance weights. Finally, the model adaptively combines current knowledge states and historically important knowledge states, and integrates question representations for the next prediction.

In summary, the contributions of this paper are as follows:

- (1) We designed a heterogeneous global graph question representation module. This module constructs a heterogeneous global graph through the inclusion and containment relations between questions and concepts, co-occurrence relations of concepts, and transition relations of questions, and

enhances the learning of question representations through graph neural network algorithms.

(2) We designed a historical state fusion module. This module adaptively learns the weights between current knowledge states and historically important knowledge states by calculating the importance coefficients of historical knowledge states, and determines the amount of information retained from different knowledge states to overcome the long-term dependency problem.

(3) We integrated the above two modules to predict students' future performance, resulting in the QRHKT knowledge tracing model. Experiments on three real-world datasets show that QRHKT outperforms baseline methods. Ablation experiments validate the effectiveness of each model component.

The organization of this paper is as follows. Section 2 reviews related work. Section 3 clarifies the problem definition and explains the notation. Section 4 presents our proposed model. Section 5 details the experiments and discussion. Section 6 demonstrates the model application and visualization. Finally, Section 7 concludes the paper.

2 Related Work

The issues of sparse question features and long-term dependencies are significant problems in knowledge tracing models, requiring researchers to focus on expanding and improving these models. We summarize the improvement methods as follows.

2.1 Knowledge Tracing Models with Enhanced Question Representations

Early knowledge tracing models used concepts as inputs, such as DKT[1], DSCMN[2], and DKVMN[3]. Since the number of concepts is much smaller than the number of questions, using concepts as inputs can effectively avoid over-parameterization and overfitting. However, this approach assumes that questions under the same concept are equivalent, neglecting fine-grained

information at the question level, which limits the model's performance.

To address this issue, researchers began using questions as inputs to capture the differences between questions. However, in some sparse datasets, the number of interactions between students and questions is low, making it difficult to effectively learn question representations and thus potentially leading to suboptimal performance. Subsequent research explored how to learn more effective question representations.

Some methods enrich question embeddings through the textual content of questions. For example, Su et al. [4] used Word2vec [5] and Bi-LSTM to learn question information from text; Pandey et al. [6] used word embedding techniques to convert words into fixed-dimension vectors and used SIF (Smooth Inverse Frequency)[7] to generate semantic information of questions; Tong et al.[8] used BERT[9] to obtain semantic information of questions. However, these methods have limitations in terms of high computational resource consumption and difficulty in obtaining question text content.

Another common approach is to use the relationships between questions and concepts to learn question representations. For instance, Ghosh et al. [10] adopted the Rasch model [11] from psychometrics to construct embeddings of concepts and questions, demonstrating significant performance improvements through ablation experiments. Wei et al. [12] combined concepts, difficulty levels, and unique features of questions to learn question representations. Additionally, there are inherent structural relationships between questions and concepts, forming a bipartite graph. A question's first-order neighboring nodes are the concepts it contains, and second-order neighboring nodes are questions sharing the same concepts. A graph structure can be used to mine relationships between questions and concepts as well as between questions themselves (as shown in Figure 2).

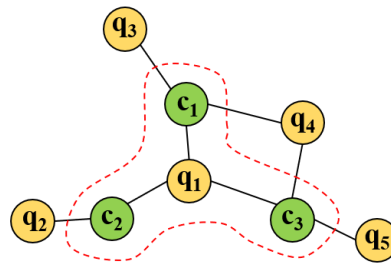


Figure 2 Relationship between and between topics and concepts

However, using only the inclusion relationships between questions and concepts may be insufficient. First, in many datasets, a question usually contains only one or two concepts, and in some datasets, each question corresponds one-to-one with a concept, making the graph structure relatively sparse and limiting information propagation. We found that most knowledge tracing methods are based on sequential models (such as RNNs and Transformers), which focus on individual students' historical responses, ignoring information sharing between students. Long *et al.* [13] proposed improving knowledge tracing through collaborative information among students, referencing the responses of other students with similar response records when predicting a student's performance.

Some research in the recommendation system field [14-15] also shows that, besides using the

current user's session, historical session patterns from other users can be referenced. Similarly, there are transition relationships between questions (as shown in Figure 3). For example, when predicting Student A's performance on Question q_5 , by analyzing the response records of Students B and C, we find that after correctly answering Question q_1 , they both correctly answered Question q_5 . For example, when predicting Student A's performance on Question q_5 , by analyzing the response records of Students B and C, we find that after correctly answering Question q_1 , they both correctly answered Question q_5 . Since Student A has already correctly answered Question q_1 , we can infer that Student A also has a high probability of answering Question q_5 correctly.

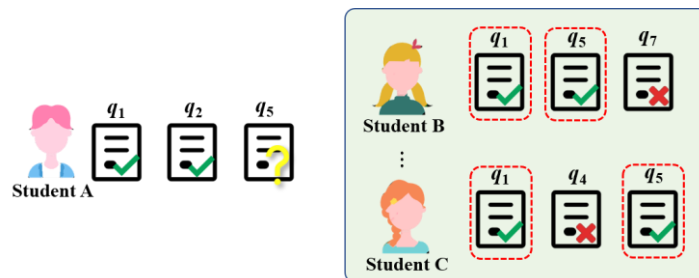


Figure 3 The transfer relationship between the questions

In terms of concepts, concepts themselves contain hierarchical information. For example, before mastering integration, one needs to successively master concepts such as sets, mappings, functions, and derivatives. However, current public datasets do not include hierarchical information of concepts, and manually annotating hierarchical information by experts is time-consuming and labor-intensive. Since students usually start with lower-level concepts and gradually master higher-level concepts during practice, exploring potential relationships between concepts solely through

students' response sequences remains to be researched.

2.2 Knowledge Tracing Models Improved for Long-term Dependencies

DKT and DKVMN are two representative models in the field of knowledge tracing, using RNN and MANN to address the knowledge tracing problem, respectively. DKVMN updates the knowledge state based on the most recent response records, making it difficult to capture long-term dependencies in question sequences.

Similarly, the RNN used in DKT also faces long-term dependency issues. Although variants such as LSTM and GRU improve sequence learning capabilities to some extent, they do not completely solve the problem.

Abdelrahman et al.[16] used Hop-LSTM to capture sequential dependencies between questions in students' response sequences by performing jump connections based on correlation to handle relevant response records. Sha et al.[17] enhanced long-term dependency capture using dual-layer LSTM and residual connections. Su et al.[4] calculated similarity between questions using cosine similarity and weighted aggregated all historical knowledge states based on this similarity. Yang et al.[18] designed a review module that selects the most relevant historical response records based on attention weights to reduce noise. Wang et al.[19] combined LSTM with an improved attention mechanism, integrating features such as time distance and the number of past attempts into the attention weight scores between questions. These methods address long-term dependency issues from the perspective of question similarity but do not effectively utilize students' historically important knowledge states. Certain historical knowledge states have a lasting

impact on subsequent predictions. For example, if a student correctly answered a derivative question at time t , since the concept of derivatives is relatively advanced, we can infer in subsequent predictions that the student is likely to correctly answer lower-level questions such as basic arithmetic and equations based on the knowledge state at time t .

3 Problem Definition

The task of QRHKT is to predict a student's performance on future questions based on their historical response data. This involves building a model to process and analyze the sequential data, extracting latent information to predict the next response. Specifically, a student's response record can be represented as:

$$\chi = ((q_1, r_1), (q_2, r_2), (q_3, r_3), \dots, (q_T, r_T)) \quad (1)$$

where q_i represents a question, and r_i is a binary variable (0 or 1) indicating whether the student answered question q_i correctly (0 for incorrect, 1 for correct). Given χ , knowledge tracing aims to predict whether the student will correctly answer q_{T+1} , i.e., compute the probability:

$$P(r_{T+1} = 1 | q_{T+1}, \chi) \quad (2)$$

Table 1 Some key terms in QRHKT

symbol	meaning
c	Knowledge concept
q	topic
r	Student's answer result (1 or 0)
h	The student's state of knowledge
G	Heterogeneous global maps
V	A collection of nodes for a heterogeneous global graph, which can be a topic or a knowledge concept
E	The set of edge relations of heterogeneous global graphs, i.e., the inclusion relationship between topics and concepts, the co-occurrence relationship between concepts, and the topic transfer relationship
r	Directed edges in the graph structure
ϕ	Relevance between topics
x	Embedding of questions
p	Embedding of knowledge concepts
\oplus	Splicing operations
σ	Sigmoid activate the function
\square	Hadama Mass

4 QRHKT Framework

This section introduces the knowledge tracing model based on a Heterogeneous Global Graph

Neural Network. The model architecture is shown in Figure 4 and can be divided into the following parts:

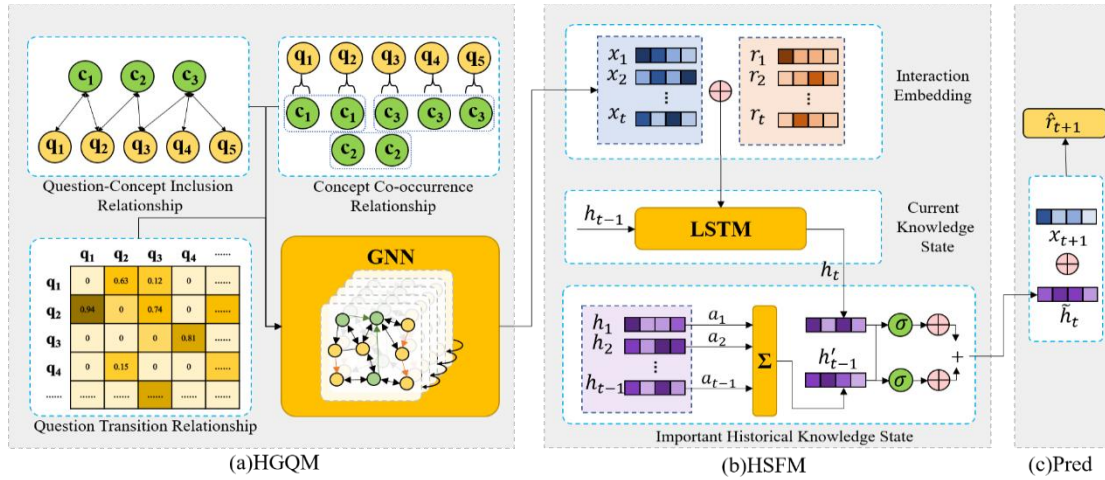


Figure 4 QRHKT framework

- **Heterogeneous Global Graph Question Representation Module (HGQM):** This module calculates the inclusion relationships between questions and concepts, co-occurrence relationships among concepts, and transition relationships between questions. It constructs a heterogeneous global graph based on these three relationships and uses graph neural network algorithms to learn the representations of questions.
- **Historical Significant Knowledge State Fusion Module (HFSM):** This module projects the learned question representations into an embedding space. It models the student's knowledge state at each time step using an LSTM, calculates the importance coefficients of historical knowledge states, and performs a weighted sum to obtain the significant historical knowledge state. An adaptive fusion module then combines the current knowledge state with the significant historical knowledge state.
- **Prediction Module (Pred):** This module concatenates the fused knowledge state with the question embedding and feeds it into a fully connected layer for prediction.

4.1 Heterogeneous Global Graph Question Representation Module (HGQM)

The HGQM module is based on a Heterogeneous Global Graph Neural Network for learning

question representations. Specifically, it leverages global information from student response sequences and computes the transition relationships between questions and the co-occurrence relationships between concepts, in conjunction with the inclusion relationships between questions and concepts, to construct a heterogeneous global graph. The Graph Neural Network algorithm is then used for message passing and aggregation to learn the features of the nodes in the graph.

The heterogeneous global graph is represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} includes question nodes $\mathcal{Q} = \{q_i\}_{i=1}^N$ and concept nodes $\mathcal{C} = \{c_i\}_{i=1}^M$. Here, q_i represents the i th question, c_i represents the i th concept, $N = |\mathcal{Q}|$ is the total number of questions involved in all student responses, and $M = |\mathcal{C}|$ is the total number of concepts covered by the questions. \mathcal{E} represents the different edge relationships in the graph, i.e., the inclusion relationships between questions and concepts, the co-occurrence relationships between concepts, and the transition relationships between questions. We will now describe how to construct these three relationships.

4.1.1 Question-Concept Inclusion Relationship

As illustrated in Figure 5, the graph consists of two types of nodes: question nodes and concept nodes. A question includes one or more concepts, and a concept is associated with multiple questions. This results in a bipartite graph

structure. Intuitively, a question is closely related to the concepts it contains, and questions that include the same concept should be more closely related. Therefore, we model the relationship

between questions and concepts as two types of directed edges in graph \mathcal{G} : $(c_j, q_i, r_{include})$ and $(q_i, c_j, r_{include_by})$, signifying that question q_i includes concept c_j .

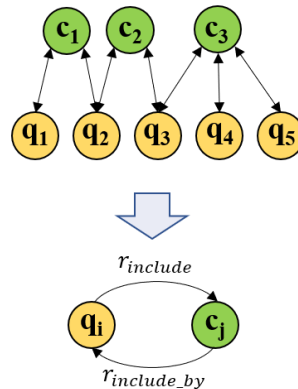


Figure 5 The inclusion relationship between the title and the concept and the corresponding directed edges

4.1.2 Concept Co-occurrence Relationship

Through our research on the data, we found that when students solve problems, they often solve multiple questions on a single concept consecutively before moving on to questions on another concept. As shown in Figure 6, from the question level, the student solved the following sequence of questions: $q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow q_4 \rightarrow q_5 \rightarrow q_6 \rightarrow q_7 \rightarrow q_8 \rightarrow q_9$. From the concept level, the student tackled questions corresponding to the following concepts in sequence: one question on concept c_1 , one on c_2 , one on c_3 , two on c_2 three on c_3 , and one on

c_4 .

Since the learning process is hierarchical, students need to start with lower-level concepts and gradually transition to more advanced ones. For example, if questions corresponding to concepts c_2 and c_3 appear repeatedly in the problem-solving sequence, it indicates a strong correlation between these concepts. In summary, frequent co-occurrence in different problem-solving sequences indicates a strong concept correlation. Therefore, we construct the relationships between concepts based on the co-occurrence information in all students' historical problem-solving records.

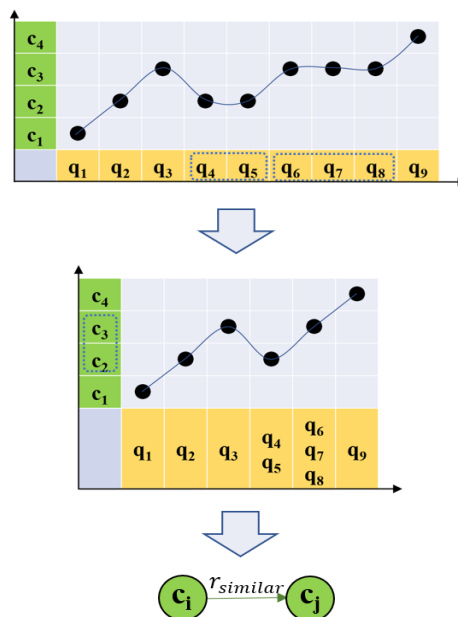


Figure 6 Conceptual co-occurrence relationship and corresponding directed edges

Specifically, we first merge consecutive concept problem-solving sequences to obtain the concept co-occurrence sequence S . For example, the sequence $\{c_1, c_2, c_3, c_2, c_2, c_3, c_3, c_3, c_4\}$ is merged into $\{c_1, c_2, c_3, c_2, c_3, c_4\}$ as shown in Figure 6.

Next, for concept c_i , we calculate its adjacent co-occurring concepts based on all historical concept co-occurrence sequences. The co-occurrence frequency $f_c(c_j, c_i)$ between concepts c_i and c_j is calculated using the following formula:

$$f_c(c_j, c_i) = \frac{\sum_{S \in N(c_j) \cap N(c_i)} \psi(S, c_j, c_i)}{\sqrt{|N(c_j)| |N(c_i)|}} \quad (3)$$

$$\psi(S, c_j, c_i) = \begin{cases} 1, & c_j \text{ is adjacent to } c_i \text{ in } S \text{ and } c_j \text{ is before } c_i \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Here, S is the merged concept co-occurrence sequence, and $N(c_j)$ is the collection of concept co-occurrence sequences in which concept c_j appears. Additionally, to avoid introducing too much noise, we only select the top k concepts with the highest co-occurrence frequency with concept c_i . If $f_c(c_j, c_i)$ is among the *top-k* co-occurrence frequencies for concept node c_i , we model the co-occurrence relationship between concepts c_i and c_j as a directed edge

$(c_j, c_i, r_{similar})$ in graph G .

4.1.3 Question Transition Relationship

In addition to the relationships between questions and concepts, and between concepts themselves, there are also correlations between questions. For example, many questions have transition relationships, where if a student correctly answers question q_1 , they are likely to also correctly answer question q_2 .

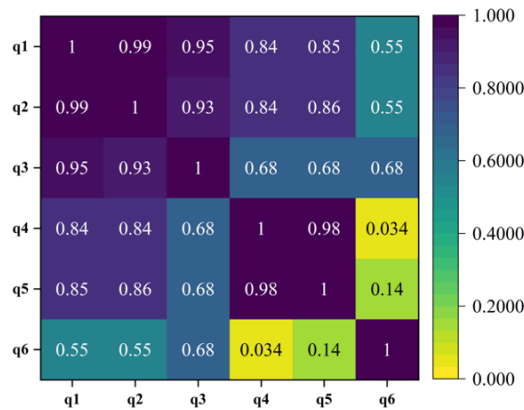


Figure 7 The transfer relationship and the corresponding directed edges

To explore this potential relationship between questions, we use the Phi coefficient to measure the correlation between two questions.

Specifically, we construct the question transition relationship table as shown in Table 2:

Table 2 Topic transfer relationship

$q_i \backslash q_j$	incorrect	correct	total
incorrect	n_{00}	n_{01}	n_{0*}
correct	n_{10}	n_{11}	n_{1*}
total	n_{*0}	n_{*1}	n

Based on the problem-solving records of all students, we count the occurrences of the pairs (incorrect, incorrect), (incorrect, correct), (correct, incorrect), and (correct, correct) between every pair of questions q_j and q_i . In the problem-solving sequence, q_j appears before q_i . If q_j appears multiple times before q_i , we only consider the most recent occurrence of q_j . Finally, we calculate their correlation using the following formula:

$$\phi_{i,j} = \frac{n_{11}n_{00} - n_{01}n_{10}}{\sqrt{n_{1*}n_{0*}n_{*1}n_{*0}}} \quad (5)$$

where the range of $\phi_{i,j}$ is $[-1,1]$. This value represents the extent to which a student's performance on question q_j affects their performance on question q_i . If $\phi_{i,j} > 0$, it indicates a positive correlation between the two questions; otherwise, it indicates a negative correlation.

Similarly, to avoid introducing too much noise, we set a threshold θ . When $\phi_{i,j} > \theta$, we model the transition relationship between questions q_i and q_j as a directed edge $(q_j, q_i, r_{transfer})$ in graph G .

4.1.4 Question Representation Aggregation

Inspired by [21] and [22], we use Graph Neural Networks (GNNs) to encode the nodes in the graph structure $G=(V,E)$. According to the definition of the graph G in the previous section, for question nodes, there are two types of connecting edges, $r_{include}$ and $r_{transfer}$; for concept nodes, there are also two types of connecting edges, $r_{include_by}$ and $r_{similar}$. In the GNN layer, the node features are updated by aggregating the

features of neighboring nodes and passing messages along the edges.

First, we use the question and concept IDs to obtain the initial embeddings $x_i^{(0)}$ and $p_i^{(0)}$ for questions and concepts, respectively. Here, $x_i^{(l)}$ and $p_i^{(l)}$ represent the node embeddings of questions and concepts after l layers of propagation. Specifically, $x_i^{(l)}$ is the embedding vector for question q_i , and $p_i^{(l)}$ is the embedding vector for concept c_i .

For question q_i , let $N_{r_x}(q_i)$ denote its neighboring nodes with edge type r_x . The aggregation formula is as follows:

$$m_{N_{r_x}(q_i)}^{(l+1)} = \frac{1}{|N_{r_x}(q_i)|} \sum_{n \in N_{r_x}(q_i)} e_n^{(l)} \quad (6)$$

$$x_{i,r_x}^{(l+1)} = f(W_{r_x}^{(l+1)}[m_{N_{r_x}(q_i)}^{(l+1)} \parallel x_i^{(l)}] + b_{r_x}^{(l+1)}) \quad (7)$$

where \parallel denotes the concatenation operation, $W_{r_x}^{(l+1)}$ is the weight matrix, $b_{r_x}^{(l+1)}$ is the bias vector, and $f(\cdot)$ is the activation function. Here, we use the ReLU activation function, and $e_n^{(l)}$ can be either $x_n^{(l)}$ or $p_n^{(l)}$.

Next, we accumulate the different messages propagated by different types of edges and update the representation of the question node using the following formula:

$$x_i^{(l+1)} = accum(x_{i,r_{include}}^{(l+1)}, x_{i,r_{transfer}}^{(l+1)}) \quad (8)$$

where $accum(\cdot)$ can be an operation such as summation or concatenation. In this paper, we use the mean operation.

The aggregation operation for concept nodes is similar to that for question nodes, and the updated concept node representation is as follows:

$$\bullet \quad p_i^{(l+1)} = \text{accum}\left(p_{i,r_{include}}, p_{i,r_{similar}}^{(l+1)}\right) \quad (9)$$

After L layers of propagation, we combine the embeddings from each layer to form the final representations of questions and concepts:

$$\bullet \quad x_i = \sum_{l=0}^L \alpha_l x_i^{(l)} \quad (10)$$

$$\bullet \quad p_i = \sum_{l=0}^L \alpha_l p_i^{(l)} \quad (11)$$

where α_l represents the importance of the output from the l -th layer of the GNN, $\alpha_l \geq 0$. In this paper, we set α_l to $1/(L+1)$.

4.2 Historical State Fusion Module (HSFM)

Certain historical knowledge states of students have a significant impact on future predictions. For example, correctly answering questions involving higher-level concepts often indicates an ability to answer questions involving lower-level concepts. We consider the knowledge state after correctly answering questions involving higher-level concepts as the important historical knowledge state. Based on this idea, we designed an adaptive knowledge state fusion module. Specifically, we divide the student's knowledge state into the current knowledge state and important historical knowledge states. The current knowledge state is obtained through an LSTM, while the important historical knowledge states are weighted and aggregated based on their importance from various historical time steps. Then, we adaptively combine the current knowledge state with the important historical knowledge states to influence the prediction for the next time step.

4.2.1 Interaction Embedding

After passing through multiple layers of graph neural networks, we obtain the representations of the questions. To capture the student's answering performance, we need to concatenate the question representation with the response. To emphasize the distinction between correct and incorrect answers, we encode the question-answer pair as follows, inspired by previous work [1]:

$$\bullet \quad a_t = \begin{cases} [x_t \oplus r_t], r_t = 1 \\ [r_t \oplus x_t], r_t = 0 \end{cases} \quad (12)$$

where \oplus denotes the concatenation operation, x_t is the embedding vector of question q_t , and r_t is an all-zero or all-one vector of the same dimension as x_t .

4.2.2 Current Knowledge State

The question-answer pair embeddings are input to the LSTM to calculate the student's knowledge state at each time step:

$$\bullet \quad h_t = \text{LSTM}(a_t, h_{t-1}) \quad (13)$$

4.2.3 Important Historical Knowledge State

At time t , we obtain the current knowledge state h_t and a series of historical knowledge states $\{h_1, h_2, h_3, \dots, h_{t-1}\}$. Considering that each historical knowledge state has a different degree of influence on the current prediction, we first calculate the importance coefficient for each historical knowledge state:

$$\bullet \quad \alpha_i = Z^T \tanh(W^T h_i + b) \quad (14)$$

$$\bullet \quad \alpha_i = \frac{\exp(\alpha_i)}{\sum_j \exp(\alpha_j)} \quad (15)$$

where \tanh is the tanh activation function, W is the weight matrix, b is the bias vector, Z is a learnable weight parameter, and α_i is the importance coefficient of the historical knowledge state h_i . Next, we obtain the important historical knowledge state by weighted summation:

$$\bullet \quad h'_{t-1} = \sum_{i=1}^{t-1} h_{att_i} \quad (16)$$

$$\bullet \quad h_{att_i} = \alpha_i h_i \quad (17)$$

Finally, we design an adaptive knowledge state fusion module that adaptively learns the weights between the current knowledge state and the important historical knowledge state, determining the amount of information retained from different knowledge states:

$$\bullet \quad z_1 = \sigma\left(W_{z1}^T [h'_{t-1} \oplus h_t] + b_{z1}\right) \quad (18)$$

$$z_2 = \sigma\left(W_{z2}^T \left[h_{t-1}' \oplus h_t\right] + b_{z2}\right) \quad (19)$$

$$z_2 = \sigma\left(W_{z2}^T \left[h_{t-1}' \oplus h_t\right] + b_{z2}\right) \quad (20)$$

where σ is the Sigmoid activation function, \oplus is the concatenation operation, \square is the Hadamard product, W_{z1} , W_{z2} are weight matrices, b_{z1} , b_{z2} are bias vectors, and \tilde{h}_t is the fused knowledge state.

4.3 Prediction Module

In the prediction module, we concatenate the fused knowledge state \tilde{h}_t with the embedding of the next question x_{t+1} and input it into a fully connected network for prediction:

$$\hat{r}_{t+1} = \sigma\left(W_{out}^T \left[\tilde{h}_t \oplus x_{t+1}\right] + b_{out}\right) \quad (21)$$

where σ is the Sigmoid activation function, W_{out}^T is the weight matrix, b_{out} is the bias vector, and \hat{r}_{t+1} is the predicted probability of the student correctly answering the next question.

Finally, we train the model using the cross-entropy loss function:

$$L = -\sum_i (r_i \log \hat{r}_i) + (1 - r_i) \log (1 - \hat{r}_i) \quad (22)$$

5 Experiment

In the experimental task, we evaluated the proposed QRHKT model on three public real-world datasets. This section describes:

Our experiments are designed to answer the following questions.

RQ1: How effective is QRHKT in predicting learners' performance compared to existing sequential models and graph-based knowledge tracking models?

RQ2: How do the modules in QRHKT affect the performance of the model?

RQ3: How can I be sure that the heterogeneous global graph title characterization module achieves the optimal effect?

RQ4: How does the Heterogeneous Global Graph Title Representation module model the potential relationship between the topic and the knowledge point?

5.1 Datasets

In this paper, three widely used benchmark datasets are employed to evaluate the predictive performance of the QRHKT model, as shown in Table 3.

Table 3 Dataset

	ASSIST2009	ASSISTChaII	EdNet-KT1:
#students	4217	1708	784,309
#questions	17,751	3162	13,169
#concepts	110	102	188
#responses	110,224	942,814	131,417,236
Responses per student	29	551	168
Responses per concept	1002	855	448,523
Responses per question	7	298	9979

- **ASSIST2009**¹: This dataset comes from the ASSISTments online education platform, recording data from students using the ASSISTments platform during the 2009-2010 period.

- **ASSISTChall**²: Among all publicly available ASSISTments datasets, this one has the richest

descriptive information, collected from a public data mining competition held in 2017. It contains a total of 942,816 learning records and has the longest average learning sequence length.

- **EdNet-KT1**³: This dataset is a large-scale hierarchical dataset collected by Santa,

capturing various student activities. It includes 131,417,236 interactions from 784,309 students, making it the largest publicly available IES dataset to date. Additionally, Santa provides a total of 13,169 questions annotated with 293 concepts. This study uses the KT1 version.

5.2 Experimental Setup

Before the experiments, we removed data records with empty problem ID and kc fields. For model training, we used 5-fold cross-validation to find the optimal parameters. For each fold, 20% of the response records were used as the test dataset, 20% as the validation set, and 60% as the training set. The embedding dimensions for question nodes and concept nodes were both set to 64, the dimension of hidden vectors in the LSTM was set to 64, the number of layers in the graph neural network was 2, and all learnable parameters were optimized using Adam. The batch size was set to 64, and the learning rate was set to 0.005, which decreased with iterations. For student practice sequences of inconsistent lengths, the length of each input to the sequence model was fixed to 50 practices. All experiments were conducted on a 64-bit Ubuntu 20.04.5 LTS server equipped with an Intel® Xeon® Gold 6238R CPU @ 2.30 GHz and a 32 GB NVIDIA A100 GPU. All code implementations were executed using the PyTorch framework.

5.3 Baselines

To evaluate the effectiveness of QRHKT, we compared it with various baseline methods. To ensure fair comparisons, we used the same training, validation, and test sets as QRHKT. All these methods were adjusted to ensure optimal performance from their implementations.

5.3.1 Non-Graph-Based Models

- **DKT** [1]: The first deep learning-based knowledge tracing model, which represents students' knowledge states using hidden vectors in an RNN.

- **DKT+F** [23]: An extension of DKT that incorporates three features—time interval since the last learning session, time interval of learning the same time point, and the number of historical learnings of the same concept—to model students' forgetting behavior.
- **DKVMN** [3]: Introduces a dynamic key-value memory network that uses static and dynamic matrices to store the static embeddings of concepts and the students' mastery levels of various concepts, respectively.
- **EKT** [24]: Considers the relationship between each question and all concepts, using the semantic information of practices as input.
- **SAKT** [25]: Utilizes a self-attention mechanism to identify interactions related to the given concept from students' past question-answering interactions and predicts their knowledge mastery based on the performance on these related interactions.
- **AKT** [10]: Uses a new monotonic attention mechanism to link students' performance on the target question with their past question-answering performances.

5.3.2 Graph-Based Models

- **GKT** [26]: Learns the relationships between knowledge concepts and models knowledge tracing as a time-series node-level classification problem, trained using GCN.
- **GIKT** [18]: Utilizes graph convolutional networks (GCN) to aggregate question embeddings and concept embeddings.
- **SGKT** [27]: Uses a relational graph to model the relationships between question concepts, employing a gated graph neural network to derive the students' knowledge states from their responses. To ensure fairness in experiments, the tests did not utilize question attributes such as difficulty level, type, and average response time.

Table 4 Compare the network structures used by the models

class	model	Network structure
Non-graph-based structure	DKT	RNN
	DKT+F	RNN
	DKVMN	MANN
	EKT	LSTM

	SAKT	Transformer
	AKT	Self-attention, FFN
Based on the graph structure	GKT	GNN
	GIKT	GCN, RNN
	SGKT	GCN, Attention
Ours	QRHKT	GNN, LSTM

5.4 Comparative Experiments

The comparative experiments were conducted to address **RQ1**.

Table 5 Comparative experimental results AUC (%)

Category	Datasets			
	Methods	ASSIST2009	ASSISTchall	EdNet-KT1:
Sequential Modeling	DKT	75.25	72.77	69.31
	DKT+F	75.73	75.70	68.67
	DKVMN	79.84	72.68	67.72
	EKT	73.32	71.06	67.34
	SAKT	67.39	69.71	68.28
	AKT	75.19	74.21	66.57
Graph-based	GKT	73.84	71.41	69.87
	GIKT	78.13	72.77	71.04
	SGKT	77.97	73.31	70.88
Ours	QRHKT	82.77	80.54	73.60

As shown in Table 5, QRHKT achieves better performance than other baseline models across all three datasets: ASSIST2009, ASSISTchall, and EdNet-KT1. The AUC values are higher than those of the best-performing models in each dataset by 2.93%, 4.84%, and 2.56%, respectively. These results indicate that QRHKT enhances predictive performance by learning representations of questions and integrating students' historical states.

Furthermore, QRHKT shows a significant advantage in ASSISTchall, likely due to the richer practice records in this dataset, which allow QRHKT to construct a denser heterogeneous graph structure and better learn question representations. The graph-based models GKT, GIKT, and SGKT also perform better on ASSISTchall than on ASSIST2009 and EdNet-KT1, demonstrating the effectiveness of graph-based models in learning question representations. QRHKT's higher AUC values compared to other graph-based models might be attributed to the

historical state fusion module using LSTM structures. This hypothesis needs to be verified through ablation experiments. Lastly, the less pronounced improvement on the ASSIST2009 dataset could be due to the higher complexity of the knowledge structure, with more questions and a higher question-to-concept ratio.

5.5 Ablation Experiments

To address RQ2, we conducted ablation experiments.

The QRHKT model learns question representations through the heterogeneous global graph question representation module and integrates students' historical and current states through the historical state fusion module. To validate the effectiveness of these two modules, we performed ablation experiments on different components of QRHKT.

The heterogeneous global graph question representation module includes three types of relationships: the inclusion relationship between

questions and concepts, the co-occurrence relationship between concepts, and the transition relationship between questions. To verify the necessity of these relationships, the experimental groups were further subdivided as follows:

- **QRHKT-RCR**: Removes the co-occurrence relationship between concepts.
- **QRHKT-RQR**: Removes the transition relationship between questions.
- **QRHKT-RQC**: Uses only the inclusion relationship between questions and concepts.

- **QRHKT-RGG**: Removes the heterogeneous global graph question representation module.
- **QRHKT-RHP**: Removes the historical state fusion module, using only the current knowledge state for prediction.
- **QRHKT-RGGHP**: Removes both the heterogeneous global graph module and the historical state fusion module.

The results of the ablation experiment are shown in Table 6.

Table 6 AUC (%) of ablation assay results

model	HGQM			HSFM	ASSIST2009	ASSISTchall	EdNet-KT1:
	q-c	c-c	q-q				
1 QRHKT	●	●	●	●	82.77	80.54	73.60
2 QRHKT-RCR	●	○	●	●	77.40	77.48	69.36
3 QRHKT-RQR	●	●	○	●	77.28	77.36	69.23
4 QRHKT-RQC	●	○	○	●	77.14	77.25	69.15
5 QRHKT-RGG	○	○	○	●	76.88	77.13	69.11
6 QRHKT-RHP	●	●	●	○	79.21	78.84	71.76
7 QRHKT-RGGHP	○	○	○	○	76.09	75.87	68.77

From the results, we can observe:

- 1) The AUC of QRHKT-RGG decreased by 5.89%, 3.41%, and 4.49% on ASSIST2009, ASSISTchall, and EdNet-KT1, respectively, after removing the heterogeneous global graph question representation module. This indicates that the graph structure built using global information effectively learns question representations and significantly enhances model prediction performance. The higher performance of QRHKT-RHP compared to QRHKT-RGGHP also supports this conclusion.
- 2) QRHKT-RCR's performance decreased after removing the concept co-occurrence relationship, suggesting that this relationship improves model performance. The higher performance of QRHKT-RQR compared to QRHKT-RQC further supports this.
- 3) QRHKT-RQR's performance decreased after removing the question transition relationship, indicating that this relationship enhances model performance. The higher performance

of QRHKT-RCR compared to QRHKT-RQC also supports this.

- 4) QRHKT-RQC's performance is higher than QRHKT-RGG, indicating that the inclusion relationship between questions and concepts enhances model performance.
- 5) The AUC of QRHKT-RHP decreased by 3.56%, 5.07%, and 4.83% on ASSIST2009, ASSISTchall, and EdNet-KT1, respectively, after removing the historical state fusion module, indicating the significant role of incorporating important historical knowledge states in the model.

5.6 Hyperparameter θ Analysis

The hyperparameter θ analysis is primarily conducted to address RQ3.

When constructing the transition relationship between questions into the heterogeneous global graph, a suitable θ value needs to be set to filter out some noise data to avoid introducing too much noise. Therefore, multiple sets of experiments were conducted to find an appropriate θ . As

shown in Table 7, the optimal θ values for the ASSIST2009, ASSISTchall, and EdNet-KT1 datasets are all 0.4. This result indicates that a smaller θ value introduces noise data that affects

model performance, while a larger θ value leads to insufficient information, which cannot effectively improve model performance.

Table 7 Hyperparameter θ analysis results

	ASSIST2009	ASSISTchall	EdNet-KT1:
$\theta=0.2$	80.20	78.59	71.93
$\theta=0.3$	81.13	79.42	72.89
$\theta=0.4$	82.77	80.54	73.60
$\theta=0.5$	81.92	79.61	72.81
$\theta=0.6$	81.23	78.43	71.13

To visually display the role of the hyperparameter θ , we selected eight different questions from the ASSISTchall dataset for analysis. The question IDs and their corresponding concepts are shown in Table 8. In Figure 8, it is clear that there are differences in the relevance values $\phi_{i,j}$ of different questions, indicating that the interaction representation module effectively represents different question nodes after training. When the

hyperparameter θ is set to 0.4, the relevance value $\phi_{i,j}$ between questions 1206 and 1207 is 0.42. In Table 8, it can be observed that questions 1206 and 1207 correspond to the same concept (perimeter), indicating that the heterogeneous graph neural network discovers deep relationships between nodes and concepts when θ is 0.4.

Table 8 The numbers of the eight items selected by ASSISTchall and their corresponding concepts

Question number	Correspondence concept	Question number	Correspondence concept
1119	Geometry attributes	1074	Read the diagram
1121	the inner corners and the sides of >3	1204	region
1090	Change rotation	1206	circumference
1092	Dot plots	1207	circumference

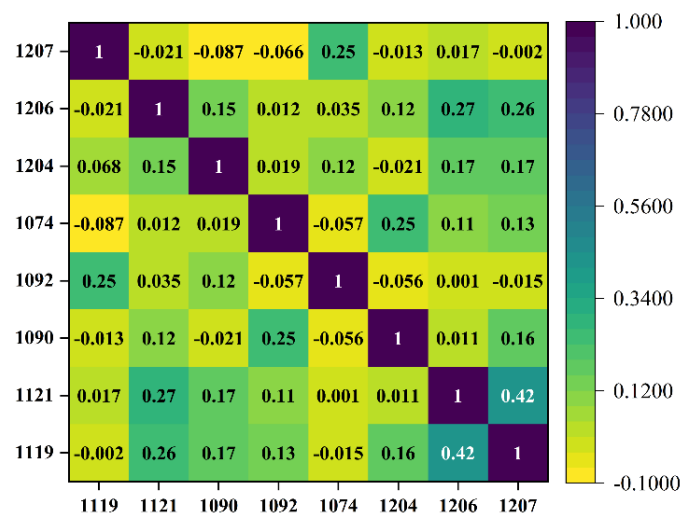


Figure 8 $\phi_{i,j}$ Differences in relevance values for different topics

5.7 Clustering Analysis

To address RQ4, this section mainly verifies whether the heterogeneous global graph question representation module has positively impacted the learning of question representations. Since GIKT and SGKT are also heterogeneous graph-based knowledge tracing models and both discuss question representations, we chose GIKT and SGKT as comparative models for the clustering experiment. The experiment labels question nodes using concept numbers, clusters question nodes using the K-Means algorithm, sets the number of clusters to the number of concept nodes in each dataset, and uses Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) as clustering evaluation metrics. To avoid experimental randomness, we conducted 10 repeated experiments and calculated the average results. Table 9 shows the clustering experiment results.

The NMI between variables (U) and (V) is calculated as:

$$NMI(U, V) = 2 \frac{MI(U, V)}{H(U) + H(V)} \quad (23)$$

where MI is the mutual information value between variables, and H is the entropy value of the variables.

The Rand Index (RI) is calculated as:

$$RI = \frac{TP + TN}{TP + FP + TN + FN} = \frac{TP + TN}{C_N^2} \quad (24)$$

The Adjusted Rand Index (ARI) is calculated as:

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} \quad (25)$$

From Table 9, it can be seen that QRHKT performs better in the clustering task. Its NMI is 0.38% to 6.21% higher than that of other models, and its ARI is 0.8% to 14.94% higher than that of other models. Compared with other baseline models, QRHKT's consideration of student-question interactions and students' answering ability results in better distinction capabilities of question nodes in terms of concepts. Therefore, incorporating student-question interactions and students' answering abilities positively influences the classification and grading of questions.

Table 9. Node clustering results

model	ASSIST2009		ASSISTchall		EdNet-KT1:	
	NMI	ARI	NMI	ARI	NMI	ARI
GIKT	68.55	44.32	68.62	43.47	44.69	15.07
SGKT	63.81	42.12	72.82	48.60	39.98	12.54
QRHKT	69.96	45.03	74.83	57.80	45.07	15.87

To more clearly demonstrate the clustering effect of question nodes, we visualized ten concepts from the ASSISTchall dataset and their associated questions. The feature vectors of question nodes learned by the three models were reduced in dimensionality using the t-SNE algorithm and projected into a two-dimensional space for visual comparison. For each question node representation vector, different colors were used

based on the concept number. From Figure 9, it can be seen that QRHKT can map questions of different concepts to different regions better than GIKT and SGKT models, with clearer boundaries between questions of different concept categories. Additionally, the overlap of question nodes within clusters of the same concept is low, better preserving the distinction between questions within the same concept.

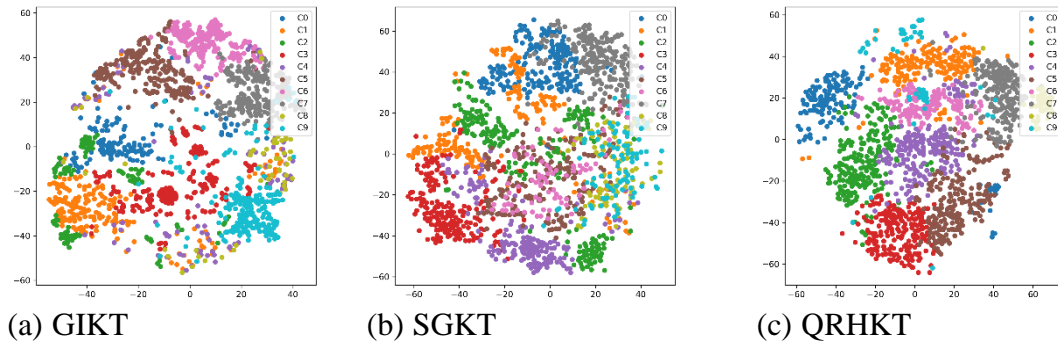


Figure 9 Visualization of the 10 concepts in the ASSISTchall data after t-SNE dimensionality reduction and the clustering of related questions

6 Visualization

To achieve the goal of personalized learner modeling, we studied the effectiveness of QRHKT in tracking knowledge states in terms of accuracy and rationality. Figure 10 shows a

visualized case of tracking knowledge states from the same learning sequence, obtained from an interaction segment of the ASSISTchall dataset. From Figure 10, we derive some important findings that can help build personalized learner profiles.

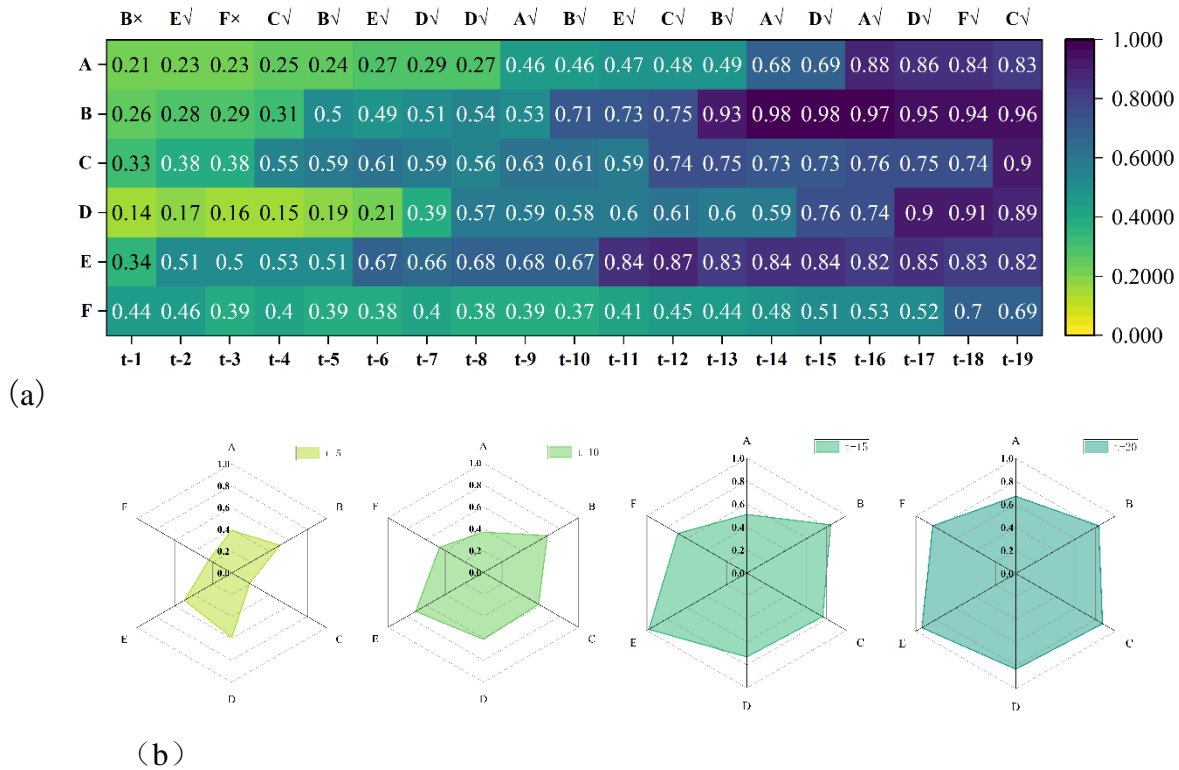


Figure 10. A visual example of a learner's knowledge state tracked by QRHKT using a learning sequence taken from the ASSISTChall dataset, in which the learner answers 20 questions on 6 concepts. In (a), the top of the heatmap shows the concepts contained in each question and the results of the students' answers, and the left side shows the 6 concepts from A to F. In (b), there are 4 radar charts, which are the students' knowledge status of the 6 concepts after completing the answers at 5, 10, 15, and 20 times

7 Conclusion

In this paper, we addressed the two limitations of

existing knowledge tracing models: the lack of utilization of global information for question representation and the challenge of overcoming

long-term dependence on students' answering sequences. To address these issues, we proposed a knowledge tracing model that enhances question representation and integrates historical performance.

First, we constructed a heterogeneous global graph combining the inclusion relationship between questions and concepts, the transition relationship between questions, and the co-occurrence relationship between concepts, and enhanced question representation through graph neural network algorithms. Then, to overcome the long-term dependence problem, we divided students' knowledge states into current knowledge states and historically important knowledge states. We obtained the current knowledge states through a long short-term memory network and produced historically important knowledge states by weighted aggregation based on importance coefficients. We used an adaptive fusion module to integrate current knowledge states and historically important knowledge states. Finally, we predicted students' future answering performance by combining knowledge states and question representations.

The comparative experiment results on three public datasets show that the proposed QRHKT achieves better predictive performance. Ablation experiments demonstrate the necessity and impact of the heterogeneous global graph question representation module and its three relationships, as well as the historical state fusion module, on overall performance. Clustering analysis results indicate that QRHKT positively impacts the representation of question nodes. Visualization results show that QRHKT can model accurate and reasonable knowledge states for learners.

In future research, we plan to explore more methods for learner modeling, including incorporating factors such as learning ability and forgetting factors into knowledge state modeling, and integrating textual information, image content, and hierarchical relationships of questions to enhance question feature representation.

Dataset linking

1

<https://sites.google.com/site/assistmentsdata/home/assistent-2009-2010-data>

2

<https://sites.google.com/view/assistmentsdatamining>

3

<https://github.com/bigdata-ustc/EduData?tab=readme-ov-file>

References

1. C. Piech et al., "Deep knowledge tracing," in Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, in NIPS'15. Cambridge, MA, USA: MIT Press, Dec. 2015, pp. 505–513.
2. S. Minn, Y. Yu, M. C. Desmarais, F. Zhu, and J.-J. Vie, "Deep Knowledge Tracing and Dynamic Student Classification for Knowledge Tracing," in 2018 IEEE International Conference on Data Mining (ICDM), Singapore: IEEE, Nov. 2018, pp. 1182–1187. doi: 10.1109/ICDM.2018.00156. Available: <https://ieeexplore.ieee.org/document/8594965/>. [Accessed: Jul. 02, 2024]
3. J. Zhang, X. Shi, I. King, and D.-Y. Yeung, "Dynamic Key-Value Memory Networks for Knowledge Tracing," in Proceedings of the 26th International Conference on World Wide Web, Perth Australia: International World Wide Web Conferences Steering Committee, Apr. 2017, pp. 765–774. doi: 10.1145/3038912.3052580. Available: <https://dl.acm.org/doi/10.1145/3038912.3052580>. [Accessed: Jul. 02, 2024]
4. Y. Su et al., "Exercise-enhanced sequential modeling for student performance prediction," in Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, in AAAI'18/IAAI'18/EAAI'18. New Orleans, Louisiana, USA: AAAI Press, Feb. 2018, pp. 2435–2443.
5. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, in NIPS'13. Red Hook, NY, USA: Curran Associates Inc., Dec. 2013, pp. 3111–3119.
6. S. Pandey and J. Srivastava, "RKT: Relation-Aware Self-Attention for Knowledge Tracing," in Proceedings of the 29th ACM

- International Conference on Information & Knowledge Management, Virtual Event Ireland: ACM, Oct. 2020, pp. 1205–1214. doi: 10.1145/3340531.3411994. Available: <https://dl.acm.org/doi/10.1145/3340531.3411994>. [Accessed: Jul. 02, 2024]
7. S. Arora, Y. Liang, and T. Ma, “A Simple but Tough-to-Beat Baseline for Sentence Embeddings,” presented at the International Conference on Learning Representations, Apr. 2017. Available: <https://www.semanticscholar.org/paper/A-Simple-but-Tough-to-Beat-Baseline-for-Sentence-Arora-Liang/3f1802d3f4f5f6d66875dac09112f978f12e1e1e>. [Accessed: Jul. 02, 2024]
 8. H. Tong, Z. Wang, Y. Zhou, S. Tong, W. Han, and Q. Liu, “Introducing Problem Schema with Hierarchical Exercise Graph for Knowledge Tracing,” in Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid Spain: ACM, Jul. 2022, pp. 405–415. doi: 10.1145/3477495.3532004. Available: <https://dl.acm.org/doi/10.1145/34774953532004>. [Accessed: Jul. 02, 2024]
 9. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423. Available: <https://aclanthology.org/N19-1423>. [Accessed: Jul. 02, 2024]
 10. A. Ghosh, N. Heffernan, and A. S. Lan, “Context-Aware Attentive Knowledge Tracing,” in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event CA USA: ACM, Aug. 2020, pp. 2330–2339. doi: 10.1145/3394486.3403282. Available: <https://dl.acm.org/doi/10.1145/3394486.3403282>. [Accessed: Jul. 02, 2024]
 11. L. Wolins, B. D. Wright, and G. Rasch, “Probabilistic Models for some Intelligence and Attainment Tests.,” *J Am Stat Assoc*, vol. 77, no. 377, p. 220, Mar. 1982, doi: 10.2307/2287805
 12. Wei S. et al. “Research on neural knowledge tracking methods incorporating generic topic representation learning,” *Chinese Journal of Information*, vol. 36, no. 4, pp. 146–155, Jun. 2022.
 13. T. Long et al., “Improving Knowledge Tracing with Collaborative Information,” in Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event AZ USA: ACM, Feb. 2022, pp. 599–607. doi: 10.1145/3488560.3498374. Available: <https://dl.acm.org/doi/10.1145/3488560.3498374>. [Accessed: Jul. 02, 2024]
 14. Z. Wang, W. Wei, G. Cong, X.-L. Li, X.-L. Mao, and M. Qiu, “Global Context Enhanced Graph Neural Networks for Session-based Recommendation,” in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event China: ACM, Jul. 2020, pp. 169–178. doi: 10.1145/3397271.3401142. Available: <https://dl.acm.org/doi/10.1145/3397271.3401142>. [Accessed: Jul. 02, 2024]
 15. Y. Pang et al., “Heterogeneous Global Graph Neural Networks for Personalized Session-based Recommendation,” in Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event AZ USA: ACM, Feb. 2022, pp. 775–783. doi: 10.1145/3488560.3498505. Available: <https://dl.acm.org/doi/10.1145/3488560.3498505>. [Accessed: Jul. 02, 2024]
 16. G. Abdelrahman and Q. Wang, “Knowledge Tracing with Sequential Key-Value Memory Networks,” in Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris France: ACM, Jul. 2019, pp. 175–184. doi: 10.1145/3331184.3331195. Available: <https://dl.acm.org/doi/10.1145/3331184.3331195>. [Accessed: Jul. 02, 2024]
 17. L. Sha and P. Hong, “Neural Knowledge Tracing,” in Brain Function Assessment in Learning, C. Frasson and G. Kostopoulos, Eds., in Lecture Notes in Computer Science, vol. 10512. Cham: Springer International Publishing, 2017, pp. 108–117. doi: 10.1007/978-3-319-67615-9_10. Available: http://link.springer.com/10.1007/978-3-319-67615-9_10.

- [Accessed: Jul. 02, 2024]
18. Y. Yang et al., “GIKT: A Graph-Based Interaction Model for Knowledge Tracing,” in *Machine Learning and Knowledge Discovery in Databases*, F. Hutter, K. Kersting, J. Lijffijt, and I. Valera, Eds., in *Lecture Notes in Computer Science*, vol. 12457. Cham: Springer International Publishing, 2021, pp. 299–315. doi: 10.1007/978-3-030-67658-2_18. Available: https://link.springer.com/10.1007/978-3-030-67658-2_18. [Accessed: Jul. 02, 2024]
 19. M. Wang, C. Peng, R. Yang, C. Wang, Y. Chen, and X. Yu, “GASKT: A Graph-Based Attentive Knowledge-Search Model for Knowledge Tracing,” in *Knowledge Science, Engineering and Management*, H. Qiu, C. Zhang, Z. Fei, M. Qiu, and S.-Y. Kung, Eds., in *Lecture Notes in Computer Science*, vol. 12815. Cham: Springer International Publishing, 2021, pp. 268–279. doi: 10.1007/978-3-030-82136-4_22. Available: https://link.springer.com/10.1007/978-3-030-82136-4_22. [Accessed: Jul. 02, 2024]
 20. K. Harbouche, C. Khentout, M. Djoudi, and A. Alti, “Measuring Similarity of Educational Items Using Data on Learners’ Performance and Behavioral Parameters: Application of New Models SCNN-Cosine and Fuzzy-Kappa,” *ISI*, vol. 28, no. 1, pp. 1–11, Feb. 2023, doi: 10.18280/isi.280101
 21. W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, in *NIPS’17*. Red Hook, NY, USA: Curran Associates Inc., Dec. 2017, pp. 1025–1035.
 22. M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, “Modeling Relational Data with Graph Convolutional Networks,” in *The Semantic Web*, A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, and M. Alam, Eds., in *Lecture Notes in Computer Science*, vol. 10843. Cham: Springer International Publishing, 2018, pp. 593–607. doi: 10.1007/978-3-319-93417-4_38. Available: https://link.springer.com/10.1007/978-3-319-93417-4_38. [Accessed: Jul. 02, 2024]
 23. K. Nagatani, Q. Zhang, M. Sato, Y.-Y. Chen, F. Chen, and T. Ohkuma, “Augmenting Knowledge Tracing by Considering Forgetting Behavior,” in *The World Wide Web Conference*, San Francisco CA USA: ACM, May 2019, pp. 3101–3107. doi: 10.1145/3308558.3313565. Available: <https://dl.acm.org/doi/10.1145/3308558.3313565>. [Accessed: Jul. 02, 2024]
 24. Q. Liu et al., “EKT: Exercise-Aware Knowledge Tracing for Student Performance Prediction,” *Ieee T Knowl Data En*, vol. 33, no. 1, pp. 100–115, Jan. 2021, doi: 10.1109/TKDE.2019.2924374
 25. S. Pandey and G. Karypis, “A Self Attentive model for Knowledge Tracing,” *ArXiv*, Jul. 2019, Available: <https://www.semanticscholar.org/paper/A-Self-Attentive-model-for-Knowledge-Tracing-Pandey-Karypis/0abdc7c7c6950a4a5bc6c93cf78d320d6c969152>. [Accessed: Jul. 02, 2024]
 26. H. Nakagawa, Y. Iwasawa, and Y. Matsuo, “Graph-based Knowledge Tracing: Modeling Student Proficiency Using Graph Neural Network,” in *IEEE/WIC/ACM International Conference on Web Intelligence*, Thessaloniki Greece: ACM, Oct. 2019, pp. 156–163. doi: 10.1145/3350546.3352513. Available: <https://dl.acm.org/doi/10.1145/3350546.3352513>. [Accessed: Jul. 02, 2024]
 27. Z. Wu, L. Huang, Q. Huang, C. Huang, and Y. Tang, “SGKT: Session graph-based knowledge tracing for student performance prediction,” *Expert Systems with Applications*, vol. 206, p. 117681, Nov. 2022, doi: 10.1016/j.eswa.2022.117681