IAENG International Journal of Applied Mathematics

A Knowledge Tracing Model Based on Concept Enhancement and Gating Mechanism

Qingsong Guo, Jun Wang, Xiaoyu Han, Zijie Li, Juxiang Zhou

***Abstract*—Existing knowledge tracing methods built upon dynamic key-value memory networks perform well, but they primarily focus on student-exercise interactions, failing to adequately represent students' true mastery, forgetting rates, and the physical significance of concepts in memory units. Therefore, we introduce a novel knowledge tracing method based on Concept Enhancement and Gating Mechanism (CEGM). Knowledge tracing assesses learners' knowledge degree and predicts students' academic outcomes by examining their historical interaction with intelligent tutoring systems. The method first introduces "knowledge absorption gate" and "knowledge updating gate" to better quantify students' knowledge mastery and forgetting level. Then, the joint embeddings of questions and associated knowledge concepts as inputs for the model, and employs a two-parameter logic model to enhance the conceptual representation and the model's interpretability. Finally, our research comparing the CEGM with 14 baseline models on six datasets show that the CEGM significantly outperforms the other models. In addition, an ablation study further validates the validity and soundness of the CEGM approach.**

***Index Terms*—deep learning, gating mechanism, item response theory, knowledge tracing, memory networks**

I. INTRODUCTION

WITH the emergence of large-scale online learning platforms, an abundance of educational data has been accumulated. Analyzing this data effectively is crucial for evaluating learners' knowledge levels and offering personalized guidance. Knowledge Tracing (KT) can analyze students' interaction data to measure their knowledge acquisition and forecast their upcoming performance. Scholars like Yu *et al*. [1] have used KT to reveal learners' potential knowledge states and make personalized recommendations. Fig.1. illustrates the KT process, showing a student's responses to six mathematical questions covering five knowledge concepts (kc_1-kc_5). Questions q_1, q_3, and q_5 are answered correctly, and q_2 and q_4 are answered incorrectly. The state of student mastery of the 5 knowledge concepts is represented by radar charts. KT analyses the responses to the previous questions for predicting the correct answer to the sixth question. Machine learning and deep learning are the two core technologies in the field of knowledge tracing.

The classic, traditional approaches to knowledge tracing encompass Bayesian Knowledge Tracing (BKT)[2] and Additive Factor Modelling (AFM)[3]. Corbett and Anderson introduced BKT in 1994, utilizing it to monitor students' knowledge acquisition as they engage in skill practice. Additionally, Hidden Markov Models (HMM)[4] and Bayesian Belief Networks(BBNs)[5] are probabilistic graphical models that are frequently employed in BKT. The AFM uses logistic regression to predict students' knowledge mastery grounded on multiple factors [6].

Recently, the Deep Knowledge Tracing (DKT) model has emerged[7], inspired by deep learning techniques[8]. Neural Network (RNN)[9] was first applied to knowledge tracing. Since then, different types of RNNs like Long Short-Term Memory Networks (LSTM)[10] and Gated Recurrent Units (GRU)[11] have been created and used a lot for predicting time series and recognizing patterns. For example, Huang[12] improved LSTM by combining the attention mechanism and constraint function to enhance the precision in prediction. Li *et al*.[13] introduced a pattern recognition method using LSTM, aimed at dealing with multimodal heterogeneous data fusion and feature learning, which improved the classification accuracy; Berradi *et al*.[14] conducted a comparative analysis of RNN, LSTM, and GRU based on mean square error and the hidden layer's node count. Because BKT and DKT did not adequately account for the relationship between individual knowledge points and questions. Zhang *et al*.[15] introduced the Dynamic Key-Value Memory Network (DKVMN) to track students' understanding of related knowledge concepts. DKVMN employs key-value matrices to store details about students' knowledge points and their mastery levels. Subsequently, many researchers have improved the DKVMN based on students' behavioral dominant factors, dependencies between knowledge points, and difficulty levels [16]-[18]. Despite the notable achievements in DKVMN modeling research, several unresolved issues remain:

Manuscript received May 27, 2024; revised December 23, 2024.

This work was supported in part by the National Natural Science Foundation of China under Grant 62266054, Yunnan Fundamental Research Projects under Grant 202401AT070122; Scientific research foundation of Yunnan Provincial Department of Education under Grant 2023Y0534; Scientific Research Foundation of Yunnan Provincial Department of Education under Grant 2023Y0535.

Qingsong Guo is a postgraduate student in the Key Laboratory of Education Informatization for Nationalities, Ministry of Education, Yunnan Normal University, Kunming 650500, China. (e-mail: 3080790461@qq.com).

Jun Wang is an associate researcher at Yunnan Normal University, Kunming 650500, China. (Corresponding author. e-mail: wangjun@ynnu.edu.cn).

Xiaoyu Han is a PhD student in the Yunnan Key Laboratory of Smart Education at Yunnan Normal University, Kunming 650500, China. (e-mail: 2387196509@user.ynnu.edu.cn).

Zijie Li is a PhD student in the Yunnan Key Laboratory of Smart Education at Yunnan Normal University, Kunming 650500, China. (e-mail: icaq@qq.com).

Juxiang Zhou is an associate professor at the Key Laboratory of Education Informatization for Nationalities, Ministry of Education and Yunnan Key Laboratory of Smart Education, Yunnan Normal University, Kunming 650500, China. (e-mail: zjuxiang@126.com).

ID	Knowledge Components(KCs)
kc_1	Integers
kc_2	Factors and Multiples
kc_3	Common Factors in Integers
kc_4	Prime Numbers
kc_5	Equations and Inequalities

KC illustrative examples

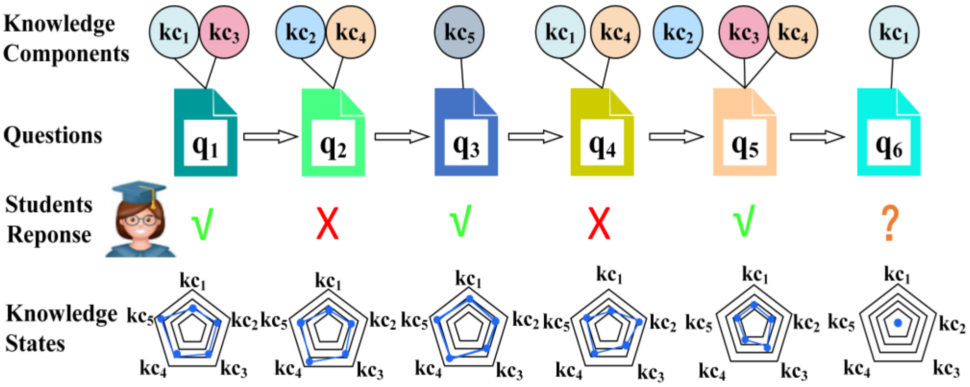


Fig.1. A descriptive example for the task of tracing knowledge. The diagram's left section details five knowledge concepts, and right section displays a student's reactions to those concepts and their corresponding knowledge states. Here, "✓" and "✗" represent the student's correct and incorrect responses, respectively.

First, the existing methods do not adequately consider the fact that students will choose different problem solving strategies according to their proficiency in the actual problem solving process, which will affect their levels of various knowledge points. Second, Second, the updating of the mastery of a single knowledge point cannot fully and accurately reflect the extent of students' forgotten knowledge over time. Third, the random initialization of knowledge concepts and mastery levels in the DKVMN model prevents the model from effectively capturing the intrinsic characteristics of each knowledge concept.

To respond to the aforementioned obstacles, this research proposes a KT method built on concept enhancement and gating mechanism. The method draws on the principles of the GRU and designs two gates to quantify students' knowledge acquisition and forgetting. Furthermore, In addition, by combining questions and knowledge concepts as inputs to the model and introducing a two-parameter logic model, not only the expressiveness of the knowledge concepts is enhanced, but also the interpretability of the model is improved.

This article's major contributions are highlighted below:

- We design the "knowledge absorption gate" and "knowledge update gate" to quantify students' actual mastery of knowledge and levels of knowledge forgetting, thereby more accurately reflecting their learning state.
- We utilize a combination of knowledge concepts and questions as the model's input, and employ a two-parameter logistic model to enhance the expressiveness of the knowledge concepts, and further enhancing the model's interpretability.
- We contrast the CEGM model against 13 benchmark models within the realm of deep knowledge tracing. The findings from the experiments indicate that the CEGM model significantly surpasses the majority of these standard models, thereby validating the efficacy and soundness of the approach.

II. Related Works

A. Problem Definition

Knowledge tracing works towards predicting students' future academic performance by analyzing their interaction data on learning platforms and modeling their current knowledge state. This process can be formalized as follows:

Suppose that the set of knowledge concepts $C=\{c_1, c_2, c_3, \ldots c_t\}$ contains the question set $Q=\{q_1, q_2, q_3, \ldots q_t\}$. Each question $q_t \in Q$ may present varying levels of difficulty. While a student is involved with the question set Q, an interaction sequence $X_t=\{x_1, x_2, x_3, \ldots x_{t-1}\}$ is generated. Here $x_t=(q_t, c_t, a_t)$ denotes the student's response to the question q_t at time t. $a_t=0$ denotes the wrong answer and $a_t=1$ denotes the correct answer. $P(a_{t+1}=1 \mid q_{t+1}, X_t)$ denotes the probability of the student giving a correct answer in the subsequent time point (t+1).

B. Deep Learning Knowledge Tracing

Over the past few years, deep learning methods have been more frequently employed in KT. At present, the predominant deep knowledge tracing models mainly belong to the subsequent classes:

1) **Knowledge tracing models based on RNNs:** DKT pioneered the use of RNNs in knowledge tracing models. Its subsequent improved versions include DKT+[19], DKT-DSC [20], and DKT-Forget [21]. Specifically, DKT+ enhanced the coherence of knowledge state prediction by merging a regularization component into the loss function; DKT-DSC used the K-Means algorithm to assess the learning level of students and dynamically assigns them to groups of similar ability; and DKT-Forget incorporated information about the forgetting mechanism into the DKT model. These improvements aim to capture students' learning process and knowledge acquisition more precisely.

2) **Knowledge tracing models based on attention mechanisms:** Attention mechanisms can efficiently process sequential data. AKT [22] first introduced monotonic attention methods into knowledge tracking models and achieved excellent prediction performance. Later, many scholars also made various optimizations rooted in the attention mechanism. As an example, Long *et al.*[23] proposed a collaborative multi-attention mechanism for peer sequence retrieval; Choi *et al.*[24] applied an attention mechanism in their encoding-decoding layer to process practice-response data; Pandey *et al.*[25] introduced a relationship-aware self-attention layer that integrates contextual information; Zhang *et al*.[26] designed a multiple attention mechanism to better model the students' learning and forgetting processes; and a sequential context-aware attention mechanism was proposed by Yin Wong *et al*.[27] , which is not dependent on the specific task. The introduction of attention mechanisms not only strengthens the ability of knowledge tracing models to capture important information, but also significantly improves the accuracy of prediction.

3) **Knowledge tracing models based on memory-enhanced networks:** Researchers have promoted the development of knowledge tracing models by optimizing external memory structures, influenced by memory augmentation networks. Both KVMN [28] and DKVMN use memory network matrix to represent students' knowledge states. Abdelrahman *et al*.[29] introduced Hop-LSTM to model the sequence of exercises. Deep-IRT model was proposed to augment the low interpretability of the DKVMN model. Chen *et al*.[30] introduced a prediction layer based on item response theory to generate interpretable prediction results. Tsutsumi *et al*. [31] further developed the learning process based on Deep-IRT by using separate networks for students and questions. Although these approaches improve the interpretability of the model, they are still inadequate in dealing with the inherent conceptual dependencies in learning.

4) **Knowledge tracing models based on graph neural networks:** Graphical representations have broad applications in knowledge tracing tasks to deal with a variety of complex knowledge structures. The graph knowledge tracing (GKT) [32] transformed knowledge structures into a graph format, which not only boosted the predictive capability of the model but also its interpretability. The structured knowledge tracing (SKT) [33] model further explores the intrinsic connections between concepts by digging deeper into the multi-layered relationships. Additionally, graph-based item knowledge tracing (GIKT) [34] and the problem-entity-based graph (PEBG) [35] models both utilize graph structures to characterize the correlations between problems or knowledge points. Zhang *et al*.[36] utilized gated heterogeneous graph convolutional networks to effectively model connections between students and knowledge points.

In summary, In the realm of KT research, deep learning approaches have yielded excellent results. However, knowledge Tracing approaches with Recurrent Neural Networks are susceptible to issues such as gradient explosion and vanishing when processing long sequences, which impedes their learning ability to effectively capture long-term dependencies. In contrast, knowledge tracing models that utilize attention mechanisms and memory augmentation networks exhibit limitations in terms of parameter interpretability. Furthermore, knowledge tracing models employing graph neural networks can enhance interpretability, but they also introduce increased complexity. Unlike other knowledge tracing models, the CEGM model integrates features from both the first and third types of knowledge tracing methods. This model is structured around two mechanisms: the knowledge absorption gate, which quantifies how well students grasp particular knowledge concepts for practical use, and the knowledge update gate, which evaluates the extent of forgetting of previously learned material. Additionally, the model's interpretability is improved by incorporating a combination of questions and related knowledge concepts as inputs, along with a two-parameter logistic model (2PLM) [37] to determine the possibility that students might respond future questions accurately.

III. CEGM MODEL ARCHITECTURE

The CEGM model's structure is exhibited in Fig.2., it primarily comprises two components: the performance predicting module and the knowledge updating module. The performance predicting module encompasses three elements: determining the correlation weight, assessing knowledge mastery, and utilizing item response theory alongside model optimization. The knowledge update module is responsible for dealing with the two processes of knowledge forgetting and knowledge growth.

A. Performance Prediction Module

1) Calculation of Correlation Weight

The question label q_t is first passed through a feature Embedding Matrix) $E \in R^{Q \times d_e}$ to get the question representation vector $e_t^q \in R^{d_e}$ (d_e signifies the dimension of e_t^q). Using Multi-hot encoding to process the knowledge concept label c_t . The $c_t \in \{0,1\}^C$ (C represents the count of knowledge concepts) corresponds to the question embedding vector e_t^q .The c_t reveals the association between the questions and each knowledge concept. The c_t passes through a linear layer to get the knowledge conceptual representation vector $e_t^c \in R^{d_e^c}$, where d_e^c represents hidden vector's dimension. The knowledge conceptual representation vector e_t^c is concatenated with the question embedding vector e_t^q , and then linearly transformed to get the integrated embedding vector $\mu_t \in R^{d_e}$. The above computational steps can be described in (1) and (2).

$$e_t^c = W_{e1} \bullet c_t + b_{e1} \tag{1}$$

$$\mu_t = W_{e2} \bullet [e_t^c, e_t^q] + b_{e2} \tag{2}$$

Where W_{e1} and W_{e2} are the weight coefficients, b_{e1} and b_{e2} reprensent the bias vectors, and $[\cdot,\cdot]$ denotes the concatenation of the two vectors. μ_t not only encompasses the semantic details of the question but also integrates the conceptual information pertaining to the relevant knowledge.

To quantify the degree of correlation between the questions and the underlying concepts. we perform an inner product operation on the integrated embedding vector μ_t and each key matrix M_i^e , and then compute the correlation weights using a fully connected layer that contains a softmax activation function. The specific expression is shown in (3).

$$\omega_t = softmax(M_i^e \bullet \mu_t) \tag{3}$$

Among them, M_i^e denotes the *i*-th row vector of the key matrix M_e ,which stores knowledge points and has a size of $N \times d_e$ (*N* indicates the quantity of key matrix). The softmax activation function converts a vector into probabilities, as displayed in (4).

$$softmax(x_i) = exp(x_i) / \sum_{j=1}^{K} exp(x_j) \tag{4}$$

In equation (4), x_i represents the *i*-th element in the input vector, *K* is used to denote the totality of categories, and *j* indicates the category's index.

2) Level of Knowledge Mastery

When students answer questions q_t, their mastery level of the knowledge concepts r_t can be determined by summing the relevant weights after multiplying them with the elements M_i^v of the value matrix M_t^v, see (5).

$$r_t = \sum_{i=1}^{N} \omega_t (M_i^v)^T \tag{5}$$

Where *N* signifies the collective number of value matrices, and *T* signifies the turn rank.

The knowledge concept summary vector v_t is derived from combining the student's knowledge proficiency r_t with the integrated embedding vector μ_t, which is then carried out by the fully connected layer with a activation function tanh. v_t incorporates the learner's proficiency in knowledge and the difficulty level of the previous exercises, as shown in (6).

$$v_t = tanh(W_v \bullet [r_t, \mu_t] + b_v) \tag{6}$$

Where W_v denotes the coefficient matrix, and b_v is the bias term. The expression for the activation function tanh is shown in (7).

$$tanh(x_i) = \frac{(exp(x_i) - exp(-x_i))}{(exp(x_i) + exp(-x_i))} \tag{7}$$

3) Item Response Theory

Although deep learning methods have demonstrated their power in improving the predictive performance of models, these models often suffer from the "black box" problem, i.e., their internal operating mechanisms are difficult to explain. The Item Response Theory (IRT) can well reveal the meaning expressed at the bottom of the model. Therefore, combining deep learning with IRT is ideal for improving the performance and interpretability of models. Nowadays, IRT has been extensively employed in the development of various models[38]-[39]. To intensify the understanding of knowledge concepts and represent their meanings more accurately in memory units, this research uses the powerful explanatory power of item response theory to predict students' final scores. Specifically, we use a two-parameter logistic model to evaluate the probability p_t that a student will answer the question without error at time t, as illustrated in (8).

$$p_t = \frac{1}{1 + exp^{-(\delta_t \theta_t - \varphi_t)}} \tag{8}$$

Where $\delta_t \in (0, \infty)$ is the item's discrimination parameter, which indicates the differentiation of the student's ability, $\theta_t \in (-\infty, \infty)$ denotes the student's ability parameter, and $\varphi_t \in (-\infty, \infty)$ is the item's difficulty parameter.

In the CEGM model, because the knowledge concept summary vector v_t is acquired by connecting and summing the knowledge mastery level r_t and the combined integrated embedding vector μ_t. Therefore, the knowledge concept summary vector v_t covers both the knowledge state information of the student in answering question q_t and the embedding information of question q_t. The student's ability characteristics θ_t can be calculated by passing the knowledge concept summary vector v_t through the neural network. After processing integrated embedding vector μ_t through a neural network, the item difficulty φ_t and item discrimination δ_t can be acquired. These calculations are described in (9-11).

$$\theta_t = tanh(w_\theta v_t + b_\theta) \tag{9}$$

$$\varphi_t = tanh(w_\varphi \mu_t + b_\varphi) \tag{10}$$

$$\delta_t = sigmoid(w_\delta \mu_t + b_\delta) \tag{11}$$

Where θ_t measures learners' knowledge acquisition, φ_t indicates concept difficulty, and δ_t represents ability differentiation at time t. The tanh and sigmoid functions are utilized in the neural network to map output values to the intervals (-1, 1) and (0, 1), respectively.

As can be seen from the formula, the difficulty and discrimination of the item are mainly affected by the combined knowledge point vector μ_t, and the students' ability characteristics are determined by the knowledge point summary vector v_t. Finally, we input the values of θ_t, φ_t, and δ_t into the IRT to determine p_t.

4) Model Optimization

The model is trained by parameters such as question embedding matrix E, question-answer embedding matrix B and key matrix M_e to derive the model's predictive capability. To validate the discrepancy among the model's predicted probability distribution and the actual labels, this study employs the Adam optimizer [40] to reduce the cross-entropy loss between the predicted values and the true values, thus optimizing the model's training. The expression method is described in (12).

$$L = -\sum_t (a_t \log(p_t) + (1-a_t)\log(1-p_t)) \tag{12}$$

Where L denotes the loss function, a_t denotes the true label and takes the value of 0 or 1, and p_t indicates the probability that the model predicts a positive class. The minus sign "-" is used to convert the value of the loss function to a positive value.

B. Knowledge Update Module

In this study, we design "knowledge absorption gates" and "knowledge update gate" to measure how well students remember and master specific knowledge concepts. Specifically, using the question-response $(q_t,\ a_t)$ interaction data to renew the learner's knowledge capability matrix M_t^v. According to (13), the interaction information o_t is obtained after the student has answered the question q_t. Then, the interaction information o_t is mapped into the question-response vector $s_t \in R^{d_v}$ through the embedding matrix $B \in R^{2Q \times d_v}$.

$$o_t = q_t + a_t \bullet c_t \tag{13}$$

In the actual question-answering process, students typically apply the mastered knowledge concepts to answer questions. By incorporating the question interaction vector s_t with the knowledge mastery state matrix $M_t^v = (m_1^v, m_2^v, m_3^v, ..., m_C^v)$ of each knowledge point, and going through a full connectivity layer to obtain the knowledge absorption gate $G_t \in R^{C \times d_v}$. G_t can measure the extent to which the students have practically applied each knowledge point. The expression is shown in (14-15).

$$S_t = concat(s_1, s_2, s_{3,} ..., s_t) \tag{14}$$

$$G_t = sigmoid(W_g \bullet [S_t + M_t^v] + b_g) \tag{15}$$

Where the model constructs the question interaction vector $S_t \in R^{C \times d_v}$ by concatenating N question interaction vectors s_t of the same dimensions. The G_t reflects the weights that students assign to different knowledge points when responding to questions relying on their individual knowledge acquisition.

The absorbed knowledge state $Q_t \in R^{C \times d_v}$. of the student at the current moment can be obtained from the product of the knowledge acquisition state matrix M_t^v and the knowledge absorption gate G_t . The question interaction vector S_t is concatenated with the learner's knowledge absorption state vector Q_t , and a linear transformation is used to compute the student's knowledge growth vector $\bar{M}_t^v \in R^{C \times d_v}$ .The specific formulas are exhibited in (16-17).

$$Q_t = G_t * M_t^v \tag{16}$$

$$\bar{M}_t^v = tanh(W_m \bullet [Q_t, S_t)] + b_m) \tag{17}$$

Where $W_m \in R^{2d_v \times d_v}$ is the learnable weight matrix that is used to linearly transform the inputs, and $b_m \in R^{d_v}$ is the bias vector to adjust the output after linear transformation.

During the acquisition of new knowledge, students often gradually lose recall of previously learned information, with the extent of forgetfulness differing among various knowledge points. Therefore, we design the knowledge update gate mechanism to evaluate the extent to which students have forgotten knowledge. The knowledge update gate $F_t \in R^{C \times d_v}$ can be obtained by adding the questions interaction vector S_t with the state m_i^v of every item of knowledge within the knowledge state matrix M_t^v through the fully connected layer. The expression is presented in (18).

$$F_t = sigmoid(W_f \bullet [S_t + M_t^v] + b_f) \tag{18}$$

The knowledge update gate F_t can distinguish the degree of forgetfulness for students' successively learned knowledge. By using the knowledge update gate F_t to update the students' knowledge mastery state, it is possible to obtain the students' new knowledge state M_{t+1}^v. The particular computation step is depicted in (19).

$$M_{t+1}^v = (1-F_t) * M_t^v + F_t * \bar{M}_t^v \tag{19}$$

Here, $M_{t+1}^v \in R^{C \times d_v}$ denotes the new knowledge mastery state, $(1-F_t) * M_t^v$ denotes the unforgotten part of the knowledge that the student has already mastered, and $F_t * \bar{M}_t^v$ denotes the unforgotten part of the new knowledge that the student has retained and demonstrated mastery of when answering the question at t time step. The M_{t+1}^v will be used in the next question-answering activity.

IV. EXPERIMENTS

For the purpose of verifying the CEGM model's effectiveness and rationality, this part will execute an experimental research to solve the following three key questions:

- **RQ1:** How does the CEGM model fare in comparison with the classical KT baseline models with regard to its predictive performance?
- **RQ2:** What are the effects of various parts(i.e., concept enhancement, forgetting mechanism, and item response theory) on the model's performance?
- **RQ3:** How does the embedding dimension (Embedding_dim) affect the CEGM model' performance?

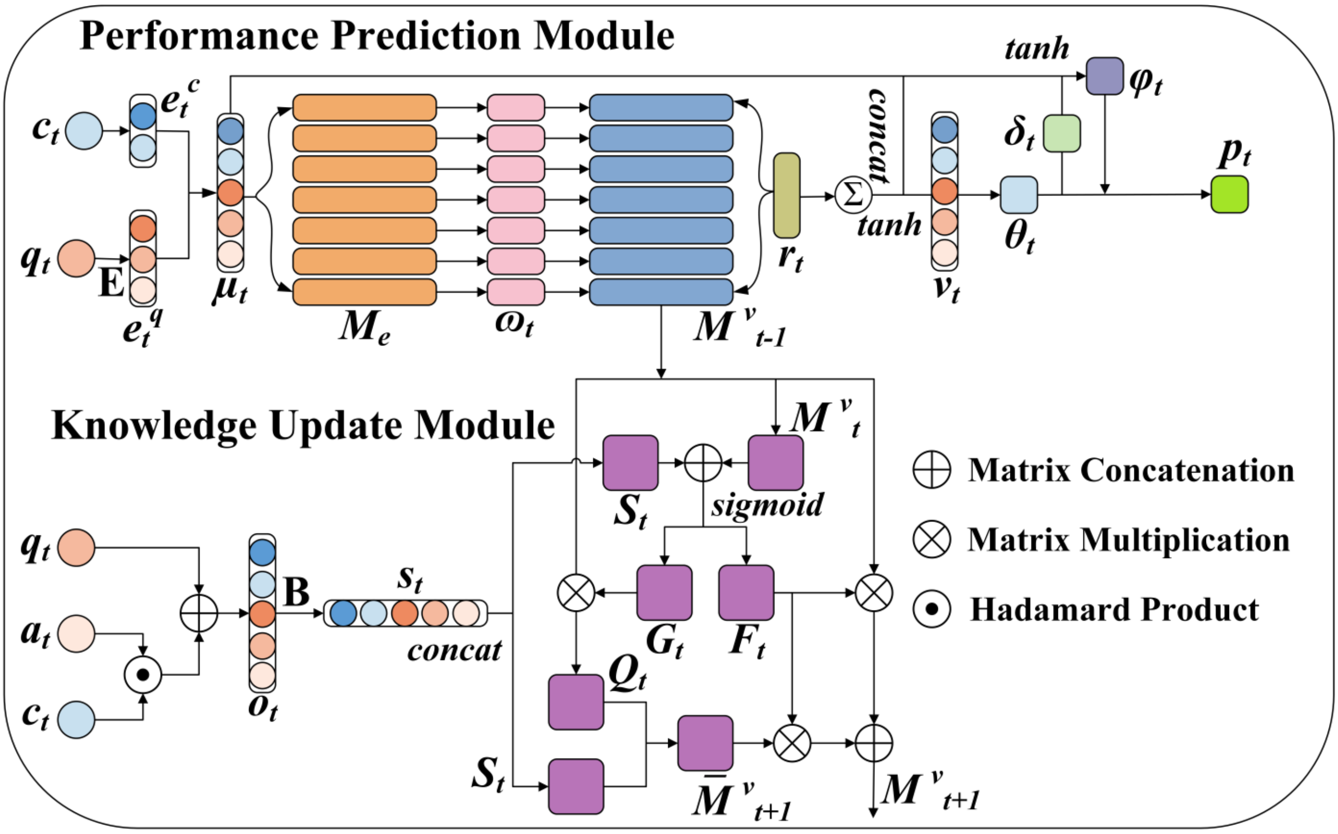


Fig.2. Framework diagram of the CEGM model.

A. Experimental Environment and Parameter Settings

The experiment utilizes the wandb (https://wandb.ai) tool integrated within the pyKT (https://pykt.org/) platform to optimize model parameters. pyKT is a Python library constructed using Pytorch and intended for training DKT models. The experimental environment setting is displayed in Table I.

TABLE I
EXPERIMENTAL ENVIRONMENT CONFIGURATION

Environmental name	Configuration information
Operating system	Windows 11, 64-bit
Development language	Python 3.9.0
Framework	Pytorch 1.11.0 + Cuda 11.1
CPU	12th Gen Intel(R) Core(TM) i7-1260P 2.10 GHz
GPU	NVIDIA GeForce RTX 3090
RAM	16.0GB

The experimental parameter settings are consistent with those in[41]. We use 80% of the dataset to train the model, and the leftover 20% acts as a validation set for evaluating the model performance. The dataset is randomly partitioned into five equally sized subsets, with four used for training and one for validation. We configure 200 epochs for training and apply an early stopping procedure to curb overfitting. The prediction layer has an embedding dimension of 64, and the hidden layer has a size of 128. The learning rates are specified as [1e-2, 1e-3, 1e-4, 1e-5], the dropout rates are set to [0.05, 0.1, 0.2, 0.3], the random seed options include [42, 3407], the sequence length is fixed at 200, the size of each batch is configured for 64, and the memory slot size is determined to be 50.

B. Datasets

To meet the needs of this experiment, we test the CEGM model's predictive performance on six publicly available datasets. For detailed information about the datasets, please refer to Table II. Table II shows the statistics of information after preprocessing for each dataset, including the number of interactions, sequence length, questions, and knowledge concepts. Where "-" indicates that the dataset Satics2011 only contains knowledge concepts, and ASSISTments2015 only contains questions.

- **ASSISTments2009(AS2009)：** The dataset originates from the online platform ASSISTments and mainly collects mathematical exercise[42]. It includes 4151 students, 337415 interactions, 4661 interaction sequences, 17737 questions, and 123 knowledge points.
- **Algebra2005(AL2005)：** The dataset is sourced from the 2010 KDD Cup EDM challenge, and comprises math questions collected from 2005 to 2006[43]. It includes information from 575 students, covering 1084 questions, 813661 interactions , and 112 knowledge concepts.
- **Bridge2006(BD2006)：** The dataset includes 1146 students who answered 19258 different questions, generating a total of 3686871 interaction records. These questions are further divided into 207790 sub-questions, each associated with one or more of 493 knowledge concepts.
- **NIPS34：** The dataset originates from the NeurIPS 2020 Education Challenge[44]. We choose Task 3 and Task 4 of the dataset to train the model. These two tasks include 1399470 interactions, 9401 sequences, 948 questions, and

57 knowledge concepts.

● **Statics2011(STA2011)**: This dataset is extracted during an engineering statics course at Carnegie Mellon University in the fall 2011[45]. It covers 189292 interactions, 1034 sequences, and 1223 different questions.

● **ASSISTments2015(AS2015)**: The dataset is collected in 2015 on the ASSISTments platform[46]. It contains 682,789 interactions from 19,917 students, 19,292 sequences, and 100 knowledge concepts.

TABLE II
INFORMATION STATICS FOR EACH DATASET

Datasets	Interactions	Sequences	Questions	Knowledge Concepts
AS2009	337,415	4,661	17,737	123
AL2005	884,098	4,712	173,113	112
BD2006	1,824,310	9,680	129,263	493
NIPS34	1,399,470	9,401	948	57
STA2011	189,292	1,034	1,223	-
AS2015	682,789	19,292	-	100

C. Evaluation Indicators

In our study, we consistently apply a 5-fold cross-validation method across all experiments to assess model performance. The metrics for evaluation reported consist of the mean Area Under the Curve (AUC)[47] and Accuracy (ACC)[48] scores. Here, ACC represents the samples' portion that the model has accurately predicted. It usually serves as an indicator of overall classification effectiveness. An elevated ACC signifies superior classifier performance. AUC, on the other hand, pertains to the region beneath the ROC curve, utilized to appraise the classification impact across various thresholds. An AUC of 50% suggests that the model's performance aligns with random guessing. Typically, AUC scores vary between 0.5 and 1, where higher numbers signify superior model performance.

D. Methods of Comparison

In this research, the performance of the proposed CEGM model is benchmarked against 13 exemplary KT baseline models. At the same time, a detailed introduction was provided for the main content of each model.

◆ **DKT**(Piech *et al.*, 2015)[7]: builds a model representing students' knowledge states through Recurrent Neural Networks (RNNs).

◆ **DKT+**(Yeung &Yeung, 2018)[19]: uses regular terms to improve the consistency of KT model.

◆ **DKT-F**(Nagatani *et al.*, 2019)[21]: adds forgetting mechanisms to predict students' performance.

◆ **KQN**(Lee & Yeung, 2019)[49]: predicts students' response scores by using dot product attention.

◆ **LPKT**(Shen *et al.*, 2021)[50]: predicts model by students' knowledge state and learning gains.

◆ **IEKT**(Long *et al.*, 2021)[51]: estimates the students' understanding of the question prior to predicting their response.

◆ **DKVMN**(Zhang *et al.*, 2017)[15]: uses static and dynamic matrices to store knowledge relationships and students' mastery respectively.

◆ **ATKT**(Guo *et al.*, 2021)[52]: uses adversarial raining methods to improve the robustness and generalization of deep learning knowledge tracing models.

◆ **GKT**(Nakagawa *et al.*, 2019)[32]: employs a graphical representation to map knowledge concept links, transforming the KT into a time-based node classification problem.

◆ **SAKT**(Pandey & Karypis, 2019)[53]: captures the connection between the KCs and the students' historical interactions by a self-attention network.

◆ **SAINT**(Choi *et al.*, 2020)[24]: applies deep self-attention layers separately for exercises and responses.

◆ **AKT(**Ghosh *et al.*, 2020**)**[22]: captures the temporal distances between questions and the previous interactions among students by attention mechanisms.

◆ **AT-DKT**(Liu *et al.*, 2023)[54]: enhances the predictive accuracy of the DKT model using question tag and personalized prior knowledge.

◆ **SimpleKT**(Liu et al., 2023)[41]: integrates the dot-product attention approach to discern the temporally relevant data within the student's learning engagements.

E. Experimental Results

1) Comparison Experiment(RQ1)

For the purpose of evaluating the CEGM model's effectiveness this study compares the CEGM model with 13 representative KT baseline models on six publicly available datasets. Table III and Table IV present the forecasting accuracy comparison results of the CEGM model and the traditional knowledge tracking benchmarks are evaluated with respect to AUC and ACC indicators, encompassing both the knowledge concept level (KC level) and the question level. It should be noted that, since the STA2011 dataset provides only information on knowledge points, and the AS2015 dataset includes only question information. Therefore, we only report the AUC and ACC values of STA2011 at the level of knowledge and AS2015 at the question level in all experiments. The experimental results for LPKT and SimpleKT are cited from[41], the results for AT-DKT come from[54], and the results for the remaining models are all taken from[55].

The '—' mark in the table indicates that the corresponding models do not report experimental results for that dataset. Additionally, the most outstanding outcomes are presented by using bold, and the second-best results are italicized. And ΔDKT illustrates the advancement in the CEGM model's performance against the DKT.

As is evident from the data displayed in Table III and Table IV, the CEGM model has achieved impressive performance on multiple datasets, both in terms of KC Level and Question Level. Specifically, from the perspective of KC Level, the performance on the AS2009, AL2005, NIPS34, and AS2015 datasets is particularly outstanding, with AUC values that are 3.62%, 2.69%, 1.82%, and 3.52% higher than those of the DKT model, respectively. This indicates that the CEGM is significantly better than the DKT model at predicting students' mastery of knowledge points.

TABLE III
AUC PERFORMANCE COMPARISION BETWEEN THE CEGMMODEL AND CLASSIC KT BASELINE MODELS

Models	KC Level(All-in-One)				Question Level(All-in-One)				STA2011	AS2015
	AS2009	AL2005	BD2006	NIPS34	AS2009	AL2005	BD2006	NIPS34		
DKT[7]	0.7419	*0.8146*	0.8013	0.7681	0.7541	0.8149	0.8015	0.7689	0.8222	0.7271
DKT+[19]	0.7424	0.8144	0.8019	0.7689	0.7547	0.8156	0.8020	0.7696	0.8279	*0.7285*
DKT-F[21]	—	0.8163	0.7984	0.7727	—	0.8147	0.7985	0.7733	0.7839	—
KQN[49]	0.7361	0.8005	0.7935	0.7677	0.7477	0.8027	0.7936	0.7684	0.8232	0.7254
DKVMN[15]	0.7330	0.7891	0.7981	0.7668	0.7473	0.8054	0.7983	0.7673	0.8093	0.7227
ATKT[52]	0.7337	0.7964	0.7885	0.7658	0.7470	0.7995	0.7889	0.7665	0.8055	0.7245
GKT[32]	0.7227	0.8025	0.8045	0.7681	0.7424	0.8110	0.8046	0.7689	0.8040	0.7258
SAKT[53]	0.7085	0.7682	0.7738	0.7516	0.7246	0.7880	0.7740	0.7517	0.7965	0.7114
SAINT[24]	0.6865	0.6662	0.7779	0.7860	0.6958	0.7775	0.7781	0.7873	0.7599	0.7026
AKT[22]	0.7650	0.8091	**0.8206**	*0.8017*	0.7853	0.8306	**0.8208**	0.8033	**0.8309**	0.7281
LPKT[50]	—	—	—	—	0.7814	0.8274	0.8055	0.8035	—	—
IEKT[51]	—	—	—	—	*0.7861*	**0.8416**	0.8125	*0.8045*	—	—
AT-DKT[54]	—	—	—	—	—	0.8246	0.8105	0.7816	—	—
SimpleKT[41]	—	—	—	—	0.7744	0.8254	0.8160	0.8035	0.8199	0.7248
CEGM(ours)	**0.7781**	**0.8315**	*0.8195*	**0.8033**	**0.7868**	*0.8357*	*0.8196*	**0.8049**	*0.8286*	**0.7320**
ΔDKT	3.62%	2.69%	1.82%	3.52%	3.27%	2.08%	1.81%	3.60%	0.64%	0.49%

TABLE IV
ACC PERFORMANCE COMPARISION BETWEEN THE CEGMMODEL AND CLASSIC KT BASELINE MODELS

Models	KC Level(All-in-One)				Question Level(All-in-One)				STA2011	AS2015
	AS2009	AL2005	BD2006	NIPS34	AS2009	AL2005	BD2006	NIPS34		
DKT[7]	0.7181	0.7882	0.8552	0.7028	0.7244	0.8097	0.8553	0.7032	0.7969	0.7503
DKT+[19]	0.7191	0.7889	0.8552	0.7034	0.7248	0.8097	0.8553	0.7039	0.7977	0.7510
DKT-F[21]	—	0.7891	0.8535	0.7071	—	0.8090	0.8536	0.7076	0.7872	—
KQN[49]	0.7179	0.7850	0.8532	0.7023	0.7228	0.8025	0.8532	0.7028	0.7978	0.7500
DKVMN[15]	0.7144	0.7778	0.8544	0.7013	0.7199	0.8027	0.8545	0.7016	0.7929	*0.7508*
ATKT[52]	0.7158	0.7774	0.8510	0.7010	0.7208	0.7998	0.8511	0.7013	0.7904	0.7494
GKT[32]	0.7077	0.7825	0.8554	0.7009	0.7153	0.8088	0.8555	0.7014	0.7902	0.7504
SAKT[53]	0.7017	0.7729	0.8460	0.6878	0.7063	0.7954	0.8461	0.6879	0.7879	0.7474
SAINT[24]	0.6885	0.7538	0.8410	0.7176	0.6936	0.7791	0.8411	0.7180	0.7682	0.7438
AKT[22]	*0.7323*	*0.7939*	**0.8586**	*0.7318*	*0.7392*	0.8124	**0.8587**	0.7323	*0.8021*	**0.7521**
LPKT[50]	—	—	—	—	0.7355	0.8145	0.8554	*0.7341*	—	—
IEKT[51]	—	—	—	—	0.7375	**0.8236**	0.8553	0.7330	—	—
AT-DKT[54]	—	—	—	—	—	0.8144	0.8560	0.7145	—	—
SimpleKT[41]	—	—	—	—	0.7320	0.8083	*0.8579*	0.7328	0.7957	*0.7508*
CEGM(ours)	**0.7441**	**0.8166**	*0.8559*	**0.7378**	**0.7489**	*0.8233*	0.8552	**0.7355**	**0.8037**	*0.7508*
ΔDKT	2.60%	4.84%	0.29%	3.50%	2.45%	1.36%	-0.00%	3.23%	1.68%	0.01%

The CEGM model performs optimally on the AS2009, AL2005, NIPS34, and STA2011 datasets in terms of ACC evaluation metrics, with an improvement of 1.18%, 2.27%, 0.60%, and 0.16% compared to the AKT model, respectively. Even on the BD2006 and AS2015 datasets, the performance of the CEGM model is only slightly inferior to that of the AKT model by 0.17% and 0.13%, showing strong competitiveness. This suggests that the CEGM model offers a significant advantage in forecasting the precision of students' responses. From the perspective of the question level, the CEGM model also shows superior performance. On the AS2009 and NIPS34 datasets, the AUC values of the CEGM model are 3.27% and 3.60% higher than those of the DKT model, and the ACC values are also improved by 2.45% and 3.23% respectively. Although the CEGM model do not achieve the highest scores on the AL2005 and BD2006 datasets, the gap with the strongest competitor is very small.

In summary, Both the AUC and ACC metrics of the CEGM model outperformed or approached those of the best baseline model across multiple datasets, showing significant improvement in almost all cases. Although the AKT baseline model is slightly better on the BD2006, STA2011, and AS2015 datasets, the main reason is that the AKT employs a monotonic attention mechanism for detecting short-term dependencies across various time scales. and implicitly models problem difficulty via embeddings based on the Rasch model, leading to superior performance when both knowledge concept level (KC level) and the question level information are available.

However, the CEGM approach follows closely behind AKT in terms of AUC and ACC, and performs equally well or even better compared to best-in-class models such as AT-DKT and SimpleKT. The outcomes demonstrate that the CEGM model is not only proficient in capturing changes in students' knowledge states but also in accurately predicting student performance. Thus, despite being slightly inferior to AKT on some specific datasets, the CEGM model demonstrates its value as a novel knowledge tracing method through its effectiveness and superiority.

2) Ablation Study(RQ2)

To ascertain the distinct impacts of each element within the CEGM model on overall effectiveness, we utilize the AS2009 dataset and perform an ablation analysis, with findings detailed in Table V. Here, 'w/o' (an abbreviation for 'without') indicates that the corresponding component is removed from the CEGM model.

- **w/o Item Response Theory (IRT):** indicates that only the Concept Enhancement and Gating Mechanism are used.
- **w/o Concept Enhancement (CE):** indicates that only the Item Response Theory and the Gating Mechanism are used.
- **w/o Gating Mechanism (GM):** indicates that only the Item Response Theory and Concept Enhancement are used.
- **w/o IRT & CE & GM:** indicates the simultaneous removal of the Item Response Theory, Concept Enhancement, and Gating Mechanism.

TABLE V
ANALYSIS OF THE CONTRIBUTION OF DIFFERENT COMPONENTS OF THE CEGM MODEL ON THE DATASET AS2009

Models	KC Level (All-in-One)		Question Level (All-in-One)	
	AUC	ACC	AUC	ACC
CEGM	**0.7781**	**0.7441**	**0.7868**	**0.7489**
w/o IRT	0.7746	0.7388	0.7798	0.7361
w/o CE	0.7751	0.7386	0.7717	0.7359
w/o GM	0.7742	0.7317	0.7728	0.7379
w/o IRT & CE & GM	0.7330	0.7144	0.7473	0.7199

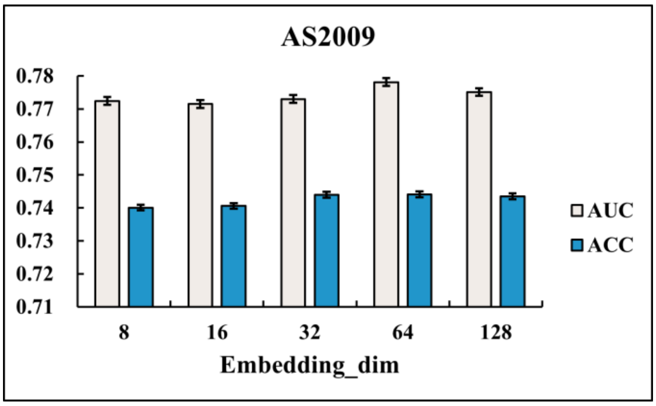
According to the data in the Table, each part of the CEGM contributes to its whole effectiveness. Specifically, the IRT component' removal leads to a slight diminution of the model's performance at the KC level, as indicated by a reduction of 0.35% in the AUC and 0.53% in the ACC. At the Question Level, the corresponding decreases in AUC and ACC are more substantial, at 0.70% and 1.28%, respectively. This highlights the important role of IRT in enhancing model accuracy.

Similarly, removing the CE component leads to a decrease of 0.30% in the AUC and 0.55% in the ACC at the KC level, whereas the declines at the question level are 1.51% for AUC and 1.30% for ACC, suggesting the importance of CE to model's performance. The absence of the GM component results in an AUC reduction of 0.39% and an ACC reduction of 1.24% at the KC level, with larger declines of 1.40% in AUC and 1.10% in ACC at the Question level, demonstrating the considerable impact of GM on model accuracy. When all three components, IRT, CE, and GM, are removed from the CEGM model, the degradation in performance becomes even more pronounced, with the AUC and ACC metrics at the KC level falling by 4.71% and 2.97%, respectively, and those at the question level declining by 3.95% and 2.90%, respectively.

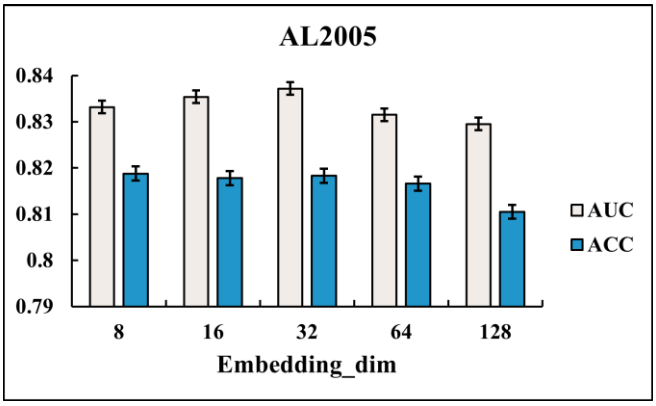
To sum up, the three components IRT, CE and GM are pivotal to the performance of the CEGM model at both the KC and Question levels. The integration of these three components significantly improves the model's predictive quality, with the GM having the greatest impact on accuracy. It is worth noting that the simultaneous removal of all components results in a substantial drop in performance, which validates the effectiveness and practicality of the CEGM model.

F. Hyperparameter Experiment(RQ3)

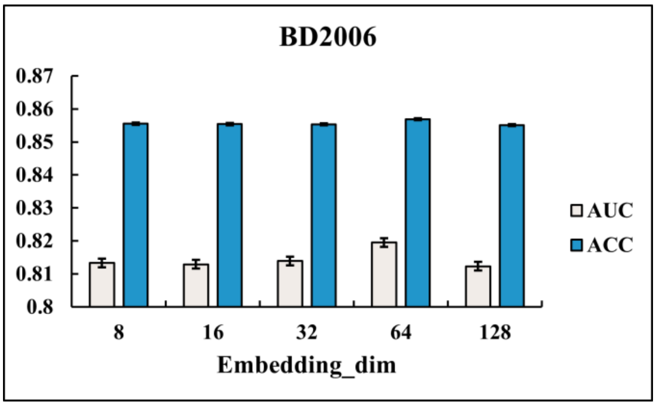
To investigate how embedding dimensions influence the CEGM method's performance, we conduct experiments using four dimensions: 8, 16, 64, and 128, ensuring study fairness and reliability. Across six datasets, we analyze and visualize AUC and ACC values for these dimensions, as illustrated in Fig.3. We also evaluate model prediction stability by examining the standard errors of our experimental outcomes.



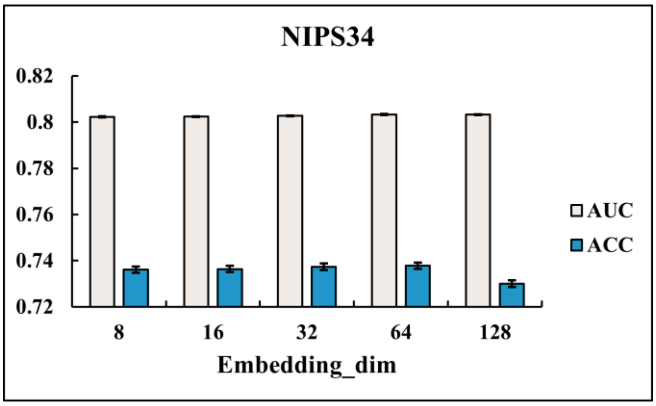
(a) AUC and ACC of the dataset AS2009



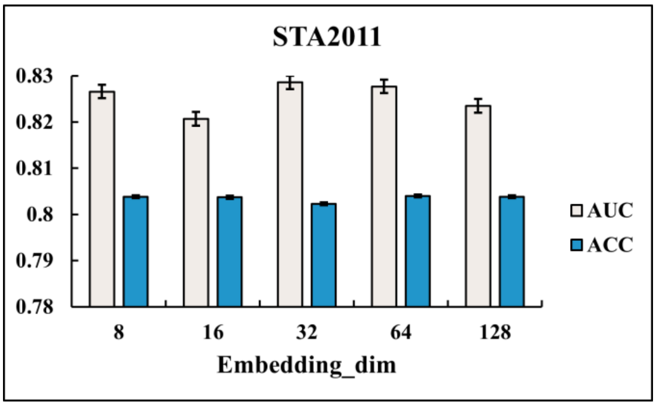
(b) AUC and ACC of the dataset AL2005



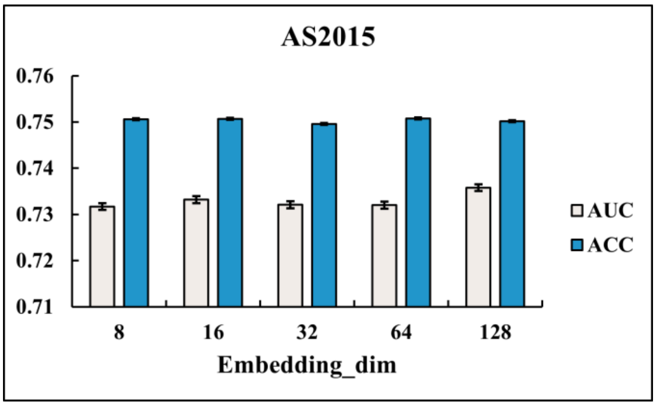
(c) AUC and ACC of the dataset BD2006



(d) AUC and ACC of the dataset NIPS34



(e) AUC and ACC of the dataset STA2011



(f) AUC and ACC of the dataset AS2015

Fig.3. Performance of CEGM model on four datasets across various dimensions. where (a-f) denote the AUC and ACC values on the six datasets.

From Fig.3., we can see that the model's performance fluctuates among various datasets because of differences in its embedding dimensions. With the growth of the embedding size, the AUC and ACC values of the CEGM model on the AS2009, AL2005, BD2006, and STA2011 datasets show an initial increase and then a decrease. In the NIPS34 dataset, although the AUC value does not change much, the ACC value also experiences an initial increase and then a decrease. It is worth noting that in the AS2015 dataset, both the AUC and ACC values have a continuous growth trend. This phenomenon indicates that different datasets display significant differences in the changes of the embedded dimensions of the model.

With an embedding dimension value of 64, the model will achieve high AUC and ACC values on the AS2009, BD2006, and NIPS34 datasets. Setting the feature dimension to 32, the model demonstrates the capability to attain high performance indicators on the AL2005 and STA2011 datasets. The AS2015 dataset obtains the best AUC and ACC performance with an embedding dimension of 128. These findings further demonstrate that the optimization of model performance is intimately connected to the qualities of a specific dataset, and that a particular embedding dimension cannot be applicable to all datasets.

From the perspective of performance stability, the length of the error line reflects the stability of the model's performance. A shorter error line equates to a more robust model performance. The model has high performance stability under different embedding dimensions on the six datasets, with a standout in the NIPS34 dataset, where the model performance is always close to the average.

In a word, the model perform better with embedding dimensions of 32, 64 and 128. Nevertheless, after considering factors such as computational resource consumption, model complexity, the risk of overfitting, and ensuring the consistency of the experimental design, we select 64 as selected as the embedding dimension in the model training process.

G.Knowledge State Visualization

In order to visually display the changing trend of students' knowledge mastery, we randomly select the interaction records of a student answering 30 questions from the AS2009 dataset for visualization and analysis, which reflects his or her mastery of five knowledge points (marked as kc_1 to kc_5). The particular outcomes are displayed in Fig.4. The top line of the graph corresponds to the knowledge areas associated with each exercise query, with various symbols used to differentiate among distinct knowledge areas. In this illustration, a student's correct answer is denoted by '1', while an incorrect response is denoted by '0'. The heatmap's core represents the student's proficiency in each knowledge area, with color intensity denoting the level of proficiency. A darker shade indicates the learners' proficiency in mastering the knowledge.

Before starting the exercise, the initial knowledge of the student is set to zero. This state is continuously updated as the exercises are completed. Correct answers will improve the mastery of related knowledge points, whereas incorrect answers may lead to a decline in mastery. For example, when the student correctly answers questions 6 and 7 involving "kc_5", the mastery level of "kc_5" displayed in the fifth row of the heat map will be improved accordingly.

Throughout the exercise, the mastery of a point is not only determined by itself, but is also influenced by the mastery of related points. For example, after answering questions 22 and 23 correctly in succession, the students' understanding of "kc_4" improves because these questions are related to one or more knowledge points.

As the exercises are completed, the level of mastery of each knowledge point improves. According to Fig.4., it can be seen that the student has a firm grasp of "kc_2", "kc_4" and "kc_5", and the mastery of "kc_1" and "kc_3" is relatively weak. With this kind of visual analysis, students can identify the knowledge points they have mastered and the parts that need further strengthening, thereby improve learning efficiency.

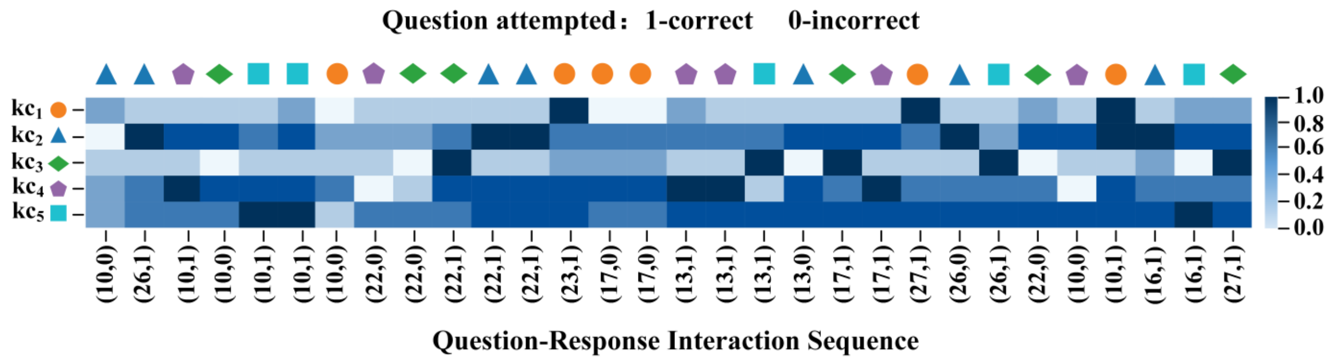


Fig.4. An example of a student's change in knowledge status when answering 30 questions containing 5 knowledge concepts. where the different shapes at the top and left indicate five different knowledge points.

V.CONCLUSION

This paper introduces a groundbreaking approach to knowledge tracing, i.e., the Concept Enhancement and Gating Mechanism-based Knowledge Tracing Model (CEGM). It effectively models students' level of knowledge and predicts their future academic performance. The method quantifies both the degree of knowledge acquisition and the rate of forgetting by designing a "knowledge absorption gate" and a "knowledge update gate", thereby capturing the dynamics of student learning with greater precision. Moreover, the technique includes details about questions and their linked knowledge concepts as inputs for the model, thereby improving the portrayal of these concepts. We also use a two-parameter logistic model to improve the model's explanatory power. Contrastive experiments across six public datasets demonstrate that the CEGM model achieves higher predictive accuracy compared to various benchmark models. Further ablation studies confirm the method's soundness and rationality. Even though the CEGM model has many advantages, there are still limitations: Fist, it has not yet fully considered the impact of factors such as solution time and number of attempts. Second the computational cost is relatively high due to its reliance on neural network structures. To overcome these constraints, forthcoming studies will concentrate on integrating acquired behavioral characteristics into the CEGM model to enhance its forecasting capabilities, and also on devising techniques to reduce computational load.

REFERENCES

[1] Simiao Yu, Ji Li, and Tiancheng Zhang, "Exercise Recommendation Algorithm Based on Reinforcement Learning," Engineering Letters, vol. 32, no. 10, pp1947-1956, 2024.

[2] Corbett A T, Anderson J R. Knowledge tracing: Modeling the acquisition of procedural knowledge[J]. User Modelling and User-Adapted Interaction, vol. 4, no. 4, pp253-278, 1995.

[3] Cen H, Koedinger K, Junker B. Comparing two IRT models for conjunctive skills[C]//Intelligent Tutoring Systems: 9th International Conference, ITS 2008, Montreal, Canada, June 23-27, 2008 Proceedings 9. Springer Berlin Heidelberg, pp796-798, 2008.

[4] Baker R S J, Corbett A T, Aleven V. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing[C]//Intelligent Tutoring Systems: 9th International Conference, ITS 2008, Montreal, Canada, June 23-27, 2008 Proceedings 9. Springer Berlin Heidelberg, pp406-415, 2008.

[5] Villano M. Probabilistic student models: Bayesian belief networks and knowledge space theory[C]//Intelligent Tutoring Systems: Second International Conference, ITS'92 Montréal, Canada, June 10-12 1992 Proceedings 2. Springer Berlin Heidelberg, pp491-498, 1992.

[6] Embretson, Susan E., and Steven P. Reise. Item response theory. Psychology Press, 2013.

[7] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J. Guibas, and Jascha SohlDickstein. Deep knowledge tracing. In NeurIPS. pp505-513, 2015.

[8] Jürgen Schmidhuber. Deep learning in neural networks: An overview. Neural Networks. Vol. 61, pp85-117, 2015.

[9] Zachary C. Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019, 2015.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural Computation, 1997.

[11] Cho K , Van Merrienboer B , Bahdanau D, et al. On the properties of neural machine translation: encoder-decoder approaches[C]//Eighth Workshop on Syntax, Sementics and Structure. Doha, Qatar: ACL, pp103-111, 2014.

[12] Huang Sen, "Time Series Prediction based on Improved Deep Learning," IAENG International Journal of Computer Science, vol. 49, no.4, pp1133-1138, 2022.

[13] Li Dongping, Yang Yingchun, Shen Shikai, He Jun, Shen Haoru, Yue Qiang, Hong Sunyan, and Deng Fei, "Research on Deep Learning Model of Multimodal Heterogeneous Data Based on LSTM," IAENG International Journal of Computer Science, vol. 49, no.4, pp1016-1022, 2022.

[14] Zahra Berradi, Mohamed Lazaar, Hicham Omara, and Oussama Mahboub, "Effect of Architecture in Recurrent Neural Network Applied on the Prediction of Stock Price," IAENG International Journal of Computer Science, vol. 47, no.3, pp436-441, 2020.

[15] Zhang J, Shi X, King I, et al. Dynamic key-value memory networks for knowledge tracing[C]//Proceedings of the 26th international conference on World Wide Web. 2017: 765-774.

[16] Xiao Y, Xiao R, Huang N, et al. Knowledge tracing based on multi-feature fusion[J]. Neural Computing and Applications, vol. 35, no.2, pp1819-1833, 2023.

[17] Zhang J, Xia R, Miao Q, et al. Explore Bayesian analysis in Cognitive-aware Key-Value Memory Networks for knowledge tracing in online learning[J]. Expert Systems with Applications, 2024: 124933.

[18] Ma C. Deep Knowledge Tracing Based on Behaviour in the Item Response Theory Framework[J]. 2024.

[19] Chun-Kit Yeung and Dit-Yan Yeung. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In Proceedings of the Fifth Annual ACM Conference on Learning at Scale, pp1-10, 2018.

[20] Sein Minn, Yi Yu, Michel C. Desmarais, Feida Zhu, and Jill-Jênn Vie. Deep knowledge tracing and dynamic student classification for knowledge tracing. In ICDM, pp1182-1187, 2018.

[21] Koki Nagatani, et al.. Augmenting knowledge tracing by considering forgetting behavior. In The World Wide Web Conference, pp 3101-3107, 2019.

[22] Ghosh, Aritra, Neil Heffernan, and Andrew S. Lan. "Context-aware attentive knowledge tracing." Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. 2020.

[23] Ting Long, Jiarui Qin, Jian Shen, Weinan Zhang, Wei Xia, Ruiming Tang, Xiuqiang He, and Yong Yu. Improving knowledge tracing with collaborative information. In WSDM, pp599-607, 2022.

[24] Choi Y, Lee Y, Cho J, et al. Towards an appropriate query, key, and value computation for knowledge tracing[C]//Proceedings of the seventh ACM conference on learning@ scale. 2020: 341-344.

[25] Shalini Pandey and Jaideep Srivastava. RKT: Relation-aware self-attention for knowledge tracing. In CIKM, pp1205-1214, 2020.

[26] Kai Zhang, Zhengchu Qin, Yue Liu, and Xinyi Qin, "Multiple Attention Modules-based Knowledge Tracing." IAENG International Journal of Computer Science, vol. 50, no. 2, pp633-640, 2023.

[27] C. S. Yin Wong, G. Yang, N. F. Chen and R. Savitha, "Incremental Context Aware Attentive Knowledge Tracing," ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, Singapore, pp3993-3997, 2022.

[28] Alexander H. Miller, Adam Fisch, Jesse Dodge, et al. Key value memory networks for directly reading documents. In EMNLP, pp1400-1409, 2016

[29] Ghodai Abdelrahman and Qing Wang. Knowledge tracing with sequential key-value memory networks. In SIGIR. pp175-184, 2019.

[30] Chen J, Liu Z, Huang S, et al. Improving interpretability of deep sequential knowledge tracing models with question-centric cognitive representations[C]//Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 12, pp14196-14204, 2023.

[31] E. Tsutsumi, Y. Guo, R. Kinoshita and M. Ueno, "Deep Knowledge Tracing Incorporating a Hypernetwork With Independent Student and Item Networks," in IEEE Transactions on Learning Technologies, vol. 17, pp951-965, 2024.

[32] Hiromi Nakagawa, Yusuke Iwasawa, and Yutaka Matsuo. Graph-based knowledge tracing: Modeling student proficiency using graph neural network. In WIC. Association for Computing Machinery, New York, NY, USA, 156163, 2019.

[33] Shiwei Tong, Qi Liu, Wei Huang, Zhenya Huang, Enhong Chen, Chuanren Liu, Haiping Ma, and Shijin Wang. Structure-based knowledge tracing: An influence propagation view. In ICDM, pp541-550, 2020.

[34] Yang Yang, Jian Shen, Yanru Qu, et al. GIKT: A graph-based interaction model for knowledge tracing. Springer International Publishing, pp299-315, 2020.

[35] Liu Y, Yang Y, Chen X, et al. Improving knowledge tracing via pre-training question embeddings[J]. arXiv preprint arXiv:2012.05031, 2020.

[36] Y. Zhang, Z. Wang, T. Yu, M. Lu, Z. Ren and J. Zhang, "Knowledge Tracing Based on Gated Heterogeneous Graph Convolutional Networks," 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, pp.6847-6849, 2022.

[37] McKinley R L, Reckase M D. An Extension of the Two-Parameter Logistic Model to the Multidimensional Latent Space[J]. 1983.

[38] Li G, Shuai J, Hu Y, et al. DKT-LCIRT: A Deep Knowledge tracing Model Integrating Learning Capability and Item Response Theory[J]. Electronics, vol. 11, no. 20, pp3364, 2022.

[39] Frances M. Yang and Solon T. Kao. Item response theory for measurement validity. Shanghai Archives of Psychiatry, vol. 26, no. 3, pp171-177, 2022

[40] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arxiv preprint arxiv: 1412.6980, 2014.

[41] Liu Z, Liu Q, Chen J, et al. simpleKT: a simple but tough-to-beat baseline for knowledge tracing[J]. arXiv preprint arXiv: 2302.06881, 2023.

[42] Feng M, Heffernan N, Koedinger K. Addressing the assessment challenge with an online system that tutors as it assesses[J]. User modeling and user-adapted interaction, 2009, 19: 243-266. url: https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data/skill-builder-data-2009-2010.

[43] Stamper J, Niculescu-Mizil A, Ritter S, et al. Algebra I 2005-2006 and Bridge to Algebra 2006-2007[J]. Development data sets from KDD Cup, 2010. url: https://pslcdatashop.web.cmu.edu/KDDCup/.

[44] Wang Z, Lamb A, Saveliev E, et al. Instructions and guide for diagnostic questions: The neurips 2020 education challenge[J]. arXiv preprint arXiv:2007.12061, 2020. url: https://eedi.com/projects/neurips-education-challenge.

[45] Paul Steif and Norman Bier. Oli engineering statics-fall 2011, 2014. url: https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=507.

[46] Xiaolu Xiong, Siyuan Zhao, Eric G. Van Inwegen, Joseph E. Beck. Going deeper with deep knowledge tracing[J]. International Educational Data Mining Society, 2016. url: https://sites.google.com/site/assistmentsdata/datasets/2015-assistments-skill-builder-data.

[47] Agarwal S, Graepel T, Herbrich R, et al. Generalization Bounds for the Area Under the ROC Curve[J]. Journal of Machine Learning Research, 2005, 6(4).

[48] Fair R C. Evaluating the predictive accuracy of models[J]. Handbook of econometrics, 1986, 3: 1979-1995.

[49] Jinseok Lee and Dit-Yan Yeung. Knowledge query network for knowledge tracing: How knowledge interacts with skills. In Proceedings of the 9th International Conference on Learning Analytics & Knowledge, pp491-500, 2019.

[50] Shuanghong Shen, Qi Liu, Enhong Chen, Zhenya Huang, Wei Huang, Yu Yin, Yu Su, and Shijin Wang. Learning process-consistent knowledge tracing. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp1452-1460, 2021.

[51] Ting Long, Yunfei Liu, Jian Shen, Weinan Zhang, and Yong Yu. Tracing knowledge state with individual cognition and acquisition estimation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp173-182, 2021.

[52] Xiaopeng Guo, Zhijie Huang, Jie Gao, Mingyu Shang, Maojing Shu, and Jun Sun. Enhancing knowledge tracing via adversarial training. In Proceedings of the 29th ACM International Conference on Multimedia, pp367-375, 2021.

[53] Pandey, Shalini, and George Karypis. "A self-attentive model for knowledge tracing." arXiv preprint arXiv: 1907.06837, 2019.

[54] Liu Z, Liu Q, Chen J, et al. Enhancing deep knowledge tracing with auxiliary tasks[C]//Proceedings of the ACM Web Conference 2023. pp4178-4187, 2023.

[55] Liu Z, Liu Q, Chen J, et al. pyKT: a python library to benchmark deep learning based knowledge tracing models[J]. Advances in Neural Information Processing Systems, vol.35, pp18542-18555, 2022.

Qingsong Guo received the B.S. degree in Computer Science and Technology from Zunyi Normal University, Zunyi, China, in 2021. She is currently pursuing the M.E. degree in Educational Technology at Yunnan Normal University Kunming, China. Her research interests include knowledge tracing and deep learning.

Jun Wang is the Director of Laboratory and Informatization Office at the Department of Education, Yunnan Normal University, and an associate researcher. He is currently pursuing the Ph.D. degree in Educational Technology at Yunnan Normal University, Kunming, China. His research interests include smart education and database engineering.

Xiaoyu Han received the M.S. degree in Educational Technology from Yunnan Normal University, Kunming, China, in 2024. She is currently pursuing the Ph.D. degree in Educational Technology at Yunnan Normal University, Kunming, China. Her research interests include knowledge tracing and deep learning.

Zijie Li received the M.S. degree in Educational Technology from Yunnan Normal University, Kunming, China, in 2022. He is currently pursuing the Ph.D. degree in Educational Technology at Yunnan Normal University, Kunming, China. His research interests include knowledge graph and knowledge tracing.

Juxiang Zhou received the Ph.D. degree in Control Theory and Control Engineering from Dalian University of Technology, Dalian, China, in 2019. She is currently the Deputy Director of the Key Laboratory of Smart Education, Yunnan Province, an associate professor, and Ph.D. supervisor at Yunnan Normal University, Kunming, China. Her main research interests are image and video processing, education informatization and smart education.