



Informative representations for forgetting-robust knowledge tracing

Zhiyu Chen¹ · Zhilong Shan¹ · Yanhua Zeng¹

Received: 14 April 2023 / Accepted in revised form: 6 January 2024 / Published online: 4 February 2024
© The Author(s), under exclusive licence to Springer Nature B.V. 2024

Abstract

Tracing a student's knowledge state is critical for teaching and learning. Knowledge tracing aims to accurately predict student performance by analyzing historical records on online education platforms. Most studies have focused on a student's skill with interactions sequence to predict the probability of correctly answering the latest question. However, they still suffer from the challenge of information sparsity and student forgetting. Specifically, the relationship between question and skill, and the features related to question texts have not been integrated to enrich information exploration. Besides, modeling forgetting behavior remains a challenge in assessing a student's learning gains. In this paper, we present a novel model, namely Informative Representations for Forgetting-Robust Knowledge Tracing (IFKT). IFKT utilizes a light graph convolutional network to capture various relational structures via embedding propagation. Then, the embeddings are assembled with rich interaction features separately as the powerful representation. Furthermore, attention weights assignments are individualized using the relative positions, in addition to the relevance between the current question with historical interaction representations. Finally, we compare IFKT against seven knowledge tracing baselines on three real-world benchmark datasets, demonstrating the superiority of the proposed model.

Keywords Knowledge tracing · Graph · Interaction features · Attention · Forgetting

✉ Zhilong Shan
zlshan@m.scnu.edu.cn

Zhiyu Chen
2020022957@m.scnu.edu.cn

Yanhua Zeng
2020023058@m.scnu.edu.cn

¹ School of Computer Science, South China Normal University, Guangzhou 510631, China

1 Introduction

The COVID-19 pandemic has challenged traditional classroom-based teaching and facilitated the rise of online education. While the intelligent tutoring system provides users with massive educational resources, it also allows users to interact with the online education platform, generating a large amount of data (Liu et al. 2021; Li et al. 2018). Educational data mining aims to minimize the cost of students' learning and teachers' instruction while maximizing learning benefits for students. Data analysis technology can facilitate collecting and analyzing student-related data on a large scale (de Andrade et al. 2021). Knowledge tracing is a task that continuously tracks students' skill level and allows for significant labor cost savings. It enables teachers to tailor their guidance based on students' proficiency level by recommending appropriate learning resources (Wang et al. 2021), and it also assists teachers to keep track of students' abilities. Therefore, knowledge tracing has been investigated for decades (Cui et al. 2022).

Many classical and mainstream knowledge tracing approaches use skills as model input. For example, Bayesian Knowledge Tracing (BKT) (Corbett and Anderson 1994) employs a hidden Markov method to model each skill independently while Deep Knowledge Tracing (DKT) (Piech et al. 2015) employs a long short-term memory (LSTM) to jointly model the states of all skills, Dynamic Key-Value Memory Network (DKVMN) (Zhang et al. 2017) stores skills in a static memory matrix and updates students' mastery level for each skill in a dynamic memory matrix. The goal of KT is to precisely trace the evolving knowledge states of students on the skills based on their performance in correctly answering the next question. Therefore, skills cannot completely substitute for questions as model input, losing much information about the questions themselves. To overcome this limitation, one direction follows graph-based approaches. Graph-based Interaction Knowledge Tracing (GIKT) (Yang et al. 2020) leverages the relation between questions and skills, constructed as a graph to learn embedding for prediction. Besides, skills are relevant to each other, and there are similarities among the skills covered by the same problem (Huang et al. 2019). Pre-training Embeddings via Bipartite Graph (PEBG) (Liu et al. 2020) obtains pre-trained exercise embeddings to improve the precise prediction of KT, which explores explicit question–skill relations and implicit question similarity as well as skill similarity simultaneously. Another direction is to use additional features from profiles on Intelligent Tutoring System, text-aware models (Su et al. 2018; Liu et al. 2021; Cheng et al. 2022) exploit the interaction features of questions themselves while student behavior models incorporate temporal features (Shin et al. 2021; Wang et al. 2021) to rich the model input. The inclusion of additional features often results in more precise predictions of student learning outcomes (Abdelrahman et al. 2022). Nonetheless, they still suffer from the inherent challenge of *information sparsity*: classical methods only use skills as model inputs, and recent studies either focus on the potential association between problems and skills or the student profiles from the online platform. In this work, various relational structures are integrated with features related to the student and question text.

After mining the information of student data records provided by the platform, one of the most critical problems is how to assess the knowledge state of students. Following the Transformer architecture (Vaswani et al. 2017), the latest works like

Self-Attentive Knowledge Tracing (SAKT) (Pandey and Karypis 2019) and Separated Self-Attentive Neural Knowledge Tracing (SAINT) (Choi et al. 2020) have been generalized to incorporate attention mechanisms into the KT domain. Both learn the attention weights of questions in a sequence of interactions, which can reflect the relative importance of these historical interactions for predicting the probability of correctly answering the next question. Studies in learning psychology (Sense et al. 2016) have shown that *forgetting* is an essential aspect to consider for an accurate assessment of students' knowledge state. The *learning curve theory* proposed by Ebbinghaus (Ebbinghaus 2013) focuses on the *forgetting effect* and the *learning effect* that affect memory retention. The more practices a student does, the higher level of related knowledge, and the more time passes, the more knowledge a student will forget. Attentive Knowledge Tracing (AKT) (Ghosh et al. 2020) differs from SAKT and SAINT in its attention, using an exponential decay rate to reduce attention weights for questions in a sequence of interactions, which means considering the *forgetting effect* in a student's memory over time. Other methods (Nagatani et al. 2019; Shen et al. 2021; Wang et al. 2021) have been developed to consider a student's forgetting behavior during tracing knowledge states. However, these approaches either simplify the forgetting behavior or reveal the retention rate decreases exponentially as time passes by. The work presented in this paper is the first approach that individualizes attention with relative position to simulate student forgetting in the KT task.

In this paper, we propose a new model called Informative Representations for Forgetting-Robust Knowledge Tracing (IFKT), which can concurrently tackle the information sparsity and student forgetting problem. Firstly, a light graph convolutional network (lightGCN) (He et al. 2020) is employed to investigate the various relational structures of questions and skills via embedding propagation. The embeddings of questions involving the same skill are similar, while the embeddings of questions involving different skills are more distinct. Meanwhile, the embeddings of skills involved in the same question are close, while the embeddings of other skills have more differences from each other. Secondly, in addition to the potential information between question and skill, we investigate rich features of question text, including *problem type*, *counts of attempt*, and *response time*. Instead of combining them all, we integrate them into the question embedding and the interaction embedding, respectively, because they can indirectly reflect the difficulty of questions and students' knowledge level separately. Thirdly, learning gains not only grow with continuous practice, but also decrease as students forget. Therefore, in addition to the relative importance of historical interactions on the current predicted question, we innovatively introduce the idea of attention with the relative position (Shaw and Uszkoreit 2018) to simulate student forgetting during tracing knowledge state. Specifically, it distinguishes the impact of historical interactions in different moments on the current predicted question, resulting in a more accurate picture of students' knowledge states. Finally, IFKT outperforms baselines in terms of prediction performance on three public datasets commonly used in the knowledge tracing domain, according to experimental results. The contributions of IFKT are summarized as follows:

- (1) We use a light graph convolutional network to explore various relational structures between questions and skills via embedding propagation, then the embeddings

are integrated with the features relating to question text separately to construct powerful representations. IFKT is capable to mitigate the information sparsity problem.

- (2) We incorporate the relative position to individualize attention weight assignments, in addition to the importance between historical interactions and the current question. IFKT can better assess a student's knowledge state considering student forgetting.
- (3) We extensively evaluate the performance of our proposed IFKT model by comparing it against the mainstream deep learning and graph KT models on three well-established KT datasets. The experimental results demonstrate that IFKT achieves state-of-the-art performance.

The rest of the paper is structured as follows. Section 2 reviews the related work. Section 3 presents the framework of IFKT. Results and discussion, parameter analysis, and visualization are reported in Sect. 4, followed by the conclusion and future work in Sect. 5.

2 Related work

In this section, we first review the related work on knowledge tracing. Then, we present a brief overview of two fields of light graph convolutional networks and relative position representations, which are closely related to our work.

2.1 Knowledge tracing

Knowledge tracing aims to monitor students' evolving knowledge states during the learning process and predicts students' future performance on the next question based on their past interactions. It can be summarized as follows: given a student's historical interaction sequence $X = \{x_1, x_2, \dots, x_t\}$, here $x_t = \{q_t, a_t\}$ represents the student's answer a_t to the question q_t at time t , to predict the probability $p(a_{t+1} = 1 | q_{t+1}, X)$ that the student would answer the question correctly at $t + 1$. Knowledge tracing has become a critical research topic in recent decades due to its advantages in supporting intelligent educational services. In summary, previous studies on knowledge tracing can be grouped into two categories: traditional models and deep learning models.

2.1.1 Traditional models

BKT is the most primitive and widely used probabilistic model in the domain, which is an application of the hidden Markov model that uses binary variables to represent students' historical responses as a sequence of observations and mastery of skills as states. The formulation of BKT considers students' guessing and slipping, but assumes that once a skill is learned it will never be forgotten. After that, BKT undergoes multifaceted and extended research, such as incorporating temporal features (Vie and Kashima 2019) to enrich information, representing students' performance on a question with continuous values rather than binary outputs (Wang and Heffernan 2013),

and examining students' cognition of both objective and subjective problems with a fuzzy system (Wu et al. 2015). Despite the initial breakthroughs in traditional models, they struggle to adequately capture highly complex cognitive processes.

2.1.2 Deep learning models

DKT pioneers deep learning to model students' learning overall, which achieves quite good performance. To enhance the interpretability of DKT, DKVMN introduces a Memory Augmented Neural Network to handle knowledge tracing task with keys being the skills and values being the mastery of students. Since students' learning behaviors and abilities are similar sometimes, Minn et al. extend DKVMN to divide students into groups for personalized predictions (Minn et al. 2018). Transformer abandons recurrence and relies entirely on the self-attention mechanism to capture global dependencies within a sequence. SAKT first adopts the self-attention mechanism to capture long-term dependencies, it learns the relevance between the current problem and previous interactions. Pu et al. (2020) modifies the structure of the self-attention mechanism and refines the attention weights considering the time interval between problems. SAINT creates a multilayer attention mechanism that leverages an encoder–decoder structure to process information about the problem and response separately. In addition, an adaptive sparse self-attentive network is proposed by Wang et al. (2021) to generate missing features and access fine-grained student performance predictions. AKT emphasizes that in addition to the strong correlation among problems involving the same skill, the difficulty of problems and the learning ability of students should be taken into account. Other numerous approaches have been developed to handle knowledge tracing with the ability of graph convolutional networks (GCN) (Kipf and Welling 2017). From a data structure perspective, various relational structures often exist within questions and skills, like the similarity of skills (Tong et al. 2020), and dependency between skills (Nakagawa et al. 2019). Moreover, SGKT models students' learning process using a dynamic session graph, and focuses on cold-start problems like a few students' training data and new students occurring (Zhengyang et al. 2022). HHSKT extracts knowledge structure features through a heterogeneous graph, and can estimate the difficulty of problems dynamically (Ni et al. 2023). However, early studies explore the graph structure between questions or skills depending on a single type of relationship information. By contrast, we adopt a light graph convolutional network to mine the heterogeneous associations between questions and skills.

2.2 Light graph convolutional network

Light graph convolutional network (LightGCN) is an effective deep learning technique that has been widely used in the recommendation domain (He et al. 2020). It simplifies the design of the GCN, only leaving the most important components, such as neighborhood aggregation and layer combination. This modification offers it an advantage in constructing heterogeneous node embeddings: easier to be trained together with better generalization ability. In the recommender system application, the embeddings obtained by LightGCN are similar for users who are interested in the

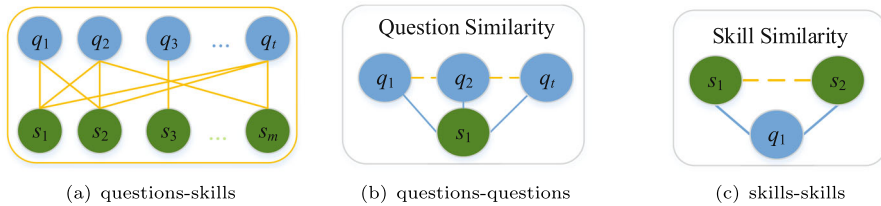


Fig. 1 Heterogeneous relation graph of questions and skills

same item. Inspired by this phenomenon, we apply LightGCN to knowledge tracing for discovering the associations between questions and skills. As shown in Fig. 1(a), there is a first-order relationship between questions and skills. In addition, there is a second-order relationship among questions. As shown in Fig. 1(b), where q_1, q_2 , and q_t all cover skill s_1 , then q_1, q_2 , and q_t are similar to each other. Correspondingly, there is a second-order relationship among skills, as shown in Fig. 1(c), where q_1 involves both skills s_1, s_2 , and s_1, s_2 are similar.

2.3 Relative position representations

The self-attention with relative position representations is primarily designed for machine translation (Shaw and Uszkoreit 2018). It extends the self-attention mechanism to efficiently consider the pairwise relationships between input elements. Specifically, it constructs a labeled, directed, fully connected graph to model the input. Edges can detect information about relative position differences between input elements for linear sequences. Beyond the correlation between the historical interactions and the predicted question, the weights require to be individualized since students may forget during learning. Therefore, self-attention with relative position representations is incorporated into our model for weight distinction.

3 The IFKT model

3.1 Model overview

The architecture of our proposed method is shown in Fig. 2. IFKT contains three components: embedding, knowledge state, and prediction. The embedding module is designed to capture potential information about questions and skills, then supplement the input with other interaction features. The knowledge state module individualizes the weights of historical questions on the current prediction by accounting for student forgetting, especially for questions that involve the same skills as the current one. The prediction module is used to predict the probability that a student would answer the question correctly at the next moment. Table 1 shows some important notations for the sake of presentation clarity.

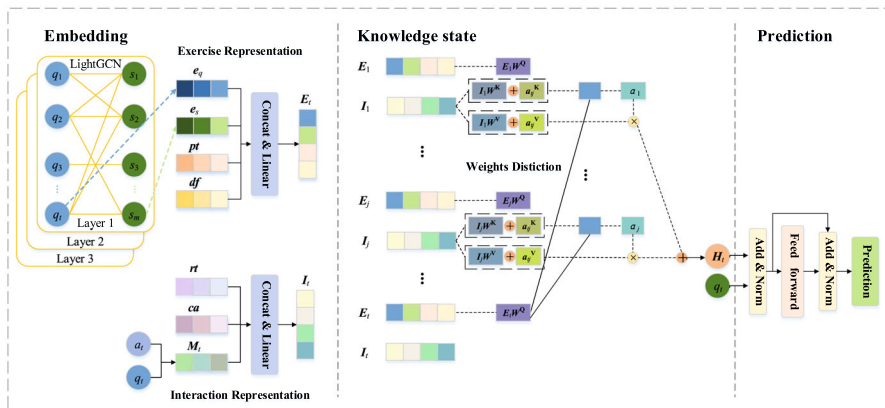


Fig. 2 The overall architecture of the proposed IFKT method

Table 1 Notations

Notations	Descriptions
$problem\ type$	Question's type
$counts\ of\ attempt$	Number of attempts of the student
$response\ time$	Time for student's first response
$correct$	Correct or incorrect answer to the question
q_i, s_i, a_i	The question, skill, and answer
K	The maximal aggregate layer
l	The maximum length

3.2 Embedding

The embedding layer transforms raw data into continuous vectors, residing in a high-dimensional space. This process crucially preserves semantic information, providing the model with enriched and informative data, thereby enhancing its performance.

3.2.1 Interaction embedding

The interaction x_t is first represented numerically as $y_t = q_t + a_t \times E$, where E is the total number of questions. As a result, the maximum possible value for an element in the interaction sequence is $2E$. Then, the sequence $\mathbf{y} = \{y_1, y_2, \dots, y_t\}$ is changed into $\mathbf{z} = \{z_1, z_2, \dots, z_n\}$, where n is the longest sequence that the model can handle. After training, interaction embedding matrix $\mathbf{M}$ is obtained, $\mathbf{M} \in R^{2E \times d}$, where d is the latent dimension, and $\mathbf{M}_t$ for each element in the sequence.

3.2.2 Question embedding and skill embedding

To collect potential information from questions–skills, questions–questions, and skills–skills. LightGCN is applied to get the question embedding and skill embed-

ding. These embeddings are learned through a linear propagation process on the question–skill graph. The final embedding is determined by aggregating the embeddings acquired at all layers using weighted sums. In a more specific context, assuming that N_q is the number of skills involved in question q and N_s is the number of questions involving skill s . $\mathbf{e}_q^{(0)}$ and $\mathbf{e}_s^{(0)}$ are the initial embeddings of the question q and skill s , respectively. LightGCN propagates the embeddings using the heterogeneous graph of question and skill as follows:

$$\begin{aligned}\mathbf{e}_q^{(k+1)} &= \sum_{s \in N_q} \frac{1}{\sqrt{|N_q| |N_s|}} \mathbf{e}_s^{(k)} \\ \mathbf{e}_s^{(k+1)} &= \sum_{q \in N_s} \frac{1}{\sqrt{|N_s| |N_q|}} \mathbf{e}_q^{(k)}\end{aligned}\quad (1)$$

where $\mathbf{e}_s^{(k)}$ and $\mathbf{e}_q^{(k)}$ denote the fine-grained embeddings of question q and skill s after k -layer propagation, respectively. The symmetric normalization term $\frac{1}{\sqrt{|N_q| |N_s|}}$ is deployed to avoid increasing the embedding scale with the convolution operation.

After k -layer light graph convolution propagation, due to the potential oversmoothing of embeddings with increasing layers, the final layer is not utilized. Instead, the ultimate representation is obtained by weighted summation across all layers. Furthermore, this approach facilitates the extraction of semantic information at different levels and inherently incorporates an effect akin to self-connection. Finally, representations of the question and skill are acquired as follows:

$$\begin{aligned}\mathbf{e}_q &= \sum_{k=0}^K \alpha_k \mathbf{e}_q^{(k)} \\ \mathbf{e}_s &= \sum_{k=0}^K \alpha_k \mathbf{e}_s^{(k)}\end{aligned}\quad (2)$$

where α_k denotes the weight coefficient of the k -layer embedding in constituting the final embedding. In this paper, α_k is set to $\frac{1}{K+1}$. The subsequent analysis experiment proves that the best choice is setting K to 3.

3.2.3 Interaction features

When students interact with the online education platform, features such as *counts of the correct answer* (N_c), or *counts of the incorrect answer* (N_w), *problem type* (pt), *response time* (rt), and *counts of attempt* (ca) are generated. Such features indirectly reflect the difficulty of questions and the students' knowledge state, so incorporating them can both improve prediction precision and help understand the model's performance better.

Difficulty. The difficulty of questions varies, even when they involve the same skill. Testing whether the student's knowledge is up to standard should be based on the

difficulty of the overall assignment. We define the *difficulty* (df) coefficient of each question as the ratio of *counts of the correct answer* in *total response*:

$$df = \frac{N_c}{N_c + N_w} \quad (3)$$

where N_c and N_w correspond to the counts of correct and wrong answer, respectively.

3.2.4 Exercise representation

Problem type can indirectly reflect the difficulty of a question, so *problem type* and *difficulty* are used as additional information for the question embedding. First, *problem type* and *difficulty* are up-dimensioned using *linear* to generate $\mathbf{df} \in \mathbb{R}^{1 \times d}$ and $\mathbf{pt} \in \mathbb{R}^{1 \times d}$. Since directly concatenating question embedding, skill embedding with *problem type* and *difficulty* may trigger high dimensionality, the *linear* layer is connected for dimensionality reduction to maximize the retention of valid information. The final exercise representation $\mathbf{E}_t, \mathbf{E}_s \in \mathbb{R}^{E \times d}$ is constructed as follows:

$$\mathbf{E}_t = \text{Concat\&Linear}(\mathbf{e}_q, \mathbf{e}_s, \mathbf{df}, \mathbf{pt}) \quad (4)$$

3.2.5 Interaction representation

Instead of directly assigning all interaction features to the question embedding, we add *counts of attempt* and *response time* as supplementary information to the interaction embedding. However, these two attributes are the behaviors of students during the interaction, which reflect both the difficulty and the student's knowledge state. The more *counts of attempt* and the longer *response time* to a question indicate that the question is more difficult or that the student has lower mastery of knowledge. Conversely, the easier the question or the higher level of the student's knowledge. Subsequent analytical experiments have shown that it is better to integrate the interaction features into the question embedding and the interaction embedding separately than directly assembling all the features into the question embedding. Similarly, *Concat & Linear* is adopted to construct the final representation of interaction $\mathbf{I}_t, \mathbf{I}_s \in \mathbb{R}^{2E \times d}$, as follows:

$$\mathbf{I}_t = \text{Concat\&Linear}(\mathbf{ca}, \mathbf{rt}, \mathbf{M}_t) \quad (5)$$

3.3 Knowledge state

Students may forget during the learning process, so the weights of historical questions assigned to the predicted ones at different moments should be distinguished, especially those involving the same skills. As shown in Fig. 3, for the prediction of the question at the moment t_{19} , some of the previous methods consider that historical moments are assigned the same weight coefficients if they involve the same skills as the current one. However, at moments t_2, t_3, t_7, t_{12} , and t_{19} , students respond to questions related to skill s_6 , and the heat map visualizes different weight assignments. Intuitively, the weights

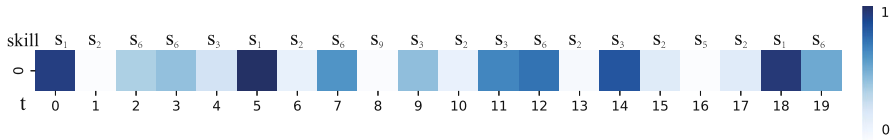


Fig. 3 The individualized attention weights heat map

assigned to the farther moment should be smaller than the nearer one, because the degree of influence is reduced due to student forgetting. While AKT uses a monotonic attention mechanism to summarize past learner performance in the right time scale, IFKT differs in that it personalizes attention weights using relative position encoding to simulate students' forgetting of previously learned skills during the learning process. We elaborate on these intuitions in the following section.

3.3.1 Weight distinction

The final representations of the question $\mathbf{E} = [\mathbf{E}_1, \dots, \mathbf{E}_t]$ and the interaction $\mathbf{I} = [\mathbf{I}_1, \dots, \mathbf{I}_t]$ are acquired from the embedding component. $\mathbf{E}_t$ and $\mathbf{I}_t$ denote the final embedding of the question and the interaction at time t , respectively. This part is intended to capture the student's knowledge state. Self-attention with relative position representations is used to distinguish the impact of historical interactions on the current prediction, especially for questions involving the same skill. It extends the self-attention mechanism to efficiently consider the pairwise relationships between input elements. Specifically, it constructs a labeled, directed, fully connected graph to model the input. Edges can encode information pertaining to the relative positional disparities among input elements in the context of sequences. Then, the edge between input elements $\mathbf{E}_t$ and $\mathbf{I}_j$ is represented by vectors $\mathbf{a}_{tj}^V, \mathbf{a}_{tj}^K \in \mathbb{R}^d$:

$$\mathbf{a}_{tj}^V = w_{clip(j-t, r)}^V, \quad \mathbf{a}_{tj}^K = w_{clip(j-t, r)}^K$$

$$clip(j - t, r) = \max(-r, \min(r, j - t)) \quad (6)$$

where relative position representations $w^K = (w_{-r}^K, \dots, w_r^K)$ and $w^V = (w_{-r}^V, \dots, w_r^V)$ are the learnable parameters during training. Because precise information is not available beyond a certain distance, the maximum relative position is clipped to a maximum absolute value of r , and r is set as the default value here.

Next, the interaction sequence $\mathbf{I} = [\mathbf{I}_1, \dots, \mathbf{I}_t]$ and the question sequence $\mathbf{E} = [\mathbf{E}_1, \dots, \mathbf{E}_t]$ are computed as the attention score $\mathbf{H} = [\mathbf{H}_1, \dots, \mathbf{H}_t]$ of each attention head, and edge information is propagated to the sublayer output, where $\mathbf{H}_t$ is a weighted sum of linearly transformed input elements:

$$\mathbf{H}_t = \sum_{j=1}^n (\alpha_{tj} (\mathbf{I}_j \mathbf{W}^V + \mathbf{a}_{tj}^V)) \quad (7)$$

The weight coefficient α_{tj} is calculated using a *softmax* function:

$$\alpha_{tj} = \frac{\exp e_{tj}}{\sum_{r=1}^n (\exp e_{tr})} \quad (8)$$

e_{tj} is calculated using a compatibility function that compares two input elements while also taking edge information into account:

$$e_{tj} = \frac{\mathbf{E}_t \mathbf{W}^Q (\mathbf{I}_j \mathbf{W}^K + \mathbf{a}_{tj}^K)^T}{\sqrt{d}} \quad (9)$$

To avoid broadcasting relative positions, we split the above equation as:

$$e_{tj} = \frac{\mathbf{E}_t \mathbf{W}^Q (\mathbf{I}_j \mathbf{W}^K)^T + \mathbf{E}_t \mathbf{W}^Q (\mathbf{a}_{tj}^K)^T}{\sqrt{d}} \quad (10)$$

3.3.2 Multi-head attention

The self-attention layer uses h heads to catch information from different subspaces. The concatenation of h attention heads is multiplied by $\mathbf{W}^O$ to aggregate the output of different attention heads:

$$\mathbf{U} = \text{MultiHead}(\mathbf{E}, \mathbf{I}) = \text{Concat}(\mathbf{head}_1, \dots, \mathbf{head}_h) \mathbf{W}^O \quad (11)$$

where the projections are parameter matrices $\mathbf{W}^Q \in \mathbb{R}^{d \times d_k}$, $\mathbf{W}^K \in \mathbb{R}^{d \times d_k}$, $\mathbf{W}^V \in \mathbb{R}^{d \times d_v}$, $\mathbf{W}^O \in \mathbb{R}^{hd_v \times d}$, and $d_k = d_v = d/h$, $\mathbf{head}_h = \mathbf{H}^h$.

3.4 Prediction

Subsequently, we implement a feed-forward neural network, including nonlinearity in the model as well as considering the interactions between different potential dimensions:

$$\mathbf{F} = \text{FFN}(\mathbf{U}) = \text{ReLU}(\mathbf{U} \mathbf{W}^{(1)} + b^{(1)}) \mathbf{W}^{(2)} + b^{(2)} \quad (12)$$

where $\mathbf{W}^{(1)}, \mathbf{W}^{(2)} \in \mathbb{R}^{d \times d}$, $b^{(1)}, b^{(2)} \in \mathbb{R}^d$ are the learnable parameters during training.

A residual network is adopted between the attention layer and feed-forward network, which contributes to propagating the information in the historical sequence that is far from the current forecast question to the last layer, and thus allows the model to trap the underlying information. Layer normalization is used afterward to stabilize and accelerate the neural network. It performs normalization on input samples within each layer, transforming them into a standard normal distribution with a mean of 0 and a variance of 1. The attention network in our model requires a masking mechanism to enforce causality constraints, ensuring that the prediction at t can depend only on the

first $t - 1$ interactions. By adhering to this causality constraint, the model effectively prevents the inclusion of information from future interactions during the attention process. This strategy enables the model to produce more accurate and contextually relevant predictions by concentrating on past interactions, thereby contributing to improved temporal coherence and overall performance in sequential tasks.

Thereafter, the student's performance is predicted through a fully connected network and the sigmoid function, with p denoting the probability of the student answering the question correctly:

$$p = \sigma(\mathbf{F}w + b) \quad (13)$$

Finally, the parameters in our model are updated using gradient descent to minimize the cross-entropy loss between the final prediction of the model and the true labels:

$$\mathbf{L} = \sum_{t=1}^T (a_t \log(p_t) + (1 - a_t) \log(1 - p_t)) \quad (14)$$

4 Experiments and results

In this section, we describe the extensive experiments that are conducted to demonstrate the effectiveness and advantage of the IFKT. We first evaluate the prediction of IFKT with several baselines on three open source datasets. Subsequently, the discrepancies between the experimental outcomes and those reported in the original paper are analyzed, along with the impact of diagnostic errors on their applicability in real educational scenarios. Following that, to observe the core module of our approach, we conduct three ablation studies. Finally, parameter analysis and visualization are helped investigate the sensitivity and better understand the ability of IFKT.

4.1 Implementation details

Since the length of users' learning sequences varies significantly among different datasets, the maximum length l is set using the average of all users' sequences after grouping each dataset by user_id. The sequence that less than the specified length is left-filled with 0. We randomly select 80% of students for training and the remaining 20% for testing. The area under the ROC curve (AUC) and the accuracy (ACC) are used as the evaluation metrics, and the higher the AUC and ACC, the better the performance of prediction.

We implement the network training and testing using *Pytorch*. The original hyperparameters are set for all baselines. In IFKT, the embedding dimension of the question, skill, interaction, and interaction features is set to 128. The maximal aggregate layer K is fixed as 3 in LightGCN. We use the Adam optimizer to update the model parameters, the initial learning rate is turned to 0.001, and the batch size is 64. We also use dropout with the keep probability of 0.8 to avoid overfitting. The limitation of the early

Table 2 Dataset statistics

Items	ASSIST09	ASSIST12	Algebra
# user	4151	28, 834	1336
# question	16, 891	50, 988	91, 759
# skill	101	198	1703
# interaction	274, 590	2, 630, 080	1, 806, 782
# problem type	5	6	153
# average sequence length	66	92	1352

stopping module is six epochs. For the sake of reproducibility, the codes and models of our work are available.

4.1.1 Datasets

To evaluate IFKT, experiments are conducted on three benchmark datasets. The statistics of the three datasets are listed in Table 2.

ASSIST2009.¹ The dataset was collected from the ASSISTments online education platform during the 2009-2010 school year. After removing the redundant data, the dataset contains 4,151 users, 274,590 interactions, and 16,891 questions involving 101 skills. The average sequence length after grouping by user_id is 66.

ASSIST2012.² The dataset was collected from the ASSISTments online tutoring platform during the school year 2012-2013. We execute the same data processing as for ASSIST09, the dataset contains 28,834 users, 2,630,080 interactions, and 50,988 questions involving 198 skills. The average sequence length after grouping by user_id is 92.

Algebra³: It is commonly used in *KDDCup* educational data mining competitions, and Algebra 2006-2007 is used in this paper. After removing null values, the dataset contains 1,336 users, 1,806,782 interactions, and 91,759 questions involving 1703 skills. The average sequence length after grouping by user_id is 1352.

To explore the actual baseline data, a snippet of student records from Assist2012 has been carefully curated, and the selected featured attributes are presented in Table 3.

4.1.2 Baselines

To evaluate the effectiveness of IFKT, we introduce several models that are used in our comparison experiments.

- DKT: DKT uses LSTM to model student skill learning, which is the first application of deep learning in knowledge tracing (Piech et al. 2015).

¹ <https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data/skill-builder-data-2009-2010>.

² <https://sites.google.com/site/assistmentsdata/datasets/2012-13-school-data-with-affect>.

³ <https://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp>.

Table 3 A glimpse into baseline data

user_id	question	skill	ca	rt	pt	correct
61394	365981	Rounding	1	21175	algebra	1
76592	404936	Parallel and Perpendicular Lines	3	126238	choose_1	0
76592	411423	Mean	1	499410	fill_in_1	1
78401	187514	Ordering Real Numbers	1	105439	choose_n	1
176421	528831	Ordering Fractions	4	145339	rank	0

- **DKVMN:** DKVMN employs a key-value memory network to store the skills and update students' knowledge states (Zhang et al. 2017).

- **SAKT:** SAKT is the first knowledge tracing method that uses the transformer's self-attention architecture. It is a single encoder-based model where question ids are used as queries and interaction ids are used as keys and values (Pandey and Karypis 2019).

- **SAINT:** SAINT is the first transformer-based knowledge tracing method that exploits an encoder–decoder structure to process information about the question and student response separately (Choi et al. 2020).

- **GIKT:** GIKT proposes a graph-based interaction model to tackle the data sparsity and multi-skill questions, which takes the relevance of questions and skills as model inputs (Yang et al. 2020).

- **AKT:** AKT adopts the transformer network to model the learning sequences of students, which improves the performance and interpretability through the monotonic attention mechanism and the Rasch model (Ghosh et al. 2020).

- **SGKT:** SGKT proposes a dynamic session graph-based knowledge tracing approach, which builds directed graph and heterogeneous relation graph to capture the local and global information. Besides, it focuses on cold-start problems like a few students' training data and new students occurring. Zhengyang et al. (2022).

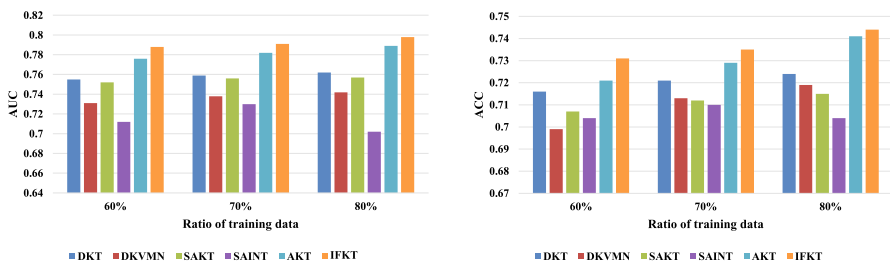
4.2 Results and discussion

4.2.1 Overall results

Table 4 lists the performance of IFKT compared to baseline methods. IFKT* represents a variant of the IFKT model that assigns all interaction features to the question embedding. Its performance is weaker compared to IFKT, which demonstrates the significance of separately concatenating interaction features with question and interaction. Because *response time* and *counts of attempt* are the behaviors of students in the learning progress, they reflect both the question's difficulty and students' mastery level. From the results, we observe that our IFKT achieves the highest performance across all datasets, which verifies the effectiveness of our model. We found that IFKT and AKT perform better on ASSIST12 and Algebra than ASSIST09, which means IFKT can handle data when the user's interaction records are enormous. The result suggests

Table 4 Performance comparison on three benchmark datasets

Methods	ASSIST09		ASSIST12		Algebra	
	AUC	ACC	AUC	ACC	AUC	ACC
DKT	0.762	0.724	0.698	0.72	0.758	0.782
DKVMN	0.742	0.719	0.665	0.710	0.756	0.787
SAKT	0.757	0.715	0.695	0.718	0.744	0.772
SAINT	0.702	0.704	0.657	0.703	0.753	0.787
GIKT	0.792	0.741	0.781	0.752	—	—
AKT	0.784	0.735	0.851	0.806	0.841	0.836
SGKT	0.795	0.742	0.821	0.801	—	—
IFKT*	0.78	0.724	0.846	0.805	0.802	0.808
IFKT	0.798	0.744	0.869	0.824	0.859	0.846

**Fig. 4** Prediction of users' performance for different training ratios under AUC and ACC

that the individualized attention mechanism can help capture sequence dependencies accurately and it is necessary to distinguish the question's difficulty.

Figure 4 depicts the results of IFKT against baselines under different training ratios. This suggests that IFKT inherently explores the characteristics of students and their interaction items, thereby leading to better prediction performance.

4.2.2 Analysis of discrepancies

It is crucial to note that disparities exist between the experimental results and the AUC reported in the original paper. Upon careful analysis, we attribute the principal factor to the proprietary and personalized nature of the data preprocessing procedures. To elaborate, conventional methods typically provide preprocessed data within publicly accessible code repositories. However, these methods exhibit considerable disparities. For example, when dealing with the maximum length, some approaches exclude users with fewer than three records, while others remove users with fewer than ten records. Furthermore, in the process of reconstructing user sequences, certain methods segment longer records beyond a specified maximum length to create new user sequences, while padding sequences that do not reach the maximum length. Conversely, other approaches retain only a user's most recent records while discarding distant historical records.

Meanwhile, Abdelrahman et al. (2022); Liu et al. (2022) have also pointed out that the evaluation of models in the past has been nonstandardized, resulting in significant variations in performance even for the same model on the same datasets, ranging from 0.709 to 0.86. Furthermore, previous studies (Liu et al. 2020; Abdelrahman et al. 2022; Ghosh et al. 2020; Ni et al. 2023) have reported results inconsistent with the original papers. However, the overall trend is similar. For instance, some models following DKT exhibited inferior performance compared to DKT.

To ensure a level playing field, our experiments are conducted in a consistent manner, including experimental equipment, data preprocessing, data splitting, and evaluation methods. The parameters for the baselines are set in accordance with configurations of the original papers, with the exception of the maximum length. This deviation is a result of our experiments, which show that simply setting the maximum sequence length to 50, 100, or 200 may constrain the model's performance. Instead, setting it as the average sequence length enhances the model's performance, at least on these three publicly available datasets, as detailed in the parameter analysis section. Any deviations between our results and those of the original papers describing the baseline experiments are due to enforcing this level playing field.

Hence, future efforts could prioritize standardization, encompassing uniform data preprocessing, standardized dataset partitioning, and the establishment of real-world evaluation protocols within the educational context, as recommended by PYKT (Liu et al. 2022). Standardization not only assists researchers in the field in easing the task of comparing existing baselines and developing new algorithms but also guarantees a fair and transparent comparison among different methods (Liu et al. 2022).

4.2.3 Effect of diagnostic error

Existing methods are outperformed by IFKT, and the diverse interaction data available on the platform is utilized more effectively. The model can be expanded to incorporate additional information that has not yet been utilized but may be collected by the platform in the future. Further research is required to refine our information selection methods. Larger datasets are particularly excelled in by IFKT, which is beneficial for educational scenarios with extensive interaction records. Additionally, longer student records can be handled by it, aiding in more accurate self-assessment of their knowledge.

However, the experimental results reveal a persisting prediction error rate exceeding 20%. In real-world educational contexts, this poses a significant challenge for the smooth implementation of personalized downstream tasks, adding burdens to both students and teachers. For instance, when systems recommend inappropriate educational resources, it not only increases the learning burden on students but may also inhibit learning gains. Moreover, imprecise teacher awareness of students' knowledge states can lead to negative teaching interventions. Additionally, students' lack of awareness of their learning gaps may result in the expenditure of significant time and effort on practicing knowledge they have already mastered. Even though having only 80% accuracy would be problematical in many situations, there are learning contexts where it would actually be pretty good, such as finding peer helpers as is done in iHelp system who can interact with the learner to clarify any misdiagnoses. To shed light



Fig. 5 Correct answer rate for each problem type

Table 5 Comparison of the performance of the IFKT based on the different submodules

Methods	ASSIST09		ASSIST12		Algebra	
	AUC	ACC	AUC	ACC	AUC	ACC
N-PESE	0.787	0.737	0.838	0.801	0.855	0.839
N-IF	0.747	0.715	0.724	0.727	0.757	0.799
N-WD	0.791	0.731	0.846	0.808	0.848	0.841
IFKT	0.798	0.744	0.869	0.824	0.859	0.846

on the underlying factors contributing to these limitations, we conduct an in-depth investigation, with particular emphasis on potential issues such as data class imbalances and the cold-start problem. The data class imbalances, for example, illustrated in Fig. 5 for Assist2012, encompass six problem types with uneven distributions of correct and incorrect answers. Addressing the cold-start problem, we partially alleviate it by excluding students with fewer than three records. These factors collectively introduce prediction biases that significantly impact the model's performance. Therefore, in future endeavors, the fundamental task of knowledge tracing, which is dedicated to enhancing the predictive accuracy of models, remains a crucial focus.

4.2.4 Ablation study

To further investigate the contributions of different components of IFKT, we perform three ablation experiments comparing several variants of the IFKT method. The results are shown in Table 5.

Table 6 Refinement of interaction features ablation experiments

Datasets	N-df	N-pt	N-ca	N-ms	N-IF	IFKT
ASSIST09	0.752	0.795	0.794	0.793	0.747	0.798
ASSIST12	0.755	0.796	0.758	0.798	0.724	0.869
Algebra	0.821	0.852	–	0.781	0.757	0.859

N-PESE is IFKT without question embedding and skill embedding from LightGCN. On all datasets, IFKT outperforms its counterpart, N-PESE, which uses vanilla embedding for question and skill. These results suggest that using LightGCN to obtain question embedding and skill embedding is effective at capturing potential information.

N-IF is IFKT without difficulty and other interaction features. The experimental results from the whole datasets show that the AUC of the model decreases to some extent. Therefore, incorporating interaction features from the educational platform can help enrich the input. Table 5 shows that interaction features play a key role in the overall model, so we conduct additional experiments to analyze the effect of each interaction feature. Notably, *counts of attempt* is not available in Algebra. Table 6 shows the results, *difficulty* and *response time* directly reflect the students' knowledge states and difficulty of the questions, which largely determine whether students can answer correctly. Implementing these two attributes can enhance the performance of our model significantly.

N-WD is IFKT without weights distinction, which uses the original self-attention mechanism. Results confirm our intuition that it is necessary to differentiate the assignments of weights, considering that students may forget during the learning process.

4.2.5 Parameter analysis

To investigate the sensitivity of IFKT, we evaluate the impact of key hyperparameters on its performance. Specifically, we consider the hyperparameter of the maximal aggregate layer K in LightGCN, which is carried out on all bench datasets. K ranges from 0 to 6, results are shown in Fig. 6. Notably, when the number of the layer is 0, it means the question embeddings and skill embeddings used in our model are indexed from embedding matrices directly. The performance of IFKT improves gradually when K increases from 0 to 3, but overly increasing the maximal aggregate layer could stagnate or even deteriorate the performance. This validates the effectiveness of LightGCN in obtaining embeddings of both questions and skills as model inputs. To avoid model complexity and overfitting, we set K to 3.

Given that the length of students' response sequences varies widely across datasets, after grouping each dataset by *user_id*, the maximum length l is fixed at an average of all users' series. To demonstrate the benefits of this modification, we simulate the performance of IFKT and some baselines set to the default length and the average length separately on Algebra. For comparison, the default length of IFKT and IFKT* is set to 200 here. The results are shown in Fig. 7, the models fit better because longer sequence information for each user is made available.

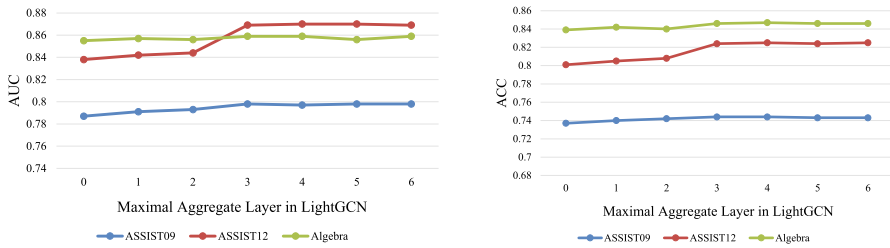


Fig. 6 Impact of the different maximal aggregate layer in LightGCN

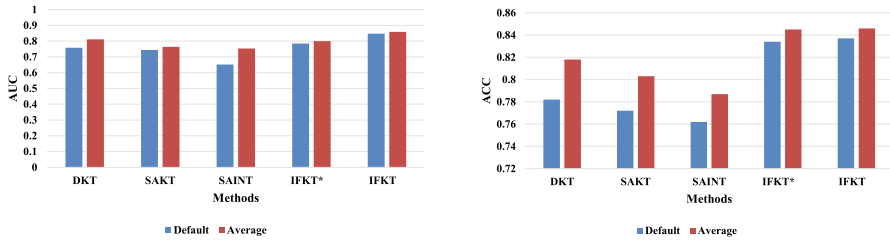


Fig. 7 Impact of different maximum length

4.2.6 Visualization

To better understand the ability of IFKT to represent students' interaction records, we visualize the embeddings for capturing potential information of questions and skills generated from LightGCN. Firstly, we randomly chose 200 questions and the whole 101 skills from ASSIST09. Then, the embeddings of questions and skills are projected into two-dimensional space using the t-SNE algorithm, the x-axis and y-axis represent two new feature dimensions for the data points. These dimensions do not have a direct correspondence to the features in the original data but strive to preserve the similarity relationships between data points as much as possible. Using t-SNE for dimensionality reduction allows us to visually observe how embeddings cluster and interact in a two-dimensional plane. Similar embeddings tend to cluster together, while dissimilar embeddings remain at a greater distance. To illustrate, in Fig. 8a, each point represents a question embedding, and different colors are used to distinguish between different questions. Some questions in the original data are similar, and after the mapping, their distances in the two-dimensional representation are relatively close, whereas questions that are less related are positioned farther apart. Similarly, in Fig. 8b, each point represents a skill embedding, and distinct colors are utilized to differentiate between different skills. Similar skills cluster together in the two-dimensional visualization, while dissimilar skills are separated by considerable distances. This confirms LightGCN's ability to catch potential information among items, such as question similarity and skill similarity.

Figure 9 comprises three subplots extracted from the Assist2009, Assist2012, and Algebra. Within each dataset, we randomly select 200 questions associated with 10 skills and present them visually. The x-axis represents the "skill_id", while the y-axis signifies the difficulty levels of the questions linked to each respective "skill_id". Each

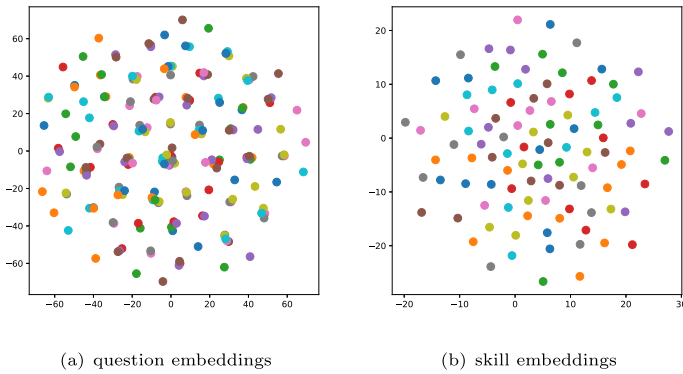


Fig. 8 Question embeddings and skill embeddings from t-SNE on ASSIST09

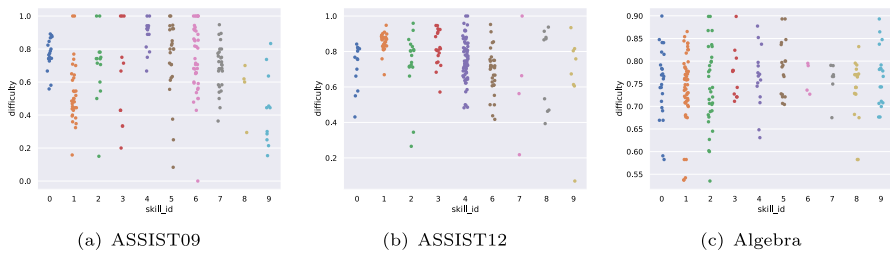


Fig. 9 Difficulty distribution of questions involving the same skill

data point in the plot corresponds to an individual question, and distinct colors are employed to differentiate among the various questions. As depicted in each subplot, each skill is associated with multiple questions, but there exists variability in the difficulty levels among questions related to the same skill. This phenomenon validates our standpoint regarding the incapability of skills to substitute questions as model inputs.

5 Conclusion and future work

In this paper, we propose a new model for knowledge tracing task, named IFKT. IFKT explores the three types of relationship information for questions and skills with a light graph convolutional network, obtaining both question embedding and skill embedding as model input. Then, we differentiate the difficulty diversity among questions and add other interaction features from the online educational platform, which can help better understand the performance of prediction from both the question's difficulty and student's knowledge states. In addition, we individualize the weights assigned to the current prediction from historical interactions, accounting for the fact that students may forget during the learning process. We perform a comparative study with seven previous methods on three real-world datasets and verify the effectiveness of IFKT for knowledge tracing. We also demonstrate that the modules of question embedding

and skill embedding, interaction features, and weight distinction indeed benefit the performance of knowledge tracing. Specifically, the interaction data between students and educational platforms can reflect students' learning behavior and knowledge proficiency. This interaction data is thoroughly leveraged to gain a better understanding of students' learning processes and accurately predict learning performance. Additionally, it has been observed that new knowledge is acquired by students, and they also experience forgetting of previously learned information, aligning with the Ebbinghaus forgetting curve theory in educational psychology. In practical educational scenarios, these findings can serve as a scientific basis for personalized learning support and learning interventions. Moreover, parameter analysis experiments are performed to show that IFKT has good model stability, and visualization studies are conducted to provide more insights into IFKT.

Finally, addressing the challenges mentioned in the preceding discussion will drive the development of personalized educational models. By effectively addressing data class imbalances and the cold-start problem, these methods can provide students with more accurate and personalized learning experiences while also reducing the burden on educators in real-world educational settings. In addition, IFKT demonstrates only a certain level of interpretability, with certain features reflecting question difficulty and student proficiency. Therefore, further research can enhance the interpretability of personalized educational models by integrating educational theories and domain knowledge for model explanation, thereby providing valuable insights into model decision-making and enhancing the model's transparency and credibility.

Funding The authors thank all the participants in this study for their time and valuable inputs. This work was supported by the National Key Research and Development Program of China under grant number 2023YFC3305704 and National Natural Science Foundation of China under grant number 62377015.

Declarations

Conflicts of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

- Liu, Qi, Shen, Shuanghong, Huang, Zhenya, Chen, Enhong, Zheng, Yonghe: A survey of knowledge tracing. ArXiv preprint, [arXiv:2105.15106](https://arxiv.org/abs/2105.15106), (2021)
- Li, Chencheng, Zhou, Pan, Xiong, Li., Wang, Qian, Wang, Ting: Differentially private distributed online learning. *IEEE Trans. Knowl. Data Eng.* **30**(8), 1440–1453 (2018)
- de Andrade, Tiago Luís, Rigo, Sandro José, Barbosa, Jorge Luis Victória: Active Methodology, Educational Data Mining and Learning Analytics: A Systematic Mapping Study. *Inf. Educat.*, (2021), 20(2)
- Wang, Chao, Zhu, Hengshu, Wang, Peng, Zhu, Chen, Zhang, Xi., Chen, Enhong, Xiong, Hui: Personalized and explainable employee training course recommendations: A bayesian variational approach[J]. *ACM Trans. Inf. Syst. (TOIS)* **40**(4), 1–32 (2021)
- Cui, Chaoran, Yao, Yumo, Zhang, Chunyun, Ma, Hebo, Ma, Yuling, Ren, Zhaochun, Zhang, Chen, Ko, James: DGEKT: A Dual Graph Ensemble Learning Method for Knowledge Tracing. arXiv preprint [arXiv:2211.12881](https://arxiv.org/abs/2211.12881), (2022)
- Corbett, Albert T., Anderson, John R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User-Adap. Inter.* **4**(4), 253–278 (1994)

- Piech, Chris, Bassen, Jonathan, Huang, Jonathan, Ganguli, Surya, Sahami, Mehran, Guibas, Leonidas J., Sohl-Dickstein, Jascha: Deep knowledge tracing. *Adv. Neural Inf. Process. Syst.* 28, (2015)
- Zhang, Jiani, Shi, Xingjian, King, Irwin, Yeung, Dit-Yan: Dynamic Key-Value Memory Networks for Knowledge Tracing. In Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich, editors, *Proceedings of the 26th International Conference on World Wide Web*, Perth, Australia, April 3-7, 2017, pages 765-774. ACM, (2017)
- Yang, Yang, Shen, Jian, Qu, Yanru, Liu, Yunfei, Wang, Kerong, Zhu, Yaoming, Zhang, Weinan, Yu, Yong: GIKT: A graph-based interaction model for knowledge tracing. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 299-315. Springer, (2020)
- Huang, Xiaoqing, Liu, Qi, Wang, Chao, Han, Haoyu, Ma, Jianhui, Chen, Enhong, Su, Yu, Wang, Shijin: Constructing educational concept maps with multiple relationships from multi-source data. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1108-1113, (2019)
- Liu, Yunfei, Yang, Yang, Chen, Xianyu, Shen, Jian, Zhang, Haifeng, Yu, Yong: Improving knowledge tracing via pre-training question embeddings. In Christian Bessiere, editor, *IJCAI-20*, pages 1577-1583. International Joint Conferences on Artificial Intelligence Organization, 7 (2020)
- Su, Yu, Liu, Qingwen, Liu, Qi, Huang, Zhenya, Yin, Yu, Chen, Enhong, Ding, Chris, Wei, Si, Hu, Guoping: Exercise-enhanced sequential modeling for student performance prediction. *The 32nd AAAI Conference on Artificial Intelligence*, pages 2435-2443, (2018)
- Liu, Qi, Zhenya Huang, Yu., Yin, Enhong Chen, Hui Xiong, YuSu., Guoping, Hu.: EKT: Exercise-aware knowledge tracing for student performance prediction. *IEEE Trans. Knowl. Data Eng.* 33(1), 100-115 (2021)
- Cheng, Song, Liu, Qi, Chen, Enhong, Zhang, Kai, Huang, Zhenya, Yin, Yu, Huang, Xiaoqing, Su, Yu: Adaptkt: A domain adaptable method for knowledge tracing[C]//*Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 123-131 (2022)
- Shin, Dongmin, Shim, Yugeun, Yu, Hangeol, Lee, Seewoo, Kim, Byungsoo, Choi, Youngduck: SAINT+: Integrating temporal features for EdNet correctness prediction. In *11th International Learning Analytics and Knowledge Conference*, pages 490-496, (2021)
- Wang, Chenyang, Ma, Weizhi, Zhang, Min, Lv, Chuancheng, Wan, Fengyuan, Lin, Huijie, Tang, Taoran, Liu, Yiqun, Ma, Shaoping: Temporal cross-effects in knowledge tracing. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 517-525, (2021)
- Abdelrahman, Ghodai, Wang, Qing, Nunes, Bernardino Pereira: Knowledge tracing: A survey. *ArXiv preprint, arXiv:2201.06953*, (2022)
- Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N., Kaiser, Łukasz, Polosukhin, Illia: Attention is all you need. In *Advances in neural information processing systems*, pages 5998-6008, (2017)
- Pandey, Shalini, Karypis, George: A self-attentive model for knowledge tracing. *arXiv preprint arXiv:1907.06837*, (2019)
- Choi, Youngduck, Lee, Youngnam, Cho, Junghyun, Baek, Jineon, Kim, Byungsoo, Cha, Yeongmin, Shin, Dongmin, Bae, Chan, Heo, Jaewe: Towards an appropriate query, key, and value computation for knowledge tracing. *L@S '20*, page 341-344, (2020)
- Sense, Florian, Behrens, Friederike, Meijer, Rob R., van Rijn, Hedderik: An individual's rate of forgetting is stable over time but differs across materials[J]. *Top. Cogn. Sci.* 8(1), 305-321 (2016)
- Ebbinghaus, Hermann: Memory: A contribution to experimental psychology. *Ann. Neurosci.* 20(2013), 155 (2013)
- Ghosh, Aritra, Heffernan, Neil T., Lan, Andrew S.: Context-aware attentive knowledge tracing. In Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash, editors, *The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Virtual Event, CA, USA, August 23-27, 2020, pages 2330-2339. ACM, (2020)
- Nagatani, Koki, Zhang, Qian, Sato, Masahiro, Chen, Yan-Ying, Chen, Francine, Ohkuma, Tomoko: Augmenting knowledge tracing by considering forgetting behavior. In Ling Liu, Ryen W. White, Amin Mantrach, Fabrizio Silvestri, Julian J. McAuley, Ricardo Baeza-Yates, and Leila Zia, editors, *The World Wide Web Conference*, San Francisco, CA, USA, May 13-17, 2019, pages 3101-3107. ACM, (2019)
- Shen, Shuanghong, Liu, Qi, Chen, Enhong, Huang, Zhenya, Huang, Wei, Yin, Yu, Su, Yu, Wang, Shijin: Learning process-consistent knowledge tracing. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1452-1460, (2021)

- Wang, Chenyang, Ma, Weizhi, Zhang, Min, Lv, Chuancheng, Wan, Fengyuan, Lin, Huijie, Tang, Taoran, Liu, Yiqun, Ma, Shaoping: Temporal cross-effects in knowledge tracing. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pages 517–525, (2021)
- He, Xiangnan, Deng, Kuan, Wang, Xiang, Li, Yan, Zhang, YongDong, Wang, Meng: Lightgcn: Simplifying and powering graph convolution network for recommendation[C]//Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. 639–648 (2020)
- Shaw, Peter, Uszkoreit, Jakob: Ashish Vaswani. Self-attention with relative position representations[J]. arXiv preprint [arXiv:1803.02155](https://arxiv.org/abs/1803.02155), (2018)
- Vie, Jill-Jênn., Kashima, Hisashi: Knowledge tracing machines: Factorization machines for knowledge tracing. Proc. AAAI Conf. Artif. Intell. **33**, 750–757 (2019)
- Wang, Yutao, Heffernan, Neil: Extending knowledge tracing to allow partial credit: Using continuous versus binary nodes[C]//Artificial Intelligence in Education: 16th International Conference, AIED: Memphis, TN, USA, July 9–13, 2013. Proceedings 16. Springer, Berlin Heidelberg **2013**, 181–188 (2013)
- Wu, Runze, Liu, Qi, Liu, Yuping, Chen, Enhong, Su, Yu, Chen, Zhigang, Hu, Guoping: Cognitive modelling for predicting examinee performance[C]//Twenty-Fourth International Joint Conference on Artificial Intelligence. (2015)
- Minn, Sein, Yu, Yi, Desmarais, Michel C., Zhu, Feida, Vie, Jill-Jenn: Deep knowledge tracing and dynamic student classification for knowledge tracing. In 2018 IEEE International Conference on Data Mining (ICDM). IEEE, (nov 2018)
- Pu, Shi, Yudelson, Michael, Ou, Lu, Huang, Yuchi: Deep knowledge tracing with transformers. In International Conference on Artificial Intelligence in Education, pages 252–256. Springer, (2020)
- Wang, Xizhe, Mei, Xiaoyong, Huang, Qionghao, Han, Zhongmei, Huang, Changqin: Fine-grained learning performance prediction via adaptive sparse self-attention networks. Inf. Sci. **545**, 223–240 (2021)
- Kipf, Thomas N., Welling, Max: Semi-supervised classification with graph convolutional networks, in 5th International Conference on Learning Representations, ICLR, (2017)
- Tong, Shiwei, Liu, Qi, Huang, Wei, Huang, Zhenya, Chen, Enhong, Liu, Chuanren, Ma, Haiping, Wang, Shijin: Structure-based Knowledge Tracing: An Influence Propagation View. In 2020 IEEE International Conference on Data Mining, pages 541–550. IEEE, (2020)
- Nakagawa, Hiromi, Iwasawa, Yusuke, Matsuo, Yutaka: Graph-based knowledge tracing: Modeling student proficiency using graph neural network. In 2019 IEEE/WIC/ACM International Conference on Web Intelligence, pages 156–163. IEEE, (2019)
- Zhengyang, Wu., Huang, Li., Huang, Qionghao, Huang, Changqin, Tang, Yong: SGKT: Session graph-based knowledge tracing for student performance prediction[J]. Expert Syst. Appl. **206**, 117681 (2022)
- Ni, Qin, Wei, Tingjiang, Zhao, Jiabao, He, Liang, Zheng, Chanjin: HHSKT: A learner-question interactions based heterogeneous graph neural network model for knowledge tracing[J]. Expert Syst. Appl. **215**, 119334 (2023)
- Liu, Zitao, Liu, Qiongqiong, Chen, Jiahao, Huang, Shuyan, Tang, Jiliang, Luo, Weiqi: pyKT: a python library to benchmark deep learning based knowledge tracing models[J]. Adv. Neural. Inf. Process. Syst. **35**, 18542–18555 (2022)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.