Multiple Graph Knowledge Tracing Based on LSTM-Attention Hypergraph Convolution and Forgetting Effect

Ruichun Kang, Xiaoyao Li, Guiyao Liu, and Lianhong Wang

***Abstract*—Due to the great potential of online education platforms, knowledge tracing (KT) has become popular in personalized learning. KT aims at tracing the dynamic change of knowledge state over time based on student's historical learning trajectory and predicting student's future performance. However, the existing methods still face some challenges, including the ignoring of forgetting behavior, the loss of higher-order information, and the limit of pair-wise representation in student's learning trajectory. To address these issues, we propose the multiple graph knowledge tracing (MGKT). Based on forgetting mechanism, MGKT introduces forgetting feature into a graph convolutional network. Considering the topological ordering and relations of exercise-response and skill-response, a dual-channel directed multigraph communication module is developed for MGKT to characterize student's hidden knowledge states. In addition, we design a hypergraph convolution module with LSTM and attention mechanism for MGKT to learn higher-order semantic information and group-wise relationship in hypergraphs. Comprehensive experiments are performed on three public datasets and the experimental results demonstrate the superiority of MGKT over some state-of-the-art KT models.**

***Index Terms*—Attention mechanism, forgetting effect, graph neural networks (GNNs), knowledge tracing (KT), multiple graph.**

I. Introduction

With the rapid development of modern information technology, the online education platforms gradually become an important supplement to the traditional education patterns and even as the mainstream in some special occasions [1], due to flexible time and space, rich course resources, and powerful learning diagnosis function. To trace the changes of developed the student's knowledge proficiency in the learning process, researchers have proposed a basic task called Knowledge Tracing (KT). As shown in Fig. 1 the KT models trace the change of student's knowledge state over time based on the student's historical learning trajectory to predict the student's future performance. KT serves as a powerful tool in personalized diagnosis, applying education-related data to monitor students' changing knowledge state [2]. It is of great significance to alleviate the problems of "information overload" and "knowledge disorientation" faced in personalized education.

Fig. 1. Simple schematic diagram of knowledge tracing.

Generally, KT methods are categorized into traditional knowledge tracing and deep knowledge tracing. Early research centered around traditional knowledge tracing, and Bayesian-based knowledge tracing [3], [4], [5]. They typically use probabilistic graphical models such as hidden Markov models (HMMs) and Bayesian belief networks (BBNs) to trace the student's dynamic knowledge state during learning process [6]. Compared with HMM, recurrent neural networks (RNN) are more suitable to build complex models due to their high dimensional and continuous representation of latent features [7]. Deep knowledge tracing (DKT) [8] first introduced RNN to capture the interaction between exercises and student's answers. Subsequently, to enhance the external memory structure, Zhang et al. [9] proposed dynamic key–value memory network (DKVMN) using static key matrices and dynamic value matrices to trace the evolutionary trend of student knowledge state over time. In addition, self-attentive knowledge tracing (SAKT) [10] first integrated attention mechanisms with knowledge tracing by utilizing multiple attention weights to indicate the importance of historical exercise sequence to the current exercise. From then on, attention mechanisms have drawn extensive attention from researchers in the KT field [11], [12], [13], such as attentive knowledge tracing (AKT) [11].

Recently, a lot of deep learning techniques have been developed to process graph data due to its great expressive power. Therefore, increasing works combine graph structure to

Received 4 November 2024; revised 27 January 2025; accepted 17 March 2025. This work was supported by the National Natural Science Foundation of China under Grant 62377010. *(Ruichun Kang and Xiaoyao Li contributed equally to this work.) (Corresponding author: Lianhong Wang.)*

Ruichun Kang, Guiyao Liu, and Lianhong Wang are with the College of Electrical and Information Engineering, Hunan University, Changsha 410082, China (e-mail: kangruichun@hnu.edu.cn; lgyswyyx@163.com; 292386791@qq.com).

Xiaoyao Li is with the College of Advanced Interdisciplinary Studies, Central South University of Forestry and Technology, Changsha 410082, China (e-mail: lxy0731@csuft.edu.cn).

Digital Object Identifier 10.1109/TCSS.2025.3554594

construct architecture dealing with various challenges in KT task. Graph-based knowledge tracing (GKT) [14] converts knowledge structure into diagrams solved by standard diagram learning techniques. With the assumption of the one-to-many relationship between skills and exercises, graph-based interactive knowledge tracing (GIKT) [15] applies graph convolutional network (GCN) for embedding characterization. Based on joint GCN, joint knowledge tracing (JKT) [16] not only establishes connections between exercises under cross-skills, but also captures high-level semantic information. In addition, Wu et al. [17] developed the session graph based knowledge tracing (SGKT) based on exercise session graphs to obtain student's knowledge states and embedding representations.

By reviewing and analyzing previous works on knowledge tracing, we find some remaining challenges as follows.

1) Since forgetting behaviors have significant impact on student's knowledge proficiency and future learning performance, there have been increasing studies incorporating forgetting effect into KT models. However, they are mainly sequence based KT models. Although a few graph-based models consider the forgetting effect, their prediction performance is not satisfying due to the mismatching of forgetting rules to complex topological structure of graph. Therefore, *how to effectively integrate forgetting features with graph structure* becomes a key to improve the model performance.
2) The student's interaction behaviors can lead to various graph structures [18]. However, most graph-based KT methods focus only on the node feature of exercises or skills, while ignoring the hidden semantic information among the two and the corresponding responses. In addition, a few approaches have introduced the hidden associations but fail to consider the influence of ordering in student's response sequence. Hence, it is a difficulty for KT models to *fully capture the relationships among exercises, skills and responses*.
3) With various graph types brought into KT field, *the intra-graph, intergraph, and cross-graph type information passing and fusion* turn into an important problem to be faced. In practice, an exercise can cover multiple skills and a skill is usually required by different exercises. Moreover, the learning trajectory varies by students. Accordingly, simple pairwise dependency and single graph structure applied by traditional graph based (e.g. graph neural networks) are not enough to characterize such higher-order correlations in student's exercise-response sequence [19].

To address the above issues, we propose a multiple graph knowledge tracing (MGKT) based on LSTM-Attention hypergraph convolution and forgetting effect. Our main contributions are summarized as follows.

1) Based on the forgetting gating mechanism, we construct a heterogeneous forgetting GCN (HFGCN) for MGKT to integrate the forgetting features in student's learning trajectory with the knowledge state information in heterogeneous graph.
2) A dual-channel directed multigraph communication (DcDMC) module is first introduced in MGKT to utilize topological ordering and correlations of both exercise-response and skill-response sequences to jointly characterize student's hidden knowledge states.
3) Taking advantage of LSTM and attention mechanism, we design a hypergraph convolution module named LSTM-hypergraph-attention convolution (LHAConv), and propose multiple graph joint association matrix to help MGKT learn higher-order semantic information and the group-wise relationship in student's learning process and improve the sensitivity of convolution networks to different hyperedges.
4) We compare our proposed MGKT model with several state-of-the-art deep learning-based KT models on three public datasets and extensive experiments demonstrate the superior performance and interpretability of MGKT.

Furthermore, our work contributes not only to the theoretical development of educational data mining but also to the practical application in education platforms (e.g. MOOC, Coursera, and Khan Academy). For instance, by capturing and learning the higher-order semantic information from student's learning trajectory, MGKT can help the platforms recommend more targeted learning schedule and exercise library for students with different levels of knowledge proficiency and forgetting rates predicted by MGKT, enable real-time adjustment to the learning material, and therefore improve student's learning efficiency and effect.

The remainder of the article is organized as follows. Section II reviews the related works on knowledge tracing. Section III describes the preliminary and specifies the overall framework of MGKT. Section IV gives the experimental results of MGKT. Finally, Section V concludes the article.

II. Related Work

A. Knowledge Tracing

Previous research has centered around traditional knowledge tracing, which consists of probabilistic and logistic models. Bayesian inference models [3] are the most representative probabilistic KT methods utilizing HMMs to track students' knowledge states. Logistic models are a class of models that estimate student performance by learning logistic functions, such as performance factor model [20], knowledge tracing machines [21], etc. Inspired by the ability of deep learning to extract hidden features of sequences, DKT [8] first utilizes RNN and its variant LSTM to encode learning sequences to obtain student's hidden knowledge states. DKT+ [22], an extended variant of DKT, utilizes two additional regular terms to address the limitations of DKT by reconstructing answer inputs and improving the consistency of skill-sharing exercises in terms of prediction. To explore the complex knowledge structures of student, DKVMN [9] uses static key matrices and dynamic value matrices to trace the change of student's knowledge state. In addition, since attention mechanisms can reflect the extent to which different exercises affect the corresponding response, SAKT [10] first integrates the scaled dot product attention mechanism in the KT model to capture the relationships among student's historical interactions. Convolutional knowledge tracing (CKT) [23]

further combines attention with a one-dimensional convolution network to model individualized student's learning state. With the hot research on graph neural networks, researchers have gradually noticed the excellent characterization ability of graph structures in the field of knowledge tracing. GKT [14] uses nodes to represent skills and edges to represent dependencies between skills and thus transforms the KT problem into a time-series node classification problem by standard graph learning techniques.

B. Forgetting Effect

From the perspective of experimental psychology, student's knowledge proficiency will decline due to forgetting [24]. Therefore, repeated learning can consolidate previously learned knowledge. Based on this phenomenon, some researchers have considered the effect of forgetting in KT model. The DKT+Forget [25] extends the DKT model by adding students' forgetting characteristics. Specifically, it adds time intervals and number of attempts to trace student's knowledge states. Wang et al. [26] introduced Hawkes process in KT model and argued that there are differences in the degree of forgetting for different skills. Abdelrahman and Wang [27] proposed the deep graph memory network incorporating a forgetting gating mechanism in the attention memory structure to dynamically capture forgetting behavior during knowledge tracing. After that, Hen et al. [2] proposed a learning process in consistent knowledge tracing utilizing the positive effect of learning gains and the negative effect of learning forgetting to assess the student's learning progress in continuous learning interactions.

C. Graph Neural Networks

Due to the multivariate node and edge structure inherent in graph structures, graphs provide a more intuitive representation of relationships between entities than sequence structures. To deal with graph structures, Scarselli et al. [28] proposed graph neural networks (GNN). It extends existing neural network approaches to process graph-structured data. Subsequently, Kipf and Welling [29] proposed GCN for semisupervised learning of graph-structured data, node classification, and graph classification problems. Graph attention networks [30] utilize a masked self-attention mechanism to assign different weights to different nodes in the neighborhood, thus obtaining the importance of each node at different levels.

D. Hypergraph Learning

Hypergraph structures have attracted much attention for their flexibility in modeling complex data associations. Zhou et al. [31] proposed hypergraphs for representing complex relationships among objects of interest. Feng et al. [32] designed a hyperedge convolution to learn hidden representations when considering higher-order data structures. Ji et al. [33] proposed jumping hypergraph convolutions to support display embedding propagation of higher-order correlations in the dual channel hypergraph collaborative filtering model. Gao et al. [34] proposed tensor-based dynamic hypergraph learning method tensor representation to describe dynamic hypergraphs. Cui et al. [35] established a dual graph ensemble learning method for knowledge tracing (DGEKT) to capture the heterogeneous exercise-skill connection and interaction transition by hypergraph and directed graph structure, respectively.

TABLE I
NOTATION SUMMARY

Notations	Descriptions
Θ, Σ, P	The set of exercises, skills and responses.
M	The total number of exercises.
N	The total number of skills.
S	The total number of students.
$\boldsymbol{q}_k$,$\boldsymbol{s}_p$	The exercise/skill node.
$\boldsymbol{f}_{ji}$	The forgetting feature vector.
$\boldsymbol{g}_{s,j}$	The directed connections between skills and responses of the jth student.
$\boldsymbol{g}_{q,j}$	The directed connections between exercises and responses of the jth student.
$\boldsymbol{\beta}_{s,j}$, $\boldsymbol{\beta}_{q,j}$	The directed multigraph feature of the jth student passing through the fully connected layer.
$\boldsymbol{\alpha}_j$	The result after aggregating the graph communication layers.
$\mathbf{H}$	The association matrix of hypergraph.
$\mathbf{H}_M$	The multiple graph joint association matrix.
$\mathbf{D}$, $\mathbf{M}$	The node degree matrix and hyperedge degree matrix obtained by $\mathbf{H}$.
$\mathbf{D}_M$, $\mathbf{B}_M$	The node degree matrix and the hyperedge degree matrix obtained by $\mathbf{H}_M$.
$\boldsymbol{X}_j$	The hypergraph convolution layer.
$\boldsymbol{a}_j$	The attention weight of hyperedge.
γ_d, γ_x, γ_h	The outputs of the DcDMC, HFGCN, and LHAConv.
$p_{Q'}$	The prediction of student future performance.
$y_{Q'}$	The actual results of student future performance.
$(\cdot)^{\mathrm{T}}$	The transpose operator.
$\oplus$	The element-wise addition.
$\odot$	The element-wise multiplication.
$\sigma(\cdot)$	The sigmoid activation function.
$\Sigma(\cdot)$	The sum of elements.

III. PROPOSED METHOD

A. Preliminary

In this section, we first give the mathematical notation used in this article as shown in Table I and some definitions related to multigraph knowledge tracing.

Definition 1 (Knowledge Tracing): The task of knowledge tracing [2] is to predict the probability that a student correctly answers the next exercise on a particular skill based on his historical learning trajectory. In the knowledge tracing, the set of exercises is denoted as $\mathcal{Q} = \{q_1,q_2,\ldots,q_M\}$, M represents the total number of exercises. The set of skills is denoted as $\mathcal{S} = \{s_1,s_2,\ldots,s_N\}$, N represents the total number of skills. $\mathcal{R} = \{r_1,r_2,\ldots,r_M\}$ denotes the set of students' responses, and the binary variable $r_i \in \{0,1\}$ $(i = 1,2,\ldots,M)$ means the score (i.e. 1 means correct and 0 means wrong) of the ith exercise.

Definition 2 (Forgetting Gating Mechanism): The forgetting behavior [27] denotes the declining trend in student's knowledge proficiency of a specific skill since the last practice, due to human memory decay in cognitive science. The forgetting gating mechanism [27] can be formulated as follows:

$$\boldsymbol{f}_{\text{out}} = \text{Tanh}(\boldsymbol{f}_{\text{in}}\boldsymbol{W} + \boldsymbol{b}) \tag{1}$$

where forgetting gating function $\text{Tanh}(x) = (e^x - e^{-x})/(e^x + e^{-x})$. $\boldsymbol{f}_{\text{out}} \in \mathbb{R}^{1\times d_{\text{out}}}$ and $\boldsymbol{f}_{\text{in}} \in \mathbb{R}^{1\times d_{\text{in}}}$ denote the forgetting vector and

input forgetting feature, respectively. A higher value in $\boldsymbol{f}_{\text{out}}$ denotes greater effect of the input feature on the current node. $\boldsymbol{W} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$ and $\boldsymbol{b} \in \mathbb{R}^{1 \times d_{\text{out}}}$ are trainable weight matrix and bias, respectively. d_{in} and d_{out} are latent dimensions.

Definition 3 (Heterogeneous Information Graph): The heterogeneous information graph [22] is denoted as $\mathcal{G}_H = (V_H, \varepsilon_H)$, where V_H is the node set and ε_H is the edge set. There is a node type mapping function φ_H: $V_H \rightarrow \Gamma_V$ and an edge type mapping function ψ_H: $\varepsilon_H \rightarrow \Gamma_\varepsilon$, where Γ_V and Γ_ε denote the node type set and the edge type set, respectively.

Definition 4 (Directed Multigraph): A directed multigraph [35] $\mathcal{G}_d = (V_d, \varepsilon_d)$ consists a node set V_d and a directed edge set ε_d. Each directed edge connects two ordered nodes. There is a mapping function f: $\varepsilon_d \rightarrow V_d \times V_d = \{(u, v) | u, v \in V_d\}$. Two edges ε_1 and $\varepsilon_2 \in \varepsilon_d$ are parallel edges (or multiple edges) if $f(\varepsilon_1) = f(\varepsilon_2)$. Edge ε is a self-loop if $f(\varepsilon) = (v, v)$.

Definition 5 (Hypergraph): A hypergraph [22] $\mathcal{G}_h = (V_h, \varepsilon_h)$ is an extension of a simple graph. In the hypergraph, a hyperedge can connect more than two nodes at the same time.

The hypergraph is associated with an association matrix $\mathbf{H}$ with the size of $|V_h| \times |\varepsilon_h|$. If hyperedge ε contains node ν, $\mathbf{H}(\nu, \varepsilon) = 1$, otherwise $\mathbf{H}(\nu, \varepsilon) = 0$. The diagonal node degree matrix $\mathbf{D}$ and the diagonal hyperedge degree matrix $\mathbf{B}$ are defined as

$$\mathbf{D}(v, v) = \sum_{\varepsilon=1}^{|\mathcal{E}_h|} \mathbf{H}(v, \varepsilon)\mathbf{W}(\varepsilon, \varepsilon) \tag{2}$$

$$\mathbf{B}(\varepsilon, \varepsilon) = \sum_{v=1}^{|V_h|} \mathbf{H}(v, \varepsilon) \tag{3}$$

where $\mathbf{W}$ is a diagonal weight matrix.

Definition 6 (Multiple Graph Joint Association Matrix): Suppose $\mathcal{G}_d = (V_d, \varepsilon_d)$ is a directed multigraph based on a given heterogeneous information graph $\mathcal{G}_H = (V_H, \varepsilon_H)$ and the corresponding node type mapping function φ_H and edge type mapping function ψ_H. The related hypergraph can be derived as $\mathcal{G}_h = (V_h, \varepsilon_h)$, where $V_h = V_d$. A joint association matrix $\mathbf{H}_M \in \mathbb{N}^{|V_h| \times |\varepsilon_h|}$ is used to represent connections between nodes and hyperedges. If hyperedge ε contains node $\nu \in V_h = V_d$, $\mathbf{H}_M(\nu, \varepsilon) = d^-(v)$, otherwise $\mathbf{H}_M(\nu, \varepsilon) = 0$. $d^-(v)$ denotes the in-degree (i.e. the number of edges coming into a node) of node $\nu \in V_d = V_h$ in the directed multigraph $\mathcal{G}_d$. The corresponding diagonal node degree matrix $\mathbf{D}_M \in \mathbb{R}^{|V_h| \times |V_h|}$ and the diagonal hyperedge degree matrix $\mathbf{B}_M \in \mathbb{N}^{|\varepsilon_h| \times |\varepsilon_h|}$ are also defined as (2) and (3) with $\mathbf{H}$ substituted by $\mathbf{H}_M$.

Take Fig. 2 for example.

1) Given $V_H = \{t_1, t_2, q_1, q_2, q_3, q_4, q_5, s_1, s_2, s_3, s_4\}$ and the related heterogeneous information graph $\mathcal{G}_H = (V_H, \varepsilon_H)$ in Fig. 2(b). Student t_1 answered exercise q_1, q_2, q_3, q_4, q_5, and student t_2 answered exercise q_2, q_4, q_5. Exercise q_1 requires skill s_1, q_2 requires skill s_1 and s_3, q_3 requires skill s_3, q_4 requires skill s_3 and s_4, and q_5 requires skill s_2 and s_4. Accordingly, the node type set and edge type set can be denoted as $\Gamma_V = \{\text{student, exercise, skill}\}$ and $\Gamma_\varepsilon = \{\text{answered, requires}\}$, respectively.
2) Suppose student t_1 answered the five exercises in Fig. 2(b) in order as $(q_1,\times) \rightarrow (q_2,\times) \rightarrow (q_3,\times) \rightarrow (q_4,\surd) \rightarrow (q_5,\surd) \rightarrow (q_2,\times) \rightarrow (q_2,\surd)$, where tuple $(q_{k_{q,ji}}, r_{k_{r,ji}})(i = 1,2,\ldots,K_{q,j}, j \in [1, S])$ consists of exercise $q_{k_{q,ji}}$ and the corresponding response $r_{k_{r,ji}}$. $k_{q,ji}$, $k_{r,ji} \in [1, M]$, $r_{k_{r,ji}} \in \{0, 1\}$, $j = 1$ and $M = 5$ in this case. Let $V_{d,q} = \left\{\left(q_{k_{q,ji}}, r_{k_{r,ji}}\right) | i = 1, 2, \ldots, K_{q,j}\right\}$, the exercise-response sequence can be transformed into a directed multigraph $\mathcal{G}_{d,q} = (V_{d,q}, \varepsilon_{d,q})$ shown in Fig. 2(c). As can be seen, there are six different tuples in the exercise-response sequence and thus $K_{q,j} = 6$ in this case. In addition, the corresponding node multiset of the exercise-response sequence can be denoted as $\overline{V}_{d,q} = \{(q_{\overline{k}_{q,ji}}, r_{\overline{k}_{r,ji}}) | i = 1, 2, \ldots, Q_j\}$, where $\overline{k}_{q,ji}, \overline{k}_{r,ji} \in [1, M]$ and $r_{\overline{k}_{r,ji}} \in \{0, 1\}$. Q_j denotes the length of the exercise-response sequence and thus $Q_j = 7$ for student t_1 in this case. Considering the relationships between exercises and skills in Fig. 2(b), we obtain the skill-response sequence as $(s_1,\times) \rightarrow (s_1,\times) \rightarrow (s_3,\times) \rightarrow (s_1,\times) \rightarrow (s_3,\times) \rightarrow (s_4,\times) \rightarrow (s_3,\surd) \rightarrow (s_2,\surd) \rightarrow (s_4,\surd) \rightarrow (s_1,\times) \rightarrow (s_3,\times) \rightarrow (s_1,\surd) \rightarrow (s_3,\surd)$, and then derive the corresponding directed multigraph $\mathcal{G}_{d,s} = (V_{d,s}, \varepsilon_{d,s})$ with $V_{d,s} = \left\{\left(s_{p_{s,ji}}, r_{p_{r,ji}}\right) | i = 1, 2, \ldots, K_{s,j}\right\}$ in Fig. 2(c). $p_{s,ji} \in [1, N]$, $p_{r,ji} \in [1, M]$, $r_{p_{r,ji}} \in \{0, 1\}$ and here $N = 4$, $K_{s,j} = 7$. Thus, the corresponding node multiset of the skill-response sequence can be denoted as $\overline{V}_{d,s} = \{(s_{p'_{s,ji}}, r_{p'_{r,ji}}) | i = 1, 2, \ldots, E_j\}$, where $p'_{s,ji} \in [1, N]$, $p'_{r,ji} \in [1, M]$ and $r_{p'_{r,ji}} \in \{0, 1\}$. E_j denotes the length of the skill-response sequence or the extended exercise-response sequence of the jth student and thus $E_1 = 13$ in this case. Moreover, we can also extend exercise-response sequence as $(q_1,\times) \rightarrow (q_2,\times) \rightarrow (q_2,\times) \rightarrow (q_3,\times) \rightarrow (q_3,\times) \rightarrow (q_3,\times) \rightarrow (q_4,\surd) \rightarrow (q_5,\surd) \rightarrow (q_5,\surd) \rightarrow (q_2,\times) \rightarrow (q_2,\times) \rightarrow (q_2,\surd) \rightarrow (q_2,\surd)$ to align with the skill-response sequence. Similarly, the related node multiset of the extended exercise-response sequence is $\overline{V}'_{d,q} = \{(q_{k'_{s,ji}}, r_{k'_{r,ji}}) | i = 1, 2, \ldots, E_j\}$, where $k'_{s,ji}, k'_{r,ji} \in [1, M]$ and $r_{k'_{r,ji}} \in \{0, 1\}$.
3) Let $V_h = V_{d,q}$ and $\varepsilon_h = V_{d,s}$, the related hypergraph $\mathcal{G}_h = (V_h, \varepsilon_h)$ is constructed in Fig. 2(d). Furthermore, we can also compute the corresponding joint association matrix $\mathbf{H}_{M,j}$, node degree matrix $\mathbf{D}_{M,j}$ and hyperedge degree matrix $\mathbf{B}_{M,j}$ of the jth student as shown in the left part of Fig. 2(d) by (2) and (3).

Definition 7 (Hypergraph Attention): Hypergraph attention [34] can capture higher-order interactions between nodes within a hyperedge by computing attention scores. Suppose embedded node matrix $\boldsymbol{X} \in \mathbb{R}^{P \times d}$ for a given hypergraph, the related scaled dot-product attention [10] is defined as follows:

$$\text{Attention}(\boldsymbol{X}) = \text{Softmax}\left(\frac{(\boldsymbol{X}\boldsymbol{W}_Q)(\boldsymbol{X}\boldsymbol{W}_K)^{\text{T}}}{\sqrt{d_K}}\right)(\boldsymbol{X}\boldsymbol{W}_V) \tag{4}$$

where trainable weights $\boldsymbol{W}_Q$, $\boldsymbol{W}_K \in \mathbb{R}^{d \times d_K}$, $\boldsymbol{W}_V \in \mathbb{R}^{d \times d_V}$ are the query, key, and value projection matrices, respectively. They linearly may $\boldsymbol{X}$ to different space. P is the total number of nodes in the hypergraph and d, d_K, d_V are the latent dimensions.

As shown in Fig. 3, the correlation matrix $\boldsymbol{Q}\boldsymbol{K}^{\text{T}} = (\boldsymbol{X}\boldsymbol{W}_Q)(\boldsymbol{X}\boldsymbol{W}_K)^{\text{T}}$ is first calculated to indicate the attention score between

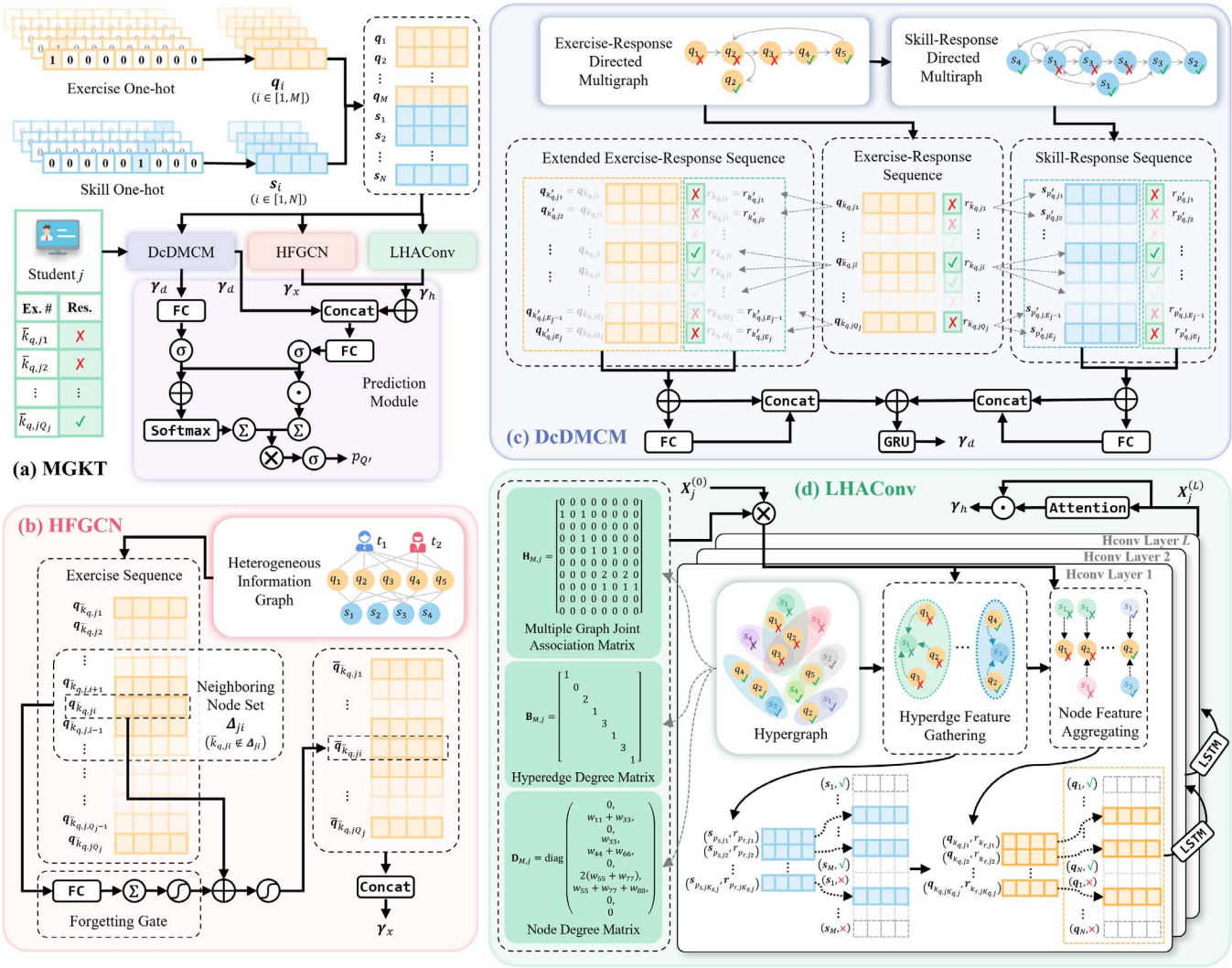

Fig. 2. Structure of MGKT. (a) MGKT. (b) HFGCN. (c) DcDMCM. (d) LHAConv.

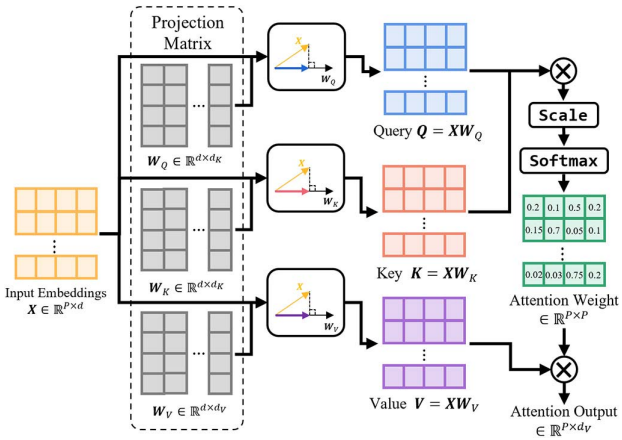

Fig. 3. Flowchart of self-attention mechanism.

nodes in the hypergraph. Then, the dot product is scaled by $1/\sqrt{d_K}$ and a softmax function is applied to normalize the attention weights on each value in $\boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_V$. Finally, each node embedding is updated by a weighted sum of the values from all nodes.

B. Algorithms

As shown in Fig. 2, the four components of MGKT are the heterogeneous forgetting GCN, the dual-channel directed multi-graph communication module, the LSTM-hypergraph-attention convolution module, and the prediction module.

1) Heterogeneous Forgetting Graph Convolutional Network (HFGCN): Following [26], the forgetting behavior diagnosis is not only influenced by time interval but also other two factors as follows, i.e. temporal cross-effect and occasional interference.

a) Temporal cross-effect indicates that the knowledge state (or exercise mastery) of a target skill is influenced by both the previous interactions of the same skill and those of the other skills. As shown in Fig. 4(a), the black line denotes the dynamic change of knowledge state, the blue line denotes the moment an interaction occurs, and the dashed line denotes the steady state after the interaction completes. For example, although the student wrongly answers the exercise of target skill $\triangle$ at the moment t_2 and breaks off the related exercises of the same skill for a long time, its knowledge state still shows upward trend due to temporal cross-effect.

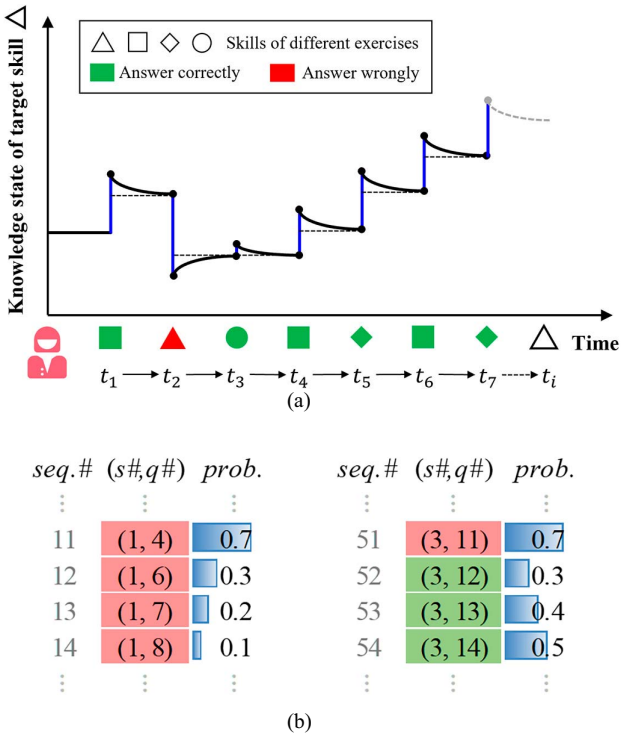

Fig. 4. Influence of: (a) temporal cross-effect; and (b) occasional interference on forgetting behavior diagnosis.

b) Occasional interference denotes that one bad performance in exercise may not result from forgetting behavior but other occasional factors, such as nervousness and carelessness. Fig. 4(b) gives an example to illustrate occasional interference case via the probabilities (*prob.*) of correct answers by some KT model. Each interaction consists of sequence number (*seq.*#), skill number (*s*#), exercise number (*q*#) and the corresponding response. The green denotes the right answer and the red denotes the wrong. As can be seen in the left part of Fig. 4(b), if a student performs poorly on a succession of exercises of same skill, then the student may have indeed forgotten the skill. And the declining prob. actually reflects the continuity of forgetting effect. Otherwise, as shown in the right part of Fig. 4(b), the student may just lose the point by accident but have mastery of the skill. However, the prob. of q_{12} by the KT model is greatly influenced by the previous bad interaction on q_{11}, while receiving no feedback from the subsequent good interactions with the same skill s_3.

Hence, considering the above respects, we introduce the neighboring nodes to update the forgetting feature of the current node in an exercise sequence by forgetting gating mechanism as defined in Definition 7.

In Fig. 2(b), a heterogeneous information graph is applied to illustrate the relationships among students, exercises, and skills. Let $\{\boldsymbol{q}_{\overline{k}_{q,ji}} | i = 1, 2, ..., Q_j\}$ denotes the node multiset corresponding to the finished exercise sequence by the jth ($j \in [1, S]$) student, the related forgetting feature vector of the current node $\boldsymbol{q}_{\overline{k}_{q,ji}}$ ($i \in [1, Q_j]$) is calculated as follows:

$$\boldsymbol{f}_{ji} = \mathrm{Tanh}\left(\frac{1}{|\Delta_{ji}|}\sum_{\substack{\overline{k}_{q,jl} \in \Delta_{ji} \\ \overline{k}_{q,ji} \notin \Delta_{ji}}} \boldsymbol{q}_{\overline{k}_{q,jl}} \boldsymbol{W} + \boldsymbol{b}\right) \tag{5}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are the trainable weight and bias, respectively. $\boldsymbol{q}_{\overline{k}_{q,jl}}$ represents neighboring nodes. $\boldsymbol{\Delta}_{ji}$ ($\overline{k}_{q,ji} \notin \Delta_{ji}$) denotes a set of the neighboring nodes of $\boldsymbol{q}_{\overline{k}_{q,ji}}$. Q_j is the length of exercise sequence of the jth student. Then, the updated node $\overline{\boldsymbol{q}}_{\overline{k}_{q,ji}}$ can be represented as

$$\overline{\boldsymbol{q}}_{\overline{k}_{q,ji}} = \mathrm{Tanh}\left(\boldsymbol{f}_{ji} \oplus \boldsymbol{q}_{\overline{k}_{q,ji}}\right). \tag{6}$$

Finally, the output of the HFGCN is obtained as

$$\gamma_x = \mathrm{Concat}\left(\overline{\boldsymbol{q}}_{\overline{k}_{q,j1}}, \overline{\boldsymbol{q}}_{\overline{k}_{q,j2}}, ..., \overline{\boldsymbol{q}}_{\overline{k}_{q,jQ_j}}\right). \tag{7}$$

2) Dual-Channel Directed Multigraph Communication (DcDMC) Module: The connection between exercise-response and skill-response is also one of the important factors for knowledge tracing [35]. However, single exercise-response or skill-response is not enough to characterize the student knowledge structure. Therefore, we construct a dual-channel directed multigraph communication module to capture the connection.

First, we use the exercise-response sequence and corresponding skill-response sequence to construct directed multigraphs as shown in Fig. 2(c). The directed multigraph propagation process can be formalized as follows:

$$\begin{cases} \boldsymbol{g}_{q,j} = (\boldsymbol{q}_{k'_{q,j1}}, \boldsymbol{q}_{k'_{q,j2}}, ..., \boldsymbol{q}_{k'_{q,jE_j}}) \oplus (r_{k'_{r,j1}}, r_{k'_{r,j2}}, ..., r_{k'_{r,jE_j}}) \\ \boldsymbol{g}_{s,j} = (\boldsymbol{s}_{p'_{s,j1}}, \boldsymbol{s}_{p'_{s,j2}}, ..., \boldsymbol{s}_{p'_{s,jE_j}}) \oplus (r_{p'_{r,j1}}, r_{p'_{r,j2}}, ..., r_{p'_{r,jE_j}}) \end{cases} \tag{8}$$

where $\overline{V}'_{d,q} = \left\{\left(\boldsymbol{q}_{k'_{q,ji}}, r_{k'_{r,ji}}\right) | i = 1, 2, ..., E_j\right\}$ is the extended exercise-response sequence and $\overline{V}_{d,s} = \left\{\left(\boldsymbol{s}_{p'_{s,ji}}, r_{p'_{r,ji}}\right) | \ i = 1, 2, ..., E_j\right\}$ is the related skill-response sequence of the jth student, where $k'_{q,ji}$, $k'_{r,ji}$, $p'_{r,ji} \in [1, M]$, $p'_{s,ji} \in [1, N]$ and $r_{k'_{r,ji}}, r_{p'_{r,ji}} \in \{0, 1\}$. E_j denotes the length of the extended exercise-response sequence or the corresponding skill-response sequence.

Then, we employ a graph communication layer integrates $\boldsymbol{g}_{s,j}$ and $\boldsymbol{g}_{q,j}$ of the jth student as

$$\boldsymbol{\beta}_{s,j} = \boldsymbol{g}_{s,j}\boldsymbol{W}_{s,j} + \boldsymbol{b}_{s,j} \tag{9}$$

$$\boldsymbol{\beta}_{q,j} = \boldsymbol{g}_{q,j}\boldsymbol{W}_{q,j} + \boldsymbol{b}_{q,j} \tag{10}$$

$$\boldsymbol{\alpha}_j = \mathrm{Concat}\left(\boldsymbol{\beta}_{s,j}, \boldsymbol{\beta}_{q,j}\right) \oplus \mathrm{Concat}\left(\boldsymbol{g}_{s,j}, \boldsymbol{g}_{q,j}\right) \tag{11}$$

where $\boldsymbol{W}_{s,j}$, $\boldsymbol{W}_{q,j}$ are the trainable weights, $\boldsymbol{b}_{s,j}$, $\boldsymbol{b}_{q,j}$ are the trainable biases, $\boldsymbol{\beta}_{s,j}$, $\boldsymbol{\beta}_{q,j}$ are directed graph features through the fully connected layer and $\boldsymbol{\alpha}_j$ is the result after aggregating the graph communication layers.

Finally, we initialize $\boldsymbol{\alpha}_j^{(1)} = \boldsymbol{\alpha}_j$ and input it into the gated recurrent unit (GRU) to extract knowledge hidden status $\boldsymbol{h}_j$ of the jth student. In the tth iteration, the update process of the new

knowledge hidden state $\boldsymbol{h}_j^{(t)}$ ($t \in [1, T]$) is formulated as follows:

$$\boldsymbol{z}_j^{(t)} = \sigma\left(\boldsymbol{\alpha}_j^{(t)} \boldsymbol{W}_z + \boldsymbol{h}_j^{(t-1)} \boldsymbol{U}_z\right) \tag{12}$$

$$\boldsymbol{r}_j^{(t)} = \sigma\left(\boldsymbol{\alpha}_j^{(t)} \boldsymbol{W}_r + \boldsymbol{h}_j^{(t-1)} \boldsymbol{U}_r\right) \tag{13}$$

$$\overline{\boldsymbol{h}}_j^{(t)} = \mathrm{Tanh}\left(\boldsymbol{\alpha}_j^{(t)} \boldsymbol{W}_h + \boldsymbol{r}_j^{(t)} \odot \boldsymbol{h}_j^{(t-1)} \boldsymbol{U}_h\right) \tag{14}$$

$$\boldsymbol{h}_j^{(t)} = \boldsymbol{z}_j^{(t)} \odot \boldsymbol{h}_j^{(t-1)} + \left(1 - \boldsymbol{z}_j^{(t)}\right) \odot \overline{\boldsymbol{h}}_j^{(t)} \tag{15}$$

where $\boldsymbol{W}_z$, $\boldsymbol{W}_r$, $\boldsymbol{W}_h$, $\boldsymbol{U}_z$, $\boldsymbol{U}_r$, $\boldsymbol{U}_h$ are trainable weights, $\boldsymbol{z}_j^{(t)}$ denotes the update gate that determines how much of the previous hidden state $\boldsymbol{h}_j^{(t-1)}$ to incorporate into the new hidden state $\boldsymbol{h}_j^{(t)}$. $\boldsymbol{r}_j^{(t)}$ denotes the reset gate that determines how much of the previous hidden state $\boldsymbol{h}_j^{(t-1)}$ to forget. $\overline{\boldsymbol{h}}_j^{(t)}$ is the candidate hidden state.

After total T iterations, the output γ_d of the dual-channel directed multigraph communication module is obtained as

$$\gamma_d = \boldsymbol{h}_j^{(T)}. \tag{16}$$

3) LSTM-Hypergraph-Attention Convolution (LHAConv) Module: Since the common pairwise representation between exercise and skill may lead to information loss in transformation of higher-order correlation into one-to-one correspondence graph, we introduce hypergraph [22] to capture the higher-order relationships between exercises and skills in the dynamic process of student learning trajectory. As shown in Fig. 2(d), for an exercise-response sequence of the jth student, we treat the combination of each exercise and its response $\left(q_{k_{q,ji}}, r_{k_{r,ji}}\right)$ as a node, and the tuple $\left(s_{p_{s,ji}}, r_{p_{r,ji}}\right)$ consisting of skill and the related response as hyperedge since each skill can be covered by multiple exercises.

Due to the limitation of the nonlinear activation function used by traditional hypergraph convolution [32] in complex information exploitation from hypergraph structure, we select LSTM instead and then the $(l + 1)$th hypergraph convolution layer $\boldsymbol{X}_j^{(l+1)}$ ($l \in [0, L\text{-}1]$) is updated as follows:

$$\boldsymbol{X}_j^{(l+1)} = \mathrm{LSTM}\left(\mathbf{D}_{M,j}^{-1/2} \mathbf{H}_{M,j} \mathbf{B}_{M,j}^{-1} \mathbf{H}_{M,j}^{\mathrm{T}} \mathbf{D}_{M,j}^{-1/2} \boldsymbol{X}_j^{(l)} \boldsymbol{\theta}^{(l)}\right) \tag{17}$$

where $\mathbf{D}_{M,j}$ and $\mathbf{B}_{M,j}$ are calculated by (2)–(3) with multiple graph joint association matrix $\mathbf{H}_{M,j}$. $\boldsymbol{\theta}^{(l)}$ indicates trainable parameters. $\boldsymbol{X}_j^{(l)}$ is initialized as $\boldsymbol{X}_j^{(0)} =$ Embedding(Concat($\boldsymbol{q}_1$, $\boldsymbol{q}_2$, …, $\boldsymbol{q}_M$, $\boldsymbol{s}_1$, $\boldsymbol{s}_2$, …, $\boldsymbol{s}_N$)). Hypergraph convolution can perform node-hyperedge-node transform as illustrated in Fig. 2(d). First, the input node embedding matrix $\boldsymbol{X}_j^{(l)}$ is filtered by $\boldsymbol{\theta}^{(l)}$ to extract features. Then, $\mathbf{D}_{M,j}^{-1/2}$ and $\mathbf{H}_{M,j}^{\mathrm{T}}$ are applied to gather the node features and form the hyperedge feature. Finally, $\mathbf{B}_{M,j}^{-1}$, $\mathbf{H}_{M,j}$, and $\mathbf{D}_{M,j}^{-1/2}$ are multiplied to update the node features by aggregating the corresponding hyperedge features. Thus, the hypergraph convolution can capture the complex and higher-order connections in hypergraph.

Considering the different importance of each previous exercise-response interaction to the future exercise performance, we apply the attention mechanism in Definition 7 to obtain the output γ_h of LHAConv module as

$$\gamma_h = \mathrm{Attention}\left(\boldsymbol{X}_j^{(L)}\right) \odot \boldsymbol{X}_j^{(L)}. \tag{18}$$

4) Prediction Module: Integrating the outputs of HFGCN, DcDMC, and LHAConv modules, we predict the probability $p_{Q'}$ of each student correctly answering the next exercise as follows:

$$\boldsymbol{p}_d = \sigma(\gamma_d \boldsymbol{W}_d + \boldsymbol{b}_d) \tag{19}$$

$$\boldsymbol{p}_r = \sigma(\mathrm{Concat}((\gamma_x \oplus \gamma_h), \gamma_d) \boldsymbol{W}_r + \boldsymbol{b}_r) \tag{20}$$

$$p_{Q'} = \sigma\left(\sum \mathrm{Softmax}(\boldsymbol{p}_d \oplus \boldsymbol{p}_r) \sum (\boldsymbol{p}_d \odot \boldsymbol{p}_r)\right) \tag{21}$$

where $\boldsymbol{W}_d$, $\boldsymbol{W}_r$ are trainable weights and $\boldsymbol{b}_d$, $\boldsymbol{b}_r$ are trainable biases.

5) Loss Function: We choose the cross-entropy log loss between the predicted $p_{Q'}$ and the true label $y_{Q'}$ as the objective function

$$\mathrm{Loss} = -\sum_{Q'} [y_{Q'} \log p_{Q'} + (1 - y_{Q'}) \log(1 - p_{Q'})]. \tag{22}$$

The training process of our proposed MGKT model is described in Algorithm 1.

Algorithm 1: MGKT.

Input: The historical learning logs of students
Output: The predicted future performance $p_{Q'}$
Initialize: Weights, bias parameters and embedding matrices
Extract the exercise sequence and the skill sequence for each student from the historical learning logs
for $j \in [1, S]$ **do**
  Select the Q_j exercise-response interactions of the jth student
  for $i \in [1, Q_j]$ **do**
    Get forgetting feature vector $\boldsymbol{f}_{ji}$ by Eq. (5)
    Get updated forgetting node $\overline{\boldsymbol{q}}_{\tilde{k}_{q,ji}}$ by Eq. (6)
  end for
  Get the output γ_x of HFGCN by Eq. (7)
  Get $\boldsymbol{g}_{q,j}$ and $\boldsymbol{g}_{s,j}$ from extended exercise sequence and the skill sequence by Eq. (8)
  Get $\boldsymbol{\alpha}_j$ after aggregating the graph communication layers by Eqs. (9)–(11)
  for $t \in [1,T]$ **do**
    Get the update gate $\boldsymbol{z}_j^{(t)}$, the reset gate $\boldsymbol{r}_j^{(t)}$, the candidate hidden state $\overline{\boldsymbol{h}}_j^{(t)}$ and the new hidden state $\boldsymbol{h}_j^{(t)}$ by Eqs. (12)–(15)
  end for
  Get the output γ_d of DcDMC by Eq. (16)
  for $l \in [0, L\text{-}1]$ **do**
    Get hypergraph convolution layer $\boldsymbol{X}_j^{(l+1)}$ by Eq. (17)
  end for
  Get the output γ_h of LHAConv based on hypergraph by Eq. (18)
  Integrate γ_x, γ_d and γ_h, and predict future performance by Eqs.(19)–(21)
end for

TABLE II
SUMMARY OF DATASETS STATISTICS

Datasets	#Exercise	#Skill	#Student	#Record
ASSIST2009	17 737	123	3852	282 619
EdNet	12 150	189	5000	713 631
ASSIST2017	3162	102	1709	942 816

TABLE III
SUMMARY OF METHODS

Variant Method	Memory Mechanism	Graph Structure	Convolution Network	Attention Mechanism
DKT				
DKT+Forget	√			
DKVMN	√			
CKT			√	√
AKT				√
GIKT		√		
SGKT	√	√		√
DGEKT		√		
MGKT	√	√	√	√

IV. EXPERIMENTS

In this section, we provide comprehensive evaluations of the advantages and performance of the proposed algorithm. There are comparison and ablation experiments on three real-world datasets, module comparison experiments, hyper-parameter sensitivity analysis, and visualization case study.

A. Datasets

Three real-world datasets are used in the experiments. Table II lists the summary of dataset statistics.

1) *ASSIST2009*[1] is a public dataset published by the ASSISTments online education platform during the school year 2009–2010 and is commonly used for accuracy validation of KT models.
2) *EdNet*[2] is a large-scale public dataset provided by an AI-guided learning system Santa and contains 131 441 538 interactions collected from 784 309 students. In our experiments, we randomly choose the response records of 5000 students, which contain 12 150 exercises and 189 knowledge skills.
3) *ASSIST2017*[3] was provided by ASSISTments Data Mining Competition 2017.

B. Baselines

Our MGKT is compared with eight baseline KT modules, including DKT [8], DKVMN [9], CKT [23], DKT+Forget [25], GIKT [15], SGKT [17], DGEKT [35], and AKT [11]. Table III summarizes the modules included by each method. All experiments are conducted in Windows 10 operating system, a host computer with an Intel(R) Core(TM) i9-9900K CPU at 3.60 GHz and 64 GB of RAM.

[1]ASSIST2009:https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data/skill-builder-data-2009-2010
[2]EDNET:https://github.com/riiid/ednet
[3]ASSIST2017:https://sites.google.com/view/assistmentsdatamining/data-mining-competition-2017

For MGKT, we set the embedding size of exercises and skills to 300, the number of aggregation layer of heterogeneous information graphs to 3, the learning rate to 0.00025, and the batch size to 9. All parameter matrices are randomly initialized.

C. Comparison Experiment

Table IV shows the overall performance with respect to the student performance prediction task, where we set the train-test split ratio as 7:3, 8:2, and 9:1, respectively. To provide a robust evaluation of the results, we use the area under the ROC curve (AUC) and accuracy (ACC) and root mean square error (RMSE) to measure the effect. As can be seen from Table IV, the bold and underline fonts respectively denote the best and second performance for each split ratio. The graph-structured methods (i.e., GIKT, SGKT, DGEKT, and MGKT) generally outperform the nongraph-structured methods (i.e., DKT, DKT + Forget, DKVMN, AKT, and CKT) in terms of AUC and ACC on EdNet and ASSIST2017. When the ratio is 8:2, compared with nongraph-structured methods, the AUC of MGKT improves by 8.38%∼14.2% on EdNet, 4.15%∼11.79% on ASSIST2009 and 9.17%∼16.24% on ASSIST2017, respectively. In terms of ACC, MGKT improves its performance on the three datasets by 2.92%∼6.8%, 1.21%∼5.51%, and 5.63%∼7.74%, respectively. Furthermore, MGKT achieves 52.27%, 50.08%, and 50.46% in RMSE on the three datasets, respectively, and shows slightly worse performance than CKT. In addition, compared with the graph-structured methods, the (ACC, AUC, RMSE) of our MGKT significantly outperforms the second-best SGKT model by (2.13%, 3.24%, –2.07%) increase on ASSIST2017 with ratio 8:2, while only (0.53%, 0.76%, –0.51%) increase on EdNet and (0.3%, 0.8%, –0.3%) increase on ASSIST2009, respectively. This is because the high complexity of learning interactions in ASSIST2009 and EdNet is more likely to produce redundant information to interfere with the modeling process, and finally affect the prediction. Overall, our MGKT performs the best on all three public datasets.

D. Ablation Study

The ablation experiments are performed on three datasets to further validate the effectiveness of each module of MGKT. Fig. 5 illustrates the AUC and ACC results with train-test split ratio 8:2. And the other three competing models are outlined as follows.

1) *MGKT-HG* contains only ordinary heterogeneous information GCN.
2) *MGKT-HFG* contains only HFGCN.
3) *MGKT-HFG&DGC* contains only HFGCN and DcDMC.

As shown in Fig. 5, the AUC of MGKT-HFG is improved by 0.09%∼0.78% and the ACC by 0.16%∼0.31% compared with MGKT-HG. It proves the advantages of forgetting mechanism. Furthermore, MGKT-HFG&DGC surpasses MGKT-HFG by 3.30%∼7.82% in AUC and 1.60%∼4.49% in ACC with the help of dual-channel directed multigraph communication module. Finally, the addition of LHAConv improves MGKT-HFG&DGC by 0.68%∼1.22% in AUC and 0.16%∼0.88% in ACC.

TABLE IV
COMPARISON EXPERIMENT RESULTS ON DIFFERENT DATASETS

Model	Ratio	EdNet			ASSIST2009			ASSIST2017			Runtime
		ACC (↑, %)	AUC (↑, %)	RMSE (↓, %)	ACC (↑, %)	AUC (↑, %)	RMSE (↓, %)	ACC (↑, %)	AUC (↑, %)	RMSE (↓, %)	(↓, min)
DKT (2015)	7:3	66.81	62.98	57.61	72.08	71.22	52.84	66.59	65.86	57.88	3
	8:2	65.88	61.87	58.41	72.60	68.12	52.35	66.79	64.94	57.63	
	9:1	68.74	64.21	55.91	72.52	73.27	52.42	66.73	65.00	57.68	
DKT+Forget (2019)	7:3	66.89	65.38	57.54	71.58	66.93	53.31	68.90	72.11	55.77	3
	8:2	67.07	65.62	57.39	72.14	68.13	52.78	68.90	72.01	55.77	
	9:1	69.38	67.58	55.33	72.23	68.28	52.69	68.36	71.77	56.25	
DKVMN (2017)	7:3	64.87	67.68	59.27	72.47	69.99	52.47	67.47	69.43	57.03	3
	8:2	66.34	65.87	58.02	72.84	70.25	52.11	67.68	69.13	56.85	
	9:1	69.29	68.07	55.42	72.90	78.57	52.06	67.35	69.32	57.14	
CKT (2020)	7:3	67.96	66.55	**45.54**	73.13	72.52	**43.04**	68.04	70.23	**45.40**	**2**
	8:2	68.60	66.29	**45.41**	73.49	75.69	**42.53**	67.98	70.10	**45.40**	
	9:1	69.32	66.64	**45.07**	73.87	75.22	**42.32**	68.27	70.25	**45.21**	
AKT (2020)	7:3	69.89	67.69	54.87	72.50	75.93	52.44	66.97	67.27	57.47	7
	8:2	69.70	67.69	55.04	73.70	75.76	51.29	67.39	67.42	57.10	
	9:1	69.96	67.69	54.81	72.21	77.69	52.72	67.57	67.30	56.94	
GIKT (2020)	7:3	71.54	74.09	53.35	73.92	77.94	51.07	72.33	77.86	52.60	8
	8:2	71.61	74.02	53.28	73.84	77.88	51.14	72.35	77.86	52.58	
	9:1	72.27	74.88	52.66	73.14	76.96	51.82	73.04	78.60	51.93	
SGKT (2022)	7:3	71.95	75.08	52.96	74.43	78.96	50.56	72.30	77.80	52.62	21
	8:2	72.15	75.31	52.78	74.61	79.11	50.38	72.40	77.94	52.53	
	9:1	72.84	75.85	52.11	73.58	77.94	51.39	72.61	78.37	52.33	
DGEKT (2024)	7:3	71.11	71.86	53.74	66.44	66.67	57.93	71.78	77.28	53.12	30
	8:2	70.72	71.57	54.11	69.24	69.40	55.32	72.13	77.81	52.70	
	9:1	71.18	71.92	53.69	67.20	67.32	57.27	72.74	78.07	52.21	
MGKT	7:3	**72.65**	**76.20**	52.30	**74.75**	**79.83**	50.25	**73.99**	**80.15**	51.00	58
	8:2	**72.68**	**76.07**	52.27	**74.91**	**79.91**	50.08	**74.53**	**81.18**	50.46	
	9:1	**73.49**	**76.68**	51.49	**75.00**	**79.35**	49.99	**74.52**	**81.17**	50.74	

Note: **Bold** font denotes the best performance, underline font denotes the second.

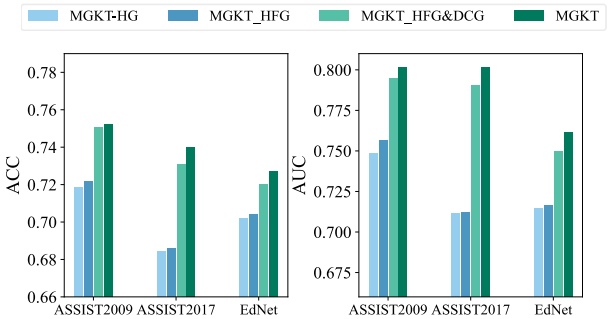

Fig. 5. AUC and ACC of ablation experiment on different datasets.

E. Runtime Comparison

The last column in Table IV lists the average running time per epoch for all the methods on ASSIST2017. As can be seen, the runtime cost of graph-structured KT methods is generally higher than the nongraph-structured ones. Moreover, the runtime of MGKT is longer than other competing methods. By analyzing the procedure, the extra time spent by MGKT is mainly related to multiple graph generation and integration, which leads to an increase in physical memory and computational requirements while improving the accuracy in KT task.

TABLE V
ACC AND AUC RESULTS OF SCDM AND DCDMC

Model	ASSIST2009		ASSIST2017	
	ACC (↑, %)	AUC (↑, %)	ACC (↑, %)	AUC (↑, %)
SA-DGC	75.18	80.05	73.90	80.11
QA-DGC	75.07	79.96	73.90	80.10
MGKT	75.21	80.15	73.97	80.13

F. Analysis of DcDMC Module

In Table V, we compare the single-channel directed multigraph (ScDM) module and DcDMC module based on ASSIST2009 and ASSIST2017. We have designed three structures as follows.

1) *SA-DGC* contains an ScDM of skill-response relationship.
2) *QA-DGC* contains an ScDM of exercise-response relationship.
3) *MGKT* contains a DcDMC of both skill-response and exercise-response relationships.

Applying dual-channel module, MGKT increases by 0.02%∼0.19% in AUC and 0.03%∼0.14% in ACC on two datasets compared with the single-channel based methods SA-DGC and QA-DGC. This is because the single-channel graph is not

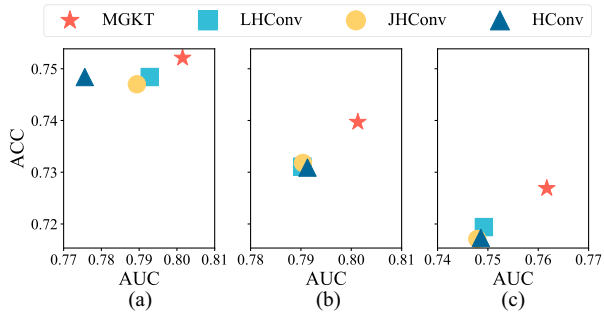

Fig. 6. AUC and ACC of hypergraph convolution modules on: (a) ASSIST2009; (b) ASSIST2017; and (c) EdNet.

enough to characterize the simultaneous changes in both exercise-response and skill-response sequences, and the relations between them, thus may lead to bias in the prediction of student's knowledge states.

G. Analysis of LHAConv Module

To demonstrate the effectiveness of the hypergraph convolution module, we compare the LHAConv with several existing hypergraph convolutions via AUC and ACC with train-test split ratio 8:2. Below describe these hypergraph convolution methods:

1) *HConv* [32] is the original hypergraph convolution method with the convolution layer defined as

$$\boldsymbol{X}_j^{(l+1)} = \sigma\left(\mathbf{D}_{M,j}^{-1/2}\mathbf{H}_{M,j}\mathbf{B}_{M,j}^{-1}\mathbf{H}_{M,j}^{\mathrm{T}}\mathbf{D}_{M,j}^{-1/2}\boldsymbol{X}_j^{(l)}\boldsymbol{\theta}^{(l)}\right). \quad (23)$$

2) *JHConv* [33] denotes jump hypergraph convolution. It considers both its original features and aggregated related representations. The inclusion of jump connections helps the model avoid the information dilution caused by additional connection integration. The convolution layer is defined as follows:

$$\boldsymbol{X}_j^{(l+1)} = \sigma\left(\boldsymbol{D}_{M,j}^{-1/2}\boldsymbol{H}_{M,j}\mathbf{B}_{M,j}^{-1}\boldsymbol{H}_{M,j}^{\mathrm{T}}\boldsymbol{D}_{M,j}^{-1/2}\boldsymbol{X}_j^{(l)}\boldsymbol{\theta}^{(l)} + \boldsymbol{X}_j^{(l)}\right). \quad (24)$$

3) *LHConv* denotes a hypergraph convolution structure of MGKT that contains only LSTM and no attention mechanism.
4) *MGKT* contains LHAConv, a hypergraph convolution structure with both LSTM and attention mechanism.

As shown in (23) and (24), we replace the association matrix $\mathbf{H}$ of hypergraph in the original HConv and JHConv with the multiple graph joint association matrix $\mathbf{H}_M$. Fig. 6 shows the AUC and ACC of the four hypergraph convolutions of LHAConv, LHConv, JHConv, and HConv on the three datasets. First, although LHConv shows slightly worse performance than HConv and JHConv on ASSIST2017, it surpasses them about 0.06%∼4.42% in AUC and 0.09%∼0.23% in ACC on the other two datasets. Second, MGKT improves LHConv by 0.87%∼2.59% in AUC and 0.37%∼0.98% in ACC with the help of attention mechanism. In all, the results demonstrate the superiority of LSTM in learning the higher-order connections and attention mechanism in capturing the potential differences within hypergraph.

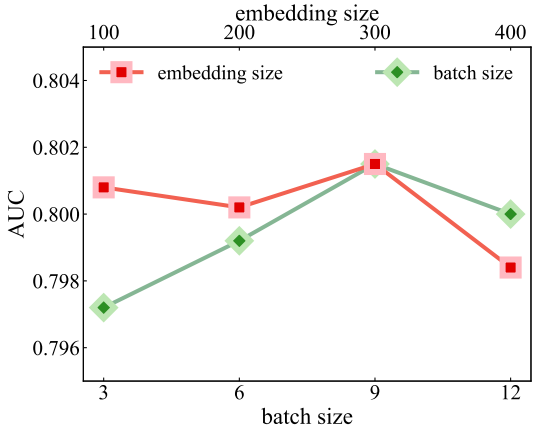

Fig. 7. Hyper-parameter sensitivity analysis of embedding size and batch size.

H. Hyper-Parameter Sensitivity Analysis

In this section, we investigate the sensitivity of hyper-parameters such as embedding size and model batch size in MGKT based on AUC. As shown in Fig. 7, the embedding size and batch size are set as [100,200,300,400] and [3], [6], [9], [12], respectively.

1) For the knowledge state embedding size, the model performance increases steadily when the embedding layer gradually increases, but there is a gradual decrease after reaching a certain level. This is due to the fact that when the embedding layer is too large, the representations of knowledge states become mixed and indistinguishable, which affects the final prediction results of the model.
2) For the model sample size, the model performance maintains a steady upward trend when the sample size ranges from 3 to 9, but the performance decreases rapidly as the sample size increases further. This is due to the reduced generalization ability of the model as a result of the large number of samples.

Overall, MGKT has good model stability and the optimal values of its hyper-parameters are easily determined.

I. Cold Start

Cold start is one of the problems to be addressed in real-world educational datasets. Following [17], we compare the performance of DKT, SGKT, and MGKT in the following two cold start situations on ASSIST2017.

1) *Scenario 1* is to train model with a few student interactions and apply it to new unseen samples. As shown in Fig. 8(a), we conduct experiments using different proportions of students in the training dataset. As the percentage of student ranges from 1% to 15%, both MGKT and SGKT increase in ACC and AUC, and decrease in RMSE. In addition, MGKT improves AUC by 7.05%∼11.92% over DKT and 2.31%∼10.74% over SGKT, ACC by 3.45%∼9.21% over DKT and

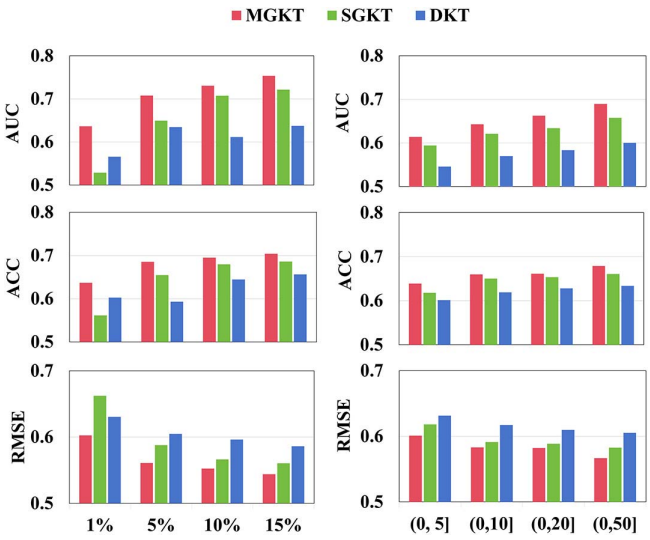

Fig. 8. Results of cold start problem in: (a) Scenario 1; and (b) Scenario 2.

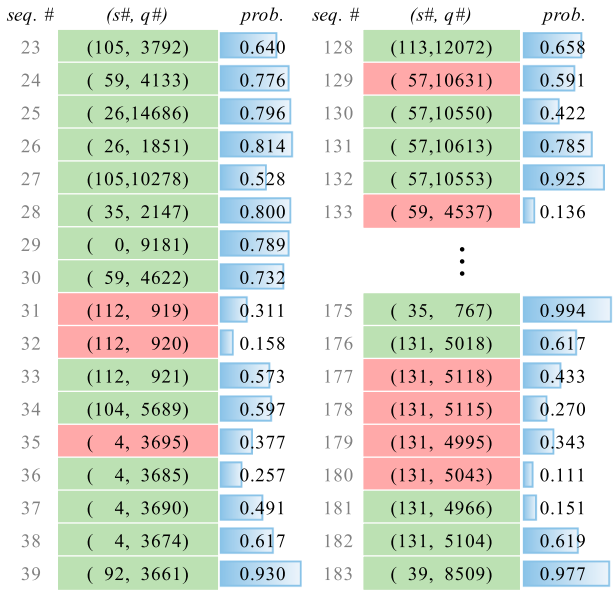

Fig. 9. Case study of a certain student in ASSIST2009.

1.56%∼7.56% over SGKT, and RMSE by –4.4% ∼ –2.79% over DKT and –5.97% ∼ –1.39% over SGKT.

2) *Scenario 2* occurs when new students with short exercise sequences result in insufficient features into the KT model. We classify the training data into four groups with different ranges of exercise sequence length, i.e. (0, 5], (0, 10], (0, 20] and (0, 50]. Each training exercise sequence in this scenario is extracted from the corresponding original sequence. Obviously, the first group meets the most difficult situation because such short exercise sequence in this group easily leads to insufficient information for graph structure construction. As can be seen in Fig. 8(b), the three models generally grow in ACC and AUC and decline in RMSE as the range of training sequence length enlarges. Besides, MGKT surpasses SGKT by about 2%, 1.5%, –1% and DKT by about 7%, 4%, –3% in terms of average AUC, ACC, and RMSE, respectively.

J. Visualization Case Study

Fig. 9 presents a knowledge tracing case of a certain student via the probabilities of correct answers by MGKT. The answering sequence has total 275 interactions. For a better illustration of the knowledge tracing process, we select some parts of the answering sequence in Fig. 9. The main findings are listed below.

1) From seq. 35 to seq. 38, MGKT produces a lower prediction for q_{3685} following a wrong answer for q_{3695} and produces higher predictions for q_{3690} and q_{3674} after two successive correct answers since MGKT detects that these exercises relate to the same latent skill s_4. The similar patterns are obtained from seq. 129 to seq. 132 and from seq. 176 to seq. 182. The slight fluctuation in the predictions for q_{5118} and q_{4995} may result from the relations between the covered skill and the other skills.
2) MGKT produces the similar predictions for q_{4133} and q_{4622} because the two exercises cover the same skill s_{59} and the corresponding interactions are close in time. By comparison, MGKT predicts a lower probability for q_{4537} because s_{59} required by q_{4537} indicates a higher forgetting chance due to the long-time interval between *seq.* 30 and *seq.* 133. However, q_{767} shows higher exercise mastery than q_{2147} with the same skill s_{35} despite of long time interval. This may indicate the different forgetting rate of different skills.

These findings not only support the effectiveness of MGKT in capturing forgetting feature and higher-order semantic information in student's learning process, but also bring the following inspirations to educators.

1) *Dig Into Knowledge Correlation:* The educators should pay attention to the relevance among skills instead of emphasis on independence of each skill or exercise, and help students construct knowledge framework. Based on the mistake collection by each student, educators can determine the range of skill weakness according to the knowledge correlation and then conduct targeted tutoring.
2) *Grasp Forgetting Pattern and Timing of Review:* Different skills show differences in forgetting rate. The educators can schedule the review session according to the forgetting rate and other features of each skill. For example, skills with high forgetting rate require an increase in review frequency, while skills with low forgetting rate enable appropriate reduction in repetition and thus the time saved can be put into the study of other important and difficult skills.
3) *Optimize Teaching Resource and Exercise Design:* When designing course assignment, the educators can distribute the exercises with same skill into different time periods to avoid student's short-term memory resulted from concentrated practice and then failure in knowledge mastery. At the same time, the assignment or

teaching resource should also cover as many types of exercises with same skill as possible. For instance, student's knowledge proficiency may differ between application and calculation problems due to the understanding of the text contents.

V. Conclusion

In this article, we proposed a MGKT model. First, we incorporated the forgetting behavior into the heterogeneous information graph convolution module of MGKT. Second, we studied the temporal sequentiality and correlations of both exercise-response and skill-response sequences and then constructed a dual-channel directed graph communication module for MGKT to characterize student's knowledge state during exercise-response process. Finally, we developed a hypergraph convolution module with LSTM and attention mechanism to capture higher-order correlations of knowledge sequences. To verify the effectiveness of MGKT, we conducted extensive experiments on three public datasets, including comparison experiment, ablation study, module analysis, hyperparameter sensitivity analysis, and case study. The results validated the advantages of MGKT in effectiveness and interpretability over some state-of-the-art KT models.

References

[1] J. R. Anderson, C. F. Boyle, A. T. Corbett, and M. W. Lewis, "Cognitive modeling and intelligent tutoring," *Artif. Intell.*, vol. 42, no. 1, pp. 7–49, 1990.

[2] S. Hen et al., "Monitoring student progress for learning process-consistent knowledge tracing," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 8, pp. 8213–8227, Aug. 2022.

[3] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Model. User-Adapted Interaction*, vol. 4, pp. 253–278, Dec. 1994.

[4] ŠG. Ines, G. Ani, and G. Angelina, "Twenty-Five Years of Bayesian knowledge tracing: a systematic review," *User Model. User-Adapted Interaction*, vol. 34, no. 4, pp.1127–1173, 2024.

[5] F. Liu, X. Hu, C. Bu, and K. Yu, "Fuzzy Bayesian knowledge tracing," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 7, pp. 2412–2425, Jun. 2021.

[6] G. Abdelrahman, Q. Wang, and B. Nunes, "Knowledge tracing: A survey," *ACM Comput. Surv.*, vol. 55, no. 11, pp. 1–37, 2023.

[7] L. Wang, X. Li, Z. Luo, Z. Hu, and Q. Yan, "Multivariate cognitive response framework for student performance prediction on MOOC," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 3, pp. 1221–1233, Mar. 2024.

[8] C. Piech, et al., "Deep knowledge tracing," in *Proc. 28th Adv. Neural Inf. Process. Syst.,* vol. 1, 2015, pp. 505–513.

[9] J. Zhang, X. Shi, I. King, and D. Y. Yeung, "Dynamic key-value memory networks for knowledge tracing," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 765–774.

[10] S. Pandey and G. Karypis, "A self-attentive model for knowledge tracing," in *Proc. 12th Int. Conf. Educ. Data Mining*, 2019, pp. 384–389.

[11] A. Ghosh, N. Heffernan, and A. S. Lan, "Context-aware attentive knowledge tracing," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 2330–2339.

[12] H. Zhan, J. J. Kim, and G. Liu, "Contrastive learning with bidirectional transformers for knowledge tracing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP),* 2024, pp. 5040–5044.

[13] S. Zu, S. Cai, W. Tang, C. Wang, L. Li, and J. Shen, "GuessKT: improving knowledge tracing via considering guess behaviors," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP),* 2024, pp. 12811–12815.

[14] H. Nakagawa, Y. Iwasawa, and Y. Matsuo, "Graph-based knowledge tracing: modeling student proficiency using graph neural network," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell.*, 2019, pp. 156–163.

[15] Y. Yang et al., "GIKT: a graph-based interaction model for knowledge tracing," in *Proc. Mach. Learn. Knowl. Discovery Databases: Eur. Conf.*, 2020, pp. 299–315.

[16] X. Song, J. Li, Y. Tang, T. Zhao, Y. Chen, and Z. Guan, "JKT: A joint graph convolutional network based deep knowledge tracing," *Inf. Sci.*, vol. 580, pp. 510–523, Nov. 2021.

[17] Z. Wu, L. Huang, Q. Huang, C. Huang, and Y. Tang, "SGKT: Session graph-based knowledge tracing for student performance prediction," *Expert Syst. Appl.*, vol. 206, 2022, Art. no. 117681.

[18] L. Wu et al., "Modeling the evolution of users' preferences and social links in social networking services," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 6, pp. 1240–1253, Jun. 2017.

[19] T. Wu and Q. Ling, "Self-supervised heterogeneous hypergraph network for knowledge tracing," *Inf. Sci.*, vol. 624, pp. 200–216, May 2023.

[20] P. I. Pavlik, H. Cen, and K. R. Koedinger, "Performance factors analysis–a new alternative to knowledge tracing," in *Proc. Conf. Artif. Intell. Educ.: Building Learn. Syst. Care: From Knowl. Representation Affect. Model.*, 2009, pp. 531–538.

[21] J. J. Vie and H. Kashima, "Knowledge tracing machines: Factorization machines for knowledge tracing," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, 2019, pp. 750–757.

[22] C. K. Yeung and D. Y. Yeung, "Addressing two problems in deep knowledge tracing via prediction-consistent regularization," in *Proc. 5th Annu. ACM Conf. Learn. Scale*, 2018, pp. 1–10.

[23] S. Shen, et al., "Convolutional knowledge tracing: modeling individualization in student learning process," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 1857–1860.

[24] A. T. Nickl and K. H. T. Bäuml, "To-be-forgotten information shows more relative forgetting over time than to-be-remembered information," *Psychonomic Bull. Rev.*, vol. 31, no. 1, pp. 156–165, 2024.

[25] K. Nagatani, Q. Zhang, M. Sato, Y. Y. Chen, F. Chen, and T. Ohkuma, "Augmenting knowledge tracing by considering forgetting behavior," in *Proc. World Wide Web Conf.*, 2019, pp. 3101–3107.

[26] C. Wang et al., "Temporal cross-effects in knowledge tracing," in *Proc. 14th ACM Int. Conf. Web Search Data Mining*, 2021, pp. 517–525.

[27] G. Abdelrahman and Q. Wang, "Deep graph memory networks for forgetting-robust knowledge tracing," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 8, pp. 7844–7855, Aug. 2022.

[28] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[29] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2016.

[30] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018.

[31] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: clustering, classification, and embedding," in *Proc. 19th Int. Conf. Neural Inf. Process. Syst.*, 2006, pp. 1601–1608.

[32] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, 2019, pp. 3558–3565.

[33] S. Ji, Y. Feng, R. Ji, X. Zhao, W. Tang, and Y. Gao, "Dual channel hypergraph collaborative filtering," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 2020–2029.

[34] Y. Gao, Z. Zhang, H. Lin, X. Zhao, S. Du, and C. Zou, "Hypergraph learning: Methods and practices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2548–2566, May 2022.

[35] C. Cui et al., "DGEKT: a dual graph ensemble learning method for knowledge tracing," *ACM Trans. Inf. Syst.*, vol. 42, no. 3, pp. 1–24, 2024.

Ruichun Kang received the B.S. degree in communication engineering in 2022, from Hunan Normal University, Changsha, China, where she is currently working toward the M.S. degree with the Communication Engineering, College of Electrical and Information Engineering.

Her research interests include educational data mining and machine learning.

Xiaoyao Li received the B.S. degree in mathematics and applied mathematics from China University of Petroleum, Beijing, China, in 2012, and the Ph.D. degree in control science and engineering from Hunan University, Changsha, China, in 2022.

Currently, she is a Lecturer with the College of Advanced Interdisciplinary Studies, Central South University of Forestry and Technology, Changsha, China. She was a Research Assistant with the Department of Computer and Information Science, University of Macau, Macau, China, in 2017. Her research interests include educational data mining, signal/image processing, and computer vision.

Guiyao Liu received the B.S. degree in electronic information engineering from Shanghai Polytechnic University, Shanghai, China, in 2023. He is currently working toward the M.S. degree with the Electronic Information, College of Electrical and Information Engineering, Hunan University, Changsha, China.

His research interests include educational data mining and large language models.

Lianhong Wang received the B.S., M.S., and Ph.D. degrees in radio technology, circuits and systems, control theory and control engineering from Hunan University, Changsha, China, in 1993, 2002, and 2009, respectively.

Currently, she is a Full Professor with the College of Electrical and Information Engineering, Hunan University. From 2011 to 2012, she was a Visiting Scholar with Brandeis University, Waltham, Massachusetts, USA. She has published more than 30 papers in journals and conferences. Her research interests include educational data mining, computer vision, and artificial intelligence.