

# HiTSKT: A hierarchical transformer model for session-aware knowledge tracing

Fucaí Ke<sup>a,\*</sup>, Weiqing Wang<sup>a</sup>, Weicong Tan<sup>a</sup>, Lan Du<sup>a</sup>, Yuan Jin<sup>a</sup>, Yujin Huang<sup>a</sup>, Hongzhi Yin<sup>b</sup>

<sup>a</sup> Faculty of Information Technology, Monash University, Melbourne, 3800, VIC, Australia

<sup>b</sup> Faculty of Engineering, The University of Queensland, Brisbane, 4072, QLD, Australia

## ARTICLE INFO

### Keywords:

User behaviour modelling  
Knowledge tracing  
Educational data mining  
Learner modelling  
Hierarchical transformer

## ABSTRACT

Knowledge tracing (KT) aims to leverage students' learning histories to estimate their mastery levels on a set of pre-defined skills, based on which the corresponding future performance can be accurately predicted. In practice, a student's learning history comprises answers to sets of massed questions, each known as a session, rather than merely being a sequence of independent answers. Theoretically, within and across these sessions, students' learning dynamics can be very different. Therefore, how to effectively model the dynamics of students' knowledge states within and across the sessions is crucial for handling the KT problem. Most existing KT models treat student's learning records as a single continuing sequence, without capturing the sessional shift of students' knowledge state. To address the above issue, we propose a novel hierarchical transformer model, named HiTSKT, comprises an interaction(-level) encoder to capture the knowledge a student acquires within a session, and a session(-level) encoder to summarize acquired knowledge across the past sessions. To predict an interaction in the current session, a knowledge retriever integrates the summarized past-session knowledge with the previous interactions' information into proper knowledge representations. These representations are then used to compute the student's current knowledge state. Additionally, to model the student's long-term forgetting behaviour across the sessions, a power-law-decay attention mechanism is designed and deployed in the session encoder, allowing it to emphasize more on the recent sessions. Extensive experiments on four public datasets demonstrate that HiTSKT achieves new state-of-the-art performance on all the datasets compared with seven state-of-the-art KT models.

## 1. Introduction

Online education breaks the temporal and spatial limitations of traditional in-person learning and brings huge educational benefits to society, especially during the Covid-19 period [1–3]. It is essential to provide personalized experience for students in online education as different students have different mastering levels of specific skills or concepts during their learning progresses.

Knowledge tracing (KT) which aims to trace students' learning histories to estimate their mastery levels on a set of pre-defined skills becomes the key in providing personalized experience [4,5]. Based on the traced students' knowledge, E-educators can gain a comprehensive understanding of the dynamics of students' knowledge states. The understanding enables them to tailor personalized learning curricula accordingly. Intelligent tutoring systems (ITS) strive to automate the task of recommending personalized curricula. With the advancements in KT techniques within machine learning, ITS have been endowed with the same tracing ability as human educators. Specifically, these systems

utilize KT techniques to model and infer students' mastery levels of skills over time, drawing insights from students' historical responses to exercise questions.

Despite rapid development, KT remains a challenging and unresolved issue in the field. In practice, the dynamics that underlie students' mastery of skills are characterized by complexity and evolution, yet the current research has not extensively explored these aspects. Over the past two decades, numerous KT models leveraging statistical and machine learning techniques have been proposed. A predominant approach involves utilizing hidden Markov models (HMMs) to capture the dynamics of students' knowledge states. These models incorporate various assumptions concerning the distributions of states and other relevant variables, such as students' response accuracy and question difficulty (e.g., as in [6–8]). Despite the efficacy and semantics introduced to model learning, these assumptions have, however, constrained the models' applicability in reality. These models are oversimplified, lacking the necessary complexity to capture the dynamics accurately.

\* Corresponding author.

E-mail address: [fucai.ke1@monash.edu](mailto:fucai.ke1@monash.edu) (F. Ke).

<https://doi.org/10.1016/j.knosys.2023.111300>

Received 6 June 2023; Received in revised form 29 November 2023; Accepted 12 December 2023

Available online 15 December 2023

0950-7051/© 2023 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

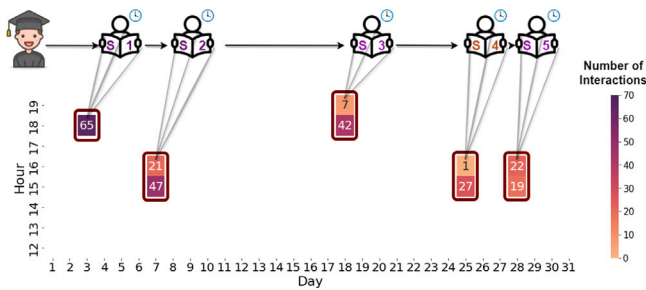


Fig. 1. Interactions heatmap for one student in one month on ASSISTments2017. Each red frame is a practice session containing several interactions, and time intervals are represented as arrow. One interaction means the process that a student interacts with ITS and provides one response to one exercise question.

For instance, Gaussian prior assumptions on student expertise and question difficulty represent simplified treatments that overlook the true statistics of these factors.

Recent KT models take advantage of the expressive power of deep neural networks (DNNs), such as the RNN-based [9–13] and the self-attention [14–17] models. These DNNs help overcome some modelling and inference limitations of traditional models like HMMs. For instance, RNN-based KT models focus on predicting each interaction based on its predecessors, with recent ones having a larger impact. In contrast, self-attention-based KT models condition predictions on all historical interactions simultaneously. With the utilization of deep neural networks, these advanced KT models enhance performance in the KT task [9,10,18–20].

Although the improvements brought by the DNNs, current KT models still have several known issues. One significant problem is that these models treat a student's learning records as a single continuous sequence, neglecting to capture the true distribution of students' learning records. As an example, we show in Fig. 1 our observations via exploration on ASSISTments2017,<sup>1</sup> a real-world KT dataset. This figure illustrates a student's learning history over 31 days, encompassing five distinct practice sessions. Each session involves answering a varying number of questions consecutively within relatively short periods, with much longer intervals between any two consecutive practice sessions. The sessional behaviour observed in a single student repeats across the entire student population. The statistical details supporting this observation will be further discussed in Section 5 to confirm the generality of our findings. Therefore, rather than being continuous, the interaction records of students in reality typically display clustering effects. These records consist of a series of separated practice sessions, with each session containing a sequence of practice interactions.

In Educational Psychology, studies on memory retention consider sessional information (e.g., in the form of massed exercises over certain spacing of time) as an indicative factor of students' learning performance [21–24]. Within-session information (e.g., questions number and difficulty) and inter-session information (e.g., spacing between different sessions) have distinct impacts on students' learning performance [25–27]. Therefore, it is reasonable to separately model the within- and inter-session dynamics underlying students' learning process to capture their separate impacts.

Existing KT models are not designed to capture the sessional information and its effects on students' performance. To do so, we propose a novel Hierarchical Transformer model for Session-Aware Knowledge Tracing (HiTSKT), that simultaneously captures the dynamics and connections underlying two types of sequences: (1) sequences of responses within a session and (2) sequences of different sessions.

HiTSKT models these two types of sequences in a bottom-up manner with a *Acquisition & Consolidation* (AC) modelling component. The AC

component is designed as a hierarchical transformer encoder architecture, in which (1) an *interaction* encoder transforms the knowledge acquired by a student within a session into an intra-session knowledge representation vector, and (2) a *session* encoder consolidates the intra-session representations of all the past sessions into a vector representing the inter-session knowledge up to the current session. The memory consolidation progress is performed by the encoder network with a forgetting mechanism that emphasizes more on the intra-session information from more recent sessions. After consolidating inter-session knowledge, HiTSKT employs a *Retrieval & Responding* (RR) modelling component to calculate current knowledge state. This component first retrieves the stored inter-session knowledge representation and then integrates it with the intra-session knowledge acquired thus far in the session.

The contributions of our work are summarized as follows:

- To our best knowledge, we are the first to exploit the underlying sessional information from the students' learning histories for the KT problem. A detailed exploratory data analysis of the sessional information on four real-world online educational datasets has been conducted in Section 5.1, which provides a new insight into the KT problem.
- A carefully-designed hierarchical transformer-based model, named HiTSKT, is proposed to model the sessional information for the KT problem. With a session-aware hierarchical transformer encoder, the model is capable of capturing students' knowledge state variations within and across the sessions. Additionally, the knowledge state retrieval encoder enables measuring the impact of stored knowledge on students' performance in the current session.
- HiTSKT also models and captures students' sessional forgetting behaviour through an power-law-decay scaled attention mechanism in the acquisition & consolidation modelling component. The sessional forgetting behaviour is evidenced by our exploratory data analysis in Section 5.6.
- Extensive experiments including overall effectiveness studies, ablation studies and model interpretation, have been conducted on four real-world online educational datasets. The results demonstrate (1) the superior response correctness prediction performance of HiTSKT by capturing the sessional information, compared with seven state-of-the-art KT models; and (2) the effectiveness of the major components of HiTSKT. In addition, the code for HiTSKT and the implementation of several state-of-the-art KT models are provided.<sup>2</sup>

The rest of the paper is organized as follows. Section 2 surveys the existing KT research related to our work. Section 3 formulates the KT problem to be tackled in this paper. The details of our proposed model, HiTSKT, are described in Section 4. Section 5 concerns the experimental evaluation of HiTSKT against seven state-of-the-art KT models over four real-world datasets and the discussion of the evaluation results. Section 6 draws a conclusion on the work completed in this paper, and envisage the future work to be extended from it.

## 2. Related work

**Traditional Knowledge Tracing Methods.** Corbett and Anderson [6] proposed the first KT model (i.e., Bayesian Knowledge Tracing, BKT) by employing the hidden Markov model. BKT predicts students' knowledge state (i.e., their proficiency) on a skill (i.e., a knowledge concept in the curriculum) based on their performance on the immediate last quiz. They also conducted a series of experiments that confirmed the efficacy of their model. Subsequently, many extended models based on BKT have been proposed (e.g., [7,8]) that aim to

<sup>1</sup> ASSISTments2017 source: <https://sites.google.com/site/assistmentsdata>

<sup>2</sup> <https://github.com/pokerme7777/HiTSKT>

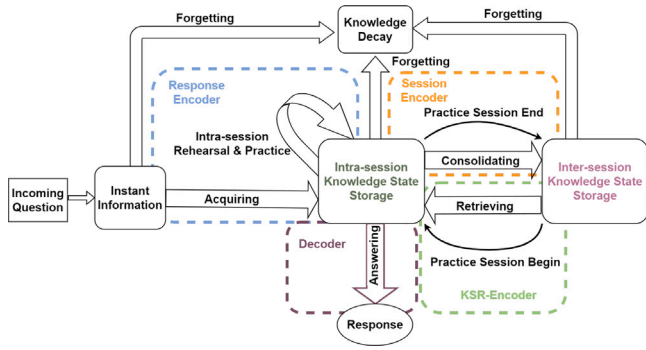


Fig. 2. HiTSKT motivation based on Atkinson Shiffrin memory model.

improve model performance and interpretability. However, BKT and its extended models are not capable of capturing more subtle and complex patterns underlying students' learning histories due to their intrinsic lack of flexibility [5,28,29].

**Deep Learning and Other Knowledge Tracing Methods.** To address the limitations of the traditional KT models, Piech et al. [9] pioneered the development of the deep neural network KT model, known as DKT, employing LSTM. DKT outperformed BKT and its extended models on various benchmark datasets [9,30]. Following this pioneering work, more KT models explore the use of deep neural networks. Zhang et al. [10] proposed a KT model with the dynamic key-value memory network (DKVMN) that leverages a memory-augmented neural network to extract the relationship among skills. Inspired by DKVMN, Abdelrahman and Wang [31] stacked a modified LSTM (i.e., the Hop LSTM) on top of a dynamic key-value network in their KT model. Furthermore, to improve model robustness and avoid model overfitting on smaller datasets, Guo et al. [32] proposed an adversarial training based KT method (ATKT). More recently, Wang et al. [33] argue that students' interactions can be considered as point processes, and what happened last will have a distinct influence on learning. Hence, they proposed to use Hawkes process to model the temporal information of students' learning history and decomposes a sequence of interactions into local dynamic processes.

A notable category of KT models is self-attentive KT models. Following the transformer's [34] success in natural language processing and computer vision tasks, KT models with self-attention mechanisms have been proposed. Pandey and Karypis [14] proposed the first KT model based on self-attention mechanisms (i.e., SAKT). The SAKT model can capture the long term dependency between the interactions of one student. Pu et al. [16] proposed a KT model based on vanilla transformer [34]. Additionally, Ghosh et al. [15] introduced a context-aware KT model named AKT, featuring a monotonically and exponentially decaying attention mechanism. AKT is designed to roughly capture the influence of a student's performance from a distant past on their current performance.

Although self-attentive KT models [18,20,32,35] have demonstrated enhanced tracing performance, their designs overlook sessional information. These models are too rigid to incorporate such information to capture the potential sessional drifts of knowledge states. Furthermore, several models [12,36–38] are designed to capture memory-decay dynamics. For example, HMN [12] uses Differentiable Neural Computer (DNC) [36] to improve the modelling of memory decay. Similarly, LBKT [37] and LFBKT [38] proposed forget gates to measure memory decay effect by processing more learning behaviours (e.g., hints, attempts, thinking time span). However, these works [12,36–38] are restricted to measuring the impact of individual interactions, overlooking the dependence of memory decay on the spacing of sessions.

**Memory Retention and Educational Psychology Studies** (e.g., [23,24,26]) have shown that a student's overall performance within

Table 1

Notations used in this paper.

| Notation    | Description                                            |
|-------------|--------------------------------------------------------|
| $I$         | The number of students                                 |
| $K$         | The number of skills                                   |
| $Q$         | The number of questions                                |
| $i$         | The $i$ th student or student $i$                      |
| $t$         | The $t$ th time step or time $t$                       |
| $ses_n^i$   | The $n$ th session of the student $i$                  |
| $x$         | An interaction $x$                                     |
| $q$         | A question $q$                                         |
| $k$         | A skill/concept $k$                                    |
| $f_q$       | The number of occurrences of the question $q$          |
| $a$         | The binary correctness of a response                   |
| $x$         | Embedding of the interaction $x$                       |
| $d_q$       | Embedding of the index of question $q$                 |
| $k$         | Embedding of the skill $k$                             |
| $f_q$       | Embedding of the number of occurrences of question $q$ |
| $a$         | Embedding of the correctness of a response             |
| $h^{inner}$ | Intra-session knowledge representation vector          |
| $h^{inter}$ | Inter-session knowledge representation vector          |
| $q$         | Attention query vector                                 |
| $K$         | Key matrix                                             |
| $V$         | Value matrix                                           |
| $W$         | Mapping matrix                                         |

practice sessions can better reflect their knowledge mastery levels. The student's knowledge states between sessions are hugely different due to the memory decay and consolidation effect of long-term memory [21, 22,25]. On the contrary, the intra-session knowledge state is relatively steady with slight fluctuation. Because the impact of a single exercise is not substantial enough to significantly affect memory. Therefore, it is reasonable to learn the intra-session and inter-session knowledge states separately. Current KT models ignore the potential sessional drifts in students' memories during their learning processes. We believe that modelling such drifts can considerably account for variations of a student's knowledge states and is the key to developing more effective KT models.

Furthermore, educational psychologists (e.g., [21,22,25]) have pointed out that the learning practice is always session based. As illustrated in Fig. 2, when students launch a session, their brain retrieves related knowledge states from their memory storage as the working states. The working states can also be viewed as intra-session knowledge states, and are ready to respond. Human intelligence acquires necessary information from the incoming question and utilizes the working states to react. Once the session practice is finished, the working states will be consolidated into the deep memory storage, which corresponds to the inter-session knowledge state, and decays over time [23,24,26].

Overall, we can argue that the sessional information underlying students' learning histories should be explicitly exploited for KT. Inspired by [23,24,26,39], in this paper, we propose a KT model with hierarchical transformer encoders to fully exploit the sessional information.

### 3. Problem formulation

Throughout the paper, the notation used to formulate our targeting KT problem are described in Table 1. Based on this notation, suppose that the learning records of the student  $i$  are composed of  $n$  historical, non-overlapping sessions and the current  $(n + 1)$ th session  $\{ses_1^i, ses_2^i, \dots, ses_n^i, ses_{n+1}^i\}$ , where  $ses_n^i$  denotes the student  $i$ 's  $n$ th session. For  $ses_n^i$ , it contains a sequence of  $t$  interactions  $x_{n,t}^i = \{x_{n,t}^i, x_{n,t}^i, \dots, x_{n,t}^i\}$ , where an interaction  $x_{n,t}^i = \{q_{n,t}^i, k_{n,t}^i, f_{n,t}^i, a_{n,t}^i\}$  consists of (1) the question  $q_{n,t}^i$  that the student  $i$  answered at time step  $t$  in session  $n$ , (2) the corresponding skill  $k_{n,t}^i$  required for the question, (3) the number of occurrences of the question (until time  $t$ )  $f_{n,t}^i$ , and (4) the graded response  $a_{n,t}^i$ . Note that the graded response  $a_{n,t}^i \in \{1, 0\}$

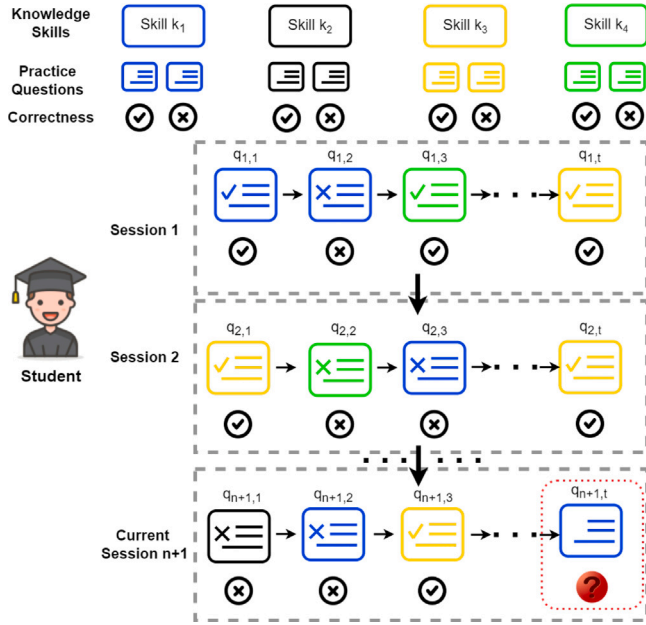


Fig. 3. An illustrative example of the proposed HiTSKT task.

denotes whether the student  $i$  has answered the  $t$ th question in the  $n$ th session correctly or not. In this case, the student  $i$ 's past learning history includes  $n$  sessions, i.e.,  $\{ses_1^i, ses_2^i, \dots, ses_n^i\}$ , the first  $t-1$  interactions in the current  $(n+1)$ th session, i.e.,  $\{x_{n+1,1}^i, x_{n+1,2}^i, \dots, x_{n+1,t-1}^i\}$ . Based on learning history and a query question  $q_{n+1,t}^i$  at the current time  $t$ , the problem of KT concerns the prediction of the graded answer  $a_{n+1,t}^i$  of student  $i$  to the query question. Fig. 3 illustrates our targeting problem formulated above.

In this paper, the time duration we use to define a session is 10 hours based on human activity studies.<sup>3</sup> Specifically, the time interval between two sessions should be longer than the session duration. In other words, if the next interaction occurs later than 10 hours, then this interaction belongs to a new session.

#### 4. Methodology

HiTSKT is motivated by the Atkinson Shiffrin memory model [21, 22, 24, 25] as shown in Fig. 2. HiTSKT consists of two main components: the acquisition & consolidation (AC) modelling component and the retrieval & responding (RR) modelling component. In the first component, a carefully designed session-aware hierarchical transformer is employed for the acquisition and consolidation processes. In the second component, a knowledge state retrieval (KSR) encoder module and a student response prediction module are utilized for retrieval and response. The structure of HiTSKT is shown in Fig. 4.

##### 4.1. Acquisition & Consolidation modelling component

The AC component aims to extract and aggregate hierarchical sessional information in a bottom-up manner. Each interaction-level encoder summarizes the acquired intra-session knowledge for individual sessions. A special token AKSS (i.e., intra-session Knowledge State Storage) is designed and placed alongside the input sequence to the encoder to store the intra-session knowledge representation. These AKSS representations, output from the interaction encoders, are then fed into an upper-level session encoder. In session encoder, all the past sessions experienced by a student get consolidated into an inter-session Knowledge State Storage (RKSS).

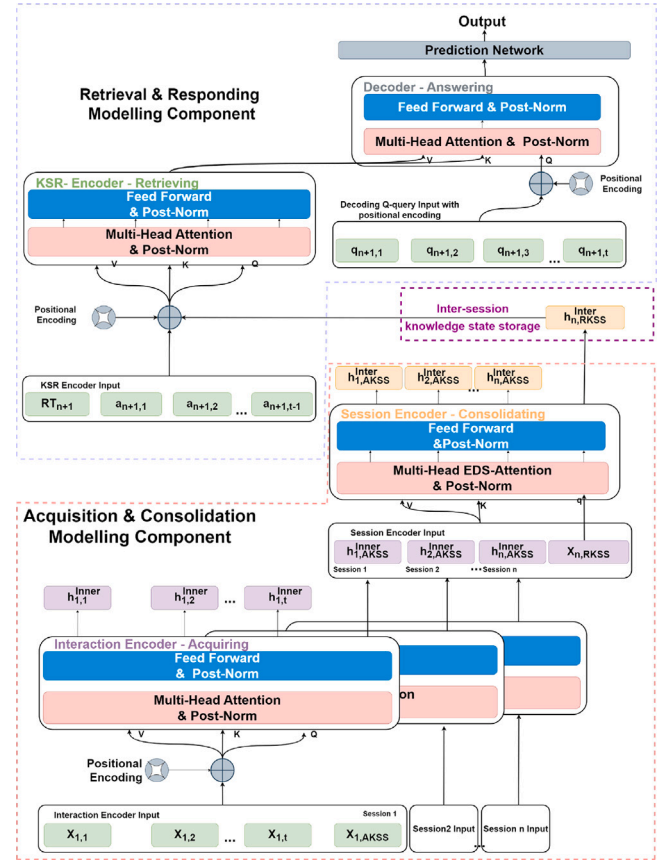


Fig. 4. The overall model architecture of HiTSKT, which consists of two components: AC component and RR component. The AC component aims to extract and aggregate the sessional information in a bottom-up manner. The RR component leverages a KSR encoder to retrieve and refine a student's stored inter-session representation, which is then fed into an answering decoder to predict the student's next interaction in the current session.

##### 4.1.1. Interaction encoder

The purpose of this encoder is to precisely capture variations in students' knowledge states within each past session and generate corresponding intra-session knowledge representations. The encoder consists of a rehearsal embedding layer, a multi-head attention layer, and a feedforward neural network layer.

**Rehearsal embedding** serves as the input to the interaction encoder, aiming to capture as much diverse information about students' individual interactions as possible. In this study, we examine information from various perspectives. This includes considering the skill or concept covered by a question, which may indicate knowledge state correlations among questions of the same concept. Additionally, we explore the index of a question, which provides insights into variations in students' knowledge states (e.g., being low for difficult questions). The number of occurrences of a question is also considered, reflecting correlations between rehearsals/practice and knowledge states. Lastly, we take into account the correctness of past responses, which serves as an indicator of a student's learning quality. In particular, the last two factors together can also be indicative of the student's knowledge state variation. According to [26, 41], students with high knowledge states in specific domains are likely to answer questions related to those domains correctly after only a few of practice attempts. Conversely, if a student continues to answer such questions incorrectly after several attempts, it suggests a likely low knowledge state in that particular area.

Therefore, we design the rehearsal embedding for the  $t$ th interaction of the  $n$ th session  $x_{n,t}$  to be the sum of the embeddings of (1) the skill  $k_{n,t} \in \mathbb{R}^d$  of the rehearsal question  $q_t$ , (2) the question's index

<sup>3</sup> People working hours per day [40].

$d_{n,q_i} \in \mathbb{R}^d$ , (3) the ordinal label for the occurrence number of the question  $f_{n,q_i} \in \mathbb{R}^d$  and (4) the student's response correctness  $a_{n,i} \in \mathbb{R}^d$ ; That is the following formulation of the rehearsal embedding  $x_{n,i}$ :

$$x_{n,i} = k_{n,i} \oplus d_{n,q_i} \oplus f_{n,q_i} \oplus a_{n,i}, \quad (1)$$

where  $\oplus$  denotes element-wise addition. In Section 5, we present our experimental result that the rehearsal embedding is better at modelling the KT problem, comparing with the Rasch model based embedding [15]. Furthermore, KT is characterized by temporal orderings of interactions. Capturing the underlying temporal dependencies between the interactions requires additional position embedding. Therefore, we apply the **positional encoding** module that uses fixed sinusoids of different frequencies [34] to model relative orderings. The position embedding has been added to the rehearsal embedding, constituting the final input to the next layer.

**Multi-head attention layer** serves as the aggregator of knowledge within a single session. By performing a weighted sum over the contextualized knowledge (state) representations of each interaction, this layer returns a within-session knowledge representation. To facilitate the within-session knowledge aggregation, we append a special token AKSS at the end<sup>4</sup> of the input sequence to the interaction encoder. This token corresponds to and outputs the within-session knowledge representation aggregated from the contextualized representations of every interaction.<sup>5</sup> In this case, the query to the attention layer is mapped from the rehearsal embedding  $x_{n,AKSS} \in \mathbb{R}^d$  of the special token AKSS in session  $n$ . The input embedding (i.e. rehearsal embedding + position embedding) of each interaction is mapped into the keys and values. More specifically, the mapping can be formulated as follows:

$$q_{n,AKSS} = W_Q^T x_{n,AKSS}; \quad K_n = W_K^T X_n; \quad V_n = W_V^T X_n \quad (2)$$

where  $W_Q, W_K, W_V \in \mathbb{R}^{d \times d_m}$  are learnable mapping matrices, with  $d_m$  being the dimension after mapping, and  $X_n^T \in \mathbb{R}^{t \times d}$  is the input embedding matrix of all the interactions within the  $n$ th session. Then, the scaled dot-product attention scores  $\alpha_{n,AKSS} \in \mathbb{R}^t$  are computed as:

$$\alpha_{n,AKSS} = \text{Softmax} \left( \frac{K_n^T q_{n,AKSS}}{\sqrt{d_{K_n^T}}} \right) \quad (3)$$

where  $d_{K_n^T} = d$  is the dimension of the key embedding matrix. After the Softmax function is applied to obtain  $\alpha_{n,AKSS}$ , the inner-session knowledge representation  $h_{n,AKSS}^{Inner} \in \mathbb{R}^d$  is output as:

$$h_{n,AKSS}^{Inner} = \sum_t \alpha_{n,t,AKSS} \times v_{n,t} \quad (4)$$

where  $v_{n,t} \in \mathbb{R}^{d_m}$  is the value embedding corresponding to the  $t$ th interaction.

**Feed-Forward neural network (FFN)** module then takes in the output  $h_{n,AKSS}^{Inner}$  from the attention layer.  $h_{n,AKSS}^{Inner}$  is passed through a two-layer multi-layer perceptron with ReLU activation to obtain the following final output of the interaction encoder:

$$h_{n,AKSS}^{Inner} := \Phi_2^T \sigma(\Phi_1^T h_{n,AKSS}^{Inner} + b_1) + b_2 \quad (5)$$

where  $\Phi_1 \in \mathbb{R}^{d \times d'_m}$ ,  $\Phi_2 \in \mathbb{R}^{d'_m \times d}$  and  $b_1 \in \mathbb{R}^{d'_m}, b_2 \in \mathbb{R}^d$  are respectively the learnable weight matrices and bias vectors of the FFN, while  $\sigma(\cdot)$  denotes the ReLU activation function.

Following the above procedures, each past session in  $\{ses_1^i, ses_2^i, \dots, ses_n^i\}$  will be encoded by the interaction encoder into a sequence of intra-session knowledge (state) representations  $\{h_{1,AKSS}^{Inner}, h_{2,AKSS}^{Inner}, \dots, h_{n,AKSS}^{Inner}\}$ . These representations will then be used to construct a historical intra-session knowledge representation matrix  $H_n^T \in \mathbb{R}^{n \times d}$  and be fed into the subsequent session encoder.

<sup>4</sup> This placement is commonly performed by the neural language models for NLP tasks [39].

<sup>5</sup> For example,  $x_{1,AKSS}$  is rehearsal embedding of AKSS in 1th session.

#### 4.1.2. Session encoder

As discussed in Section 2, the time intervals and the intensity of the practice sessions significantly affect students' memory retention of their acquired within-session knowledge. This process is governed by the memory consolidation and forgetting mechanism. According to studies [21–26], students' long-term learning performance is dependent on the retrieved knowledge across sessions. Therefore, we propose to design a session-level encoder capable of modelling the sessional consolidation and forgetting mechanism of students' memory.

Following each practice session, the consolidation process summarizes and aggregates all acquired knowledge. The knowledge state heavily relies on memory retention during the consolidation process. Consequently, we model the sessional knowledge “consolidation” by employing an attention mechanism and address forgetting behaviour by explicitly adjusting attention to decay further into the past. According to the findings from the memory studies, forgetting curves are closely approximated by power-law curves [42–44]. Therefore, it is reasonable to assume that the attention, cast by the memory consolidation, is subject to such forgetting curves. In other words, the influence of the past practice sessions on the consolidation will decay with a power-law effect. This leads to our design of a new monotonic, power-law-decay session-level attention mechanism.

Our proposed **memory-decay session-level attention** mechanism is based on (1) an Inter-session Knowledge State Storage (RKSS) token, and (2) a power-law-decay function on the time gaps between two sessions. More specifically, the RKSS token is a session-level special artificial token, and the rehearsal embedding representation ( $x_{n,RKSS}$ ) of the RKSS token in session  $n$  corresponds to the inter-session knowledge (state) representation  $h_{n,RKSS}^{Inter}$  that aggregates inner-session knowledge representations  $\{h_{1,AKSS}^{Inner}, h_{2,AKSS}^{Inner}, \dots, h_{n,AKSS}^{Inner}\}$  across all the previous sessions. The power-law-decay function, drawing inspiration from memory decay research [45–47], computes the decay factor  $\xi_{n,j}$  from the time of the  $j$ th session  $T_j$  to that of the  $n$ th session  $T_n$  is shown as follows:

$$\xi_{n,j} = \frac{1}{(T_n - T_j) * S} \quad (6)$$

where  $S$  is a tunable hyper-parameter for the stability of memory retention and  $(T_n - T_j)$  is the time gap between the target session  $n$  and the previous session  $j$ . Finally, the power-law-decay session-level attention is formulated as follows:

$$q_{n,RKSS} = W_Q'^T x_{n,RKSS}; \quad K'_n = W_K'^T H_n; \quad V'_n = W_V'^T H_n$$

$$\alpha_{n,RKSS} = \text{Softmax} \left( \frac{K_n'^T q_{n,RKSS}}{\sqrt{d_{K_n'^T}}} \right) \quad (7)$$

$$h_{n,RKSS}^{Inter} = \sum_{n'=1}^n \alpha_{n,n',RKSS} \times \xi_{n+1,n'} \times v'_{n'}$$

where  $W_Q', W_K', W_V' \in \mathbb{R}^{d \times d_m}$  are learnable mapping matrices of the session-level attention;  $x_{n,RKSS}$  is the rehearsal embedding of the RKSS token at the end of session  $n$ , which “stores” the inter-session knowledge before the start of session  $n + 1$ ;  $v'_{n'} \in \mathbb{R}^{d_m}$  is the  $n'$ -th column of the value (embedding) matrix  $V_n' \in \mathbb{R}^{d_m \times n}$ . Finally, the acquisition & consolidation modelling component produces  $h_{n,RKSS}^{Inter}$ , an inter-session knowledge representation encompassing all the preceding practice sessions. The output representation undergoes layer normalization to reshape its size of dimension to  $d$ .

#### 4.2. Retrieval & Responding modelling component

The RR component leverages a **knowledge state retrieval (KSR)** encoder to retrieve and refine a student's stored inter-session knowledge representation. The representation is then fed into an **answering decoder** module to predict the student's next interaction in the current session. The decoder computes the student's current knowledge state by integrating the inter-session representation with the intra-session information from the previous interactions in the current session.

#### 4.2.1. Knowledge state retrieval encoder

Memory retrieval is a crucial mechanism facilitating humans to respond based on past experiences. As part of the learning process, a student's brain initially retrieves relevant knowledge states from their memory storage. These retrieved states serve as working states, preparing the student for the upcoming study task when a new practice session commences [21,22,25]. This process is illustrated in Fig. 2.

The KSR encoder models the human memory retrieval process, in which the representation of a student's inter-session knowledge state is retrieved. The representation is then refined by each and every previous response in the current  $(n + 1)$ th session, i.e.,  $\{a_{n+1,0}^i, a_{n+1,1}^i, a_{n+1,2}^i, \dots, a_{n+1,t-1}^i\}$ , before fed into the decoder module to be used to predict the corresponding next answers  $\{a_{n+1,1}^i, a_{n+1,2}^i, \dots, a_{n+1,t}^i\}$ . In this case,  $a_{n+1,0}^i \in \mathbb{R}^d$  is artificially padded. The padding does not correspond to the correctness of any response but serves as a symbol that activates the retrieval at the start of the current session. For clarity, we replace/rename it by a "Retrieval Trigger" (RT) token. The corresponding representation of which gets integrated with the inter-session knowledge representation to be fed into the decoder module for predicting the very first interaction. In other words, the RT token enables the retrieval of all the necessary intra-session information preceding to the current question to be answered. In addition, the refinement of the inter-session knowledge representation  $h_{n,RK,SS}^{\text{Inter}}$  at every position before the current one includes an element-wise addition. The addition involves the student's response correctness embedding  $a_{n+1,t-1}$  at those positions and their corresponding positional embeddings. Furthermore, similar to the interaction encoder, the KSR encoder pipelines a multi-head attention layer and a feed-forward neural layer. The structure enables the KSR encoder to further exchange the refined knowledge at every position up to the current interaction. The final outputs of the KSR encoder serve as the key and value knowledge representations for the current query question, all of which are inputs to the decoder module.

#### 4.2.2. Decoder

We leverage a transformer decoder to predict the outcome of the current interaction. The output of the KSR encoder from every position up to the current one consists of the key and value embedding matrices. The aggregated skill and index embedding of the current question  $q_t$  (i.e.,  $k_{n+1,t} \oplus d_{n+1,q_t}$ ) serves as the query embedding. Through the multi-head attention layer and the FFN layer of the transformer decoder, we obtain the final knowledge state representation for the current position (i.e.,  $h_{n+1,t}^{\text{Answer}} \in \mathbb{R}^{d_m}$ ). The representation is then fed into a FFN-based prediction layer that produces the corresponding correctness prediction ( $\hat{a}_{n+1,t} \in [0,1]$ ) for the query question. Finally, for the training of HiTSKT, the following binary cross-entropy loss between the ground-truth answers and the corresponding predicted answers over each student's individual learning histories needs to be minimized:

$$\mathcal{L} = - \sum_i \sum_n \sum_t (a_{n,t}^i \log(\hat{a}_{n,t}^i)) + ((1 - a_{n,t}^i) \log(1 - \hat{a}_{n,t}^i)) \quad (8)$$

## 5. Experiments

In this section, we will introduce the experiments conducted to evaluate our proposed model. We train and test our model on four benchmark datasets and compare the performance of our model with state-of-the-art KT models. Note that, the code of our model and also the implementation of the compared models are publicly available.<sup>6</sup>

### 5.1. Datasets

Four real-world benchmark KT datasets with time stamp information (i.e., Algebra2005,<sup>7</sup> ASSISTments2017,<sup>8</sup> Junyi<sup>9</sup> and Riiid's EdNet dataset [48]) are used to evaluate the performance of HiTSKT. The Algebra2005 dataset contains 13–14-year-old students' responses to Algebra questions presented at KDDcup 2010 Educational Data Mining challenge. The ASSISTments dataset is collected from an online high school mathematics tutoring platform. Junyi dataset consists of millions of exercise attempt logs on Junyi Academy Foundation Platform. Riiid's EdNet data is the world's largest KT related open database containing hundreds of millions of students' interactions (records).

**Pre-processing.** As for all four datasets, we remove all the interactions where skill information is Null. All interactions where the time spent is more than 9999 s are dropped, as these interactions might not be answered or completed. The number of occurrences of each question are also labelled by simply counting on all four datasets as well.

**Session Division.** According to the start time of the next interaction and the end time of the previous interaction information, we first calculate the time interval between each two neighbouring interactions. Then, the two consecutive interactions whose time intervals are larger than 10 h are separated as two different sessions based on human activity studies.<sup>10</sup>

**Datasets Statistics.** The detailed statistics of these datasets after preprocessing are shown in Table 2. Avg.Sessions denotes the average number of practice sessions for each student in the dataset and Avg. Interactions means number of interactions in one session on average. EdNet is the largest dataset with more than 88,000,000 interactions, while ASSISTments2017 has the most average number of interactions (70) for each session.

**Session Information Analysis.** To complement with Fig. 1 where a single student's interaction time gap distribution is shown, we conduct a statistical analysis of all the students' logs time gap distribution in Fig. 5 on the same dataset. In this figure, we only show the time intervals that are smaller than 100 h as even larger intervals clearly mean different sessions. From this figure, we can see that the time intervals between two consecutive interactions follow a long-tail distribution. The majority of consecutive interactions happen within a short time interval, while much fewer of instances have a large time interval. Therefore, we could conclude that histories of students' interaction records in reality usually exhibit clustering effects. Each learning history comprises a series of separated practice sessions, with each session containing a sequence of practice interactions. This together with Fig. 1 consolidate the universality and naturality of session information in KT task.

Compared with the other three datasets, **the information for each session in ASSISTments2017 is more abundant.** On the other hand, the information for each student in Junyi and EdNet are sparser compared with ASSISTments2017 dataset as there are only 20 and 23 sessions on average for each student separately and merely 24 and 28 interactions in one session on average correspondingly as shown in Table 2.

From Fig. 6, the distribution of the number of practice over sessions on ASSISTments2017 is close to the normal distribution, which means that the number of interactions for most sessions are around the mean number, which is 54. On the contrary, the distributions are skewed to less than the mean numbers on both Junyi and EdNet dataset, which means that only very few sessions on Junyi and EdNet datasets have abundant practice information. Obviously, the students' histories records on Junyi and EdNet are sparser.

<sup>7</sup> Algebra2005 source: <https://pslcdatashop.web.cmu.edu/KDDCup>.

<sup>8</sup> ASSISTments2017 source: <https://sites.google.com/site/assistmentsdata>

<sup>9</sup> Junyi source: <https://www.kaggle.com/junyiacademy>

<sup>10</sup> People working hours per day [40].

<sup>6</sup> <https://github.com/pokerme7777/HiTSKT>

**Table 2**  
Datasets statistics.

| Features/Datasets           | Algebra2005 | ASSISTments2017 | Junyi      | EdNet      |
|-----------------------------|-------------|-----------------|------------|------------|
| Interactions                | 596,963     | 942,807         | 14,660,217 | 88,597,714 |
| Students                    | 460         | 1709            | 29,865     | 131,538    |
| Questions                   | 171,143     | 3162            | 25,630     | 13,523     |
| Skills                      | 271         | 102             | 1326       | 10,000     |
| Avg.Sessions                | 28          | 8               | 20         | 23         |
| Avg.Interactions/Session    | 45          | 70              | 24         | 28         |
| Median.Interactions/Session | 37          | 54              | 16         | 16         |

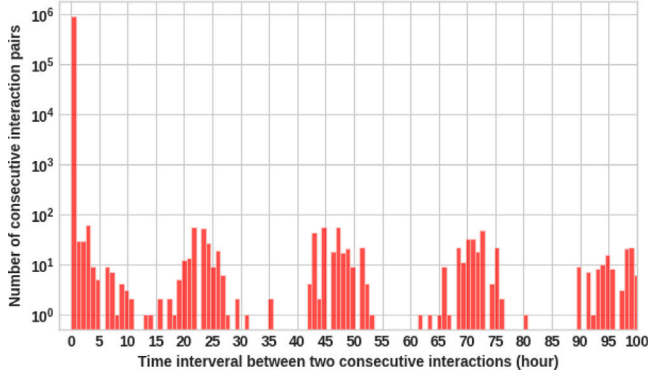


**Fig. 5.** Histogram of time intervals between two consecutive interactions on ASSISTments2017 dataset. The x-axis means the time gap between two consecutive interactions and the y-axis means the number of interaction pairs.

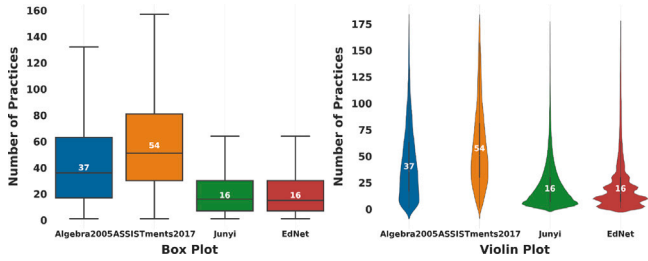


**Fig. 6.** Box-Plot and Violin-Plot for the number of interactions in each session on four datasets.

**Session Sequence Length and Interaction Sequence Length.** Our model is transformer-based, thus the number of history sessions before the current session and the number of interactions inside each session are fixed in our model. The number of history sessions is determined by the third quartile (Q3) of all students' session numbers and we round this value to the closest power of 2. Similarly, we choose Q3 of the number of interactions inside each session as the interaction sequence length of sessions. The number of history sessions is 16 on all four datasets. The number of interactions in each session is 64 on Algebra2005 and ASSISTments2017 dataset, and 32 on Junyi and EdNet datasets.

For longer interaction or session sequences, a strategy of truncating earlier interactions or sessions is employed. Conversely, in the case of shorter sequences, padding techniques are applied to fill the vacancy. For instance, in the context of the ASSISTments2017 dataset, 16 is used as the number of history sessions (i.e., session sequence length) and earlier sessions are trimmed if a student has more than 16 sessions. "session padding" is then applied to fill the blank if a student has less than 16 sessions. Session-level padding consists of a sequence of "interaction padding", designed to align with the dimensions of other sessions. Similarly, the most recent 64 interactions are utilized to construct the interaction sequences, enabling the modelling of intra-session information. Interaction sequences that lack sufficient interactions are

augmented using "interaction padding" to ensure the completion of the sequence.

## 5.2. Baselines and evaluation metric

We compare our HiTSKT model with the following advanced KT models:

- DKT [9] is the first KT model with the deep neural network. It uses long-short term memory recurrent neural network (i.e., LSTM) to make sequential prediction of students' performance. The hidden state of the LSTM can be extracted to infer the student's mastery level of different skills.
- DKVMN [10] is an extended DKT model with an extra memory augmented neural network to evaluate questions and users' knowledge state separately by two matrices. Its "key" matrix contains the fixed representation of each skill, and its dynamic "value" matrix models each learner's mastery level of each skill. Also, DKVMN leverages separate "read" and "write" processes on these two matrices and be more flexible than DKT.
- SAKT [14] is the first model utilizing the self-attention mechanism to assign weights to students' responses and its structure is similar to the transformer model.
- AKT [15] is the state-of-the-art KT model that achieved the highest performance score on several benchmark datasets. It adopts a novel monotonic attention mechanism to model the influence of a skill learned from a distant past on a student's knowledge state. It also incorporated the difficulty coefficient to encode the skills.
- HawkesKT [33] is the first model that leverages Hawkes process to model the temporal information in students' learning history. More specifically, it adopts the mutual excitation mechanism and the kernel function to model the temporal cross-effect and control adaptive temporal evolution.
- ATKT [32] adopts adversarial training with attentive LSTM to model the students' knowledge state. This model has a knowledge hidden state attention module that could adaptively aggregate information. Moreover, the model was trained with perturbation to improve model robustness.
- RKT [49] combines attention weight with the exercise relation extracted from textual questions' information and the forgetting ratio, assuming the student's record is a continuous single sequence.

The metrics used to evaluate the model performance in this paper are the widely-used area under the ROC (Receiver Operating Characteristics) which is known as AUC (Area Under the Curve), F1-score (F1), accuracy (ACC) and mean squared error (MSE) [10,14,15,18,30,32,33,37,38,50].

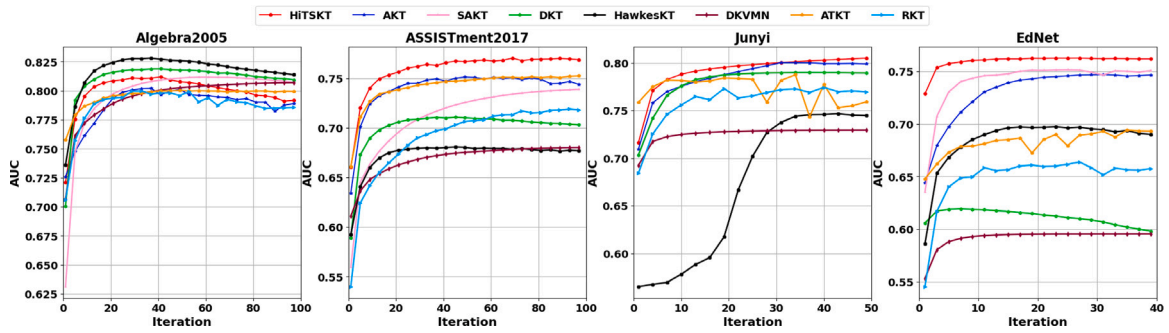
## 5.3. Training and testing

**Dataset Division.** We use the first 60% sessions for each student to train, the next 20% sessions to do validation and the last 20% sessions to test. Our model does not predict the first session for all students on all datasets because our model structure needs to encode previous sessions information.

**Table 3**

The performance of HiTSKT and all the baselines on four datasets.

| Dataset         | Metric | DKVMN  | DKT    | HawkesKT | RKT    | SAKT   | ATKT   | AKT    | HiTSKT        |
|-----------------|--------|--------|--------|----------|--------|--------|--------|--------|---------------|
| Algebra 2005    | AUC    | 0.7730 | 0.7898 | 0.7802   | 0.7741 | 0.7927 | 0.7989 | 0.8039 | <b>0.8142</b> |
|                 | F1     | 0.8636 | 0.8611 | 0.8574   | 0.8561 | 0.8734 | 0.8712 | 0.8635 | <b>0.8744</b> |
|                 | ACC    | 0.7833 | 0.7811 | 0.7798   | 0.7780 | 0.8026 | 0.7960 | 0.7896 | <b>0.8027</b> |
|                 | MSE    | 0.1523 | 0.1461 | 0.1520   | 0.1558 | 0.1423 | 0.1435 | 0.1495 | <b>0.1420</b> |
| ASSISTments2017 | AUC    | 0.6695 | 0.6888 | 0.6762   | 0.7005 | 0.7187 | 0.7417 | 0.7392 | <b>0.7553</b> |
|                 | F1     | 0.4200 | 0.4776 | 0.4565   | 0.5558 | 0.5200 | 0.5135 | 0.5477 | <b>0.6131</b> |
|                 | ACC    | 0.6572 | 0.6687 | 0.6549   | 0.5848 | 0.6800 | 0.6846 | 0.6886 | <b>0.6943</b> |
|                 | MSE    | 0.2170 | 0.2104 | 0.2240   | 0.2285 | 0.2061 | 0.2028 | 0.1998 | <b>0.1988</b> |
| Junyi           | AUC    | 0.7264 | 0.7386 | 0.7296   | 0.7635 | 0.7852 | 0.7850 | 0.7850 | <b>0.7911</b> |
|                 | F1     | 0.8108 | 0.8153 | 0.8044   | 0.8195 | 0.8249 | 0.8233 | 0.8286 | <b>0.8302</b> |
|                 | ACC    | 0.7178 | 0.7271 | 0.7182   | 0.7366 | 0.7479 | 0.7431 | 0.7540 | <b>0.7541</b> |
|                 | MSE    | 0.1872 | 0.1818 | 0.1970   | 0.1792 | 0.1694 | 0.1705 | 0.1654 | <b>0.1652</b> |
| EdNet           | AUC    | 0.5967 | 0.6209 | 0.6880   | 0.6569 | 0.7513 | 0.6935 | 0.7464 | <b>0.7615</b> |
|                 | F1     | 0.8050 | 0.8050 | 0.7988   | 0.8087 | 0.8062 | 0.8030 | 0.8109 | <b>0.8141</b> |
|                 | ACC    | 0.6796 | 0.6840 | 0.7012   | 0.6788 | 0.7247 | 0.6882 | 0.7259 | <b>0.7268</b> |
|                 | MSE    | 0.2135 | 0.2092 | 0.2093   | 0.2202 | 0.1820 | 0.2020 | 0.1819 | <b>0.1816</b> |

**Fig. 7.** Convergence curves based on validation AUC for all models on four datasets.**Table 4**

Models complexity and memory usage.

|                       | DKVMN  | DKT   | HawkesKT | RKT   | SAKT  | ATKT | AKT   | HiTSKT |
|-----------------------|--------|-------|----------|-------|-------|------|-------|--------|
| FLOPs (M)             | 17     | 13    | 51       | 92    | 260   | 100  | 508   | 199    |
| Parameters (M)        | 0.043  | 0.103 | 0.27     | 11.10 | 17.52 | 0.26 | 3.02  | 45.68  |
| GPU memory usage (MB) | 173.27 | 0.48  | 2.50     | 44.50 | 70.17 | 1.11 | 12.13 | 183.04 |

**Implementation.** HiTSKT is implemented with PyTorch, and we used Adam optimiser to train our model. All experiments are conducted on a server of which the computing core is NVIDIA V100 GPU with 16 GB memory. Constrained by the memory of the GPU, we train the model with a batch size of 64 for all datasets.

**Parameters Settings.** As for comparison fairness, for HiTSKT and all the baselines, we tune embedding size among {64, 128, 256, 512} and learning rate among  $\{1e-5, 5e-5, 8e-5, 1e-4, 1e-3\}$  on all datasets. For parameters that are specific to one baseline, we tune them in the recommendation range presented in the paper of that baseline. For each model, we run it five times on all four datasets, and the average results are reported.

#### 5.4. Main results and discussion

Before we introduce the effectiveness of HiTSKT in KT problem compared with other baselines, we first have a look at the convergence rate of HiTSKT compared with other baselines. The results are shown in Fig. 7 and they show that HiTSKT is robust and converges quickly than other compared models.

Table 3 lists the performance of HiTSKT and other KT methods across four datasets and the following observations are made:

- HiTSKT outperforms all the compared state-of-the-art models on all four datasets (i.e., Algebra2005, ASSISTments2017, Junyi and EdNet) which validates the effectiveness of HiTSKT in KT problem.

- The performance of HiTSKT is related to the session information enrichment degree. As shown in Table 3, compared with baselines, HiTSKT achieves more improvement on ASSISTments2017 than the improvements on Junyi and Ednet as ASSISTments2017 has richer session information as shown in Fig. 6.
- Overall, the attention based models (i.e., AKT, SAKT, ATKT and HiTSKT) perform better than the other models across all the datasets. This is consistent with the superior performance of attention based models in other areas and also confirms the correctness of designing our model based on attentive models.
- In general, self-attention baseline models (e.g., AKT and SAKT) perform better on larger datasets when compared with other baseline models. The gaps between self-attention based models and other baseline models are minimal on the Algebra2005 and ASSISTments2017 dataset which have less number of interactions as shown in Table 2. On the other hand, AKT and SAKT show more obvious improvements compared with the other baseline models on the EdNet dataset which has the most interactions as shown in Table 2. The underlying reason might be that these self-attention models are more complex containing more parameters that require more data to prevent the overfitting problem.
- Different from the other attention based models, ATKT performs better on smaller datasets (e.g., Algebra2005, ASSISTments2017) compared with its performance on larger datasets (e.g., EdNet). This might be due to the adversarial design of ATKT. The adversarial examples from the adversarial attack create notorious noise

**Table 5**  
HiTSKT performance (AUC) table — Ablation study.

| Validation focus       | Subject                                       | Algebra2005   | ASSISTments2017 | Junyi         | EdNet         |
|------------------------|-----------------------------------------------|---------------|-----------------|---------------|---------------|
| Rehearsal embedding    | Replaced by Rasch Model-Based Embeddings [15] | 0.7923        | 0.7417          | 0.7857        | 0.7569        |
|                        | Without $k_{n,l}$                             | 0.7777        | 0.7407          | 0.7890        | 0.7456        |
|                        | Without $d_{n,q_i}$                           | 0.7777        | 0.6929          | 0.7438        | 0.6369        |
|                        | Without $f_{n,q_i}$                           | 0.8003        | 0.7459          | 0.7897        | 0.7554        |
|                        | Without $a_{n,l}$                             | 0.7916        | 0.7529          | 0.7766        | 0.7437        |
| AKSS & RKSS            | Without AKSS                                  | 0.8042        | 0.7536          | 0.7897        | 0.7520        |
|                        | Without RKSS                                  | 0.8035        | 0.7540          | 0.7717        | 0.7601        |
|                        | Without AKSS and RKSS                         | 0.8034        | 0.7433          | 0.7774        | 0.7417        |
| Positional encoding    | Without Positional Encoding                   | 0.7921        | 0.7552          | 0.7904        | 0.7599        |
| Memory-decay attention | Replaced by monotonic Attention               | 0.7845        | 0.7390          | 0.7841        | 0.7331        |
| KSR encoder            | Without KSR                                   | 0.7905        | 0.7443          | 0.7819        | 0.7589        |
|                        | Replaced by right-shift correctness           | 0.7916        | 0.7520          | 0.7879        | 0.7588        |
| HiTSKT                 | –                                             | <b>0.8142</b> | <b>0.7553</b>   | <b>0.7911</b> | <b>0.7615</b> |

to confuse the neural network during training which makes the model more robust on small datasets.

Table 4 lists the complexity and memory usage of HiTSKT and other KT methods. To measure complexity and GPU memory usage, we employ a batch size of ten from the Algebra2005 dataset and report the average FLOPs, the number of parameters in the model, and GPU memory usage throughout the training process.

The major complexity of our proposed HiTSKT model comes from the modelling of session information. More specifically, for a student's learning history, HiTSKT models multiple session information using multiple action encoders. However, the action encoders are naturally independent which leads to a natural parallel computation for these action encoders. Thus, we can see from Table 4, compared with the other models, HiTSKT achieves a better performance with higher space complexity but no higher time complexity. More specifically, HiTSKT has lower time complexity compared to AKT (with 508M FLOPs), while surpassing AKT in terms of AUC, F1, ACC, and MSE. Additionally, the actual time gaps between training HiTSKT and training other lighter KT models in our experiment are relatively small. On the largest dataset (EdNet) with size of 88,597,714 records, the difference is less than two hours, and there is no significant difference on the smallest dataset (Algebra2005). Improving model complexity and memory usage will be left for future work.

ATKT, HawkesKT, DKT, and DKVMN are considered relatively lightweight models due to their lower parameter counts. However, it is worth noting that ATKT exhibits a higher time complexity (FLOPs) of 100 M, attributed to increased computational load in the LSTM and fully connected network layers, with FLOPs amounting to 87.04 M. Additionally, DKVMN exhibits higher GPU memory consumption (173.27 MB) relative to other lightweight KT models. The increased memory usage can be ascribed to specific features inherent in the training process, encompassing activation maps, intermediate representations, and activation functions. The distinctive design of DKVMN, featuring key and value memory components, contributes to its enhanced performance but concurrently demands more memory resources.

### 5.5. Ablation study

In this section, we present our ablation studies to validate the effectiveness of the key components of our model, including the implementation of rehearsal embedding, artificial tokens (i.e., AKSS and RKSS), Power-law Decay Attention and KSR encoder. Table 5 lists the results of ablation studies.

**Rehearsal Embedding.** Rehearsal embedding generates the initial embedding for the inputs by mining the potential relationship between

two groups of inputs including skills and questions difficulty and the student's learning ability. In this part, we compare our rehearsal embedding with the Rasch model-based embedding which is another very popular way of learning the representation for relational data in KT. Specifically, for the Rasch model-based embedding approach, we utilized the method from [15] and IRT models [51]. Applying two scalars, the model measures both the questions' difficulty and the students' ability. However, the model's performance reduced by around 0.8% on four datasets, which means rehearsal embedding is more suitable to address KT problem. Furthermore, HiTSKT's performance is still higher than others' when using Rasch model-based embedding method. That reveal the HiTSKT successfully model session-aware memory retention process.

Based on the experiment result table presented in Table 5, it is evident that all four embeddings ( $a_{n,l}$ ,  $d_{n,q_i}$ ,  $f_{n,q_i}$ ,  $k_{n,l}$ ) exert a significant impact on the performance of HiTSKT across all four datasets. Within the context of the ablation study encompassing these datasets, the embedding representing question index  $d_{n,q_i}$  exhibits a pronounced influence on the model's performance. Concurrently, the influence stemming from the ordinal label pertaining to the occurrence number of the question  $f_{n,q_i}$  or the correctness of the student's response  $a_{n,l}$  is slightly less pronounced in comparison to  $d_{n,q_i}$ . Overall, the amalgamation of all four embeddings enhances the model's performance and imparts robustness to the KT model.

**AKSS and RKSS.** AKSS and RKSS are designed to model the hierarchical structure. The underlying technical challenge in modelling this hierarchical structure is that we need to condense a sequence of interaction representations matrix into one vector for each session. Apart from AKSS and RKSS, Pooling is a standard method to do this [52–54]. As introduced by Pan et al. [55], it has been demonstrated that the utilization of an extra class token is not necessary. Instead, the direct application of average pooling over patch tokens has the potential to yield improved performance. Therefore, in this experiment, we use Average-Pooling (Eq. (9)) to replace the AKSS and RKSS tokens. The  $\kappa$  is the kernel size. In our experiment, the default value of  $\kappa$  is equal to the number of preceding practice sessions  $n$ .

$$h_{n,RKSS}^{Inter} = \frac{1}{\kappa} \sum_{m=1}^{\kappa} h_{m,AKSS}^{Inner} \quad (9)$$

However, different with the argument with [52,55], our experiment result illustrate the importance of two designed tokens (AKSS and RKSS). The AUC of the model without AKSS reduced by around 0.3% compared with HiTSKT. Our experimentation reveals that the RKSS token holds substantial influence over the performance of the KT model. As evident from Table 5, omitting the RKSS token results in a

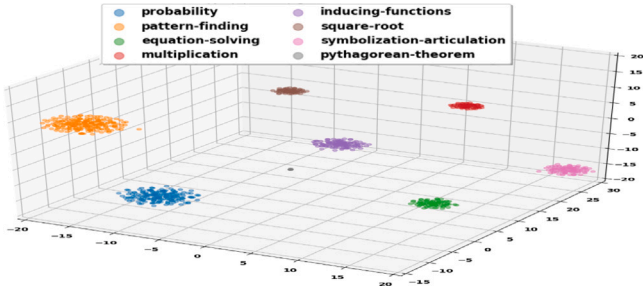


Fig. 8. Question Embedding Visualization. All question indexes are presented as dot points in three-dimensional space, and the colours represent the corresponding skills.

minimum performance drop of 0.13% AUC for HiTSKT. Additionally, if both the AKSS and RKSS tokens are excluded, the performance reduction reaches at least 1.2% AUC. Theoretically, average pooling can capture information from all the new representations of  $h_{n,t}^{\text{inner}}$  or all the session histories representations  $z_{n,t}$ . However, according to our experiment results, artificial tokens are more robust and significantly improve model performance.

**Power-law Decay Attention.** We designed the Power-law Decay Attention mechanism to capture students' forgetting behaviour over time. In this section, we compare this attention mechanism with the scaled dot-product attention used in other attention-based KT models [14,15,49]. The results indicate that our designed power-law decay attention mechanism can enhance the model's AUC performance. Specifically, the AUC performance improved from 78.45% to 81.42% in the Algebra2005 dataset, from 73.9% to 75.53% in the ASSISTments2017 dataset, from 78.41% to 79.11% in the Junyi dataset, and from 73.31% to 76.15% in the EdNet dataset.

**KSR encoder.** HiTSKT is the pioneering KT model that incorporates session information. Consequently, we are the first to introduce a method (KSR encoder) that integrates representations from previous sessions and interactions within the current session. In order to validate the efficacy of this encoder, we have conducted two sets of experiments. The first experiment involves replacing the KSR encoder with a right shift correctness embedding sequence, while the second experiment entails removing the KSR encoder entirely.

When we replaced the KSR encoder with a sequence of right shift correctness embeddings, there was a drop in the model's performance. The adoption of a right shift correctness embedding sequence involves the inclusion of a padding token and the correctness of the student's all previous interactions in the current session. This inclusion enhances the availability of current session information. The application of a padding token serves to prevent any inadvertent leakage of the student's correctness on ongoing questions while concurrently allowing access to all prior interaction correctness. It is noteworthy that the initial token within the right shift correctness embedding represents an artificial token. Notably, the approach of employing right shift correctness embedding to attain instant feedback for the current session is similar to the technique employed in the AKT model. In our experiment, we directly perform element-wise addition between the right shift correctness sequence and the output  $h_{n,RKSS}^{\text{inter}}$  of the Acquisition and Consolidation modelling component, without involving the attention layer. When KSR encoder is placed by right shift correctness, the models' performance dropped to 79.16%, 75.20%, 78.79% and 75.88% on Algebra2005, ASSISTments2017, Junyi and EdNet respectively. If the KSR encoder is eliminated and  $h_{n,RKSS}^{\text{inter}}$  is directly transmitted to the decoder without incorporating previous correctness within the current session, the model's performance experiences a reduction to 79.05%, 74.43%, 78.19%, and 75.89%, respectively. All of these results collectively confirm the effectiveness of the KSR encoder.

## 5.6. Further model interpretation through visualization

In this session, we will interpret HiTSKT by visualizing questions' index  $d_{n,j}$  embedding and exploring example students' knowledge state using ASSISTments2017.

**Visualizing Questions' Index Embedding.** We first select eight high-frequency skills, as shown in Fig. 8 legend, and get all corresponding questions' index embedding in ASSISTments2017. For these questions' index, we can get their 256-dimensional embeddings after training the model. Then t-SNE [56] is used to reduce the dimension to 3 to better visualize them. In Fig. 8, all questions indexes are presented as dot points in three-dimensional space, and the colours represent the corresponding skills. We observe that all questions with the same skill are clustered together in space, which means questions with the same skill will be projected to close value in the embedding space. Thus, predicted students' performance on the questions will highly relate to skill mastery, which is consistent with cognitive. In other words, HiTSKT reasonably projects question indexes into embedding space which will enable a more effective calculation of students' knowledge state.

**Investigation into model performance and memory decay involves tuning the value of the stability of memory retention ( $S$ ).** We explore the relationship between model performance and the stability of memory retention  $S$ . In light of our experimental outcomes, the optimal range for the stability of memory retention ( $S$ ) is observed to fall within the interval [0.05, 1], as illustrated in Fig. 9(a). Notably, the associated memory decay factor ( $\xi$ ), as evidenced by the memory decay curve in the ASSISTments2017 dataset, bears a noteworthy resemblance to the curve depicted in Fig. 9(b).

When the value of the tunable hyper-parameter  $S$ , becomes too large, the model's performance tends to deteriorate. Building upon Eq. (6) for power-law decay attention, if the stability of memory retention ( $S$ ) assumes a significantly large value, approaching infinity, the decay factor ( $\xi_{n,j}$ ) tends to approach 0. Consequently, the outputs of the acquisition & consolidation modelling component ( $h_{n,RKSS}^{\text{inter}}$ ) would tend to converge towards 0. This outcome implies that the interactions from previous practice sessions would be deprioritized, allowing the model to emphasize solely the current session. This condition leads to a model performance closely resembling that of the AKT model as illustrated in Fig. 9(a). However, HiTSKT is a session-aware model that incorporates innovative artificial inter- and inner-session tokens. These tokens serve to aggregate learning patterns across various student proficiency levels. This unique feature enables HiTSKT to outperform AKT, even without factoring in records from previous practice sessions.

Conversely, when the stability of memory retention ( $S$ ) is excessively small, the impact of the temporal gap weight ( $T_n - T_j$ ) is diminished, resulting in all previous sessions being treated as contributing approximately equally to the current session. This contradicts the principles of memory decay [21,22,25]. Consequently, the model's performance experiences a slight reduction when the stability of memory retention ( $S$ ) values close to 0.

**Exploration on Knowledge State Predicted by HiTSKT.** This part is to show that HiTSKT could effectively predict students' knowledge state after each practice session and the prediction is interpretable.

We conduct a random sampling of five high-frequency skills (e.g., multiplication, equation-solving, pattern-finding, subtraction and substitution) from among the top twenty most popular skills. Then, we randomly select two students (e.g., student 8 and student 954) from the pool of students who have accumulated more than nine practice session records and have studied only four of these skills (four learnt skills and one unlearned skill) during these practice sessions. Our selection of four positive cases and one negative case is a deliberate choice to ensure the effectiveness of our student sampling. Subsequently, we explore their predicted knowledge state on five skills as a demonstration. For the two selected students, we calculate the weighted arithmetic mean of their

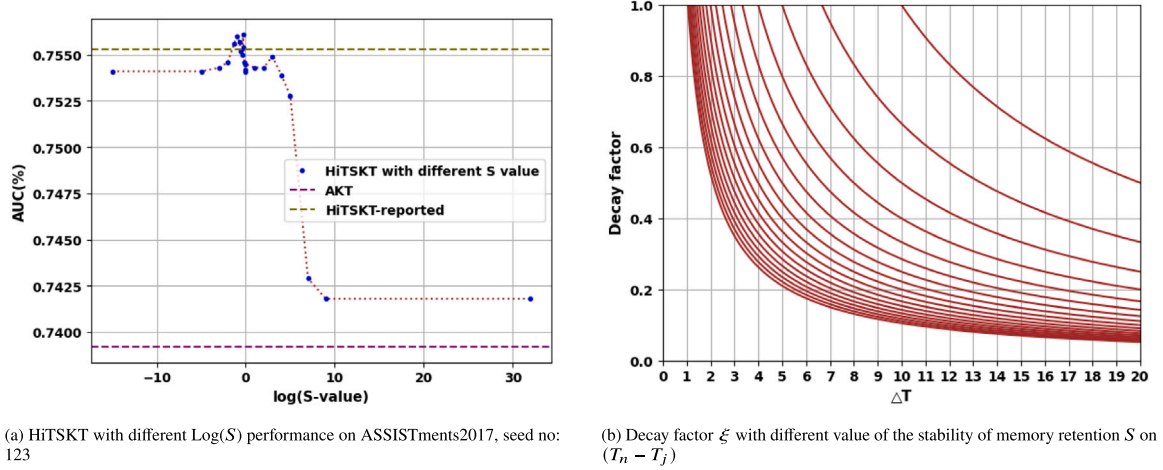
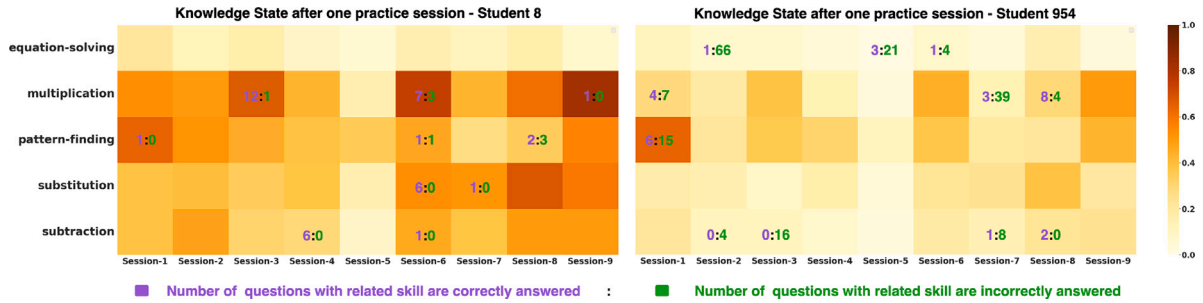
Fig. 9. Exploration on the Stability of Memory Retention ( $S$ )

Fig. 10. Knowledge State Visualization. This is an exploration figure about two students' predicted knowledge state on five skills (e.g., multiplication, equation-solving, pattern-finding, subtraction and substitution).

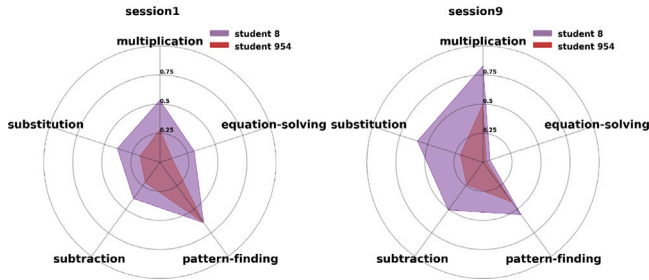


Fig. 11. Radar plots for example students' Knowledge State on five skills. Two demonstrated students' skills have been improved or forgotten after nine practice sessions and students' learning rates are different.

probabilities of correctly answering related questions for each skill after each practice session.

These mean probabilities are used as the predicted students' knowledge state (i.e., mastery level) on the corresponding skills after each practice session as shown in Fig. 10. Each block represents the predicted knowledge state of corresponding skills after a practice session. For instance, the top left block in the first subplot means the predicted knowledge state on skill (equation-solving) after session-1 is close to 0.3. The numbers in Fig. 10 represent the number of questions with related skills that are correctly/incorrectly answered. Based on this, we can tell example student 8 is an excellent student with 0.7656 overall accuracy on these five skills from session 1 to 10, while example student 954 only has 0.1776 overall accuracy. Based on Fig. 10, two observations can be made.

First, forgetting and memory re-consolidation are existing phenomena in KT and HiTSKT is able to model these behaviours. For example,

student 8 did many practices in terms of multiplication in session 3 and the accuracy rate was  $\frac{12}{12+1}$ . If there is no forgetting, then she/he should be able to well answer questions related to multiplication in all the later sessions. However, we can see that she/he showed worse performance (i.e.,  $\frac{7}{7+3}$  compared with  $\frac{12}{12+1}$ ) in session 6. This validates the existence of forgetting behaviour. Later, as more interactions were done related to this skill, student 8's master level in this skill got re-consolidated in session 9. Our model is able to capture these forgetting and re-consolidation behaviours as we can see that the colour got lighter and lighter after session 3 and the colour got darker and darker after session 6 as time went by.

Second, our model is able to effectively capture students' knowledge state change as practice goes on. This actually echoes with our first observation as it shows that our model is able to capture how student 8's knowledge state changes as she/he does more practice. Another support is the example of student 954 learning equation-solving. As this student showed bad performance in all the interactions related to this skill, our model always predicts a bad master level of this student in terms of this skill.

Overall, we know two students' skills have been improved after nine practice sessions and students' learning rates are different from Fig. 11. Student 8 made more remarkable progress, although only half the quantity questions of student 945 did during this period. Effective practice sessions can boost student knowledge (i.e., Multiplication of student 8), while recommending practice questions casually will waste students' time and have no benefit to students (i.e., Equation-solving of student 954). Meanwhile, the recommendation system should be cautious about forgetting phenomena and help students revisit a suitable time stamp (i.e., Equation-solving of student 8).

According to the visualization and analysis above, we can tell that the prediction from HiTSKT is highly consistent with the students' actual situation and fits the memory law mentioned.

## 6. Conclusion

In this paper, our emphasis is on leveraging session information extracted from students' learning histories to address the KT problem. The focus stems from a detailed exploratory data analysis of session information conducted across various real-world KT datasets. To do so, we have introduced a carefully designed hierarchical transformer model for session-aware KT. Our proposed model, HiTSKT, consists of two main components. The first is the *acquisition & consolidation* modelling component, which employs a hierarchical transformer encoder architecture to summarize two types of acquired knowledge: the (lower-level) intra-session and (higher-level) inter-session knowledge. The second is the *retrieval & responding* modelling component, which includes a knowledge retrieval module (i.e., the KSR encoder) and an answering (transformer) decoder module. Furthermore, we have designed a power-law-decay attention mechanism for HiTSKT that captures students' forgetting behaviour over their acquired inter-session knowledge as a result of long-term sessional drifts.

Extensive experiments have been conducted on four large-scale real-world KT datasets and the results show that HiTSKT achieves new state-of-the-art performance on all the datasets in terms of future response correctness prediction. Furthermore, our ablation studies have also validated the effectiveness of all the key components of HiTSKT. Visualization has also been provided which shows that HiTSKT is interpretable and is able to learn students' knowledge states effectively. Directions of future work include (1) designing more sophisticated memory-decay attention mechanism based on state-of-the-art memory retention and decay studies in Educational Psychology, and (2) investigating whether masking language modelling, a powerful technique of training transformer-based deep language models in NLP, can be adapted and applied to enable more effective training of HiTSKT for KT.

## CRedit authorship contribution statement

**Fucaí Ke:** Writing – original draft, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Weiqing Wang:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Weicong Tan:** Validation, Methodology, Conceptualization. **Lan Du:** Writing – review & editing, Software. **Yuan Jin:** Writing – review & editing, Validation, Resources. **Yujin Huang:** Writing – review & editing, Software. **Hongzhi Yin:** Writing – review & editing, Formal analysis.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

I have shared the link to my code and data.

## References

- [1] Q. Liu, S. Shen, Z. Huang, E. Chen, Y. Zheng, A survey of knowledge tracing, 2021, arXiv preprint arXiv:2105.15106.
- [2] Y. Suo, N. Miyata, H. Morikawa, T. Ishida, Y. Shi, Open smart classroom: Extensible and scalable learning system in smart space using web service technology, *IEEE Trans. Knowl. Data Eng.* 21 (6) (2008) 814–828.
- [3] E.J. Emanuel, Moocs taken by educated few, *Nature* 503 (7476) (2013) 342.
- [4] X. Song, J. Li, T. Cai, S. Yang, T. Yang, C. Liu, A survey on deep learning based knowledge tracing, *Knowl.-Based Syst.* 258 (2022) 110036.
- [5] M. Khajaj, R.V. Lindsey, M.C. Mozer, How deep is knowledge tracing? *Int. Educ. Data Min. Soc.* (2016).
- [6] A.T. Corbett, J.R. Anderson, Knowledge tracing: Modeling the acquisition of procedural knowledge, *User Model. User-Adapt. Interact.* 4 (4) (1994) 253–278.
- [7] R.S. d Baker, A.T. Corbett, V. Alevan, More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing, in: *International Conference on Intelligent Tutoring Systems*, Springer, 2008, pp. 406–415.
- [8] Z.A. Pardos, N.T. Heffernan, Kt-idem: Introducing item difficulty to the knowledge tracing model, in: *International Conference on User Modeling, Adaptation, and Personalization*, Springer, 2011, pp. 243–254.
- [9] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L.J. Guibas, J. Sohl-Dickstein, Deep knowledge tracing, *Adv. Neural Inf. Process. Syst.* 28 (2015) 505–513.
- [10] J. Zhang, X. Shi, I. King, D.-Y. Yeung, Dynamic key-value memory networks for knowledge tracing, in: *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 765–774.
- [11] T. Long, Y. Liu, J. Shen, W. Zhang, Y. Yu, Tracing knowledge state with individual cognition and acquisition estimation, in: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 173–182.
- [12] S. Liu, R. Zou, J. Sun, K. Zhang, L. Jiang, D. Zhou, J. Yang, A hierarchical memory network for knowledge tracing, *Expert Syst. Appl.* 177 (2021) 114935.
- [13] Y. Su, Z. Cheng, P. Luo, J. Wu, L. Zhang, Q. Liu, S. Wang, Time-and-concept enhanced deep multidimensional item response theory for interpretable knowledge tracing, *Knowl.-Based Syst.* 218 (2021) 106819.
- [14] S. Pandey, G. Karypis, A self-attentive model for knowledge tracing, 2019, arXiv preprint arXiv:1907.06837.
- [15] A. Ghosh, N. Heffernan, A.S. Lan, Context-aware attentive knowledge tracing, in: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 2330–2339.
- [16] S. Pu, M. Yudelson, L. Ou, Y. Huang, Deep knowledge tracing with transformers, in: *International Conference on Artificial Intelligence in Education*, Springer, 2020, pp. 252–256.
- [17] Y. Ren, K. Liang, Y. Shang, Y. Zhang, Muloer-san: 2-layer multi-objective framework for exercise recommendation with self-attention networks, *Knowl.-Based Syst.* 260 (2023) 110117.
- [18] Q. Liu, Z. Huang, Y. Yin, E. Chen, H. Xiong, Y. Su, G. Hu, Ekt: Exercise-aware knowledge tracing for student performance prediction, *IEEE Trans. Knowl. Data Eng.* 33 (1) (2019) 100–115.
- [19] L. He, J. Tang, X. Li, P. Wang, F. Chen, T. Wang, Multi-type factors representation learning for deep learning-based knowledge tracing, *World Wide Web* 25 (3) (2022) 1343–1372.
- [20] W. Tan, Y. Jin, M. Liu, H. Zhang, Bidkt: Deep knowledge tracing with bert, in: *International Conference on Ad Hoc Networks*, *International Conference on Testbeds and Research Infrastructures*, Springer, 2022, pp. 260–278.
- [21] P. Brown, H. Roediger, M. McDaniel, M.I. Stick, *The Science of Successful Learning*, Cambridge, MA, 2014.
- [22] R.A. Bjork, Retrieval practice and the maintenance of knowledge, *Pract. Aspects Mem.: Curr. Res. Issues* 1 (1988) 396–401.
- [23] C.A. Rowland, E.L. DeLosh, Mnemonic benefits of retrieval practice at short retention intervals, *Memory* 23 (3) (2015) 403–419.
- [24] R.C. Atkinson, R.M. Shiffrin, Human memory: A proposed system and its control processes, in: *Psychology of Learning and Motivation*, Vol. 2, Elsevier, 1968, pp. 89–195.
- [25] H.L. Roediger III, A.C. Butler, The critical role of retrieval practice in long-term retention, *Trends Cogn. Sci.* 15 (1) (2011) 20–27.
- [26] K.B. Lyle, C.R. Bego, R.F. Hopkins, J.L. Hieb, P.A. Ralston, How the amount and spacing of retrieval practice affect the short-and long-term retention of mathematics knowledge, *Educ. Psychol. Rev.* 32 (1) (2020) 277–295.
- [27] A. Baddeley, Working memory: Theories, models, and controversies, *Annu. Rev. Psychol.* 63 (2012) 1–29.
- [28] G. Abdelrahman, Q. Wang, Deep graph memory networks for forgetting-robust knowledge tracing, *IEEE Trans. Knowl. Data Eng.* (2022).
- [29] S. Shen, E. Chen, Q. Liu, Z. Huang, W. Huang, Y. Yin, Y. Su, S. Wang, Monitoring student progress for learning process-consistent knowledge tracing, *IEEE Trans. Knowl. Data Eng.* (2022).
- [30] T. Gervet, K. Koedinger, J. Schneider, T. Mitchell, et al., When is deep learning the best approach to knowledge tracing? *J. Educ. Data Min.* 12 (3) (2020) 31–54.
- [31] G. Abdelrahman, Q. Wang, Knowledge tracing with sequential key-value memory networks, in: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 175–184.
- [32] X. Guo, Z. Huang, J. Gao, M. Shang, M. Shu, J. Sun, Enhancing knowledge tracing via adversarial training, in: *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 367–375.
- [33] C. Wang, W. Ma, M. Zhang, C. Lv, F. Wan, H. Lin, T. Tang, Y. Liu, S. Ma, Temporal cross-effects in knowledge tracing, in: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2021, pp. 517–525.
- [34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.
- [35] S. Shen, Q. Liu, E. Chen, H. Wu, Z. Huang, W. Zhao, Y. Su, H. Ma, S. Wang, Convolutional knowledge tracing: Modeling individualization in student learning process, in: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 1857–1860.

- [36] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S.G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, et al., Hybrid computing using a neural network with dynamic external memory, *Nature* 538 (7626) (2016) 471–476.
- [37] B. Xu, Z. Huang, J. Liu, S. Shen, Q. Liu, E. Chen, J. Wu, S. Wang, Learning behavior-oriented knowledge tracing, in: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 2789–2800.
- [38] M. Chen, Q. Guan, Y. He, Z. He, L. Fang, W. Luo, Knowledge tracing model with learning and forgetting behavior, in: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 3863–3867.
- [39] X. Zhang, F. Wei, M. Zhou, Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization, in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5059–5069.
- [40] A. Nakata, Work. hours, sleep. sufficiency, And prevalence of depression among full-time employees: a community-based cross-sectional study, *J. Clin. Psychiatry* 72 (5) (2011).
- [41] P.K. Agarwal, J.R. Finley, N.S. Rose, H.L. Roediger III, Benefits from retrieval practice are greater for students with lower working memory capacity, *Memory* 25 (6) (2017) 764–771.
- [42] C. Donkin, R.M. Nosofsky, A power-law model of psychological memory strength in short-and long-term recognition, *Psychol. Sci.* 23 (6) (2012) 625–634.
- [43] L. Averell, A. Heathcote, The form of the forgetting curve and the fate of memories, *J. Math. Psychol.* 55 (1) (2011) 25–35.
- [44] J.T. Wixted, E.B. Ebbesen, On the form of forgetting, *Psychol. Sci.* 2 (6) (1991) 409–415.
- [45] H. Ebbinghaus, Memory: A contribution to experimental psychology, *Ann. Neurosci.* 20 (4) (2013) 155.
- [46] J.M. Murre, J. Dros, Replication and analysis of ebbinghaus' forgetting curve, *PLoS One* 10 (7) (2015) e0120644.
- [47] D. Maraun, H. Rust, J. Timmer, Tempting long-memory-on the interpretation of dfa results, *Nonlinear Process. Geophys.* 11 (4) (2004) 495–503.
- [48] Y. Choi, Y. Lee, D. Shin, J. Cho, S. Park, S. Lee, J. Baek, C. Bae, B. Kim, J. Heo, Ednet: A large-scale hierarchical dataset in education, in: *International Conference on Artificial Intelligence in Education*, Springer, 2020, pp. 69–73.
- [49] S. Pandey, J. Srivastava, Rkt: relation-aware self-attention for knowledge tracing, in: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 1205–1214.
- [50] Y. Su, Q. Liu, Q. Liu, Z. Huang, Y. Yin, E. Chen, C. Ding, S. Wei, G. Hu, Exercise-enhanced sequential modeling for student performance prediction, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32, 2018.
- [51] M.M. Khajah, Y. Huang, J.P. González-Brenes, M.C. Mozer, P. Brusilovsky, Integrating knowledge tracing and item response theory: A tale of two frameworks, in: *CEUR Workshop Proceedings*, Vol. 1181, University of Pittsburgh, 2014, pp. 7–15.
- [52] Y. Liu, M. Lapata, Hierarchical transformers for multi-document summarization, in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5070–5081.
- [53] M. Riesenhuber, T. Poggio, Hierarchical models of object recognition in cortex, *Nature Neurosci.* 2 (11) (1999) 1019–1025.
- [54] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, 2016.
- [55] Z. Pan, B. Zhuang, J. Liu, H. He, J. Cai, Scalable vision transformers with hierarchical pooling, in: *Proceedings of the IEEE/Cvf International Conference on Computer Vision*, 2021, pp. 377–386.
- [56] L. Van der Maaten, G. Hinton, Visualizing data using t-sne, *J. Mach. Learn. Res.* 9 (11) (2008).