HiTSKT: A Hierarchical Transformer Model for Session-Aware Knowledge Tracing

Fucai Ke[a], Weiqing Wang[a], Weicong Tan[a], Lan Du[a], Yuan Jin[a], Yujin Huang[a], Hongzhi Yin[b]

[a] *Faculty of Information Technology, Monash University, Melbourne, 3800, VIC, Australia*
[b] *Faculty of Engineering, The University of Queensland, Brisbane, 4072, QLD, Australia*

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. As an important way of providing personalized experience for online education, KT has gained increased attention in recent years. 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 summarise acquired knowledge across the past sessions. To predict an interaction in the current session, a knowledge retriever integrates the summarised 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 three public datasets demonstrate that HiTSKT achieves new state-of-the-art performance on all the datasets compared with six state-of-the-art KT models.

Keywords: User behaviour modelling, Knowledge tracing, Educational data mining, Learner modeling, Hierarchical Transformer
PACS: 0000, 1111
2000 MSC: 0000, 1111

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, 2, 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 predefined skills is essential in providing personalized experience in online education and has attracted increased attention these years [4, 5]. Based on the traced students' knowledge, E-educators can properly understand the dynamics of students' knowledge states and provide personalized learning curricula accordingly. Intelligent tutoring systems (ITS), which aim to automate the task of personalized curricula recommendation, have been endowed with the same tracing ability as human educators thanks to the advancements of the knowledge tracing (KT) techniques in machine learning. More specifically, an ITS leverages such techniques to model and infer students' mastery levels of skills over time from their historical responses to exercise questions, based on which their corresponding future performance can be accurately predicted.

Despite the notable achievements, knowledge tracing 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 knowledge tracing 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, 7, 8]). Despite the efficacy and semantics brought to model learning, these assumptions, however, have also limited the models' applicability in reality as they are mostly too simplified to capture enough complexity of the dynamics (e.g., Gaussian prior assumptions on student expertise and question difficulty are mostly simplified treatments which ignore the actual statistics of the two).

Recent KT models take advantage of the expressive power of deep neural networks (DNNs), such as the RNN-based [9, 10, 11, 12, 13] and the self-attention [14, 15, 16, 17] models, to overcome the modelling and inference limitations of the traditional models like HMMs. More specifically, RNN-based

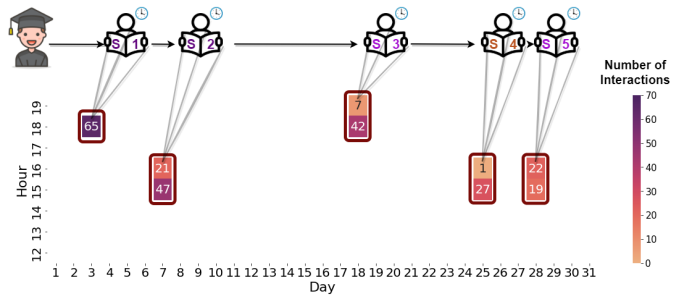

Figure 1: Interactions heatmap for one student in one month on ASSISTments2017. Each block in the red frame is a practice session containing several interactions, and time intervals are represented as arrow. One interaction means the process that a student interacts with ITS and provides one response to one exercise question. For instance, the example student has 65 interactions at $6pm$ on the third day of that month. These 65 interactions are relative dense and can be considered as a session practice. The following session practice happened on three days later, which contains 68 interactions during 2 hours on the seventh day of that month.

models predict each interaction based on their predecessors, where more recent ones have larger impacts on the current interaction. On the other hand, self-attention-based KT models condition the prediction for each interaction on all the other historical interactions by attending to them all at once. It has been shown by the previous research that these recent DNN-based models can bring substantial performance improvements to the knowledge tracing task in terms of students' response accuracy prediction and question recommendation [9, 10, 18, 19, 20].

Despite the improvements brought by the DNNs, current KT models still have several known issues. One is that they model a student's historical learning records as a continuing sequence, which fails to capture how these records are distributed during the students' learning progress in reality. As an example, we show in Fig. 1 our observations via exploration on ASSISTments2017[1], a real-world knowledge tracing dataset. This figure illustrates a student's learning history within 31 days, which has 5 distinct practice sessions, each containing a different number of questions answered consecutively within relatively short periods of time; while the intervals between any two consecutive practice sessions are much longer. More specifically, with a spectrum of colours that indicates the intensity levels of interactions per time unit, a session is characterized by a distinct burst of interactions with the ITS, followed by a distinct large time gap (presented as directed arrows in the figure). Such sessional behaviour of a single student repeats across the entire student population, the statistical details of which will be further introduced in Section 5 to confirm the generality of our findings. Therefore, instead of being continuing, histories of students' interaction records in reality usually exhibit clustering effects by consisting of a series of separated practice sessions where each session includes a sequence of practice interactions.

In Educational Psychology, studies of 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, 22, 23, 24]. According to these studies, within-session information, such as number and difficulty of the questions, and inter-session information, such as the spacing of different sessions, have different impacts on students' learning performance[25, 26, 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 **Hi**erarchical **T**ransformer model for **S**ession-Aware **K**nowledge **T**racing (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 receives the intra-session representations of all the past sessions, and consolidates them 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 the inter-session knowledge gets consolidated, HiTSKT uses a *Retrieval & Responding* (RR) modelling component to retrieve the stored inter-session knowledge representation, and integrates it with the intra-session knowledge acquired so far in the session to compute the student's current knowledge state.

The contributions of our work are summarised 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 three 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 model and a knowledge state retrieval encoder, the model is capable of capturing students' knowledge state variations within and across the sessions, as well as their impacts on students' performance in the current session.
- HiTSKT also models and captures students' sessional forgetting behaviour, which is evidenced by our exploratory data analysis in Section 5.6, through an power-law-decay scaled attention mechanism, designed and deployed in the acquisition & consolidation modelling component.
- Extensive experiments including overall effectiveness studies, ablation studies and model interpretation have been

[1] ASSISTments2017 source: https://sites.google.com/site/assistmentsdata

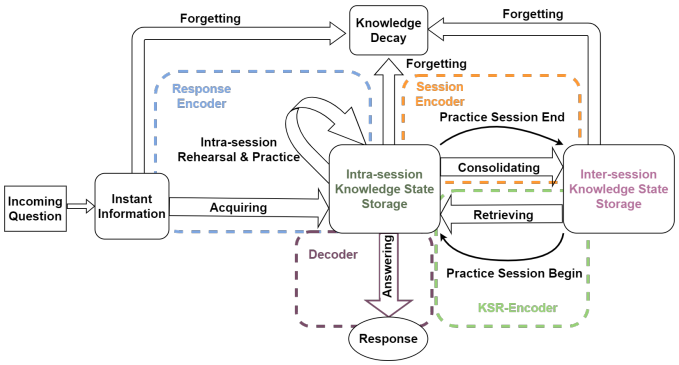

Figure 2: HiTSKT motivation based on Atkinson Shiffrin Memory Model. When students launch a session, their brain retrieves related knowledge states from their deep memory storage as the working states, which can also be viewed as intra-session knowledge states in this figure, and are ready to respond. 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].

conducted on three real-world online educational datasets. The results show that (1) the superior response correctness prediction performance of HiTSKT by capturing the sessional information, compared with six state-of-the-art KT models; (2) the effectiveness of the major components of HiTSKT. In addition, the code of HiTSKT and the implementation of several state-of-art KT models is provided[2].

The rest of the paper is organised 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 six state-of-the-art KT models over three 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 Knowledge Tracing 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 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].

[2]https://github.com/pokerme7777/HiTSKT

Deep Learning and Other Knowledge Tracing Methods. To address the lacks of model flexibility and prediction power encountered by the traditional KT models, Piech et al. [9] proposed the first KT model based on deep neural networks (i.e., the DKT model) with long-short term memory recurrent neural networks (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 an 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 knowledge tracing 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]. More recently, Ghosh et al. [15] proposed a context-aware KT model with a monotonically and exponentially decay attention mechanism to model how the student's performance from a distant past affects their current performance.

Although self-attentive KT models [18, 20, 32, 35] have shown desirable tracing performance, their designs overlook sessional information and are too rigid to incorporate such information to capture the potential sessional drifts of knowledge states. Furthermore, there have also been some models designed to capture the memory-decay dynamics underlying the variations of knowledge states caused by the students' forgetting behaviour. For example, AKT [15] has proposed a context-aware exponentially decay attention mechanism and HMN [12] uses Differentiable Neural Computer (DNC) [36] to improve the modeling of memory decay. However, they fail to consider the memory decay to be dependent 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 practice sessions can better reflect their knowledge mastery levels than their performance in a single question. They have also found that 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 as one single exercise can not make a

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
$\boldsymbol{x}$	Embedding of the interaction x
$\boldsymbol{d}_q$	Embedding of the difficulty of question q
$\boldsymbol{k}$	Embedding of the skill k
$\boldsymbol{f}_q$	Embedding of the number of occurrences of question q
$\boldsymbol{a}$	Embedding of the correctness of a response
$\boldsymbol{h}^{\text{Inner}}$	Intra-session knowledge representation vector
$\boldsymbol{h}^{\text{Inter}}$	Inter-session knowledge representation vector
$\boldsymbol{q}$	Attention query vector
$\boldsymbol{K}$	Key matrix
$\boldsymbol{V}$	Value matrix
$\boldsymbol{W}$	Mapping matrix

significant difference in 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 modeling 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, which 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 knowledge tracing. Recently, to adapt the transformer model to more complex tasks, there has emerged research work (e.g., [37]) that hierarchically stacks up transformer encoders to extract individual representations from partial sequences of the entire long sequence. Inspired by these studies, in this paper, we propose a KT model with hierarchical transformer encoders to fully exploit the sessional information for better modelling the variations of students' knowledge states.

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 non-overlapping sessions $\{ses_1^i, ses_2^i, \ldots, ses_n^i\}$, where ses_n^i denotes the student i's n-th session. For ses_n^i, it contains a sequence of t interactions $\boldsymbol{x}_n^i = \{x_{n,1}^i, x_{n,2}^i, \ldots, x_{n,t}^i\}$, where an interaction $x_{n,t}^i = \{q_{n,t}^i, k_{n,t}^i, f_{n,q_t}^i, a_{n,t}^i\}$ 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,q_t}^i, and (4) the graded response $a_{n,t}^i$. Note that the graded response $a_{n,t}^i \in \{1, 0\}$ denotes whether the student i has answered the t-th question in the n-th session correctly or not. In this case, given the student i's past learning history, which includes his/her first n sessions, i.e., $\{ses_1^i, ses_2^i, \ldots, ses_n^i\}$, the first $t-1$ interactions in the current $(n+1)$-th session, i.e., $\{x_{n+1,1}^i, x_{n+1,2}^i, \ldots, x_{n+1,t-1}^i\}$, and a query question $q_{n+1,t}^i$ at the current time t, the problem of knowledge tracing concerns the prediction of the graded answer $a_{n+1,t}^i$ of student i to the query question. Fig. 3 illustrates our targeting KT problem formulated above where, as its distinctive feature, the differences in sessions have been highlighted with dashed frames.

In this paper, the time duration we use to define a session is 10 hours based on human activity studies [3]. 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. It suggests that when students begin to practice or rehearse, their brains will first retrieve relevant knowledge from the current inter-session knowledge state storage into their intra-session knowledge state storage. The intra-session knowledge will then be integrated with the new coming information, acquired within the session thus far, as the students' current knowledge states to perform the responses. When the session ends, its final intra-session knowledge will be consolidated back (with certain extents of forgetting) into the students' inter-session storage.

Therefore, HiTSKT consists of two main components: the acquisition & consolidation (AC) modelling component, which is a session-aware hierarchical transformer encoder model, and the retrieval & responding (RR) modelling component with a knowledge state retrieval (KSR) encoder module and a student response prediction module. The structure of HiTSKT is shown in Fig. 4.

4.1. Acquisition & Consolidation Modelling Component

The AC component aims to extract and aggregate the sessional information in a bottom-up manner. More specifically,

[3] People working hours per day [38]

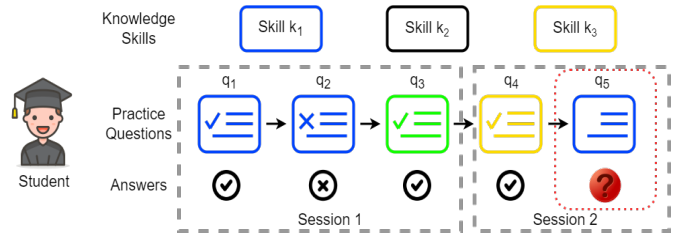

Figure 3: An illustrative example of the proposed HiTSKT Task. Given a student's past learning history including (1) first n sessions i.e., $\{ses_1^i, ses_2^i, \ldots, ses_n^i\}$ and (2) the first $t-1$ interactions in the current $(n+1)$-th session, i.e., $\{x_{n+1,1}^i, x_{n+1,2}^i, \ldots, x_{n+1,t-1}^i\}$, HiTSKT calculates out the student's current knowledge state first. The following step is to predict student's graded answer $a_{n+1,t}^i$ to the query question $q_{n+1,t}^i$ based on the knowledge state.

an interaction(-level) encoder first computes the knowledge representations that summarize the acquired intra-session knowledge for each individual session from a student's learning history. Correspondingly, 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. Then, these representations (of the AKSS tokens output from the interaction encoder) are fed into an upper-level session encoder. It proceeds to summarise all the past sessions experienced by a student into an inter-session knowledge representation, which gets consolidated into an inteR-session Knowledge State Storage (RKSS).

4.1.1. Interaction encoder

This encoder aims to accurately capture the variations of students' knowledge states within each past session, and summarise them into the corresponding within-session knowledge representations. It comprises a rehearsal embedding layer, a multi-head attention layer and a feed forward neural network layer.

Rehearsal embedding serves as the input to the interaction encoder, which should be able to capture as diverse information regarding students' individual interactions as possible. In this work, we consider such information to come from the following aspects; the skill/concept covered by a question, which can be indicative of the knowledge state correlations among questions of the same concept; the difficulty of a question, indicative of the variations of students' knowledge states, e.g. being low for difficult questions; the number of occurrences of a question, reflecting the correlations between rehearsals/practice and knowledge states; lastly, the correctness of past responses, indicative of a student's learning quality. In particular, the last two together can also be indicative of the student's knowledge state variation. For example, according to [26, 39], it is very likely for students with high knowledge states in certain domains to correctly answer the domain questions in few times of practice. On the other hand, if a student still has low correctness on those questions after several attempts, his/her knowledge state in the particular area is more likely to be low.

Therefore, we design the rehearsal embedding for the t-th interaction of the n-th session $\boldsymbol{x}_{n,t}$ to be the sum of the embeddings of (1) the main skill $\boldsymbol{k}_{n,t} \in \mathbb{R}^d$ of the rehearsal question q_t, (2)

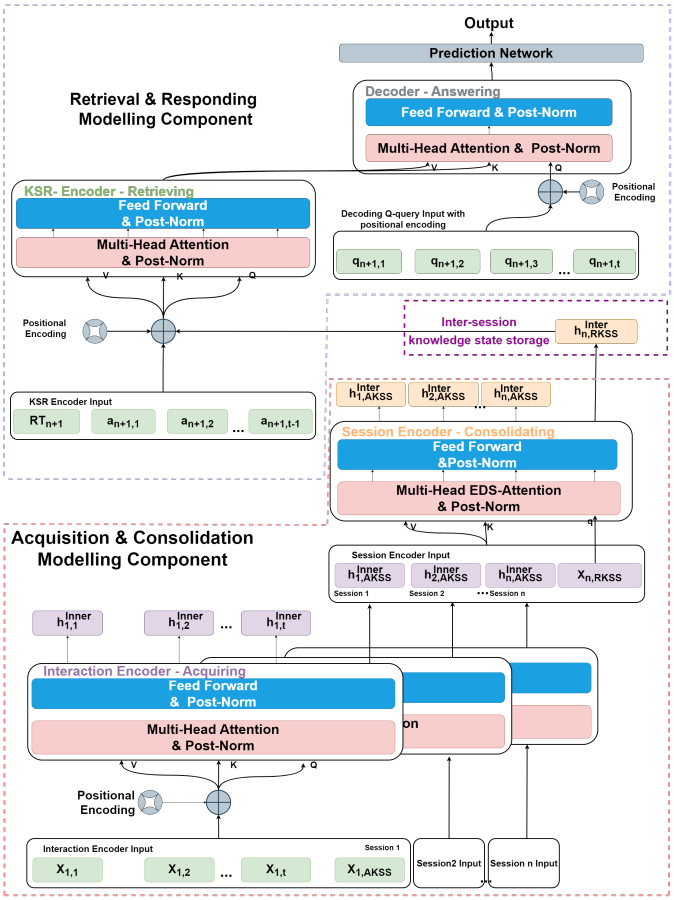

Figure 4: The overall model architecture of HiTSKT, which consists of two components: AC Component and RR Component. The AC component aims to extract and aggregate the sessional information in a bottom-up manner. Specifically, an interaction(-level) encoder first computes the knowledge representations (i.e., $\boldsymbol{h}_{n,\mathrm{AKSS}}^{\mathrm{Inner}}$) that summarize the acquired intra-session knowledge for each session. These representations are fed into an upper-level session encoder to summarise all the past sessions experienced by a student into an inter-session knowledge representation (i.e., $\boldsymbol{h}_{n,RKSS}^{\mathrm{Inter}}$).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.

the question's difficulty $\boldsymbol{d}_{n,q_t} \in \mathbb{R}^d$, (3) the ordinal label for the occurrence number of the question $\boldsymbol{f}_{n,q_t} \in \mathbb{R}^d$ and (4) the student's response correctness $\boldsymbol{a}_{n,t} \in \mathbb{R}^d$; That is the following formulation of the rehearsal embedding $x_{n,t}$:

$$\boldsymbol{x}_{n,t} = \boldsymbol{k}_{n,t} \oplus \boldsymbol{d}_{n,q_t} \oplus \boldsymbol{f}_{n,q_t} \oplus \boldsymbol{a}_{n,t} \tag{1}$$

In Section 5, we present our experimental result that the rehearsal embedding is better at modeling the KT problem, comparing with the Rasch model based embedding [15]. Furthermore, knowledge tracing 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, and add the corresponding position embedding to the rehearsal embedding as the final input to the interaction encoder.

Multi-head attention layer serves as the aggregator of the knowledge acquired within one session by performing a weighted sum over the contextualized knowledge (state) representations of each interaction into a within-session knowledge representation. To facilitate the within-session knowledge aggregation, we append a special token AKSS at the end[4] 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. In this case, the query to the attention layer is mapped from the rehearsal embedding $\boldsymbol{x}_{n,\text{AKSS}} \in \mathbb{R}^d$ of the special token, while 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

$$\boldsymbol{q}_{n,\text{AKSS}} = \boldsymbol{W}_Q^T \boldsymbol{x}_{n,\text{AKSS}};\ \boldsymbol{K}_n = \boldsymbol{W}_K^T \boldsymbol{X}_n;\ \boldsymbol{V}_n = \boldsymbol{W}_V^T \boldsymbol{X}_n \tag{2}$$

where $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d \times d_m}$ are learnable mapping matrices, with d_m being the dimension after mapping, and $\boldsymbol{X}_n^T \in \mathbb{R}^{t \times d}$ is the input embedding matrix of all the interactions within the n-th session. Then, the scaled dot-product attention scores $\boldsymbol{\alpha}_{n,\text{AKSS}} \in \mathbb{R}^t$ are computed as:

$$\boldsymbol{\alpha}_{n,\text{AKSS}} = \text{Softmax}\left(\frac{\boldsymbol{K}_n^T \boldsymbol{q}_{n,\text{AKSS}}}{\sqrt{d_{\boldsymbol{K}_n^T}}}\right) \tag{3}$$

where $d_{\boldsymbol{K}_n^T} = d_m$ is the dimension of the key embedding matrix. After the Softmax function is applied to obtain $\boldsymbol{\alpha}_{n,\text{AKSS}}$, the inner-session knowledge representation $\boldsymbol{h}_{n,\text{AKSS}}^{\text{Inner}} \in \mathbb{R}^{d_m}$ is output as:

$$\boldsymbol{h}_{n,\text{AKSS}}^{\text{Inner}} = \sum_t \alpha_{n,t,\text{AKSS}} \times \boldsymbol{v}_{n,t} \tag{4}$$

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

Feed-Forward neural network (FFN) module then takes in the output $\boldsymbol{h}_{n,\text{AKSS}}^{\text{Inner}}$ from the attention layer, passes it through a two-layer multi-layer perceptron with ReLU activation to obtain the following final output of the interaction encoder:

$$\boldsymbol{h}_{n,\text{AKSS}}^{\text{Inner}} := \boldsymbol{\Phi}_2^T \sigma(\boldsymbol{\Phi}_1^T \boldsymbol{h}_{n,\text{AKSS}}^{\text{Inner}} + \boldsymbol{b}_1) + \boldsymbol{b}_2 \tag{5}$$

where $\boldsymbol{\Phi}_1 \in \mathbb{R}^{d_m \times d'_m}, \boldsymbol{\Phi}_2 \in \mathbb{R}^{d'_m \times d_m}$ and $\boldsymbol{b}_1 \in \mathbb{R}^{d'_m}, \boldsymbol{b}_2 \in \mathbb{R}^{d_m}$ 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 $\{ses_1^i, ses_2^i, \ldots, ses_n^i\}$ will be encoded by the interaction encoder into a sequence of intra-session knowledge (state) representations $\{\boldsymbol{h}_{1,\text{AKSS}}^{\text{Inner}}, \boldsymbol{h}_{2,\text{AKSS}}^{\text{Inner}}, \ldots, \boldsymbol{h}_{n,\text{AKSS}}^{\text{Inner}}\}$. These representations will then be used to construct a historical intra-session knowledge representation matrix $\boldsymbol{H}_n^T \in \mathbb{R}^{n \times d}$ and be fed into the subsequent session encoder.

[4]This placement is commonly performed by the neural language models for NLP tasks.

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, via the memory consolidation and forgetting mechanism, thereby their long-term learning performance, which is dependent on the retrieved knowledge across sessions [21, 22, 25, 26, 23, 24]. Therefore, we propose to design a session-level encoder that is able to model the sessional consolidation and forgetting mechanism of students' memory by analysing their acquired within-session knowledge during their learning processes.

Based on the memory studies mentioned above, the consolidation process mainly occurs after a "practice & rehearsal" session, which summarises and aggregates all the acquired knowledge from the session, and the knowledge state, to be consolidated with such knowledge, is highly dependent on the memory retention rate during the consolidation. As a result, we model the sessional knowledge "consolidation" by leveraging the attention mechanism, and the forgetting behaviour by explicitly modifying the attention to decay further back in time. According to the findings from the memory studies, forgetting curves are closely approximated by power-law curves [40, 41, 42]. 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 corresponds to the inter-session knowledge (state) representation $\boldsymbol{h}_{n,\text{RKSS}}^{\text{Inter}}$ that aggregates inner-session knowledge representations $\{\boldsymbol{h}_{1,\text{AKSS}}^{\text{Inner}}, \boldsymbol{h}_{2,\text{AKSS}}^{\text{Inner}}, \ldots, \boldsymbol{h}_{n,\text{AKSS}}^{\text{Inner}}\}$ across all the previous sessions. The power-law-decay function that 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:

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

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:

$$\begin{aligned} \boldsymbol{q}_{n,\text{RKSS}} = \boldsymbol{W}_Q^{'T} \boldsymbol{x}_{n,\text{RKSS}};\ \boldsymbol{K}_n' = \boldsymbol{W}_K^{'T} \boldsymbol{H}_n;\ \boldsymbol{V}_n' = \boldsymbol{W}_V^{'T} \boldsymbol{H}_n \\ \boldsymbol{\alpha}_{n,\text{RKSS}} = \text{Softmax}\left(\frac{\boldsymbol{K}_n^{'T} \boldsymbol{q}_{n,\text{RKSS}}}{\sqrt{d_{\boldsymbol{K}_n^{'T}}}}\right) \\ \boldsymbol{h}_{n,RKSS}^{\text{Inter}} = \sum_{n'=1}^{n} \alpha_{n,n',\text{RKSS}} \times \xi_{n+1,n'} \times \boldsymbol{v}'_{n'} \end{aligned} \tag{7}$$

where $\boldsymbol{W}_Q', \boldsymbol{W}_K', \boldsymbol{W}_V' \in \mathbb{R}^{d \times d_m}$ are learnable mapping matrices of the session-level attention; $\boldsymbol{x}_{n,\text{RKSS}}$ is the rehearsal embed-

ding of the RKSS token at the end of session n, which "stores" the inter-session knowledge before the start of session $n+1$; $\boldsymbol{v}'_{n'} \in \mathbb{R}^{d_m}$ is the n'-th column of the value (embedding) matrix $\boldsymbol{V}'_n \in \mathbb{R}^{d_m \times n}$. Finally, the acquisition & consolidation modelling component outputs $\boldsymbol{h}^{\text{Inter}}_{n,RKSS}$, an inter-session knowledge representation for all the previous practice sessions.

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, which 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

The KSR encoder models the human memory retrieval process, in which the representation of a student's inter-session knowledge state is retrieved, and then refined by each and every previous response in the current $(n+1)$-th session, i.e., $\{a^i_{n+1,0}, a^i_{n+1,1}, a^i_{n+1,2}, \ldots, a^i_{n+1,t-1}\}$, before fed into the decoder module to be used to predict the corresponding next answers $\{a^i_{n+1,1}, a^i_{n+1,2}, \ldots, a^i_{n+1,t}\}$. In this case, $a^i_{n+1,0}$ is artificially padded as it does not correspond to the correctness of any question 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 $a^i_{n+1,1}$. 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 $\boldsymbol{h}^{\text{Inter}}_{n,RKSS}$ at every position before the current one is its element-wise addition with the student's response correctness embedding $\boldsymbol{a}_{n,t}$ at those positions as well as 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 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, where the output representations of the KSR encoder from every position up to the current one consist of the key and value embedding matrices, and the aggregated skill and difficulty embedding of the current question q_t (i.e., $\boldsymbol{k}_{n+1,t} \oplus \boldsymbol{d}_{n+1,q_t}$) 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. $\boldsymbol{h}^{\text{Answer}}_{n+1,t} \in \mathbb{R}^{d_m}$), which 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 minimised:

$$\mathcal{L} = -\sum_i \sum_n \sum_t (a^i_{n,t} \log(\hat{a}^i_{n,t})) + ((1 - a^i_{n,t}) \log(1 - \hat{a}^i_{n,t})) \tag{8}$$

Table 2: Datasets statistics

Features/Datasets	ASSISTments2017	Junyi	EdNet
Interactions	942,807	14,660,217	88,597,714
Students	1,709	29,865	131,538
Questions	3,162	25,630	13,523
Skills	102	1,326	10,000
Avg.Sessions	8	20	23
Avg.Interactions/Session	70	24	28
Median.Interactions/Session	54	16	16

5. Experiments

In this section, we will introduce the experiments conducted to evaluate our proposed model. We train and test our model on three benchmark datasets and compare the performance of our model with state-of-the-art knowledge tracing models. Note that, the code of our model and also the implementation of the compared models are publicly available[5].

5.1. Datasets

Three real-world benchmark KT datasets with time stamp information (i.e., ASSISTments2017 [6], Junyi [7] and Riiid's EdNet dataset [43]) are used to evaluate the performance of HiTSKT. 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 three datasets, we remove all the interactions where skill information is Null and drop all the interactions where the time spent is more than 9999 seconds as these interactions might be not answered or completed. The number of occurrences of each question are also labelled by simply counting on all three 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 hours are separated as two different sessions based on human activity studies [8].

[5] https://github.com/pokerme7777/HiTSKT

[6] ASSISTments2017 source: https://sites.google.com/site/assistmentsdata

[7] Junyi source: https://www.kaggle.com/junyiacademy

[8] People working hours per day [38]

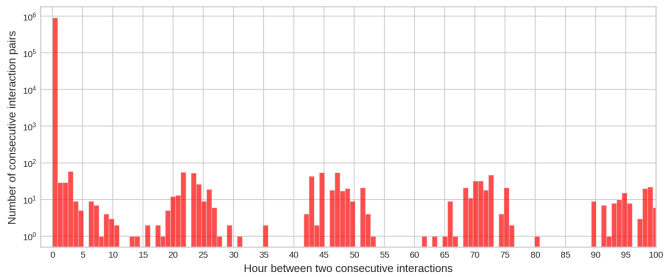

Figure 5: Histogram of time intervals between two consecutive interactions on ASSISTments2017 dataset. The x-axis means the time gap between two consecutive interactions and the y-axis means the number of interaction pairs. The time intervals between two consecutive interactions follow a long-tail distribution which means that most consecutive interactions happen within a short time interval while much fewer of them have a long time interval. Therefore, we could conclude that histories of students' interaction records in reality usually exhibit clustering effects by consisting of a series of separated practice sessions where each session includes a sequence of practice interactions.

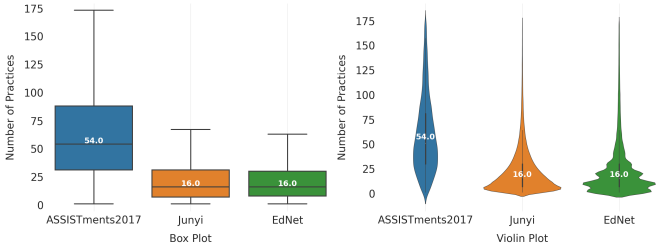

Figure 6: Box-plot and Violin-plot for the number of interactions in each session on three datasets. The distribution of the number of practice over sessions on ASSISTments2017 is close to the normal distribution, which means that the number of interactions for most sessions is around the mean number. On the contrary, the distributions are skewed to less than the mean numbers on both Junyi and EdNet datasets, which means that the session sizes are smaller and the students' histories records are sparser on these two datasets.

Datasets Statistics. The detailed statistics of these datasets after preprocessing are shown in Table 2, where Avg.Sessions is 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 the 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 hours 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 which means that most consecutive interactions happen within a short time interval while much fewer of them have a large time interval. This together with Fig. 1 consolidate the universality and naturality of session information in KT task.

Compared with the other two 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.

Session Sequence Length and Practice 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 ASSISTments2017, Junyi and EdNet. The number of interactions in each session is 64 on ASSISTments2017 dataset, and 32 on Junyi and EdNet datasets.

For longer interaction or session sequences, we trim the earlier interactions or sessions. For shorter sequences, padding is used to fill the vacancy. For instance, on ASSISTments2017, 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 while "session padding" is used to fill the blank if a student has less than 16 sessions. Similarly, the most recent 64 interactions in each session is used as the interaction sequences to model the intra-session information and interaction sequences with insufficient interactions will be filled up by "interaction padding".

5.2. Baselines and Evaluation Metric

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

- DKT [9] is the first knowledge tracing 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.

Table 3: The performance of HiTSKT and all the baselines on three datasets.

	ASSISTments2017	Junyi	EdNet
DKVMN	0.6695 ± 0.0032	0.7264 ± 0.0018	0.5967 ± 0.0007
DKT	0.6888 ± 0.0096	0.7386 ± 0.0131	0.6209 ± 0.0004
HawkesKT	0.6762 ± 0.0104	0.7296 ± 0.0124	0.6880 ± 0.0035
SAKT	0.7187 ± 0.0024	0.7852 ± 0.0014	0.7513 ± 0.0004
ATKT	0.7417 ± 0.0030	0.7850 ± 0.0038	0.6935 ± 0.0038
AKT	0.7392 ± 0.0019	0.7850 ± 0.0026	0.7464 ± 0.0037
HiTSKT	**0.7553 ± 0.0004**	**0.7911 ± 0.0001**	**0.7615 ± 0.0003**

- SAKT [14] is the first model utilising 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 perturbance to improve model robustness.

The metric used to evaluate the model performance in this paper is the widely-used area under the ROC (Receiver Operating Characteristics) which is known as AUC (Area Under the Curve) [10, 14, 15, 18, 30, 32, 33, 44].

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.

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 three 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 the 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 three datasets and the following observations are made:

- HiTSKT outperforms all the compared state-of-the-art models on all three datasets (i.e., 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 ASSISTments2017 dataset which has the least 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., 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 to confuse the neural network during training which makes the model more robust on small datasets.

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 4 lists the results of ablation studies.

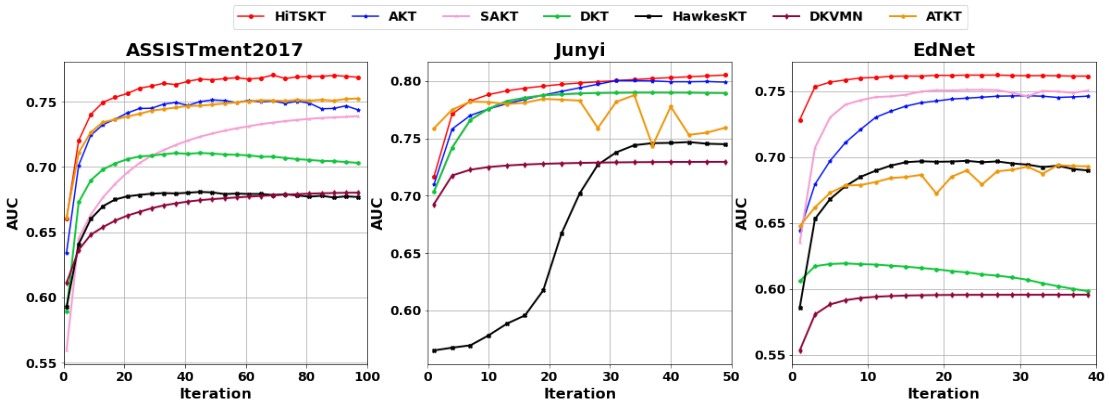

Figure 7: Convergence Curves by Validation AUC. Two self-attention baseline models (i.e., AKT, SAKT) perform better on larger datasets when compared with other baseline models. Different from the other attention based models, ATKT performs better on smaller datasets (e.g., ASSISTments2017) compared with its performance on larger datasets (e.g., EdNet). HiTSKT is robust and converges quickly than other compared models.

Table 4: HiTSKT Performance Table - Ablation Study

	ASSISTments2017	Junyi	EdNet
HiTSKT using Rasch Model-Based Embeddings [15]	0.7417	0.7857	0.7569
HiTSKT replaces KSR encoder with right-shift answer	0.7520	0.7879	0.7588
HiTSKT without AKSS	0.7536	0.7897	0.7520
HiTSKT without AKSS and RKSS	0.7433	0.7774	0.7417
HiTSKT without Positional Encoding	0.7552	0.7904	0.7599
HiTSKT using monotonic Attention	0.7390	0.7841	0.7331
HiTSKT	0.7553	0.7911	0.7615

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 [45]. 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 three 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.

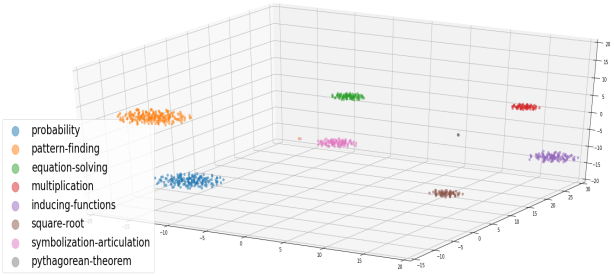

Figure 8: Question Embedding Visualisation. All questions' difficulty are presented as dot points in three-dimensional space, and the colours represent the corresponding skills. All questions with the same skill are clustered together in space, which means questions with the same skill will be projected to close value in the embedding space.

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. In this experiment, we use Average-Pooling (Eq.9) to replace the AKSS and RKSS tokens. The κ is the kernel size and the stride is the size of the window.

$$\text{out}(d_1, d_2) = \frac{1}{\kappa} \sum_{m=0}^{\kappa-1} \text{input}(\text{stride} \times d_1 + m, d_2) \tag{9}$$

The AUC of the model without AKSS reduced by around 0.3% compared with HiTSKT, and a further 0.6% fall can be observed if HiTSKT does not have both artificial tokens on three datasets. Theoretically, average pooling can capture information from all the new representations of $\boldsymbol{h}_{n,t}^{\text{Inner}}$ or all the session histories representations $\boldsymbol{z}_{n,t}$. However, according to our exper-

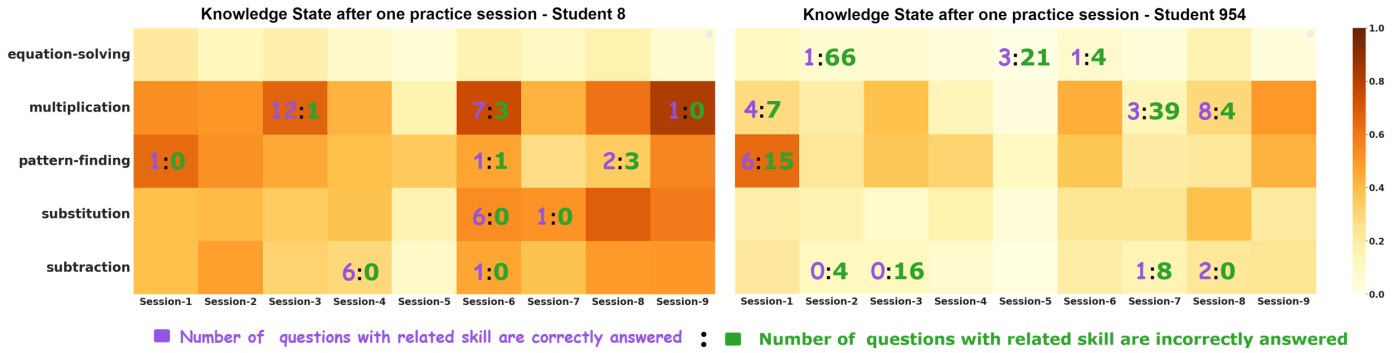

Figure 9: Knowledge State Visualisation. This is an exploration figure about two students' predicted knowledge state on five skills (e.g., multiplication, equation-solving, pattern-finding, subtraction and substitution). Each block colour represents the predicted knowledge state (the weighted arithmetic mean of students' probabilities of correctly answering related questions) of corresponding skills after a practice session. The numbers represent the number of questions with related skills that are correctly/incorrectly answered in reality.

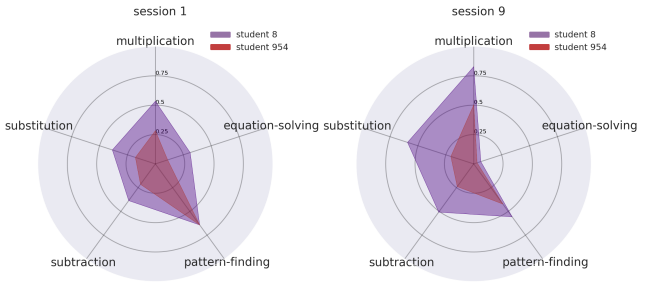

Figure 10: Radar plots for example students' Knowledge State on five skills. Two demonstrated students' skills have been improved or forgotten after nine practice sessions and students' learning rates are different.

iment 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 part, we compare this attention mechanism with standard scaled dot-product attention. The performance dropped to 73.90%, 78.41% and 73.31% respectively on ASSISTments2017, Junyi and EdNet after we replaced the power-law decay attention with the standard one.

KSR encoder. When we replaced the KSR encoder with right shift answer embedding, the models' performance dropped to 75.20%, 78.79% and 75.88% on ASSISTments2017, Junyi and EdNet respectively.

5.6. Further Model interpretation through Visualisation

In this session, we will interpret HiTSKT by visualising questions' difficulty $d_{n,t}$ embedding and exploring example students' knowledge state using ASSISTments2017.

Visualising Questions' Difficulty Embedding. We first select eight high-frequency skills, as shown in Fig. 8 legend, and get all corresponding questions' difficulty embedding in ASSISTments2017. For these questions' difficulty, we can get their 256-dimensional embeddings after training the model. Then t-SNE [46] is used to reduce the dimension to 3 to better visualise them. In Fig. 8, all questions' difficulty 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 questions' difficulty into embedding space which will enable a more effective calculation of students' knowledge state.

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 select two example students (i.e., student 8 and student 954) and explore their predicted knowledge state on five skills (e.g., multiplication, equation-solving, pattern-finding, subtraction and substitution) as a demonstration. For the two students, we calculate the weighted arithmetic mean of their 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. 9. 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. 9 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. 9, two observations can be made.

First, forgetting and memory re-consolidation are existing phenomena in knowledge tracing 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 $12/(12+1)$. 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., $7/(7+3)$ compared with $12/(12+1)$) 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. 10. 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 visualisation 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, we focus on exploiting the session information from students' learning histories for tackling the knowledge tracing problem, as inspired by the detailed exploratory data analysis of session information over several real-world knowledge tracing datasets. To do so, we have introduced a carefully designed hierarchical transformer model for session-aware knowledge tracing. Our proposed model, HiTSKT, consists of two main components, which are the *acquisition* & *consolidation* modelling component with a hierarchical transformer encoder architecture to summarise two types of acquired knowledge: the (lower-level) intra-session and (higher-level) inter-session knowledge, and the *retrieval* & *responding* modelling component with 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 three large-scale real-world knowledge tracing 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. Visualisation 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 modeling, 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 knowledge tracing.

References

[1] Q. Liu, S. Shen, Z. Huang, E. Chen, Y. Zheng, A survey of knowledge tracing, arXiv preprint arXiv:2105.15106 (2021).

[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 transactions on knowledge and data engineering 21 (6) (2008) 814–828.

[3] E. J. Emanuel, Moocs taken by educated few, Nature 503 (7476) (2013) 342–342.

[4] X. Song, J. Li, T. Cai, S. Yang, T. Yang, C. Liu, A survey on deep learning based knowledge tracing, Knowledge-Based Systems 258 (2022) 110036.

[5] M. Khajah, R. V. Lindsey, M. C. Mozer, How deep is knowledge tracing?., International Educational Data Mining Society (2016).

[6] A. T. Corbett, J. R. Anderson, Knowledge tracing: Modeling the acquisition of procedural knowledge, User modeling and user-adapted interaction 4 (4) (1994) 253–278.

[7] R. S. d Baker, A. T. Corbett, V. Aleven, 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, Advances in Neural Information Processing Systems 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 Systems with Applications 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, Knowledge-Based Systems 218 (2021) 106819.

[14] P. Pandey, G. Karypis, A self attentive model for knowledge tracin, in: Proceedings of the 12th International Conference on Educational Data Mining, 2019.

[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, Knowledge-Based Systems 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 Transactions on Knowledge and Data Engineering 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, Practical aspects of memory: Current research and 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 in cognitive sciences 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, Educational Psychology Review 32 (1) (2020) 277–295.
[27] A. Baddeley, Working memory: Theories, models, and controversies, Annual review of psychology 63 (2012) 1–29.
[28] G. Abdelrahman, Q. Wang, Deep graph memory networks for forgetting-robust knowledge tracing, IEEE Transactions on Knowledge and Data Engineering (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 Transactions on Knowledge and Data Engineering (2022).
[30] T. Gervet, K. Koedinger, J. Schneider, T. Mitchell, et al., When is deep learning the best approach to knowledge tracing?, Journal of Educational Data Mining 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, . 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] 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.
[38] A. Nakata, Work hours, sleep sufficiency, and prevalence of depression among full-time employees: a community-based cross-sectional study, The Journal of clinical psychiatry 72 (5) (2011) 0–0.
[39] 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.
[40] C. Donkin, R. M. Nosofsky, A power-law model of psychological memory strength in short-and long-term recognition, Psychological Science 23 (6) (2012) 625–634.
[41] L. Averell, A. Heathcote, The form of the forgetting curve and the fate of memories, Journal of mathematical psychology 55 (1) (2011) 25–35.
[42] J. T. Wixted, E. B. Ebbesen, On the form of forgetting, Psychological science 2 (6) (1991) 409–415.
[43] 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.
[44] 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.
[45] 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.
[46] L. Van der Maaten, G. Hinton, Visualizing data using t-sne., Journal of machine learning research 9 (11) (2008).