scientific reports

Check for updates

OPEN

Deep learning based knowledge tracing in intelligent tutoring systems

Xin Zhou[1,5], Zhuoxu Zhang[2,5], Xike Xie[3] & Jiawei Zhang[4]✉

The emergence of online education, e.g., intelligent tutoring system (ITS), complements or partially replaces conventional offline education, especially during the COVID-19 pandemic. Knowledge tracing (KT) plays a pivotal role in the intelligent tutoring system in capturing the knowledge states of students. By analyzing a series of students' interaction records of questions and answers in ITS, KT is able to provide personalized feedbacks to students. Recent advances in deep learning techniques, such as deep knowledge tracing, apply recurrent neural networks over students' interaction records for knowledge state modeling and achieve great improvement in the prediction of performance on future tasks and assessment questions. However, in practice, KT is often in lack of sufficient student interaction records to accurately model and predict students' knowledge states, the so-called data sparsity issue. Meanwhile, the data sparsity issue is generally overlooked in the existing knowledge tracing systems. In this paper, we propose a quality-aware deep learning framework for knowledge tracing, based on the sparse attention techniques and generative decoding. Extensive experiments are conducted over a series of real datasets showing that our proposal accurately captures students' knowledge states.

Keywords Knowledge tracing, Deep learning, Sparse attention, Generative decoder

Over the past decades, e-learning systems, including intelligent tutoring systems (ITS) and massive open online courses (MOOCs), have played an important role in online education systems[1].

Recent multidisciplinary analyses of ITS evolution[2] reveal increasing convergence between educational technologies and recommendation systems, particularly in handling sparse learner interactions.

With the sweep of COVID-19, students around the world are going through a migration from offline to online education[3]. Undoubtedly, online learning provides us with many advantages, such as massive learning resources, an unrestricted learning environment, and fast and efficient communication platforms. However, there are also challenges for us to handle, such as the difficulty in tailoring teaching programs according to the student's needs and in assessing students' mastery of knowledge.

This complexity mirrors challenges in scientific article recommendation systems[4], where sparse user interactions require sophisticated modeling approaches.

As pointed out in the OECD report, low quality, diversity, and availability of teaching materials online, as well as the lack of pedagogical continuity, particularly for the most disadvantaged students, risks undermining learning during the COVID-19 period[5]. In such circumstances, tracking the knowledge state of students to monitor and evaluate the impact of online learning, identify gaps and collect disaggregated data is considered important initiative[6]. For example, Riiid labs[7] have provided a series of solutions in the field of online education, including the deep learning-based knowledge tracing.

Existing works of knowledge tracing can be roughly classified into two categories, modeling the state of students' knowledge mastery and predicting students' performance in future questions.

Recent advances in interpretable educational AI[8] have shed light on how deep knowledge tracing (DKT) models make predictions, while graph-based approaches like jumping knowledge networks[9] demonstrate potential for capturing hierarchical skill relationships.

For the former, Bayesian knowledge tracking (BKT)[10] uses binary variables to characterize the students' mastery of knowledge concept, and deep knowledge tracking (DKT)[11] uses hidden variables to indicate the

[1]State University of New York at Binghamton, Binghamton, New York, USA. [2]The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China. [3]University of Science and Technology of China, Hefei, China. [4]Soochow University, Suzhou, China. [5]Xin Zhou, Zhuoxu Zhang have authors contributed equally to this work. ✉email: jwzhang@suda.edu.cn

students' mastery levels. For the latter, conventional strategies present students' priori knowledge with additional storage units[12] or embed the input knowledge concepts into hidden units[13].

These methods parallel developments in recommendation system architectures[14,15], where matrix factorization and neural approaches address similar sparse interaction challenges.

Given recent advances, the practical deployment of knowledge tracing still suffers. The key issue is data deficiency or sparsity[16], meaning that there often lacks of sufficient interaction records for accurately modeling knowledge states and predicting students' performance in addressing future questions. Of student-system interaction records, a question takes several skills as prerequisites, establishing profound connections with knowledge states. On the other hand, students' knowledge states, as well as corresponding migration processes, are latent and are well interpretable. Thus, without sufficient interaction records (i.e., training data), data-driven deep learning techniques would fall short in capturing the knowledge states and predicting students' performance, the so-called data sparsity[17] or "cold start" challenge[18,19]. The problem of data sparsity is well recognized[20]. According to[21], students may occasionally log in the system and selectively conduct online exercises that partially cover the knowledge mastery of course contents without formal supervision. Moreover, the fact is prevalent and may exist for long, due to:

1) The unbalanced popularization and distribution of ITS, as observed in graph-based recommendation systems[22] handling uneven node distributions
2) The subordinate role of ITS in an official education system[23,24];
3) The lack of consensus and guidelines in online-offline collaborative education[25].

This mirrors challenges in session-based recommendations[26] where temporal learning patterns require adaptive modeling.

Thus, it is important to technically study the knowledge tracing under the scenario of data deficiency and sparsity, for the practical system deployment.

Motivated by the challenges above, in this study, we propose a novel DKT framework integrated in an ITS.

Drawing inspiration from dynamic graph attention networks[27] that adapt to edge relationships,

we devise a new sparse attention mechanism to tackle the problem of data sparsity, and then apply a heuristic generative decoding method to predict students' performance in unknown problems, which simulates the learning process in realistic scenarios. To evaluate our proposals, we conduct extensive experiments with public datasets. From our empirical studies, we find that our proposed sparse attention process can deal with sparse data and improve the performance of knowledge tracing.

The rest of the paper is organized as follows. In Section "Related works" , we give an overview of existing KT methods as well as some widely used KT architectures. Section "DKT framework" proposes a DKT framework in the context of an ITS and discusses the details about the module deployment. Section "Methodology" makes a comprehensive explanation of our proposed MSKT model. Section "Experimental analysis" evaluates the performance of MSKT deployed in an ITS. Finally, Section "Conclusion" concludes the paper.

Related works

ITS has been an active research field since its early work in 1970s. Its goal is to provide students with personalized guidance. As the exploration of Artificial Intelligence and Data Mining, mainstream techniques in these fields have found their way to support ITS, specially for the problem of Knowledge Tracing[28]. KT problem is defined to figure out how to trace the learning state of students in an effective manner with interactions with online teaching materials or exercises, e.g., the mastery level of skills required needed by specific course contents. The term of KT was first proposed by[29] for intelligent teaching and cognitive study. Since then, lots of researches have been carried out to design machine learning models to solve the challenges in KT. Generally, there are mainly two categories of methods. One is the structures based on recurrent neural networks (RNN)[30]. The other is based on attention mechanisms. Both of them have make contributions to knowledge state modeling and student future performance prediction.

Before the rise of deep learning, there were plenty of researches for knowledge tracing, such as BKT[10,31,32]. BKT utilizes a probabilistic graphical structure, i.e., Hidden Markov Model[33] and Bayesian Belief Network[34] to capture the knowledge states of students with their interactions in online learning. But BKT assumes that student's current knowledge state is only related to the previous state and will not be forgotten after learning a certain knowledge concept. Hence, it is not sufficient to trace the knowledge state by solely using unit variables. On the other hand, DKT[11] adopted deep learning in knowledge tracing. It replaces univariate units with multidimensional hidden units which eliminate the need for experts to manually provide labelled concepts. Additionally, DKT is able to manipulate vectorized data, which is suitable for online education environments. However, DKT is a data-driven model and is not applicable to context-specific teaching. Based on this,[35] construct a *key* matrix and a *value* matrix respectively to store knowledge concepts and update knowledge states at the expense of a higher storage cost. This dynamic key-value memory network captures not only the knowledge state, but also the potential relationships between concepts. Nevertheless, none of these networks can solve the "cold start" problem (i.e., the inability to predict students' answers to questions that do not appear previously) until[18] proposed the exercise-enhanced recurrent neural network and Exercise-aware Knowledge Tracing (EKT). Both structures represent query content from a semantic perspective using a bidirectional Long short-term memory (LSTM). EKT has its advantages in updating the knowledge state in real time and handling "cold start" problems with good interpretability. Whereafter,[36] proposed a neural cognitive diagnosis framework. It embeds students and questions to different dimensions and uses neural networks to learn the corresponding interaction functions.

Recently, a set of novel approaches focus on attention mechanisms which generally have a better interpretability. For example, an attention network with bidirectional LSTM was used in[37] to implement real-time user Q&A prediction. A Self-Attentive model for Knowledge Tracing (SAKT) was proposed in[38], which achieved state-of-the-art results with only question and answer interaction tuples as the input of the model. In[7], it extracts suitable key-valued inputs using LSTM and solves the problem of inter container leakage when multiple problems share a common id.[39–41] made improvements in the feature inputs by adding relevant attributes other than the answers to the questions, such as the category of the questions, the answering time consumption, etc.

The above frameworks have a good performance in the basic task of knowledge tracing. Besides, there are also strategies, which simulate realistic learning scenarios heuristically. For example, kernel functions were used in[42,43] to model the forgetting process of students, i.e., the correlation between two interactions sequence can be expressed as the function of their distance. In[44], the correlation between questions is also integrated to trace the forgetting process more accurately.[45] used a novel inference-generating and knowledge-tracing framework that construct multi-dimensional relationship from problems to concepts. Also, to maintain the retention of information in the original data, the framework studied a influence factor map, which represents potential relationship from questions to concepts. In[46], a self-attentive fusion approach were used and achieved a comparative performance. In this work, we upgrade the self-attention mechanism to the sparse attention mechanism with a focus on tackling the data sparsity challenge. We also devise a generative decoding method for quality student performance prediction.

DKT framework

In this section, we first explain how our proposed DKT framework fits into the modern ITS. Then, we give detailed descriptions about the specific modules in our DKT.

Intelligent tutoring system

Figure 1 illustrates a typical ITS with KT. In general, ITS is mainly comprised several parts: Online Testing System, Knowledge Tracing System, Learning Situation Analysis System, and other upper layered applications. In the sequel, we make an explanation about the interactions in-between the three key parts, with a specific focus on the challenges arising from the ITS to motivate the necessity of studying DKT.

Online testing system

Online Testing System works as a platform to interact with students. In particular, the Question Bank (represented by a cylinder in Fig. 1) generates a series of questions to students. Accordingly, students work out the questions and submit their answers to the system. We denote this interactive process as a Q&A Session, which is integrated in the Online Testing System and aims to elaborate the questions given to students. Suppose there is a set of question $\{q_1, q_2, q_3, q_4\}$ taken from the Question Bank. From the students' side, to correctly answer these questions, some prerequisite skills $\{k_1, k_2, k_3\}$ are needed (e.g., a question about quadratic equation may have prerequisites of factorization and polynomials). Formally, $q_1 = k_1 \wedge k_2$ denotes that question q_1(e.g., quadratic equation) takes skill k_1(e.g., factorization) and skill k_2(e.g., polynomials) as its prerequisites. In a Q&A Session, questions are generated by different combination of skills and dispensed to students, in order to test the performance of KT and the improvement of students based on the guidance of KT. After finishing the above process, various information are produced and will be recorded to the Knowledge Tracing System for further analysis.

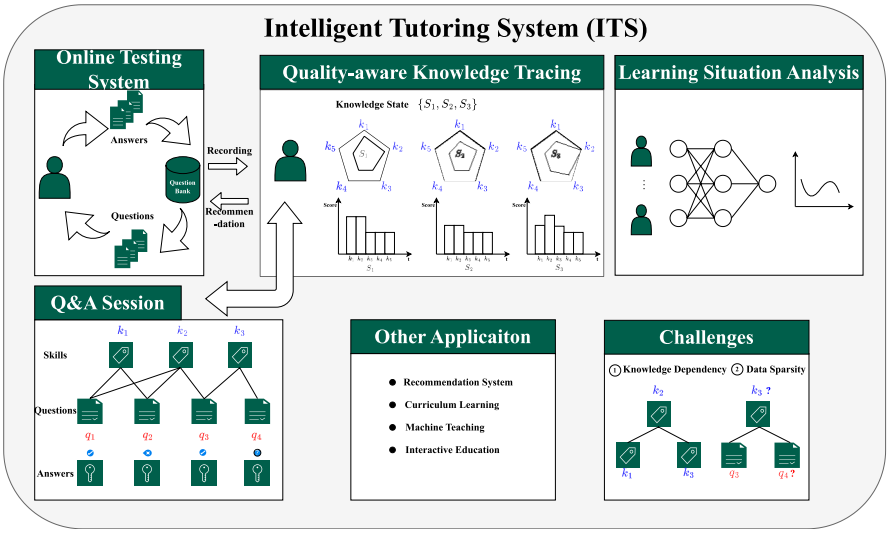

Figure 1. Intelligent tutoring system(ITS).

Knowledge tracing
Based on the interaction records in the Online Learning System, the goal of KT is to extract the knowledge state of each student. For example, in the KT diagram of Fig. 1, each knowledge state in $\{S_1, S_2, S_3\}$ represents the mastery of skills $\{k_1, k_2, k_3, k_4, k_5\}$ of a student. Specifically, proficiency in skill k_2 leads to a higher score in the knowledge state related to k_2. The accurate capturing of knowledge states facilitates the application in Learning Situation Analysis and so on. Particularly, our proposed method enables quality knowledge tracing which yields a better performance, even the given interaction records from Online Testing System are sparse, i.e., the quality of training data is not high. Details about Quality-aware KT can be found in Section "Methodology".

Learning situation analysis
Upper streaming applications, such as Learning Situation Analysis, are also supported by KT, the core part of an ITS. It takes knowledge states from Quality-aware KT as input and then aggregates them for learning situation analysis. Unlike the knowledge state of a single person extracted in KT, Learning Situation is designed to integrate the states of all students and produce an overall situation for the student groups, such as elementary students, secondary school students, and college students. The granularity of student groups is application dependent and can be specified by analysts. In general, knowledge states can be fed into an average pool to produce representative learning situations of student groups. Also, we can put knowledge states into deep learning networks for such aggregation.

Other applications
The knowledge state extracted in Knowledge Tracing can be further applied to many other fields such as recommendation system, curriculum learning, machine teaching, and interactive education. In recommendation system, knowledge state can be used to recommend relevant learning materials by a recommendation model[47]. In order to get the maximum knowledge gain for students, personalized curriculum learning can be designed[48]. Besides, with the usage of knowledge state, machine teaching can train up the teacher model together with the student model, to minimize the teaching cost[12]. Last but not the least, "knowledge state" has been utilized[49] in education game or the so-called interactive education, where players' knowledge states play an important role in the training process.

Challenges
Although ITS has its applications in many areas, there are also quite a few challenges to be considered. First and foremost, as discussed in Section "Online testing system", one question usually takes several skills as its prerequisite. Also, each skill might depend on more than one other skills, as shown in the Challenges diagram of Fig. 1. Thus, the complex knowledge dependency raise challenges to knowledge state tracing to a great extent. Moreover, data collected from Online Learning System is often sparse. This is because (i) the number of interactions in the historical Q&A Session is limited; (ii) for each question, there is a number of relevant skills, but not all skills are required for every attempt at the question, so that the storage matrix is of sparsity (e.g., the traditional one-hot matrix). It is well accepted that the sparsity of data may reduce the quality of prediction and increase the unnecessary memory overhead; (iii) the proportion of online and offline teaching varies significantly in different places. For example, in developed regions online teaching accounts for a large proportion, while in underdeveloped regions offline teaching is more favorable. As a result, the diverse combination of online and offline teaching may lead to data sparsity. Targeted at these challenges, we apply a specific DKT model to solve the difficulties, as discussed in Section "Module deployment".

Module deployment
In Section "Intelligent tutoring system", we propose our DKT framework from the system aspect of an ITS. Here, we offer further illustration of the DKT mechanism, including its deployed modules. The overview of the DKT framework is shown in Fig. 2, which consists of three important modules, evaluation module, MSKT module, and the cognition module.

Evaluation module
As shown in Fig. 2, the Evaluation Module plays a role of the interface between Online Testing System to Knowledge State Extraction module (MSKT). A large number of students log in the Online Testing System and participate in the Q&A Section for data collection. After students finishing all the requested questions, the Evaluation Module records and collects data in the Online Testing System and then outputs a chunk of vectorized log messages, e.g., question sequence, question categories, student's response sequence, and corresponding timestamps. It is worth noting that some of the raw log messages will be processed (e.g., data embedding) before being stored. All the processed information will be collected and fed to the next module.

MSKT module
In this paper, we proposed a module named MSKT (Multi-head Sparse attention for Knowledge Tracing) to implement the knowledge state extraction of DKT. MSKT takes its input from the output of Evaluation Module. Then these sequence data will be processed by a series of self-attention network[50] in MSKT. After processing, MSKT outputs the corresponding knowledge states, which can be used to make the prediction for future answer accuracy, or be put into the Cognition Module for advanced analysis. More details about MSKT will be discussed in Section "Methodology".

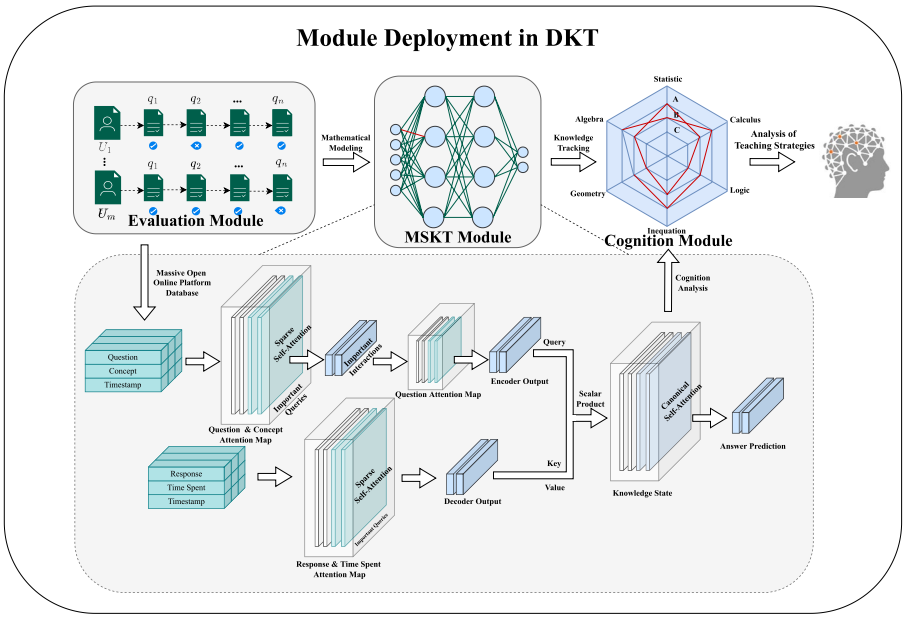

Figure 2. DKT module.

Cognition module

As the subsequent step of MSKT, Cognition Module plays the role of an output unit of our DKT framework. From Fig. 2, we know that in MSKT, the product of the output of encoder and decoder is considered as knowledge states, which are the input of Cognition Module. Essentially, the knowledge state is presented as a multi-dimension vector. The length of the vector is the same as the output of encoder or decoder of MSKT, due to the property of vector multiplication. Knowledge state takes its advantages in the rich presentation of students' learning state. For example, the value of the i-th dimension in the knowledge state may show the mastery of Geometry in maths, while the j-th dimension denotes the ability in Algebra. As a result, the ample knowledge states can be utilized for learning situation analysis or teaching strategy supporting. For instance, a low value in the dimension of Statistics implies that the student needs to improve the mastery of Statistics. Hence, more exercises about Statistics are expected to be given to the student. In addition, with the input of knowledge states, it makes the Cognition Module easier to be adapted to other application-driven tasks. As mentioned in[51], knowledge states are capable of combining techniques as reinforcement learning and computer vision. In a way, we can connect Cognition Module to other deep learning modules to accomplish these tasks as aforementioned.

Methodology

In this section, we introduce the method of Multi-head Sparse attention for Knowledge Tracing (MSKT). We first make a description about the architecture of MSKT and compare with the two state-of-arts, SAKT[38] and SAINT[39,40]. After that, we make analysis for each component of MSKT to cover the details about the design mechanism.

Hereby, we make a comprehensive explanation of the architecture of MSKT. First, we list the input and output of MSKT, then we describe the structure MSKT in details. Meanwhile, we make a comparison between MSKT and the two state-of-art self-attention structure to further show the merits of MSKT.

Model input and output

According to Section "Evaluation module", MSKT gets its vectorized input from the output of the Evaluation Module. Here, we list some of the inputs which are taken by MSKT. Without losing generality, we assume that the length of data sequence is n. The length of the sequence n can be set to an integer multiple of RAM, so that the in-memory computation can be done efficiently.

- Question sequence $E = (q_1, q_2, ..., q_n)$: q_i denotes the ID of the i-th question.
- Question category $C = (c_1, c_2, ..., c_n)$.
- Student's response $R = (r_1, r_2, ..., r_n)$: $r_i \in \{0, 1\}$ is a binary value, such that it is equal to 0 if the i-th response is wrong and vice versa.
- Timestamp sequence $T = (t_1, t_2, ..., t_n)$: t_i denotes the timestamp when question q_i is answered.
- Position sequence $P = (p_1, p_2, ..., p_n)$: p_i refers to position of question q_i and its corresponding response, e.g., 1st, 2nd, ...
- Interaction sequence $I = (i_1, i_2, ..., i_n)$: it is the summarization (e.g., concatenation or summation) of the above inputs for the internal use of neural networks.

From the above inputs, we choose question sequence, question category, student's response, and timestamp sequence, as the input of MSKT. The output is two-fold, including the knowledge states, represented by the product of the encoder and decoder (as mentioned in Section "Cognition module"), and the output of the final fully connected layer (depicted in Fig. 3), predicting students' performance in addressing future questions.

Model structure

Figure 3 displays the structure of MSKT. It mainly contains two parts, the "sparse attention" encoder and the "generative" decoder. The "sparse attention" is designed specially for the sparse data, which improves the efficiency of the sparse data computation. The "generative" decoder is used to enhance the prediction accuracy. In this way, the structure of MSKT is able to increase both the efficiency and effectiveness. Both of them adopt the classical self-attention mechanism[52].

A self-attention function is defined as the mapping from a query and a set of key-value pairs to the output. Intuitively, one compares the query with keys and retrieves the weights of the values. Particularly for the self-attention function, the output is the aggregation of weights of retrieved values, a.k.a., the summation of weights. Given a set of sequence data, we select the corresponding query, key and value sequence from the input. Then, we do the following to get the output: (i) conduct an dot-product of the query with all keys; (ii) derive the weight of value sequence by applying a softmax function; (iii) do computation on the value and its corresponding weight to get the output. The output value indicates how much importance a query is. A higher output value reflects a more important status of the single query in the whole sequence. The above process can be summarized as the following formula[50].

$$SelfAttention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}}\right)\boldsymbol{V} \tag{1}$$

$\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ represent the queries, keys, and values matrices, respectively, and d denotes the dimension of $\boldsymbol{Q}$ and $\boldsymbol{K}$, which serves to scale the dot-product attention, thus stabilizing gradient magnitudes throughout the training process. In the encoder, the self-attention mechanism transforms three distinct input sequences into these matrices: E, C, and P. Specifically, E (element-wise embeddings) is designated as the query $\boldsymbol{Q}$, capturing token-level semantic features, while C (contextual embeddings) acts as $\boldsymbol{K}$ to capture inter-token relationships and structural dependencies. Additionally, P (positional encodings) is assigned to $\boldsymbol{V}$, thereby preserving sequential order and positional awareness. This combination enables the encoder to compute comprehensive attention scores that effectively integrate semantic relevance, syntactic structures, and positional cues within the input sequences.

In contrast, in the decoder, the attention mechanism is adapted specifically for autoregressive generation tasks. Here, the decoder utilizes sequence R, representing its current state (e.g., embeddings of previously generated tokens), as the queries $\boldsymbol{Q}$. The sequence T, encoding target-aware contextual features commonly derived from encoder outputs or prior temporal steps, serves as the keys $\boldsymbol{K}$. Meanwhile, positional encodings P continue to function as the values $\boldsymbol{V}$ to maintain spatial and sequential coherence and facilitate accurate positional awareness in autoregressive decoding.

The attention mechanism computes scores via scaled dot products between the matrices $\boldsymbol{Q}$ and $\boldsymbol{K}$, normalized by $\sqrt{d}$ to mitigate gradient instability issues, ensuring effective backpropagation through deeper layers. Consequently, the resulting attention scores are utilized to dynamically weigh and aggregate the values

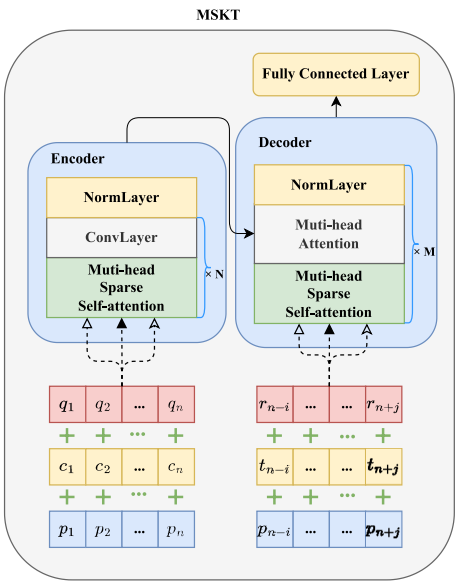

Figure 3. Structure of MSKT.

V, synthesizing comprehensive output representations. This adaptive mechanism effectively balances positional awareness, spatial coherence, and contextually driven relationships across tokens, allowing the model to generate output representations that optimally capture the sequential, spatial, and structural information necessary for downstream tasks. Thus, this attention-based paradigm elegantly accommodates both the semantic coherence required for accurate encoding and the autoregressive predictive capability necessary for precise decoding.

Instead of using a single self-attention function, here we apply multi-head self attention function. The term multi-head refers to the mechanism that queries, keys, and values are mapped multiple times with varied projections. With this design, we can get multi-dimensional output values which enriches the subspace representation and thus improves the performance.

Comparison with SAKT and SAINT. Fig. 4 compares the structures of SAKT (left) and SAINT (right). From the picture, we know that the attention layer of SAKT is shallow and it does not consider appropriate features such as categories, positions and timestamps. It is hard for the simple encoder structure of SAKT to extract deep conceptual information and to handle knowledge state modeling. Compared to SAKT, the structure of SAINT is refined. Firstly, SAINT enriches its input by adding categories, position and timestamps sequence. Secondly, SAINT adopts an encoder-decoder model rather than the single encoder of SKAT. These two improvements brings the benefit of fully exploiting the deep conceptual information.

In contrast with the two state-of-art models, MSKT employs multi-head sparse self-attention layer to deal with the input sequences and it is equipped with a generative sparse decoder. The MSKT architecture we proposed aims to mainly address the two shortcomings of previous models: (i) Reduce the cost of sparse matrix multiplication (details will be discussed Section "Encoder and decoder") (ii) prevents step-by-step prediction and supports alternative prediction. Next, we explain how MSKT can refine the above deficiency.

Encoder and decoder

We study two important components, the "sparse attention" encoder and the "generative" decoder, for our proposed MSKT model.

Sparse attention encoder. To illustrate the mechanism of the sparse attention encoder, we first explain the concept of the attention map. We derive the attention map by the dot production of $\boldsymbol{Q}$ and $\boldsymbol{K}$ matrix. The value of the result matrix represents the attention score of the two members. For example, Fig. 5 depicts the attention map of the ten questions (q_0 to q_9), sampled from our dataset. The first column in Fig. 5 represents the attention score of q_0 to all others queries from q_0 to q_9. A darker color indicates a higher attention score and therefore a closer relation between the two queries, and vice versa. Thus, we call the query with low attention score with others the *lazy query*. On the contrary, queries with high attention score with other counterparts are *active queries*. From the chart, we learn that q_1 is an active query while q_6 is a lazy query. In fact, the attention score of most query pairs is low. As a result, if the lazy queries can be omitted, we can focus more on the important queries with an easier and faster computation.

Based on the above discussion, we apply the sparse self-attention encoder, as depicted in Fig. 6. Compared to the baseline, our encoder have two improvements, including (i) the sparse attention method which accelerates the attention layer computation, and (ii) integrating convolution and pooling layers between each n-head layer as depicted in Fig. 6 to prevent excessive memory consumption due to layer stacking, as shown in Fig. 6. These two improvement can achieve a better performance in both efficiency and accuracy.

Generative decoder. Fig. 7 describes our generative decoder, in comparison to the step-by-step decoder applied in SAINT. The traditional step-by-step decoder needs to "look forward" many steps, starting from

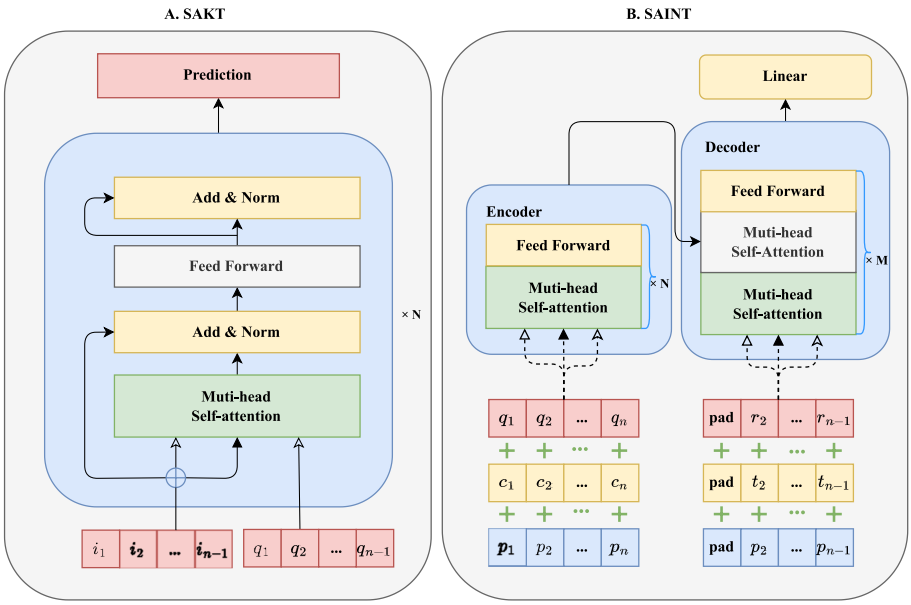

Figure 4. Two state-of-art structures.

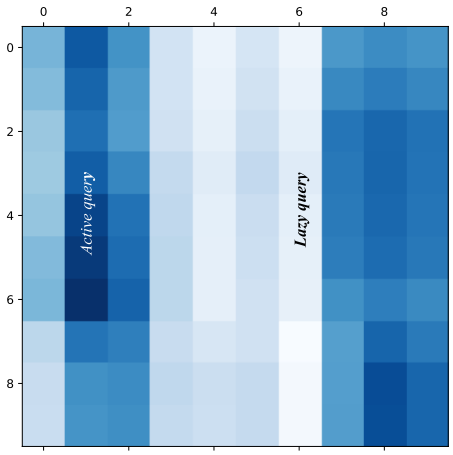

Figure 5. Attention map.

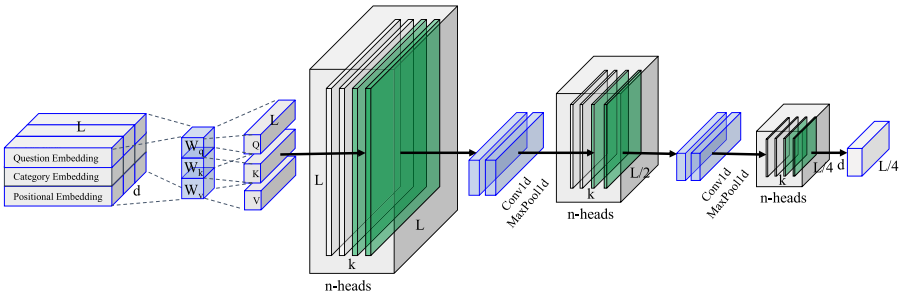

Figure 6. Single layer sparse attention encoder.

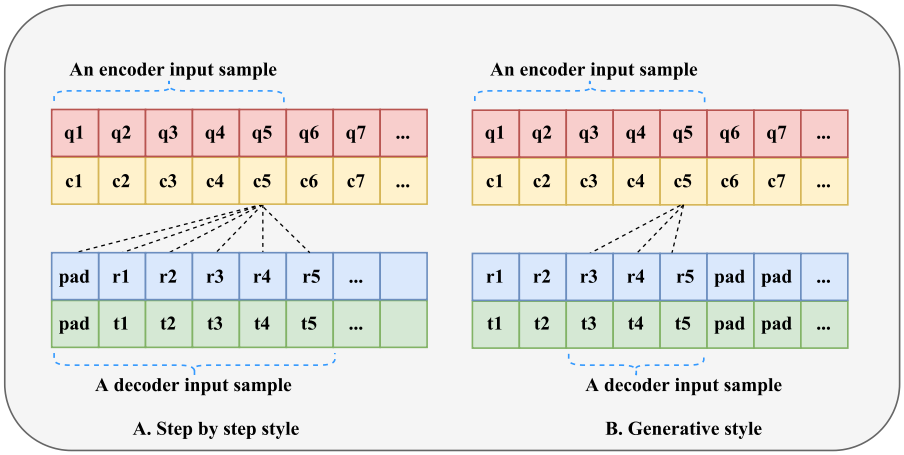

Figure 7. Two different decoding strategies (**A**) Decoding style of step-by-step (**B**) Generative decoding style.

a specific flag. In contrast, the generative decoder only needs to "look forward" n steps where n is a tunable parameter. Take Fig. 7 as an example. In order to make prediction of answer to q_6, the step-by-step decoder "look forward" five steps (q_1 is the start flag), while the generative style only needs to "look forward" three steps ($n = 3$). It is obvious that the generative decoder can make prediction in a more efficient and controllable way, because it is capable in skipping unnecessary input samples, so as to accelerate the model training.

In this section, we investigate the MSKT model and study its key components. In the sequel, we conduct detailed empirical studies to evaluate the model performance.

In summary, MSKT refines the structure of SAKT and SAINT with sparse attention and generative decoder. These improvements not only gain the computing efficiency but also increase the effectiveness compared to the state-of-art methods.

Experimental analysis

To evaluate the performance of our DKT framework with the deployment of MSKT, we use a series of benchmark datasets as the input of the system. Then we perform the prediction task on the DKT and make an comparison of two baselines as well as the ablation study.

Datasets

Four benchmark datasets and Riiid's EdNet data[53] were used in our experiments, and the latter was from the Riiid answer correctness prediction challenge, including $1M+$ students. Table 1 presents the details of these datasets used in our experiments.

- ASSIST2009 (https://www.etrialstestbed.org/resources/featured-studies/dataset-papers): This dataset is one of the most effective and most frequently used benchmark datasets for knowledge tracing. It contains about $4k$ students, $18k$ questions, and $328k$ interaction records. More, the data density is low so that it requires a high model generalization capability.
- ASSIST2012 (https://www.etrialstestbed.org/resources/featured-studies/dataset-papers): The original data set contains 180k question types, with $45k$ remaining after processing. Besides, it includes $46k$ students and $5{,}819k$ interactions.
- ASSIST2015 (https://www.etrialstestbed.org/resources/featured-studies/dataset-papers): This dataset has the lowest data density among the dataset we used, containing $19k$ students, 100 questions, and $708k$ interactions.
- ASSIST2017 (https://www.etrialstestbed.org/resources/featured-studies/dataset-papers): The dataset is collec ted by the ASSISTments Longitudinal Data Mining Competition and each user has a high average number of answer records, with a data density of over 80%. It contains $1k$ students, 102 questions and $249k$ interactions.
- Riiid's EdNet (https://www.kaggle.com/c/riiid-test-answer-prediction/data): The dataset is from the Riiid AIEd Challenge 2020 and uses $1M+$ students' learning records to track the state of knowledge. It is the largest publicly available dataset of the world for smart education, containing 99 million student interactions.
- SIAT-DB1[54,55]: This dataset collects human physiological signals for motion state estimation and can be used for rehabilitation exoskeleton robots. It was collected from 8 healthy volunteers and contains surface electromyography (sEMG), force myography (FMG) and hip-knee-ankle joint angles.

Baseline

As we have discussed before, Fig. 4 illustrates the architectures of SAKT and SAINT.

- **SAKT** employs a shallow self-attention encoder that operates solely on basic interaction sequences (e.g., question-response pairs). Its structure lacks explicit modeling of critical metadata such as problem categories, positional contexts, or temporal dynamics (timestamps). The limited depth of attention layers restricts its capacity to capture hierarchical knowledge states or conceptual relationships.
- **SAINT**, as a traditional decoder-based architecture, adopts an encoder-decoder framework. The encoder integrates enriched inputs with embedded metadata (categories, positions, timestamps), while the decoder iteratively refines knowledge representations through multi-head attention layers. This dual-stage design enables SAINT to progressively decode latent knowledge states and temporal dependencies, achieving deeper feature interaction than single-encoder models like SAKT.

Data preprocessing

Datasets described in Section "Datasets" can be regarded as the raw data stored in Evaluation Module of DKT as depicted in Fig. 2. Before it can be fed into the next module, three simple steps are required. Firstly, we clean the dataset by removing duplicate records and filling in missing items. Secondly, the data is grouped by each user's id and sorted in chronological order. Thirdly, it is processed into a quintuple form (sequence length, question id, category, elapsed time, correctness) so as to facilitate subsequent operations. Next, we follow the data flow chart as in Fig. 8 before the process of MSKT:

1. Determine whether the length of the sequence exceeds the maximum length. Here, we set a threshold to control the maximum length of the input sequence, because the computation overhead increases dramatically when the maximum length of sequence grows. In order to achieve a better accuracy within an acceptable

Datasets	Records	Students	Questions	Categories
ASSIST2009	328K	4K	18K	124
ASSIST2012	5819K	46K	150K	266
ASSIST2015	708K	19K	100	100
ASSIST2017	249K	1K	1K	86
Riiid's EdNet	99M	393K	13K	10K

Table 1. Dataset statistics.

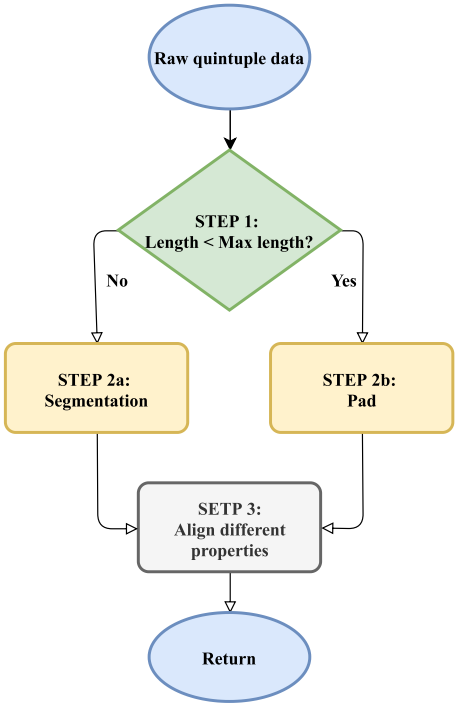

Figure 8. Data flow chart.

Training hyperparameters	Values
Batch size	64
Dropout rate	0.1
Max length	[100,120,140,160,180,200]
Hidden dimension	512
Number of head	8
Encoder layers	3
Decoder layers	2

Table 2. Hyperparameter setting.

runtime, we set up the threshold for the sequence length. Also, in order to study the impact on sequence length, we test a series of max sequence length, from 100 to 200. If it is greater than the maximum length, we cut the perform a cut to remove the extra-long part. If the sequence length is oversized, the oversized part will be cut for meeting the length requirement. If the sequence length is undersized, zero-padding is applied for normalizing it to the length required.

2. Align the sequences with different attributes. For example, SAKT and SAINT are aligned according to the step-by-step approach, as shown in Fig. 7 part A. MSKT is aligned according to the generative alignment, as shown in Fig. 7 part B.
3. Divide the quintuple data into training and test sets at the ratio of 4:1.

Parameters setting

During the training phase, we used the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$. In ASSIST2009, 2015, and 2017 datasets, the number of iterations is 20 and the learning rate is set to $1e$-3. When processing ASSIST2012 and Riiid's EdNet dataset, we set the number of iterations to 3, and the learning rate is set to $1e$-4. Details of other parameters are shown in Table 2.

Quantitative analysis

This section is divided into three parts. First, we compare MSKT with several state-of-the-art models (SAKT, SAINT, SAINT+) on Riiid's EdNet dataset when they are deployed in the DKT. Besides, we use different length sequences to verify the superiority of MSKT on large open datasets. Then, We compare the aforementioned models on multiple benchmark datasets to test the generalization ability of our proposed DKT model. Finally, to prove the effectiveness of our proposed sparse attention and generative decoding approach, we conduct a

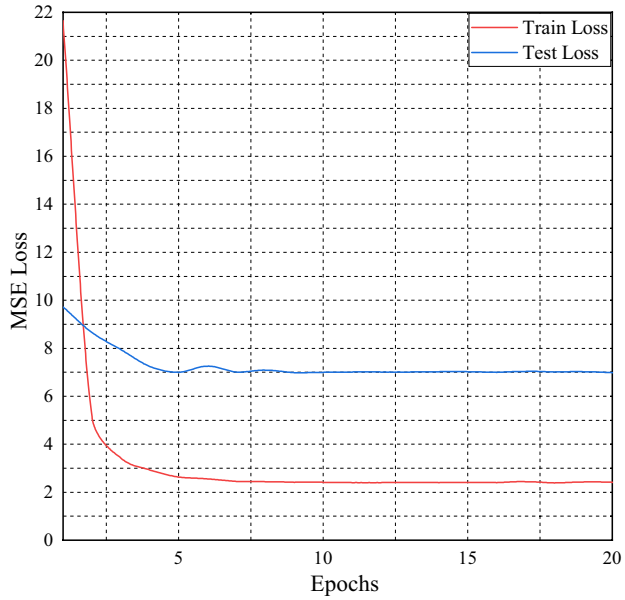

Figure 9. PCC curve.

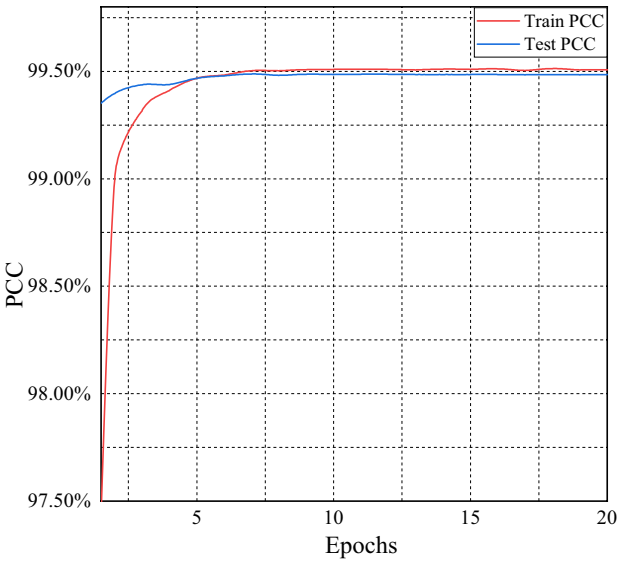

Figure 10. Loss curve.

series of ablation studies. Accuracy (ACC) and Area Under Curve (AUC) are used as evaluation metrics in our experiments, which are common in examining the prediction accuracy for knowledge tracing.

Training dynamics and performance analysis

The training curves (Figs. 9, 10) reveal critical insights into MSKT's performance on the SIAT-DB1 dataset. Both training and testing PCC values exhibit rapid convergence, surpassing 99.5% within the first 10 epochs, demonstrating efficient learning of motion patterns. The final PCC stabilizes near saturation at $\approx$ 99.8%, with a minimal train-test deviation (<0.3%), indicating near-perfect correlation between predicted and ground-truth joint kinematics (hip, knee, ankle). Concurrently, the loss curve shows sharp optimization dynamics: training MSE plummets from 18.2 to 2.1 within 5 epochs before stabilizing at 0.5 (training) and 2.0 (testing) post-epoch 15, confirming stable convergence without oscillations.

Comparison with state-of-the-art methods

Table 3 compares the performance of each model deployed in the MSKT module of DKT in terms of ACC and AUC with sequences of different lengths. As can be seen from the table, ACC and AUC are in the range of 70%-

Metrics	ACC				AUC			
Methods	SAKT	SAINT	SAINT+	MSKT	SAKT	SAINT	SAINT+	MSKT
100	72.00%	70.55%	69.68%	**85.73%**	75.61%	74.78%	73.22%	**91.10%**
120	72.20%	70.89%	69.62%	**86.86%**	75.67%	75.12%	73.38%	**91.95%**
140	72.08%	71.08%	70.41%	**85.75%**	75.66%	75.44%	74.34%	**91.48%**
160	72.12%	71.27%	70.53%	**89.19%**	75.69%	75.73%	74.51%	**93.64%**
180	71.98%	71.43%	70.66%	**89.81%**	75.61%	75.97%	74.70%	**94.14%**
200	72.20%	71.67%	70.71%	**88.19%**	75.82%	76.01%	74.71%	**92.74%**

Table 3. Comparison with several state-of-the-art methods in different sequence length on Riiid's EdNet dataset. Significant values are in bold.

Metrics	Accuracy		Efficiency		
Methods	RMSE	PCC	Time cost (ms)	Flops (M)	Parameters (K)
SAKT	7.51	89.50%	0.89	932.25	146.05
SAINT	7.28	89.94%	0.67	684.03	107.21
MSKT	2.71	**98.86%**	1.13	687.72	107.78

Table 4. Comparison with several state-of-the-art methods in accuracy and efficiency on SIAT-DB1 dataset. The *Accuracy* metric represents the average score of all joints, *e.g.*, hip, knee and ankle. The *Time Cost (ms)* represents the time to infer one sample Significant values are in bold.

Metrics	ACC				AUC			
Datasets	ASSIST2009	ASSIST2012	ASSIST2015	ASSIST2017	ASSIST2009	ASSIST2012	ASSIST2015	ASSIST2017
SAKT	70.25%	73.72%	74.31%	68.88%	72.15%	74.65%	71.39%	70.37%
SAINT	70.14%	70.46%	72.09%	64.38%	71.59%	69.97%	68.86%	67.85%
SAINT+	64.82%	70.04%	72.16%	63.30%	65.34%	65.70%	69.59	67.79%
MSKT	**89.63%**	**81.30%**	**96.43%**	**74.54%**	**94.54%**	**88.53%**	**95.59%**	**73.16%**

Table 5. Comparison with several state-of-the-art methods on different benchmark. Significant values are in bold.

73% and 73%-76% respectively, for SAKT, SAINT, and SAINT+. Meanwhile, ACC and AUC of MSKT are in the range of 85%-90% and 91%-95%, respectively. It shows that MSKT has more than 10% improvement in terms of ACC and AUC, indicating that MSKT has better potential for large-scale educational datasets. In addition, the analysis in the table shows that the three baseline models are lack of sensitivity to variation in sequence length. In contrast, MSKT is sensitive in capturing the changes in sequence length, reaching a maximum when the sequence length is 180. Based on the above observations, we can conclude that MSKT has advantages in the deployment of DKT when dealing with long sequence data. It also facilitates the tracing of students' knowledge states in the study of learning situation analysis.

Comparison on efficiency

Especially, we also test more features of MSKT on the well-known healthcare datasets SIAT-DB1 (see Table 4). On the SIAT-DB1 dataset, the MSKT method demonstrates significant advantages in both accuracy and efficiency. As shown in the table, MSKT achieves a Root Mean Square Error (RMSE) of 2.71, substantially lower than SAKT (7.51) and SAINT (7.28), while its Pearson Correlation Coefficient (PCC) reaches 98.86%, outperforming the state-of-the-art methods (SAKT: 89.50%; SAINT: 89.94%). This highlights MSKT's superior accuracy in predicting joint movements (e.g., hip, knee, ankle). Although MSKT's inference time per sample (1.13 ms) is slightly higher than SAINT (0.67 ms), its computational cost (Flops) (687.72 M) and parameter size (107.78 K) remain comparable to SAINT. Notably, MSKT achieves a breakthrough in accuracy with even lower computational complexity, demonstrating that its structural optimization effectively enhances performance while maintaining a lightweight architecture.

Comparison on different benchmarks

Table 5 compares MSKT with SAKT, SAINT, and SAINT+ on different benchmark datasets. We can see that MSKT improves about 20%, 10%, 20%, and 5% on ASSIST2009, ASSIST2012, ASSIST2015, and ASSIAT2017, respectively, in contrast to other models. This also shows that MSKT has strong robustness and strong generalization ability under different benchmark datasets. Notably, for ASSIST2009 and ASSIST2015, MSKT increases by about 20% in terms of ACC and AUC. In fact, the data density of these two datasets are 0.06 and

Metrics	ACC				AUC			
Datasets	ASSIST2009	ASSIST2012	ASSIST2015	ASSIST2017	ASSIST2009	ASSIST2012	ASSIST2015	ASSIST2017
MSKT	**89.63%**	81.30%	**96.43%**	**74.54%**	**94.54%**	**88.53%**	**95.59%**	**73.16%**
MSKT-GD	88.39%	81.75%	95.44%	72.22%	93.19%	88.01%	95.00%	69.84%
MSKT-GD-SA	64.82%	70.04%	72.16%	63.30%	65.34%	65.70%	69.59%	67.79%

Table 6. Comparison with MSKT's variants. MSKT: Multi-head sparse attention for knowledge tracing. MSKT-GD: MSKT without generative decoder. MSKT-GD-SA: MSKT without generative decoder and sparse attention

0.05, meaning that the two are the sparsest in all the datasets. In knowledge tracing, how to deal with sparse data sequences has been a troublesome problem. The good performance result of MSKT indicates that the sparse attention mechanism has its advantages in dealing with sparse data.

Ablation study

In this section, we qualitatively analyze our two proposed strategies, namely, the sparse attention mechanism and the generative decoder. For ease of presentation, we use MSKT-GD to denote the MSKT model without generative decoder and use MSKT-GD-SA to denote the MSKT model without the generative decoder and sparse attention mechanism. Table 6 shows that the accuracy of MSKT decreases by more than 10%, if the generative decoder or sparse attention is solely used, indicating that these two components have a significant impact on the model performance. Additionally, the decrease in accuracy of MSKT-GD shows that our generative decoder can improve the quality of the prediction results. Moreover, in contrast to MSKT-GD, the correct prediction rate of MSKT-GD-SA declines after removing the sparse attention, indicating that the sparse attention encoder largely refines the quality of the model. In summary, both strategies have good effect over the prediction results, thus verifying the rational of our model designing.

Conclusion

In this study, we propose a novel multi-head sparse attention model called MSKT for knowledge tracing in DKT. MSKT alleviates the problem of data sparsity with a sparse attention mechanism. In additional, we substitute the traditional step-by-step decoder with a generative prediction decoder. The experimental results show that our proposed MSKT model fits well in the Deep Knowledge Tracing Framework and achieves a better results on both large-scale datasets and multiple benchmark datasets, compared to the state-of-the-art methods. But it should be noted that blended learning is more common in real-world situations, which means in-person schooling is often combined with distance learning and this has raised questions on the effectiveness of learning as students and teachers alternate between these two modes of delivery[5]. In such circumstances, following students' real-time learning trajectories to provide high-quality, real-time collaborative feedback on students' work is essential to teaching[56]. For future work, we would like to further improve the interpretability and the generalization of our models and extensively examine the effectiveness of our proposals with on more benchmark datasets and real-world deployment scenarios.

Data availability

The data that support the findings of this study are publicly available at: https://www.etrialstestbed.org/resources/featured-studies/dataset-papers. https://www.kaggle.com/c/riiid-test-answer-prediction/data

Received: 10 October 2024; Accepted: 16 June 2025
Published online: 01 July 2025

References

1. Basnet, R. B., Johnson, C. & Doleck, T. Dropout prediction in moocs using deep learning and machine learning. *Edu. Infor. Technol.* 1–15 (2022).
2. Guo, L. et al. Evolution and trends in intelligent tutoring systems research: A multidisciplinary and scientometric view. *Asia Pac. Educ. Rev.* **22**(3), 441–461 (2021).
3. Kamal, M. I., Zubanova, S., Isaeva, A. & Movchun, V. Distance learning impact on the English language teaching during covid-19. *Educ. Inf. Technol.* **26**(6), 7307–7319 (2021).
4. El Alaoui, D., Riffi, J., Aghoutane, B., Sabri, A., Yahyaouy, A. & Tairi, H. Overview of the main recommendation approaches for the scientific articles. In *International Conference on Business Intelligence*, pp. 107–118 (2021). Springer
5. OECD: The State of Global Education: 18 Months Into the Pandemic. OECD Publishing, Paris, France (2021).
6. Cerna, L., Rutigliano, A. & Mezzanotte, C. The impact of covid-19 on student equity and inclusion: Supporting vulnerable students during school closures and school re-openings. *Organisation for Economic Co-operation and Development*. 34 (2020).
7. Oya, T. & Morishima, S. Lstm-sakt: Lstm-encoded sakt-like transformer for knowledge tracing, 2nd place solution for riiid! answer correctness prediction. arXiv e-prints, 2102 (2021).
8. Lu, Y., Wang, D., Chen, P., Meng, Q. & Yu, S. Interpreting deep learning models for knowledge tracing. *Int. J. Artif. Intell. Educ.* **33**(3), 519–542 (2023).
9. El Alaoui, D. et al. Deep graphsage-based recommendation system: Jumping knowledge connections with ordinal aggregation network. *Neural Comput. Appl.* **34**(14), 11679–11690 (2022).
10. Corbett, A. T. & Anderson, J. R. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User-Adap. Inter.* **4**(4), 253–278 (1994).

11. Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J. & Sohl-Dickstein, J. Deep knowledge tracing. *Adv. Neural Infor. Process. Syst.***28** (2015).
12. Trifa, A., Hedhili, A. & Chaari, W. L. Knowledge tracing with an intelligent agent, in an e-learning platform. *Educ. Inf. Technol.* **24**(1), 711–741 (2019).
13. Wang, P., Li, L., Wang, R., Xie, Y. & Zhang, J. Complexity-based attentive interactive student performance prediction for personalized course study planning. *Edu. Infor. Technol.*, 1–23 (2022).
14. El Alaoui, D., Riffi, J., Aghoutane, B., Sabri, A., Yahyaouy, A. & Tairi, H. Collaborative filtering: comparative study between matrix factorization and neural network method. In *Networked Systems: 8th International Conference, NETYS 2020, Marrakech, Morocco*, June 3–5, 2020, Proceedings 8, pp. 361–367 (2021). Springer
15. El Alaoui, D. et al. Comparative study of filtering methods for scientific research article recommendations. *Big Data Cognit. Comput.* **8**(12), 190 (2024).
16. Vie, J.-J. & Kashima, H. Knowledge tracing machines: Factorization machines for knowledge tracing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 750–757 (2019).
17. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016).
18. Liu, Q. et al. Ekt: Exercise-aware knowledge tracing for student performance prediction. *IEEE Trans. Knowl. Data Eng.* **33**(1), 100–115 (2019).
19. Pian, Y., Lu, Y., Huang, Y. & Bittencourt, I. I. A gamified solution to the cold-start problem of intelligent tutoring system. In *International Conference on Artificial Intelligence in Education*, pp. 376–381 (2020).
20. Chen, P., Lu, Y., Zheng, V. W. & Pian, Y. Prerequisite-driven deep knowledge tracing. In *2018 IEEE International Conference on Data Mining (ICDM)*, IEEE, pp. 39–48 (2018).
21. Nguyen, V. A. The impact of online learning activities on student learning outcome in blended learning course. *J. Infor. & Knowl. Manag.* **16**(04), 1750040 (2017).
22. El Alaoui, D., Riffi, J., Sabri, A., Aghoutane, B., Yahyaouy, A. & Tairi, H. Social recommendation system based on heterogeneous graph attention networks. *Int. J. Data Sci. Anal.*, 1–17 (2024)
23. Reimers, F., Schleicher, A., Saavedra, J. & Tuominen, S. Supporting the continuation of teaching and learning during the covid-19 pandemic. *OECD* **1**(1), 1–38 (2020).
24. OECD: Schooling Disrupted, Schooling Rethought: How the Covid-19 Pandemic Is Changing Education. (2020).
25. United Nations Educational, S. & Organization, C. COVID-19 Response – Hybrid Learning: Hybrid Learning as a Key Element in Ensuring Continued Learning. (2020).
26. El Alaoui, D., Riffi, J., Sabri, A., Aghoutane, B., Yahyaouy, A. & Tairi, H. A novel session-based recommendation system using capsule graph neural network. *Neural Netw.*, 107176 (2025).
27. El Alaoui, D., Riffi, J., Sabri, A., Aghoutane, B., Yahyaouy, A. & Tairi, H. Contextual recommendations: dynamic graph attention networks with edge adaptation. *IEEE Access* (2024).
28. Minn, S., Yu, Y., Desmarais, M. C., Zhu, F. & Vie, J.-J. Deep knowledge tracing and dynamic student classification for knowledge tracing. In *2018 IEEE International Conference on Data Mining (ICDM)*, IEEE, pp. 1182–1187 (2018).
29. Anderson, J. R., Boyle, C. F., Corbett, A. T. & Lewis, M. W. Cognitive modeling and intelligent tutoring. *Artif. Intell.* **42**(1), 7–49 (1990).
30. Pandey, S., Karypis, G. & Srivastava, J. An empirical comparison of deep learning models for knowledge tracing on large-scale dataset. arXiv preprint arXiv:2101.06373 (2021).
31. Kasurinen, J. & Nikula, U. Estimating programming knowledge with Bayesian knowledge tracing. *ACM SIGCSE Bull.* **41**(3), 313–317 (2009).
32. Van De Sande, B. Properties of the Bayesian knowledge tracing model. *J. Edu. Data Min.* **5**(2), 1–10 (2013).
33. D Baker, R. S., Corbett, A. T. & Aleven, V. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *International Conference on Intelligent Tutoring Systems*, Springer, pp. 406–415 (2008).
34. Villano, M. Probabilistic student models: Bayesian belief networks and knowledge space theory. In *International Conference on Intelligent Tutoring Systems*, Springer. pp. 491–498 (1992).
35. Zhang, J., Shi, X., King, I. & Yeung, D.-Y. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th International Conference on World Wide Web*, pp. 765–774 (2017).
36. Wang, F., Liu, Q., Chen, E., Huang, Z., Chen, Y., Yin, Y., Huang, Z. & Wang, S. Neural cognitive diagnosis for intelligent education systems. In *Proceedings of the AAAI Conference on Artificial Intelligence* vol. 34, pp. 6153–6161 (2020).
37. Lee, Y., Choi, Y., Cho, J., Fabbri, A.R., Loh, H., Hwang, C., Lee, Y., Kim, S.-W. & Radev, D. Creating a neural pedagogical agent by jointly learning to review and assess. arXiv preprint arXiv:1906.10910 (2019).
38. Pandey, S. & Karypis, G. A self-attentive model for knowledge tracing. arXiv preprint arXiv:1907.06837 (2019).
39. Choi, Y., Lee, Y., Cho, J., Baek, J., Kim, B., Cha, Y., Shin, D., Bae, C. & Heo, J. Towards an appropriate query, key, and value computation for knowledge tracing. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*, pp. 341–344 (2020).
40. Shin, D., Shim, Y., Yu, H., Lee, S., Kim, B. & Choi, Y. Saint+: Integrating temporal features for ednet correctness prediction. In *LAK21: 11th International Learning Analytics and Knowledge Conference*, pp. 490–496 (2021).
41. Zeng, J., Zhang, Q., Xie, N. & Yang, B. Application of deep self-attention in knowledge tracing. arXiv preprint arXiv:2105.07909 (2021).
42. Tran, D. K. L. Riiid! answer correctness prediction kaggle challenge: 4th place solution summary. arXiv preprint arXiv:2102.04250 (2021).
43. Pandey, S. & Srivastava, J. Rkt: Relation-aware self-attention for knowledge tracing. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 1205–1214 (2020).
44. Ghosh, A., Heffernan, N. & Lan, A. S. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2330–2339 (2020).
45. Song, X. et al. Jkt: A joint graph convolutional network based deep knowledge tracing. *Inf. Sci.* **580**, 510–523 (2021).
46. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. & Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. *Adv. Neural Infor. Process. Syst.***30** (2017).
47. Khanal, S. S., Prasad, P., Alsadoon, A. & Maag, A. A systematic review: Machine learning based recommendation systems for e-learning. *Educ. Inf. Technol.* **25**(4), 2635–2664 (2020).
48. Schodde, T., Bergmann, K. & Kopp, S. Adaptive robot language tutoring based on bayesian knowledge tracing and predictive decision-making. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 128–136 (2017).
49. Kantharaju, P., Alderfer, K., Zhu, J., Char, B., Smith, B. & Ontanón, S. Tracing player knowledge in a parallel programming educational game. In *Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference* (2018).
50. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. Attention is all you need. *Adv. Neural Infor. Process. Syst.***30** (2017).
51. Liu, Q., Shen, S., Huang, Z., Chen, E. & Zheng, Y. A survey of knowledge tracing. arXiv preprint arXiv:2105.15106 (2021).
52. Luong, M.-T., Pham, H. & Manning, C. D. Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025 (2015).

53. Choi, Y., Lee, Y., Shin, D., Cho, J., Park, S., Lee, S., Baek, J., Bae, C., Kim, B. & Heo, J. Ednet: A large-scale hierarchical dataset in education. In *International Conference on Artificial Intelligence in Education*, Springer, pp. 69–73 (2020).
54. Zhou, X. et al. Continuous estimation of lower limb joint angles from multi-stream signals based on knowledge tracing. *IEEE Robot. Autom. Lett.* **8**(2), 951–957 (2023).
55. Zhou, X., Lin, C., Wang, C. & Peng, X. semg-based joint angle estimation via hierarchical spiking attentional feature decomposition network. *IEEE Robot. Autom. Lett.* (2025)
56. Reimers, F. & Opertti, R. *Learning to build back better futures for education: Lessons from educational innovation during the COVID-19 pandemic* (International Bureau of Education, Geneva. UNESCO, 2021).

Author contributions

Conceptualization: [Jiawei Zhang and Xike Xie]; Methodology: [Xin Zhou, Zhuoxu Zhang, Xike Xie, and Jiawei Zhang]; Formal analysis and investigation: [Xin Zhou, Zhuoxu Zhang]; Writing - original draft preparation: [Xin Zhou, Zhuoxu Zhang]; Writing - review and editing: [Jiawei Zhang and Xike Xie]; Funding acquisition: [Jiawei Zhang and Xike Xie]; Resources: [Jiawei Zhang and Xike Xie]; Supervision: [Jiawei Zhang and Xike Xie].

Declarations

Competing interests

The authors declare that they have no conflicts of interest.

Additional information

Correspondence and requests for materials should be addressed to J.Z.

Reprints and permissions information is available at www.nature.com/reprints.

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