SLPKT: A Novel Simulated Learning Process Model for Knowledge Tracing

Jingxia Zeng[1], Mianfan Chen[1], Jianing Liu[1], Yuncheng Jiang[1,2]*
[1] School of Computer Science, South China Normal University, Guangzhou, China
[2] School of Artificial Intelligence, South China Normal University, Foshan, China
{2021023230, jiangyuncheng }@m.scnu.edu.cn

***Abstract*—Knowledge Tracing (KT) traces students' changing knowledge states and predicts future performance based on their past performance. However, most existing methods underestimate the impact of students' learning processes on performance prediction, and thus do not model the learning process well. To address this issue, we propose a novel Simulated Learning Process Model for Knowledge Tracing, which simulates the student's learning process by reviewing historical performance and enhancing the role of related knowledge in prediction before solving exercises. We first use a state acquisition module to obtain the knowledge state. Then we mine important historical information to assist in solving the target exercise. Finally, a knowledge enhancement module is used to improve the knowledge prediction of the target exercise. Extensive experiments on four real-world datasets demonstrate that our method is effective and outperforms the state-of-the-art models.**

***Index Terms*—Knowledge tracing, Learning process, Historical performance, Knowledge enhancement**

I. Introduction

In recent years, the usage of various online education systems has increased significantly, and students generate a significant amount of learning data on these platforms and systems [3]. The primary research task for these learning data is to infer the student's mastery level of knowledge based on these learning records, and then provide him with follow-up personalized services, such as learning path suggestions [6], course recommendations [5], and adaptive testing [18].

Knowledge Tracing (KT) [2] is an emerging research area in online learning, which uses the student's past performance to predict the student's future performance. In recent years, KT has gained widespread applications, drawing increasing attention from academia, and various methods for dealing with it have been proposed [4], [7], [9], [16], [19], [20].

Most existing KT methods employ Recurrent Neural Networks (RNNs) to process students' learning records [4], [13], [21]. To predict student's performance more accurately, there are some methods that have considered the influence of the student's learning process. Existing methods for studying student's learning process primarily suggest that student's forgetting behavior and learning ability will influence final performance.

However, the existing methods are primarily restricted to the forgetting effect and the learning ability difference [10], [14], [15], which do not fully model student's learning process,

DOI reference number: 10.18293/SEKE23-049

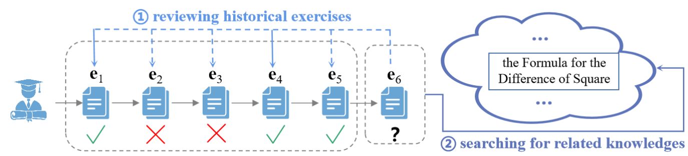

Fig. 1. The learning process after students get a new exercise. In particular, $\mathbf{e_1}$, $\mathbf{e}_4$, and $\mathbf{e}_6$ are exercises that involve the FDS.

thus limiting the model's predictive accuracy. In addition to the factors described above, we believe that the following learning process also affects the performance: After receiving a new exercise, students will go through a process of reviewing whether they have done similar problems, and then search for related knowledge and information to generate an idea to solve this new exercise. For example, as shown in Fig. 1, there has a student who is given a new exercise that involves the Formula for the Difference of Square (FDS). To complete this new exercise, first, he will review whether he has done exercises about the FDS before. If the student has solved similar exercises correctly or used the same concept (FDS) correctly to solve some exercises, the probability of the student solving this new exercise correctly should be higher, and vice versa. Furthermore, whether he did exercises related to the FDS or not, he will go through a process of searching for the FDS in his mind to solve this new exercise.

Therefore, we propose a novel Simulated Learning Process Model for KT (SLPKT), which can model the student's learning process described above. Specifically, after the student get a new(target) exercise, we first use a knowledge state acquisition module to obtain the knowledge state. Then, to mine important historical information, we trace historical states based on the similarities between the target's knowledge concept and the knowledge concepts of the historical exercises. Finally, a knowledge enhancement module is used to improve the prediction accuracy of the target exercise. Furthermore, we conduct a series of experiments on four public datasets and compare them to existing KT methods. Our results indicate that SLPKT outperforms existing KT methods in predicting student performance.

We summarize the key contributions of SLPKT below:

- To our knowledge, SLPKT is the first time that students' learning processes are fully simulated to predict their performance after they get the new exercises.

- We design a historical review module to retrieve historical important information to assist in solving the target exercise, and a related knowledge enhancement module that can enhance the role of knowledge related to the target exercise in predicting performance.
- Numerous experimental results on four public datasets show that SLPKT outperforms the state-of-the-art models, demonstrating the effectiveness of our method.

II. Related Works

The existing KT methods can be roughly divided into two types: traditional knowledge tracing and deep knowledge tracing. Since most of the current works are deep knowledge tracing, thus we mainly introduce deep knowledge tracing.

A. Traditional knowledge tracing

Bayesian Knowledge Tracing (BKT) [2] is a classic probability model on KT, which can be regarded as a special case of the hidden Markov model (HMM). Performance Factors Analysis (PFA) [12] and Learning Factors Analysis (LFA) [1] are essentially traditional psychometric models.

B. Deep knowledge tracing

Deep Knowledge Tracing (DKT) [13] introduces deep learning to KT for the first time. Augmenting Knowledge Tracing by Forgetting (DKT+Forgetting) [9] adds forgetting features to the DKT. Dynamic Key-Value Memory Network (DKVMN) [21] introduces memory enhancement neural networks into KT. To obtain knowledge state, Exercise-Enhanced sequential modeling for student performance prediction (EERNNA) [17] calculates the attention weights between hidden states by calculating the cosine similarity between the exercises. Context-aware Attentive Knowledge Tracing (AKT) [4] uses the same text feature extraction method as the EERNNA model, and further mines students' guessing and mistaken behavior from the proposed semantic features. A Graph-based Interaction model for Knowledge Tracing (GIKT) [19] uses graph convolutional networks to capture exercise representations and knowledge concepts from the diagram of exercises, and uses a recap module to review relevant historical exercises to help students solve problems. Learning Process-consistent Knowledge Tracing (LPKT) [15] directly uses a learning-gain module and a forgetting module to model the student's learning process to monitor the student's knowledge state.

Although small parts of these methods modeled student's learning process, they are all limited to the forgetting effect and learning ability difference. These methods do not take good account of the impact of student's learning process. Therefore, we proposed SLPKT to simulate the learning process of students after they receive a new exercise to improve the performance and interpretability of the model.

III. Preliminary

In an online education system, assuming there are the set of students $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_I\}$, the set of exercises $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_J\}$, and the set of knowledge concepts $\mathbf{L} = \{\mathbf{l}_1, \mathbf{l}_2, \ldots, \mathbf{l}_T\}$. Since each exercise is related to a specific knowledge concept, a dictionary is used to represent the relationship between the exercise and the concept, such as dictionary $\mathbf{dict} = \{\ldots, \mathbf{e}_i : [\mathbf{l}_m, \ldots, \mathbf{l}_n], \ldots\}$, indicating that $\mathbf{e}_i$ is related to $\mathbf{l}_m$, ..., and $\mathbf{l}_n$, and so on.

In the KT task, students answer a series of exercises provided by the online learning platform sequentially, and the system will provide feedback on each exercise's accuracy once the student has responded. Given an interaction sequence $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t\}$ and a new question $\mathbf{e}_{t+1}$, where $\mathbf{x}_t = (\mathbf{e}_t, \mathbf{c}_t, \mathbf{a}_t)$, $\mathbf{e}_t$ is the exercise, $\mathbf{c}_t$ is the value corresponding to $\mathbf{e}_t$ in the dictionary $\mathbf{dict}$, and $\mathbf{a}_t$ indicates whether the $\mathbf{e}_t$ is answered correctly (1 means correct, 0 means false). The goal of KT is to monitor students' changing knowledge state during the learning process and predict their performance at the next learning step $t+1$.

IV. Methodology

In this section, we will introduce SLPKT in detail. As shown in Fig. 2, the model consists of four modules : (a) knowledge state acquisition module, (b) historical knowledge state review module, (c) related knowledge enhancement module, and (d) prediction module. Specifically, after the student is given a new exercise, the knowledge state acquisition module first obtains the knowledge state based on his past performance. Then the historical state review module reviews his historical performances to mine important historical information. Furthermore, the knowledge enhancement module enhances the role of related knowledge of this new exercise in predicting performance. Finally, we use the prediction module to predict student's performance on this new exercise.

A. Knowledge State Acquisition

During the learning process, since student's mastery of knowledge is constantly changing, we need to model the entire process of student learning to capture the changing knowledge state. For each learning step, we connect the exercise and the answer, and then project them into the d-dimension through linear transformations. So the input of the model is:

$$\mathbf{x}_t = \mathbf{W}_x^T[\mathbf{e}_t \oplus \mathbf{a}_t] + \mathbf{b}_x, \tag{1}$$

where $\oplus$ denotes the vector concatenation operation, $\mathbf{W}_x \in \mathbb{R}^{2d \times d}$ is the weight and $\mathbf{b}_x$ is the bias. Then, we use LSTM to obtain the basic knowledge state $\mathbf{h}$:

$$\mathbf{h}_t = LSTM(\mathbf{x}_t, \mathbf{h}_{t-1}), \tag{2}$$

where $\mathbf{h}_t$ and $\mathbf{h}_{t-1}$ represent student's knowledge states.

B. Historical Knowledge State Review

Whenever students are faced with a new problem, most of them first will review whether they have done similar exercises or used the same knowledge concept to solve some exercises. Thus, when completing the prediction task of KT, it should not only depend on the current knowledge state but also consider whether the historical knowledge states have reference significance for the current prediction work.

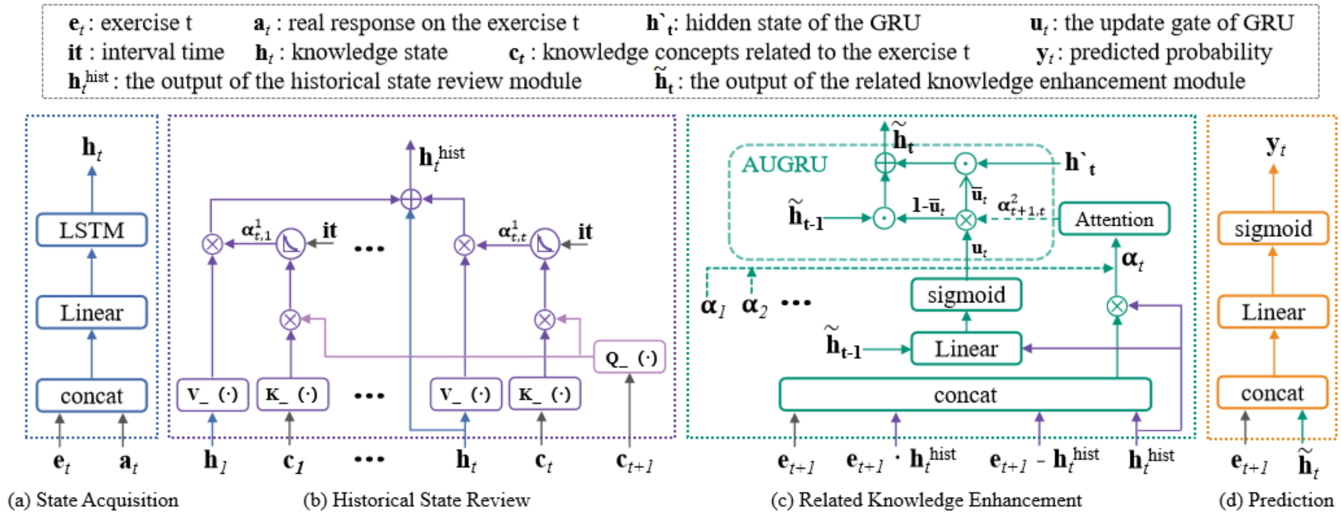

Fig. 2. The architecture of SLPKT. Assuming that t is the current learning step, our task is to obtain the student's knowledge state generated at this learning step, and then predict student's performance in exercise $\mathbf{e}_{t+1}$ based on the knowledge state $\tilde{\mathbf{h}}_t$.

Therefore, we use an attention-based neural network model to calculate the similarities between the knowledge concept involved in the target exercise and the knowledge concepts involved in all historical exercises. Then, according to these similarities, the corresponding historical knowledge states are integrated into the current knowledge state. In addition, we do not use the basic scaled dot attention mechanism because learning is temporary and memory declines.

Specifically, there are a key, query and value embedding layer that map inputs to keys, queries and values respectively. Let $\mathbf{q}_t$ and $\mathbf{k}_t$ represent the query and the key, which correspond to the knowledge concept involved in the exercise answered by the learner, and $\mathbf{v}_t$ is the value corresponding to the knowledge state at the learning step t:

$$\mathbf{q}_t = K\_(\cdot)(\mathbf{c}_t),\ \ \mathbf{k}_t = Q\_(\cdot)(\mathbf{c}_t),\ \ \mathbf{v}_t = V\_(\cdot)(\mathbf{h}_t), \tag{3}$$

where $\mathbf{c}_t$ and $\mathbf{h}_t$ are the knowledge concept and the knowledge state respectively. And we use the softmax function to calculate the $\alpha_{t,m}$ of the scaled dot product attention value:

$$\alpha^1_{t+1,m} = Softmax(\frac{exp(-\theta \cdot D(t+1,m))\mathbf{q}^T_{t+1}\mathbf{k}_m}{\sqrt{d}}), \tag{4}$$

where $\mathbf{k}_m$ is the concept embedding through $K\_(\cdot)$ at the learning step m, and $1 \leq m \leq t$ means we depend on all the past learning steps. θ >0 is a learnable decay rate parameter and $D(t+1,m)$ is temporal distance measure between time steps $t+1$ and m.

Considering that a student may complete all the exercises many days apart, the relative distance of the exercises and the interval time are used to control the decay rate of the decay function simultaneously:

$$it = \begin{cases} 1, & if\ the\ interval\ time > 0, \\ 0, & otherwise. \end{cases} \tag{5}$$

$$D(t+1,m) = (\mid t+1-m \mid + it) \cdot \sum_{n=m+1}^{t+1} \gamma_{t+1,n},$$
$$\gamma_{t+1,n} = Softmax(\frac{\mathbf{q}^T_{t+1}\mathbf{k}_n}{\sqrt{d}}),\ \ \forall\ n\ \leq\ t+1. \tag{6}$$

Then, according to the similarities of the calculated concepts, the corresponding historical knowledge states are integrated into the current knowledge state, and theses important historical information is used to assist in problem-solving:

$$\mathbf{h}^{hist}_t = \mathbf{h}_t + \sum_{m=1}^{t}(\alpha^1_{t+1,m} \cdot \mathbf{v}_m). \tag{7}$$

C. Related Knowledge Enhancement

Since students' mastery of each knowledge concept has its own independent evolution process, and students also will have a process of searching for related knowledge in their minds to solve the exercise after they get a new exercise. Therefore, to be able to better simulate this learning process, we need to enhance the role of knowledge mastery related to the target exercise on prediction.

To model the above behavior, we first calculate the correlation between the target exercise and the knowledge state, and the module's inputs are the target exercise and the knowledge state, as shown in the Fig. 2 (c). Then, we normalize the correlation of each learning step with the correlation calculated by all previous learning steps:

$$\begin{aligned} &\alpha_t = \mathbf{h}^{hist}_t \mathbf{W}[\mathbf{e}_{t+1} \oplus \mathbf{h}^{hist}_t \oplus (\mathbf{e}_{t+1} - \mathbf{h}^{hist}_t) \\ &\oplus (\mathbf{e}_{t+1} \cdot \mathbf{h}^{hist}_t)], \\ &\alpha^2_{t+1,t} = Softmax(\alpha_t), \end{aligned} \tag{8}$$

where $\mathbf{e}_{t+1}$ is the exercise at the learning step $t+1$ and $\mathbf{h}^{hist}_t$ is the knowledge state which has reviewed the historical information at the learning step t, $\mathbf{W} \in \mathbb{R}^{d \times 4d}$, d is the dimension of $\mathbf{e}_{t+1}$'s embedding vector.

Then, we put this correlation and knowledge state into the Gate Recurrent Unit (GRU) with an attentional update gate (AUGRU) for updating, which is a module transformed by DIEN [22].

$$\begin{aligned} \mathbf{u}_t &= \sigma(\mathbf{W}_u\,[\tilde{\mathbf{h}}_{t-1}, \mathbf{h}_t^{hist}] + \mathbf{b}_u), \\ \bar{\mathbf{u}}_t &= \alpha_{t+1,t}^2 \mathbf{u}_t, \end{aligned} \tag{9}$$

where $\mathbf{u}_t$ is the update gate of GRU, $\mathbf{W} \in \mathbb{R}^{d\times d}$ is the weight, $\mathbf{b}$ is the bias, $\bar{\mathbf{u}}_\mathbf{t}$ is the attention update gate and $\mathbf{h}_t'$ is the hidden states of AUGRU. And $\tilde{\mathbf{h}}_t$ is the final knowledge state generated by SLPKT:

$$\tilde{\mathbf{h}}_t = (1 - \bar{\mathbf{u}}_t)\cdot \tilde{\mathbf{h}}_{t-1} + \bar{\mathbf{u}}_t \cdot \mathbf{h}_t'. \tag{10}$$

D. Prediction

In the student's learning process, after the student is given a new exercise $\mathbf{e}_{t+1}$, he will get a knowledge state to solve this new exercise according to the above process. Therefore, we use the relevant knowledge state $\tilde{\mathbf{h}}_t$ to infer the student's performance on $\mathbf{e}_{t+1}$. We first connect the knowledge state $\tilde{\mathbf{h}}_t$ and the exercise $\mathbf{e}_{t+1}$, and then project them to the output layer through a fully connected network activated by sigmoid:

$$\mathbf{y}_{t+1} = \sigma(\mathbf{W}_y^T[\mathbf{e}_{t+1} \oplus \tilde{\mathbf{h}}_t] + \mathbf{b}_y) \in [0, 1], \tag{11}$$

where $\mathbf{W}_y$ is the weight and $\mathbf{b}_y$ is the bias. Output $\mathbf{y}_{t+1}$ indicates the expected performance of the student in $\mathbf{e}_{t+1}$.

TABLE I
DATASET STATISTICS.

Dataset	ASSIST2009	ASSIST2012	ASSIST2017	EdNet1
Students	3,852	27,485	1,709	5,000
Exercises	17,737	53,065	3,162	12,022
Concepts	123	265	102	142
Interactions	282,606	2,709,436	942,816	676,985
Avg.length	82.72	93.64	551.68	126.32

V. EXPERIMENT

In this section, we conducted several experiments to investigate the effectiveness of SLPKT. First, we assess prediction error by comparing our model to other baselines on four common datasets. Then, we performed an ablation study on the historical knowledge state review module and the related knowledge enhancement module to demonstrate their effectiveness. Finally, we investigate the effect of different fixed lengths on our model and visualize the knowledge state obtained from each module to demonstrate that SLPKT has learned a more meaningful knowledge state.

A. Datasets

Four public datasets are used to evaluate the model's validity which are commonly used in KT tasks. Table I shows the statistics for all datasets. We filter records without knowledge concepts. A brief description of all datasets is listed below:

ASSIST2009 comes from ASSISTments, an online tutoring system created in 2004. ASSIST2012 is collected from the same platform as ASSIST2009 during the school year 2012-2013. In this dataset, each exercise is only related to one knowledge concept, but one knowledge concept corresponds to several exercises. ASSIST2017 was utilized in the 2017ASSIST data mining competition. EdNet1 is a dataset of all student interactions with the system that Santa collected over two years.

B. Training Details

Data Preprocessing. We first sort all of the students' learning records based on the learning steps of the answers, and then set all input sequences of the dataset to a fixed length of 200. For sequences shorter than a fixed length, it is filled to a fixed length with a vector of zeros. And student interactions with a sequence length of less than 3 are removed from all datasets.

Training Settings. We performed a standard 5-fold cross-validation on all models for all datasets. For each fold, 80% of the students are divided into the training set (60%) and the validation set (20%), and the remaining 20% as the test set. We randomly initialize all parameters in an even distribution, all hyperparameters are learned on the training set, and the test set is evaluated using the model that performs best on the validation set. Finally, we set some necessary parameters: The number of dimensions d mentioned in the text is 128, the batch size is 64, and the learning rate of the Adam algorithm is 0.002 for all trainable parameters. Finally, we select the area under the ROC curve (AUC) and accuracy (ACC) as evaluation indicators.

C. Baseline Methods

We compare the model with several previous methods. For a fair comparison, all of these methods are tuned for optimal performance.

- **DKT** [13] uses RNNs to assess student's knowledge state, but we use LSTM in our implementation.
- **DKT+** [9] adds forgetting features to the DKT, which are the interval time of the same exercises, the interval time of adjacent exercises, and the number of historical exercises of the target exercise.
- **DKVMN** [21] proposes a new dynamic key-value storage network model, which has a static matrix and a dynamic matrix to store and update the mastery of corresponding concepts respectively.
- **SAKT** [11] introduces the self-attention model to capture the correlation of relevant exercises from previous interactions to make predictions.
- **AKT** [4] uses two self-attention encoders to learn context-aware representations of exercises and answers, and it uses hyperparameters to retrieve knowledge gained in the past that is relevant to the current exercise.
- **GIKT** [19] uses a graph convolutional network to capture exercise representations from the relation graph of exercise and knowledge concepts, and uses a recap module to capture long-term dependencies.
- **LPKT** [15] is designed to complete KT tasks by simulating students' learning and memory processes, and the model monitors students' knowledge state by directly

TABLE II
AUC AND ACC VALUES OF ALL COMPARISON METHODS ON FOUR DATASETS.

Method	ASSIST2009		ASSIST2012		ASSIST2017		EdNet1	
	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
DKT	0.7455	0.7265	0.7291	0.7351	0.7235	0.6910	0.6857	0.6898
DKT+	0.7528	0.7283	0.7405	0.7395	0.7286	0.6996	0.6723	0.6684
DKVMN	0.7345	0.7181	0.7186	0.7251	0.7142	0.6802	0.6752	0.6861
SAKT	0.6874	0.6857	0.7188	0.7258	0.6683	0.6980	0.6846	0.6983
AKT	0.7627	0.7293	0.7698	0.7553	0.7582	0.7190	0.7302	0.7130
GIKT	0.7686	0.7242	0.7719	0.7466	0.7652	0.7090	0.7380	0.7166
LPKT	-	-	*0.7770*	*0.7568*	*0.7935*	0.7338	0.7371	0.7156
KSGAN	*0.7740*	*0.7355*	0.7736	0.7557	0.7791	0.7247	*0.7426*	*0.7181*
CoKT	0.7685	0.7329	0.7435	0.7385	0.7911	*0.7339*	0.7399	0.7102
SLPKT	**0.7813**	**0.7387**	**0.8038**	**0.7679**	**0.8016**	**0.7387**	**0.7636**	**0.7300**

[1] '-' indicates that the model is not suitable for the dataset.
[2] The best results are bold, and the second-best results are italics.

modeling students' learning processes using learning gain module and forgetting module.

- **KSGAN** [8] uses a Graph-Attention Network (GAT) based model that leverages the knowledge structure between concepts and exercises to predict students' performance.
- **CoKT** [7] retrieves the sequences of peer students who have similar question-answering experiences to obtain the inter-student information to make predictions.

D. Student Performance Prediction

One of the most important metrics for evaluating the KT method is the experimental results of student's performance prediction, so we conduct extensive experiments on all datasets to compare SLPKT to all baselines of student performance prediction and report the results of the five test folds in Table II. To provide robust evaluation results, performance was evaluated using AUC and ACC in all experiments. From Table II, we can see that SLPKT outperforms all other KT methods across all datasets and metrics, thus we believe that SLPKT is more aligned with students' learning processes, resulting in more accurate predictions of their future performance. In addition, we note that on the ASSIST2012 dataset, the model is significantly better than the state-of-the-art LPKT model (AUC is improved by 2.6%), which indicates that the model has good adaptability to model the learning process of a large number of students.

E. Ablation Experiments

In this section, we conduct some ablation experiments to further show how each module in SLPKT affects final results. The prediction results of these variants are provided in Table III:

- $-$**HSRRNE** (Remove Historical State Review Module and Related Knowledge Enhancement Module): there is no simulation of the learning process after the students get a new exercise.

TABLE III
ABLATION STUDY ON FOUR DATASETS.

Dataset		-HSRRNE	-HSR	-RNE	SLPKT
ASSIST2009	AUC	0.7537	0.7708	0.7612	**0.7813**
	ACC	0.7251	0.7304	0.7366	**0.7387**
ASSIST2012	AUC	0.7454	0.7907	0.7664	**0.8038**
	ACC	0.7316	0.7588	0.7515	**0.7679**
ASSIST2017	AUC	0.7547	0.7839	0.7694	**0.8016**
	ACC	0.7062	0.7279	0.7198	**0.7387**
EDNET1	AUC	0.7139	0.7499	0.7349	**0.7636**
	ACC	0.6946	0.7125	0.7152	**0.7300**

- $-$**HSR** (Remove Historical State Review Module): this variant model can only simulate the process of searching for related knowledge in student's mind and enhance related knowledge in the prediction.
- $-$**RNE** (Remove Related Knowledge Enhancement Module): this variant model can only simulate the process of students reviewing the historical states to mine historical important informance.

Table III shows that SLPKT is superior to all variant models, proving that the modules we added are practical. First, the role of the Related Knowledge Enhancement Module plays a crucial role in SLPKT, and if we removed it, it would lead to the greatest decline in the results. Then, reviewing the historical state in the KT task is more effective than not considering it. Finally, adding either of these two modules will be better than not adding them, which also shows that reasonable modeling of the student's learning process will improve the accuracy of the prediction results.

F. Length Analysis and Knowledge State Visualization

To investigate the effect of different fixed lengths on our model, we evaluate our method's performance on all datasets with four different lengths: 50, 100, 150 and 200, as shown

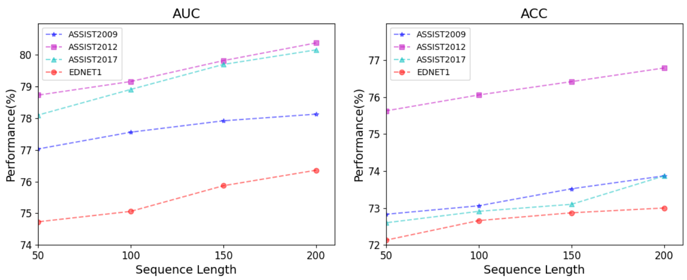

Fig. 3. Comparison results of the influence of sequence lengths on four datasets.

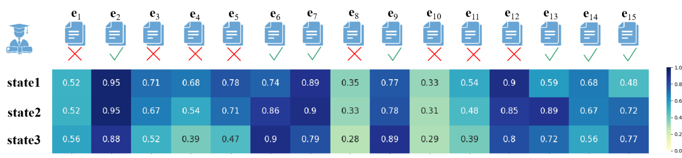

Fig. 4. Student's knowledge state evolution process, in which state1, state2, and state3 are the knowledge state output of the knowledge state acquisition module, the historical state review module, and the related knowledge enhancement module respectively.

in the Fig. 3. Shorter learning sequences of students often determine that the model cannot learn better performance. As the length increases, SLPKT can also maintain good performance.

To show SLPKT has obtained a more reasonable knowledge state, we visualize the knowledge state evolution process, as shown in Fig. 4. The probability in the knowledge state represents the student's mastery of the current exercise and knowledge concept. In addition, correct answers are indicated by 1 and incorrect answers are indicated by 0. If the probability is closer to the actual answer, that means we have obtained a more accurate knowledge state. For the historical state review module, the state2 of $\mathbf{e}_3$-$\mathbf{e}_{12}$ obtain a more accurate knowledge state based on the historical information than state1. For the related knowledge enhancement module, the changes of state3 and state2 in $\mathbf{e}_3$-$\mathbf{e}_5$, $\mathbf{e}_8$, and $\mathbf{e}_{11}$-$\mathbf{e}_{12}$ explain well that we have successfully enhanced the role of related knowledge.

VI. Conclusion

In this paper, we propose SLPKT by modeling the student learning process after they get a new exercise. Compared to the existing KT methods, we consider the students' learning processes after they get a new exercise. To mine historical important information, we trace historical states based on the similarities between the target's knowledge concept and the history exercises' knowledge concepts. And a knowledge enhancement module is used to improve the role of the target exercise's knowledge in performance predicting. We validated the performance of SLPKT on four public datasets and compared it to 9 excellent methods. Experimental results show that our method achieves better performance. In future work, we will further explore the better ways to simulate the learning process to improve the performance.

Acknowledgment

The works described in this paper are supported by The National Natural Science Foundation of China under Grant Nos. 61772210 and U1911201; The Project of Science and Technology in Guangzhou in China under Grant No. 202007040006.

References

[1] Cen, H., Koedinger, K., Junker, B.: Learning factors analysis–a general method for cognitive model evaluation and improvement. In: ITS. pp. 164–175 (2006)

[2] Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. USER MODEL USER-ADAP **4**(4), 253–278 (1994)

[3] Fischer, C., Pardos, Z.A., Baker, R.S., Williams, J.J., Smyth, P., Yu, R., Slater, S., Baker, R., Warschauer, M.: Mining big data in education: Affordances and challenges. Review of Research in Education **44**(1), 130–160 (2020)

[4] Ghosh, A., Heffernan, N., Lan, A.S.: Context-aware attentive knowledge tracing. In: KDD. pp. 2330–2339 (2020)

[5] Jiang, W., Pardos, Z.A., Wei, Q.: Goal-based course recommendation. In: LAK. pp. 36–45 (2019)

[6] Liu, Q., Tong, S., Liu, C., Zhao, H., Chen, E., Ma, H., Wang, S.: Exploiting cognitive structure for adaptive learning. In: KDD. pp. 627–635 (2019)

[7] Long, T., Qin, J., Shen, J., Zhang, W., Xia, W., Tang, R., He, X., Yu, Y.: Improving knowledge tracing with collaborative information. In: WSDM. pp. 599–607 (2022)

[8] Mao, S., Zhan, J., Li, J., Jiang, Y.: Knowledge structure-aware graph-attention networks for knowledge tracing. In: KSEM. Lecture Notes in Computer Science, vol. 13368, pp. 309–321 (2022)

[9] Nagatani, K., Zhang, Q., Sato, M., Chen, Y.Y., Chen, F., Ohkuma, T.: Augmenting knowledge tracing by considering forgetting behavior. In: WWW. pp. 3101–3107 (2019)

[10] Nedungadi, P., Remya, M.: Incorporating forgetting in the personalized, clustered, bayesian knowledge tracing (pc-bkt) model. In: CCIP. pp. 1–5 (2015)

[11] Pandey, S., Karypis, G.: A self-attentive model for knowledge tracing. EDM (2019)

[12] Pavlik, P.I., Cen, H., Koedinger, K.R.: Performance factors analysis - A new alternative to knowledge tracing. In: AIED. vol. 200, pp. 531–538 (2009)

[13] Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L.J., Sohl-Dickstein, J.: Deep knowledge tracing. In: NIPS. pp. 505–513 (2015)

[14] Qiu, Y., Qi, Y., Lu, H., Pardos, Z.A., Heffernan, N.T.: Does time matter? modeling the effect of time with bayesian knowledge tracing. In: EDM. pp. 139–148 (2011)

[15] Shen, S., Liu, Q., Chen, E., Huang, Z., Huang, W., Yin, Y., Su, Y., Wang, S.: Learning process-consistent knowledge tracing. In: KDD. pp. 1452–1460 (2021)

[16] Sheng, D., Yuan, J., Zhang, X.: Grasping or forgetting? MAKT: A dynamic model via multi-head self-attention for knowledge tracing. In: SEKE. pp. 399–404 (2021)

[17] Su, Y., Liu, Q., Liu, Q., Huang, Z., Yin, Y., Chen, E., Ding, C., Wei, S., Hu, G.: Exercise-enhanced sequential modeling for student performance prediction. In: AAAI. vol. 32 (2018)

[18] Wang, F., Liu, Q., Chen, E., Huang, Z., Chen, Y., Yin, Y., Huang, Z., Wang, S.: Neural cognitive diagnosis for intelligent education systems. In: AAAI. vol. 34, pp. 6153–6161 (2020)

[19] Yang, Y., Shen, J., Qu, Y., Liu, Y., Wang, K., Zhu, Y., Zhang, W., Yu, Y.: Gikt: a graph-based interaction model for knowledge tracing. In: ECML/PKDD. pp. 299–315 (2020)

[20] Yu, Y., Huang, C., Chen, L., Chen, M.: Using multi-feature embedding towards accurate knowledge tracing. In: SEKE. pp. 287–292 (2022)

[21] Zhang, J., Shi, X., King, I., Yeung, D.Y.: Dynamic key-value memory networks for knowledge tracing. In: WWW. pp. 765–774 (2017)

[22] Zhou, G., Mou, N., Fan, Y., Pi, Q., Bian, W., Zhou, C., Zhu, X., Gai, K.: Deep interest evolution network for click-through rate prediction. In: AAAI. vol. 33, pp. 5941–5948 (2019)