Learning Process-consistent Knowledge Tracing

Shuanghong Shen[1], Qi Liu[1], Enhong Chen[1,*], Zhenya Huang[1], Wei Huang[1], Yu Yin[1], Yu Su[2], Shijin Wang[2]
[1]Anhui Province Key Laboratory of Big Data Analysis and Application, School of Data Science & School of Computer Science and Technology, University of Science and Technology of China, [2]iFLYTEK Research, iFLYTEK, Co., Ltd
{qiliuql,cheneh,huangzhy}@ustc.edu.cn; {closer,ustc0411,yxonic}@mail.ustc.edu.cn; {yusu,sjwang3}@iflytek.com

ABSTRACT

Knowledge tracing (KT), which aims to trace students' changing knowledge state during their learning process, has improved students' learning efficiency in online learning systems. Recently, KT has attracted much research attention due to its critical significance in education. However, most of the existing KT methods pursue high accuracy of student performance prediction but neglect the consistency of students' changing knowledge state with their learning process. In this paper, we explore a new paradigm for the KT task and propose a novel model named Learning Process-consistent Knowledge Tracing (LPKT), which monitors students' knowledge state through directly modeling their learning process. Specifically, we first formalize the basic learning cell as the tuple *exercise—answer time—answer*. Then, we deeply measure the learning gain as well as its diversity from the difference of the present and previous learning cells, their interval time, and students' related knowledge state. We also design a learning gate to distinguish students' absorptive capacity of knowledge. Besides, we design a forgetting gate to model the decline of students' knowledge over time, which is based on their previous knowledge state, present learning gains, and the interval time. Extensive experimental results on three public datasets demonstrate that LPKT could obtain more reasonable knowledge state in line with the learning process. Moreover, LPKT also outperforms state-of-the-art KT methods on student performance prediction. Our work indicates a potential future research direction for KT, which is of both high interpretability and accuracy.

CCS CONCEPTS

• **Information systems** → *Data mining*; • **Computing methodologies** → **Neural networks**.

KEYWORDS

knowledge tracing, learning process, learning gain, forgetting effect

*Enhong Chen is the corresponding author.

KDD '21, August 14–18, 2021, Virtual Event, Singapore

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467237

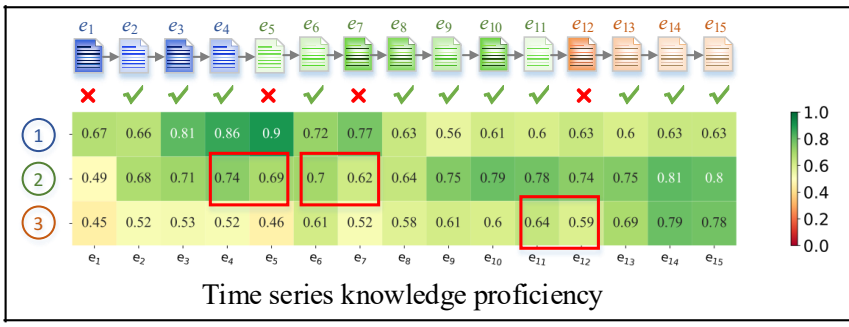
ACM Reference Format:
Shuanghong Shen, Qi Liu, Enhong Chen, Zhenya Huang, Wei Huang, Yu Yin, Yu Su, Shijin Wang. 2021. Learning Process-consistent Knowledge Tracing. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14-18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 9 pages.
https://doi.org/10.1145/3447548.3467237.

1 INTRODUCTION

The recent threats of COVID-19 have triggered the outbreak of online learning [26], whose various forms (such as intelligent tutoring systems and massive open online courses) play indispensable roles in minimizing the disruption to education [25, 32]. Knowledge tracing (KT) [4] is an emerging research area in online learning, which utilizes machine learning sequence models that are capable of using educationally related data to monitor the changing knowledge state of students. In recent decades, KT has been widely applied and received growing attention from the scientific community [17, 30, 34, 37].

Generally, due to the real knowledge state of students at each learning interaction is hard to be recorded and quantified explicitly, their performance on exercises is almost the only way to infer their knowledge state. Therefore, most existing KT models are optimized by minimizing the cross-entropy log loss of the predicted answers and students' actual answers. There is a default assumption that higher accuracy on future performance prediction is approximately be considered as equal to better estimations of knowledge state.

Following the above ideas, existing KT models have achieved impressive results on student performance prediction. For example, Bayesian Knowledge Tracing (BKT) [4], Performance Factor Analysis (PFA) [29], Deep Knowledge Tracing (DKT) [31] and Exercise-aware Knowledge Tracing [16]. However, in our experiments, we have noticed that the only pursuit of high accuracy of future performance prediction could lead to inconsistency between students' knowledge state and their learning process. For better illustration, we give a visualization case of the knowledge state traced by DKT in Figure 1. DKT is a popular model based on RNN or LSTM [10] for the KT task and has achieved impressive performance [2, 31]. In the figure, while the student is answering 15 exercises on 3 different knowledge concepts, DKT keeps tracing his/her knowledge state and depicts the changing process. We find one common but unreasonable observation from the figure, that is once the student has answered wrongly, DKT argues that his/her knowledge state on corresponding knowledge concepts will decline. Although such a downward trend of students' knowledge state after mistakes may bring high accuracy of future performance prediction, it is not in



Exercises	Knowledge Concepts
e_1, e_2, e_3, e_4	①. Absolute Value
$e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}$	②. Addition and Subtraction Integers
$e_{12}, e_{13}, e_{14}, e_{15}$	③. Ordering Integers

Figure 1: A toy example of the evolution process of a student's knowledge states that are traced by DKT, where the student has answered 15 exercises on 3 knowledge concepts. In the left figure, the color of the heatmap or the number in the small box refers to the knowledge state of the student after answering the exercise. The red boxes indicate that DKT thinks the knowledge state will decline after wrong answers. The right table gives the relations between exercises and knowledge concepts.

line with the cognitive theory, because students can also acquire knowledge even if they get wrong answers. Previous research has pointed out that mistakes are seen as natural elements of learning processes [13] and students can learn from errors and foster learning progress through a favorable error climate [35].

To this end, we argue that it is valuable to keep the consistency of students' learning process in knowledge tracing and give equal attention to both right and wrong learning interactions, instead of solely pursuing high accuracy of student performance prediction. In other words, we should focus more on the quality of the knowledge tracing results. In this paper, we explore a new paradigm for the KT task by directly modeling students learning process. However, there are many challenges to be solved along this line. Firstly, how to define the learning process and convert it into a proper form for modeling. Secondly, the learning gain, which represents the knowledge that students acquire in learning, is implicit and changeable in the learning process. Although Mao [20] applied binary Quantized Learning Gain (QLG) to instantiate students' learning gains as *High* or *Low*, such simple instantiation of the learning gain is not enough to capture its diversity. For instance, students may have different learning gains even if they have the same performance on the same exercises. Thirdly, in contrast to the learning gain, students' knowledge will also decrease over time, which commonly manifests as forgetting, is also necessary to be considered in the KT task.

To conquer the above challenges, we propose a novel method named Learning Process-consistent Knowledge Tracing (LPKT), which reaches our goal to assess the knowledge state of students by modeling their learning process. Specifically, as the learning process can be seen as the learning sequence of students in a timeline, we first define the basic learning cell in the learning process as a tuple *exercise—answer time—answer* and adjacent cells are separated by the interval time. Notably, the learning cell is different from the learning interaction (i.e., *exercise—answer*) in that the former contains the time that students spent on answering the exercise. Therefore, the learning cell is more capable to reflect the complete learning process. Then, for measuring the learning gain, which broadly represents the difference of students' performance at two points in time, we compute them directly from the difference of present and previous learning cells. Besides, to capture the diversity of learning gains, we also model another two factors, which are the interval time between two continuous learning cells and students' related knowledge state respectively. In LPKT, the learning gain is set to be always positive, so that students can consistently get knowledge at each learning interaction. Furthermore, considering that not all learning gains can be transformed as the growth of students' knowledge, we design a learning gate in LPKT for controlling students' absorptive capacity of knowledge. Finally, for the common phenomenon of forgetting in the learning process, we design a forgetting gate in LPKT to determine the decrease of knowledge state over time. Therefore, LPKT realizes a novel way to assess students' knowledge state by modeling their learning process. Extensive experiments on three public real-world datasets demonstrate that LPKT gets more reasonable knowledge state in line with students' cognitive process. Moreover, LPKT can also significantly outperform existing KT models on student performance prediction. Our idea to solve the KT problem by modeling students' learning process indicates a potential future research direction, which is of both high interpretability and high accuracy. We also conduct a case study to show that LPKT can learn meaningful representations of exercises automatically.

2 RELATED WORKS

2.1 Knowledge tracing

Most of the existing knowledge tracing models can be classified into traditional probabilistic models, logistic models, and deep learning-based methods. Bayesian Knowledge Tracing (BKT) [4] is a classic and widely-used probabilistic model for KT, which can be seen as a special case of the Hidden Markov Model (HMM). Logistic models are a large class of models based on logistic functions, which utilizes a logistic function to estimate the probability of knowledge state [30], such as Performance Factor Analysis (PFA) [29]. DKT introduces deep learning into KT for the first time [31]. DKT takes the learning sequence as the input of RNN or its variant LSTM and represents student knowledge states by the hidden states. Dynamic Key-Value Memory Networks (DKVMN) [39] introduces memory-augmented neural networks into KT. It defines a static matrix called *key* to store latent knowledge concepts and a dynamic matrix called *value* to store and update the knowledge mastery [39]. Exercise-aware Knowledge Tracing [16] introduces text contents to enhance the performance on the KT task. Convolutional knowledge tracing (CKT) [33] applies the convolutional windows to model students'

individualized learning rates within several continuous learning interactions. The self-attentive model for knowledge tracing (SAKT) [27] presents transformer to knowledge tracing directly for the first time. Pandey and Srivastava [28] presents a relation-aware self-attention layer that incorporates the contextual information for knowledge tracing. Ghosh et al. [6] presents a context-aware attentive knowledge tracing (AKT) model for KT, which utilizes contextualized representations of both exercises and knowledge acquisitions and incorporates attention mechanism with cognitive and psychometric models.

2.2 Learning Gain

The learning gain broadly means the difference between the skills, competencies, content knowledge, and personal development at two points in time [22]. Learning gain is different from learning outcomes in that learning gain compares performance at two points in time, while learning outcome concentrates on the output level at a single point in time. For example, students may not benefit from the exercise even if he/she performs well on it. Luckin et al. [19] calculated learning gain as LG = $post - pre$, where pre and post refer to a student's pre-test and post-test scores. Normalized Learning Gain (NLG) [8] is a widely used adjusted measurement: NLG = $\frac{post-pre}{1-pre}$, where 1 is the maximum score for pre- and post-tests. However, NLG can be problematic in certain circumstances, such as even a small decline in post-test score from the pre-test can result in a large negative in NLG if the student has high pretest scores. Mao [20] proposes a qualitative measurement called Quantized Learning Gain (QLG), which is a binary qualitative measurement on students' learning gains from pretest to the posttest: *High* or *Low*. They first split students into three groups based on their scores. Then, if a student moves from a lower performance group to a higher performance group, he/she is a *High* QLG. On the contrary, he/she will be a *Low* QLG. But such simple instantiation of the learning gain is still not enough to capture its diversity.

2.3 Forgetting Effect

In a real learning environment, forgetting is inevitable [21]. The *Ebbinghaus forgetting curve theory* indicates that students' knowledge proficiency may decline due to the forgetting factor [11]. Nedungadi and Remya [24] incorporate forgetting based on the assumption that the learned knowledge decays exponentially over time [18]. They utilize an exponential decay function to update the knowledge mastery level. Huang et al. [11] proposes the Knowledge Proficiency Tracing (KPT) model to model students' knowledge proficiency with both learning and forgetting theories, which dynamically captures the change of students' proficiency level over time. Nagatani et al. [23] make attempts to improve DKT by considering forgetting effects, but they only extend DKT by incorporating multiple types of time or counts information.

3 PRELIMINARY

In this section, we formalize the learning process of students and give a brief introduction to the definition of knowledge tracing. Besides, we also present some important embeddings in LPKT.

3.1 Problem Definition

In an intelligent tutoring system, supposing there are the set of students $S = \{s_1, s_2, ..., s_i, ..., s_I\}$, the set of exercises $E = \{e_1, e_2, ..., e_j, ..., e_J\}$, and the set of knowledge concepts $K = \{k_1, k_2, ..., k_m, ..., k_M\}$, where each exercise is related to specific knowledge concepts. The Q-matrix $Q \in \mathbb{R}^{J \times M}$, which is consisted of zeros and ones, indicates the relationship between exercises and knowledge concepts, where $Q_{jm} = 1$ if knowledge concept k_m is required for exercise e_j and $Q_{jm} = 0$ otherwise. Generally, when an exercise is assigned to the student, he/she spends a certain time on answering it according to his/her learned knowledge. The learning process keeps repeating the above answering behavior on different exercises, where there is an interval time between adjacent answering interaactions. Therefore, we denote the learning process of a student as $\mathrm{x} = \{(e_1, at_1, a_1), it_1, (e_2, at_2, a_2), it_2, ..., (e_t, at_t, a_t), it_t\}$, where the tuple (e_t, at_t, a_t) represents a basic learning cell in learning process, e_t is the exercise, at_t is the answer time the student spent on answering e_t, and a_t represents the binary correctness label (1 represents correct and 0 for wrong), it_t stands for the interval time between the learning cells.

Problem Definition. *Given students' learning sequence* $\mathrm{x} = \{(e_1, at_1, a_1), it_1, (e_2, at_2, a_2), it_2, ..., (e_t, at_t, a_t), it_t\}$*, the KT task aims to monitor students' changing knowledge state during the learning process and predict their future performance at the next learning step* $t + 1$*, which can be further applied to individualize students' learning scheme and maximize their learning efficiency.*

3.2 Embeddings

In LPKT, to realize our goal of modeling students' learning process, we consider the following elements: exercises, answer time, answers, interval time, knowledge concepts, and knowledge state. We define the basic cell of the learning process as a tuple *exercise—answer time—answer* and each learning cell is separated by the interval time. To better understanding the whole structure of LPKT before presenting its details, we give a simple introduction to the embeddings of those elements from three categories as below.

3.2.1 Time Embedding. Time embedding refers to the embedding of answer time and interval time. Generally, the answer time and interval time are both important elements in the learning process, which can influence the learning gain and forgetting effects of students to some degree. Nagatani et al. [23] discretized all time features by minutes at $\log_2$ scale and represented them as one-hot vectors. *Forgetting curve theory* was also introduced to model the decreasing knowledge state of students as time goes on [11, 18]. In LPKT, due to the interval time could be much longer than the answer time, we discretize the former by the minutes and the latter by the seconds. Besides, we set all the interval time longer than one month as one month. Then, we represent the discretized answer time by an embedding matrix $\boldsymbol{at} \in \mathbb{R}^{d_{at} \times d_k}$, the discretized interval time is similarly represented by an embedding matrix $\boldsymbol{it} \in \mathbb{R}^{d_{it} \times d_k}$, where d_{at} and d_{it} are the number of the discretized answer time and interval time respectively. Then, at_t and it_t in learning interaction x_t will be represented as the vector $\boldsymbol{at}_t \in \mathbb{R}^{d_k}$ and $\boldsymbol{it}_t \in \mathbb{R}^{d_k}$.

3.2.2 Learning Embedding. Learning embedding is the embedding of the basic learning cell, which is the main part of students' learning

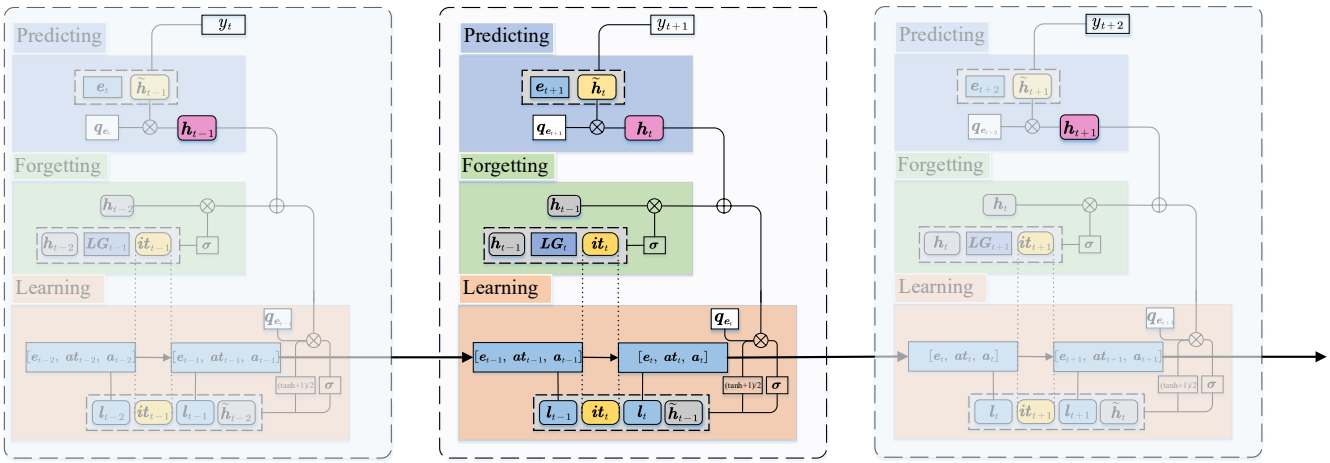


Figure 2: The architecture of the LPKT model.

process and characterizes the knowledge them acquire by answering exercises. We first represent the exercise set by an embedding matrix $\boldsymbol{E} \in \mathbb{R}^{J \times d_e}$, where J is the number of exercises and d_e is the dimension. Then each exercise e_t in learning cell x_t will be represented as the vector $\boldsymbol{e_t} \in \mathbb{R}^{d_e}$. For the answer a_t, i.e., 0 or 1, we expand it to a all-zero or all-one vector $\boldsymbol{a_t} \in \mathbb{R}^{d_a}$, d_a is the dimension as well. Finally, for getting the learning embedding $\boldsymbol{l_t} \in \mathbb{R}^{d_k}$ of the basic learning cell (e_t, at_t, a_t), we concatenate $\boldsymbol{e_t}$, $\boldsymbol{at_t}$, and $\boldsymbol{a_t}$ together and apply a multi-layer perceptron (MLP) to deeply fuse the exercise embeddings, answer time embeddings, and answer embeddings as follows:

$$\boldsymbol{l_t} = \boldsymbol{W}_1^T[\boldsymbol{e_t} \oplus \boldsymbol{at_t} \oplus \boldsymbol{a_t}] + \boldsymbol{b}_1, \tag{1}$$

where $\oplus$ is the operation of concatenating, $\boldsymbol{W}_1 \in \mathbb{R}^{(d_e+d_k+d_a)\times d_k}$ is the weight matrix, $\boldsymbol{b}_1 \in \mathbb{R}^{d_k}$ is the bias term, d_k is the dimension.

3.2.3 Knowledge Embedding. Knowledge embedding is served to store and update the knowledge state of students during the learning process. In LPKT, the knowledge embedding is initialized as an embedding matrix $\boldsymbol{h} \in \mathbb{R}^{M \times d_k}$, where M is the number of knowledge concepts. Therefore, each row of the matrix $\boldsymbol{h}$ represents the knowledge mastery of the corresponding knowledge concept. At each learning interaction, the learning gain on each knowledge concept modeled by LPKT is updated into the knowledge embedding, the forgetting effects are also included in it simultaneously.

The Q-matrix indicates the relations between exercises and knowledge concepts, which controls the updated row in the knowledge embedding after answering related exercises. For instance, after answering the exercise e_j with knowledge concept k_m, the row m of the student's knowledge matrix will be updated. Traditionally, if the knowledge concept k_m is not contained in the exercise e_j, Q_{jm} will be set as 0, which shows students' performance on exercise e_j has no influence on their knowledge mastery $\boldsymbol{h}_m$ on knowledge concept k_m. However, manually-labeled Q-matrix may be deficient because of inevitable errors and subjective bias [15, 36]. In order to make up for possible omissions or mistakes, we define an enhanced Q-matrix $\boldsymbol{q} \in \mathbb{R}^{J \times M}$, where $\boldsymbol{q}_{jm}$ will be set as a small positive value γ rather than 0 even if k_m is not in e_j. Although this unified setting is simple as well, as the focus of our paper is on the learning process modeling part, we leave the exploration to learn the specific weights in the Q-matrix as future works.

4 THE LPKT MODEL

In this section, we present the LPKT model in detail. As shown in Figure 2, LPKT is consisted of three modules at each learning step: (1) learning module, (2) forgetting module, and (3) predicting module. Specifically, after a student has answered an exercise, the learning module models his/her learning gains compared with the previous learning interaction. The forgetting module is utilized to measure how much knowledge will be forgotten as time goes on. Then, the learning gains and forgotten knowledge will be taken advantage of to update the student's previous knowledge state for achieving their latest knowledge state. Finally, the predicting module is proposed to predict the student's performance on the next exercise according to his/her latest knowledge state.

4.1 Learning Module

As mentioned in our primary goal to model the learning process for the KT task, after formalizing the learning process as the basic learning cell and the interval time, the next problem is to measure the implicit and dynamic learning gain. Traditionally speaking, the learning gain can be defined as 'distance traveled' [22], which stands for the difference of students' performance at two points in time. For modeling learning gains precisely, we should consider the differences in students' performance at two continuous learning interactions of students. In LPKT, we realize the modeling of learning gain through concatenating students' previous learning embedding $\boldsymbol{l}_{t-1}$ and present learning embedding $\boldsymbol{l}_t$ as the basic input element in LPKT. However, although we can capture the differences in students' performance with two continuous learning embeddings, it is unable to capture the diversity of learning gains in the learning process. For example, not all students share the same learning gains even if they have the same performance on part of overlapped learning sequences (i.e., the same continuous learning embeddings). Therefore, we consider two influencing factors of the learning gains in LPKT, which are the interval time and students' previous knowledge state respectively. On the one hand, the interval time between two learning cells is a key element in learning process, which can

reflect the distinctions of learning gains. Generally, students tend to acquire more knowledge with shorter interval time, which means their learning process is compact and continuous. On the other hand, the previous knowledge state can also influence students' learning gains, such as students with worse mastery have greater possibilities of improvement. Therefore, we incorporate the above two factors into LPKT for modeling the evolution of learning gains. Specifically, for the interval time, we concatenate $\boldsymbol{it}_t$ into the basic input element in the timeline between the two continuous learning embeddings. For previous knowledge state, to focus on the knowledge state on the related knowledge concepts of the present exercise, we first multiply $\boldsymbol{h}_{t-1}$ and the knowledge concept vector $\boldsymbol{q}_{e_t}$ of present exercise and get the related knowledge state $\widetilde{\boldsymbol{h}}_{t-1}$:

$$\widetilde{\boldsymbol{h}}_{t-1} = \boldsymbol{q}_{e_t} \cdot \boldsymbol{h}_{t-1}, \tag{2}$$

where $\cdot$ denotes the element-wise product between vectors. Then the learning gains $\boldsymbol{lg}_t$ will be modeled as follows:

$$\boldsymbol{lg}_t = tanh(\boldsymbol{W}_2^T[\boldsymbol{l}_{t-1} \oplus \boldsymbol{it}_t \oplus \boldsymbol{l}_t \oplus \widetilde{\boldsymbol{h}}_{t-1}] + \boldsymbol{b}_2), \tag{3}$$

where $\boldsymbol{W}_2 \in \mathbb{R}^{(4d_k)\times d_k}$ is the weight matrix, $\boldsymbol{b}_2 \in \mathbb{R}^{d_k}$ is the bias term, $tanh$ is the non-linear activation function.

Considering that not all learning gains can be transformed into the growth of students' knowledge completely, we further design a learning gate $\boldsymbol{\Gamma}_t^l$ to control the students' absorptive capacity of knowledge:

$$\boldsymbol{\Gamma}_t^l = \sigma(\boldsymbol{W}_3^T[\boldsymbol{l}_{t-1} \oplus \boldsymbol{it}_t \oplus \boldsymbol{l}_t \oplus \widetilde{\boldsymbol{h}}_{t-1}] + \boldsymbol{b}_3), \tag{4}$$

where $\boldsymbol{W}_3 \in \mathbb{R}^{(4d_k)\times d_k}$ is the weight matrix, $\boldsymbol{b}_3 \in \mathbb{R}^{d_k}$ is the bias term, σ is the non-linear *sigmoid* activation function.

Then $\boldsymbol{\Gamma}_t^l$ will be multipied to $\boldsymbol{lg}_t$ to get the actual learning gains $\boldsymbol{LG}_t$ in the $t-$th learning interaction. Similarly, to focus on the learning gain of the related knowledge concepts of exercise e_t, we multiply $\boldsymbol{LG}_t$ by $\boldsymbol{q}_{e_t}$ to get the related learning gains $\widetilde{\boldsymbol{LG}}_{t-1}$:

$$\begin{aligned} \boldsymbol{LG}_t &= \boldsymbol{\Gamma}_t^l \cdot ((\boldsymbol{lg}_t + 1)/2), \\ \widetilde{\boldsymbol{LG}}_t &= \boldsymbol{q}_{e_t} \cdot \boldsymbol{LG}_t, \end{aligned} \tag{5}$$

due to the output range of $tanh$ function is $(-1, 1)$, we apply a linear transformation $((\boldsymbol{lg}_t + 1)/2$ to project the range of $\boldsymbol{lg}_t$ from $(-1, 1)$ to $(0, 1)$. Therefore, the learning gains $\boldsymbol{LG}_t$ will be always positive, which is in line with our assumption that students' can consistently acquire knowledge at each learning interaction.

4.2 Forgetting Module

After computing $\widetilde{\boldsymbol{LG}}_t$, which plays an enhanced role in students' knowledge state, the opposite forgetting phenomenon affects how much knowledge will be forgotten as time goes on. According to the *forgetting curve theory* [18], the amount of learned material that is remembered decays exponentially over time. Nevertheless, a simple manual-designed exponential decay function is not sufficient for capturing complex relations between knowledge state and interval time. For modeling the complex forgetting effects, we design a forgetting gate $\boldsymbol{\Gamma}_t^f$ in LPKT, which applies a MLP to learn the degree of loss information in knowledge matrix based on three factors: (1) students' previous knowledge state $\boldsymbol{h}_{t-1}$, (2) students' present learning gains $\boldsymbol{LG}_t$, and (3) interval time $\boldsymbol{it}_t$:

$$\boldsymbol{\Gamma}_t^f = \sigma(\boldsymbol{W}_4^T[\boldsymbol{h}_{t-1} \oplus \boldsymbol{LG}_t \oplus \boldsymbol{it}_t]) + \boldsymbol{b}_4), \tag{6}$$

where $\boldsymbol{W}_4 \in \mathbb{R}^{(3d_k)\times d_k}$ is the weight matrix, $\boldsymbol{b}_4 \in \mathbb{R}^{d_k}$ is the bias term, σ is the non-linear *sigmoid* activation function.

Then, we can eliminate the influence of forgetting by multiplying $\boldsymbol{\Gamma}_t^f$ to $\boldsymbol{h}_{t-1}$ and the knowledge state $\boldsymbol{h}_t$ after students accomplish the $t-$th learning interaction will be updated as follows:

$$\boldsymbol{h}_t = \widetilde{\boldsymbol{LG}}_t + \boldsymbol{\Gamma}_t^f \cdot \boldsymbol{h}_{t-1}. \tag{7}$$

4.3 Predicting Module

Through modeling the learning gain and forgetting effect in the learning process, we have got students' knowledge state $\boldsymbol{h}_t$ after the $t-$th learning interaction. In this part, we will use $\boldsymbol{h}_t$ to predict students' performance on the next exercise e_{t+1}.

In a real learning environment, given $\boldsymbol{e}_{t+1}$ to the student, he/she will try to solve it by applying his/her knowledge to the corresponding knowledge concepts. Therefore, we use the related knowledge state $\widetilde{\boldsymbol{h}}_t$ to infer the student's performance on $\boldsymbol{e}_{t+1}$. We first concatenate the exercise embedding $\widetilde{\boldsymbol{h}}_t$ and $\boldsymbol{e}_{t+1}$, then project them to the output layer by a fully connected network with sigmoid activation:

$$y_{t+1} = \sigma(\boldsymbol{W}_5^T[\boldsymbol{e}_{t+1} \oplus \widetilde{\boldsymbol{h}}_t] + \boldsymbol{b}_5), \tag{8}$$

where $\boldsymbol{W}_5 \in \mathbb{R}^{(2d_k)\times d_k}$ is the weight matrix, $\boldsymbol{b}_5 \in \mathbb{R}^{d_k}$ is the bias term. The output y_{t+1}, which is the range of $(0, 1)$, represents the predicted performance of the student on next exercise e_{t+1}. We can further set a threshold to determine whether the student can answer e_{t+1} correctly, that is he/she can get right answer if y_{t+1} is greater than the threshold, otherwise, the answer is wrong.

4.4 Objective Function

To learn all parameters in LPKT, we also choose the cross-entropy log loss between the prediction y and actual answer a as the objective function:

$$\mathbb{L}(\theta) = -\sum_{t=1}^{T}(a_t \log y_t + (1 - a_t)\log(1 - y_t)) + \lambda_\theta ||\theta||^2, \tag{9}$$

where θ denotes all parameters of LPKT and λ_θ is the regularization hyperparameter. The objective function was minimized using Adam optimizer [12] on mini-batches. More details of settings will be specified in the part of experiments.

5 EXPERIMENTS

In this section, we first describe the real-world datasets used in the experiments. Then we conduct experiments with the aim of answering the following research questions:

- **RQ1** Does our proposed LPKT model keep the consistency of students' changing knowledge state to their learning process?
- **RQ2** Does our proposed LPKT model outperform the state-of-the-art knowledge model on student performance prediction?
- **RQ3** How does the learning module, forgetting module, and time information in LPKT impact the knowledge tracing result?
- **RQ4** Can LPKT learn meaningful representations of exercises?

Statistics	Datasets		
	ASSIST2012	ASSISTchall	EdNet-KT1
Students	29,018	1,709	784,309
Exercises	53,091	3,162	12,372
Concepts	265	102	141
Answer Time	26,747	1,326	9,292
Interval Time	29,748	2,839	41,830
Avg.length	93.45	551.68	121.48

Table 1: Statistics of all datasets.

5.1 Datasets

Three real-world public datasets have been used to evaluate the effectiveness of LPKT. Table 1 shows the statistics of all datasets. A simple description of all datasets is listed as follows:

- **ASSISTments 2012**[1] **(ASSIST2012)** is collected from the ASSISTments [5], an online tutoring system created in 2004. The data is gathered from skill builder problem sets where students need to work on similar exercises to achieve mastery, which contains data for the school year 2012-2013 with affect predictions. We have filtered the records without knowledge concepts.
- **ASSISTments Challenge**[2] **(ASSISTChall)** is utilized in the 2017 ASSISTments data mining competition. Researchers collected it from a longitudinal study, which tracks students from their use of ASSISTments blended learning platform in middle school in 2004-2007. In this dataset, students have a much longer learning sequence than ASSIST2012.
- **EdNet-KT1**[3] is the dataset of all student-system interactions collected over 2 years by Santa, a multi-platform AI tutoring service with more than 780K users in Korea available through Android, iOS and web [3]. To provide various kinds of actions in a consistent and organized manner, EdNet offers the datasets in four different levels of abstraction. In this paper, we use its simplest form, i.e., EdNet-KT1, which consists of students' exercise-solving logs. If the exercise has more than one knowledge concepts, we only use the first knowledge concept as its knowledge concept.

5.2 Training Details

Preprocessing. We first sorted all learning records of the student by the timestamp of answering. Then, we set all input sequences to a fixed length based on the average sequence length of the dataset. Specifically, for datasets ASSIST2012 and EdNet-KT1, we set the fixed lengths to be 100. For ASSISTchall, the fixed lengths were set as 500. For sequences longer than the fixed length, we cut them into several unique sub-sequences according to the fixed length. For the sequences shorter than the fixed length, zero vectors were used to pad them up to the fixed length.

Training setting. For all datasets, we performed standard 5-fold cross-validation for all models. Thus, for each fold, 80% of the students were split as the training set (80%) and validation set (20%), the rest 20% were used as the testing set. To set up the training process, we randomly initialize all parameters in the uniform distribution [7]. All the hyper-parameters are learned on the training set, and the model that performed best on validation set was used to evaluate the testing set. In LPKT, we added a dropout layer [9] with a dropout rate of 0.2 to prevent overfitting. Parameter d_k, d_e are all set to be 128 and d_a is 50 in our implementation. The small positive value γ in the enhanced Q-matrix $\boldsymbol{q}$ is 0.03. Our code is available at https://github.com/bigdata-ustc/EduKTM.

[1]https://sites.google.com/site/assistmentsdata/home/2012-13-school-data-with-affect
[2]https://sites.google.com/view/assistmentsdatamining/dataset
[3]http://ednet-leaderboard.s3-website-ap-northeast-1.amazonaws.com/

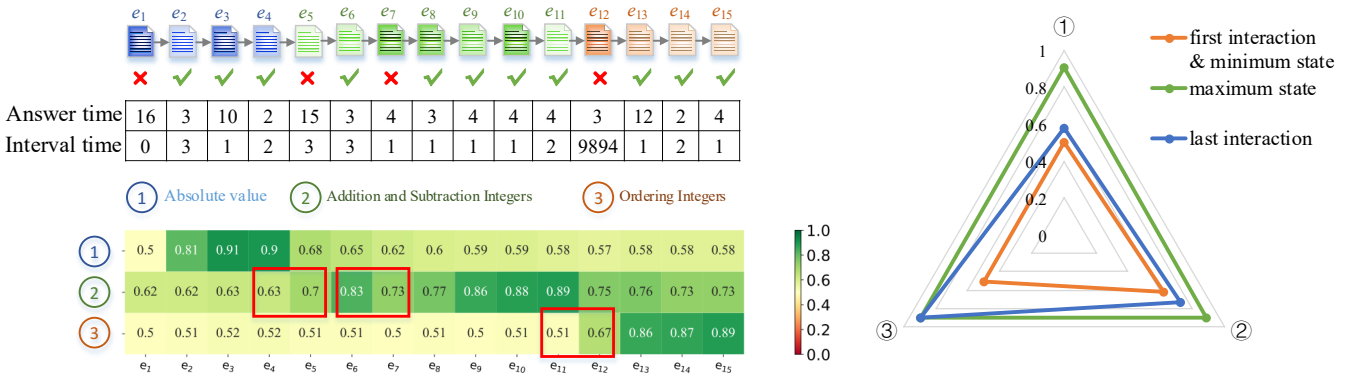
5.3 Baseline Methods

We compare LPKT with several previous methods. For a fair comparison, all these methods are tuned to have the best performances. All models are implemented by Tensorflow [1], and trained on a cluster of Linux servers with TITAN V100 GPUs. The details of comparison methods are:

- **DKT** leverages recurrent neural network to assess student knowledge state [31]. We utilized LSTM in our implementation.
- **DKT+** is an extended variant of DKT [38], which attempts to solve two major problems in DKT. The first problem is that DKT fails to reconstruct the observed input and the second one is the predicted performance of DKT across time-steps is not consistent.
- **DKVMN** takes advantage of memory network to get interpretable student knowledge state [39]. It defines a static matrix called key matrix to store latent knowledge concepts and a dynamic matrix called $value$ matrix to store and update the corresponding knowledge state through read and write operations over time.
- **SAKT** applies the transformer structure to the KT task [27]. It proposes a self-attentive model for knowledge tracing.
- **CKT** introduces convolutional windows in CNN to model the individualized learning rate of students in learning process [33].
- **AKT** is the context-aware attentive knowledge tracing model [6]. It uses the two self-attentive encoders to learn context-aware representations of the exercises and answers. The knowledge evolution model is referred to the knowledge retriever, which uses an attention mechanism to retrieve knowledge acquired in the past that is relevant to the current exercise.

5.4 Knowledge State Visualization (RQ1)

As our primary goal is to model students' learning process for KT, we will show that LPKT can capture reasonable knowledge state of students, which is inconsistent with their learning process. Figure 3 shows the changing knowledge state traced by LPKT of the same student in Figure 1. There are several important observations in the figure. Firstly, our proposed LPKT method can capture the student's learning gains from both wrong and right learning interactions. For example, even the student answer exercise e_5 and e_{12} wrongly, LPKT thinks his/her knowledge state on related knowledge concepts (i.e., *Addition and Subtraction Intergers* and *Ordering Intergers*) can also get promotion. We note that after answering exercise e_7 wrongly, his/her knowledge state is reductive, the reason is that his/her performance on *Addition and Subtraction Intergers* is not stable in this stage and LPKT is trying to modify his/her knowledge state. Secondly, if the student does not practice on some knowledge concepts, his/her knowledge state on these concepts will reduce gradually as time goes on. For instance, the student's



(a) Time series knowledge proficiency (b) Radar diagram

Figure 3: The evolution process of a student's (the same student in Figure 1) knowledge states traced by LPKT. In sub-figure (a), the top part indicates his/her performance at each time step, the answer time and interval time. Sub-figure (b) is the radar diagram of the student's knowledge state at the first interaction and the last interaction, his/her maximum and minimum knowledge state in learning process are also depicted on it.

Methods	ASSIST2012				ASSISTchall				EdNet-KT1			
	RMSE	AUC	ACC	r^2	RMSE	AUC	ACC	r^2	RMSE	AUC	ACC	r^2
DKT	0.4241	0.7289	0.7360	0.1468	0.4471	0.7213	0.6907	0.1425	0.4508	0.6836	0.6889	0.1008
DKT+	0.4239	0.7295	0.7254	0.1497	0.4502	0.7101	0.6842	0.1308	0.4601	0.6429	0.6733	0.0635
DKVMN	0.4261	0.7228	0.7329	0.1398	0.4503	0.7108	0.6842	0.1302	0.4538	0.6741	0.6843	0. 0913
SAKT	0.4258	0.7233	0.7339	0.1403	0.4626	0.6605	0.6694	0.0822	0.4524	0.6794	0.6862	0.0964
CKT	0.4234	0.7310	0.7365	0.1497	0.4455	0.7263	0.6924	0.1488	0.4519	0.6811	0.6871	0.0984
AKT	0.4100	0.7740	0.7554	0.2035	0.4317	0.7655	0.7141	0.2015	0. 4241	0. 7701	0.7287	0.2059
LPKT	**0.4069**	**0.7772**	**0.7583**	**0.2145**	**0.4153**	**0.8008**	**0.7424**	**0.2609**	**0.4234**	**0.7721**	**0.7300**	**0.2085**

Table 2: Results of comparison methods on student performance prediction. LPKT outperforms all baselines on all datasets.

Methods	learning	forgetting	time	RMSE	AUC	ACC	r^2
LPKT-L	✓		✓	0.4112	0.7659	0.7531	0.1980
LPKT-F		✓	✓	0.4087	0.7734	0.7554	0.2075
LPKT(no time)	✓	✓		0.4077	0.7759	0.7571	0.2115
LPKT	✓	✓	✓	**0.4069**	**0.7772**	**0.7583**	**0.2145**

Table 3: Results of ablation experiments on ASSIST2012.

knowledge state on *Absolute Value* and *Addition and Subtraction Intergers* is dropping by degrees after answering exercise e_4 and e_{11} respectively. Thirdly, the general changing process of the student's knowledge state is consistent with his/her learning process. At the first learning interaction, his/her knowledge state is the minimum. During the learning process, the student keeps absorbing new knowledge and his/her knowledge state achieves the maximum, which can be reflected by the increased areas of the radar diagram that indicates the student's knowledge proficiency. At the last learning interaction, the student's knowledge state presents a certain degree of reduction in comparison with the maximum but is still better than the beginning.

5.5 Student Performance Prediction (RQ2)

Although our goal for proposing LPKT is to get more reasonable KT, the experimental results on student performance prediction are still one of the most important metrics for evaluating KT methods. Therefore, we compare LPKT with all baselines on student performance prediction and report the average results across five test folds in Table 2. In order to evaluate the performance of all models comprehensively, we conduct extensive experiments on all datasets. For providing robust evaluation results, the performance was evaluated in terms of Root Mean Squared Error (RMSE), Area Under Curve (AUC), Accuracy (ACC), and the square of Pearson correlation (r^2) in all experiments. From Table 2, we can see that LPKT outperforms all other KT methods on all datasets and metrics, which indicates better results in line with students' learning process are positively related to predicting their future performance more accurately. In addition, there are also some other important observations. First, we have noticed that LPKT significantly outperforms (i.e., improves the AUC by 4.6%) state-of-the-art AKT model on the ASSISTchall dataset, which suggests that LPKT is more capable of capturing students' historical learning information in long sequences. Second, in the biggest EdNet-KT1 dataset, which contains the learning records of 784,309 students, LPKT also marginally outperforms other methods, which indicates that the modeling of learning process in LPKT has good adaptability for a large number of students.

5.6 Ablation Experiments (RQ3)

In this section, we conduct some ablation experiments to further show how each module in LPKT affects final results. In Table 3,

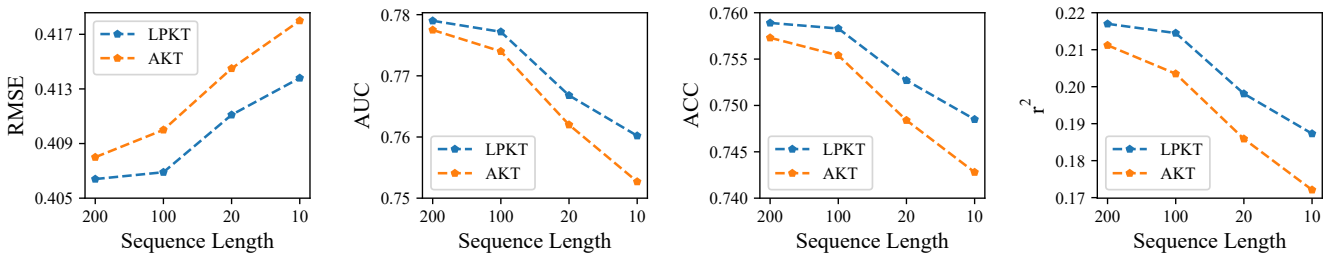
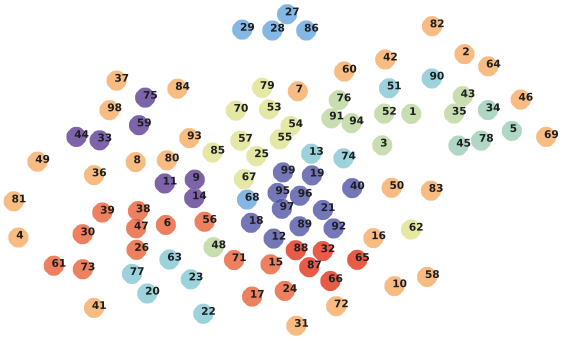


Figure 4: Comparison results of the influence of learning sequence length of LPKT and AKT on ASSISTchall.



(a) Exercises clustering results. Randomly selected 100 exercises in the 3162 exercises of ASSISTchall are clustered into ten concepts. Exercises under the same concept are labeled in the same color and the number stands for the index of exercises.

Knowledge Concepts	Index of Exercises	Knowledge Concepts	Index of Exercises
pattern-finding	13, 17, 18, 19, 20, 22, 28, 29, 30, 31, 32, 91, 92, 93, 94, 99, 100	equivalent-fractions-decimals-percents	47, 49, 51, 52
square-root	33, 34, 35, 36, 37, 38, 39, 40, 41	subtraction	62, 87, 88, 89, 90, 95, 96, 97, 98
symbolization-articulation	7, 8, 9, 10	transformations-rotations	23, 24
point-plotting	1, 2, 3, 5, 25, 26, 27	reading-graph	4, 14, 15, 16
supplementary-angles	58, 73, 74	inducing-functions	11, 12, 21, 42, 43, 44, 45, 46, 64, 69, 71
evaluating-functions	53, 54, 55, 56, 57, 63	addition	65, 66, 67, 68
transversals	59, 72	isosceles-triangle	60, 61
equation-solving	70, 82, 85, 86	equation-concept	6
interpreting-numberline	79, 80, 81	venn-diagram	75, 76
percent-of	77, 78	percents	83, 84
of-means-multiply	48	rate	50

(b) Manually-labeled knowledge concepts of the above 100 exercises.

Figure 5: Exercises clustering.

there are three variations of LPKT, each of which takes out one module from the full LPKT. Concretely, LPKT-L refers to LPKT without considering forgetting, i.e., the forgetting gate is removed. LPKT-F refers to LPKT without modeling learning gains, where the basic input element in LPKT is replaced by a single learning embedding, instead of two continuous learning embeddings. Therefore, LPKT-L can only measure students' learning outcomes, rather than learning gains. LPKT (no time) refers to LPKT that does not utilize any time information, i.e., the answer time and interval time are dropped. The result in Table 3 shows some interesting conclusions. First, the common phenomenon of forgetting plays a critical role in learning process, which can cause the biggest decline of the predictive results if we do not consider it. Second, modeling the learning gain indeed performs better than modeling only the learning outcomes in knowledge tracing, because the learning gain can better reflect the dynamic changes of students' knowledge state. Third, the answer time and interval time are essential and necessary information in the whole learning process, which is harmful to accurately model the learning process if omitted.

Moreover, we also conduct experiments to evaluate that if LPKT can better model students' learning process than the state-of-the-art KT method. Generally, a longer learning sequence represents a more complete learning process. Therefore, we compare the results of LPKT and state-of-the-art AKT on student performance prediction under different learning sequence lengths in dataset ASSISTchall. Figure 4 indicates the comparison results. Specifically, we set four different lengths: 200, 100, 20, and 10, respectively. The shorter the learning sequence, the more incomplete the learning process. From Figure 4, we can see that the gap between LPKT and AKT becomes wider (i.e., the reduction of experimental results of LPKT is less than AKT) as the learning sequence is going shorter. This observation demonstrates that LPKT is less affected by incomplete learning sequences so that LPKT indeed better models students' learning process. In real learning environments, it is hard to get access to the complete learning sequences of students, therefore LPKT has more potential application values due to its robustness.

5.7 Exercises Clustering (RQ4)

In LPKT, the embeddings of exercises are randomly initialized. As LPKT can get students' knowledge state with high interpretability and high accuracy, the learned embeddings of exercises should also show some meanings after training. In Figure 5, we randomly choose 100 exercises among the 3162 exercises in dataset ASSISTchall and visualize the embeddings of these exercises utilizing the T-SNE method [14]. As shown in Figure 5, we can see that the learned embeddings of exercises in LPKT can be split into 10 concepts and the clustering results show well meanings. For example, exercises 89, 95, 96, 97 with same concept *subtraction* are split together and exercises 53, 54, 55, 57 with same concept *evaluating-functions* are also in the same cluster. Although not all the clustering results are correct, these automatically learned representations of exercises can serve as meaningful supplements for the educational experts.

6 CONCLUSIONS AND FUTURE WORKS

In this paper, we explored a new paradigm for knowledge tracing through modeling students' learning process and presented a novel model named Learning Process-consistent Knowledge tracing (LPKT). Specifically, we first formalized the learning process as

the basic learning cell and interval time, where the former was the tuple *exercise—answer time—answer*. Then we modeled the learning gain in learning process by capturing the difference in two continuous learning cells. The diversity of learning gains was measured by students' related knowledge state and the interval time. We also designed a learning gate to distinguish students' absorptive capacity of knowledge. For the common forgetting phenomenon, we designed a forgetting gate to determine the reduction of students' knowledge over time. With extensive experiments on three public datasets, we proved that LPKT can get a more reasonable knowledge state that keeps consistent with students' cognitive process. Moreover, LPKT also outperformed state-of-the-art KT method on student performance prediction. Our work reveals a potential future research direction for the KT task by modeling students' learning process, which is of both high interpretability and high accuracy.

In future, we will keep exploring better ways to model students' learning process. For example, we may use pre-trained meaningful representations of exercises, which contain information about the difficulty. Besides, for measuring the learning gain more precisely, we can give more attention to related previous learning interactions. Finally, we will study how to automatically learn the specific weights in the Q-matrix to represent the relation between exercises and knowledge concepts more precisely.

ACKNOWLEDGMENT

This research was partially supported by grants from the National Natural Science Foundation of China (No.sU20A20229 and 61922073), the Foundation of State Key Laboratory of Cognitive Intelligence (No. iED2020-M004), and the Iflytek joint research program.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *OSDI*. 265–283.

[2] Ghodai Abdelrahman and Qing Wang. 2019. Knowledge Tracing with Sequential Key-Value Memory Networks *(SIGIR'19)*. New York, NY, USA, 175–184.

[3] Youngduck Choi, Youngnam Lee, Dongmin Shin, Junghyun Cho, Seoyon Park, Seewoo Lee, Jineon Baek, Chan Bae, Byungsoo Kim, and Jaewe Heo. 2020. Ednet: A large-scale hierarchical dataset in education. In *International Conference on Artificial Intelligence in Education*. Springer, 69–73.

[4] Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *UMUAI* 4, 4 (1994), 253–278.

[5] Mingyu Feng, Neil Heffernan, and Kenneth Koedinger. 2009. Addressing the assessment challenge with an online system that tutors as it assesses. *USER-ADAP* 19, 3 (2009), 243–266.

[6] Aritra Ghosh, Neil Heffernan, and Andrew S. Lan. 2020. Context-Aware Attentive Knowledge Tracing *(KDD '20)*. New York, NY, USA, 2330–2339.

[7] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*. 249–256.

[8] Richard R Hake. 2002. Relationship of individual student normalized learning gains in mechanics with gender, high-school physics, and pretest scores on mathematics and spatial visualization. In *Physics education research conference*, Vol. 8. 1–14.

[9] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580* (2012).

[10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[11] Zhenya Huang, Qi Liu, Yuying Chen, Le Wu, Keli Xiao, Enhong Chen, Haiping Ma, and Guoping Hu. 2020. Learning or Forgetting? A Dynamic Approach for Tracking the Knowledge Proficiency of Students. *ACM Transactions on Information Systems (TOIS)* 38, 2 (2020), 1–33.

[12] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[13] Julia KäFer, Susanne Kuger, Eckhard Klieme, and Mareike Kunter. 2019. The significance of dealing with mistakes for student achievement and motivation: results of doubly latent multilevel analyses. *European Journal of Psychology of Education* (2019).

[14] Van Der Maaten Laurens and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 2605 (2008), 2579–2605.

[15] Jingchen Liu, Gongjun Xu, and Zhiliang Ying. 2012. Data-driven learning of Q-matrix. *Applied psychological measurement* 36, 7 (2012), 548–564.

[16] Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Hui Xiong, Yu Su, and Guoping Hu. 2019. EKT: Exercise-aware Knowledge Tracing for Student Performance Prediction. *IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE)* (2019).

[17] Qi Liu, Shuanghong Shen, Zhenya Huang, Enhong Chen, and Yonghe Zheng. 2021. A Survey of Knowledge Tracing. *arXiv preprint arXiv:2105.15106* (2021).

[18] Geoffrey R Loftus. 1985. Evaluating forgetting curves. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 11, 2 (1985), 397.

[19] R Luckin et al. 2007. Beyond the code-and-count analysis of tutoring dialogues. *Artificial intelligence in education: Building technology rich learning contexts that work, R. Luckin, KR Koedinger, and J. Greer, Eds. IOS Press* (2007), 349–356.

[20] Ye Mao. 2018. Deep Learning vs. Bayesian Knowledge Tracing: Student Models for Interventions. *Journal of educational data mining* 10, 2 (2018).

[21] Shaul Markovitch and Paul D Scott. 1988. The role of forgetting in learning. In *Machine Learning Proceedings 1988*. Elsevier, 459–465.

[22] Cecile Hoareau McGrath, Benoit Guerin, Emma Harte, Michael Frearson, and Catriona Manville. 2015. Learning gain in higher education. *Santa Monica, CA: RAND Corporation* (2015).

[23] Koki Nagatani, Qian Zhang, Masahiro Sato, Yan-Ying Chen, Francine Chen, and Tomoko Ohkuma. 2019. Augmenting Knowledge Tracing by Considering Forgetting Behavior. In *WWW*. ACM, 3101–3107.

[24] Prema Nedungadi and MS Remya. 2015. Incorporating forgetting in the personalized, clustered, bayesian knowledge tracing (pc-bkt) model. In *2015 International Conference on cognitive computing and information processing (CCIP)*. IEEE, 1–5.

[25] Tuan Nguyen. 2015. The effectiveness of online learning: Beyond no significant difference and future horizons. *MERLOT Journal of Online Learning and Teaching* 11, 2 (2015), 309–319.

[26] Ebba Ossiannilsson. 2020. Sustainability: Special Issue "The Futures of Education in the Global Context: Sustainable Distance Education". *Sustainability* (07 2020).

[27] Shalini Pandey and George Karypis. 2019. A Self-Attentive model for Knowledge Tracing. *arXiv preprint arXiv:1907.06837* (2019).

[28] Shalini Pandey and Jaideep Srivastava. 2020. RKT: Relation-Aware Self-Attention for Knowledge Tracing *(CIKM '20)*. New York, NY, USA, 1205–1214.

[29] Philip I Pavlik Jr, Hao Cen, and Kenneth R Koedinger. 2009. Performance Factors Analysis–A New Alternative to Knowledge Tracing. *Online Submission* (2009).

[30] Radek Pelánek. 2017. Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User Modeling and User-Adapted Interaction* 27, 3-5 (2017), 313–350.

[31] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *NeurIPS*. 505–513.

[32] Cristóbal Romero and Sebastián Ventura. 2010. Educational data mining: a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40, 6 (2010), 601–618.

[33] Shuanghong Shen, Qi Liu, Enhong Chen, Han Wu, Zhenya Huang, Weihao Zhao, Yu Su, Haiping Ma, and Shijin Wang. 2020. Convolutional Knowledge Tracing: Modeling Individualization in Student Learning Process. *SIGIR '20: The 43rd International ACM SIGIR conference on research and development in Information Retrieval Virtual Event China July, 2020* (2020), 1857–1860.

[34] Dongmin Shin, Yugeun Shim, Hangyeol Yu, Seewoo Lee, Byungsoo Kim, and Youngduck Choi. 2020. SAINT+: Integrating Temporal Features for EdNet Correctness Prediction. *arXiv preprint arXiv:2010.12042* (2020).

[35] Gabriele Steuer and Markus Dresel. 2015. A constructive error climate as an element of effective learning environments. *Psychological Test and Assessment Modeling* 57, 2 (2015), 262 – 275.

[36] Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yuying Chen, Yu Yin, Zai Huang, and Shijin Wang. 2020. Neural Cognitive Diagnosis for Intelligent Education Systems. In *AAAI 2020*.

[37] Le Wu, Yong Ge, Qi Liu, Enhong Chen, Richang Hong, Junping Du, and Meng Wang. 2017. Modeling the evolution of users' preferences and social links in social networking services. *IEEE Transactions on Knowledge and Data Engineering* 29, 6 (2017), 1240–1253.

[38] Chun-Kit Yeung and Dit-Yan Yeung. 2018. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. *arXiv preprint arXiv:1806.02180* (2018).

[39] Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. 2017. Dynamic key-value memory networks for knowledge tracing. In *WWW*. 765–774.