This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221985

Monitoring Student Progress for Learning Process-consistent Knowledge Tracing

Shuanghong Shen, Enhong Chen, Qi Liu, Zhenya Huang, Wei Huang, Yu Yin, Yu Su, Shijin Wang

Abstract—Knowledge tracing (KT) is the task of tracing students' evolving knowledge state during learning, which has improved the learning efficiency. To facilitate KT's development, most existing methods pursue high accuracy of student performance prediction but neglect the consistency between students' dynamic knowledge state with their learning process. Moreover, they focus on learning outcomes at a single learning interaction, while student progress at continuous learning interactions is more instructive. 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 captures the evolution of students' knowledge state through monitoring their learning progress. Specifically, we utilize both the positive effect of the learning gain and the negative effect of forgetting in learning to calculate student progress in continuous learning interactions. Then, considering that the rate of progress is student-specific, we extend LPKT to LPKT-S by explicitly distinguishing the individual progress rate of each student. Extensive experimental results on three public datasets demonstrate that LPKT and LPKT-S could obtain more appropriate knowledge states in line with the learning process. Moreover, LPKT and LPKT-S outperform state-of-the-art KT methods on student performance prediction. Our work indicates a promising future research direction for KT, which is highly interpretable and accurate.

Index Terms—Educational Data Mining, Knowledge Tracing, Student Progress, Learning Process, Learning Gain, Forgetting Effect

1 Introduction

Recent years have witnessed the rapid development of online learning [1], which plays an indispensable role in realizing better education [2, 3]. 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 students' changing knowledge states [5, 6, 7, 8]. In online learning systems, students can achieve knowledge mastery by answering different exercises. In turn, we can also infer students' knowledge states and predict their future performance by their learning sequences, which is formalized as the KT task [4, 9]. Specifically, given students' historical learning sequence, including exercises and answers, the KT task aims to measure students' knowledge states at different time steps, and their future performance can be predicted by the related knowledge states [10]. Meanwhile, after understanding students' knowledge states, students and instructors can avoid wasting time on well-mastered knowledge concepts and pay more attention to those with poor mastery. In this way, KT can enhance learning and teaching simultaneously.

Existing KT models measure students' knowledge states by their learning sequences. For example, Bayesian Knowledge Tracing (BKT) applied Hidden Markov Model in the KT task [4], Deep Knowledge Tracing (DKT) introduced RNNs/LSTMs [11] to model student learning [12] and Exercise-aware Knowledge Tracing utilized the text information of exercises [13], which helped to further understand the exercises for better learning sequence modeling. They assumed that higher accuracy in future performance prediction is equivalent to a better estimate of the knowledge state. However, in the experiments of our previous work [14], we have noticed that pursuing only high accuracy of future performance prediction could lead to inconsistency between students' knowledge states and their learning process. For better illustration, we give a visualization case of the knowledge state traced by DKT, a popular KT model that has achieved impressive performance [12]. In Figure 1, while the student was answering 12 different exercises on 3 different knowledge concepts, DKT traced the evolving process of his/her knowledge state. It is easy to find an obvious observation from the figure: Once the student got wrong answers (e.g., e_2, e_3, e_4, e_8, and e_{12}), DKT thinks that his/her knowledge state will correspondingly decline (e.g., the knowledge state on the knowledge concept *70: Square Root* drops from 0.85 to 0.7 after wrongly answering e_8). Although such a downward trend after mistakes may bring higher accuracy to students' future performance prediction,

- *S. Shen, E. Chen (corresponding author), Z. Huang, W. Huang, and Y. yin are with the Anhui Province Key Laboratory of Big Data Analysis and Application, School of Data Science & School of Computer Science and Techonology, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence, Hefei, Anhui, 230026, China. Email: {closer, ustc0411, yxonic}@mail.ustc.edu.cn, {cheneh, huangzhy}@ustc.edu.cn*
- *Q. Liu is with the Anhui Province Key Laboratory of Big Data Analysis and Application, School of Data Science & School of Computer Science and Techonology, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence & Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, Anhui, 230026, China. Email: qiliuql@ustc.edu.cn*
- *Y. Su is with the School of Computer Science and Technology, Hefei Normal University & Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, Anhui, 230601, China. Email: yusu@hfnu.edu.cn*
- *S. Wang is with the iFLYTEK AI Research (Central China), iFLYTEK, Co., Ltd & State Key Laboratory of Cognitive Intelligence, Hefei, Anhui, 230088, China. Email: sjwang3@iflytek.com*

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Science & Technology of China. Downloaded on March 15,2023 at 17:09:25 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221985

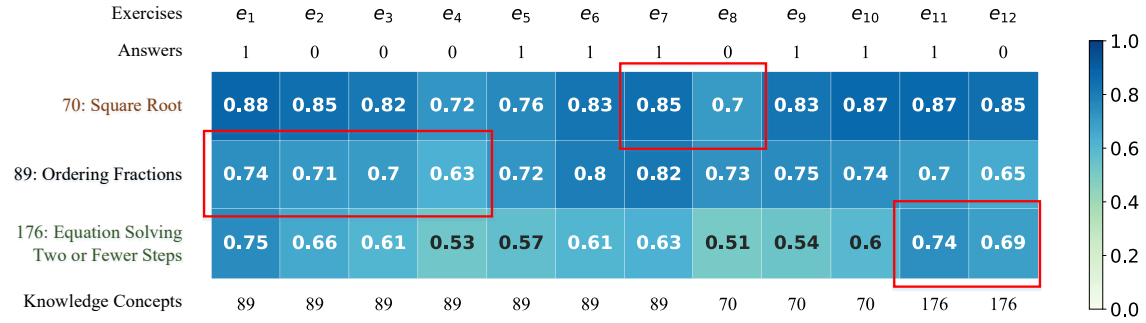


Fig. 1. A toy example of the evolution process of a student's knowledge state traced by DKT, where the student has answered 12 exercises on three knowledge concepts. The number in the small box refers to the knowledge state after answering the exercise, where the darker the color in the heat map, the higher the knowledge mastery. The red boxes emphasize that DKT thinks the knowledge state will decline after wrong answers.

it is not in line with students' cognitive processes. Actually, students can consistently acquire knowledge as long as they have practiced, whether the answer is correct or not. Previous research has also pointed out that mistakes are seen as natural and significant elements of the learning process [15], and students can learn from errors and foster knowledge acquisition through a favorable error climate [16].

On the other hand, under the guidance of seeking more accurate future performance prediction, existing KT methods choose to focus on students' learning outcomes at a single learning interaction (e.g., the student got the correct answer on e_1 at the first learning interaction in Figure 1). In this paper, we further find that the student progress in continuous learning interactions brings more significance to making effective learning schemes. In other words, modeling the learning outcome can only capture if the student has learned the particular knowledge concepts, but not if the student is learning at a pace that will allow completing his/her learning goals [17]. Besides, it may be too late to make effective changes to improve students' achievement when they receive their learning results [18]. Therefore, monitoring student progress is more instructive in teaching and learning practice.

To this end, we argue that although the predicted accuracy of student future performance and students' learning outcomes are significant for the KT task, it is essential to maintain the consistency of the knowledge state and the learning process. Besides, more attention should be given to students' learning progress in continuous learning interactions, which helps improve the quality of online education. In this paper, we aim to explore a new paradigm for the KT task by monitoring student progress in learning. However, there are many challenges to be solved along this line. First, it is unclear how to define the learning process and convert it into a proper form for modeling. Second, the student progress in continuous learning interactions is hard to be quantified. Specifically, two opposite factors will influence student progress: (1) the positive effect of the learning gain, which represents the knowledge that students acquire in learning, is implicit and changeable in the learning process. Although Mao [19] 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. (2) the negative effect of forgetting [20], which means students' knowledge will also decrease over time. This common forgetting phenomenon is also complicated but necessary to be considered. Third, the rate of learning progress differs from student to student, which should be distinguished to capture the individual's knowledge state.

To conquer the first two challenges, in our preliminary work [14], we proposed a novel method named Learning Process-consistent Knowledge Tracing (LPKT), which calculated students' learning gains and forgetting to reach our primary goal of assessing students' knowledge states by monitoring their learning progress. Specifically, as the learning process is presented as the learning sequence of students, we first defined the basic learning cell in the learning sequence as a tuple *exercise—answer time—answer*, and adjacent cells were connected by the interval time, forming the learning process. Notably, the learning cell contains the time that students spend on answering the exercise so that it is more capable of reflecting the complete learning process. Then, to measure student progress, we first directly computed the positive effect of the learning gain from the difference between present and previous learning cells. Besides, to capture the diversity of the learning gain, we also modeled two significant factors: the interval time between two continuous learning cells and students' related knowledge state. It is worth noting that the learning gain is always positive in LPKT, which means that students can consistently get knowledge at each learning interaction, whether their answers are right or wrong. Furthermore, considering that not all learning gains can be transformed as the growth of students' knowledge, we designed a learning gate to control students' knowledge absorptive capacity. Subsequently, for the negative effect of forgetting, we developed a forgetting gate to determine the decrease of knowledge state over time. Finally, considering both the learning gain and the forgetting, LPKT realized a novel way to assess students' knowledge state by modeling their learning process.

In LPKT, we assumed that students have the same progress rate, which does not hold in reality. Therefore, in this paper, we extend the LPKT model to LPKT-S by introducing a student-specific learning feature to explicitly distinct students' various progress rates. Specifically, when calculating the learning gain, LPKT-S considers students' individual knowledge absorptive capacity. For the forgetting, LPKT-S will compute how much knowledge will be forgotten by different students. Extensive experiments on three public real-world datasets demonstrate that LPKT and LPKT-S get more appropriate knowledge states in line

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Science & Technology of China. Downloaded on March 15,2023 at 17:09:25 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221985

with students' learning process. Moreover, LPKT and LPKT-S can also significantly outperform existing KT models on student performance prediction. LPKT-S achieves better performance than LPKT as it has introduced the individual student progress rate. Our idea to solve the KT problem by modeling students' learning progress indicates a potential future research direction of high interpretability and accuracy. In summary, compared with our preliminary model of LPKT [14], the main contributions of this paper are summarized as follows:

- We propose to conduct knowledge tracing by monitoring student progress at continuous learning interactions, which has not been explored in our preliminary work [14]. The proposed LPKT model obtains a more reasonable evolution of students' knowledge state and state-of-the-art results on student performance prediction.
- We further extend LPKT to LPKT-S, which explicitly distinguishes students' different rates of learning progress. Benefiting from introducing this student-specific feature, LPKT-S achieves better performance than LPKT.
- We conduct extensive experiments to show the interpretability of both LPKT and LPKT-S. The results indicate that modeling student progress can give more instructive feedback to students about their learning. Besides, LPKT and LPKT-S can also learn meaningful representations of exercises automatically.

2 Related Works

In this section, we briefly introduce the related works from the following four categories.

2.1 Knowledge tracing

Most of the existing KT models can be classified into three categories including probabilistic models, logistic models, and deep learning-based methods. More specifically, BKT [4] was a classic and widely-used probabilistic model for KT, which could be seen as a specific application of the Hidden Markov Model (HMM). Logistic models were a large class of models based on logistic functions, which utilized a logistic function to estimate the probability of knowledge state [5], such as Performance Factor Analysis (PFA) [21]. DKT introduced deep learning into KT for the first time [12], which took the learning sequence as the input of RNN or its variant LSTM and represented student knowledge states by the hidden states. Dynamic Key-Value Memory Networks (DKVMN) [22] introduced memory-augmented neural networks into KT. It defined a static matrix called *key* to store latent knowledge concepts and a dynamic matrix called *value* to store and update the knowledge mastery [22]. EKT [13] introduced text contents to enhance the performance of the KT task. Convolutional knowledge tracing (CKT) [23] applied the convolutional windows to model students' individualized learning rates within several continuous learning interactions. The self-attentive model for knowledge tracing (SAKT) [24] presented the self-attention mechanism to model the long-term dependencies between learning interactions. Pandey and Srivastava [25] developed a relation-aware self-attention layer that incorporates the contextual information. Ghosh et al. [26] presented a context-aware attentive knowledge tracing (AKT) model, which utilized contextualized representations of both exercises and knowledge acquisitions and incorporated the attention mechanism with cognitive and psychometric models. Shen et al. [27] considered the impact of question difficulty on student learning in KT. Readers can refer to Liu et al. [10] and Schmucker et al. [28] for a detailed survey of recent development of KT.

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 [29]. Learning gain differs from learning outcomes in that learning gain compares performance at two points in time, while learning outcome only concentrates on the output at a single point in time. For example, students may not benefit from the exercise even if he/she has already performed well on it. Luckin et al. [30] calculated learning gain as LG = $post - pre$, where pre and post were a student's pre-test and post-test scores. Normalized Learning Gain (NLG) [31] was a widely used adjusted measurement, which was calculated as follows:

$$NLG = \frac{post - pre}{1 - pre}, \tag{1}$$

where 1 is the maximum score for pre- and post-tests. However, NLG may be problematic in certain circumstances, such as even a slight decline in post-test score from the pre-test could result in a significant negative in NLG if the student had high pre-test scores. Mao [19] proposed a qualitative measurement called Quantized Learning Gain (QLG), which was a binary qualitative measurement of students' learning gains from pre-test to the post-test: *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, he/she is a *High* QLG. On the contrary, he/she will be a *Low* QLG. However, such a simple instantiation of the learning gain is not enough to capture its diversity.

2.3 Forgetting Effect

In real-world learning environments, forgetting is inevitable [20]. The *Ebbinghaus forgetting curve theory* indicated that students' knowledge proficiency may decline due to the forgetting factor [32]. Nedungadi and Remya [33] incorporated forgetting based on the assumption that the learned knowledge decays exponentially over time [34]. They utilized an exponential decay function to update the knowledge mastery level. Huang et al. [32] proposed the Knowledge Proficiency Tracing (KPT) model to model students' knowledge proficiency with both learning and forgetting theories, which dynamically captured the change in students' proficiency level over time. Nagatani et al. [35] made attempts to improve DKT by considering forgetting effects, but they only extended DKT by incorporating multiple types of time or counts information.

2.4 Student Progress

Student progress is of great significance for teachers and students to adjust teaching and learning strategies [18, 36].

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Science & Technology of China. Downloaded on March 15,2023 at 17:09:25 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221985

In contrast to mastery measurement, which tells whether a student has understood the particular knowledge concepts, monitoring student progress can evaluate the effectiveness of the teaching strategy, and teachers can promptly adjust instruction if the rate at which a specific student is learning seems insufficient [17]. There are two different interpretations of student progress: one compares a student's performance in the course to the student's performance at the same point from previous editions of the course. The other evaluates the students' distances from achieving goals expected to accomplish when finishing a course. Generally, student progress is monitored by periodic quizzes and tests (i.e., weekly, biweekly, or monthly). Ashenafi et al. [37] proposed to use peer-assessment data to build linear regression models for predicting students' weekly progress. In our paper, we combine the positive effect of learning gain and the negative effect of forgetting in the learning process together to measure student progress. In this way, student progress monitoring can go together with their daily learning, and no additional tests are required, avoiding time-consuming and potential disgust of the students.

3 PRELIMINARY

In this section, we first formalize the learning process and briefly introduce the definition of knowledge tracing. Subsequently, we present some essential embeddings from four categories in LPKT and LPKT-S. The mathematical notations utilized in our paper are summarized in Table 1.

3.1 Problem Definition

In an intelligent tutoring system, supposing there are the set of students $S = \{s_1, s_2, ..., s_i, ..., s_I\}$ with I different students, the set of exercises $E = \{e_1, e_2, ..., e_j, ..., e_J\}$ with J different exercises, and the set of knowledge concepts $K = \{k_1, k_2, ..., k_m, ..., k_M\}$ with M different knowledge concepts, where each exercise is related to specific knowledge concepts. The Q-matrix $\boldsymbol{Q} \in \mathbb{R}^{J \times M}$, which is consisted of zeros and ones, indicates the relationship between exercises and knowledge concepts. Here $\boldsymbol{Q}_{jm} = 1$ stands for that knowledge concept k_m is required for exercise e_j and $\boldsymbol{Q}_{jm} = 0$ if not. Generally, when an exercise is assigned to the student, he/she spends a certain time 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 interactions. Therefore, we denote the learning process of a student as $\mathbf{x} = \{(e_1, at_1, a_1), it_1, (e_2, at_2, a_2), it_2, ..., (e_t, at_t, a_t), it_t\}$, 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* $\boldsymbol{X} = \{(e_1, at_1, a_1), it_1, (e_2, at_2, a_2), it_2, ..., (e_t, at_t, a_t), it_t\}$, *the KT task aims to monitor students' changing knowledge state during the learning process and predict their future performance at the next learning interaction* $t+1$, *which can be further applied to individualize students' learning scheme and maximize their learning efficiency.*

TABLE 1
Mathematical notations and descriptions.

Notations	Descriptions
S, E, KC	The set of students, exercises, and knowledge concepts.
I, J, M	The number of students, exercises, and knowledge concepts.
$\boldsymbol{X}$	Students' learning sequence.
$\boldsymbol{Q}$	The Q-matrix.
$\boldsymbol{Q}^e$	The defined enhance Q-matrix.
$\boldsymbol{E}$	The exercise embedding matrix.
$\boldsymbol{l}$	The learning embedding of the learning cell.
$\boldsymbol{q}$	The knowledge concept vector.
$\boldsymbol{s}$	The student embedding.
$\boldsymbol{h}, \widetilde{\boldsymbol{h}}$	The knowledge state and related knowledge state.
x	One step of students' learning sequence.
k	The knowledge concept.
e, $\boldsymbol{e}$	The exercise and its embedding.
at, $\boldsymbol{at}$	The answer time and its embedding.
it, $\boldsymbol{it}$	The interval time and its embedding.
a, $\boldsymbol{a}$	Students' actual answer and its embedding.
$\boldsymbol{lg}$	The initial learning gain in LPKT and LPKT-S.
$\boldsymbol{LG}$, $\widetilde{\boldsymbol{LG}}$	The learning gain and related learning gain in LPKT.
$\boldsymbol{\Gamma}^l$, $\boldsymbol{\Gamma}^f$	The learning gate and forgetting gate in LPKT.
$\boldsymbol{p}$	The student progress in LPKT.
$\boldsymbol{LG}_s$, $\widetilde{\boldsymbol{LG}}_s$	The learning gain and related learning gain in LPKT-S.
$\boldsymbol{\Gamma}_s^l$, $\boldsymbol{\Gamma}_s^f$	The learning gate and forgetting gate in LPKT-S.
$\boldsymbol{p}_s$	The student progress in LPKT-S.
y	The prediction of student future performance.

3.2 Embeddings

In LPKT, to realize our goal of modeling student progress in the learning process, 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. In addition, we also consider some other elements, such as the knowledge concepts and students' knowledge state. In LPKT-S, we further introduce a student-specific element to capture the differences in the progress rate of students. To better understand the structure of LPKT and LPKT-S before presenting their details, we briefly introduce the embeddings of those elements from the following four categories.

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, influencing students' learning gains and the forgetting effect. For example, a longer answer time is more likely to bring more learning gains, and a longer interval time generally causes more forgetting. Huang et al. [32] and Loftus [34] introduced the *Forgetting curve theory* to model the decreasing knowledge state of students as time goes on. Nagatani et al. [35] discretized all time features by minutes at the log_2 scale and represented them as one-hot vectors. In LPKT and LPKT-S, because the interval time could be much longer than the answer time, we discretize the former by the minutes and the latter by the seconds [38]. Besides, we set the maximum interval time as ten days. 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_k is the dimension, d_{at} and d_{it} are the number of the discretized answer time and interval time. Then, at_t and it_t

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Science & Technology of China. Downloaded on March 15,2023 at 17:09:25 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221985

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 process and characterizes the knowledge they acquire by answering exercises. We first represent the exercise set by an embedding matrix $\boldsymbol{E} \in \mathbb{R}^{J \times d_e}$, where 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:

$$\boldsymbol{l}_t = \boldsymbol{W}_1^T[\boldsymbol{e}_t \oplus \boldsymbol{at}_t \oplus \boldsymbol{a}_t] + \boldsymbol{b}_1, \tag{2}$$

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 students' knowledge state during their learning process. In LPKT and LPKT-S, 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 positive effect of learning gain and the negative effect of forgetting on each knowledge concept are both calculated to get student progress, which then will be updated into the knowledge embedding.

The Q-matrix indicates the relations between exercises and knowledge concepts, determining 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 m-th row of the student's knowledge matrix will be updated. Traditionally, if the knowledge concept k_m is not contained in the exercise e_j, $\boldsymbol{Q}_{jm}$ will be set as 0, which shows students' performance on exercise e_j has no influence on their knowledge mastery $\boldsymbol{h}_m$ on k_m. However, manually-labeled Q-matrix may be deficient because of inevitable errors and subjective bias [39, 40]. In order to make up for possible omissions or mistakes, we define an enhanced Q-matrix $\boldsymbol{Q}^e \in \mathbb{R}^{J \times M}$, where $\boldsymbol{Q}^e_{jm}$ will be set as a small positive value γ rather than 0 even if k_m is not in e_j. Then, for each knowledge concept k_m, we can get the knowledge concept vector $\boldsymbol{q}_{k_m}$. Although this unified setting is simple, we leave the exploration to learn the specific weights in the Q-matrix as future works as this paper focuses on the learning process modeling part.

3.2.4 Student Embedding

In our previous work [14], we assumed that all students have the same progress rate. This assumption is unreasonable to some extent, and the progress rate should be a student-specific feature. For example, some students make faster progress on geometry, and others are more good at learning algebra. Therefore, we extend LPKT to LPKT-S and proposes the student embedding $\boldsymbol{S} \in \mathbb{R}^{I \times d_s}$, which assigns a specific vector $\boldsymbol{s}_i$ for student i. Here $\boldsymbol{s}_i$ is utilized to distinguish the individual progress rate of the student. LPKT-S can learn vectors containing different progress rate information for different students in the training process, which will be used to make more precise and personalized predictions.

Algorithm 1 The LPKT Model.

Input: The embedding of the learning cell, $\boldsymbol{l}_t \in \mathbb{R}^{d_k}$; The embedding of the interval time, $\boldsymbol{it}_t \in \mathbb{R}^{d_k}$; The student's previous knowledge state, $\boldsymbol{h}_{t-1} \in \mathbb{R}^{M \times d_k}$; The next exercise to be answered, $\boldsymbol{e}_{t+1} \in \mathbb{R}^{d_e}$.

Output: The student's updated knowledge state, $\boldsymbol{h}_t \in \mathbb{R}^{M \times d_k}$; The prediction of student's answer on the next exercise, y_{t+1}.

compute the related knowledge state by Eq. (3);
obtain the student's initial learning gain from Eq. (4);
compute the learning gate by Eq. (5);
obtain the related learning gain from Eq. (6);
define the forgetting gate by Eq. (7);
compute student progress and update the student's knowledge state by Eq. (8);
predict the student's performance by Eq. (10);
return $\boldsymbol{h_t}$, y_{t+1};

4 The LPKT Model

In this section, we present the LPKT model in detail. The main structure of LPKT is depicted in Figure 2 and Algorithm 1. Precisely, LPKT consists of three modules at each learning interaction: (1) the learning gain module, (2) the forgetting module, and (3) the predicting module. After a student has answered an exercise, the learning module models his/her learning gains compared with the previous learning interaction. The forgetting module measures how much knowledge will be forgotten over time. Then, the positive effect of learning gains and the negative effect of forgotten knowledge will be utilized to output the student's learning progress and update his/her previous knowledge state to obtain the 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 Gain Module

As our primary goal is to model the student progress for the KT task, after formalizing the learning process as alternate combinations of the basic learning cell and the interval time, the next problem is to measure the implicit and dynamic learning progress in the learning process. Traditionally, a practice or a learning effect occurs when students answer questions, i.e., the positive effect of the learning gain. Previous studies have defined the learning gain as 'distance traveled' [29], which stands for the difference in students' performance at two points in time. Based on this definition, we should consider the differences in students' performance during two continuous learning interactions to model the

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Science & Technology of China. Downloaded on March 15,2023 at 17:09:25 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221985

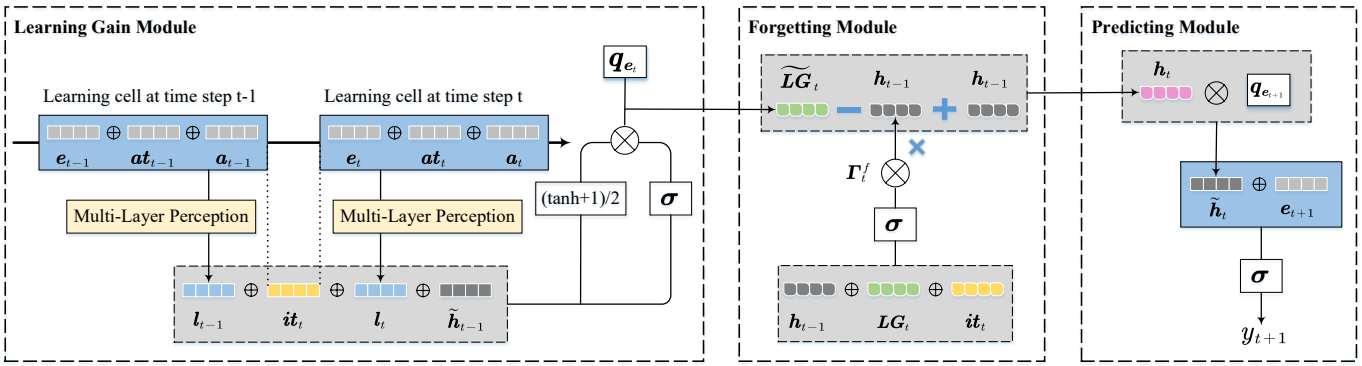


Fig. 2. The architecture of the LPKT model. For convenience, we only give the processing at timestamp t and conduct similar computations recurrently on the student's learning sequence. Specifically, the input is the current learning cell $e_t, \boldsymbol{at}_t, \boldsymbol{a}_t$, its previous neighbor $e_{t-1}, \boldsymbol{at}_{t-1}, \boldsymbol{a}_{t-1}$, and their interval time $\boldsymbol{it}_t$. LPKT calculates the learning progress by quantifying the student's learning gain and forgetting. Then, the learning progress will be utilized to update the student's knowledge state.

learning gain precisely. 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. However, although we can capture the differences in students' performance with two continuous learning embeddings, it cannot 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 the part of overlapped learning sequences (i.e., the same continuous learning embeddings). Therefore, we consider two influencing factors of the learning gains: the interval time and students' previous knowledge state. On the one hand, the interval time between two learning cells is a critical element of the learning process, reflecting the distinctions of learning gains. Generally, students acquire more knowledge at shorter intervals, making their learning process 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 for improvement. Therefore, we incorporate the above two factors into LPKT to model learning gains' evolution. 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 the previous knowledge state, to focus on the knowledge state of 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 $\widetilde{\boldsymbol{h}}_{t-1}$:

$$\widetilde{\boldsymbol{h}}_{t-1} = \boldsymbol{q}_{e_t} \cdot \boldsymbol{h}_{t-1}, \tag{3}$$

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

$$\boldsymbol{lg}_t = tanh(\boldsymbol{W}_2^T[\boldsymbol{l}_{t-1} \oplus \boldsymbol{it}_t \oplus \boldsymbol{l}_t \oplus \widetilde{\boldsymbol{h}}_{t-1}] + \boldsymbol{b}_2), \tag{4}$$

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:

$$\boldsymbol{\Gamma}_t^l = \sigma(\boldsymbol{W}_3^T[\boldsymbol{l}_{t-1} \oplus \boldsymbol{it}_t \oplus \boldsymbol{l}_t \oplus \widetilde{\boldsymbol{h}}_{t-1}] + \boldsymbol{b}_3), \tag{5}$$

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 multiplied to $\boldsymbol{lg}_t$ to get the actual learning gain $\boldsymbol{LG}_t$ in the $t-$th learning interaction. Similarly, to expand the learning gain to other knowledge concepts, we multiply $\boldsymbol{LG}_t$ by $\boldsymbol{q}_{e_t}$ to get the overall learning gains $\widetilde{\boldsymbol{LG}}_t$:

$$\begin{aligned} \boldsymbol{LG}_t &= \boldsymbol{\Gamma}_t^l \cdot ((\boldsymbol{lg}_t + 1)/2), \\ \widetilde{\boldsymbol{LG}}_t &= \boldsymbol{q}_{e_t} \cdot \boldsymbol{LG}_t, \end{aligned} \tag{6}$$

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 $\boldsymbol{LG}_t$ and $\widetilde{\boldsymbol{LG}}_t$ will always be 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* [34], the amount of learned material that is remembered decays exponentially over time. Nevertheless, a simple manual-designed exponential decay function is insufficient 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 $\boldsymbol{it}_t$. The greater power of non-linearity makes MLP more capable of capturing the complex students' forgetting behavior in learning. The specific calculating formula is as follows:

$$\boldsymbol{\Gamma}_t^f = \sigma(\boldsymbol{W}_4^T[\boldsymbol{h}_{t-1} \oplus \boldsymbol{LG}_t \oplus \boldsymbol{it}_t]) + \boldsymbol{b}_4), \tag{7}$$

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 utilize both the positive effect of the learning gain and the negative effect of forgetting to assess the student progress in learning, which is then used to update the student's knowledge state. Specifically, we first eliminate

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Science & Technology of China. Downloaded on March 15,2023 at 17:09:25 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221985

the influence of forgetting by multiplying $\boldsymbol{\Gamma}_t^f$ to $\boldsymbol{h}_{t-1}$ and the knowledge state $\boldsymbol{h}_t$ after students have accomplished the $t-$th learning interaction will be updated as follows [1]:

$$\begin{aligned} \boldsymbol{p}_t &= \widetilde{\boldsymbol{LG}}_t - \boldsymbol{\Gamma}_t^f \boldsymbol{h}_{t-1}, \\ \boldsymbol{h}_t &= \boldsymbol{p}_t + \boldsymbol{h}_{t-1}. \end{aligned} \tag{8}$$

As the neural network has shown great potential to model the non-linearity relationship [41], we have also attempted to update the knowledge state by the neural network as follows:

$$\boldsymbol{h}_t = \sigma(\boldsymbol{W}_N^T[\boldsymbol{h}_{t-1} \oplus \widetilde{\boldsymbol{LG}}_t \oplus \boldsymbol{\Gamma}_t^f] + \boldsymbol{b}_N), \tag{9}$$

where $\boldsymbol{W}_N \in \mathbb{R}^{(3d_k)\times d_k}$ is the weight matrix, $\boldsymbol{b}_N \in \mathbb{R}^{d_k}$ is the bias term. In this way, the neural network will automatically combine the positive effect of the learning gain, the negative effect of forgetting, and the student's previous knowledge state to output the latest knowledge state. However, in the experiments, the results of the neural combination are slightly worse than the means of addition in Eq. 8, we will show it in section 6.7.

4.3 Predicting Module

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

In a real learning environment, after reading a new exercise $\boldsymbol{e}_{t+1}$, the student 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 $\widetilde{\boldsymbol{h}}_t$ and the exercise embedding $\boldsymbol{e}_{t+1}$, then project them to the output prediction by a fully connected network with averaging operation and sigmoid activation:

$$y_{t+1} = \sigma(\frac{\sum(\boldsymbol{W}_5^T[\boldsymbol{e}_{t+1} \oplus \widetilde{\boldsymbol{h}}_t] + \boldsymbol{b}_5)}{d_k}), \tag{10}$$

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 in 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, where he/she can get the 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:

$$\mathbb{L}(\theta) = -\sum_{t=1}^{T}(a_t \log y_t + (1-a_t)\log(1-y_t)) + \lambda_\theta ||\theta||^2, \tag{11}$$

where θ denotes all parameters of LPKT and λ_θ is the regularization hyperparameter. The objective function was minimized using Adam optimizer [42] on mini-batches. More details of settings will be specified in the experiments.

1. In our previous version [14], we directly updated the knowledge state $\boldsymbol{h}_t$ without calculating the student progress $\boldsymbol{p}_t$.

Algorithm 2 The LPKT-S Model.

Input: The embedding of the learning cell, $\boldsymbol{l}_t \in \mathbb{R}^{d_k}$; The embedding of the interval time, $\boldsymbol{it}_t \in \mathbb{R}^{d_k}$; The student's previous knowledge state, $\boldsymbol{h}_{t-1} \in \mathbb{R}^{M\times d_k}$; The student embedding, $\boldsymbol{s}_i \in \mathbb{R}^{d_s}$; The next exercise to be answered, $\boldsymbol{e}_{t+1} \in \mathbb{R}^{d_e}$.

Output: The student's updated knowledge state, $\boldsymbol{h}_t \in \mathbb{R}^{M\times d_k}$; The prediction of student's answer on the next exercise, y_{t+1}.

```
compute the related knowledge state by Eq. (3);
obtain the student's initial learning gain from Eq. (4);
compute the learning gate by Eq. (12);
obtain the related learning gain from Eq. (13);
define the forgetting gate by Eq. (14);
compute student progress and update the student's knowledge state by Eq. (15);
predict the student's performance by Eq. (16);
return h_t, y_{t+1};
```

5 The LPKT-S Model

In this paper, we aim to assess students' evolving knowledge state by monitoring their learning progress, where their different progress rates have significant impacts. In LPKT, the differences between students' progress rates are implicitly captured by their knowledge state, the answer time, and the interval time, which is insufficient to distinguish the student-specific progress rates. In other words, students with similar knowledge states, answer time, and interval time could significantly differ in progress rates. Therefore, explicitly distinguishing students' progress rates in learning is necessary. In this section, we extend LPKT to LPKT-S by introducing the student embedding to assign an individual progress rate for each student.

Figure 3 depicts the main structure of LPKT-S, and Algorithm 2 gives its processing flow. In contrast to LPKT, LPKT-S makes changes in all three modules. Specifically, in the learning gain module, the progress rate mainly influences students' knowledge absorptive capacity, i.e., students with faster progress rates are better at transforming the initial learning gain into their knowledge growth. Therefore, the student's initial learning gain in LPKT-S is also calculated by Eq. (4), while the learning gate $\boldsymbol{\Gamma}_{s,t}^l$ is determined by the student embedding $\boldsymbol{s}_i$, two continuous learning embeddings and their interval time as follows:

$$\boldsymbol{\Gamma}_{s,t}^l = \sigma(\boldsymbol{W}_6^T[\boldsymbol{l}_{t-1} \oplus \boldsymbol{it}_t \oplus \boldsymbol{l}_t \oplus \boldsymbol{s}_i] + \boldsymbol{b}_6), \tag{12}$$

where $\boldsymbol{W}_6 \in \mathbb{R}^{(3d_k+d_s)\times d_k}$ is the weight matrix, $\boldsymbol{b}_6 \in \mathbb{R}^{d_k}$ is the bias term. Similarly to LPKT, $\boldsymbol{\Gamma}_{s,t}^l$ then will be multiplied to $\boldsymbol{lg}_t$ to get the actual learning gains $\boldsymbol{LG}_{s,t}$ in the $t-$th learning interaction. We also multiply $\boldsymbol{LG}_{s,t}$ by $\boldsymbol{q}_{e_t}$ to get the overall learning gains $\widetilde{\boldsymbol{LG}}_{s,t}$:

$$\begin{aligned} \boldsymbol{LG}_{s,t} &= \boldsymbol{\Gamma}_{s,t}^l \cdot ((\boldsymbol{lg}_t + 1)/2), \\ \widetilde{\boldsymbol{LG}}_{s,t} &= \boldsymbol{q}_{e_t} \cdot \boldsymbol{LG}_{s,t}. \end{aligned} \tag{13}$$

Subsequently, in the forgetting module, since the forgetting behaviors are also variations among students, we further introduce the student embedding to monitor how much knowledge will be forgotten by different students

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Science & Technology of China. Downloaded on March 15,2023 at 17:09:25 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221985

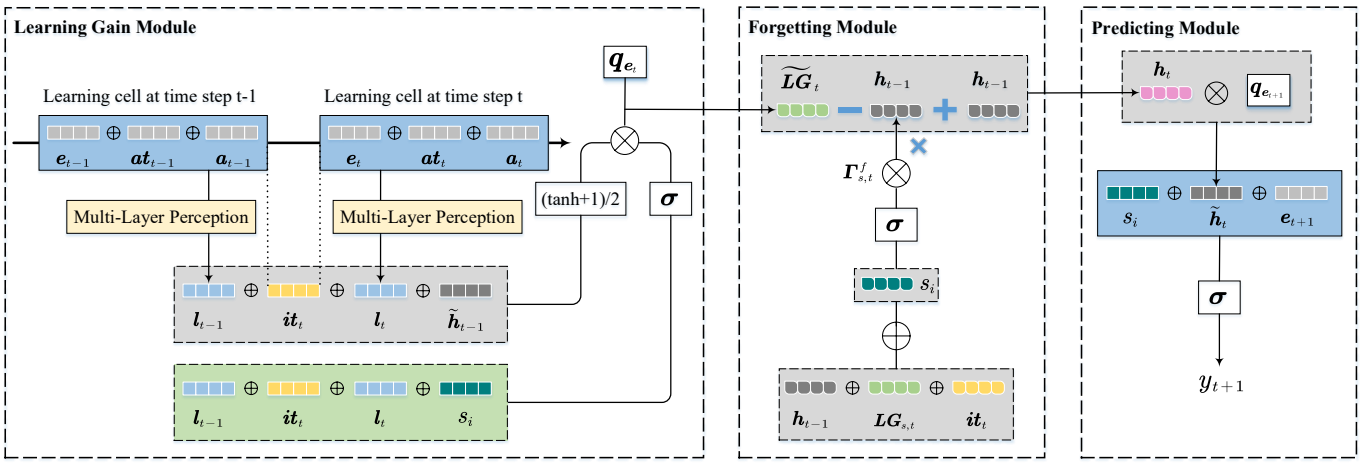


Fig. 3. The architecture of the LPKT-S model, which is similar to LPKT with three modules: learning gain module, forgetting module, and predicting module. The difference is that LPKT-S considers the student-specific progress rate and explicitly measures its influence on the above three modules.

based on the original three elements in LPKT. As a result, the forgetting gate $\boldsymbol{\Gamma}^f_{s,t}$ in LPKT-S is computed as follows:

$$\boldsymbol{\Gamma}^f_{s,t} = \sigma(\boldsymbol{W}_7^T[\boldsymbol{h}_{t-1} \oplus \boldsymbol{LG}_{s,t} \oplus \boldsymbol{it}_t \oplus \boldsymbol{s}_i]) + \boldsymbol{b}_7), \tag{14}$$

where $\boldsymbol{W}_7 \in \mathbb{R}^{(3d_k+d_s)\times d_k}$ is the weight matrix, $\boldsymbol{b}_7 \in \mathbb{R}^{d_k}$ is the bias term. Then, we can also utilize the student-specific positive effect of the learning gain and the negative effect of forgetting to assess the learning progress and update the student's knowledge state. The computing method is the same as Eq. (8):

$$\begin{aligned} \boldsymbol{p}_{s,t} &= \widetilde{\boldsymbol{LG}}_{s,t} - \boldsymbol{\Gamma}^f_{s,t}\boldsymbol{h}_{t-1}, \\ \boldsymbol{h}_t &= \boldsymbol{p}_{s,t} + \boldsymbol{h}_{t-1}. \end{aligned} \tag{15}$$

Finally, in the predicting module, we also consider the impacts of the student embedding on students' styles of answering exercises. In other words, students may have different characteristics when applying their knowledge to solve problems. Therefore, we add the student embedding $\boldsymbol{s}_i$ to Eq. (10) in LPKT-S as follows:

$$y_{t+1} = \sigma(\frac{\sum(\boldsymbol{W}_8^T[\boldsymbol{e}_{t+1} \oplus \widetilde{\boldsymbol{h}}_t \oplus \boldsymbol{s}_i] + \boldsymbol{b}_8)}{d_k}), \tag{16}$$

where $\boldsymbol{W}_8 \in \mathbb{R}^{(2d_k+d_s)\times d_k}$ is the weight matrix, $\boldsymbol{b}_8 \in \mathbb{R}^{d_k}$ is the bias term. Then we can train LPKT-S by minimizing the same objective function in Eq. (11).

By introducing the student embedding into the three modules in LPKT, we realize the extended LPKT-S model that explicitly distinguishes students' different progress rates. In the next section, we will show that LPKT-S can perform better than LPKT.

6 Experiments

In this section, we first describe the real-world datasets used in the experiments. We then introduce the training details of LPKT and LPKT-S and the baseline models. Subsequently, we conduct several experiments to show their interpretability from the following aspects: (1) Both LPKT and LPKT-S can keep the consistency of students' changing knowledge state to their learning process by monitoring student progress. (2) LPKT outperforms the state-of-the-art knowledge model on student performance prediction. LPKT-S can obtain better results than LPKT as it considers students' different progress rates. (3) The learning gain module, the forgetting module, and the time information utilized in LPKT and LPKT-S impact the knowledge tracing results differently. (4) The student progress measured by LPKT and LPKT-S has an excellent guiding significance for making study schemes. (5) LPKT and LPKT-S can learn meaningful exercise representations in the training process.

6.1 Datasets

To evaluate the effectiveness of LPKT and LPKT-S, we utilize three real-world public datasets for experiments. Table 3 shows the statistics of all datasets. Figure 4 presents the distribution of the interval time and the answer time in all datasets, which approximates the logarithmic normal distribution. It is worth noting that there are some small peaks in the distribution of the interval time, which is related to students' practice patterns [45]. Besides, we have segmented the learning sequences by interval time longer than ten days. Therefore, the average length after segmentation is shorter than the initial. A simple description of all datasets is listed as follows:

- **ASSISTments 2012**[2] **(ASSIST2012)** is collected from the ASSISTments [46], 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**[3] **(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 from 2004 to 2007. In this dataset, we also excluded the records without related knowledge concepts.

2. https://sites.google.com/site/assistmentsdata/home/2012-13-school-data-with-affect
3. https://sites.google.com/view/assistmentsdatamining/dataset

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Science & Technology of China. Downloaded on March 15,2023 at 17:09:25 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221985

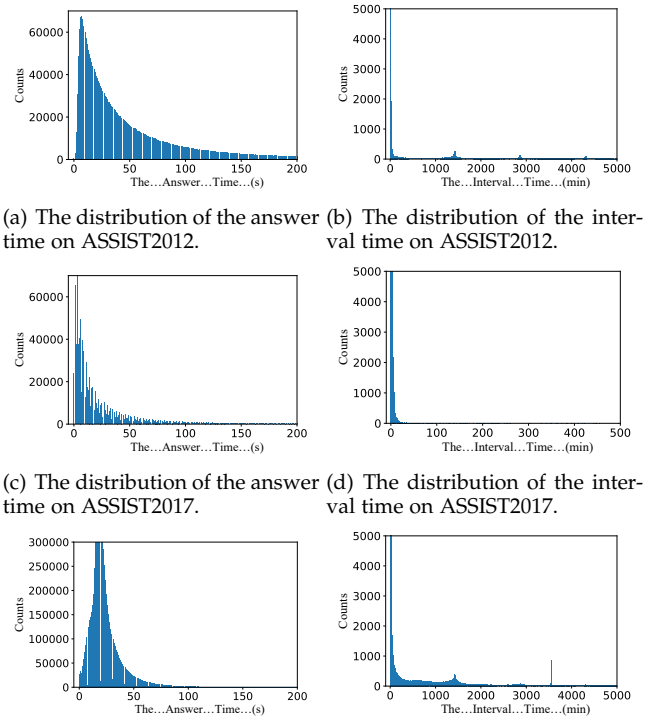
TABLE 2
The introduction of all baselines, including their pros and cons.

Methods	Introduction	Pros	Cons
DKT [12]	using RNNs/LSTMs to model students' learning sequence, where the hidden state is seen as the knowledge state.	introducing deep learning to KT for the first time.	directly applying RNNs/LSTMs in KT without considering the learning property.
DKT+ [43]	solving two problems in DKT: (1) failing to reconstruct the observed input, (2) the predicted performance across time steps is inconsistent.	enhancing the interpretability of DKT.	the performance is limited.
DKVMN [22]	utilizing the memory network to get interpretable knowledge states.	storing and updating students' knowledge state on specific KCs.	the defined read and write process to store and update the knowledge state is complex.
SAKT [24]	applying the self-attentive mechanism for KT.	capturing the long-term dependencies between different learning records.	the utilization of self-attentive is insufficient.
CKT [23]	introducing CNNs to solve the KT problem.	implicitly modeling the student-specific learning rate and prior knowledge.	the interpretability is insufficient.
AKT [26]	using Rasch model-based embeddings to represent exercises and designing the self-attentive encoder to learn context-aware representations of exercises and answers. Based on the attention mechanism, proposing the knowledge retriever to retrieve knowledge acquired in the past relevant to the current exercise.	achieving quite good performance based on effective utilization of the attention mechanism.	the reason for performance promotion is students' repeated attempts [44]

TABLE 3
Statistics of all datasets.

Statistics	Datasets		
	ASSIST2012	ASSISTchall	EdNet-KT1
Students	29,018	1,709	78,431
Exercises	53,091	2,210	12,372
Concepts	265	102	141
Answer Time	26,715	1,265	4,030
Interval Time	13709	1,151	13,142
Initial Avg.length	93.45	505.98	125.45
Avg.length after segmentation	30.58	73.85	124.04

- **EdNet-KT1**[4] is the dataset of all student-system interactions collected over two years by Santa, a multi-platform AI tutoring service with more than 780K users in Korea available through Android, iOS and web [47]. To provide various 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. Since the knowledge tags in this dataset are hierarchical, we only use the most fine-grained tag as its knowledge concept. Moreover, this dataset is rather big, with more than 780,000 unique students. Nevertheless, so many records are unnecessary for our experiments, which brings a big computational burden. Therefore, we use only 10% of the whole data, as shown in Table 3.



(a) The distribution of the answer time on ASSIST2012. (b) The distribution of the interval time on ASSIST2012.

(c) The distribution of the answer time on ASSIST2017. (d) The distribution of the interval time on ASSIST2017.

(e) The distribution of the answer time on EdNet-KT1. (f) The distribution of the interval time on EdNet-KT1.

Fig. 4. Distributions of answer times and interval times in all datasets.

6.2 Training Details

We first sorted all learning records of the student by the answering timestamp. As mentioned above, we split the learning sequence by interval times longer than ten days. Then, we set all input sequences to a fixed length of 50. For sequences longer than the fixed length, we cut them into several unique sub-sequences according to the fixed length. Zero vectors were used to pad the sequences up to the fixed length for the sequences shorter than the fixed length.

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 [48]. All the hyper-parameters are learned on the training set, and the model that performed best on the validation set was used to evaluate the testing set. In LPKT and LPKT-S, we added a dropout layer [49] with a dropout rate of 0.2 to prevent overfitting. Parameter d_k, d_e, and d_s are all set to be 128 and d_a is 50 in our implementation. The small positive value γ in the enhanced Q-matrix $\boldsymbol{Q}^e$ is 0.03. The regularization hyperparameter λ_θ in the objective function is 1e-6. Our code is available at https://github.com/shshen-closer/LPKT-S.

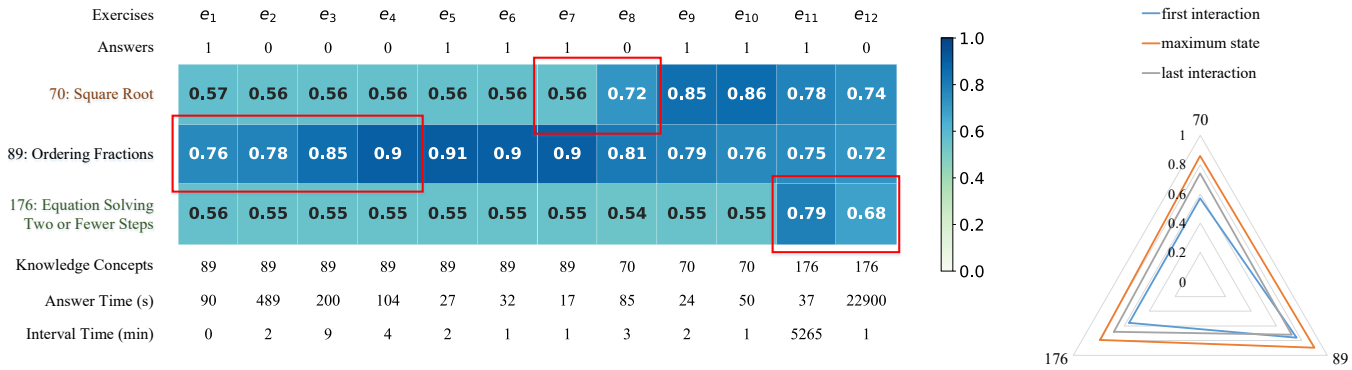
4. http://ednet-leaderboard.s3-website-ap-northeast-1.amazonaws.com/

6.3 Baselines

We compare LPKT with several previous methods. All these methods are tuned to have the best performances for a fair comparison. All models are trained on a cluster of Linux servers with TITAN V100 GPUs. We have summarized the characteristic of all baselines in Table 2.

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Science & Technology of China. Downloaded on March 15,2023 at 17:09:25 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221985



(a) The evolution process of the same student's knowledge state traced by LPKT-S in Figure 1.

(b) The radar diagram.

Fig. 5. The evolution process of a student's (the same student in Figure 1) knowledge state traced by LPKT-S. In sub-figure (a), we give more information about 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 knowledge state in the whole learning process is also depicted.

TABLE 4
Results of comparison methods on student performance prediction. The best results are bold and the existing state-of-the-art are underlined.

Methods	ASSIST2012				ASSISTchall				EdNet-KT1			
	RMSE	AUC	ACC	r^2	RMSE	AUC	ACC	r^2	RMSE	AUC	ACC	r^2
DKT	0.4253	0.7276	0.7335	0.1456	0.4486	0.7136	0.6895	0.1346	0.4555	0.6663	0.6812	0.0832
DKT+	0.4265	0.7256	0.7321	0.1409	0.4505	0.7088	0.6873	0.1272	0.4602	0.6547	0.6719	0.0642
DKVMN	0.4287	0.7188	0.7285	0.1347	0.4530	0.6978	0.6826	0.1166	0.4563	0.6625	0.6798	0.0797
SAKT	0.4266	0.7238	0.7320	0.1405	0.4597	0.6733	0.6759	0.0909	0.4556	0.6658	0.6808	0.0828
CKT	0.4253	0.7279	0.7334	0.1458	0.4510	0.7063	0.6860	0.1253	0.4560	0.6642	0.6803	0.0809
AKT	0.4121	0.7706	0.7515	0.2004	0.4364	0.7501	0.7080	0.1801	0.4297	0.7557	0.7204	0.1842
LPKT	0.4089	0.7740	0.7551	0.2103	0.4179	0.7939	0.7385	0.2491	0. 4290	0. 7577	0.7218	0.1867
LPKT-S	**0.4065**	**0.7803**	**0.7584**	**0.2195**	**0.4160**	**0.7979**	**0.7420**	**0.2558**	**0.4261**	**0.7662**	**0.7259**	**0.1976**

TABLE 5
Results of ablation experiments on EdNet-KT1.

Methods	learning	forgetting	time	combination	RMSE	AUC	ACC	r^2
LPKT_L	✓		✓	addition	0.4318	0.7503	0.7172	0.1760
LPKT_F		✓	✓	addition	0.4295	0.7561	0.7210	0.1846
LPKT (no time)	✓	✓		addition	0.4298	0.7556	0.7203	0.1837
LPKT_N	✓	✓	✓	neural network	0.4307	0.7536	0.7192	0.1804
LPKT	✓	✓	✓	addition	0.4290	0.7577	0.7218	0.1867
LPKT-S_L	✓		✓	addition	0.4283	0.7604	0.7223	0.1893
LPKT-S_F		✓	✓	addition	0.4267	0.7644	0.7249	0.1953
LPKT-S (no time)	✓	✓		addition	0.4268	0.7644	0.7248	0.1950
LPKT-S_N	✓	✓	✓	neural network	0.4281	0.7611	0.7228	0.1902
LPKT-S	✓	✓	✓	addition	0.4261	0.7662	0.7259	0.1976

6.4 Knowledge State Visualization

As our primary goal focuses on maintaining the consistency between the traced knowledge state of students and their learning process, we will first show that our proposed method can capture reasonable knowledge state of students that is consistent with their learning process as expected. Specifically, Figure 5 shows the evolving knowledge state traced by LPKT-S of the same student in Figure 1. There are several important observations in the figure. First, our proposed model can capture students' learning progress from both wrong and right learning interactions. For example, even the student answered exercise e_2, e_3, e_4, and e_8 wrongly, LPKT thinks his/her knowledge state on related knowledge concepts (i.e., *70: Square Root* and *89: Ordering Fractions*) can also get promotion. After wrongly answering exercise e_{12}, we note that LPKT-S thinks that his/her knowledge state will decrease. The reason is that his/her performance on *176: Equation Solving Two or Fewer Steps* is not stable in this stage, and LPKT-S needs more interactions to modify the estimation of his/her knowledge state. Second, if the student does not practice some knowledge concepts, his/her knowledge state on these concepts will

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Science & Technology of China. Downloaded on March 15,2023 at 17:09:25 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221985

gradually reduce as time goes on. For instance, the student's knowledge state on *70: Square Root* and *89: Ordering Fractions* is dropping by degrees after answering exercise e_7 and e_{10} respectively. Third, the general evolving process of the student's knowledge state is consistent with his/her learning process. At the first learning interaction, his/her knowledge state is almost 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, because of forgetting, the student's knowledge state presents some reduction compared to the maximum but is still better than the beginning.

6.5 Student Performance Prediction

Although our goal for proposing LPKT and LPKT-S is to get more reasonable knowledge states of students, the experimental results on student performance prediction are still one of the most critical metrics for evaluating KT methods. Therefore, we compare LPKT and LPKT-S with all baselines on student performance prediction and report the average results across five test folds in Table 4. To evaluate the performance of all models comprehensively, we conduct extensive experiments on all datasets. To provide robust evaluation results, we utilize four evaluation metrics from both regression and classification perspectives in all experiments. Specifically, as a regression task, we quantify the distance between the predicted and actual performance with (1) Root Mean Square Error (RMSE) and (2) the square of Pearson correlation (r^2). Then, from the classification perspective, we adopt (3) Area Under ROC Curve (AUC) and (4) Accuracy (ACC) to measure the effectiveness. We set a threshold of 0.5 for the predictions when calculating the accuracy. From Table 4, we can see that LPKT outperforms all other KT methods on all datasets and metrics, which indicates that better results in line with students' learning process are positively related to predicting their future performance more accurately. Moreover, LPKT-S achieves better results than LPKT as it explicitly distinguishes students' personalized progress rates. It is worth mentioning that LPKT-S gets the most boost on the dataset EdNet-KT1 and the least on the dataset ASSISTchall compared to LPKT. This observation indicates that the advantage of LPKT-S compared to LPKT grows with the increase of students since it has student-specific progress rates.

6.6 Visualization of the Student Progress and Knowledge State

In LPKT and LPKT-S, we assess students' knowledge state through monitoring their learning progress, as the student progress is more instructive for learning and teaching, which tells not only the status quo of students' knowledge state but also if they are learning at a pace that will meet the target goals. To indicate that LPKT and LPKT-S have captured meaningful student progress, we give four cases of student progress and knowledge state monitored by LPKT-S and LPKT on the dataset ASSIST2012 in Figure 6. As shown in the figure, we offer the learning progress and knowledge state of four students who have practiced different knowledge concepts to gain knowledge. LPKT and LPKT-S have measured their different learning progress and knowledge state at each learning step, where we can find some progress patterns. First, as expected, LPKT and LPKT-S give positive feedback to students' knowledge states even if they got wrong answers at the beginning. The knowledge state measured by LPKT and LPKT-S is in line with students' learning process as they generally get correct answers after several failures, where the secret of success is that students can learn from mistakes. Second, students tend to learn more when they meet the knowledge concept for the first time. The learning progress is also greater in this case. When the practice times on the same knowledge concepts increase, the learning progress decreases accordingly. This phenomenon reflects a marginal utility, which can be utilized to program a proper exercise sequence to maximize learning efficiency. Third, some practices are redundant and unnecessary for students. For example, student s_1 has answered 8 times on exercises related to the knowledge concept *44*. However, he/she has mastered this knowledge concept by the first four practices and obtains almost no progress from the last four practices. Therefore, we should leave he/she more time to learn other knowledge concepts.

6.7 Ablation Experiments

In this section, we conduct ablation experiments to show how each module in LPKT and LPKT-S affects the final results. In Table 5, there are three variations of LPKT and LPKT-S, each of which takes out one module from the full model. The details are as follows:

- **LPKT_L / LPKT-S_L** refers to LPKT / LPKT-S without considering forgetting, i.e., the forgetting gate is removed.
- **LPKT_F / LPKT-S_F** refers to LPKT / LPKT-S 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_F / LPKT-S_F can only measure students' learning outcomes rather than learning gains.
- **LPKT (no time) / LPKT-S (no time)** refers to LPKT / LPKT-S that does not utilize any time information, i.e., the answer time and interval time are dropped.
- **LPKT_N / LPKT-S_N** refers to LPKT / LPKT-S that utilizes the neural network to combine the learning gain, the forgetting effect, and students' previous knowledge states, as shown in Eq. 9.

The results in Table 5 show some interesting conclusions. First, the common phenomenon of forgetting plays a critical role in the learning process, negatively affecting student progress. It can cause the most significant decline in the performance of the predictive results if we do not consider forgetting. Second, modeling the positive effect of learning gains on student progress performs better than modeling only learning outcomes. The learning gain can better reflect the dynamic increment of students' knowledge state. Third, the answer time and interval time are essential information in learning, which is harmful to accurately modeling the learning process if omitted. Finally, when combining the learning gain, the forgetting effect, and students' previous

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Science & Technology of China. Downloaded on March 15,2023 at 17:09:25 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221985

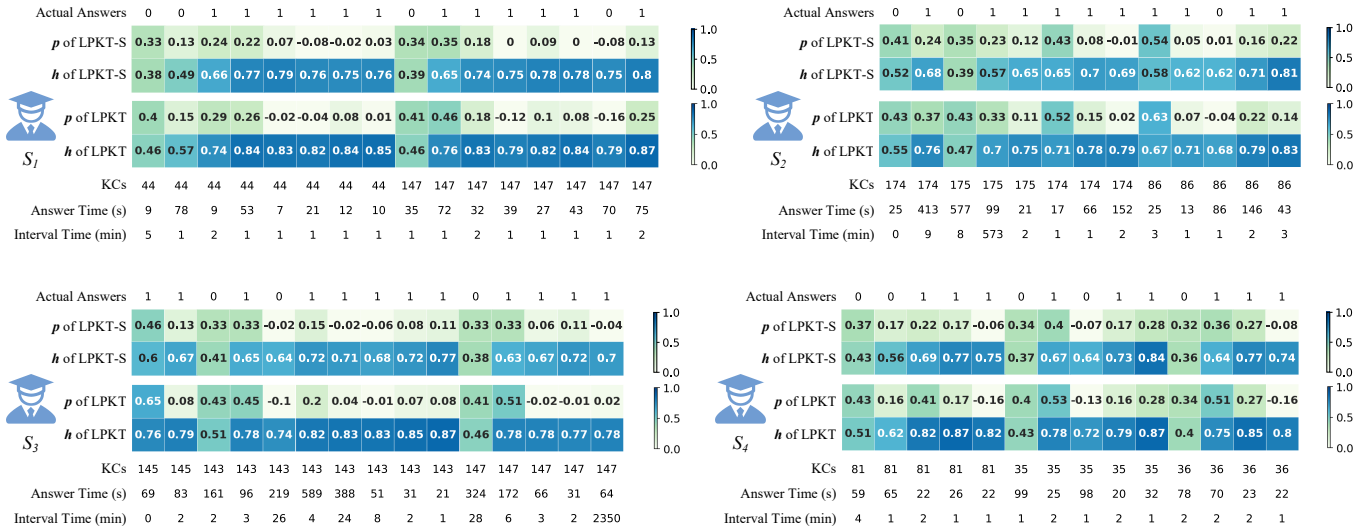


Fig. 6. Visualizations of student progress and knowledge state monitored by LPKT-S and LPKT for two students on the dataset ASSIST2012. Here the learning progress is for the corresponding knowledge at each learning step. We have normalized the learning progress to *(-1,1)*, where the larger value stands for faster progress and the negative number means a certain regress.

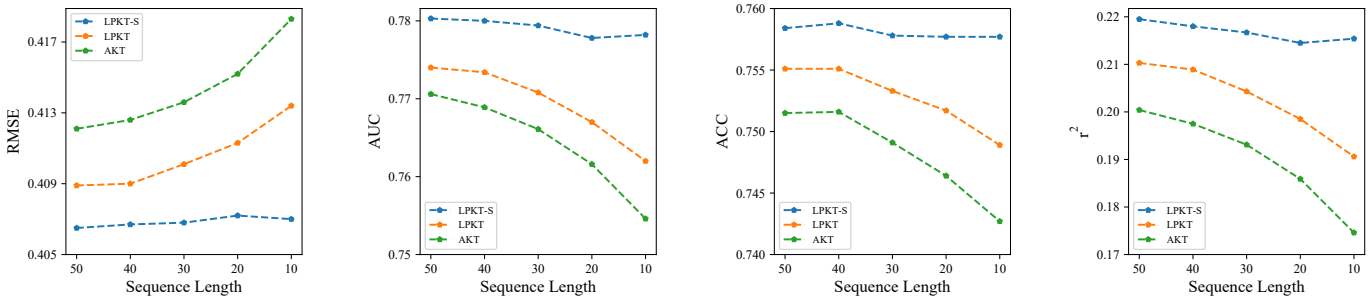


Fig. 7. Comparison results of the influence of learning sequence length of LPKT-S, LPKT, and AKT on ASSIST2012

knowledge states, the simple way of addition is more natural and effective than the neural network.

In our previous work [14], we evaluated that LPKT can better model students' learning process than the state-of-the-art KT method. In the following, we will show that LPKT-S also has similar characteristics. Generally, a longer learning sequence represents a complete learning process. Therefore, we compare the LPKT-S, LPKT, and state-of-the-art AKT on student performance prediction under different learning sequence lengths in the dataset ASSIST2012. Figure 7 indicates the comparison results. Specifically, we respectively set five lengths: 50, 40, 30, 20, and 10. Here, the shorter the learning sequence, the more incomplete the learning process. From Figure 7, 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 and better models students' learning processes. Moreover, the performance of LPKT-S shows good stability, which is almost unaffected by shorter learning sequences. The reason is that the student embedding learned by LPKT-S contained valuable student-specific information to compensate for the loss of information caused by the shorter learning sequence. In real learning environments, it is usually hard to access students' complete learning sequences. Therefore LPKT and LPKT-S have more potential application values as they have better robustness.

Besides, we have proposed the enhanced Q-matrix with the parameter γ to make up for possible omissions or mistakes in the original Q-matrix. To evaluate the impact of γ on LPKT and LPKT-S, we conduct experiments with five different values of γ: 0, 0.01, 0.03, 0.05 and 0.1. The experimental results are shown in Figure 8. As we can see from Figure 8, setting γ as a small positive value improves the performance of LPKT and LPKT-S, where we can get the maximum gain when γ is around 0.03. Specifically, when closing to zero, it is hard for γ to play the role of bridging the possible correlation between different knowledge concepts. On the other hand, if γ grows larger, there will be much more mistakes in the enhanced Q-matrix, which will damage the performance of LPKT and LPKT-S.

6.8 Exercises Clustering

The embeddings of exercises are randomly initialized in LPKT. As LPKT can get students' knowledge state with high interpretability and accuracy, the learned embeddings of exercises should also show some meaning after training. In Figure 9, we randomly choose 100 exercises from the dataset

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Science & Technology of China. Downloaded on March 15,2023 at 17:09:25 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221985

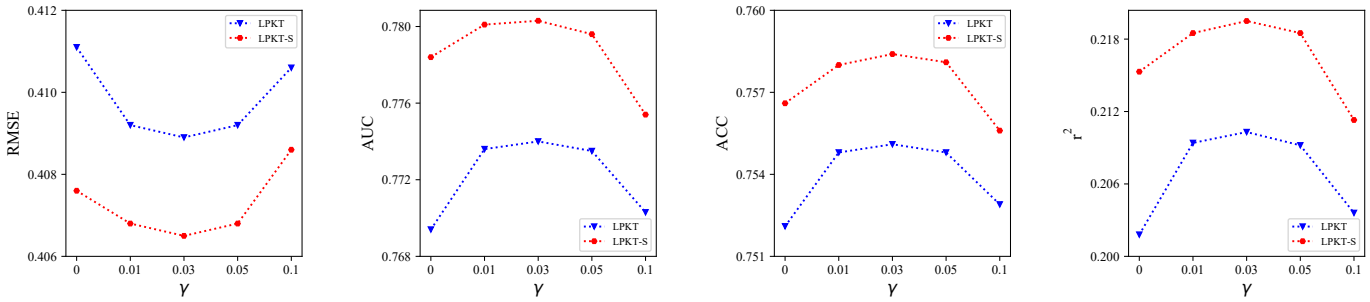
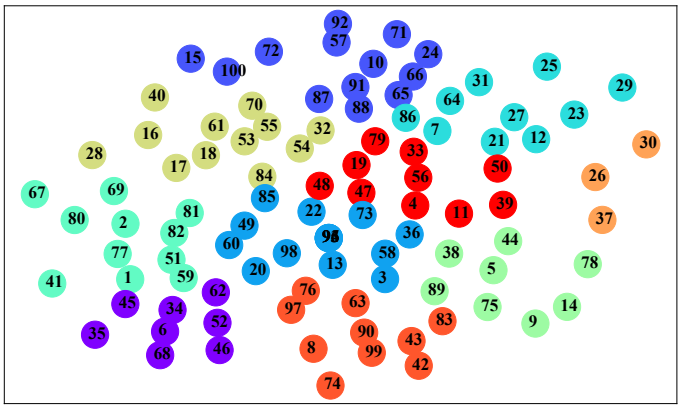


Fig. 8. The influence of the small positive value γ in the enhanced Q-matrix on the performance of LPKT and LPKT-S on ASSIST2012.



(a) The exercises clustering results.

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) The manually-labeled knowledge concepts.

Fig. 9. The exercises clustering results and the corresponding labels of KCs. We randomly selected 100 exercises from ASSISTchall and clustered them into ten concepts by k-means. Exercises with the same KC are labeled in the same color and the number stands for the exercise index.

ASSISTchall and visualize the embeddings of these exercises utilizing the T-SNE method [50]. As shown in Figure 9, we can see that the learned embeddings of exercises in LPKT can be split into ten concepts by the k-means algorithm [51] and the clustering results show well meanings. For example, the exercises 89, 95, 96, 98 with same concept *subtraction* are split together, the exercises 53, 54, 55 with same concept *evaluating-functions* are split together and the exercises 91, 92, 100 with same concept *pattern-finding* 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 educational experts.

7 Conclusions and Future Works

In this paper, we presented a novel model named Learning Process-consistent Knowledge Tracing (LPKT), which explored a new paradigm for knowledge tracing through monitoring student progress in learning. Specifically, we first defined the basic learning cell as a tuple *exercise—answer time—answer*. Then we formalized the learning process as combinations of basic learning cells and interval times. Subsequently, to monitor students' learning progress and update their knowledge state, we modeled the positive effect of the learning gain and the negative effect of forgetting in the learning process. Moreover, considering that students generally have different progress rates, we extended LPKT to LPKT-S by introducing the student embedding, which contained student-specific progress rates. Finally, we conducted extensive experiments on three public datasets to prove that LPKT and LPKT-S can get a more appropriate knowledge state consistent with students' learning process. Besides, LPKT also outperformed the state-of-the-art KT method on student performance prediction. LPKT-S was better than LPKT as it explicitly distinguished students' progress rates. Our work indicates a potential future research direction for the KT task by monitoring students' learning progress, which gives more instructive results for enhancing learning and teaching.

In the future, we will continue to explore better ways to measure students' learning progress. For example, we may use the feedback of both instructors and students about their learning process. We can also explore potential constraints in the objective function that help model learning progress. Besides, to measure more fine-grained student progress rates, we will consider assessing the progress rate on the knowledge concept level. 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 paper was an expanded version of [14], which appeared in *the proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD 2021)*. This research was partially supported by grants from the National Key Research and Development Program of China (Grant No. 2021YFF0901003), the National Natural Science Foundation of China (Grants No. U20A20229, No. 61922073, and No. 62106244).

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Science & Technology of China. Downloaded on March 15,2023 at 17:09:25 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221985

REFERENCES

[1] Ebba Ossiannilsson. Sustainability: Special issue "the futures of education in the global context: Sustainable distance education". *Sustainability*, 07 2020.

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

[3] Cristóbal Romero and Sebastián Ventura. 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):601–618, 2010.

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

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

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

[7] Dongmin Shin, Yugeun Shim, Hangyeol Yu, Seewoo Lee, Byungsoo Kim, and Youngduck Choi. Saint+: Integrating temporal features for ednet correctness prediction. *arXiv preprint arXiv:2010.12042*, 2020.

[8] Zhenya Huang, Xin Lin, Hao Wang, Qi Liu, Enhong Chen, Jianhui Ma, Yu Su, and Wei Tong. Disenqnet: Disentangled representation learning for educational questions. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 696–704, 2021.

[9] Ting Long, Jiarui Qin, Jian Shen, Weinan Zhang, Wei Xia, Ruiming Tang, Xiuqiang He, and Yong Yu. Improving knowledge tracing with collaborative information. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 599–607, 2022.

[10] Qi Liu, Shuanghong Shen, Zhenya Huang, Enhong Chen, and Yonghe Zheng. A survey of knowledge tracing. *arXiv preprint arXiv:2105.15106*, 2021.

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

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

[13] Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Hui Xiong, Yu Su, and Guoping Hu. Ekt: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering*, 33(1):100–115, 2019.

[14] Shuanghong Shen, Qi Liu, Enhong Chen, Zhenya Huang, Wei Huang, Yu Yin, Yu Su, and Shijin Wang. Learning process-consistent knowledge tracing. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1452–1460, 2021.

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

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

[17] Safer, Nancy, Fleischman, and Steve. How student progress monitoring improves instruction. *Educational Leadership*, 2005.

[18] K. Mclane. What is student progress monitoring and how will it help me?. *National Center on Student Progress Monitoring*, page 4, 2006.

[19] Ye Mao. Deep learning vs. bayesian knowledge tracing: Student models for interventions. *Journal of educational data mining*, 10(2), 2018.

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

[21] Philip I Pavlik Jr, Hao Cen, and Kenneth R Koedinger. Performance factors analysis–a new alternative to knowledge tracing. *Online Submission*, 2009.

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

[23] Shuanghong Shen, Qi Liu, Enhong Chen, Han Wu, Zhenya Huang, Weihao Zhao, Yu Su, Haiping Ma, and Shijin Wang. 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*, pages 1857–1860, 2020.

[24] Shalini Pandey and George Karypis. A self-attentive model for knowledge tracing. *arXiv preprint arXiv:1907.06837*, 2019.

[25] Shalini Pandey and Jaideep Srivastava. Rkt: Relation-aware self-attention for knowledge tracing. CIKM '20, page 1205–1214, New York, NY, USA, 2020.

[26] Aritra Ghosh, Neil Heffernan, and Andrew S. Lan. Context-aware attentive knowledge tracing. KDD '20, page 2330–2339, New York, NY, USA, 2020.

[27] Shuanghong Shen, Zhenya Huang, Qi Liu, Yu Su, Shijin Wang, and Enhong Chen. Assessing student's dynamic knowledge state by exploring the question difficulty effect. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 427–437, 2022.

[28] Robin Schmucker, Jingbo Wang, Shijia Hu, and Tom M Mitchell. Assessing the knowledge state of online students–new data, new approaches, improved accuracy. *arXiv preprint arXiv:2109.01753*, 2021.

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

[30] R Luckin et al. 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*, pages 349–356, 2007.

[31] Richard R Hake. 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*, volume 8, pages 1–14, 2002.

[32] Zhenya Huang, Qi Liu, Yuying Chen, Le Wu, Keli Xiao, Enhong Chen, Haiping Ma, and Guoping Hu. Learning or forgetting? a dynamic approach for tracking the knowledge proficiency of students. *ACM Transactions on Information Systems (TOIS)*, 38(2): 1–33, 2020.

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

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

[35] Koki Nagatani, Qian Zhang, Masahiro Sato, Yan-Ying Chen, Francine Chen, and Tomoko Ohkuma. Augmenting knowledge tracing by considering forgetting behavior. In *WWW*, pages 3101–3107. ACM, 2019.

[36] Sip, J., and Pijl. Practices in monitoring student progress. *International Review of Education*, 38(2):117–131, 1992.

[37] Michael Mogessie Ashenafi, Marco Ronchetti, and Giuseppe Riccardi. Predicting student progress from peer-assessment data. In *The 9th International Conference on Educational Data Mining EDM 2016*, 2016.

[38] Wenling Li, Yingmin Jia, and Junping Du. Distributed filtering for discrete-time linear systems with fading measurements and time-correlated noise. *Digital Signal Processing*, 60:211–219, 2017.

[39] Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yuying Chen, Yu Yin, Zai Huang, and Shijin Wang. Neural cognitive diagnosis for intelligent education systems. In *AAAI 2020*, 2020.

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

[41] Chuan Shi, Xiaotian Han, Li Song, Xiao Wang, Senzhang Wang, Junping Du, and S Yu Philip. Deep collaborative filtering with multi-aspect information in heterogeneous networks. *IEEE transactions on knowledge and data engineering*, 33(4):1413–1425, 2019.

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

[43] Chun Kit Yeung and Dit Yan Yeung. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. 2018.

[44] Shi Pu and Lee Becker. Self-attention in knowledge tracing: Why it works. In *International Conference on Artificial Intelligence in Education*, pages 731–736. Springer, 2022.

[45] Jan D Vermunt and Yvonne J Vermetten. Patterns in student learning: Relationships between learning strategies, conceptions of

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Science & Technology of China. Downloaded on March 15,2023 at 17:09:25 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3221985

learning, and learning orientations. *Educational psychology review*, 16(4):359–384, 2004.
[46] Mingyu Feng, Neil Heffernan, and Kenneth Koedinger. Addressing the assessment challenge with an online system that tutors as it assesses. *USER-ADAP*, 19(3):243–266, 2009.
[47] Youngduck Choi, Youngnam Lee, Dongmin Shin, Junghyun Cho, Seoyon Park, Seewoo Lee, Jineon Baek, Chan Bae, Byungsoo Kim, and Jaewe Heo. Ednet: A large-scale hierarchical dataset in education. In *International Conference on Artificial Intelligence in Education*, pages 69–73. Springer, 2020.
[48] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010.
[49] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.
[50] Van Der Maaten Laurens and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2605):2579–2605, 2008.
[51] Liping Jing, Michael K Ng, and Joshua Zhexue Huang. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Transactions on knowledge and data engineering*, 19(8):1026–1041, 2007.

Shuanghong Shen recieved the B.E. degree from Wuhan University, Wuhan, China, in 2018. He is currently working toward the Ph.D. degree in the School of Computer Science and Technology, University of Science and Technology of China. His main research interests include data mining, knowledge discovery, natural language processing and intelligent tutoring systems. He won the first prize in task 2 of the NeurIPS 2020 Education Challenge. He has published papers in referred conference proceedings, such as SIGIR2020, KDD 2021, AAAI 2022, SIGIR2022.

Enhong Chen (SM'07) received the B.S. degree from Anhui University, Hefei, China, the M.S. degree from the Hefei University of Technology, Hefei, China, and the Ph.D. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China, in 1989, 1992 and 1996 respectively. He is currently a Professor and the Executive Dean of School of Data Science, the Director of Anhui Province Key Laboratory of Big Data Analysis and Application. He has published a number of papers on refereed journals and conferences, such as TKDE, TIST, TMC, KDD, ICDM, NIPS and CIKM. His current research interests include data mining and machine learning, social network analysis, and recommender system. Dr. Chen was a recipient of the National Science Fund for Distinguished Young Scholars of China, the Best Application Paper Award on KDD-2008, the Best Student Paper Award on KDD-2018 (Research), and the Best Research Paper Award on ICDM-2011. He is a senior member of the IEEE.

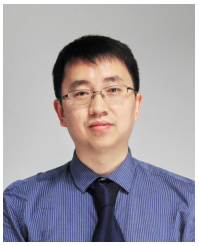
Qi Liu received the Ph.D. degree from University of Science and Technology of China (USTC), Hefei, China, in 2013. He is currently a Professor in the School of Computer Science and Technology at USTC. His general area of research is data mining and knowledge discovery. He has published prolifically in refereed journals and conference proceedings (e.g., TKDE, TOIS, KDD). He is an Associate Editor of IEEE TBD and Neurocomputing. He was the recipient of KDD' 18 Best Student Paper Award and ICDM' 11 Best Research Paper Award. He is a member of the Alibaba DAMO Academy Young Fellow. He was also the recipient of China Outstanding Youth Science Foundation in 2019.

Zhenya Huang recieved the B.E. degree from Shandong University, Ji'nan, China, in 2014 and the Ph.D. degree from University of Science and Technology of China, Hefei, China, in 2020. He is currently working as an associate researcher of the School of Computer Science and Technology, University of Science and Technology of China. His main research interests include data mining, knowledge discovery, representation learning and intelligent tutoring systems. He has published more than 30 papers in refereed journals and conference proceedings, including TKDE, TOIS, AAAI2018, CIKM2019, KDD2021.

Wei Huang received his B.E. degree in software engineering from Sichuan University (SCU), Chengdu, China, in 2017. He is currently working toward his Ph.D. degree in the School of Data Science, University of Science and Technology of China (USTC), Hefei. His research interests include data mining, deep learning, natural language processing and applications in text classification, such as patent annotation. He has published papers in referred conference proceedings, such as CIKM'2019, ACM MM'2020, ICDM'2020, IJCAI'2021 and KDD'2021.

Yu Yin received the B.E. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China, in 2017. He is currently working toward the Ph.D. degree in the School of Computer Science and Technology, USTC. His main research interests include data mining, intelligent education systems, and image recognition. He won the first prize in the Second Student RDMA Programming Competition in 2014. He has published papers in referred conference proceedings, such as AAAI'2018, KDD'2018 and KDD'2019.

Yu Su received the Ph.D. degree from Anhui University, Hefei, China. He is currently working as an associate professor of the School of Computer Science and Technology, Hefei Normal University. His main area of research includes data mining, machine learning, recommender systems, and intelligent education systems. He has published several papers in referred conference proceedings and journals, such as IJCAI'2015, AAAI'2018, ICDM'2020, KDD'2020, KDD'2021, and ACM TIST.



Shijin Wang received the B.E. degree in computer science from University of Science and Technology of China (USTC), Hefei, China, in 2003 and the Ph.D. degree from Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2008. He is the Vice Dean of the AI Research, IFLYTEK CO., LTD. His main area of research includes artificial intelligence, pattern recognition, data minning, and intelligent education system. He has published several papers in referred conference proceedings and journals, such as ICDM'2020, SIGIR'2020, WSDM'2021, AAAI'2021, and KDD'2021.

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: University of Science & Technology of China. Downloaded on March 15,2023 at 17:09:25 UTC from IEEE Xplore. Restrictions apply.