A Dynamic Knowledge Diagnosis Approach Integrating Cognitive Features

TAO HUANG1, HUALI YANG2, ZHI LI2, HEKUN XIE1, JING GENG1, AND HAO ZHANG1

1National Engineering Laboratory for Educational Big Data, Central China Normal University, Wuhan 430079, China
2National Engineering Research Center for E-Learning, Central China Normal University, Wuhan 430079, China

Corresponding author: Huali Yang (yanghuali@mail.ccnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61977033, and in part by the Central China Normal University’s Basic Natural Science Research Project, in 2020, under Grant 30106200505.

ABSTRACT The rapid development of data analytic technologies has advanced personalized learning and increased its popularity in K-12 education significantly. Specifically, one fundamental step in personalized learning is knowledge proficiency diagnosis which reveals blind spots in students’ knowledge. However, existing approaches to diagnosis either exploit data from a one-time assessment for the cognitive diagnosis (ignoring the previous historical interactions) or trace the knowledge state using recurrent neural networks to predict students’ future performance (ignoring the cognitive features). To this end, this study proposes a dynamic approach to knowledge diagnosis integrating cognitive features with a key-value memory network to store latent exercise information and capture long-term temporal features based on cognitive psychology. Specifically, given the characteristics of assessment data in China, our approach mainly aims to model sequence data with cognitive features, including forgetting and learning. Two corresponding gates are used to weaken the knowledge memory and strengthen the repeated knowledge memory over time, respectively, in the memory updating process. Finally, to evaluate our approach, we conducted extensive experiments on four real-world datasets collected from K-12 education. The results show that the approach can effectively process the time sequence in education, whose prediction results are better and more stable than other existing baseline models. We also conducted experiments for parameter sensitivity, different feature integration methods, and the effectiveness of cognitive features to ensure that the models achieved the best results. The application visualization further confirms the practicability of our approach in dealing with problems of dynamic knowledge diagnosis.

INDEX TERMS Cognitive features, dynamic knowledge diagnosis, key-value memory, performance prediction.

I. INTRODUCTION

With the rapid development of deep learning and big data technology in recent years, there have emerged many intelligent tutoring systems (ITS), such as KNEWTON, ALEKS, and ASSESSMENT, that can provide students personalized learning services including open access to millions of online education resources [1]–[3]. Due to their convenience and importance for education, these platforms have attracted a large number of teachers and students.

In fact, knowledge proficiency diagnosis [4] is a crucial step for personalized learning, with the goal of detecting students’ hidden state (mastery level) on each knowledge area in the exercise interaction process. Specifically, the results of diagnosis can help students discover areas of knowledge weakness and then obtain personalized services, such as targeted knowledge training [3], dynamic planning of learning paths [5], and personalized resource recommendations [6].

Fig. 1 shows an example of the interaction process of typical Chinese students on K12 math subjects over time. We can see that two students (S1 and S2) practice different exercises to learn three knowledge concepts (K1: Integers, K2: Fractions, and K3: Irrational numbers) from March to April, where the exercises are organized by simultaneous assessments. The knowledge corresponding with an exercise is usually annotated by educational experts. In practice, the main task in such educational platforms is to predict student performance [7]; that is, to forecast whether a student can answer exercises...
(e.g., e7, e8) correctly in a future assessment. Meanwhile, it also requires tracking the change in students’ knowledge mastery level [8] (K1, K2, and K3) in their assessment process.

In the literature, there have been a series of research achievements for knowledge proficiency diagnosis, such as cognitive diagnosis models (CDMs) [9], and knowledge tracing (KT) [1]. However, most research ignores two main factors: cognitive features and latent features of exercises, which have an important influence on students’ learning in the assessment process.

On one hand, in the cognitive psychology field, students’ knowledge mastery level evolves. Thus, knowledge proficiency is affected by two cognitive features: forgetting [10]–[12] previous knowledge and learning [13]–[15] the same knowledge. The forgetting theory posits that students’ memory of knowledge learned decreases as time goes on, such that their knowledge proficiency is correlated with the time interval factor. For example, in Fig. 1, as student S1 did not practice exercises related to knowledge K2 over 26 days in the last assessment, which has a time interval longer than knowledge K1 (5 days), the student may perform worse on e8 (K2) than e7 (K1). Learning theory emphasizes that if a student learns the same task repeatedly, their understanding of the task will be strengthened. For instance, in Fig. 1, as student S1 strengthened their memory of knowledge K1 by reviewing it more times, there is a greater probability they will answer correctly on exercise e7 (3 times) than on e8 (1 time). Fortunately, several studies [13]–[15], [17], [18] have attempted to dynamically diagnose knowledge proficiency by adding cognitive factors, which perform better in the experiment. However, these models have some limitations. In particular, DKT+forget (DKT+F) [10] considers several factors of forgetting to capture students’ hidden knowledge mastery state over time simply by integrating these factors into recurrent neural networks (RNN), ignoring the deep influence on the neural network structure. In summary, our work aims to combine the two cognitive factors and students’ exercise records to better track knowledge proficiency dynamically.

On the other hand, the exercise consisting of the same knowledge may have different latent features [8], [19], [20], such as difficulty. In fact, most existing knowledge tracing methods, such as Bayesian knowledge tracing (BKT) [21] and DKT [22], usually use the corresponding knowledge instead of exercises themselves, ignoring important latent exercise features in their exercise process. For example, even though exercises e1, e3, and e5 have the same knowledge, student S2 performed conversely on e1 and e3 in assessment 1 due to the latent exercise feature (i.e., difficulty). This student is more likely to perform worse on K1 in future assessment 3 than on exercise e3 with a similar latent feature. Although some prior studies have considered the latent feature, such as dynamic key-value memory networks (DKVMN [19]), and sequential key-value memory networks (SKVMN [20]), to the best of our knowledge, few have noticed the effect of updating the latent feature state covering the same knowledge with a different time interval for tracking knowledge proficiency.

In summary, in this work, we focus on several limitations for knowledge proficiency diagnosis as follows. First, due to the complexity and changeability of students’ cognitive processes, how do we quantitatively extract common cognitive features from the complex, previous historical interactions? How do we integrate these cognitive features (e.g., learning, memorizing, and forgetting) for the knowledge proficiency diagnosis task to improve the accuracy of the prediction results?

Hence, to address the above challenges for knowledge proficiency diagnosis, we proposed a dynamic knowledge
diagnosis approach integrating cognitive features (CF-DKD) to predict student performance by incorporating learning and forgetting theories with the long-term data of the historical interaction sequence. The major contributions of this article are as follows.

1. Although a key-value memory (i.e., DKVMN) can help trace the knowledge states by implementing add gate and erase gate [19], it is not effective in modeling the long-term dependencies. Therefore, we solve this issue by incorporating cognitive features (forgetting and learning information) into the memory networks to track dynamic changes in knowledge mastery states over time.

2. We attempt three interaction methods to combine the cognitive features and response interactions by encoding a unified feature in advance. Thus, it enhances the model accuracy for student performance predictions by exploring the optimized combination of all features.

3. We extend the updating process of the memory network through the forget and learn gate mechanism, where the forget gate erases the long-term, useless dependent information, and the learn gate enhances the short-term information and long-term dependent memory.

We have adopted a 5-fold cross-validation method to evaluate the CF-DKD model extensively on four established datasets and compare them with state-of-the-art models. Experiments show that the CF-DKD model is superior to other baselines for predicting student performance.

The remainder of this paper is organized as follows. Section II discusses related work on cognitive diagnosis. Section III presents our proposed diagnostic model CF-DKD. Section IV explains the experimental design, results, and the application of knowledge diagnosis in education. Section V is the conclusion.

II. RELATED WORK

This section provides a brief review of the relevant literature from four aspects: cognitive diagnosis, dynamic knowledge diagnosis, diagnosis with memory network, and knowledge diagnosis with cognitive features.

A. COGNITIVE DIAGNOSIS

Existing studies on cognitive diagnosis models such as Item Response Theory (IRT) and the Deterministic Input Noisy Output “AND” gate (DINA) model mainly focus on discovering students’ knowledge proficiency by predicting students’ responses. IRT [16], which is a unidimensional model with exercise distinction and difficulty parameters, uses a logistic-like function to model students’ latent traits. In DINA [23], the Q-matrix is labeled by a human to present the correlation between exercise and knowledge, such that students’ knowledge proficiency can be achieved by a function containing two parameters (i.e., guessing and slipping). However, to the best of our knowledge, all of these methods just focus on exercise interaction data from one assessment, ignoring the fact that knowledge proficiency is changeable over time with previous historical temporal records, let alone considering cognitive features for more precise diagnosis.

B. DYNAMIC KNOWLEDGE DIAGNOSIS

To overcome the challenges faced by original CDMs, researchers have attempted to propose many dynamic CDMs, considering the time factors in the interaction sequence. Particularly, the variations in CDMs combine more information, such as the time interval factor [16], [25] and response time [13], into the modeling. In addition, another representative work is DKT, which is a popular sequence model to trace students’ knowledge states by using deep learning [26] over time. DKT could capture the changes of students’ latent knowledge state through a hidden layer in RNNs with no human-labeled annotation for the prediction task. In fact, some extensions further considered other factors, such as forgetting [10], exercise content [28], or complex exercise interactions [4]. Despite the importance of these efforts, there are still some limitations in practice: First, CDMs only introduce a response time variable so they cannot discover the historical knowledge mastery state over time. Second, although DKT has considered temporal factors by the forget gate, as knowledge is learned at different times, the one hidden layer means the time interval is identical between two adjacent interactions, causing severe time information loss. In summary, existing dynamic models neglect the influence of temporal factors in the exercise process, which does not easily explain the degree of forgetting or learning with a different time interval.

C. DIAGNOSIS WITH MEMORY NETWORK

Recently, researchers have attempted to leverage the use of memory networks for student performance prediction. Memory-augmented neural networks (MANN) [29] introduce an external memory matrix to store information, which extends the one hidden layer in DKT. In order to enhance the memory representation, a key-value memory network [30] is proposed for complex problems, which allows the storage of static data in a key memory and dynamic data in a value memory. For the knowledge diagnosis task, memory network technique is first used in DKVMN [19], which exploits the relationships between the exercise and latent concepts (correlation weight) to directly output a student’s mastery level of each concept (read process). Specifically, the key memory matrix stores the knowledge, and the other dynamic matrix stores the mastery state of the corresponding concept, which is updated using the response on future exercises (write process). On the basis of DKVMN, many extensions were conducted by introducing other factors, such as long-time dependence among the same knowledge in SKVMN [20], students’ ability cluster in DSCMN [31], and knowledge tag in DKVMN_CA [32]. Experimental results show that DKVMN-based models can achieve better prediction accuracy because they can discover the correlation between latent concepts and exercises better and capture the sequential dependencies in the interaction sequence effectively.
Currently, many scholars have introduced graph neural networks in knowledge modeling. For instance, Nakagawa et al. [24] uses a graph convolutional network to represent the correlation between knowledge, and updates the latent knowledge state by a weak-forgetting and weak-learning gate mechanism, thereby finally achieving good performance. Although both interpretability and performance are good, these models consider forgetting or learning factors just by using the erase gate and add gate, whose effectiveness remains weak, ignoring the time interval information. In our work, we retain the key-value memory network structure in our CF-DKD model because of the superior representation of the key-value memory; thus, more cognitive features are needed to enhance the prediction performance.

D. KNOWLEDGE DIAGNOSIS WITH COGNITIVE FEATURES

Modeling students’ knowledge diagnosis with cognitive features is one of the key issues in cognitive psychology [25], [33] and has been discussed for decades. In the cognitive processes, students’ knowledge mastery will decrease as time goes on (forgetting) and become enhanced with repeated practice (learning) [15], [27]. On the one hand, a typical forgetting theory is the Ebbinghaus forgetting curve [34], which shows that students will forget the mastered knowledge at an exponential rate as the time interval increases. On the other hand, a learning curve [34] is a mathematical description of students’ performance in repetitive knowledge, which explains that with more repetitions, learners tend to demand less time to achieve good performance because of familiarity. Overall, knowledge diagnosis with cognitive features could be grouped into the following three categories.

Traditional probabilistic diagnosis models with cognition, such as the original BKT model, assume that students never forget once they acquire knowledge, which does not conform to the forgetting rules in cognitive psychology. Qiu proposed the “KT-Forget” [18] model that introduces the forgetting factor as a parameter into the model, and assumes that time interval by unit of “days” will affect the parameter (i.e., students may forget the mastered knowledge learned before a few days). Mohammad et al. [17] models the learning process by adding similar knowledge counts to the model, which assumes that the greater the repetition of similar techniques, the stronger the memory retained. Experiments show that it performs better than more advanced deep learning models (e.g., DKT).

First proposed by Pelánek [13], performance factor analysis (PFA) with cognition introduces a time effect function to express the influence of memory by lag time. Settles and Meeder [14] proposed a PFA extended model via the half-life regression, based on the Ebbinghaus forgetting curve by modeling repeated exercises at different time intervals, which partly considers the forgetting factor. Furthermore, many other researchers focus on the forgetting curve and learning curve [26], [35], [27]. In particular, Schmidhuber [26] proposed an interpretable probability matrix factorization framework using the two curves to track students’ knowledge proficiency, which is more accurate than other CDMs (i.e., BKT or DINA).

Moreover, through the implementation of deep learning with cognition, several attempts have been made to observe the performance result. Nagatani et al. [10] assumes that knowledge memory retention is mainly related to two forgetting behaviors: the lag time from the previous record and the number of past trials on the same knowledge. Ghosh et al. [36] adopts the attention mechanism in attentive knowledge tracing to obtain the context-awareness of the exercise with an exponential decay to model students’ forgetting behavior and ability.

Despite the good interpretability of these models, there are still some limitations. For one thing, the time interval between similar knowledge plays a very important role in forgetting, whereas some other learning features (i.e., repetitions) that could enhance memory are usually ignored. Also, due to the complexity of calculation, it is difficult to incorporate other
interactive information into the long-term temporal sequence in deep learning models. In contrast, our models improve deep learning models by incorporating cognitive factors (i.e., learning feature, forgetting feature) and memory network, which strengthen some important features, weaken unimportant ones, and track long-term memory dynamically.

III. METHOD

A. PROBLEM DEFINITION

CF-DKD is a supervised learning approach to track students’ knowledge proficiency through their exercise response sequence in each assessment over time.

The task of the CF-DKD is described as follows: let the sequence of $X = \langle x_0, x_1, \ldots, x_{t-1} \rangle$ be a student’s response interactions with exercises at different times. The response interaction cell $x_t = (e_t, r_t, f_t, l_t)$ is a four-item tuple, representing that a student performs an exercise $e_t (e_t \in \{1, \ldots, |E|\})$ at time $t$. The corresponding response is $r_t$, which is a binary variable, $r_t \in \{0, 1\}$. When a student answers correctly, $r_t = 1$; otherwise, $r_t = 0$. In addition, we describe the cognitive factors with $f_t, l_t$ representing the forgetting feature and learning feature, respectively. Thus, the definition of the problem is described as follows.

Definition 1: Regarding the knowledge proficiency diagnosis problem, given the sequence of student-exercise interactions and the cognitive features from step 1 to $t$, our goal is two-fold: (1) Predict students’ response $r_t$ to the next exercise $e_t$, i.e., the probability $p(r_t = 1 | e_t, X)$ of getting a correct answer on the next exercise $e_t$; (2) Diagnose the knowledge proficiency $M_{stu}$ of students on different knowledge.

Further, to trace the changes in knowledge proficiency dynamically in the diagnosis process, we propose two assumptions:

Assumption 1: We assume that if a student interacts with an exercise regardless of whether they respond correctly, they will enhance the memory of the related knowledge concept through the teacher’s explanation.

Assumption 2: An exercise may be associated with multiple knowledge concepts, but we assume that the first knowledge concept is the most important for the exercise. Thus, we select the first one to calculate the cognitive features by default.

B. OUR SOLUTION WITH CF-DKD FRAMEWORK

An overview of the proposed solution is illustrated in Fig. 2. As shown in the figure, given all students’ response interaction record $\{(e_1, r_1), (e_2, r_2), \ldots, (e_T, r_T)\}$ and the corresponding learning and forgetting factor record $\{(f_1, l_1), (f_2, l_2), \ldots, (f_T, l_T)\}$, we propose a dynamic knowledge diagnosis approach integrating cognitive features (CF-DKD) to trace the change in students’ latent knowledge mastery state. Specifically, we then conduct two applications with the trained models, such as predicting students’ response on the next exercises for the subsequent period considering cognitive factors, and obtaining a stable knowledge mastery state with the historical interaction sequences.

In the following discussion, we specify the knowledge modeling process of our proposed approach. CF-DKD is a time series model that employs a memory network to store latent knowledge proficiency dynamically, which provides an interface for reading and writing the hidden information based on the cognitive rules. The critical processes at step $t$ of CF-DKD are summarized in Fig. 3, which consists of four layers as follows:

- The embedding layer is responsible for representing various types of input data uniformly into a multi-dimensional vector space via one-hot encoding or embedding.
- The memory layer involves two processes: key reading and value reading. Our goal in the key reading process is to allocate the relevant hidden knowledge weight

![FIGURE 3. The CF-DKD framework in one step.](image-url)
through the addressing mechanism of the input exercise $e_i$, and the corresponding latent knowledge from the key matrix $M'_{stu}$. In the value reading process, we can retrieve the current latent knowledge state by the relevant weight from the value matrix $M_{stu}$.

- The updating layer provides two kinds of gates to update the state of the hidden knowledge $M'_{stu}$.
- The predicting layer provides three integration methods for multiple features and uses a fully connected neural network to predict the probability of the learner’s performance at the next step with the integration vector.

### C. COGNITIVE FEATURE EXTRACTION

Thus, CF-DKD enhances the students’ mastery of knowledge by introducing forgetting and learning factors. The calculation method is shown in Fig. 4. Three assessments at time t1, t2, and t3 take place on March 10, March 31, and April 4, respectively (Fig. 1). The exercise record of student S1 is used as an example. Obviously, exercises in these assessments are associated with different knowledge concepts where the same knowledge concept is marked by one color. $t_{mn}$ represents the time interval between the timepoint $m$ and timepoint $n$, $\Delta t_{mn} = t_n - t_m$ where $m > n$.

1) **FORGETTING FEATURES**

Inspired by the DKT+F [10] model, the forgetting feature involves sequence time interval ($\Delta s_t$) and repeated time interval ($\Delta r_t$).

The former ($\Delta s_t$) is the time interval between the current and last assessment (two adjacent assessments). Due to the latent relevance among the exercises at different assessments, the student is more likely to perform better with a shorter sequence time interval. For instance, as shown in Fig. 4, the student has a shorter sequence time interval on exercise e7 than exercise e5 with the same knowledge K1 as described by the following formulas:

$$\Delta s_{t=2, k=K1, e=e5} = \Delta t_{21} = t_2 - t_1 = 21 \text{ days},$$
$$\Delta s_{t=3, k=K1, e=e7} = \Delta t_{32} = t_3 - t_2 = 5 \text{ days},$$  \hspace{1cm} (1)

Thus, the possibility of answering correctly on e7 is greater than on e5.

The latter ($\Delta r_t$) is the minimum time interval of the same knowledge concept between the current and previous assessments, reported in previous studies [13], [14]. For example, as shown in Fig. 4, the student has a shorter repeated time interval on knowledge K1 than K2 as described by the following formulas:

$$\Delta r_{t=3, k=K1, e=e7} = \Delta t_{31} = t_3 - t_1 = 26 \text{ days},$$  \hspace{1cm} (2)

Thus, the possibility of answering correctly on e7 (corresponding knowledge is K1) for student S1 is greater than on e8 (corresponding knowledge is K2).

In summary, according to the forgetting theory [10], the longer the time interval ($\Delta r_t$ or $\Delta s_t$), the greater the probability that a student will forget, and the poorer the performance for this student on the next exercise. Then, we select the maximum value of the variables, $\Delta s_t$, $\Delta r_t$, respectively, as the dimension of the one-hot encoding. Therefore, the forgetting feature $f_t = [\Delta s_t, \Delta r_t]$ is represented by concatenating the two features.

2) **LEARNING FEATURES**

According to assumption 1, the student acquires knowledge each time they answer an exercise, which is called the learning process. In particular, learning features ($l_t$) are determined by past trial counts ($\Delta c_t$), $l_t = \Delta c_t$, which are the repeated counts with the same knowledge in the previous interactive sequence. If a student practices the same knowledge more frequently, their memory of that knowledge will be

| Features | variables | e1 | e2 | e3 | e4 | e5 | e6 | e7 | e8 |
|----------|-----------|----|----|----|----|----|----|----|----|
| Forgetting features | $\Delta s_t$ | - | - | - | - | - | - | - | - |
| Learning features | $\Delta c_t$ | 0 | 0 | 2 | 0 | 3 | 1 | - | - |

Note: if $\Delta t_{21}=21$ days, $\Delta t_{32}=5$ days, then $\Delta t_{31}=26$ days.

**FIGURE 4.** Cognitive feature extraction.
more profound. According to the learning theory [13], larger $c_t$ means memory enhancement of the knowledge. In Fig. 4, the student learns K1 more frequently ($\Delta c_1 = 3$ times) than K2 ($\Delta c_1 = 1$ times); thus, they would perform better on e7 (corresponding knowledge is K1) than e8 (corresponding knowledge is K2).

D. MEMORY LAYER

As shown in Fig. 3(b), we use key-value matrix pairs to store the knowledge mastery state in our CF-DKD model rather than a single hidden layer in the more traditional DKT model, which is represented by $M_{exe}$ and $M_{stu}$. $M_{exe}$ is an immutable key matrix to store latent knowledge, and $M_{stu}$ is a dynamic value matrix that stores the knowledge proficiency of each student. These two memory matrices have the same slots, each of which represents a latent knowledge. The memory layer consists of two steps:

1) KEY READING

Given the exercise input $e_t$ ($e_t \in E$) at step $t$, as shown in Fig. 3(a), we first encode the exercise with a one-hot vector, where $E$ is the exercise set and $|E|$ represents the number of the exercise. Due to the sparseness of the one-hot vector, we then map it into a dense space, with $e_t$ multiplied by an embedding matrix $A \in \mathbb{R}^{|E| \times d_k}$ to obtain a continuous vector $k_t \in \mathbb{R}^{d_k}$:

$$k_t = e_t \ast A.$$  

In order to obtain the correlation between the current exercise and latent knowledge, we employ the attention mechanism by calculating the inner product vector of the current exercise embedding vector $k_t$ and the key matrix $M_{exe}$ as follows:

$$w_t (i) = \text{softmax} \left( k_t^T M_{exe} (i) \right),$$

where $\text{softmax}(z_i) = e_i / \sum_{i=1}^{n} e_i$, and $w_t (i) \in [0, 1]$.

2) VALUE READING

Given the correlation weight $w_t$, we retrieve the latent knowledge states at exercise $e_t$ from the value matrix $M_{stu}$ for the student. Thus, the exercise mastery state $read_t$ is computed by the weighted sum of all related latent knowledge states multiplied by the corresponding correlation weight:

$$\text{read}_t = \sum_{i=1}^{N} w_t (i) M_{stu} (i),$$

E. UPDATING

According to the forgetting theory and learning theory, a student’s knowledge proficiency is influenced by exercise, response, and corresponding cognitive factors. However, how can we use these factors to update the hidden state?

First, after a student practices a new exercise, the updating process occurs to update the student’s latent knowledge state from $M_{stu}^{t-1}$ to $M_{stu}^t$ according to the objective response $r_t$ of the student on the exercise $e_t$, as shown in Fig. 3(c). Thus, we set the response interaction tuple $(e_t, r_t)$ to be a one-hot encoding. Since $r_t$ is a binary variable $r_t \in \{0, 1\}$, we can extend the response $r_t$ to a 0 vector with the same $|E|$ dimensions as $e_t$ [28]. Thus, the combined vector is calculated as

$$(e_t, r_t) = \begin{cases} \{e_t, 0\}, & \text{if } r_t = 1, \\ \{0, e_t\}, & \text{if } r_t = 0, \end{cases}$$

where $(e_t, r_t) \in \mathbb{R}^{2|E|}$ combines both the exercise and the response to a unified vector. Given the sparseness of one-hot encoding, we embed $(e_t, r_t)$ with an embedding matrix $B \in \mathbb{R}^{2|E| \times d_v}$ to obtain a dense vector $v_t \in \mathbb{R}^{d_v}$:

$$v_t = (e_t, r_t) \ast B.$$
TABLE 1. Integration methods for the exercise and cognitive features.

| Integration methods          | Expression                  | Instance: predicting     |
|------------------------------|----------------------------|--------------------------|
| Concatenation                | $\varphi(a, b) = [a, b]$   | $\varphi_{in} = [k_i, f_i, l_i]$ |
| Multiplication               | $\varphi(a, b) = [a \odot b]$ | $\varphi_{in} = [k_i \odot [f_i, l_i]]$ |
| Concatenation + Multiplication| $\varphi(a, b) = [a \odot [a, b]]$ | $\varphi_{in} = [k_i \odot [f_i, l_i], [f_i, l_i]]$ |

learners’ knowledge mastery. Second, as shown in Fig. 5 (b), although a variety of forgetting variables have been incorporated to affect the updating process in the DKT+I model, only one hidden layer is not able to accurately represent the real hidden learning process by the RNNs. Finally, in Fig. 5 (c), we use an extra key-value memory to store the latent state. Thus, to balance the learning and forgetting factors, we proposed two gates to fuse the two features adaptively, inspired by the idea of the threshold mechanism, such as the forget gate from LSTM [22], update gate from GRU [38], and add and erase gates from GKT [24].

In our CF-DKD updating method, the forget gate $F_t$ controls what information to erase from the value matrix $M_{stu}^{t-1}$ after a new exercise comes with embedded current response $v_t$ and long-term forgetting factor $f_t$. A temporary forgetting vector $\tilde{f}_t$ is generated by combining the students’ response $v_t (v_t \in \mathbb{R}^{[d_i]})$ and forgetting features $f_t (f_t \in \mathbb{R}^{[d_f]})$. The forgetting information $F_t$ can be computed via a full-connection layer with the sigmoid activation as

$$\tilde{f}_t = \varphi (f_t, v_t),$$
$$F_t = \text{sigmoid} \left( F^T \cdot \tilde{f}_t + b_f \right),$$

where $f_t = \varphi (\triangle s_t, \triangle r_t)$ and $\varphi (\cdot)$ are the integration functions explained in section “E. PREDICTING.” $F^T$ is a weight matrix $F^T \in \mathbb{R}^{(d_f + d_i) \times (d_f + d_i)}$, and each entry is a scalar from 0 to 1.

Similar to the forget gate, the learn gate $L_t$ controls what information to enhance in the current knowledge state $M_{stu}^{t-1}$ by the current response $v_t$ and long-term memory of learning factor $l_t$. Therefore, we integrate the two factors into a temporary learning vector $\tilde{l}_t$, using the same integration method. We can obtain the learning information $L_t$ through a full connection layer with the tanh activation. The $L_t$ vector is calculated as

$$\tilde{l}_t = \varphi (\triangle c_t, v_t),$$
$$L_t = \text{tanh} \left( L^T \cdot \tilde{l}_t + b_l \right),$$

where $L^T$ is the weight matrix, and $L^T \in \mathbb{R}^{(d_i + d_f) \times (d_i + d_f)}$.

However, for the gate mechanism, not all information has the same influence on reducing or strengthening the previous information. Thus, we multiply the $M_{stu}^{t-1} (i)$ with the correlation weight $w_t$ which addresses the exercise with the related exercise in the reading process. Hence, the value memory $M_{stu}^{t-1} (i)$ is updated by:

$$M_{stu}^{t} (i) = M_{stu}^{t-1} (i) \left[ 1 - w_t (i) F_t \right] + w_t (i) L_t,$$

where $1$ is a row vector filled with ones. The former part in Equation (10) represents the forgetting operation, which weakens the previous memory; however, the latter part represents the learning process, which strengthens the memory with new learning information.

F. PREDICTING

The students’ performance on a new exercise is not only determined by the current knowledge proficiency but also the same cognitive factors (i.e., learning or forgetting features). Thus, how can we integrate various factors to predict the students’ performance on the next exercise?

1) INTEGRATION OF VARIOUS FEATURES

Given the various features of a new exercise $(k_i, f_i, l_i)$, the first important task is to integrate them into a unified tensor. Thus, we explored three methods to integrate these features [10], [37], such as concatenation, multiplication, and concatenation and multiplication. First, the most popular integration method is concatenation which stacks all the feature vectors without any change in the original vector. Second, multiplication modifies the original vector by multiplying the contextual information. Thus, we conduct this method using matrix multiplication $\odot$. Third, concatenation and multiplication combine the former two methods, further enhancing cognition-related information.

In the predicting process, the integration vector $\varphi_{in}$ is determined by $k_i$, $f_i$, and $l_i$, which is calculated by three methods as shown in Table 1.

As shown in Fig. 3(d), we take an integration example of the four features by the concatenation method to predict students’ response.

2) STUDENTS’ PERFORMANCE PREDICTION

After combining all the features by a unified representation vector $\varphi_{in}$, we concatenate it with the read content vector $read_i$. A two-layer feedforward neural network is used to obtain the possibility of answering the exercise correctly, as shown in the following equations:

$$\text{forward}_i = \text{tanh} \left( W^T_{1} \cdot [\text{read}_i, \varphi_{in}] + b_1 \right),$$
$$p_i = \text{sigmoid} \left( W^T_{2} \cdot \text{forward}_i + b_2 \right),$$

where the first layer employs the tanh action function, $\text{tanh} (z_i) = (e^{z_i} - e^{-z_i}) / (e^{z_i} + e^{-z_i})$. The second layer
TABLE 2. Statistics of the four datasets.

| Dataset          | Exercises | Students | Interactions | Interactions per Student |
|------------------|-----------|----------|--------------|--------------------------|
| ASSISTments2015  | 100       | 19,917   | 708,631      | 35.6                     |
| ASSISTments2017  | 3,162     | 1,709    | 942,816      | 551.6                    |
| Slepeamap2015    | 1,683     | 18,198   | 1,336,210    | 73.4                     |
| EAnalyst         | 2,763     | 1,763    | 525,638      | 298.1                    |

employs a sigmoid action function to obtain the final prediction result, which is a scalar representing the probability of correctly answering exercise $e_t$. Moreover, sigmoid ($z_i$) = $1 / (1 + e^{−z_i})$.

G. MODEL OPTIMIZATION

Our CF-DKD model is an end-to-end model, which requires a total loss function to adjust the parameters through backward transmission, such as the exercise embedding matrix $A$, student embedding matrix $B$, learning transformation matrix $L^T$, and forgetting transformation matrix $F^T$. Therefore, we optimize the CF-DKD model by a cross-entropy loss function between the real response $r_t$ recorded in the interaction sequence and the predicted probability $p_t$. The loss function is defined as

$$L = - \sum [r_t \log p_t + (1 − r_t) \log (1 − p_t)]. \quad (12)$$

FIGURE 6. Number distribution of dataset.

The initialization of $M_{stu}$ and $M_{exe}$ employs a random Gaussian distribution, $M_{stu} \sim N(0, \sigma)$, $M_{exe} \sim N(0, \sigma)$. We employ a stochastic gradient descent to accelerate the convergence and weight decay (L2 regularization) to avoid overfitting. The learning rate is dynamically updated by the exponential decay method with a decreasing parameter of 0.95. Thus, we can obtain the best effect when the initial value is 0.09.

IV. EXPERIMENT AND RESULT

Further, to verify the effectiveness of our proposed CF-DKD and its implementations, we conducted comparative experiments on four datasets from the following aspects: (1) the accuracy of predicting performance on a new exercise, (2) the parameter sensitivity test for the best optimal size of the memory model, (3) the impact of different feature integration methods, and (4) the effectiveness of the cognitive factor in the CF-DKD models.

A. DATASETS

We employed four datasets to evaluate our proposed model from the diagnostic literature [20], [39], [10], [40] since the datasets contain temporal features. Table 2 presents detailed statistics of the four datasets, and Fig. 6 illustrates the data analyses.

1) ASSISTMENTS2015

This dataset was collected from ASSISTments system in 2015, which has been widely used in students’ performance prediction tasks. We removed the records with fewer than two interactions in the original dataset before the experiment. After data preprocessing, the dataset includes 708,631 interactions of 19,197 students and 100 exercises. This dataset has the lowest average number of exercises per student because it has the highest number of students relative to the other datasets.

2) ASSISTMENTS2017

This dataset was gathered from the same system as ASSISTments2015. We also preprocessed the data using the same method as above. The pruned dataset includes 942,816 interactions for 1,709 students with 3,162 exercises, and has the largest number of exercises and largest average number of exercises per student.

3) SLEPEAMAP2015

This dataset was derived from an online test system [41] for geography in 2015. After removing the nonconforming interactions, the dataset included 18,198 students with 1,336,210 interactions on 1,683 exercises, and each student answered 73.4 exercises on average.
The accuracy of knowledge proficiency is difficult to evaluate via observations. Therefore, we usually evaluate the accuracy of the diagnostic model through students’ performance prediction. Area Under ROC Curve (AUC) is the main evaluation metric in knowledge tracing models. Generally, the AUC score ranges from 0 to 1. The larger the scores, the better the result. The value of 0.5 represents a random prediction, similar to coin-flipping. All the datasets were divided into 70% and 30% based on the students, where the former was used for training and validation, and the latter for testing. Thus, to avoid the randomness of the evaluation results, we implemented 5-fold cross-validation for the training and validation datasets (e.g. further divided by the ratio of 8:2 for training and validating, respectively) to tune the hyper-parameters. Moreover, we selected the average score as the final result for comparison.

We implemented most models, such as DKT, DKVMN, DKT+F, and our CF-DKD model, using Pytorch 1.4. Considering that interaction sequences are of different lengths, we set each sequence to be a length of 200 [22], [19], [20], which is a fixed size, by padding short sequences with null symbols to improve computing efficiency.

We implemented all the models via deep learning with CUDA 10.2 on GPU NVIDIA GeForce RTX 2080 Ti. For the traditional BKT model, we obtained the result from reference [20] on the ASSISTments2015 dataset. For the IRT model, we selected the qualified records in the dataset to implement the models because the model required the same number of exercises by students, where we removed many sparse data. However, we did not run the IRT model in the ASSISTments2015 and ASSISTments2017 datasets as they have either too many students or questions, which does not meet the requirements of IRT.

D. RESULTS

1) PREDICTION ACCURACY ASSESSMENT

We compared our CF-DKD model with the five other baselines. Table 3 presents the AUC results for the performance-predicting task on the four datasets. Thus, there are four observations. First, our proposed CF-DKD model outperforms the other baselines over all four datasets. Particularly, our CF-DKD model achieved an average AUC value that is 1%–2% higher than the best performing model on all four datasets. The results indicate that the CF-DKD model can make full use of both the cognitive features and the memory network, which enhances the prediction performance. Second, among the models based on deep learning,

### TABLE 3. AUC comparison table of different dynamic knowledge diagnosis models on four datasets.

| AUC        | IRT  | BKT  | DKT  | DKT+F | DKVMN | CF-DKD | Improvement |
|------------|------|------|------|-------|-------|--------|-------------|
| ASSISTments2015 | 0.642 | 0.6891 | 0.7083 | 0.6792 | 0.7096 | +0.13% |
| ASSISTments2017 | -    | -    | 0.6622 | 0.7227 | 0.7189 | 0.7390 | +1.60% |
| Stepemapy2015 | 0.714 | -    | 0.6078 | 0.7086 | 0.7433 | 0.7580 | +1.50% |
| EAnalyst    | 0.8503 | 0.69 | 0.7997 | 0.8626 | 0.8978 | 0.9144 | +1.66% |

4) EANALYSTX4

This dataset was derived from an offline-to-online test system [40] widely applied in China. We selected response interaction records on mathematics for the experiment. Unlike the previous three datasets, the main component of EAnalyst data is collected from the pre-class quiz, post-class quiz, homework, unit-test, and term-test. Students check the answer immediately after every exercise in the adaptive test. Meanwhile, they are required to answer all exercises in the test offline or online before checking the answers in the EAnalyst test. The application of EAnalyst accords with the reality of current Chinese education. Meanwhile, the distribution of student interactions changes sequentially because the interactions of students in a group are the same. After removing records from only one test, the dataset includes 525,638 interactions from 1,763 students on 2,763 exercises, and each student answered 298.1 exercises on average.

B. BASELINE

Moreover, to evaluate the effectiveness of our proposed model, we compared our model with several diagnose models as the baselines.

IRT [16] is a popular cognitive diagnostic model that models students’ latent traits using a logistic-like function. BKT [21] uses a set of binary variables to represent students’ knowledge states and traces the change by the hidden Markov model. DKT [22] is the first model to introduce deep learning technologies to model the learning process. It takes the knowledge ID as an exercise and traces knowledge proficiency by a hidden layer of RNN to predict a future response. We follow the original hyper-parameters from this study. DKVMN [19] uses a key-value memory network to model students’ learning process, extending the MANN model. The key matrix stores the latent knowledge, while the value matrix stores the knowledge state of the student, whose mechanism makes it possible to track the state of students on multiple knowledge states dynamically. DKT+F [10] is an extension of the DKT model by considering forgetting behavior to predict performance. Our proposed model is CF-DKD in this study.

C. MEASURES AND EVALUATION SETTINGS

The accuracy of knowledge proficiency is difficult to evaluate since the true knowledge state of students cannot be obtained via observations. Therefore, we usually evaluate the accuracy of the diagnostic model through students’ performance prediction. Area Under ROC Curve (AUC) is the main evaluation metric in knowledge tracing models. Generally, the AUC score ranges from 0 to 1. The larger the scores, the better the result. The value of 0.5 represents a random prediction, similar to coin-flipping. All the datasets were divided into 70% and 30% based on the students, where the former was used for training and validation, and the latter for testing. Thus, to avoid the randomness of the evaluation results, we implemented 5-fold cross-validation for the training and validation datasets (e.g. further divided by the ratio of 8:2 for training and validating, respectively) to tune the hyper-parameters. Moreover, we selected the average score as the final result for comparison.

We implemented most models, such as DKT, DKVMN, DKT+F, and our CF-DKD model, using Pytorch 1.4. Considering that interaction sequences are of different lengths, we set each sequence to be a length of 200 [22], [19], [20], which is a fixed size, by padding short sequences with null symbols to improve computing efficiency.

We implemented all the models via deep learning with CUDA 10.2 on GPU NVIDIA GeForce RTX 2080 Ti. For the traditional BKT model, we obtained the result from reference [20] on the ASSISTments2015 dataset. For the IRT model, we selected the qualified records in the dataset to implement the models because the model required the same number of exercises by students, where we removed many sparse data. However, we did not run the IRT model in the ASSISTments2015 and ASSISTments2017 datasets as they have either too many students or questions, which does not meet the requirements of IRT.

D. RESULTS

1) PREDICTION ACCURACY ASSESSMENT

We compared our CF-DKD model with the five other baselines. Table 3 presents the AUC results for the performance-predicting task on the four datasets. Thus, there are four observations. First, our proposed CF-DKD model outperforms the other baselines over all four datasets. Particularly, our CF-DKD model achieved an average AUC value that is 1%–2% higher than the best performing model on all four datasets. The results indicate that the CF-DKD model can make full use of both the cognitive features and the memory network, which enhances the prediction performance. Second, among the models based on deep learning,
we found that DKT+F and CF-DKD models, considering the cognitive features, generate better performance results than the original DKT and DKVMN models, indicating the effectiveness of introducing cognitive features to the students’ performance prediction task by introducing more time-related factors, which is consistent with the observations from DKT+F [10]. Third, we note that the AUC value of ASSISTments2015 is the lowest in all datasets, regardless of any model, because the average number of exercises per student on this dataset is the lowest, which increases the difficulty of the tracing task. Fourth, the AUC value for traditional models (IRT and BKT) is lower than that of the new models with the deep learning method in most cases. IRT assumes that all students interact with the same exercises in a single assessment, which ignores the dynamic time sequence information. The IRT result is not the lowest (as we have discarded many records), which is consistent with the result in [28] regarding the other datasets. Similarly, BKT does not perform better than other deep learning models because the RNN models can likely capture temporal information more effectively than the traditional Bayesian method. Finally, the state-of-the-art models (DKVMN and CF-DKD) based on an extra memory network strengthen the performance of the models, where the key-value matrix captures more details with latent knowledge. Therefore, our CF-DKD, with a key-value memory network and cognitive factors, is more suitable for student performance prediction modeling.

Moreover, Fig. 7 shows the trend in loss changes of each model on all four datasets on both the training and validation sets. First, the training loss value of DKT+F can
TABLE 5. Different integration methods affect the performance of the model.

| Integration methods   | $D_{KT_{mul+cat}}$ | $CF - D_{KD_{cat}}$ | $CF - D_{KD_{mul}}$ | $CF - D_{KD_{mul+cat}}$ |
|-----------------------|--------------------|--------------------|--------------------|-------------------------|
| ASSISTments2015       | 0.7083             | 0.7096             | 0.6804             | 0.6794                  |
| ASSISTments2017       | 0.7227             | 0.7390             | 0.6952             | 0.6965                  |
| Slepeemapy2015        | 0.7086             | 0.758              | 0.7448             | 0.7411                  |
| EAnalyst              | 0.8626             | 0.9144             | 0.8922             | 0.8953                  |
| Average               | 0.7506             | 0.7803             | 0.7532             | 0.7531                  |

TABLE 6. AUC comparison table of diagnosis models with different characteristics.

| Model                  | DKT        | DKT+       | DKT+fl       | DKT+fl+lt     | DKVMN      | DKT $\text{fl}$   | DKT $\text{lt}$ | CF-DKD         |
|------------------------|------------|------------|--------------|--------------|------------|-------------------|----------------|----------------|
| ASSISTments2015        | 0.6891     | 0.6809     | 0.6867       | 0.7083       | 0.6792     | 0.7046            | 0.7052         | 0.7096         |
| ASSISTments2017        | 0.6622     | 0.6766     | 0.6775       | 0.7227       | 0.7189     | 0.7322            | 0.7279         | 0.739          |
| Slepeemapy2015         | 0.6078     | 0.6383     | 0.6435       | 0.7086       | 0.7433     | 0.7407            | 0.7553         | 0.758          |
| EAnalyst               | 0.7997     | 0.8136     | 0.8136       | 0.8626       | 0.8978     | $\text{0.9171}$   | 0.9144         | 0.9144         |
| Average                | 0.6897     | 0.7024     | 0.7053       | 0.7506       | 0.7598     | 0.7737            | 0.7757         | 0.7803         |

be comparable with CF-DKD, but it is quite far from the validation set, especially on EAnalyst as shown in Fig. 7(a). Thus, the DKT+F (the distance of loss value between training and validation is 0.11) suffers severe overfitting, while our CF-DKD model (the distance is 0.01) does not encounter such problems. Second, the fluctuation of DKT+F is quite violent, which is especially obvious on the Slepeemapy2015 dataset shown in Fig. 7(d), while CF-DKD has a flatter curve. Finally, although DKVMN performs better on the training set for the last two datasets, after 100 epochs, the loss value of the DKVMN increases gradually on ASSISTments2017 and slepeemapy2015, as shown in Fig. 7(c) and (d). In general, the loss value of CF-DKD decreases smoothly and reaches a stable convergence value at 200 steps, which avoids the overfitting problem effectively. Thus, CF-DKD performs best on the testing dataset. Therefore, the generalization ability of CF-DKD is relatively strong.

2) PARAMETER SENSITIVITY TEST
This section explores several different combinations of four hyper-parameters, including the initialization of the learning rate ($\gamma$), the number of hidden layer units ($h$), batch size ($b$), and the number of latent memory slots ($s$). Thus, we conduct sensitive experiments on different hyper-parameters, $\gamma$, $b$, $h$, and $s$, respectively, to explore their effectiveness. In each experiment, we adjusted the main parameter and fixed the remaining parameters to explore the performance.

First, we focus on $\gamma$ initialization. We fix other parameters, $s = 50, h = 128$, and $b = 32$, according to the setting from previous studies [19], [20]. Specifically, we attempt the initialization in $\gamma \in \{0.1, 0.01, 0.001\}$, and the best result appears at nearly 0.1. Next, we further attempt $\gamma \in [0.05 - 0.09]$ and find the best performance to be $\gamma = 0.09$ in most cases.

We then focus on $h$, and try several common hidden unit numbers: $h = 32, h = 64, h = 128$, and $h = 256$. Our CF-DKD model performs best when $h = 128$.

Next, we focus on $b$ and $s$, and perform a fair comparison of CF-DKD and DKVMN by fixing $\gamma$ and $h$. Table 4 shows the AUC result of different combinations of the two parameters, where $b \in \{16, 32, 64\}$ and $s \in \{5, 10, 50, 100\}$.

Table 4 shows all AUC values on all four datasets. We find that CF-DKD performs better than DKVMN. In particular, we focus on our EAnalyst dataset, where our CF-DKD can achieve better performance, $AUC_{CF-DKD} = 91.44\%$ when $s = 50$, as the number of latent knowledge features on this dataset is greater than that of other datasets as shown in Table 2. In comparison, the best AUC of the DKVMN is 89.78% when $s = 10$ and $b = 32$. Similarly, for Slepeemapy2015, CF-DKD achieved the best performance: $AUC_{CF-DKD} = 75.71\%$ when $c = 10, b = 64$. Meanwhile, the DKVMN can achieve $AUC_{DKVMN} = 73.9\%$ when $s = 10, b = 64$. We note that hyper-parameters can cause huge fluctuations in the AUC value, ranging from 0% to 4%; therefore, the adjustment of hyper-parameters is particularly important.

3) THE IMPACT OF DIFFERENT DATA INTEGRATION METHODS
Further, to explore the performance of different integration methods, we compared CF-DKD with three different methods with DKT+F, which use the “concatenation and
multiplication \((\text{mul} + \text{cat})\)" method to achieve the best performance, as demonstrated by [10] and [37].

As shown in Table 5, our CF-DKD with concatenation method achieved the best AUC on the four datasets, with an average AUC value of 0.7803. In fact, our CF-DKD with any method can perform better than DKT+F on average AUC. Meanwhile, we also find that the AUC value of our model with the combination of concatenation and multiplication is not the best, which may be related to the overly complex neural network. Notably, DKT only uses a layer of RNN to represent its knowledge changes, while CF-DKD is more complex because it uses an additional memory network to store the latent knowledge state. From this perspective, the relatively simple integration method achieves the optimal AUC result, where the AUC order is \(\text{CF}_{\text{cat}} > \text{CF}_{\text{mul}} > \text{CF}_{\text{mul}+\text{cat}} > \text{DKT} + \text{F}\).

4) THE EFFECTIVENESS OF COGNITIVE FEATURES
We also investigated the effectiveness of each cognitive feature in our CF-DKD model. In particular, for the performance-predicting task, we first computed the AUC score of the DKT with different cognitive features, including DKT+ft (adding forgetting factors), DKT+lt (adding learning factors), DKT+ft+lt (adding forgetting factors and learning factors). We then considered different variants of CF-DKD, such as DKD_ft, DKD_lt, and CF-DKD, and compared the improved model integrating cognitive features with the original models (DKT and DKVMN). Finally, Table 6 reveals the different results of the models on all datasets.

From the table, we find that the AUC value with more extra features performs better than the original models on the four datasets. In particular, the addition of any feature (forgetting or learning) results in a higher AUC: among CF-DKD-like models, the average AUC value of the models with one feature (DKD+ft or DKD+lt) increases 1.4%-1.6% compared with the original DKVMN. Thus, we can obtain the same trend in DKT variants, which has been verified by [10]. Moreover, the models with both learning and forgetting features have a better AUC value than models with only one feature. For instance, among DKT variants, DKT with both of the two features performs better than DKD+ft and DKD+lt, showing a 4.5%-4.8% improvement in the AUC result. This finding illustrates that the greater the number of cognitive features, the better the performance of this model when predicting the response on a new exercise.

Finally, our CF-DKD model with a memory network performs better than DKT+F; that is, \(\text{AUC}_{\text{CF}} > \text{AUC}_{\text{DKT+FT}}\) as the memory network can learn more latent information in the sequence, enhancing the prediction performance. In summary, these results support the idea that the introduction of cognitive features and the extra memory network are important for modeling the learning process of students.

E. THE RESULT DISCUSSION AND EDUCATION APPLICATION
The application of CF-DKD is essential for personalized education. It is possible to not only discover students at risk by using the predicted response of a single exercise but also to find similar exercises for recommendation by clustering latent knowledge. We can also obtain the weaknesses in students’ knowledge to conduct personalized learning simultaneously.

FIGURE 8. Visualization of the exercise results by clustering.

FIGURE 9. The corresponding knowledge of the exercise.

1) PREDICTION OF STUDENTS’ PERFORMANCE ON A TEST
As there is an urgent need for personalized education, students’ response prediction can help teachers discover at-risk students [43] and then provide these students with better early warning services. As we have predicted the response of every exercise thus far, the comprehensive performance of future tests requires the total score of all the exercises. Relying on the CF-DKD model being previously trained, we feed the new exercises and corresponding cognitive factors into the prediction process by a loop mechanism to output all the scores of each exercise. Then, we summarize the result to obtain the sum score. That is, \(Y_j = \sum_{i}^{N} y_{ij} \cdot s_{ij}\) where \(y_{ij}\) represents the
predicted response of student $j$ on exercise $i$, and $s_i$ is the full score of the exercise. The prediction of students’ total scores could be used to discover students at risk to help them achieve better performance in future tests.

2) THE SIGNIFICANCE OF SIMILAR EXERCISES
IN THE APPLICATION
Finding similar exercises is a meaningful task in education. For example, we can recommend similar exercises to students for remedial learning [6], teachers can retrieve these exercises for consolidating knowledge, and we can conduct detailed cognitive analysis with the help of these exercises [9]. In fact, whether a student answers an exercise correctly is determined not only by the corresponding knowledge but also by latent factors, such as the difficulty of the exercise. Therefore, we selected ASSISTments2017 to find similar exercises, clustering 100 randomly chosen exercises into different sets. We first extracted the $k_i$ vector of the exercise representation from the fine-tuned end-to-end neural network and used the “mean shift” method to cluster the exercises into 10 clusters. Finally, we visualized the clustering results, as in Fig. 8. Similar exercises in one cluster are labeled with the same color.

Each exercise is provided with a description, as depicted in Fig. 9, which is useful for validating how well our model discovers correlations between exercises and their latent factors. As shown in Fig. 8 and 9, the exercises in the same cluster are similar to each other based on certain knowledge. For instance, exercises such as 43, 77, 88, and 96 are marked light blue in the same cluster with the corresponding knowledge as “fraction-division,” “fraction-decimals-percent,” “reduce-fraction,” and “adding-decimals,” respectively, which are related to the operation of fractions. Similarly, other clusters can be observed, such as exercises 5, 9, and 90 with “area,” “application: isosceles triangle,” and “area-of-circle,” respectively, which are related to geometry. Therefore, the results indicate the effectiveness of CF-DKD in discovering similar exercises related to latent knowledge.

3) THE INFLUENCE OF COGNITION ON THE EVOLUTION OF LATENT KNOWLEDGE STATE
Another application is to analyze the knowledge structure [9] and discover weak knowledge in order to develop personalized learning. Students’ knowledge state is the degree of mastery of each type of knowledge in a value range from 0 to 1. The knowledge state changes over time, especially over a long interval, where the mastery declines. For repeated knowledge, mastery improves. Thus, for an in-depth analysis, we selected one sequence of a student to track the evolution trend over 30 timesteps. Five slots representing the latent features were selected for visualization. Fig. 10 shows an example of depicting a student’s five changing latent knowledge states while they interacted with 30 exercises.

In Fig. 10 (a), the first column represents the initial state of each type of latent knowledge of this student, generated randomly. The knowledge state changes because the state of students transitions gradually over time, rather than alternates between master or non-master. Specifically, each time a student answers an exercise right (wrong), latent-knowledge proficiency increases (decreases). For example, the student masters latent-knowledge 5 after answering the exercise correctly at the second and third epoch. However, the student masters latent-knowledge 5 but cannot understand latent-knowledge 4 since they responded well to the former knowledge but not the latter one. However, an inconsistent phenomenon exists in that the students’ knowledge state of latent-knowledge 2 is slightly lower at the tenth interaction step even when they get a correct answer on the exercise. The possible reason is that the model needs more interaction records for the knowledge proficiency diagnosis, which becomes more certain as the interactions increase in subsequent steps. Moreover, when a student answers right (or wrong) on an exercise, more than one knowledge state is affected. For example, the state of latent-knowledge 3 increases when a student answers latent-knowledge 5-related exercises correctly because the exercises correlate with the two types of latent knowledge. Thus, after answering 30 exercises, the student mastered latent-knowledge 3 and 5 but failed to master latent-knowledge 1, 2, and 4, as shown in Fig. 10(b).

V. CONCLUSION
This study proposed a CF-DKD approach incorporating students’ cognition and extra memory matrix to predict students’ performance on future exercises and diagnose their knowledge mastery state simultaneously. Although the RNN can...
model effectively, it cannot track students’ mastery of the latent knowledge on different exercises with severe information loss caused by the same time interval between the interactions. Thus, we extended a memory network to a cognition-aware model of CF-DKD by further combining the cognitive information in the interaction sequence. Three integration methods were utilized to incorporate the features. Our CF-DKD model tracked the temporal information related to students’ cognition for the prediction task, which is comparatively superior to other state-of-the-art baselines. We then conducted extensive experiments on four datasets regarding parameter sensitivity, the impact of different data integration methods, and the cognitive factors’ influence. Finally, we applied the CF-DKD to educational applications for personalization. The results demonstrate the effectiveness of CF-DKD.

In future work, we will consider the relationship between knowledge concepts as well as the implementation of multiple knowledge concepts to capture knowledge structure in-depth. In addition, we will consider an education theory-based neural network to diagnose knowledge proficiency based on cognitive process theories in educational psychology.

REFERENCES

1. A. T. Corbett and J. R. Anderson, “Knowledge tracing: Modeling the acquisition of procedural knowledge,” User Model. User-Adapted Interact., vol. 4, no. 4, pp. 253–278, 1995.
2. S. D. Craig, “The impact of a technology-based mathematics after-school program using ALEKS on student’s knowledge and behaviors,” Comput. Educ., vol. 68, pp. 495–504, Oct. 2013.
3. R. Grossman and E. Salas, “The transfer of training: What really matters?” Int. J. Training Develop., vol. 15, no. 2, pp. 103–120, Jun. 2011.
4. F. Wang, Q. Liu, E. Chen, Z. Huang, Y. Chen, Y. Yin, Z. Huang, and S. Wang, “Neural cognitive diagnosis for intelligent education systems,” in Proc. AAAI Conf. Artif. Intell., 2020, vol. 34, no. 4, pp. 6153–6161.
5. Z. Shou, X. Lu, Z. Wu, H. Yuan, H. Zhang, and J. Lai, “On learning path planning algorithm based on collaborative analysis of learning behavior,” IEEE Access, vol. 8, pp. 119863–119879, Jul. 2020.
6. S. Schiaffino, P. Garcia, and A. Amandi, “eTeacher: Providing personalized assistance to e-learning students,” Comput. Educ., vol. 51, no. 4, pp. 1744–1754, Dec. 2008.
7. P. M. Moreno-Marcos, T. C. Pong, P. J. Muñoz-Merino, and C. D. Kloos, “Analysis of the factors influencing learners’ performance prediction with learning analytics,” IEEE Access, vol. 8, pp. 5264–5282, Jan. 2020.
8. Y. Liu, Y. Yang, X. Chen, J. Shen, H. Zhang, and Y. Yu, “Improving knowledge tracing via pre-training question embeddings,” 2020, arXiv:2012.05031. [Online]. Available: http://arxiv.org/abs/2012.05031
9. Q. Liu, R. Wu, E. Chen, G. Xu, Y. Su, Z. Chen, and G. Hu, “Fuzzy cognitive diagnosis for modelling examinee performance,” ACM Trans. Intell. Syst. Technol., vol. 9, no. 4, pp. 1–26, Feb. 2018.
10. K. Nagatani, Q. Zhang, M. Sato, Y.-Y. Chen, F. Chen, and T. Okhuma, “Augmenting knowledge tracing by considering forgetting behavior,” in Proc. World Wide Web Conf., San Francisco, CA, USA, May 2019, pp. 3101–3107.
11. L. Averell and A. Heathcote, “The form of the forgetting curve and the fate of memories,” J. Math. Psychol., vol. 55, no. 1, pp. 25–35, Feb. 2011.
12. H. Ebbinghaus, “Memory: A contribution to experimental psychology,” Ann. Neurosci., vol. 20, no. 4, pp. 155–156, Oct. 2013.
13. R. Pelánek, “Modeling students’ memory for application in adaptive educational systems,” in Proc. EDM, UNED, Madrid, Spain, 2015, pp. 480–483.
14. B. Settles and B. Meeder, “A trainable spaced repetition model for language learning,” in Proc. 54th Annu. Meeting Assoc. Comput. Linguistics, San Diego, CA, USA, 2016, pp. 1848–1858.
[40] T. Huang, Z. Li, H. Zhang, H. Yang, and H. Xie, “EAnalyst: Toward understanding large-scale educational data,” in Proc. EDM, Jul. 2020, pp. 620–623.

[41] J. Papousek, R. Pelánek, and V. Stanislav, “Adaptive geography practice data set,” J. Learn. Anal., vol. 3, no. 2, pp. 317–321, Sep. 2016.

[42] D. Liu, Y. Zhang, J. Zhang, Q. Li, C. Zhang, and Y. Yin, “Multiple features fusion attention mechanism enhanced deep knowledge tracing for student performance prediction,” IEEE Access, vol. 8, pp. 194894–194903, Nov. 2020.

[43] D. Baneres, M. E. Rodriguez, and M. Serra, “An early feedback prediction system for learners at-risk within a first-year higher education course,” IEEE Trans. Learn. Technol., vol. 12, no. 2, pp. 249–263, Apr. 2019.

TAO HUANG received the M.S. and Ph.D. degrees from Central China Normal University, Wuhan, China, in 2004 and 2007, respectively.

Since 2015, he has been a Professor with the National Engineering Laboratory for Educational Big Data. He has authored four books and over 30 articles. His research interests include educational data mining, cognitive diagnosis, and artificial intelligence in education.

HUALI YANG was born in Zhijiang, Hubei, China, in 1988. She received the M.S. degree from the National Engineering Research Center for E-Learning, Central China Normal University, in 2014.

Since 2013, she has been an Assistant Researcher in computer application technology with Central China Normal University. Her research interests include natural language processing, cognitive diagnosis, and intelligent tutor systems.

ZHI LI was born in Jingmen, Hubei, China, in 1994. She received the bachelor’s degree in logistics management from Shanghai University of Finance and Economics, in 2016. She is currently pursuing the master’s degree in computer science with Central China Normal University.

Her research interests include educational data mining, deep learning, and artificial intelligence.

HEKUN XIE was born in Yulin, Guangxi, China, in 1996. He received the bachelor’s degree in computer science and technology from Guangxi Normal University, in 2019. He is currently pursuing the master’s degree in computer science and technology with Central China Normal University.

His research interests include educational data mining, deep learning, and intelligent education systems.

JING GENG was born in Yuncheng, Shanxi, China, in 1990. She received the master’s degree from the College of Science, Northwestern Polytechnical University, in 2016. She is currently pursuing the Ph.D. degree in education technology with Central China Normal University.

Her research interests include education assessment, cognitive diagnosis, and educational data mining.

HAO ZHANG is currently an Associate Professor with Central China Normal University. His main research interests include multimedia resources recommendation, the privacy of big data, security of cloud computing based on virtualization, and learning behavior analysis based on machine learning and deep learning.

Hao Zhang
Associate Professor
Central China Normal University

* * *