Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 9153697, 16 pages
https://doi.org/10.1155/2022/9153697

Research Article

NTM-Based Skill-Aware Knowledge Tracing for Conjunctive Skills

Qiang Huang,[1] **Wei Su**,[1] **Yuantao Sun**,[2] **Tianyuan Huang**,[1] **and Juntai Shi**[1]

[1]*School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China*
[2]*Ant Group, Hangzhou, China*

Correspondence should be addressed to Wei Su; suwei@lzu.edu.cn

Received 17 March 2022; Revised 10 May 2022; Accepted 12 June 2022; Published 27 July 2022

Academic Editor: Huihua Chen

Knowledge tracing (KT) is the task of modelling students' knowledge state based on their historical interactions on intelligent tutoring systems. Existing KT models ignore the relevance among the multiple knowledge concepts of a question and characteristics of online tutoring systems. This paper proposes a neural Turing machine-based skill-aware knowledge tracing (NSKT) for conjunctive skills, which can capture the relevance among the knowledge concepts of a question to model students' knowledge state more accurately and to discover more latent relevance among knowledge concepts effectively. We analyze the characteristics of the three real-world KT datasets in depth. Experiments on real-world datasets show that NSKT outperforms the state-of-the-art deep KT models on the AUC of prediction. This paper explores details of the prediction process of NSKT in modelling students' knowledge state, as well as the relevance of knowledge concepts and conditional influences between exercises.

1. Introduction

With the development of intelligent tutoring systems (ITSs) and the emergence of massive open online courses (MOOCs) [1, 2], knowledge tracing plays an important role in improving the efficiency of personalized learning platforms. Knowledge tracing is the task of modelling students' knowledge state based on their historical interactions to predict students' mastery of knowledge concepts (KCs), where a KC can be an exercise, a skill, or a concept [3, 4].

In order to better model students' knowledge state, various knowledge-tracing models have been proposed. In previous studies, Bayesian knowledge tracing (BKT) is a powerful knowledge-tracing model. BKT models students' knowledge concept state by using a hidden Markov model (HMM) for each KC [5].

As deep learning develops, a lot of deep learning models have been applied in KT. Chris Piech applies the recurrent neural network (RNN) to model the student learning process for the first time and proposes deep knowledge tracing (DKT) [6–9]. The dynamic key-value memory network (DKVMN) uses a static memory called key and a dynamic memory called value to discover latent relations between exercises and knowledge concepts [10, 11]. Self-attentive knowledge tracing (SAKT) proposes a self-attention-based KT model to model the students' knowledge state, with exercises as attention queries and students' past interactions as attention keys/values [3, 12–15].

However, the aforementioned works only focus on students' exercise interactions and ignore the relations between questions and skills. It cannot model students' knowledge state accurately by merely focusing on students' interactions. Knowledge tracing models begin to pay attention to the structure of the knowledge concepts [16–18].

Deep hierarchical knowledge tracing models students' knowledge state by capturing the hierarchical structure of questions and knowledge concepts [16]. Neil Heffernan's latest work considers the question information to which the knowledge concept belongs [17]. Graph-based knowledge tracing considers the influence among neighboring knowledge concepts [19–22]. The bipartite graph is an effective structural model to capture latent relations between questions and skills [18]. This method is effective, but the amount of calculation is huge because it needs to extract questions and skills, respectively. Thus, it is difficult to be regarded as a streamlined and effective knowledge-tracing model.

None of the above KT models make full use of the multiknowledge concept information of the questions.

Existing knowledge tracing models cannot capture latent relations between questions and concepts concisely and effectively. We know that questions are generally composed of multiple knowledge concepts, which are actually closely related. In order to better model the students' learning process, our model is constructed by using neural Turing machines (NTMs), which are an instance of memory-augmented neural networks (MANNs) that have a large external memory capacity [23–25]. Therefore, on the basis of above deep knowledge tracing models, we propose an NTM-based skill-aware knowledge-tracing model. The highlight of our work is to utilize the knowledge concept composition information of questions to model the students' knowledge state more accurately and to discover more latent relevance among knowledge concepts effectively. The contributions of this paper are concluded as follows:

(i) We process the real-world KT datasets in detail and discover new characteristics of online tutoring systems and knowledge tracing datasets.

(ii) We design a question-skill dictionary algorithm to obtain the conjunctive skills of questions. The input encoding contains both students' answering interaction information and the related knowledge concept information.

(iii) We apply neural Turing machines into knowledge tracing innovatively to enhance the memory capacity of our model and to predict students' mastery of knowledge concepts accurately and discover knowledge concept substructure effectively.

(iv) We propose a novel NTM-based skill-aware knowledge-tracing model for conjunctive skills and apply a novel loss optimization function to deep knowledge tracing to enhance the model's ability of skill awareness. Our model considers the conjunctive knowledge concept information contained in a question in the process of modelling the students' knowledge state; thus, our model outperforms existing KT models.

The rest of this paper is organized as follows: Section 2 presents a brief overview of related work in the field of knowledge tracing. In Section 3, we formulate the process for NSKT to perform the knowledge-tracing task. Then, Section 4 introduces the characteristics and classifications of online tutoring systems. The details of the NSKT model are provided in Section 5. The experimental results and the comparison of models' performance in the real-world datasets are given in Section 6. In Section 7, we discuss in detail the process of NSKT in modelling the students' knowledge state. Section 8 presents the conclusions and future studies of this work.

2. Related Work

In this section, we present a brief overview of the models and methods of related work in the field of knowledge tracing, which can be classified into two main categories, as shown in Table 1.

Table 1: Related work.

Models	Methods	Categories
IRT	Logistic model	Statistical model
BKT	Bayesian model	
DKT	Long short-term memory network	Deep learning models
DHKT	Long short-term memory network	
DKVMN	Memory-augmented neural network	
SAKT	Self-attention	

2.1. Item Response Theory. Item response theory is the most commonly used cognitive model to predict students' mastery of knowledge concepts before knowledge tracing was proposed in 1995 [26, 27]. On the basis of IRT, the students' knowledge state cognitive model based on factor analysis was later proposed: LFA [28] and PFA [29]. These logistic regression models predict students' mastery of knowledge concepts by analyzing the relationship among factors that have an impact on students' answering accuracy [30, 31].

2.2. Knowledge Tracing. Bayesian knowledge tracing (BKT) models the students' knowledge state by using the hidden Markov model (HMM) for a single knowledge concept, which is represented as a set of binary latent variables [5].

With the rise of deep learning, deep knowledge tracing (DKT) was proposed in [6], which regards students' historical interactions as time sequences and models the students' knowledge state by the recurrent neural network (RNN). The experimental results show that DKT has the powerful ability of modelling the students' knowledge state. After DKT, a lot of deep KT models have been proposed to improve the AUC of the prediction of students' mastery of knowledge concepts. However, most of these deep knowledge-tracing models only focus on students' interactions on knowledge concepts and ignore the structural relationship between questions and knowledge concepts.

2.3. Question-KC Relation in Knowledge Tracing. Cen et al. proposed the two IRT models (additive factor model (AFM) and conjunctive factor model (CFM)) to model the conjunctive skills in the student datasets [32]. Both the AFM and CFM consider the conjunctive skills information contained in an item to predict the probability of students answering the item correctly.

Deep hierarchical knowledge tracing begins (DHKT) to focus on the hierarchical relationship between knowledge concepts and questions to predict the performance of students [16]. DHKT trains a question embedding by the average embeddings of the skills belonging to the question. The model using the bipartite graphs can capture relationships between knowledge concepts and questions effectively and systematically to pretrain question embeddings for each question [18]. Neil Heffernan's latest work begins to focus on the architecture of knowledge concepts and questions too [17].

Table 2: Notations.

Notations	Description
p	Problem/question
q	KC (skill/concept)
a	Answer correctness to the knowledge concept (KC) q
c	Answer correctness to related knowledge concept (RKC)
M	Number of unique KCs in the KT dataset
P	Probability
S	The RKCs
H	Interaction sequence of a student: $\{h_1, \ldots, h_{\lvert H\rvert}\}$
D	Dataset
E	Encoding
KC	Knowledge concept
RKC	Related knowledge concept
Dic	Dictionary

3. Problem Formulation

Generally, KT can be formulated as a supervised sequence learning problem: the student's interaction tuple at the timestamp t, $h_t = (q_t, a_t)$ that represents the combination of which skill (exercise) was answered and if the skill was answered correctly, so $a_t \in \{0, 1\}$, $q_t \in \{q_i\}_{i=1}^{M}$, where M is the number of unique exercises in datasets. Given the student's past exercise interactions, $H_t = \{h_0, \ldots, h_t\}$, the goal of KT is to predict the probability that the student will answer question q_{t+1} correctly at the next timestamp $t+1$, $P(a_{t+1} = 1/q_{t+1}, H_t)$ [3, 6, 10].

It can be seen that existing KT models only focus on students' exercise interactions, so they are difficult to predict students' mastery of skills effectively. The notations used in this paper are shown in Table 2.

Definition 1. Related knowledge concepts (RKCs): the related knowledge concepts (RKCs) refer the other knowledge concepts S that compose the question p with a knowledge concept q, where S and q are mutual conjunctive knowledge concepts (skills).

The Algorithm 1 processes the skills and the questions of the dataset to obtain a dictionary Dic with the question number as the key and conjunctive skills of the question as the value, while conjunctive skills are the skills that make up the same question. The time complexity of Algorithm 1 is $\mathcal{O}(n^2)$. In this paper, we use KC shown in Table 2 to represent skill. Let S be the RKCs related to KC q of the answering question p, where $S = \{x/x \in \mathrm{Dic}_p, x \neq q\}$is illustrated in Figure 1(a).

The skill-aware knowledge tracing model can be formulated as follows: the student's interaction at the timestamp t, $h'_t = (p_t, q_t, a_t, S_t, c_t)$, where a_t is the correctness to the question p_t on skill q_t, S_t are the of RKCs of KC q_t, c_t is the correctness to RKCs S_t. Given the student's past interactions, $H'_t = (h'_0, \ldots, h'_t)$, we can predict the probability that the student will answer next KC q_{t+1} correctly at the timestamp $t+1$, $P(q_{t+1}) = P(a_{t+1} = 1/q_{t+1}, H'_t)$ or predict students' mastery of holistic knowledge concepts, $\{P(q_i)\}_{i=1}^{M}$.

4. Online Tutoring Systems

The online tutoring systems can be classified into two categories:

4.1. Question-Level Online Tutoring Systems. In question-level online tutoring systems, students answer the question directly. If the question is answered correctly or incorrectly, all KCs (skills) of the question are answered correctly or incorrectly too. So if a student has answered q_t correctly or incorrectly, then they must answer the RKCs S_t correctly or incorrectly too, which is illustrated in Figure 1(b). Because q_t and S_t are from the same question, so in question-level online tutoring systems, for a student's interaction at the timestamp t: $(p_t, q_t, a_t, S_t, c_t)$,

$$c_t = a_t. \tag{1}$$

4.2. Skill-Level Online Tutoring Systems. The question-answering situation in skill-level online tutoring systems is much more complicated than that of the question-level online tutoring system. Students can individually answer one of the skills in the question and can answer this skill once or multiple times. So if a student answers KC_1 correctly, it does not mean that the student must answer KC_2 correctly, which is shown in Figure 1(c).

Superficially, there is no obvious answering correctness relationship between skill q_t and the related skill set S_t. However, there are a large number of students answering examples shown in Table 3 in skill-level online tutoring systems, indicating that if a student answers q_t incorrectly many times, even if he finally answers q_t correctly, which demonstrates that his mastery of skill q_t is very poor, and similarly, he has poor mastery of S_t. It is very likely that he will answer q_t's-related skills S_t incorrectly. So the student's mastery of q_t, $P(q_t)$ and the student's mastery of S_t, $P(S_t)$ are close:

$$P(q_t) \approx P(S_t). \tag{2}$$

This finding is strongly supported by the actual responses of students in skill-level online tutoring systems. So in skill-level online tutoring systems, according to formula (2), we can assume.

$$c_t \approx a_t, \tag{3}$$

as shown in Table 4.

5. Method

In this section, we will give a detailed introduction of our NSKT framework, of which, the overview architecture is given in Figure 2.

5.1. Model. The model consists of an encoding layer and a neural network layer. In order to better model the students' knowledge state, the model is constructed with

Input: question list L^p; skill list L^q; dictionary Dic; dataset $D = \{(q_i, p_i)\}_{i=0}^{|D|-1}$ composed of skills and questions.
Output: Dic

```
(1) Dicinitialization
(2) L^p initialization
(3) μ←0
(4) for each i←0, 1, ..., |D| − 1 do
(5)    if D(p)_i not inL^p then
(6)       L^p_μ←D(p)_i
(7)       μ←μ + 1
(8)    end
(9) end
(10) for each i←0, 1, ..., μ − 1 do
(11)    η←0
(12)    L^q initialization
(13)    for each j←0, 1, ..., |D| − 1 do
(14)       ifD(p)_j == L^p_i andD(q)_j not inL^q then
(15)          L^q_η←D(q)_j
(16)          η←η + 1
(17)       end
(18)    end
(19)    Dic_{L^p_i}←L^q
(20) end
(21) return Dic
```

Algorithm 1: Question-skill dictionary algorithm.

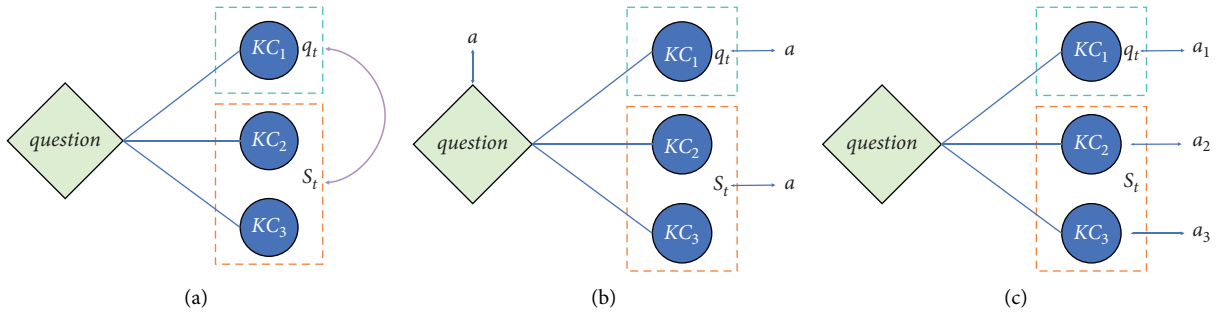

Figure 1: Illustrations. (a) Illustration of RKC S related to KC q, where S and q are mutual conjunctive skills. (b) Illustration of the question-level online tutoring system, where a denotes answer correctness to a question. (c) Illustration of the skill-level online tutoring system, a_1 denotes answer correctness to KC1, a_2 denotes answer correctness to KC_2, and so on a_3.

Table 3: Example of a student answering question 33 in skill-level online tutoring systems. Question 33 is composed of skill $s11$ and $s21$. The student answers $s11$ incorrectly three times $t_1 - t_3$ in succession. Even if he answers $s11$ correctly at the timestamp t_4, his mastery of $s11$ is very poor and his mastery of $s11$'s-related skill $s21$ is not good too, so it is very likely to answer $s21$ incorrectly, in fact, he answers $s21$ incorrectly at the timestamp t_5.

Timestamp	Skill	Correctness
t_1	$s11$	0
t_2	$s11$	0
t_3	$s11$	0
t_4	$s11$	1
t_5	$s21$	0

the neural Turing machine, which is an instance of memory-augmented neural networks (MANNs) that offer the ability to quickly encode and retrieve new information [23].

Table 4: The relationship between a_t and c_t in skill-level online tutoring systems.

timestamp	q_t	a_t	S_t	c_t
t_1	$s11$	0	$s21$	0
t_2	$s11$	0	$s21$	0
t_3	$s11$	0	$s21$	0
t_4	$s11$	1	$s21$	1
t_5	$s21$	0	$s11$	0

5.2. Input Features

5.2.1. Answer Information Encoding. Let E^q be the encoding of the student's interaction tuple (q, a), thus $E^q = [e_i^q] \in \{0, 1\}^{2M}$:

$$e_i^q = \begin{cases} 1, & \text{if } i = q + a \times M, \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

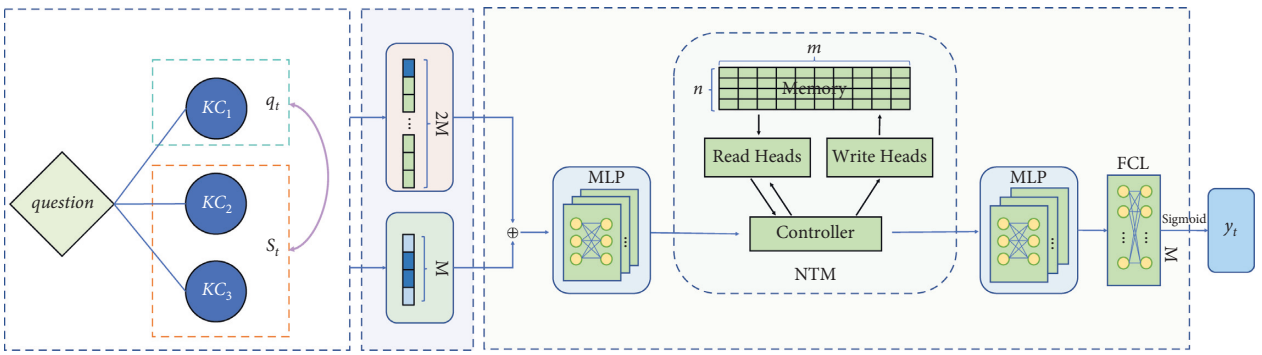

Figure 2: The NSKT framework overview.

5.2.2. RKC Information Encoding. The information of the set S of RKCs related to KC q is encoded E^s with a length of M: $E^s = [e_i^s] \in \{0,1\}^M$:

$$e_i^s = \begin{cases} 0 & i \notin S, \\ 1 & i \in S. \end{cases} \tag{5}$$

5.3. Neural Turing Machines. Neural Turing machines are an instance of memory-augmented neural networks (MANNs) that extend the capabilities of neural networks by coupling them to external memory resources. Experiments show that neural Turing machines have stronger memory capabilities than the LSTM [23], which is very suitable for modeling the students' knowledge state [33–35]. Figure 3 shows a high-level diagram of the neural Turing machine architecture.

As can be seen from Figure 3, the NTM is composed of 4 modules: controller, read heads, write heads, and memory. The controller can be a feed-forward neural network or a recurrent neural network [23, 34] and has read and write heads that access the external memory matrix.

5.4. Reading. Let $\mathbf{M}_t$ be the external memory content which is a $n \times m$ memory matrix at the timestamp t, where n is the number of memory locations and m is the vector dimension at each memory location. The n elements $w_t(i)$ of $\mathbf{w}_t$, which is a vector of weightings over the n locations emitted by a read head at the timestamp t, obey the following constraints:

$$\sum_i w_t(i) = 1, 0 \le w_t(i) \le 1, \forall i. \tag{6}$$

Let $\mathbf{r}_t$ be the read vector of a length m returned by the head at the timestamp t:

$$\mathbf{r}_t \leftarrow \sum_i w_t(i)\mathbf{M}_t(i). \tag{7}$$

5.5. Writing. The memory matrix $\mathbf{M}_t$ at the timestamp t is modified by the erase vector $\mathbf{e}_t$ and the add vector $\mathbf{a}_t$:

$$\begin{aligned} \tilde{\mathbf{M}}_t(i) &\leftarrow \mathbf{M}_{t-1}(i)\left[1 - w_t(i)\mathbf{e}_t\right] \\ \mathbf{M}_t(i) &\leftarrow \tilde{\mathbf{M}}_t(i) + w_t(i)\mathbf{a}_t. \end{aligned} \tag{8}$$

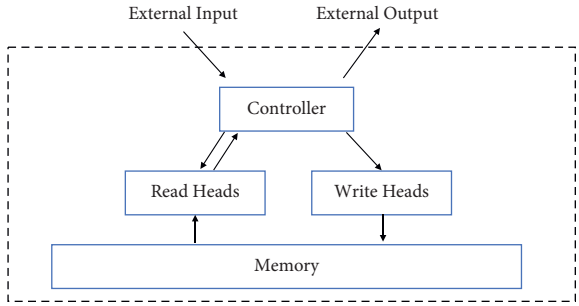

Figure 3: Neural Turing machine architecture.

5.6. Addressing Mechanisms

5.6.1. Focusing on Content. Each head produces a length m key vector $\mathbf{k}_t$ that is used to compute the normalised weighting w_t^c as follows:

$$w_t^c(i) \leftarrow \frac{\exp\left(\beta_t K\left[\mathbf{k}_t, \mathbf{M}_t(i)\right]\right)}{\sum_j \exp\left(\beta_t K\left[\mathbf{k}_t, \mathbf{M}_t(j)\right]\right)}, \tag{9}$$

where β_t is a positive key strength generated by the controller and the similarity measure K is cosine similarity:

$$K[\mathbf{u}, \mathbf{v}] = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \times \|\mathbf{v}\|}. \tag{10}$$

5.6.2. Focusing on Location. The location-based addressing mechanism is designed to facilitate both simple iterations across the locations of the memory and random-access jumps. It does so by implementing a rotational shift of a weighting as follows [23].

Firstly, the interpolation gate g_t is used to blend between the weighting $\mathbf{w}_{t-1}$ and the weighting $\mathbf{w}_t^c$:

$$\mathbf{w}_t^g \leftarrow g_t \mathbf{w}_t^c + (1 - g_t)\mathbf{w}_{t-1}. \tag{11}$$

Furthermore, the model uses a one-dimensional convolution shift kernel to convolve the current weighting w_t^g:

$$\tilde{w}_t(i) \leftarrow \sum_{j=0}^{n-1} w_t^g(j) s_t(i-j), \tag{12}$$

where $\mathbf{s}_t$ is the shift weighting generated by the controller.

To correct the blur that occurs due to the convolution operation, each head emits one further scalar $\gamma_t \geq 1$ whose effect is to sharpen the final weighting as follows:

$$w_t(i) \leftarrow \frac{\widetilde{w}_t(i)^{\gamma_t}}{\sum_j \widetilde{w}_t(j)^{\gamma_t}}. \tag{13}$$

5.7. Controller. The NTM controller in our model is the long short-term memory network [36], which can be formulated by the formulas as follows:

$$\begin{aligned}
\mathbf{f}_t &= \sigma\left(\mathbf{w}_f\left[\mathbf{h}_{t-1}, \mathbf{x}_t\right] + \mathbf{b}_f\right),\\
\mathbf{i}_t &= \sigma\left(\mathbf{w}_i\left[\mathbf{h}_{t-1}, \mathbf{x}_t\right] + \mathbf{b}_i\right),\\
\mathbf{o}_t &= \sigma\left(\mathbf{w}_o\left[\mathbf{h}_{t-1}, \mathbf{x}_t\right] + \mathbf{b}_o\right),\\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh\left(\mathbf{w}_c\left[\mathbf{h}_{t-1}, \mathbf{x}_t\right] + \mathbf{b}_c\right),\\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh\left(\mathbf{c}_t\right).
\end{aligned} \tag{14}$$

$\mathbf{i}, \mathbf{f}, \mathbf{o}, \mathbf{c}, \mathbf{h}$ are the activation matrices of the input gate, the forget gate, the output gate, the memory cell, and the hidden state matrix, respectively. $\mathbf{w}$ and $\mathbf{b}$ are the weight matrix and the bias vector of the corresponding gate, respectively. $\odot$ denotes the Hadamard product. σ and tanh denote the sigmoid and hyperbolic tangent function, respectively:

$$\sigma(x) = \frac{1}{1 + \exp(-x)}, \tag{15}$$

let logits $\in \mathbb{R}^M$ be the output of the last neural network of the NSKT model, the student's mastery of knowledge concepts predicted by the model at the timestamp t is

$$\mathbf{y}_t = \sigma(\text{logits}), \tag{16}$$

where $\mathbf{y}_t \in \mathbb{R}^M$.

5.8. Optimization. The loss function of the model consists of two parts, namely, the answering interaction loss $\mathscr{L}_1$ and the related knowledge concept information loss $\mathscr{L}_2$. Let ℓ be the binary cross entropy loss:

$$\ell(p, a) = -(a \log p + (1 - a)\log(1 - p)). \tag{17}$$

We optimize the average cross entropy loss of the student's interactions as follows:

$$\mathscr{L}_1 = \frac{\sum_t \ell\left(\mathbf{y}_t^{\mathbf{T}} \delta(q_{t+1}), a_{t+1}\right)}{|H| - 1}, \tag{18}$$

where $\delta(q_{t+1})$ is the one-hot encoding of KC q_{t+1} at the timestamp $t + 1$, $|H|$ is the total number of the student's interactions, and $\mathbf{T}$ denotes transpose operation.

The average cross-entropy loss of the related knowledge concept information is

$$\mathscr{L}_2 = \frac{\sum_t \sum_{i=1}^{|S_{t+1}|} \ell\left(\mathbf{y}_t^{\mathbf{T}} \delta(q_i), c_i\right)}{\sum_t |S_{t+1}|}, \tag{19}$$

where $q_i \in S_{t+1}$, c_i is the correctness to skill q_i.

The loss for a single student is represented by $\mathscr{L}$, which is as follows:

$$\begin{aligned}
\mathscr{L} &= \lambda \mathscr{L}_1 + (1 - \lambda)\mathscr{L}_2\\
&= \lambda \frac{\sum_t \ell\left(y_t^T \delta(q_{t+1}), a_{t+1}\right)}{|H| - 1} + (1 - \lambda)\frac{\sum_t \sum_{i=1}^{|S_{t+1}|} \ell\left(y_t^T \delta(q_i), c_i\right)}{\sum_t |S_{t+1}|}\\
&= \sum_t \left(\lambda \frac{\ell\left(y_t^T \delta(q_{t+1}), a_{t+1}\right)}{|H| - 1} + (1 - \lambda)\frac{\sum_{i=1}^{|S_{t+1}|} \ell\left(y_t^T \delta(q_i), c_i\right)}{\sum_t |S_{t+1}|}\right),
\end{aligned} \tag{20}$$

where the hyperparameter λ is the coefficient that determines the proportion of the answering information loss and the related information loss. We use an optimizer to optimize our model. Let Θ be the minimum of $\mathscr{L}$, thus, the training objective of NSKT is as follows:

$$\Theta \leftarrow \text{optimizer}(\mathscr{L}). \tag{21}$$

5.9. Skill Awareness. The student's past interactions in online tutoring systems: $H'_t = \{h'_0, \ldots, h'_t\}$, where $h' = (q_t, a_t, S_t, c_t)$ denotes that the student interaction tuple at the timestamp t. The set of knowledge concepts Set_q that students have answered actually so far is represented as follows:

$$Set_t^q = \{q_i\}_{i=1}^t. \tag{22}$$

The set of knowledge concepts (skills) Set^S answered by NSKT so far is represented as follows:

$$Set_t^S = S_1 \cup \ldots \cup S_t. \tag{23}$$

As shown in Figure 4, when the student answers the next skill q_{t+1} at the next timestamp $t + 1$, even if the student has not answered questions related to skill q_{t+1} before, $q_{t+1} \notin \text{Set}_t^q$, but if NSKT has awareness of skill q_{t+1} so far, $q_{t+1} \in \text{Set}_t^S$, NSKT can predict the student's mastery of skill q_{t+1} accurately.

6. Experiments

In this section, we give a detailed explanation of datasets and experiments conducted to evaluate the performance of the NSKT model and other KT models in three real-world open-source knowledge tracing datasets.

6.1. Datasets. To evaluate KT models' performance, we use three datasets collected from online learning platforms. These three datasets are widely used real-world datasets in KT.

(i) ASSISTments2009 (https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data) (ASSIST09) is provided by the ASSISTment online tutoring platform and is the most widely used dataset in knowledge tracing.

(ii) ASSISTments2017 (https://sites.google.com/view/assistmentsdatamining/dataset/) (ASSIST17) is

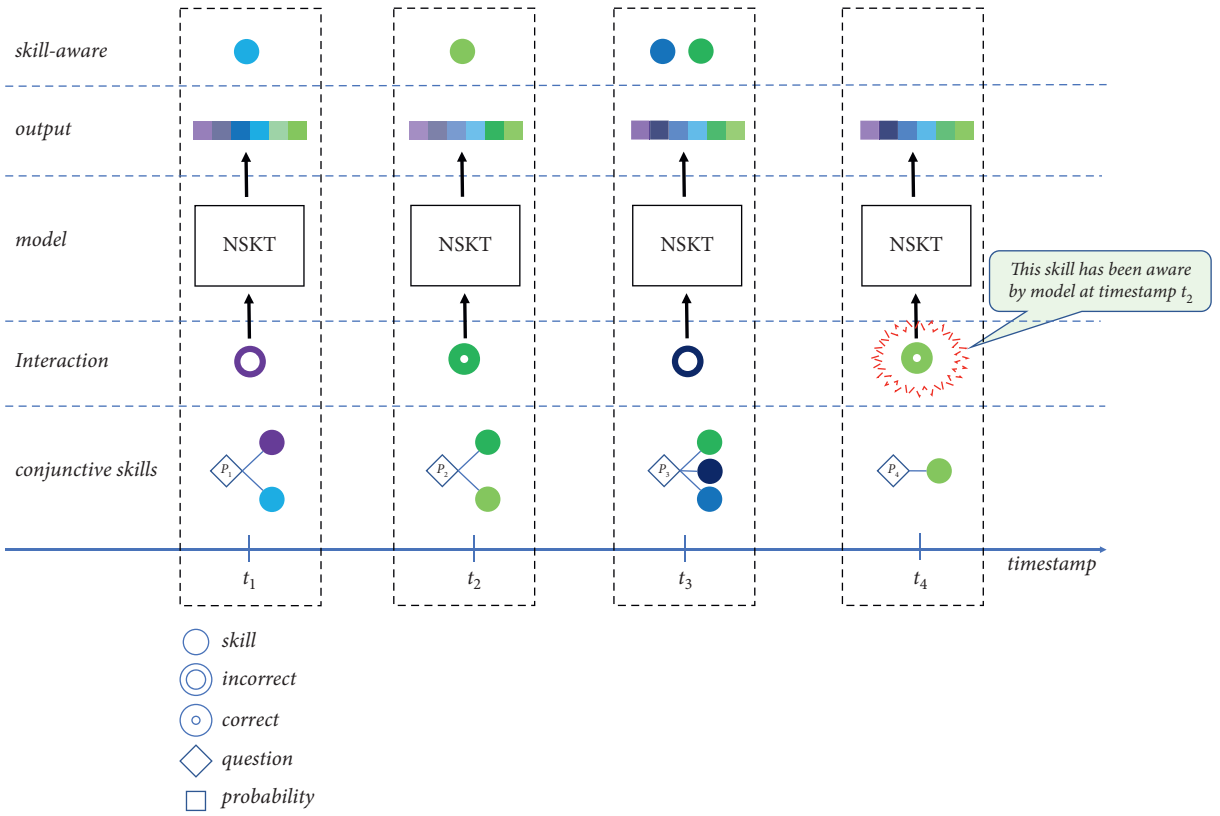

Figure 4: The process of skill awareness in NSKT. Different skills are indicated by different colors.

Table 5: The statistics of the three datasets.

	Students	Skills	Questions	Interactions (K)	MIN[a]	MAX[b]	AVG[c]
ASSIST09	4,162	124	26,688	526[d]	0	3	0.316
ASSIST17	1,709	102	3,162	942	0	2	0.472
EdNet	784,309	189	13,169	962	0	6	1.388

The symbol [a] indicates the minimum value of $|S|$, where 0 means that the question is a single KC (skill) question. The symbol [b] indicates the maximum value of $|S|$. The symbol [c] indicates the average value of $|S|$. The symbol [d]K stands for a thousand.

provided by the 2017 ASSISTments data mining competition and is the latest ASSISTments dataset with the most student responses.

(iii) EdNet (https://github.com/riiid/ednet) is the dataset of all student-system interactions collected over 2 years by Santa, a multiplatform AI tutoring service with more than 780 K users in Korea available through Android, iOS, and Web [37]. We conducted our experiments on EdNet-KT1 which consists of students' question-solving logs and is the record of Santa collected since April 2017 by following the question-response sequence format.

The complete statistical information for the three datasets is shown in Table 5.

The details about the columns in datasets are shown as follows:

ASSISTments:

(i) user_id: the ID of the student
(ii) problem_id: the ID of the problem
(iii) skill_id: the ID of the skill associated with the problem
(iv) 1: Correct on the first attempt
0: Incorrect on the first attempt,

EdNet:

(i) user_id: the ID of the student.
(ii) question_id: the ID of the question.
(iii) tags: the expert-annotated tags for each question.
(iv) correct_answer: the correct answer of each question recorded as a character between a and d inclusively.
(v) user_answer: the answer that the student submitted was recorded as a character between a and d inclusively.

6.2. Dataset Characteristics

(i) ASSIST09 and EdNet: For multiple skill questions, the records of students' interactions will be repeated with different skill taggings and each record

represents the student response to a skill of the question [38].

(ii) ASSIST17: similar to the ASSIST09 dataset, each record in ASSIST17 represents the student response to a skill of the question. However, we noticed the special features of this dataset. A large number of users in the ASSIST17 dataset only answered one skill of multiple skill questions and answered this skill one or more times. The number of multiple skill questions in this situation accounted for 44.88% of the total number of questions answered by students. That is, the student answers one or more skills of multiple skill questions, and the number of responses to a skill may be given once or multiple times.

6.3. Compared Models and Implementation Details. To show the performance of our model and demonstrate the improvement of our model to existing KT models, we compare NSKT against the state-of-the-art KT models. We give the reference GitHub repositories of some KT models.

(i) BKT [5]: Bayesian knowledge tracing uses the hidden Markov model (HMM) to model the students' latent knowledge state as a set of binary variables. We use pyBKT (https://github.com/CAHLR/pyBKT) to implement BKT and set the model parameters: seed = 42, num_fits = 1.

(ii) DKT-LSTM [6]: the DKT-LSTM is the standard deep knowledge-tracing mode. We implemented DKT (https://github.com/chrispiech/DeepKnowledgeTracing) with the LSTM with tanh activation.

(iii) DKT-NTM: DKT is implemented by using the neural Turing machine (https://github.com/MarkPKCollier/NeuralTuringMachine).

(iv) DKVMN [10]: the DKVMN (https://github.com/jennyzhang0215/DKVMN) is a variation of MANNs, which uses a static memory called key and a dynamic memory called value to model the students' knowledge state.

(v) SAKT [3]: SAKT (https://github.com/shalini1194/SAKT) is the KT model based on the self-attention architecture with exercises as attention queries and students' past interactions as attention keys/values.

(vi) DSKT: the skill-aware deep knowledge-tracing model is implemented with the LSTM and tanh activation. We dynamically set the value of the coefficient λ to explore the best performance of DSKT.

(vii) NSKT: NSKT is an NTM-based skill-aware knowledge tracing. We test the performance of NSKT with different values of the coefficient λs to optimize the model's performance.

For all models, we use the Adam optimizer with learning_rate = 0.001, beta1 = 0.9, beta2 = 0.999, and epsilon = $1e-8$ to optimize. The minibatch size and the maximum length of the sequence for all datasets are set to 32 and 100, respectively. We perform standard five-fold cross-validation to evaluate all the KT models in this paper. We conduct experiments on the server with an 8-core 2.50 GHz Intel(R) Xeon(R) Platinum 8163 CPU and 64 GB memory.

6.4. Experimental Results

6.4.1. Models' Performance. We use the area under the receiver operating characteristic curve (AUC) as an evaluation metric to compare prediction performance among the KT models mentioned in Section 6.3. A higher AUC indicates better performance. The test AUC results in the three real-world datasets for all KT models are shown in Table 6. From the experiment results, we can find the following observations:

(i) NSKT performs better than the other competing KT models in all datasets and achieves the average test AUC of 85.38%, 82.35%, and 80.81% in ASSIST09, ASSIST17, and EdNet, respectively.

(ii) DSKT performs better than the DKT-LSTM, achieves the average test AUC of 84.88%, 81.27%, and 79.71% in datasets ASSIST09, ASSIST17, and EdNet, respectively, gaining an average performance improvement of 0.82% (DKT-LSTM achieves the AUC of 84.45%, 80.04%, and 78.91%). NSKT performs better than the DKT-NTM, gaining an average performance improvement of 1.33% (DKT-NTM achieves the AUC of 84.53%, 80.51%, and 79.49%).

(iii) The DKT-NTM model has a better performance than the standard DKT-LSTM in knowledge tracing. The DKT-NTM achieves the average test AUC of 84.53%, 80.51%, and 79.49% in the three datasets, respectively, while the standard DKT-LSTM achieves the average test AUC of 84.45%, 80.04%, and 78.91% in the three datasets, respectively.

(iv) The performance of NSKT is better in dataset ASSIST17, which has more complex data features than those of ASSIST09 and EdNet. NSKT gains an average performance improvement of 2.31% in ASSIST17 compared to the standard DKT-LSTM while improving AUC by 0.93% and 0.90% in ASSIST09 and EdNet, respectively. It proves that NSKT is better in mining hidden information from complex educational data features to improve the accuracy of prediction.

Figure 5 shows the training process of KT models in the three KT datasets. It shows that the DKVMN and SAKT can learn faster than other KT models. The training speed of the DKT-LSTM, DKT-NTM, DSKT, and NSKT is close, but the test AUC of NSKT is the best.

We set the probability to KC q_t predicted by KT models: $P(q_t)$, and assume that students will answer KC q_t correctly if $P(q_t) >= 0.5$ and if $P(q_t) < 0.5$, the student will answer q_t incorrectly:

Table 6: Test AUC results for all datasets (%).

	BKT	DKT-LSTM	DKT-NTM	DKVMN	SAKT	DSKT	NSKT
ASSIST09	72.06	84.45	84.53	84.37	84.70	84.88	**85.38**
ASSIST17	65.25	80.04	80.51	80.55	81.25	81.27	**82.35**
EdNet	66.28	78.91	79.49	79.72	79.83	79.71	**80.81**

Bold values indicate the best performance.

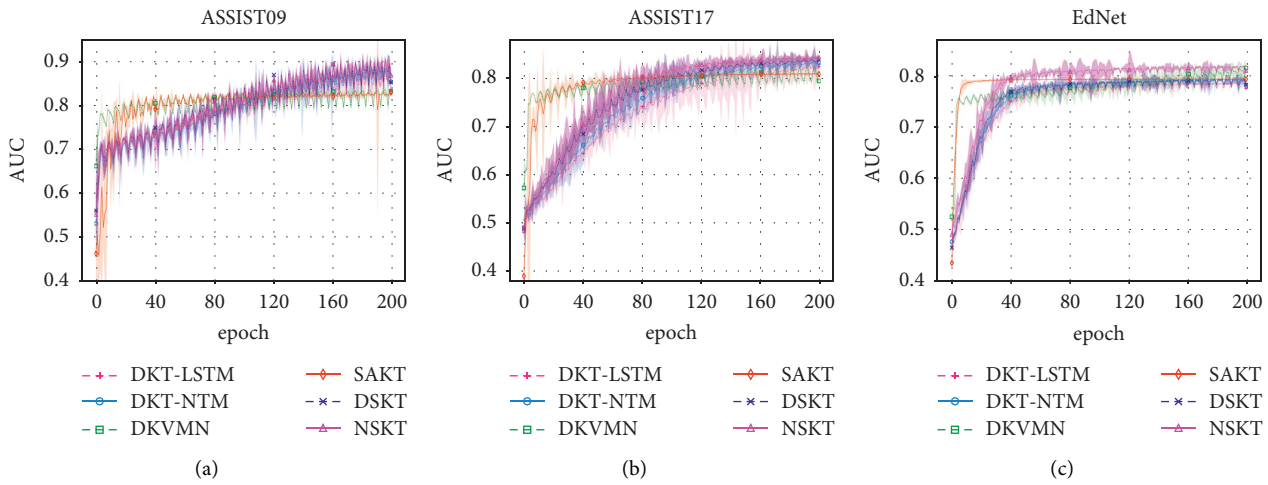

Figure 5: The dynamics for training models in the KT datasets with the best hyperparameter selection for each model and each dataset individually. Each line plots the mean over multiple runs, while the corresponding root mean square error (RMSE) is shown as the shaded area around the mean. (a) ASSIST09. (b) ASSIST17. (c) EdNet.

$$a_t' = \begin{cases} 0 & P(q_t) < 0.5, \\ 1 & P(q_t) > = 0.5. \end{cases} \tag{24}$$

If $a_t' = a_t$, it means that models can predict correctly. Thus, the accuracy of prediction for KT models in the datasets is shown in Figure 6.

Figure 7 shows the performance of DSKT and NSKT under different λ values and the value of λ when models achieve the best performance. From Figure 7, we can draw the following conclusions: the test AUC of DSKT and NSKT is not ideal with a small λ value. However, as the value of λ increases, the test results of DSKT and NSKT get better and better; thus, we recommend $\lambda \geq 0.9$.

6.5. Friedman-Aligned Rank Test. We perform the Friedman-aligned rank test [39] on the AUC test results of the KT models shown in Table 6 by the following formula:

$$X^2 = \frac{12}{nk(k+1)} \sum R_i^2 - 3n(k+1), \tag{25}$$

where R_i is the sum of the ranks of the i-th sample, k is the number of groups of samples, and n is the number of samples in each group. The probability distribution of X^2 can be approximated by that of the chi-squared distribution with $k-1$ degrees of freedom χ_{k-1}^2. Now, we test the null hypothesis, which is as follows:

H_0: there is no significant difference in the performance of the KT models.

The P value P of the Friedman-aligned rank test on test AUC results is

$$P\left(\chi_{k-1}^2 \geq X^2\right) = 0.013 < 0.05. \tag{26}$$

Then, we reject the null hypothesis H_0, which indicates a significant difference in the performance of the KT models.

6.6. Execution Time. We compared the execution time of KT models per 200 batches in each dataset shown in Figure 8. As shown in Figure 8, the BKT model requires the least execution time to train the same size of data. This is because the BKT is not a deep learning knowledge tracing model, and it needs to train fewer parameters. For deep learning knowledge tracing models, the execution times of the DKT-LSTM, DKVMN, and SAKT are close and the execution times of the DKT-NTM and DSKT are close. The execution time of the DKT-NTM is more than that of the DKT-LSTM. The reason can be that the NTM takes more time to access its own external memory matrix. NSKT considers the conjunctive skills of the questions during the training process and needs to access the NTM's external memory matrix to enhance the memory ability of the model. Hence, NSKT has the most execution time, but this is also the reason why NSKT performs better in modelling the students' knowledge state.

The experimental results show that the NTM-based skill-aware knowledge-tracing model has a strong ability to capture the relevance among knowledge concepts and can enhance the model's ability of skill awareness for conjunctive skills and improve the accuracy of prediction in modelling

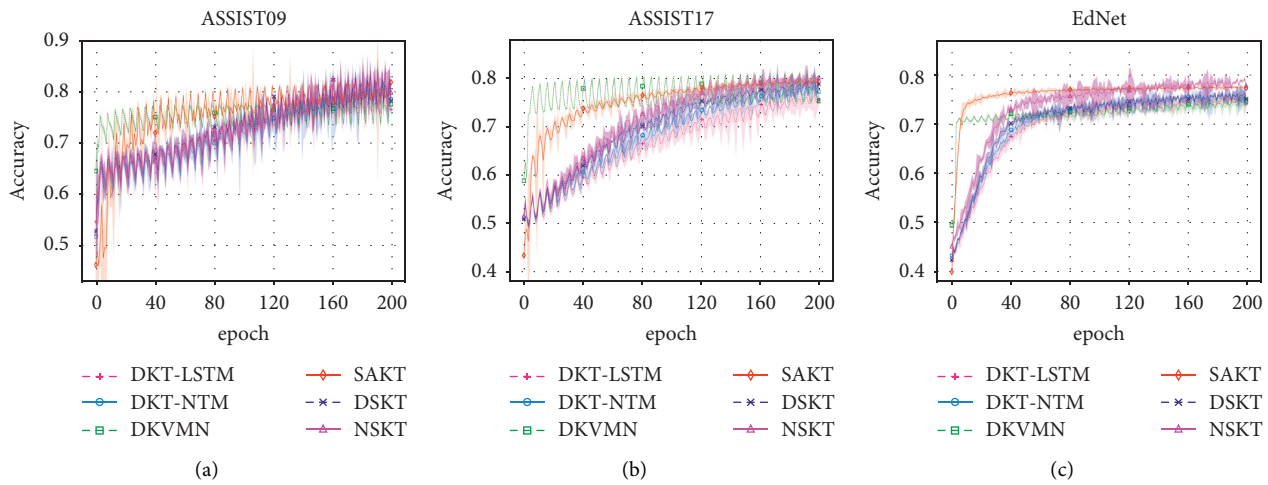

Figure 6: The accuracy of prediction for training models in the KT datasets with the best hyperparameter selection for each model and each dataset individually. Each line plots the mean over multiple runs, while the corresponding root mean square error (RMSE) is shown as the shaded area around the mean. (a) ASSIST09. (b) ASSIST17. (c) EdNet.

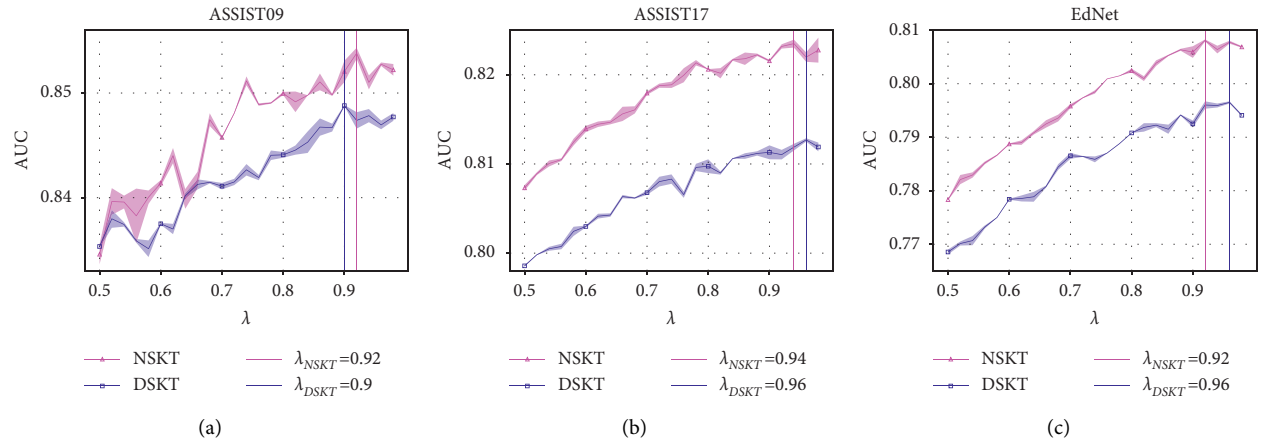

Figure 7: The average test AUC of NSKT and DSKT with different λ values in datasets. Each line plots the mean over multiple runs, while the corresponding root mean square error (RMSE) is shown as the shaded area around the mean. (a) ASSIST09. (b) ASSIST17. (c) EdNet.

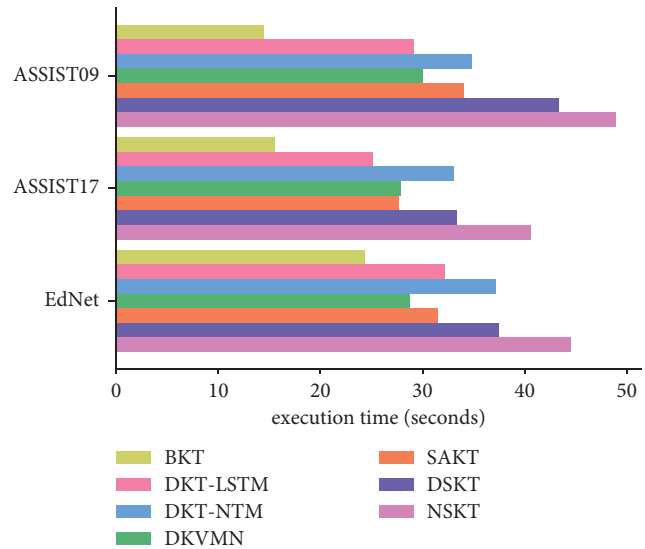

Figure 8: The comparison of the KT models' execution time per 200 batches in each dataset.

the students' knowledge state. Experiments demonstrate that NSKT is effective.

7. Discussion

In this section, we discuss the details of the prediction process of the KT model in modelling the students' knowledge state, as well as the relevance of knowledge concepts and conditional influence between exercises.

7.1. Prediction Process. In our opinion, an excellent KT model not only can predict the probability that students will answer questions correctly at the next timestamp accurately but also can perform well in modelling the students' holistic knowledge concept state.

Analyzing the prediction process of KT models can show the performance of NSKT. We randomly select a student sample U_1 from the ASSIST09 dataset, and the detailed process of DKT and NSKT modelling U_1's knowledge state is shown in Figure 9.

It can be seen from Figure 9(a) that although DKT performs fairly well in prediction, DKT only focuses on the knowledge concepts to be predicted at the next timestamp and does not care about the U_1's mastery of other knowledge concepts. Therefore, after U_1 answers $s32$ correctly $(s32, 1)$ at the timestamp t_3, the model's predicted probability of $s32$ decreases rapidly, indicating that U_1's mastery of $s32$ is getting worse and worse, which should not be consistent with the U_1's real knowledge state shown in Table 7. Because of lacking related knowledge concept (RKC) information, DKT's prediction accuracy and prediction breadth are not ideal.

As shown in Figure 9(b), we use two heatmap subfigures to show the process of modelling the U_1's knowledge state on NSKT. The x-axis of the lower subfigure is the sequence of U_1's interactions (q_t, a_t) and the y-axis is the skill index. The x-axis of the upper subfigure is the RKC S_t and the y-axis is the index of the RKCs S_t.

Because U_1 answers skill 32 (abbreviated as $s32$) correctly $(s33, 1)$ in the first three timestamps $t_1 - t_3$, the predicted probability of $s32$ gets higher and higher and the color of $s32$ in the y-axis of the lower subfigure gets brighter and brighter. As shown in the x-axis of the upper subfigure, $s33$ is the related knowledge concept of $s32$ in the first three timestamps $t_1 - t_3$; thus, the predicted probability of $s33$ gets higher and higher and the color of the $s33$ in the y-axis of the upper subfigure gets brighter and brighter too.

In the next three timestamps $t_4 - t_6$, U_1 answers s33 correctly $(s33, 1)$ in succession, the predicted probability of $s33$ gets higher and higher and the color of $s33$ in the y-axis of the lower subfigure gets brighter and brighter. $s32$ is the related knowledge concept of $s33$, so the predicted probability of $s32$ continues to increase, and the color of $s32$ in the y-axis of the upper subfigure gets brighter and brighter too and remains at a relatively high value.

In the next three timestamps $t_7 - t_9$, U_1 continues to answer $s33$ correctly $(s33, 1)$; however, this $s33$ is a single skill without related knowledge concepts, so only the predicted probability of $s33$ gets higher and higher and the color of $s33$ in the y-axis of the lower subfigure gets brighter and brighter.

At the last timestamp t_{10}, U_1 answer $s37$ correctly $(s37, 1)$, so the predicted probability of $s37$ gets higher and higher and the color of $s32$ in the y-axis of the lower subfigure gets brighter and brighter. Because $s55$ is the related knowledge concept of $s37$, so the predicted probability of $s55$ gets higher and higher and the color of $s55$ in the y-axis of the upper subfigure gets brighter and brighter too.

In contrast, we randomly select a student sample U_2 with a low answering accuracy shown in Table 8.

The process of DKT and NSKT modelling the U_2's knowledge state is shown in Figure 10. It can be seen from Figure 10(a) that DKT models the U_2's knowledge state almost accurately, but the prediction breadth is not enough.

As shown in Figure 10(b), NSKT, like DKT, models the U_2's knowledge state accurately and performs better in prediction breadth. At the timestamp t_4, U_2 answers $s95$ incorrectly many times, the predicted probability of $s95$ gets lower and lower and the color of $s95$ in the y-axis of the lower subfigure gets darker and darker. As shown in the x-axis of the upper subfigure, $s2$ is the related knowledge concept of $s95$; thus, the predicted probability of $s2$ gets lower and lower and the color of $s33$ in the y-axis of the upper subfigure gets darker and darker too.

It can be concluded from Figures 9 and 10 that NSKT performs better in prediction accuracy and prediction breadth and can better model the students' knowledge state. NSKT not only focuses on students' mastery of the knowledge concept to be predicted at the next timestamp but also focuses on the students' mastery of the related knowledge concepts. This is where NSKT is superior to other existing KT models, and NSKT performs better in modelling the students' knowledge state than DKT [4].

7.2. Pearson Correlation Coefficient. In this paper, we use the Pearson correlation coefficient as the metric to measure the correlation among skills. By estimating the covariance and standard deviation of the sample, we can get the sample Pearson coefficient r:

$$r_{(X,Y)} = \frac{\sum_{i=1}^{n} (X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n} (X_i - \overline{X})^2} \sqrt{\sum_{i=1}^{n} (Y_i - \overline{Y})^2}}. \tag{27}$$

Figures 11 and 12 show the comparison of skill Pearson correlations of U_1's interactions and U_2's interactions on DKT and NSKT, respectively. Figures 11(a) and 12(a) show the skill Pearson correlation on DKT, and Figures 11(b) and 12(b) show the skill Pearson correlation on NSKT. It can be seen from the figures that DKT can only mine the correlation among the skills that have been answered in the past, indicating that DKT cannot effectively discover the relevance among knowledge concepts. As shown in Figures 11(b) and 12(b), NSKT can discover the correlation among four skills, while DKT can only discover among three. For example, it can be seen from Figure 11(b) that the Pearson correlation between $s32$ and $s55$ on NSKT of U_1's interactions is

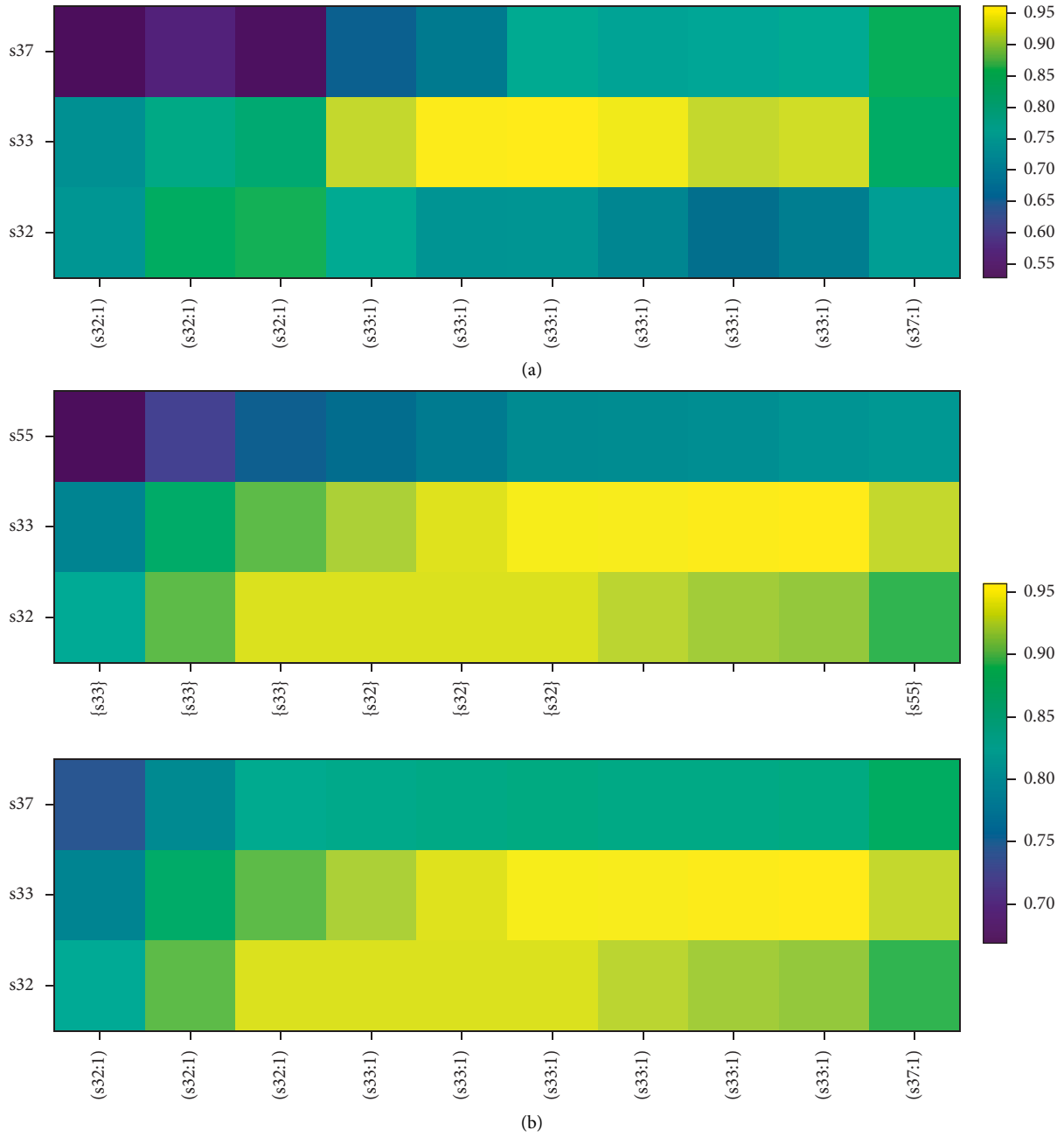

Figure 9: Comparison of the prediction process of U_1 on DKT and NSKT. The color of the heatmap indicates the predicted probability that U_1's mastery of skills after interaction (q_t, a_t) at the timestamp t. The yellower the color, the higher the probability. (a) Heatmap for the prediction process of DKT. The x-axis is the sequence of U_1's interactions $(q_t; a_t)$ and the y-axis is the skill index. (b) Heatmap for the prediction process of NSKT. The x-axis of the lower subfigure is the sequence of U_1's interactions $(q_t; a_t)$ and the y-axis is the skill index. The x-axis of the upper subfigure is the RKC S_t and the y-axis is the skill index of the RKCs S_t.

Table 7: Skill maps of ASSIST09 and U_1's interaction accuracy.

Skill index	Skill name	Accuracy (%)
32	Ordering positive decimals	100
33	Ordering fractions	100
37	Addition whole numbers	100
55	Absolute value	100

Table 8: Skill maps of ASSIST17 and U_2's interaction accuracy.

Skill index	Skill name	Accuracy (%)
2	Point plotting	30
4	Reading graph	100
34	Equation solving	50
95	Divisibility	30

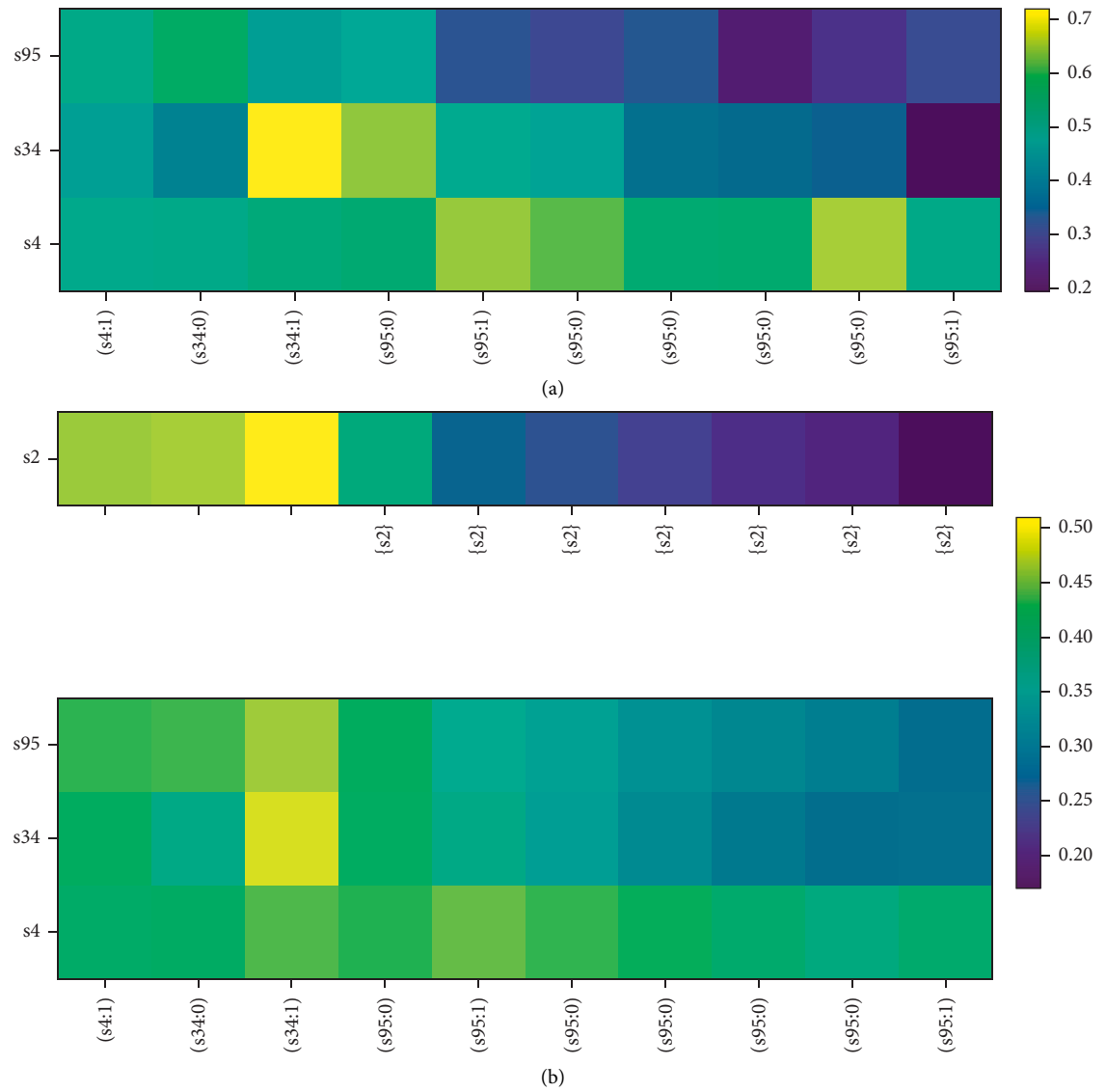

Figure 10: Comparison of the prediction process of U_2 on DKT and NSKT. The color of the heatmap indicates the predicted probability that U_2's mastery of skills after interaction (q_t, a_t) at the timestamp t. The yellower the color, the higher the probability. (a) Heatmap for the prediction process of DKT. The x-axis is the sequence of U_2's interactions $(q_t; a_t)$ and the y-axis is the skill index. (b) Heatmap for the prediction process of NSKT. The x-axis of the lower subfigure is the sequence of U_2's interactions $(q_t; a_t)$ and the y-axis is the skill index. The x-axis of the upper subfigure is the RKC S_t and the y-axis is the skill index of the RKCs S_t.

$r_{(s32,s55)} = 0.31$, which means there is a weak positive correlation between $s32$ and $s55$.

The Pearson correlation between $s33$ and $s55$ on NSKT of U_1's interactions is $r_{(s33,s55)} = 0.92$, which means there is a strong positive correlation between $s33$ and $s55$. Through the above examples, we can conclude that NSKT performs better in the ability of discovering latent relevance among knowledge concepts than existing KT models.

7.3. Knowledge Concepts' Discovery. NSKT can learn latent knowledge concept substructure among skills without expert annotations and can cluster related skills into a cluster, which denotes a knowledge concept (KC) class [6].

Figure 13 shows the visualization of using k-means to cluster the skill representation vectors, which have been performed by the t-SNE method [40, 41]. All skills are clustered into eight clusters, and each cluster can represent a knowledge concept class. Skills in the same cluster are labeled with the same color, and those skills have strong relevance and similarity. For example, $s32$ and $s33$ do have a strong relevance and similarity because they are very close in Figure 13, which further proves that NSKT has a stronger ability of discovering skill latent relevance information than existing KT models.

We have explored latent conditional influence between exercises by

$$J_{i \longrightarrow j} = \frac{y(j/i)}{\sum_k y(j/k)}, \tag{28}$$

where $y(j/i)$ is the correctness probability assigned by NSKT to exercise j when exercise i is answered correctly in the first

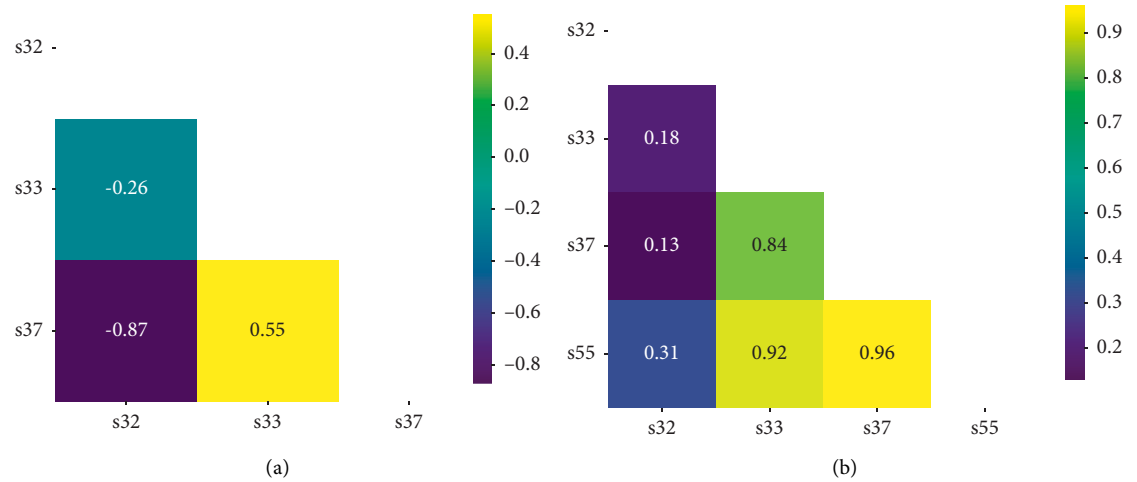

Figure 11: Comparison of U_1's interaction skill Pearson correlations on DKT and NSKT ((a) DKT and (b) NSKT). Both the x-axis and the y-axis are the skills in U_1's interactions.

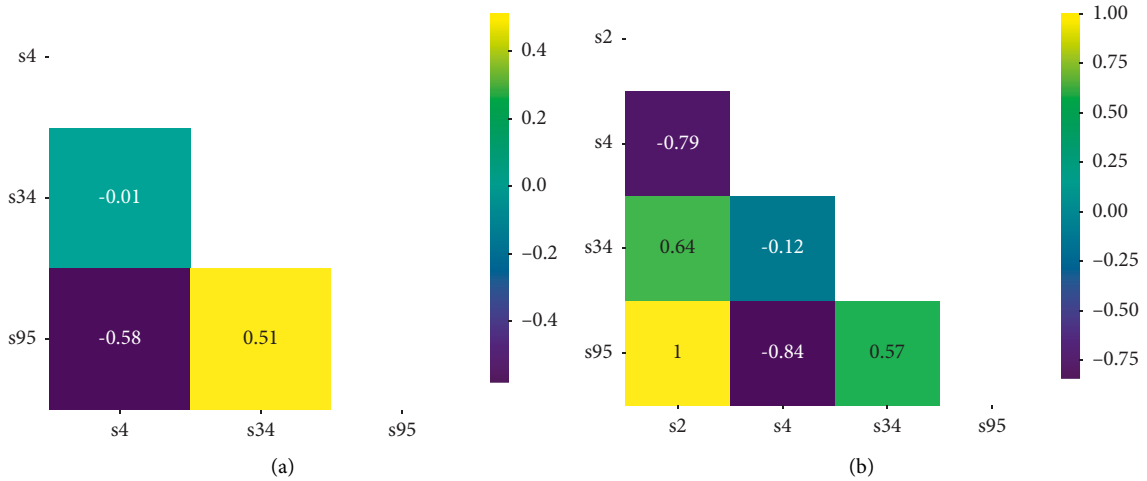

Figure 12: Comparison of U_2's interaction skill Pearson correlations on DKT and NSKT ((a) DKT and (b) NSKT). Both the x-axis and the y-axis are the skills in U_2's interactions.

Figure 13: Knowledge concept visualization in ASSIST09 and conditional influences between exercises.

time step [6]. We have shown a latent conditional influence relationship among the exercises corresponding to Figure 9(b) interactions. We have marked them with arrow symbols in Figure 13. The line width indicates connection strength, and nodes may be connected in both directions. We only show edges with an influence threshold greater than 0.08. Attached ASSIST09 skill maps are shown in Figure 13 (we only show 110 skills with the skill name).

8. Conclusion

In this work, we proposed a novel NTM-based skill-aware knowledge-tracing model for conjunctive skills, which can capture the relevance among the multiple knowledge concepts of questions to predict students' mastery of knowledge concepts (KCs) more accurately and to discover more latent relevance among knowledge concepts effectively. In order to better model the students' knowledge state, we adopt the neural Turing machines, which use the external memory matrix to augment memory ability. Furthermore, NSKT relates knowledge concepts (KCs) to related knowledge concepts (RKCs) as a whole to enhance the model's ability of skill awareness and improve prediction accuracy and prediction breadth. Experiments in the real-world KT datasets demonstrate that the NTM-based knowledge concept skill-aware knowledge-tracing model (NSKT) outperforms existing state-of-the-art KT models in modelling the students' knowledge state and discovering latent relevance among knowledge concepts.

For future studies, we will focus on mining hidden associations among knowledge concepts and building students' personalized answering paths in intelligent tutoring systems. Furthermore, we will construct the holistic structure of knowledge concepts to enhance students' understanding of how the overall knowledge affects each other.

Data Availability

The datasets used to support the findings of this study are included within the article and are available from the corresponding author on reasonable request too.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the CCF-AFSG Research Fund (Grant number: CCF-AFSG RF20200014) and the Science and Technology Project of Gansu (Grant numbers: 21YF5GA102, 21YF5GA006, 21ZD8RA008).

References

[1] J. Self, "Theoretical foundations for intelligent tutoring systems," *Journal of Artificial Intelligence in Education*, vol. 1, no. 4, pp. 3–14, 1990.

[2] M. Feng, N. Heffernan, and K. Koedinger, "Addressing the assessment challenge with an online system that tutors as it assesses," *User Modeling and User-Adapted Interaction*, vol. 19, no. 3, pp. 243–266, 2009.

[3] S. Pandey and G. Karypis, "A self-attentive model for knowledge tracing," in *Proceedings of the 12th International Conference on Educational Data Mining*, pp. 384–389, Montréal, Canada, July 2019.

[4] C. K. Yeung and D. Y. Yeung, "Addressing two problems in deep knowledge tracing via prediction-consistent regularization," in *Proceedings of the 5th Annual ACM Conference on Learning at Scale*, pp. 1–10, London, UK, June 2018.

[5] A. T. Corbett and J. R. Anderson, "Knowledge tracing: modeling the acquisition of procedural knowledge," *User modelling and user-adapted interaction*, vol. 4, no. 4, pp. 253–278, 1995.

[6] C. Piech, J. Spencer, J. Huang et al., "Deep knowledge tracing," in *Proceedings of the 28th Advances in Neural Information Processing Systems*, pp. 505–513, Montreal, Canada, December 2015.

[7] S. Shen, Q. Liu, E. Chen et al., "Convolutional knowledge tracing: modeling individualization in student learning process," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1857–1860, New York, NY, USA, July 2020.

[8] W. Wang, T. Liu, L. Chang, T. Gu, and X. Zhao, "Convolutional recurrent neural networks for knowledge tracing," in *Proceedings of the 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 287–290, Chongqing, China, July 2020.

[9] V. Mandalapu, J. Gong, and L. Chen, "Do we need to go deep? Knowledge tracing with big data," 2021, https://arxiv.org/abs/2101.08349.

[10] J. Zhang, X. Shi, I. King, and D. Y. Yeung, "Dynamic key-value memory networks for knowledge tracing," in *Proceedings of the 26th international conference on World Wide Web*, pp. 765–774, Perth, Australia, April 2017.

[11] S. Liu, R. Zou, J. Sun et al., "A hierarchical memory network for knowledge tracing," *Expert Systems with Applications*, vol. 177, Article ID 114935, 2021.

[12] T. Oya and S. Morishima, "LSTM-SAKT: LSTM-encoded SAKT-like transformer for knowledge tracing," 2021, https://arxiv.org/abs/2102.00845.

[13] S. Pu, M. Yudelson, L. Ou, and Y. Huang, "Deep knowledge tracing with transformers," in *Proceedings of the 21st International Conference on Artificial Intelligence in Education*, pp. 252–256, Ifrane, Morocco, July 2020.

[14] S. Pandey and J. Srivastava, "RKT: relation-aware self-attention for knowledge tracing," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 1205–1214, Ireland, Europe, October 2020.

[15] Y. Choi, Y. Lee, J. Cho et al., "Towards an appropriate query, key, and value computation for knowledge tracing," in *Proceedings of the Seventh ACM Conference on Learning @ Scale*, pp. 341–344, Chicago, IL, USA, August 2020.

[16] T. Wang, F. Ma, and J. Gao, "Deep hierarchical knowledge tracing," in *Proceedings of the 12th International Conference on Educational Data Mining*, pp. 671–674, Montréal, Canada, July 2019.

[17] A. Ghosh, N. Heffernan, and A. S. Lan, "Context-aware attentive knowledge tracing," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2330–2339, New York, NY, USA, August 2020.

[18] Y. Liu, Y. Yang, X. Chen, J. Shen, H. Zhang, and Y. Yu, "Improving knowledge tracing via pre-training question

embeddings," in *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pp. 1577–1583, Yokohama, Japan, January 2020.

[19] H. Nakagawa, Y. Iwasawa, and Y. Matsuo, "Graph-based knowledge tracing: modelling student proficiency using graph neural network," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 156–163, Thessaloniki, Greece, October 2019.

[20] Q. Liu, Z. Huang, Y. Yin et al., "Ekt: exercise-aware knowledge tracing for student performance prediction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 1, pp. 100–115, 2021.

[21] S. Tong, Q. Liu, W. Huang et al., "Structure-based knowledge tracing: an influence propagation view," in *Proceedings of the 20th International Conference on Data Mining*, pp. 541–550, Sorrento, Italy, November 2020.

[22] Y. Yang, J. Shen, Y. Qu et al., "GIKT: a graph-based interaction model for knowledge tracing," in *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pp. 299–315, Bilbao, Spain, September 2020.

[23] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," 2014, https://arxiv.org/abs/1410.5401.

[24] M. Collier and J. Beel, "Implementing neural turing machines," in *Proceedings of the 27th International Conference on Artificial Neural Networks*, pp. 94–104, Rhodes, Greece, October 2018.

[25] A. Graves, G. Wayne, M. Reynolds et al., "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, no. 7626, pp. 471–476, 2016.

[26] F. Lord, *Applications of Item Response Theory to Practical Testing Problems*, Erlbaum Associates, Mahwah, NJ, USA, 1980.

[27] F. Lord, "A Theory of Test Scores," *Psychometric monographs*, vol. 17, 1952.

[28] H. Cen, K. Koedinger, and B. Junker, "Learning factors analysis - a general method for cognitive model evaluation and improvement," in *Proceedings of the International Conference on Intelligent Tutoring Systems*, pp. 164–175, Jhongli, Taiwan, June 2006.

[29] P. I. Pavlik, H. Cen, and K. R. Koedinger, "Performance factors analysis–a new alternative to knowledge tracing," in *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, pp. 531–538, Brighton, England, July 2009.

[30] K. H. Wilson, Y. Karklin, B. Han, and C. Ekanadham, "Back to the basics: Bayesian extensions of irt outperform neural networks for proficiency estimation," in *Proceedings of the 9th International Conference on Educational Data Mining*, pp. 539–544, Raleigh, North Carolina, June 2016.

[31] Y. Su, Z. Cheng, P. Luo et al., "Time-and-Concept enhanced deep multidimensional item response theory for interpretable knowledge tracing," *Knowledge-Based Systems*, vol. 218, Article ID 106819, 2021.

[32] H. Cen, K. Koedinger, and B. Junker, "Comparing two IRT models for conjunctive skills," in *Proceedings of the Intelligent Tutoring Systems*, pp. 796–798, Montreal, Canada, June 2008.

[33] S. Malekmohamadi Faradonbe, F. Safi-Esfahani, and M. Karimian-kelishadrokhi, "A review on neural turing machine (NTM)," *SN Computer Science*, vol. 1, no. 6, p. 333, 2020.

[34] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "One-shot learning with memory-augmented neural networks," 2017, https://arxiv.org/abs/1605.06065.

[35] J. Zhao, S. Bhatt, C. Thille, N. Gattani, and D. Zimmaro, "Cold start knowledge tracing with attentive neural turing machine," in *Proceedings of the Seventh ACM Conference on Learning @ Scale*, pp. 333–336, Chicago, IL, USA, June 2020.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] Y. Choi, Y. Lee, D. Shin et al., "EdNet: a large-scale hierarchical dataset in education," in *Proceedings of the 21st International Conference on Artificial Intelligence in Education*, pp. 69–73, Ifrane, Morocco, July 2020.

[38] X. Xiong, S. Zhao, E. G. Van Inwegen, and J. E. Beck, "Going deeper with deep knowledge tracing," in *Proceedings of the 9th International Conference on Educational Data Mining*, pp. 545–550, Raleigh, NC, USA, June 2 2016.

[39] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.

[40] L. Van Der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 11, pp. 2579–2605, 2008.

[41] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the 5th Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14, pp. 281–297, Oakland, California, January 1967.