Improving Knowledge Tracing with Collaborative Information

Ting Long[1], Jiarui Qin[1], Jian Shen[1], Weinan Zhang[1], Wei Xia[2],
Ruiming Tang[2], Xiuqiang He[2], Yong Yu[1]*
[1]Shanghai Jiao Tong University, [2]Huawei Noah's Ark Lab
{longting,r_ocky,wnzhang}@sjtu.edu.cn,qinjr@icloud.com,yyu@apex.sjtu.edu.cn
{xiawei24,tangruiming,hexiuqiang1}@huawei.com

ABSTRACT

Knowledge tracing, which estimates students' knowledge states by predicting the probability that they correctly answer questions, is an essential task for online learning platforms. It has gained much attention in the decades due to its importance to downstream tasks like learning material arrangement, etc. The previous deep learning-based methods trace students' knowledge states with the explicitly intra-student information, *i.e.*, they only consider the historical information of individuals to make predictions. However, they neglect the inter-student information, which contains the response correctness of other students who have similar question-answering experiences, may offer some valuable clues. Based on this consideration, we propose a method called Collaborative Knowledge Tracing (CoKT) in this paper, which sufficiently exploits the inter-student information in knowledge tracing. It retrieves the sequences of peer students who have similar question-answering experiences to obtain the inter-student information, and integrates the inter-student information with the intra-student information to trace students' knowledge states and predict their correctness in answering questions. We validate the effectiveness of our method on four real-world datasets and compare it with 10 baselines. The experimental results reveal that CoKT achieves the best performance.

CCS CONCEPTS

• **Information systems** → **Personalization**; • **Applied computing** → **E-learning**.

KEYWORDS

knowledge tracing; sequence retrieval; correctness prediction

ACM Reference Format:
Ting Long[1], Jiarui Qin[1], Jian Shen[1], Weinan Zhang[1], Wei Xia[2],, Ruiming Tang[2], Xiuqiang He[2], Yong Yu[1]. 2022. Improving Knowledge Tracing with Collaborative Information. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining (WSDM '22), February 21–25, 2022, Tempe, AZ, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3488560.3498374

*Corresponding author.

WSDM '22, February 21–25, 2022, Tempe, AZ, USA

ACM ISBN 978-1-4503-9132-0/22/02…$15.00
https://doi.org/10.1145/3488560.3498374

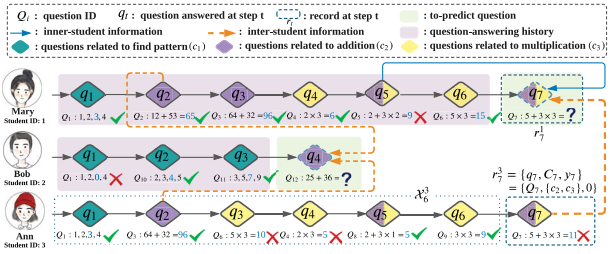


Figure 1: An example of knowledge tracing.

1 INTRODUCTION

With the popularity of online education, automatically estimating the level of students mastering concepts becomes important for online learning platforms, since it is a prerequisite for these sites to offer the services like question recommendation [20], adaptive testing [38], learning path suggestion [10, 17] and learning material arrangement [39]. *Knowledge tracing* is studied to address this issue, which assesses the level of students mastering concepts by predicting the probability that they correctly answer the concept-related questions. That is, the inputs of a knowledge tracing model are a new question and the question-answering history of a student, and the output is the probability that the student correctly answers the question. An example is shown in Figure 1, where we aim to use the question-answering history of Mary and Bob to predict the probabilities that they correctly answer new questions.

Many outstanding deep learning-based works are proposed to solve the knowledge tracing task. Some of them adopt the auto-regressive models [1, 4, 15, 21, 25, 33, 40], like Recurrent Neural Networks (RNNs) [14] or Memory Networks [13, 41], to exploit the sequential information in question-answering history of students. These methods use the hidden states of auto-regressive models to represent the level of students mastering concepts, which are usually termed by *knowledge states* [8]. The knowledge states are learned by feeding the question representation and correctness of students' responses to the RNN (memory) cells at each time step, and will be used to predict the probability of students correctly answering questions. The attention mechanism is another major approach for modeling students' question-answering history [6, 12, 23, 24, 31]. They rely on the attention to identify the importance of historical question-answering records and then make prediction based on students' historical performance.

Although the previous methods have significant achievements, their prediction results are based on the explicite intra-student information, *i.e.*, they use the information in individual question-answering history to make prediction. The intra-student information does offer help when students answer questions which are relevant to the historical questions, *e.g.*, when Mary answers the

question Q_7 in Figure 1, her performance on Q_5 could act as a clue for the prediction on Q_7. However, the help is limited when students answer questions that are irrelevant to the historical questions. Considering the question Q_{12} Bob aims to answer in Figure 1, since the concept *find pattern* is irrelevant to *addition*, it is difficult to use Bob's performance on *find pattern*-related questions to estimate the probability that Bob correctly answering *addition*-related question. In that case, the inter-student information, composed by the records of other students who have similar question-answering experiences, offers some valuable clues for the prediction. Since both Mary and Ann could correctly answer the questions related to *addition* after they correctly answer the questions related to *find pattern*, it is highly possible that answering the questions related to *addition* is easy for the students with solid mastery on *find pattern*. Thus, there is a high probability for Bob to answer Q_{12} correctly. Hence, both the intra-student information and inter-student information benefit the performance of knowledge tracing models. However, the previous methods only explicitly consider the intra-student information, while the inter-student information has never been explicitly considered in knowledge tracing solutions.

In this paper, we proposed a method called *Collaborative Knowledge Tracing (CoKT)*, which introduces the inter-student information to the previous intra-student knowledge tracing. Specifically, we design a retrieval-based mechanism, which retrieves the question-answering sequences of other students who have similar question-answering experiences, to obtain the inter-student information. We integrate the inter-student information with the intra-student information as a clue to trace students' knowledge states and estimate the probability of students correctly answering questions. We compare our method with 10 existing knowledge tracing models on four public datasets. The experimental results demonstrate the efficacy and superiority of our proposed method.

The contributions of our paper are summarized as:

- We explicitly integrate the intra-student information and inter-student information to trace students' knowledge states. To our knowledge, CoKT is the first work which explicitly considers both intra-student and inter-student information simultaneously.
- We design an effective mechanism which obtains inter-student information by retrieving the question-answering sequences of the students who have similar question-answering experiences. The mechanism is model-agnostic and easy-to-deploy, thus could be incorporated with different base models.
- Extensive experiments on four real-world datasets show that CoKT outperforms the state-of-the-art models. Further investigations verify our method is efficient to boost the performance of knowledge tracing models.

2 RELATED WORKS

The knowledge tracing methods can be grouped into *traditional methods* and *deep learning-based methods*. The deep learning-based methods are more effective in general [43].

Most *traditional methods* consider the factors related to learning. One group is built based on *Bayesian Knowledge Tracing (BKT)* [8, 44]. BKT considers four factors affecting students' responses: initial knowledge states, learning rate, slip probability, and guess probability, and it uses Hidden Markov Model to estimate students' knowledge states. Another typical type is *Factor Analysis methods* [36, 42]. The simplest model is the *Item Response Theory (IRT)* [9]. It measures students' ability and the difficulty of questions to make prediction. Recent works elaborate the factors related to learning. For instance, Vie and Kashima [36] introduced the factors like school ID, teacher ID, and they find that the performance becomes better as the number of factors increases.

One representative group in *deep learning-based methods* is *single-state methods*. The single-state methods maintain one vector to represent students' knowledge states. DKT [25] is a typical single-state method, which uses the hidden state of LSTM [14] to represent students' knowledge state, and estimates the probability of students correctly answering questions according to their knowledge states. Many works extend based on DKT: Nagatani et al. [21] considered the forgetting behavior; Chen et al. [4] labeled the prerequisite relations among concepts; Su et al. [32] and Huang et al. [15] encoded question embedding with text description; Liu et al. [18] pre-trained the embeddings of questions. There are also some works maintaining multiple vectors to represent students' knowledge states, which are denoted as *multi-state methods*. Zhang et al. [45] proposed a model called DKVMN, which uses a Key-value memory network to store students' knowledge states, and it makes prediction based on all the vectors of knowledge states. Following DKVMN, Nakagawa et al. [22] and Tong et al. [34] introduced concept graph; Abdelrahman and Wang [1] incorporated the LSTM and Memory network in knowledge states update. Another representative group in deep learning-based methods is *attention-based methods*. They apply attention to identify the relevance between the to-predict question and the historical questions, while making predictions based on students' historical performance. One typical of them is SAKT [23], which adopts self-attention [35] to obtain the weight of the historical performance. There are also many extensions: Pandey and Srivastava [24] introduced the relation of questions and students' forgetting behavior, Choi et al. [6] replaced the self-attention with Transformer [35], Ghosh et al. [12] introduced the decay in the weight of attention. Except for the self-attention and Transformer, some works use the dot product [30] or cosine similarity [15] to perform the attention mechanism.

Although these deep learning-based methods have achieved sound results, they only explicitly consider the intra-student information in individual question-answering history, which limits the performance of models. In this paper, we explicitly integrate the inter-student information with the intra-student information to address the limitation in previous knowledge tracing models and improve their performance.

3 PROBLEM DEFINITION

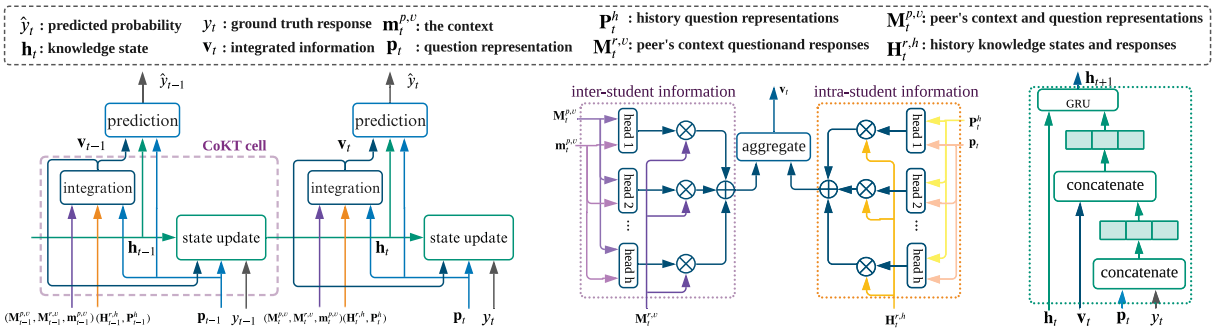
In this section, we briefly introduce the knowledge tracing task with the notations used throughout the paper. We summarize them in Table 1.

Suppose a student, who is denoted as u [1], uses an online learning platform to answer questions. We define her knowledge state as:

Definition 3.1. (**Knowledge State**). *The student's knowledge state denotes the level she masters concepts. We represent the knowledge state of u at step t as a d_h dimensional vector $\boldsymbol{h}_t$, $\boldsymbol{h}_t \in \boldsymbol{R}^{d_h}$.*

We define the question-answering records of u as

[1] For the ease of presentation, we omit u in our notation when there is no ambiguity.



(a) The framework of CoKT (b) The integration module (c) The state update module

Figure 2: The Collaborative Knowledge Tracing (CoKT) model.

Definition 3.2. (**Record**). *Given a student u, the record that her answers question at step i is r_i^u. r_i^u is a triple, $r_i^u = \{q_i, C_i, y_i\}$. q_i represents the question she answers at step i. C_i denotes the set of the concepts which are related to q_i. $C_i = \{c_j\}^{|C_i|}$ and $|C_i| \geq 1$, c_j represents the ID of the concepts in C_i. y_i denotes the correctness of the student's response on q_i:*

$$y_i = \begin{cases} 1, & \text{if the student's answer is right;} \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

An example is illustrated in Figure 1, in which Ann's 7th question-answering record is represented as r_7^3. For an arbitrary record, we define its question-answering history and context as:

Definition 3.3. (**Question-answering History of Record and Context of Record**). *Given a record r_i^u, the question-answering history of r_i^u is the records which are owned by the same student u and generated before r_i^u. We represent the question-answering history of r_i^u as $\mathcal{X}_{i-1}^u = \{r_1^u, r_2^u, ..., r_{i-1}^u\}$. The context of r_i^u is the vector representation of $\mathcal{X}_{i-1}^u$, which represents the aggregation of the previous $|i-1|$ records and is denoted as $\boldsymbol{h}_i^v \in \boldsymbol{R}^{d_h}$.*

Considering the r_7^3 in Figure 1, the question-answering history of r_7^3 is $\mathcal{X}_6^3 = \{r_1^3, r_2^3, r_3^3, r_4^3, r_5^3, r_6^3\}$.

For an arbitrary student u in the online learning platform, there are many other students who have similar question-answering experiences with u, which offers clues for estimating the probability of u correctly answering questions. Thus, we define:

Definition 3.4. (**Similar Peer Record and Similar Peer Subsequence**). *Given a record of student u with unknown correctness of response, $r_t^u = \{q_t, C_t, NA\}$ (NA denotes the answer has not been given by the student), its similar peer record $r_i^{\bar{u}}$ should satisfy three conditions: (1) its corresponding question is the same to q_t or its corresponding concepts are the same to C_t; (2) r_t^u and $r_i^{\bar{u}}$ are generated by different students, i.e., $u \neq \bar{u}$. (3) student $\bar{u}$ have answered same questions or the questions with same concepts with student u before step i. We denote the similar peer records set of r_t^u as $R_t^u = \{r_i^{\bar{u}}\}^{|R|}$, where $|R|$ denotes the number of similar peer records. $\forall\, r_i^{\bar{u}} \in R_t^u$, its question-answering history $\mathcal{X}_{i-1}^{\bar{u}}$ is a similar peer sub-sequence of r_t^u. We denote the similar peer sub-sequences set of r_t^u as $\mathcal{S}_t^u = \{\mathcal{X}_{i-1}^{\bar{u}}\}^{|R|}$.*

Considering the example in Figure 1, we can observe that (1) r_7^1 and r_7^3 share same question Q_7; (2) r_7^1 is generated by Mary, and r_7^3 is generated by Ann; (3) both Ann and Mary have answered the questions related to *addition, multiplication and find pattern*. Thus, r_7^3 is a similar peer record of r_7^1, and $\mathcal{X}_6^3$ is a similar peer sub-sequence of r_7^1 accordingly.

The task of knowledge tracing is formulated as predicting the probability that the student u will correctly answer the question q_t given the question-answering history of $r_t^u = \{q_t, C_t, NA\}$, *i.e.*, $Pr(y_t = 1|q_t, C_t, \mathcal{X}_{t-1}^u)$. Since the question-answering records of other students are available in real-world scenario, and there are some students who have similar question-answering experiences with student u, we extract the inter-student information from the similar peer records and similar peer sub-sequences of r_t^u as a clue for prediction. Thus, in our model the task is formulated as predicting $Pr(y_t = 1|q_t, C_t, \mathcal{X}_{t-1}^u, \mathcal{S}_t^u, R_t^u)$. We approach the probability by learning a function f_Θ parameterized by Θ:

$$\hat{y}_t = f_\Theta(\cdot) \tag{2}$$

Here, $\hat{y}_t = Pr(y_t = 1|q_t, C_t, \mathcal{X}_{t-1}^u, \mathcal{S}_t^u, R_t^u; \Theta)$, and $(\cdot)$ denotes the features we use to predict.

Table 1: Notations and descriptions.

Notations	Descriptions
$\mathcal{X}_{i-1}^u$	Student u's question-answering history.
q_i, c_j	The question and the concept.
C_i	The set of concepts that are related to q_i.
$\hat{y}_t, y_t$	The predicted probability and the true label.
$\mathbf{h}_t$	The knowledge state.
$\mathbf{h}_t^v$	The context of record.
r_t^u, R_t^u	The question-answering record and similar peer records.
$\mathcal{S}_t^u$	The similar peer sub-sequence.

4 METHOD

Unlike the previous methods, which only explicitly considers the intra-student information in knowledge tracing, our CoKT explicitly utilizes both intra-student and inter-student information in knowledge tracing. The intra-student information is extracted from the students' historical question-answering records, and the inter-student information is extracted from the similar peer records and similar peer sub-sequences (in Definition 3.4). Our framework is illustrated in Figure 2. We integrate the intra-student and inter-student information in the integration module, and we feed the integration of intra-student and inter-student information to the prediction module to estimate the probability of students correctly answering questions, and we feed the integration to state update module to update students' knowledge states. In the following sections, we will first present the integration of intra-student and inter-student information (the integration module in Figure 2(a)). Then, we will discuss the prediction module and state update module in Figure 2(a).

4.1 The integration module

The integration module integrates the intra-student and inter-student information, as it is shown in Figure 2(b). Since the inter-student information is extracted from the similar peer records and

similar peer sub-sequences, we will first discuss the obtaining of similar peer records and similar peer sub-sequences. Subsequently, we will present the inter-student information representation and intra-student information representation. Finally, we will discuss the integration of two types of information.

4.1.1 Obtaining similar peer records and sub-sequences. Inspired by the works of Qin et al. [26, 27], we devise a retrieval-based mechanism to obtain the similar peer records and sub-sequences. First, for an arbitrary record $r_i^{\bar{u}} = \{q_i, C_i, r_i\}$ in dataset, we store the followings to the database of a search engine: (1) record ID: the ID of $r_i^{\bar{u}}$; (2) student ID: the ID of $\bar{u}$; (3) question ID: the ID of q_i; (4) concept string: the string concatenation of the concepts in C_i; (5) historical string: the string concatenation of the questions and concepts in the question-answering history (Definition 3.3) of $r_i^{\bar{u}}$.

Then, for an arbitrary record r_t^u, we obtain its similar peer records by retrieving the records which satisfy the three conditions: (1) have same question ID or same concept string with r_t^u; (2) have different student ID with r_t^u; (3) have similar historical string. We evaluate the similarity score of historical strings by BM25 [29]:

$$score(s_t^u, s_j^{\bar{u}}) = \sum_{i=1}^{n} IDF(k_i) \cdot \frac{tf(k_i, s_j^{\bar{u}}) \cdot (b_1+1)}{tf(k_i, s_j^{\bar{u}}) + b_1 \cdot (1 - b_2 + b_2 \cdot \frac{|s_j^{\bar{u}}|}{L})}, \quad (3)$$

where s_t^u denotes the historical string of r_t^u. $s_j^{\bar{u}}$ denotes the historical string of other students' records, whose length is represented as $|s_j^{\bar{u}}|$. k_i denotes the keywords like the question ID, concept ID in s_t^u. $tf(k_i, s_j^{\bar{u}})$ denotes the k_i's term frequency in $s_j^{\bar{u}}$. L is the average length of the historical strings for all the records. b_1 and b_2 are free parameters, and we use $b_1 = 1.2$, $b_2 = 0.75$ in our case. The $IDF(k_i)$ is the inverse document frequency weight of k_i:

$$IDF(k_i) = \ln\left(\frac{N - n(k_i) + 0.5}{n(k_i) + 0.5} + 1\right), \quad (4)$$

in which N is the total number of records, and $n(k_i)$ is the number of the records which contains k_i.

We rank the records according to the similarity score, and take the top $|R|$ records as the similar peer records (R_t^u) of r_t^u. For each record $r_i^{\bar{u}} \in R_t^u$, we take out its question-answering history $\mathcal{X}_{i-1}^{\bar{u}} = \{r_1^{\bar{u}}, r_2^{\bar{u}}, ..., r_{i-1}^{\bar{u}}\}$ to obtain the similar peer sub-sequence.

4.1.2 Inter-student information representation. The inter-student information aims to collect the peer students' correctness of responses to the similar questions under the context of similar peer records, and we apply attention to obtain it. Thus, to obtain the inter-student information, these representation are necessary: (1) the context (Definition 3.3) of similar peer records; (2) the peer students' correctness to similar questions under the context of similar records; (3) the importance of peer students to student u.

For each records $r_i^{\bar{u}} \in R_t^u$, we take following steps to obtain the its context:

First, we initialize the context of $r_1^{\bar{u}}$ with $\mathbf{0} \in \mathbf{R}^{d_h}$. That is $\mathbf{h}_1^v = \mathbf{0}$. For an arbitrary record $r_j^{\bar{u}} \in \mathcal{X}_{i-1}^{\bar{u}}$ (in Definition 3.3, $\mathcal{X}_{i-1}^{\bar{u}}$ is the question-answering history of $r_i^{\bar{u}}$), $r_j^{\bar{u}} = \{q_j, C_j, y_j\}$, we represent the question q_j with the its embedding and the embedding of the concepts in C_j:

$$\mathbf{p}_j = \left[\mathbf{e}_j^q : \mathbf{e}_m\right], \quad (5)$$

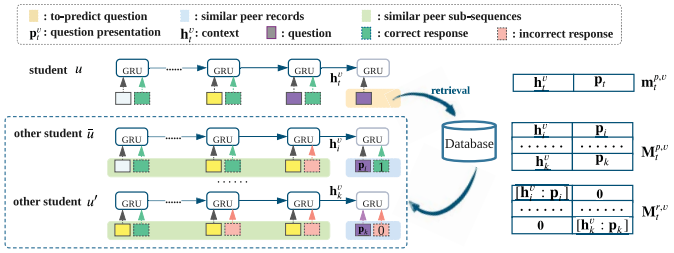


Figure 3: The obtaining of the inter-student information.

where $\mathbf{e}_j^q \in \mathbf{R}^{d_q}$ denotes the embedding of question q_j, $\mathbf{e}_m \in \mathbf{R}^{d_c}$ denotes the mean embedding of the concepts in C_j. Both the question embedding and concept embedding are randomly initialized and trained with the model. [:] denotes the vector concatenation. So we have $\mathbf{p}_j \in \mathbf{R}^{d_p}, d_p = d_c + d_q$.

Then, we define the concatenation of question representation and the correctness of students' responses as:

$$g_c(\mathbf{p}_j, y_i) = \begin{cases} \left[\mathbf{p}_j : \mathbf{0}\right], & \text{if } y_j = 1, \\ \left[\mathbf{0} : \mathbf{p}_j\right], & \text{otherwise,} \end{cases} \qquad \mathbf{z}_j^v = g_c(\mathbf{p}_j, y_i) \quad (6)$$

where $\mathbf{0}$ has the same dimension with $\mathbf{p}_j$. Thus $\mathbf{z}_j^v \in \mathbf{R}^{2d_p}$.

Subsequently, we feed the concatenation to RNN to obtain the context representation of record $r_{j+1}^{\bar{u}}$,

$$\mathbf{h}_{j+1}^v = GRU(\mathbf{z}_j^v, \mathbf{h}_j^v), \quad (7)$$

where *GRU* denotes the Gated Recurrent Unit [5]. Using Eq. 7 in $\mathcal{X}_{i-1}^{\bar{u}}$ repeatedly, we can obtain the context of $r_i^{\bar{u}}$ ($r_i^{\bar{u}} = \{q_i, C_i, y_i\}$), which is represented as $\mathbf{h}_i^v$, as it is shown in Figure 3.

To obtain the representation of peer students' correctness to similar questions under the context of similar records, for $r_i^{\bar{u}} \in R_t^u$, we concatenate its context, question representation and correctness of response into context-question-correctness concatenation:

$$\mathbf{m}_i^{r,v} = g_c([\mathbf{h}_i^v : \mathbf{p}_i], y_i), \quad (8)$$

where g_c is a function defined in Eq. 6, and $\mathbf{m}_i^{r,v} \in \mathbf{R}^{2d_m}, d_m = d_h + d_c + d_q$. We integrate the context-question-correctness concatenations of similar peer records into a matrix $\mathbf{M}_t^{r,v} \in \mathbf{R}^{|R| \times 2d_m}$, in which each row represents the context-question-correctness concatenation of one similar peer record (shown Figure 3).

To obtain the importance of peer students to the student u, we compute the relevance of the context and questions they answer. Thus, for an arbitrary $r_i^{\bar{u}} \in R_t^u$, we concatenate its context with question representation into context-question concatenation:

$$\mathbf{m}_i^{p,v} = \left[\mathbf{h}_i^v : \mathbf{p}_i\right], \quad (9)$$

where $\mathbf{p}_i$ is the question representation of q_i as it is defined in Eq. 5 and $\mathbf{m}_i^{p,v} \in \mathbf{R}^{d_m}$. We integrate the context-question concatenations of similar peer records into a matrix $\mathbf{M}_t^{p,v} \in \mathbf{R}^{|R| \times d_m}$ as it is illustrated in Figure 3. We also concatenate the context of the record which we aim to predict with its question representation:

$$\mathbf{m}_t^{p,v} = \left[\mathbf{h}_t^v : \mathbf{p}_t\right], \quad (10)$$

where $\mathbf{m}_t^{p,v} \in \mathbf{R}^{d_m}$.

Finally, we apply the attention mechanism over $\mathbf{M}_t^{r,v}$ [35] as the left-side of Figure 2(b) illustrates to obtain the inter-student

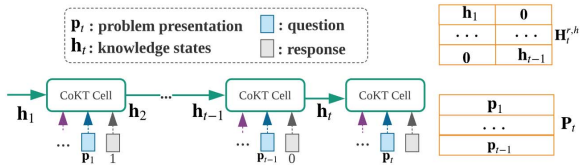


Figure 4: The obtaining of the intra-student information.

information representation. Specifically, we define

$$\mathbf{Q}_v = \mathbf{m}_t^{p,v}, \mathbf{K}_v = \mathbf{M}_t^{p,v}, \mathbf{V}_v = \mathbf{M}_t^{r,v}. \tag{11}$$

The attention is

$$g_a(\mathbf{Q}_v, \mathbf{K}_v, \mathbf{V}_v) = softmax(\frac{\mathbf{Q}_v \mathbf{K}_v^{\mathrm{T}}}{\sqrt{(d)}})\mathbf{V}_v, \tag{12}$$

where d is the dimension of $\mathbf{Q}_v$. (Here d is a variable which denotes the dimension of query in attention.).

The inter-student information is represented by the multi-head attention:

$$\mathbf{v}_t^v = f_v(\left[head_1^v, head_2^v, ..., head_{n_v}^v\right]^{\mathrm{T}} \mathbf{W}_h^v), \tag{13}$$

where $\mathbf{v}_t^v \in \mathbf{R}^{2d_m}$ is the inter-student information, $\mathbf{W}_h^v \in \mathbf{R}^{n_v \times 1}$ is the weight of attention heads, n_v is the number of heads and f_v is multiple layer perceptron (MLP). $head_{n_v}^v$ is an attention head,

$$head_{n_v}^v = g_a(f_q^v(\mathbf{Q}_v), f_k^v(\mathbf{K}_v), f_v^v(\mathbf{V}_v)), \tag{14}$$

where f_q^v, f_k^v and f_v^v are the MLPs and g_a is defined in Eq. 12.

4.1.3 Intra-student information representation. The intra-student information aims to collect the student's historical correctness to questions under the historical knowledge states. Since the historical questions has different importance to the to-predict question, we also apply attention mechanism to represent the intra-student information as the right-side of Figure 2(b) illustrates. First, for an arbitrary record $r_i^u (1 \le i < t)$, we concatenate the corresponding knowledge state with its correctness as Figure 4 illustrates (the obtaining of knowledge state will be discussed in 4.3):

$$\mathbf{h}_i^{r,h} = g_c(\mathbf{h}_i, y_i), \tag{15}$$

where $\mathbf{h}_i^{r,h} \in \mathbf{R}^{2d_h}$, g_c is a defined in Eq. 6. Then we define:

$$\mathbf{Q}_h = \mathbf{p}_t,\ \mathbf{K}_h = \left[\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_{t-1}\right]^{\mathrm{T}}, \mathbf{V}_h = \left[\mathbf{h}_1^{r,h}, \mathbf{h}_2^{r,h}, \cdots, \mathbf{h}_{t-1}^{r,h}\right]^{\mathrm{T}}, \tag{16}$$

where $\mathbf{Q}_h \in \mathbf{R}^{d_p}$, $\mathbf{K}_h \in \mathbf{R}^{(t-1)\times d_p}$, and $\mathbf{V}_h \in \mathbf{R}^{(t-1)\times 2d_h}$.

The intra-student information is represented by:

$$\mathbf{v}_t^h = f_h(\left[head_1^h, head_2^h, ..., head_{n_h}^h\right]^{\mathrm{T}} \mathbf{W}_h^h), \tag{17}$$

where $\mathbf{v}_t^h \in \mathbf{R}^{2d_m}$ denotes the intra-student information representation, $\mathbf{W}_h^h \in \mathbf{R}^{n_h \times 1}$ is weight on attention heads, and f_h is MLP. $head_i^h$ is an attention head, which is defined as

$$head_{n_h}^h = g_a(f_q^h(\mathbf{Q}_h), f_k^h(\mathbf{K}_h), f_v^h(\mathbf{V}_h)), \tag{18}$$

where f_q^h, f_k^h and f_v^h are the MLPs and g_a is defined in Eq. 12.

4.1.4 The integrated information representation. We integrate the representation of intra-student information and inter-student information with:

$$\mathbf{v}_t = softmax(\mathbf{w}_r)\left[\mathbf{v}_t^h, \mathbf{v}_t^v\right]^{\mathrm{T}}, \tag{19}$$

to obtain the integrated information, where $\mathbf{v}_t \in \mathbf{R}^{2d_m}$, $\mathbf{w}_r$ is a two-dimensional parameter vector, and $softmax(\mathbf{w}_r) \in \mathbf{R}^{1\times 2}$ indicates the relative importance of intra-student information and inter-student information.

4.2 Prediction module

We predict the probability of student u correctly answering q_t by the integrated information $\mathbf{v}_t$ (Eq. 19), her knowledge state ($\mathbf{h}_t$) and the question representation ($\mathbf{p}_t$), as it is shown in Figure 2(a).

More specifically, we concatenate the integrated information with the student's knowledge state and question representation, and feed the concatenation to a classifier to predict the probability that student u correctly answer q_t:

$$\hat{y}_t = Sigmoid(f_p(\left[\mathbf{v}_t : \mathbf{h}_t : \mathbf{p}_t\right])) \tag{20}$$

where f_p denotes a MLP, $Sigmoid(\cdot)$ denotes the *sigmoid* function.

4.3 State update module

The student acquires some knowledge after she answers questions. Thus, her knowledge state transfers from $\mathbf{h}_t$ to $\mathbf{h}_{t+1}$ after she answered q_t. We consider the question representation, correctness and the integrated information to update the knowledge states, as it is shown in Figure 2 (a) and (c). First, we combine the question representation, the correctness and the integrated information by:

$$\mathbf{z}_t^u = f_u(\left[\mathbf{v}_t : g_c(\mathbf{p}_t, y_t)\right]), \tag{21}$$

where g_c is the function defined in Eq. 6, f_u is MLP, and $\mathbf{z}_t^u \in \mathbf{R}^{2d_p}$. Subsequently, we feed $\mathbf{z}_t^u$ to GRU to update the student's knowledge state:

$$\mathbf{h}_{t+1} = GRU(\mathbf{z}_t^u, \mathbf{h}_t). \tag{22}$$

4.4 Model Learning

The objective function of our model is to minimize the negative log-likelihood of the observed sequences. The learning parameters of our method are the embedding of concepts and questions, $\mathbf{w}_r$ in Eq. 19, the weights in GRU and the parameters of all the MLPs. The parameters are jointly learned by minimizing the cross-entropy between the predicted probability $\hat{y}_t$ and the true label y_t as

$$\mathcal{L} = -\sum_u \sum_{t=1}^{T_u} (y_t\ log\ \hat{y}_t + (1 - y_t)\ log(1 - \hat{y}_t)), \tag{23}$$

where T_u denotes the length of student u's question-answering sequence. Algorithm 1 shows training procedure of our method, and implementation details will be discussed in the next section.

5 EXPERIMENT

In this section, we present our experimental settings and the results in detail [2]. We also make some discussions with extended investigations to illustrate the effectiveness of our model.

5.1 Dataset

We evaluate our method on four public real-world datasets: ASSIST09 , ASSIST12, EdNet and Junyi.

ASSIST09 is gathered from the ASSISTments online tutoring platform [11]. We filter out the records without concept tags. Each question in this dataset is related to one to four concepts.

[2]The source code is available at https://github.com/githubg0/CoKT

Algorithm 1: Training procedure of CoKT

```
Store the transformational records in a search engine;
Pre-processing: for each record, search its similar peer
  records and similar peer sub-sequences ;
Randomly initialize the learning parameters;
while not converged do
    for batch in data do
        for (t = 0; t < seq_length; t = t + 1) do
            Obtain the context of records (Eq. 7);
            Obtain the inter-student information (Eq.13);
            Obtain the intra-student information (Eq.17);
            Predict the correctness of responses (Eq. 20);
            Update the knowledge states (Eq. 22);
        end
        Compute the gradient and update the parameters
          w.r.t the loss ℒ (Eq. 23);
    end
end
```

Table 2: Dataset Statistics. The statistics are the actual sample data we used for the experiment after pre-processing.

Dataset	ASSIST09	ASSIST12	EdNet	Junyi
Students	2,968	22,422	4,700	7,000
Records	185,110	1,839,429	326,037	622,781
Questions	15,003	45,543	11,060	1,978
Concepts	121	99	189	39
Questions Per Concept	150.76	460.03	128.73	50.72
Concepts Per Question	1.22	1.0	2.21	1.0
Attempts Per Question	12.34	40.39	29.48	314.85
Attempts Per Concept	1914.21	18,580.10	4023.72	15,968.74
Positive Label Rate	63.80%	69.60%	59.69 %	67.30%

ASSIST12 is also gathered from the ASSISTments online tutoring platform [11]. Different with ASSIST09, each question corresponds only one concept. We do the same pre-processing as ASSIST09.

EdNet is a large scale datatset contributed by Choi et al. [7]. We randomly sample some students by the same way used in [18].

Junyi is collected from Junyi e-learning website [3]. we also randomly sample some students by the same way as EdNet.

The statistic details of the four datasets after pre-processing are shown in Table 2. The maximum length of students' question-answering history is set to 200. We split 80% data for training and validation, and 20% for testing.

5.2 Baselines

According to the research of [42], the deep learning-based methods have better performance than traditional methods in knowledge tracing task in general. To evaluate the effectiveness of our model, we follow [1, 12] and compare our method with three groups of 10 representative deep leanring-based models. The first group are the *single-state methods*, which make predictions based on the students' knowledge states, each of which is represented by one vector.

- DKT [25] obtains students' knowledge states by feeding students' individual question-answering records to a LSTM [14], and it outputs the prediction based on the knowledge state.
- EERNNA [15] is also an extension of DKT by considering the relevance between the historical questions and the to-predict question.

Table 3: The performance on four public datasets. * indicates p-value < 0.05 in the significance test.

Model	ASSIST09		ASSIST12		EdNet		Junyi	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
CKT	0.6870	0.7067	0.7036	0.6542	0.6285	0.6424	0.8203	0.8803
SAKT	0.6691	0.6779	0.7160	0.6959	0.6655	0.6982	0.7543	0.7963
SAINT	0.6720	0.6803	0.7123	0.6876	0.6632	0.6981	0.8075	0.8536
AKT	0.7069	0.7241	0.7355	0.7377	0.6792	0.7217	0.8189	0.8668
DKVMN	0.6489	0.6463	0.7117	0.6824	0.6579	0.6808	0.8306	0.8846
SKVMN	0.6441	0.6384	0.7083	0.6769	0.6558	0.6816	0.8229	0.8793
GKT	0.7270	0.7547	0.7181	0.6875	0.6644	0.6940	0.8074	0.8587
DKT	0.6633	0.6723	0.7128	0.6901	0.6666	0.7025	0.8337	0.8873
EERNNA	0.7053	0.7258	0.7352	0.7370	0.6671	0.7066	0.8122	0.8685
DHKT	0.7170	0.7390	0.7361	0.7391	0.6636	0.6940	0.8420	0.8961
CoKT	**0.7324***	**0.7682***	**0.7380***	**0.7401**	**0.6887***	**0.7374***	**0.8448**	**0.8980**

- DHKT [40] is another extension of DKT, by considering the hierarchical structure constraint of question embedding and concept embedding.

The second group are *multi-state methods*, which maintain multiple vectors to represent students' knowledge states.

- DKVMN [45] makes prediction based on students' knowledge states, and it obtains the knowledge states by feeding students' question-answering records to a memory network.
- SKVMN [1] improves DKVMN by applying a LSTM on the concept-similar records.
- GKT [22] extends DKVMN by updating the knowledge states according to a graph.

The third group are *attention-based methods*, which trace students' knowledge with attention mechanism.

- CKT [30] adopts the dot product to identify students' historical relevant performance to make prediction.
- SAKT [23] introduces the self-attention [35] to capture the relevance between historical questions and the to-predict question.
- SAINT [6] uses the Transformer [35] to capture the relevance between the historical questions and the to-predict question.
- AKT-NR [12] also applies the Transformer to model students' question-answering sequences. However, it adopts the exponential decay in multi-head attention score.

5.3 Implementation Details

Following [37], we set the dimension of the knowledge states, question embedding and concept embedding to 64 for all the models. In our CoKT, the dimension of context is also 64. The hidden layers of all MLPs except for f_q^h, f_k^h, f_v^h are one. The number of attention heads in Eq. 13 is 4. The number of attention heads in Eq. 17 is 1, and $f_q^h(\mathbf{Q}_h) = \mathbf{Q}_h, f_k^h(\mathbf{K}_h) = \mathbf{K}_h, f_v^h(\mathbf{V}_h) = \mathbf{V}_h$. We select the number of similar peer records (sub-sequences) from {3, 5, 7, 9, 10, 11, 12, 15} for all the datasets. For the baselines which use RNN to update the knowledge states (like DKT, DHKT, EERNNA), we also apply GRU for fair comparison. The other hyperparameters for all the baselines are carefully tuned to their best performances. The optimizer is Adam [16], and we choose learning rate from $\{1 \times 10^{-4}, 3 \times 10^{-4}, 6 \times 10^{-4}, 1 \times 10^{-3}\}$.

5.4 Experiment Result

We measure the ACC, AUC and the statistical significance to evaluate the performance of models. Specifically, AUC is the proportion of true positives (resp. negatives) that are correctly predicted to be

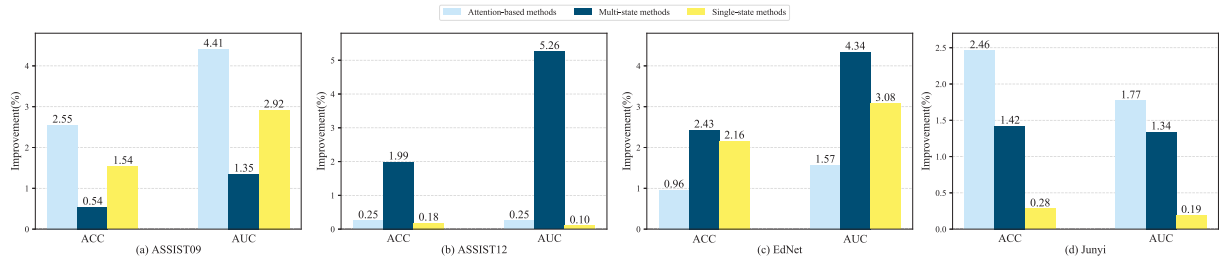


Figure 5: The improvement on different types of methods.

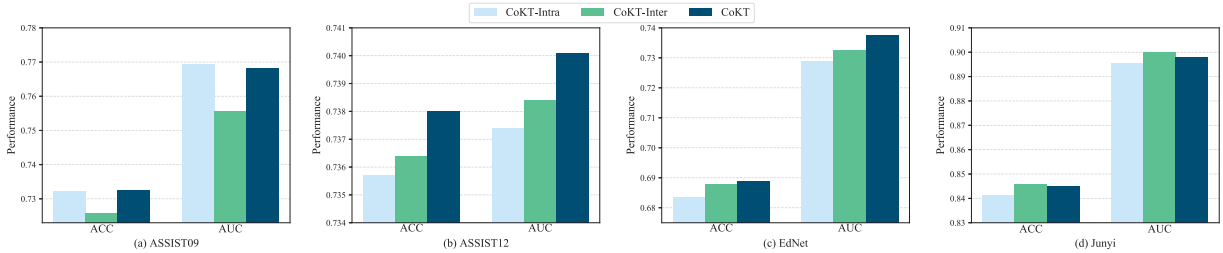


Figure 6: The contribution of different modules for different datasets.

positive (resp. negative). ACC is the proportion of predictions that are correct. A higher AUC (ACC) indicates a better performance. For statistical significance, following [28], we deploy a MannWhitney U test [19] under AUC metric, and a t-test [2] under ACC metric. Table 3 shows the converged ACC, AUC and the statistical significance of our model against the baseline models. Figure 5 illustrates our improvement on different groups of methods.

From table 3, we can observe that our method outperforms all the baselines and achieves the best results on all the datasets. Compared with previous methods, our method has 0.18% - 0.96% improvement in ACC, and 0.10% - 1.57% improvement in AUC. From Figure 5, we can observe that: (1) Our method is better than attention-based methods by 0.25%-2.55% in ACC and 0.25% - 4.41% in AUC, better than multi-state methods by 0.54% - 2.43% in ACC and 1.34% - 5.26% in AUC, better than single-state methods by 0.18%-2.16% in ACC and 0.10% -3.08% in AUC. Since the difference between our method and other deep-learning based methods is the inter-student information, the results demonstrate that the inter-student information benefits model performance; (2) There is no clear performance boundary among baseline models, different groups of methods are good at different datasets. For instance, the best multi-state method has closer performance to our CoKT than the best attention-based method in ASSIST09. However, it has opposite results in EdNet;

5.5 Ablation Study

To further investigate the contributions of different information in our method, we conduct some ablation studies. Specifically, we have two different variants:

- **CoKT-Intra** removes the intra-student information. That means, the $\mathbf{v}_t^h$ is removed in Eq. 19, and there is no $\mathbf{w}_r$ accordingly.
- **CoKT-Inter** removes the inter-student information. That means, the $\mathbf{v}_t^v$ is removed in Eq. 19, and there is no $\mathbf{w}_r$ accordingly.

Figure 6 illustrates the results. From Figure 6, we can identify that: (1) Both intra-student and inter-student information are beneficial for the model's performance. We can observe that removing the intra-student information or inter-student information from the integration module will decrease the performance in most cases.

Table 4: The performance of introducing inter-student information (Model+ Inter) or both intra-student and inter-student information (Model + Both) to other single-state baselines.

Model	ASSIST09		ASSIST12		EdNet		Junyi	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
DKT	0.6633	0.6723	0.7128	0.6901	0.6666	0.7025	0.8337	0.8873
DKT+Inter	0.7005	0.7204	0.7288	0.7224	0.6743	0.7142	0.8352	0.8887
Inter_Imprv	3.72%	4.81%	1.60%	3.23%	0.77%	1.17%	0.15%	0.14%
DKT+Both	0.7233	0.7564	0.7324	0.7273	0.6868	0.7334	0.8414	0.895
Both_Imprv	6.00%	8.41%	1.96%	3.72%	2.02%	3.09%	0.77%	0.77%
DHKT	0.717	0.739	0.7361	0.7391	0.6636	0.694	0.842	0.8961
DHKT+Inter	0.7301	0.7653	0.7347	0.7342	0.6836	0.7291	0.8364	0.8892
Inter_Imprv	1.31%	2.63%	-0.14%	-0.49%	2.00%	3.51%	-0.56%	-0.69%
DHKT+Both	0.7307	0.7653	0.7373	0.7374	0.6895	0.7383	0.8421	0.8955
Both_Imprv	1.37%	2.63%	0.12%	-0.17%	2.59%	4.43%	0.01%	-0.06%
EERNNA	0.7053	0.7258	0.7352	0.737	0.6671	0.7066	0.8122	0.8685
EERNNA+Inter	0.7328	0.7683	0.7378	0.7401	0.6845	0.7282	0.8456	0.8983
Inter_Imprv	2.75%	4.25%	0.26%	0.31%	1.74%	2.16%	3.34%	2.98%

That implies both of them offer help for prediction. (2) Different information have different importance to different datasets. In Junyi, removing inter-student information (*CoKT-Inter*) has no negative impact on the model performance, but removing the intra-student information (*CoKT-Intra*) decreases the performance more significantly, which is contrary in ASSIST09. Moreover, removing either of them will decrease model's performance in ASSIST12 and EdNet. That means different datasets have different preferences on intra-student and inter-student information.

5.6 Compatibility Analysis

Our method *CoKT* is a single-state method, which explicitly uses both intra-student information and inter-student information to improve the performance. To investigate whether our mechanism benefits the performance of other single-state baselines, we also apply our intra-student and inter-student information representation to other single-state baselines. Specifically, we introduce our consideration of only inter-student information, and both inter-student and intra-student information to *DKT* and *DHKT*. We only introduce our consideration of inter-student information to *EERNNA*, since it has explicitly considered the intra-student information

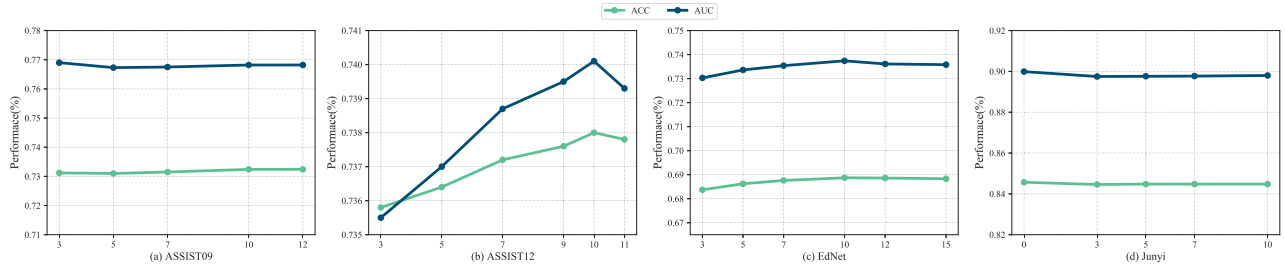


Figure 7: The impact of peer similar records number.

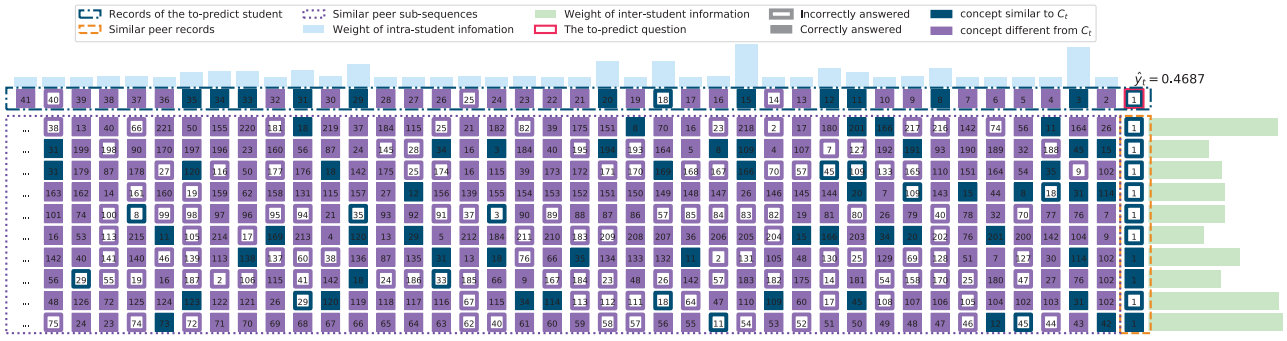


Figure 8: The impact of inter-student information on the correctness prediction

by cosine similarity. The results is presented in Table 4. We can observe that our representation of intra-student information and inter-student information benefits the performance of other sing-state baselines in most cases. Introducing our intra-student and inter-student information brings 2.85% AUC improvement in average, and introducing our inter-student information brings 2.00% AUC improvement in average.

5.7 Hyper-parameter Study

To investigate the sensitivity of our model, we evaluate the impact of the different number of similar peer records (sub-sequences) on the performance. The other hyperparameters remain unchanged when we test the number of similar peer records (sub-sequences) in the experiments. We illustrate the results in Figure 7. From Figure 7, we can observe that the performance of our model changes smoothly around the peak region. That means our method has good stability. Therefore, the optimal hyperparameters of our method are easily obtained.

5.8 Case Study

To further investigate the impact of inter-student information on the correctness prediction, we visualize the prediction of a student's t-th ($t = 40$) question-answering, as it is shown Figure 8. Each grid in Figure 8 denotes a record. The first line of the grids represents the student's t records, and the 2nd - 11th lines represent the records of peer students identified by our method, *i.e.*, the similar peer sub-sequences and similar peer records. Each grid is annotated by the question ID (number), concepts (color: similar to the to-predicted are dark green, otherwise purple) and the ground truth correctness (solid or hollow). From intra-student information presented in Figure 8, we can observe: (1) the to-predict student has never answered the question with ID 1 previously; (2) the to-predicted student has $35/39 = 89.74\%$ accuracy in previous $t-1$ steps; (3) the to-predicted students has $11/12 = 91.67\%$ accuracy in the historical questions which has similar concepts with the to-predict question. The evidence in intra-student information indicates that the student has a high probability to correctly answer q_t. However, the student gives the wrong answer as it is presented in Figure 8, q_t is annotated by hollow grid. On the other hand, from the inter-student information presented in Figure 8, we can observe that $7/10 = 70\%$ peer students incorrectly answered the question with ID 1. It implies that the student has a high probability of incorrectly answering q_t, which is consistent with the ground truth. By integrating the inter-student information with the intra-student information, our CoKT predicts that the student has only 46.87% probability of correctly answering the question. That means, the inter-student information is beneficial for the prediction, especially when intra-student information is less informative.

6 CONCLUSION

In this paper, we proposed a model called CoKT to trace students' knowledge states. Different from the previous methods, which consider only intra-student information, our model explicitly introduces the inter-student information in knowledge tracing. We obtain the inter-student information by retrieving the records and sub-sequences of the students who have similar question-answering experiences. CoKT makes predictions based on the integration of the intra-student and inter-student information. We validate the performance of our model on four public datasets, and compare it with 10 outstanding methods. The experiment results demonstrate that our method achieves the state-of-art performance. Since our inter-student information is based on the context representation, future works could further explore the context representation to improve the performance.

ACKNOWLEDGEMENT

We would like to appreciate the kind help offered by the colleagues of Huawei Noah's Ark Lab. The corresponding author Yong Yu thanks the support of NSFC (62177033).

REFERENCES

[1] Ghodai Abdelrahman and Qing Wang. 2019. Knowledge Tracing with Sequential Key-Value Memory Networks. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 175–184.

[2] Bhaskar Bhattacharya and Desale Habtzghi. 2002. Median of the p value under the alternative hypothesis. *The American Statistician* 56, 3 (2002), 202–206.

[3] Haw-Shiuan Chang, Hwai-Jung Hsu, and Kuan-Ta Chen. 2015. Modeling Exercise Relationships in E-Learning: A Unified Approach.. In *EDM*. 532–535.

[4] Penghe Chen, Yu Lu, Vincent W Zheng, and Yang Pian. 2018. Prerequisite-driven deep knowledge tracing. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 39–48.

[5] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[6] Youngduck Choi, Youngnam Lee, Junghyun Cho, Jineon Baek, Byungsoo Kim, Yeongmin Cha, Dongmin Shin, Chan Bae, and Jaewe Heo. 2020. Towards an appropriate query, key, and value computation for knowledge tracing. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*. 341–344.

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

[8] Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction* 4, 4 (1994), 253–278.

[9] Paul De Boeck and Mark Wilson. 2004. *Explanatory item response models: A generalized linear and nonlinear approach*. Springer Science & Business Media.

[10] Pragya Dwivedi, Vibhor Kant, and Kamal K Bharadwaj. 2018. Learning path recommendation based on modified variable length genetic algorithm. *Education and Information Technologies* 23, 2 (2018), 819–836.

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

[12] Aritra Ghosh, Neil Heffernan, and Andrew S Lan. 2020. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2330–2339.

[13] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature* 538, 7626 (2016), 471–476.

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

[15] Zhenya Huang, Yu Yin, Enhong Chen, Hui Xiong, Yu Su, Guoping Hu, et al. 2019. Ekt: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering* (2019).

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

[17] Qi Liu, Shiwei Tong, Chuanren Liu, Hongke Zhao, Enhong Chen, Haiping Ma, and Shijin Wang. 2019. Exploiting Cognitive Structure for Adaptive Learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 627–635.

[18] Yunfei Liu, Yang Yang, Chen Xianyu, Shen Jian, Zhang Haifeng, and Yu Yong. 2020. Improving Knowledge Tracing via Pre-training Question Embeddings. In *Twenty-Ninth International Joint Conference on Artificial Intelligence*.

[19] Simon J Mason and Nicholas E Graham. 2002. Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical significance and interpretation. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography* 128, 584 (2002), 2145–2166.

[20] Dina Fitria Murad and Bambang Dwi Wijanarko. [n.d.]. Question Recommendation for Online Learning Recommendation Systems based on Rule-path in Knowledge Domain. ([n. d.]).

[21] Koki Nagatani, Qian Zhang, Masahiro Sato, Yan-Ying Chen, Francine Chen, and Tomoko Ohkuma. 2019. Augmenting knowledge tracing by considering forgetting behavior. In *The World Wide Web Conference*. 3101–3107.

[22] Hiromi Nakagawa, Yusuke Iwasawa, and Yutaka Matsuo. 2019. Graph-based Knowledge Tracing: Modeling Student Proficiency Using Graph Neural Network. In *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE, 156–163.

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

[24] Shalini Pandey and Jaideep Srivastava. 2020. Rkt: Relation-aware self-attention for knowledge tracing. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1205–1214.

[25] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *Advances in neural information processing systems*. 505–513.

[26] Jiarui Qin, Weinan Zhang, Rong Su, Zhirong Liu, Weiwen Liu, Ruiming Tang, Xiuqiang He, and Yong Yu. 2021. Retrieval & Interaction Machine for Tabular Data Prediction. In *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[27] Jiarui Qin, Weinan Zhang, Xin Wu, Jiarui Jin, Yuchen Fang, and Yong Yu. 2020. User Behavior Retrieval for Click-Through Rate Prediction. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2347–2356.

[28] Kan Ren, Jiarui Qin, Yuchen Fang, Weinan Zhang, Lei Zheng, Weijie Bian, Guorui Zhou, Jian Xu, Yong Yu, Xiaoqiang Zhu, et al. 2019. Lifelong sequential modeling with personalized memorization for user response prediction. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 565–574.

[29] Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

[30] Shuanghong Shen, Qi Liu, Enhong Chen, Han Wu, Zhenya Huang, Weihao Zhao, Yu Su, Haiping Ma, and Shijin Wang. 2020. Convolutional Knowledge Tracing: Modeling Individualization in Student Learning Process. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1857–1860.

[31] Dongmin Shin, Yugeun Shim, Hangyeol Yu, Seewoo Lee, Byungsoo Kim, and Youngduck Choi. 2021. Saint+: Integrating temporal features for ednet correctness prediction. In *LAK21: 11th International Learning Analytics and Knowledge Conference*. 490–496.

[32] Yu Su, Qingwen Liu, Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Chris Ding, Si Wei, and Guoping Hu. 2018. Exercise-enhanced sequential modeling for student performance prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[33] Hanshuang Tong, Yun Zhou, and Zhen Wang. 2020. HGKT: Introducing Problem Schema with Hierarchical Exercise Graph for Knowledge Tracing. *arXiv preprint arXiv:2006.16915* (2020).

[34] Shiwei Tong, Qi Liu, Wei Huang, Zhenya Huang, Enhong Chen, Chuanren Liu, Haiping Ma, and Shijin Wang. 2020. Structure-based Knowledge Tracing: An Influence Propagation View. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 541–550.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[36] Jill-Jênn Vie and Hisashi Kashima. 2019. Knowledge tracing machines: Factorization machines for knowledge tracing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 750–757.

[37] Chenyang Wang, Weizhi Ma, Min Zhang, Chuancheng Lv, Fengyuan Wan, Huijie Lin, Taoran Tang, Yiqun Liu, and Shaoping Ma. 2021. Temporal Cross-Effects in Knowledge Tracing. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 517–525.

[38] Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yuying Chen, Yu Yin, Zai Huang, and Shijin Wang. 2020. Neural cognitive diagnosis for intelligent education systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 6153–6161.

[39] Shuhan Wang, Hao Wu, Ji Hun Kim, and Erik Andersen. 2019. Adaptive learning material recommendation in online language education. In *International Conference on Artificial Intelligence in Education*. Springer, 298–302.

[40] Tianqi Wang, Fenglong Ma, and Jing Gao. 2019. Deep hierarchical knowledge tracing. In *Proceedings of the 12th International Conference on Educational Data Mining*.

[41] Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916* (2014).

[42] Kevin H Wilson, Yan Karklin, Bojian Han, and Chaitanya Ekanadham. 2016. Back to the basics: Bayesian extensions of IRT outperform neural networks for proficiency estimation. *arXiv preprint arXiv:1604.02336* (2016).

[43] Kevin H Wilson, Xiaolu Xiong, Mohammad Khajah, Robert V Lindsey, Siyuan Zhao, Yan Karklin, Eric G Van Inwegen, Bojian Han, Chaitanya Ekanadham, Joseph E Beck, et al. 2016. Estimating student proficiency: Deep learning is not the panacea. In *In Neural Information Processing Systems, Workshop on Machine Learning for Education*. 3.

[44] Michael V Yudelson, Kenneth R Koedinger, and Geoffrey J Gordon. 2013. Individualized bayesian knowledge tracing models. In *International conference on artificial intelligence in education*. Springer, 171–180.

[45] Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. 2017. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web*. 765–774.