scientific reports

Check for updates

OPEN

Embedding cognitive framework with self-attention for interpretable knowledge tracing

Yanjun Pu[1]✉, Wenjun Wu[2], Tianhao Peng[2], Fang Liu[1], Yu Liang[1], Xin Yu[1], Ruibo Chen[1] & Pu Feng[1]

Recently, deep neural network-based cognitive models such as deep knowledge tracing have been introduced into the field of learning analytics and educational data mining. Despite an accurate predictive performance of such models, it is challenging to interpret their behaviors and obtain an intuitive insight into latent student learning status. To address these challenges, this paper proposes a new learner modeling framework named the EAKT, which embeds a structured cognitive model into a transformer. In this way, the EAKT not only can achieve an excellent prediction result of learning outcome but also can depict students' knowledge state on a multi-dimensional *knowledge component*(KC) level. By performing the fine-grained analysis of the student learning process, the proposed framework provides better explanatory learner models for designing and implementing intelligent tutoring systems. The proposed EAKT is verified by experiments. The performance experiments show that the EAKT can better predict the future performance of student learning(more than 2.6% higher than the baseline method on two of three real-world datasets). The interpretability experiments demonstrate that the student knowledge state obtained by EAKT is closer to ground truth than other models, which means EAKT can more accurately trace changes in the students' knowledge state.

Over the past decade, the *Intelligent tutoring system* (ITS) has become increasingly important in online education because it can offer a personalized and adaptive learning experience for a large scale of students. The core component of the ITS is a cognitive learner model, which can infer the latent knowledge state of individual students so that other components can provide personalized guidance for improving their learning efficiency[1–3]. In recent years, many knowledge tracing models have been developed, and they can be roughly divided into structured knowledge tracing models and deep knowledge tracing models[4]. A *Knowledge Tracing*(KT) task can be regarded as a supervised sequence learning problem. For instance, for a given sequence of a student's historical exercise interactions $X_t = (x_1, x_2, \ldots\ldots, x_t)$, the KT model can predict the probability of answering correctly in the next interaction $p(r_{t+1} = 1|q_{t+1}, X)$ and infer a student's knowledge state in each interaction. Input x_t is usually represented as a tuple (q_t, a_t), where q_t represents the question that a student encounters in timestamp t, and a_t indicates whether the answer of q_t is correct or not. Structured knowledge tracing methods, such as *Bayesian knowledge tracing* (BKT)[5], define their parameters and variables based on the principles of cognitive and education science. Therefore, it is simple to interpret their predictive results to instructors when they are used to assess a student's learning performance. Traditional structured models have certain limitations in modeling multi-dimensional knowledge states because they typically assume that every item q_t is designed around a single knowledge concept. However, an item q_t involves multiple skill requirements, which can be formulated as a Q-matrix[6], which is a sparse matrix for measurement of cognitive mastery. As an extension of BKT, an *Automatic Temporal Cognitive* (ATC) method integrates the Q-matrix into a nonlinear state-space model to trace multi-dimensional student knowledge states accurately[7].

Recently, deep neural network-based cognitive learner models have been proposed to solve the KT tasks, such as *Deep Knowledge Tracing* (DKT)[8], *Dynamic Key-Value Memory Network* (DKVMN)[9], *Exercise-aware Knowledge Tracing* (EKT)[10], and Deep-IRT[11]. These models adopt different DNN frameworks to realize knowledge tracing, and compared to structured models, they can achieve better prediction performance. However, deep neural networks are often deemed as black-box models whose complex inner representations are difficult to associate with an explicit description of latent skill states and their relations in the Q-matrix form. The DKT dismisses information about concepts and abstracts the students' ability as a hidden vector, resulting in the invisibility of

[1]School of Computer Science and Engineering, Beihang University, Beijing 100191, China. [2]Institute of Artificial Intelligence, Beihang University, Beijing 100191, China. ✉email: buaapyj@buaa.edu.cn

students' cognitive structure. In addition, a number of deep learning-based learner models, such as DKVMN and EKT, assume that each question is related only to one skill without considering the difficulty coefficient of questions. Researchers have attempted to apply the IRT model to the KT tasks and combined it with the DKVMN to develop a deep-IRT model, assigning neural network parameters with psychological significance. However, a simple IRT model cannot accurately describe the knowledge requirements of exercises in complex multi-skill learning scenarios[12]. Although the Deep-IRT adds difficulty attributes to every *Knowledge Component*(KC), it still follows the assumption of a single KC for each question.

Although structured knowledge tracing models have explainable parameters based on cognitive science theories, they show certain difficulties in handling complex model structures and datasets. In contrast, deep knowledge-based tracing models can achieve good predictive performance, but their encoding of the cognitive state is very hard to interpret in the context of intelligent tutoring. To address this challenge, this study proposes the EAKT model by incorporating the cognitive structure of a structured ATC model into a DNN-based knowledge tracing framework. In this way, the predictive power of the DNN-based knowledge tracing models is combined with the strength of structured models to generate an interpretable knowledge state.

The major contributions of this paper can be summarized as follows:

1. A structured cognitive model is used to constrain a DNN-based knowledge tracing framework so that the model parameters can be assigned with explainable meanings while guaranteeing the predictive performance;
2. A Q-matrix is introduced to describe the fine-grained relationship between knowledge components and every question. The EAKT represents the multi-dimensional KC vector as a student's knowledge state and the Q-matrix as a skill requirement for every question.

The rest of this paper is organized as follows. The related work on the Q-matrix discovery method and KT task, including Bayesian KT models and deep learning-based KT models are discussed in "Related work" section. Section "Proposed model" describes the structure of the EAKT model in detail. In "Implementation and experimental results" section, we presents the implementation details,experimental results and compares the EAKT's performance with that of the state-of-art deep knowledge tracing models. Last section concludes this work and presents future work directions.

Related work

Structured knowledge tracing model. The BKT model tracks students' knowledge states over time using the *Hidden Markov Model* (HMM). However, it can track only students' mastery of a single cognitive skill without specifying the difficulty of learning items. Recent research efforts on the BKT have focused on the multiple-subskill extension of the BKT. Brenes[13] proposed Dynamic Cognitive Tracing to construct a cognitive model and a student model of longitudinal student data. In his later work, he introduced the *Feature-Aware Student Knowledge Tracing* (FAST)[14] to different incorporate skill features such as subskills and used problem's difficulty and student ability as parameters of the KT model. All these feature-based extensions of the BKT strongly rely on experts' knowledge when predefining the skill and subskill features without using any automatic Q-matrix discovery method. The *Automatic Temporal Cognitive Model* (ATC) represents an evolution of the *Cognitive Diagnosis Model*(CDM) and the KT Model. It aims to incorporate the multi-dimensional knowledge state and temporal changes, including skill enhancement and forgetting factors. In the ATC model, a nonlinear state-space framework is used to encode multi-dimensional KC levels and Q-matrix of learning items. The ATC model is also capable of deriving the detailed values of the Q-matrix from student learning trajectories in a data-driven approach. Therefore, the ATC model could be an ideal candidate for governing deep neural networks for knowledge tracing and improving their interpretability.

DNN-based knowledge tracing models. *DKT and its extensions.* *Deep knowledge tracing* (DKT) uses *Recurrent Neural Networks* (RNNs) to model student learning and achieves impressive predictive advantages without the need for human-engineered features, such as recency effect and contextualized trial sequence. However, latent encoding of the knowledge state in the DKT cannot consistently depict students' mastery of KCs and predict temporal changes in knowledge state across time. Aiming to the DKT's major problems in KC modeling, the DKT+ introduces regularization terms, which correspond to the reconstruction and waviness, to the loss function of the original DKT model to enhance the consistency in prediction. Experiments have shown that the regularized loss function can effectively alleviate the two problems without degrading the original task of DKT[15]. Chen[16] aimed to address the data sparse problem by incorporating the prerequisite concept pairs as constraints in the DKT model, thus improving the prediction performance of students' concept mastery and offering a partial interpretation of predictive results. However, despite these advantages, hidden state variables of a neural network cannot explicitly represent explainable educational meanings without inducing prior cognitive structure and constraints. As a result, how to characterize changes in the students' knowledge state accurately using a deep neural network has still been a challenge.

Deep knowledge tracing with attention mechanism. Attention mechanism[17] has been shown to be effective in tasks involving sequence modeling. The idea behind this mechanism is to focus on relevant elements of the input signals when predicting the output. The *self-attentive knowledge tracing* (SAKT)[18] has been the first method to adopt attention mechanisms in the context of KT. Attention mechanisms are more flexible than recurrent and memory-based neural networks. Extensive experiments on a variety of real-world datasets suggest that the SAKT model can outperform the state-of-the-art methods and is one order of magnitude faster than the RNN-based

	Multi-dimension skill in KC level	Temporal knowledge state tracing	Processing power for large datasets	Explainability of Knowledge state
BKT	No	Yes	No	Yes
CDM	Yes	No	No	Yes
ATC	Yes	Yes	No	Yes
DKT/SAKT	No	Yes	Yes	No
DKVMN/Deep-IRT	No	Yes	Yes	Yes
AKT	No	Yes	Yes	Yes
EAKT	Yes	Yes	Yes	Yes

Table 1. Comparison of the BKT, CDM, ATC,DKT/SAKT, DKVMN/Deep-IRT, AKT, EAKT frameworks.

approaches. Ghosh et al.[19] presented a *context-aware attentive knowledge tracing* (AKT) model, incorporating the self-attention mechanism with cognitive and psychometric models. They defined context-aware representations of questions and responses using a monotonic attention mechanism to summarize every learner's historical performance in the right time scale. They used the Rasch model to capture individual differences between questions covering the same concept. However, none of the existing methods have quantitatively analyzed the interpretability of students' knowledge state.

Despite of the recent progress in the research of knowledge tracing, most modeling frameworks haven't presented an interpretable mult-dimension knowledge tracing solution. Table 1 summarizes the status of the past major proposals in the research community.

Proposed model

In this section, the EAKT model, which is developed based on the attentive knowledge tracing model and the skill encoding and prediction methods of the ATC model, is presented. First, the ATC model and cognitive diagnosis models for Q-matrix are briefly introduced, and then the EAKT model is described in detail.

ATC model. The ATC framework can be described as two parts: the first part is the probability of students answering the exercises correctly, in which the students' knowledge state and exercise KC are both represented as multi-dimensional vectors. The second part depicts the dynamic changes of students' knowledge states by Eq. (2). The specific calculation process of Eq. (1) is as follows. First, calculate the projection length of the student's knowledge state $\boldsymbol{\theta}_{st}$ on the exercise KC $\boldsymbol{a}_i$ and make a difference with the norm of exercise KC, then use the logistic function to normalize the difference q_{sit} between 0 and 1 as the probability of student s correctly answering the exercise i.

$$\begin{aligned} q_{sit} &= \frac{\theta_{st} \cdot \mathbf{a}_i}{\|\mathbf{a}_i\|} - \|\mathbf{a}_i\| \\ p_{sit} &= Pr(R_{sit} = 1|\theta_{st}, \mathbf{l}_i, \mathbf{a}_i) = \phi(q_{sit}) \end{aligned} \tag{1}$$

- $\mathbf{a}_i$ represents the required KC vector of an exercise i;
- θ_{st} represents the knowledge state vector of a student s at time t;
- R_{sit} is the response of a student s on a exercise i at time t;
- p_{sit} is the probability of a student s giving a correct response on an exercise i at time t.

An exercise is represented as $\boldsymbol{a}_i = (a_{i1}, a_{i2}, \ldots, a_{ik}, \ldots, a_{iM})$, where a_{ik} represents a latent KC of a Q-matrix Q_{MN}. Traditionally, an element a_{ik} is defined as a binary value determining whether a knowledge component KC_k associates with an exercise a_i or not[20]. In the ATC model, the binary Q-matrix is extended to a new matrix with real numbers to indicate the degree of correlation between exercises and all KCs.

Equation (2) assumes that a student's knowledge state at a time step $(t + 1)$ follows the Gaussian distribution with the mean $\mu_{s(t+1),n}$, which depends on the temporal change in $\theta_{st,n}$ in the previous time step. Such a state transition represents an interplay between knowledge acquisition and exponential forgetting between the two states. Equation (3) defines a nonlinear transformation function to formulate the state transition in the learning process over the exercise-answering sequence.

$$\theta_{s(t+1),n} \sim N\left(\mu_{s(t+1),n}, \sigma^2\right) \tag{2}$$

$$\begin{aligned} \mu_{s(t+1),n} &= \left(\theta_{st,n} + l_{i,n} * \phi\left(q_{sit}\right)\right) * f_{st,n} \\ f_{st,n} &= \exp\left\{-\left[\frac{1}{1+\theta_{st,n}} * r + \beta\right] * \Delta t\right\} \end{aligned} \tag{3}$$

- $\theta_{st,n}$ indicated the ability of the nth skill in the dimension of θ_{st};
- $l_{i,n}$ denotes the value of the nth dimension of a vector $\mathbf{l}_i$;

- r and β are fitting parameters;
- $f_{st,n}$ is the forgetting coefficient of a student s from time t to time $(t+1)$;
- Δ_t is the interval between time t and time $(t+1)$.

Cognitive diagnosis models for Q-matrix. The input to the EAKT model requires a Q-matrix, and there are two ways to discover the Q-matrix. In addition to the ATC model described above, the *non-negative matrix factorization* (NMF) models have been proposed to discover the Q-matrix. These factorization techniques can implicitly encode the "slip" and "guess" factors, which means "learner effect" and the "task effect". It divides a large unit into small sections, which are further divided into small problems, and, finally, into small steps, so that tasks can be described as specific skills required to solve the problem. NMF method approximates a matrix X by the product of two smaller matrices W and H, where $X \approx WH^T$, and $W \in R^{U\times K}$ is a matrix where each row u is a vector containing the K latent factors describing the learner u and $H \in R^{I\times K}$ is a matrix where each row i is a vector containing K factors describing task i. Let w_{uk} and h_{ik} be the elements of W and H, respectively; then, the performance p of a learner u on a task i is predicted by:

$$\hat{p}_{ui} = \sum_{k=1}^{K} w_{uk} h_{ik} = \left(WH^T\right)_{u,i} \tag{4}$$

Although both the ATC model and the NMF method can obtain the Q-matrix for the input of EAKT. ATC model is difficult to apply this model to scenarios with thousands of students and long exercise sequences with hundreds of problems due to using sampling for training. Such scenarios often generate large-scale datasets and complex distribution of KCs in exercises, the training process of the ATC model can be very computationally demanding. In the implementation, we use a lightweight NMF method to generate the Q-matrix, while the ATC model is embedded in the prediction layer only.

Particularly, the Q-matrix generation process includes three main steps. First, a student's response data are pre-processed to obtain the difficulty matrix $\left[E_{i,j}\right]_{M\times N}$ of the student exercises, which is expressed as follows:

$$D_{i,j} = \begin{cases} 1 - \frac{1}{T_{i,j}}, \text{ if } T_{ij} \geq 2, & T_{ij} \text{ represents the number of attempts by student i on exercise j} \\ 0, \text{ else} \end{cases} \tag{5}$$

Then, the matrix D is decomposed using the NMF method in Eq. (6).

$$\begin{aligned} E_{M\times N} &\approx W_{M\times K} \times H_{K\times N} = \widehat{\mathrm{E}}_{\mathrm{M}\times\mathrm{N}} \\ W_{M\times K} &\geq 0, H_{K\times N} \geq 0 \end{aligned} \tag{6}$$

Eventually, considering that there may be large similarities among the candidate KCs, the obtained matrix $U_{M\times K}$ need to be merged by a standard K-means clustering operation to construct the final Q-matrix.

EAKT model. Recent studies have demonstrated that the DNN-based knowledge tracing models have higher prediction performance than the Bayesian-based models. Particularly, a transformer with a self-attention mechanism can significantly enhance psychometric models in characterizing changes and interrelations of complex students' knowledge states. Therefore, it is the best choice to capture the complexity of knowledge acquisition and development as defined in Eq. (1) of the ATC framework. In view of that, the EAKT model that embeds cognitive framework of the ATC into the transformer structure is proposed. This design combines the benefits of the transformer's supreme predictive powers for the sequential learning process and the ATC's interpretability. The operating mechanism of the EAKT model is presented in Fig. 1, where it can be seen that it includes input embedding, knowledge state updating, and response prediction.

The workflow of the EAKT can be described as follows. At each timestamp, the EAKT model receives the current interaction information $x_t = (a_t, r_t)$ and updates a student's knowledge state s_t, and then predicts the possibility of answering question a_{t+1} correctly in the next timestamp according to the updated student state. In the implementation, it is assumed that N questions are related to M potential knowledge components, which can be formalized as an $N * M$ Q-matrix. It is worth mentioning that an element Q_{ij} is a float value instead of a binary value, representing the capability requirement value of a question i to a knowledge component j. A student's knowledge state and the requirements of questions are expressed in the form of a KC vector, whose dimensions represent the ability values related to the corresponding KC.

Input embedding with multi-dimensional Q-matrix. At time t, the model receives the input $x_t = (a_t, r_t)$, where a_t is thee N-dimensional one-hot encoding of questions answered at the current moment, and r_t is a binary variable representing the answering response to question a_t. First, it is needed to process a_t according to the Q-matrix to obtain the KC vector $\mathbf{c_t}$ that represents the KC requirement of question a_t. Inspired by the dAFM, this study adds a single fully-connected layer to the input stage of the EAKT model instead of taking the Q-matrix as a fixed constant. Input layer weight is initialized by the Q-matrix and is constantly adjusted during the model training. The KC vector of the question requirement $\mathbf{c_t}$ is expressed by:

$$\mathbf{c_t} = a_t \cdot Q_t^M \tag{7}$$

where Q^M represents the weight matrix of the fully-connected layer, which is initialized by the Q-matrix obtained in advance.

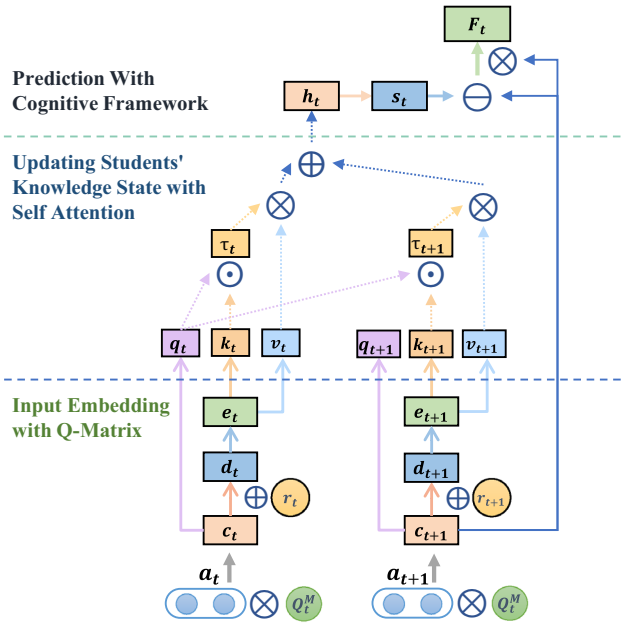

Figure 1. The overall structure of the EAKT model, including three parts: input embedding with the Q-matrix, updating students' knowledge states using the attention mechanism, and making predictions by the multi-KC cognitive framework.

Considering that the answer result has a certain influence on the change in a student's knowledge state, the consistent operation of the DKT and its variants is to extend the one-hot encoding a_t to a $2N$-dimensional vector as an input to the RNN. However, different requirements of a_t for KCs can affect the change in a student's knowledge state with each KC. Therefore, instead of using a_t, this study extends c_t to a a $2M$-dimensional vector d_t according to the value of r_t as follows:

$$d_t = \begin{cases} [\mathbf{c_t} \oplus \mathbf{0}], & r_t = 1, \\ [\mathbf{0} \oplus \mathbf{c_t}], & r_t = 0. \end{cases} \tag{8}$$

where $\oplus$ is the operation that concatenates two vectors, and $\mathbf{0}$ is a zero vector in the M dimension; $\mathbf{d_t}$ goes through a fully-connected layer to generate e_t, which represents a neural network input, so that the network can encode more information about the interaction at time t.

Updating student knowledge state by attention mechanism. Define $\mathbf{D} = (d_1, d_2, \ldots, d_l)$, $\mathbf{C} = (c_1, c_2, \ldots, c_l)$, $\mathbf{D} \in \mathbb{R}^{2M \times l}$, and $\mathbf{C} \in \mathbb{R}^{M \times l}$, where M denotes the knowledge component dimension, and l represents the input sequence length. The query, key, and value can be respectively calculated by:

$$\mathbf{Q} = \mathbf{C}\mathbf{W}^Q, \mathbf{K} = \mathbf{D}\mathbf{W}^K, \mathbf{V} = \mathbf{D}\mathbf{W}^V \tag{9}$$

Then, the scaled dot product[21] is used to generate $\mathbf{H}$ by Eq. (10), where h_t is a row t of $\mathbf{H}$.

$$\mathbf{H} = \text{ Attention } (\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V} \tag{10}$$

The dimension parameter is data-driven and determined by the training goal, which is higher prediction accuracy. This means the size of a hidden state h_t is not directly related to the students' knowledge states.

Multi-KC cognitive framework prediction effect.

$$\begin{aligned} \mathbf{c}_{t+1} &= a_{t+1} \cdot Q_t^M \\ F_t &= (\mathbf{s}_t - \mathbf{c}_{t+1}) \otimes \mathbf{c}_{t+1} \\ p_t &= Pr(R_t = 1 | \mathbf{s}_t, \mathbf{c}_{t+1}) = \phi'(F_t) \end{aligned} \tag{11}$$

where $\otimes$ represents the element-wise multiplication operation.

Equation (11) specifies a three-step calculation. First, s_t is subtracted from $\mathbf{c}_{t+1}$ to compute the difference between a student's knowledge state and a KC dimension of questions, which is denoted as a KC difference. The

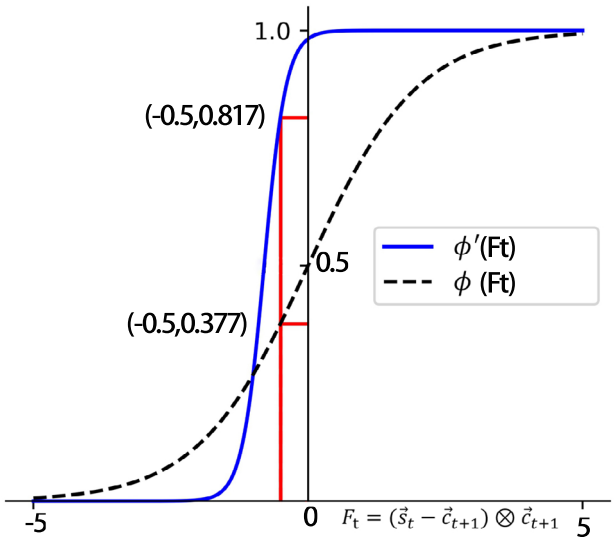

Figure 2. Comparison results of the two sigmoid functions.

measurement result of the KC difference directly affects the answering result. Second, $\mathbf{c}_{t+1}$ is set as a weight of the KC difference, and each element is multiplied to obtain the synthesis vector F_t. Third, the Sigmoid activation function Eq. (12) is modified to ϕ' given by Eq. (13) to adapt the structured model. This adjustment introduces a constant value m and a coefficient k; m is a hyperparameter, which is usually set empirically to approximately 6.9 to ensure that the student's response probability sigmoid (F_t) equals one when F_t equals zero. Because at that point, a student's mastery of knowledge state should satisfy the skill requirements of the question and be able to give a correct answer for a certainty, k is set for adjusting the slope of the sigmoid function and empirically set to 10 for better experimental performance.

$$\phi(\mathrm{F_t}) = 1/\left(1 + e^{-F_t}\right) \tag{12}$$

$$\phi'(\mathrm{F_t}) = 1/\left(1 + e^{-m-k*F_t}\right) \tag{13}$$

The difference between the original sigmoid activation function and the modified activation function is presented in Fig. 2. The effectiveness of this adjustment is verified by experiments.

Overall, we provide a pseudo code in algorithm 1 of the EAKT framework to explain the EAKT model better.

Algorithm 1: EAKT model Framework

Input : Students' historical response dataset $\mathbb{D} = \{S_1, S_2, \ldots, S_N\}$, $S = \{r_{1:T}\}$
N: Student number, T: Sequence length of student response, I: question number, L: KC number
Output : Student Knowledge state set $\Phi = \{\theta_{1:N}\}$; prediction result y_t at time t, $t \in \{1, T\}$

/* Q-matrix generation */
$\mathbf{E}_{N\times I} \leftarrow$ Convert($\mathbb{D}$) using equation (5)
$\mathbf{U}_{K\times I} \leftarrow$ NMF($\mathbf{E}_{N\times I}$) using equation (6)
$\mathbf{Q}_{L\times I} \leftarrow$ K-means($\mathbf{U}_{K\times I}$)
repeat
  foreach S_n *in* $\mathbb{D}$ **do**
    foreach *response* $r_t \in S_n$ **do**
      /* Input Embedding With Multi-dimension */
      $\vec{c}_t \leftarrow a_t \cdot Q^T$
      $\vec{d}_t \leftarrow$ Extension($\vec{c}_t$) using equation (8)
      /* Updating Student Knowledge State by Attention */
      $\mathbf{q} \leftarrow \mathbf{c_t} \cdot \mathbf{W}^Q, \mathbf{k} \leftarrow \mathbf{d_t} \cdot \mathbf{W}^K, \mathbf{v} \leftarrow \mathbf{d_t} \cdot \mathbf{W}^V$
      $\mathbf{h_t} \leftarrow \text{softmax}\left(\frac{\mathbf{q}\mathbf{k}^T}{\sqrt{d}}\right)\mathbf{v}$
      /* Multi-KC cognitive framework prediction */
      $\vec{s}_t \leftarrow \text{FC}(\vec{h}_t)$ ▷ FC is fully-connected layer
      $F_t \leftarrow (\vec{s}_t - \vec{c}_{t+1}) \otimes \vec{c}_{t+1}$
      $p_t \leftarrow Pr(R_t = 1 | \vec{s}_t, \vec{c}_{t+1}) = \phi'(F_t)$
  $\mathscr{L} = -\Sigma_n \Sigma_t \left(r_t \log\left(p_t\right) + \left(1 - r_t\right) \log\left(1 - p_t\right)\right)$
until $\mathscr{L}$ *reach the convergence condition*

Implementation and experimental results

Implementation details. *Computing infrastructure and framework setting.* We now specify the network initializations in EAKT model. The model dimension in attention as 128 and the maximum allowed sequence length l as 50. The model is trained with a mini-batch size of five. We use Adam optimizer with a learning rate of 0.001. The dropout rate is set to 0.1 to reduce overfitting. The L2 weight decay is set to 0.000001. All the model parameters unless otherwise specified are normally initialized with 0. All the experiments are conducted on 2 * Tesla V100 PCIe 32GB GPUs. Other configuration includes 2 * Intel Xeon Gold 6148 CPU, 128GB DDR4 RAM and 1* 1024GB SATA SSD. Software environment is python 3.7.4 and pytorch 1.7.1.

Evaluation methodology. *Metrics.* The prediction task is considered in a binary classification setting i.e., answering an exercise correctly or not. Hence, we compare the prediction performance using the *Area Under Curve* (AUC) metric. The cross entropy loss is also presented in the experimental results to reflect the degree of convergence of the model. To verify the interpretability of students' knowledge state, we used *Word Mover's Distance* (WMD) and *Word Rotator' Similarity* (WRS) to compare the similarity of knowledge state between ground-truth and and obtained form different models.

Network training. The objective of training is to minimize the negative log likelihood of the observed sequence of student responses under the model. The parameters are learned by minimizing the cross entropy loss between $\phi'(\mathrm{F_t})$ and r_t.

$$\mathcal{L} = -\Sigma_t \left(r_t \log\left(p_t\right) + \left(1 - r_t\right) \log\left(1 - p_t\right)\right) \tag{14}$$

Datasets. Three publicly accessible datasets were used to evaluate the prediction accuracy of the EAKT model, and a simulated dataset was used to verify the interpretability of the EAKT model. The EAKT-Q and EAKT-P represented variants of the EAKT model, of which the former denoted a model with the Q-matrix embedding but without the cognitive framework, and the latter was a model without the Q-matrix embedding but with the cognitive framework. The statistical information of the datasets is given in Table 2. To avoid the problem of sparse inputs, the common practice of the majority of the DKT-related studies to initialize the number of KCs to the total number of problems in the ASSISTments datasets was adopted. The first dataset was ASSISTments2009 [https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data], which was obtained by the ASSISTments online tutoring platform, contained 325k rows of responses of 4151 students answering 110 questions. The second dataset was ASSISTment2015 [https://sites.google.com/site/assistmentsdata/datasets/2015-assistments-skill-builder-data], which was obtained by the same platform, consisted of

Dataset	Student	Question	Interaction	Average length	Length variance	Maximum length
ASSIST2009	4,151	110	325,637	78	24293.4	1261
ASSIST2015	19,917	100	708,631	35	2542.6	632
ASSIST2017	1,709	102	942,816	551	175529.1	3057
Simu	20,000	30	1,000,000	50	0	50

Table 2. Overview of the experimental datasets.

	DKT			SAKT			EAKT-Q			EAKT-P			EAKT		
Method	KC.num	AUC	Loss	KC.num	AUC	Loss	KC.num	AUC	Loss	KC.num	AUC	Loss	KC.num	AUC	Loss
ASSIST2009	30	81.4	0.31	30	79.9	0.32	30	84.0	1.47	30	84.0	1.61	30	**84.6**	1.48
ASSIST2015	10	70.2	2.39	10	74.1	1.66	10	78.7	5.05	10	75.0	5.35	10	**80.0**	5.62
ASSIST2017	10	**71.8**	10.03	10	66.6	10.30	10	69.0	1.57	10	69.5	2.93	10	69.5	1.54
Simu	10	81.7	20.39	10	90.0	5.67	10	90.3	4.72	10	90.0	4.60	10	**90.5**	4.59

Table 3. Comparison of experimental results. Significant values are in [bold].

2014–2015 school years' student response records, containing 708k rows of non-repeating records. The records were generated by 19,917 students answering questions involving 100 questions. The third dataset was ASSISTment2017 [https://sites.google.com/view/assistmentsdatamining/dataset] containing 942,816 interactions, 1709 students, and 102 questions. These three real datasets have been widely used to evaluate the performance of the DKT and its variants.

To create a simulated dataset, a simulator with the ATC model was developed. It included a group of student agents interacting with 30 hypothetical questions and 10 KCs. In the beginning, the simulator randomly generated the initial state and a question list for each student. Then, at each simulation timestamp, it selected a question from the list to be answered for each student. Once a student solved the problem correctly, the simulator removed the problem from the student's question list. Furthermore, the simulator updated each student's knowledge state based on the student's response result after each simulation iteration by:

$$s_t(k) = max((s_{t-1}(k) + l_k) * exp(-\beta - r * \Delta t), 0) \tag{15}$$

- $s_t(k)$ denotes the ability of a student s for a knowledge component k at time t;
- l_k denotes the improvement in students' ability for a knowledge component k after answering questions at time $(t-1)$;
- β and r are the forgetting parameters;
- Δt denotes the time difference between the time t and time $(t-1)$.

Based on the students' knowledge state at moment t and the KC requirement of a question i, the probability of answering the question correctly was obtained by the ATC model defined by Eq. (1).

Student performance prediction. Experimental results on all datasets are presented in Table 3. The AUC and loss values of all models were calculated to evaluate their prediction performances in the experiment. To verify the prediction accuracy of student abilities, the training and test datasets of the experiment were defined as follows. The first 80% of each student's answer sequence was set as a training dataset, and the remaining 20% of data denoted the test dataset. There are usually two approaches to divide the test dataset, one according to the student cut and one according to the sequence cut, because in the experiment we need to verify the prediction performance of the model in addition to verify the prediction performance on the knowledge state, and the sequence cut according to the sequence can allow the previous sequence of each student to participate in the training, so that the prediction of the knowledge state is more accurate. In the experiment of the EAKT model, the KC number values of the three real datasets were 30, 10, and 30, and that of the simulated dataset was 10. The Q-matrix of the simulated dataset was generated by the ATC model. According to the comparison experiment on the simulated dataset, the Q-matrix generated by the ATC model could improve the prediction accuracy of the model. Since the ATC model cannot handle large-scale datasets, the Q-matrix of the three real datasets was obtained by the NMF.

The AUC and loss of the EAKT were compared with those of the DKT, SAKT, EAKT-Q and EAKT-P models on four datasets. For the Assist2009 dataset, the average AUC value achieved by the EAKT model was 84.6%, which was higher than those four models. The predictive performances of the five models on the ASSIST2015 dataset were similar to those on the ASSIST2009 dataset. The AUC values of the three models were 70.2%, 74.1%, 78.7%,75.0%, and 80.0%. The results indicated that the EAKT model outperformed the DKT model based the LSTM structure. The EAKT model also outperformed the SAKT model with the transformer structure on three

datasets. Experimental results showed that embedding cognitive structures in a DNN-based knowledge tracing framework could improve prediction performance.

Interpretable knowledge state. The main contribution of the EAKT model is the ability to present every student's knowledge state in an explanatory way, which is vital for implementing adaptive personalized learning. The DKT model abstracts knowledge state representation in hidden states of an RNN, resulting in the difficulty in interpreting every student's cognitive skill level and dynamic state changes. The EAKT model constrains the transformer by embedding the major elements of the ATC model to reveal the latent knowledge state and the changing trend over time from the hidden state of the neural network. To verify that the students' knowledge states obtained by the EAKT model are explicable and accurate, the simulated dataset was used to compare the output result of the EAKT with the ground-truth state of each student agent at each timestamp. Two evaluation metrics, namely the word mover's distance[22] and word rotator's distance[23] were employed to calculate the similarity between the students' latent state and inferred s_t from the hidden units of the DKT, SAKT, and EAKT. Assume that a student S has a knowledge state sequence of $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_n$, then a student S' has the knowledge state sequence $\boldsymbol{w}'_1, \boldsymbol{w}'_2, \ldots, \boldsymbol{w}'_m$.

1. Word mover's distance: p_i and q_j are defined by Eq. (16):

$$p_i \equiv \frac{1}{n}, q_j \equiv \frac{1}{m} \tag{16}$$

The word mover's distance was defined by:

$$\begin{aligned} \mathrm{WMD}(S, S') &= \min_{\gamma_{i,j} \geq 0} \sum_{i,j} \gamma_{i,j} \left\| \boldsymbol{w}_i - \boldsymbol{w}'_j \right\| \\ \text{s.t.} \quad \sum_j \gamma_{i,j} &= \frac{1}{n}, \sum_i \gamma_{i,j} = \frac{1}{m} \end{aligned} \tag{17}$$

2. Word rotator's distance: p_i and q_j were calculated by Eq. (18):

$$\begin{aligned} p_i &= \frac{\|\boldsymbol{w}_i\|}{Z}, \quad Z = \sum_{i=1}^{n} \|\boldsymbol{w}_i\| \\ q_j &= \frac{\left\|\boldsymbol{w}'_j\right\|}{Z'}, \quad Z' = \sum_{j=1}^{n'} \left\|\boldsymbol{w}'_j\right\| \end{aligned} \tag{18}$$

The word rotator's similarity was calculated by:

$$\begin{aligned} d_{i,j} &= 1 - \frac{\boldsymbol{w}_i \cdot \boldsymbol{w}'_j}{\|\boldsymbol{w}_i\| \times \left\|\boldsymbol{w}'_j\right\|} \\ \mathrm{WRS}(S, S') &= 1 - \min_{\lambda_{i,j} \geq 0} \sum_{i,j} \lambda_{i,j} d_{i,j} \\ \text{s.t.} \quad \sum_j \lambda_{i,j} &= p_i, \quad \sum_i \lambda_{i,j} = q_j \end{aligned} \tag{19}$$

According to the above definitions, the word mover's distance indicates a disproportion to the state similarity, while the word rotator's similarity means the opposite. In the experiments, 4,000 students knowledge states form test dateset obtained by the EAKT model, EAKT-O model, SAKT model, and DKT model were compared with the ground-truth states. The two evaluation metrics were calculated. The EAKT-O model represented the EAKT model with the original sigmoid activation function. The sum values of the WMD and WRS were denoted by the WMD.T and WRS.T, and their average values were denoted by WMD.A and WRS.A, respectively. They were calculated by Eq. (20), where N is the total number of students.

$$\begin{aligned} WMD.T &= \sum_{i=1}^{N} \mathrm{WMD}(S_i, S_i') \qquad WMD.A = \sum_{i=1}^{N} \mathrm{WMD}(S_i, S_i')/N \\ WRS.T &= \sum_{i=1}^{N} \mathrm{WRS}(S_i, S_i'), \qquad WRS.A = \sum_{i=1}^{N} \mathrm{WRS}(S_i, S_i')/N \end{aligned} \tag{20}$$

The student's knowledge state obtained by the EAKT model was the closest to the ground-truth state, having the highest average value of WRS.A, which was higher than those of the SAKT, EAKT-O, EAKT-Q, EAKT-P, and DKT models. The similarity matrices between the four models and the ground truth for the two distance metrics are presented in Fig. 3. The results indicated that the student's knowledge state obtained by the EAKT model outperformed those of the other models. Interestingly, the EAKT-O performed the worst in terms of the

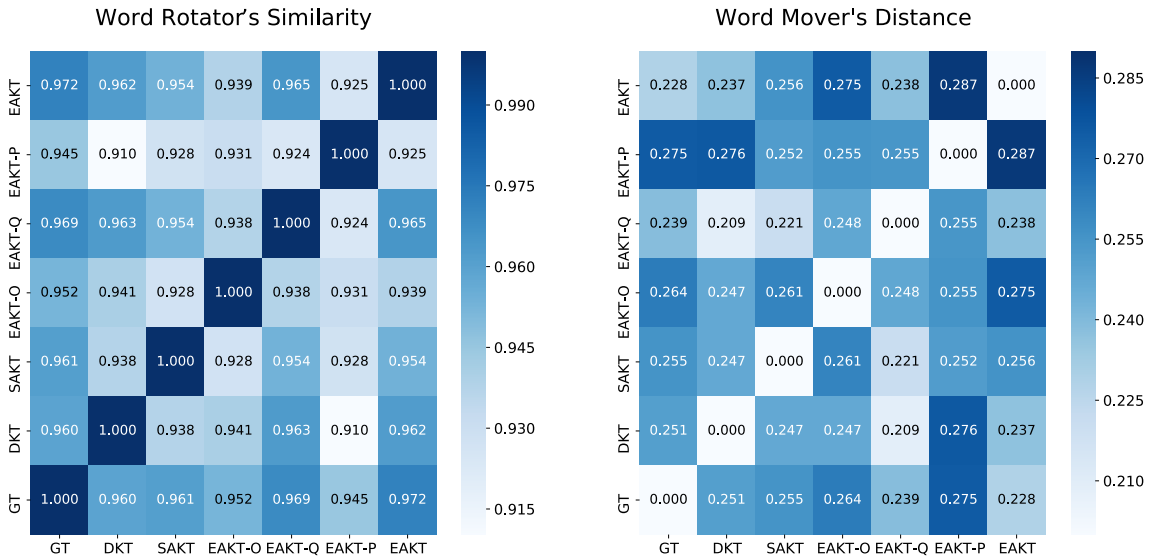

Figure 3. The similarity comparison of the knowledge states obtained by the four models and the ground-truth knowledge states in terms of the distance metrics.

two similarity metrics, which confirmed that the optimized activation function was effective in computing the student's knowledge state (Supplementary information).

Conclusions and future work

In this paper, a knowledge tracing model named the EAKT is developed using self-attention mechanism and a structured ATC model. The proposed model can trace the knowledge state of students in every timestamp while predicting their future performance. Particularly, this paper introduces a multi-dimensional KC vector to represent students' knowledge states and a Q-matrix to represent the KC requirements of questions in deep neural networks. The experiments on real datasets and simulated datasets verify that the proposed EAKT model can obtain an interpretable multi-dimensional sequence of students' knowledge states on the premise of preserving the prediction power of the self-attentive transformer framework. A combination of explanatory and predictive power in the EAKT model enables the better design of intelligent tutoring applications. In the future, we plan to explore the use of deep learning frameworks to enhance cognitive models, such as using adjustable weights to represent Q-matrix and enhance it through introducing exercise texts.

Data availability

The datasets generated and analysed during the current study are available in the EAKT repository, https://github.com/ranydb/EAKT.

Received: 2 June 2022; Accepted: 17 October 2022
Published online: 20 October 2022

References

1. Stamper, J. C. & Koedinger, K. R. Human-machine student model discovery and improvement using datashop. In *Artificial Intelligence in Education—15th International Conference, AIED 2011, Auckland, New Zealand* (2011).
2. Koedinger, K. R., Stamper, J. C., McLaughlin, E. A. & Nixon, T. Using data-driven discovery of better student models to improve student learning. In *International Conference on Artificial Intelligence in Education* 421–430 (Springer, 2013).
3. Velmahos, G. C. *et al.* Cognitive task analysis for teaching technical skills in an inanimate surgical skills laboratory. *Am. J. Surg.* **187**, 1–119 (2004).
4. Khajah, M., Lindsey, R. V. & Mozer, M. C. *How Deep is Knowledge Tracing?* aarXiv:1604.02416 (2016).
5. Corbett, A. T. & Anderson, J. R. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User-Adapted Interact.* **4**, 253–278 (1994).
6. Tatsuoka, K. K. Rule space: An approach for dealing with misconceptions based on item response theory. *J. Educ. Measure.* **20**, 345–354 (1983).
7. Pu, Y., Wu, W. & Jiang, T. Atc framework: A fully automatic cognitive tracing model for student and educational contents. *EDM* **2019**, 5 (2019).
8. Piech, C. *et al.* Deep knowledge tracing. *Adv. Neural Inf. Process. Syst.* **2015**, 505–513 (2015).
9. Zhang, J., Shi, X., King, I. & Yeung, D.-Y. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web* 765–774 (2017).
10. Huang, Z. *et al.* Ekt: Exercise-aware knowledge tracing for student performance prediction. *IEEE Trans. Knowl. Data Eng.* **33**, 100–115 (2019).
11. Yeung, C.-K. *Deep-irt: Make Deep Learning Based Knowledge Tracing Explainable Using Item Response Theory*. arXiv:1904.11738 (2019).
12. Bier, N., Lip, S., Strader, R., Thille, C. & Zimmaro, D. An approach to knowledge component/skill modeling in online courses. *Open Learn.* **2014**, 1–14 (2014).

13. González-Brenes, J. P. & Mostow, J. Dynamic cognitive tracing: Towards unified discovery of student and cognitive models. *Int. Educ. Data Min. Soc.* **2012**, 5 (2012).
14. González-Brenes, J., Huang, Y. & Brusilovsky, P. General features in knowledge tracing to model multiple subskills, temporal item response theory, and expert knowledge. In *The 7th International Conference on Educational Data Mining* 84–91 (University of Pittsburgh, 2014).
15. Yeung, C.-K. & Yeung, D.-Y. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale* 1–10 (2018).
16. Chen, P., Lu, Y., Zheng, V. W. & Pian, Y. Prerequisite-driven deep knowledge tracing. In *2018 IEEE International Conference on Data Mining (ICDM)* 39–48 (IEEE, 2018).
17. Vaswani, A. *et al. Attention is All You Need*. arXiv:1706.03762 (2017).
18. Pandey, S. & Karypis, G. *A Self-Attentive Model for Knowledge Tracing*. arXiv:1907.06837 (2019).
19. Ghosh, A., Heffernan, N. & Lan, A. S. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* 2330–2339 (2020).
20. Pardos, Z. A. & Dadu, A. dafm: Fusing psychometric and connectionist modeling for q-matrix refinement. *JEDM J. Educ. Data Min.* **10**, 1–27 (2018).
21. Vaswani, A. *et al.* Attention is all you need. In *Advances in Neural Information Processing Systems*,5998–6008 (2017).
22. Kusner, M., Sun, Y., Kolkin, N. & Weinberger, K. From word embeddings to document distances. In *International Conference on Machine Learning* 957–966 (PMLR, 2015).
23. Yokoi, S., Takahashi, R., Akama, R., Suzuki, J. & Inui, K. *Word Rotator's Distance*. arXiv:2004.15003 (2020).

Acknowledgements

This work is supported in part by the National Key Research and Development Program of China (Funding No. 2018YFB1004502) and the State Key Laboratory of Software Development Environment (Funding No. SKLSDE- 2020ZX-01).

Author contributions

Y.P. and W.W. wrote the main manuscript text, T.P., F.L., and Y.L. worked on data curation, X.Y., R.C., P.F. prepared all the figures and tables, all authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-022-22539-9.

Correspondence and requests for materials should be addressed to Y.P.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.