Dynamic Cognitive Diagnosis: An Educational Priors-Enhanced Deep Knowledge Tracing Perspective

Fei Wang, Zhenya Huang, *Member, IEEE,* Qi Liu, *Member, IEEE,* Enhong Chen, *Senior Member, IEEE,* Yu Yin, Jianhui Ma, and Shijin Wang.

***Abstract*—To provide personalized support on educational platforms, it is crucial to model the evolution of students' knowledge states. Knowledge tracing is one of the most popular technologies for this purpose, and deep learning-based methods have achieved state-of-the-art performance. Compared to classical models, such as Bayesian knowledge tracing, which track students' knowledge proficiencies, deep learning-based knowledge tracing is usually modeled to predict students' performances on questions, while ignoring the interpretability of students' knowledge states. However, for many practical applications such as learning resource recommendation, it would be more helpful if we could explicitly track the students' abilities or knowledge proficiencies separately from performance prediction. Researchers in psychometric area already designed cognitive diagnosis solutions to quantify the knowledge states of students in static conditions (e.g., examination), where the educational priors (i.e., factors related to students' learning process) were proved beneficial for student modeling. Inspired by this, we propose *Dynamic Cognitive Diagnosis*, which integrates the interpretability of educational priors from cognitive diagnosis into deep learning-based knowledge tracing methods. We first discuss and provide evidence of which educational priors can be integrated, including question attributes and interaction function. Then we show the effects of using the educational priors in deep learning-based knowledge tracing from two aspects, i.e., interpretability and accuracy. Through extensive experiments and analyses, we prove that properly chosen priors can enable deep learning-based methods to evaluate students' knowledge states in a manner that is consistent with domain knowledge or human experience. Moreover, educational priors also improve the accuracy of student performance prediction.**

***Index Terms*—Intelligent tutoring systems, personalized e-learning, cognitive diagnosis, knowledge tracing, deep learning**

I. Introduction

APPLYING artificial intelligence (AI) to enhance the educational technologies has been an important tide in recent years and has attracted intensive interest around the world [1], [2]. Compared to traditional face-to-face learning, students can access a greater variety of learning resources and receive personalized support for their learning in intelligent educational platforms (e.g., intelligent tutoring systems). Advanced AI technologies are required to handle the massive amounts of data from learners and learning resources in the learning platforms and provide effective services for each learner, such as learning resource recommendation [3], early warning of failure [4], learning path recommendation [5], and adaptive learning [6]. Among these technologies, modeling the evolution of students' knowledge states is a crucial task that serves as the backbone of numerous personalized supports. Knowledge tracing, one of the most promising solutions for this task, aims to track the students' knowledge states and predict their future performances (e.g., scores) through mining their historical learning activities (especially question answering).

The most traditional approach to knowledge tracing is Bayesian knowledge tracing (BKT) [7] and its variations, which track the student's mastery of each knowledge concept using a hidden Markov process. After recurrent neural network was first used for knowledge tracing in DKT model [8], deep learning-based methods [9], [10] have achieved state-of-the-art performance in knowledge tracing due to their advantages in sequence modeling (with recurrent neural networks, memory networks, etc.).[1] However, despite their success in student performance prediction, most existing deep knowledge tracing works cannot provide the explicit states of the students that indicate their levels of mastery of specific knowledge components (knowledge proficiency). Tracking students' knowledge proficiencies can facilitate the generation of more detailed reports about the students and will be more helpful for practical applications than simply knowing what scores they would get. For example, when recommending learning resources, the tutoring system needs to first know the knowledge components or skills that the student is poor at, and then recommend relevant resources (e.g., teaching videos). This stresses the importance of diagnosing students' knowledge proficiencies.

This work was supported by the National Natural Science Foundation of China (Grant 61922073, U1605251, and U20A20229), and the Iflytek joint research program. (*Corresponding author: Qi Liu)*

Fei Wang, Zhenya Huang, Qi Liu, Enhong Chen and Yu Yin, Jianhui Ma are with the Anhui Province Key Lab. of Big Data Analysis and Application, University of Science and Technology of China, Hefei, 230026, Anhui, China. They are also with the State Key Laboratory of Cognitive Intelligence, Wangjiang West Road, Hefei, 230088, Anhui, China. Qi Liu is also with the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Wangjiang West Road, Hefei, 230088, Anhui, China. (email: wf314159@mail.ustc.edu.cn; huangzhy@ustc.edu.cn; qiliuql@ustc.edu.cn; cheneh@ustc.edu.cn; yxonic@mail.ustc.edu.cn; jianhui@ustc.edu.cn).

Shijin Wang is with the iFLYTEK AI Research (Central China), Checheng North Road, Wuhan, 430058, Hubei, China. He is also with the State Key Laboratory of Cognitive Intelligence, Wangjiang West Road, Hefei, 230088, Anhui, China (email: sjwang3@iflytek.com).

[1]For convenience, in the remainder of this paper, we use **deep knowledge tracing** to represent deep learning-based knowledge tracing methods and use **DKT** as the abbreviation of the model proposed in [8].

The main weakness of deep knowledge tracing lies in its modeling of the relation between student knowledge states and student performance, which lacks adequate interpretability. In the academic field of psychometrics and education, test theory has been widely studied to model students' knowledge states in static scenes (e.g., examination), where the students' knowledge states are assumed to be unchanged when they answer a set of questions [11]. Among them, **cognitive diagnosis** is the most typical. This approach aims to obtain the students' knowledge proficiency on each predefined Knowledge Component (KC; e.g., *Addition*) based on their test performance. Representative works include item response theory (IRT) [12], multi-dimensional item response theory (MIRT) [11], deterministic input, noisy-and model (DINA) [13], reparameterized unified model (RUM) [14], etc. In these works, educational priors such as question attributes and interaction functions are used to describe the cognitive patterns behind the question-answering process. Question attributes model the characteristics of questions that influence this answering process (such as question difficulty and discrimination in IRT and MIRT, or knowledge component, slip and guess probability in DINA and RUM). Interaction functions model the relation between students and questions, which take student knowledge states and question attributes as input and output the probability of giving a correct answer. For example, IRT gives the probability of a correct answer with function $f(\theta; a, \gamma) = 1/(1 + \exp(-a(\theta - \gamma)))$; here, θ is the student's ability, while a, γ denote question discrimination and difficulty respectively [15]. The function indicates that the more difficult the question is, the higher ability is required to correctly answer the question. Recently a new cognitive diagnostic framework named NeuralCD [16] has been proposed, which combines neural networks and the monotonicity assumption to further improve the fitting capability of the response function while retaining the interpretability of the parameters.

In order to compensate for the limitations of deep knowledge tracing, we propose to combine the educational priors from cognitive diagnosis with the advantage of sequential modeling in deep knowledge tracing, an approach that we refer to as **Dynamic Cognitive Diagnosis**. Fig. 1 illustrates a toy example. Each time the student answers a question (can either be recommended by the systems or chosen by self), our goal is to evaluate the student's mastery level on each knowledge component (e.g., *Addition*) from their answering history. The diagnosis results can be clearly reported, and at the same time used for further services, such as personalized recommendations of learning resources. To achieve this goal, we need to answer two questions:

Research Question 1: *What educational priors can be brought to deep knowledge tracing?*

Research Question 2: *What effects do educational priors bring to deep knowledge tracing?*

By answering these questions, the contributions of this work can be summarized as follows:

- We discuss the educational priors that can be integrated to deep knowledge tracing, and propose a dynamic cognitive diagnosis framework that integrates educational priors from cognitive diagnosis with deep knowledge tracing.

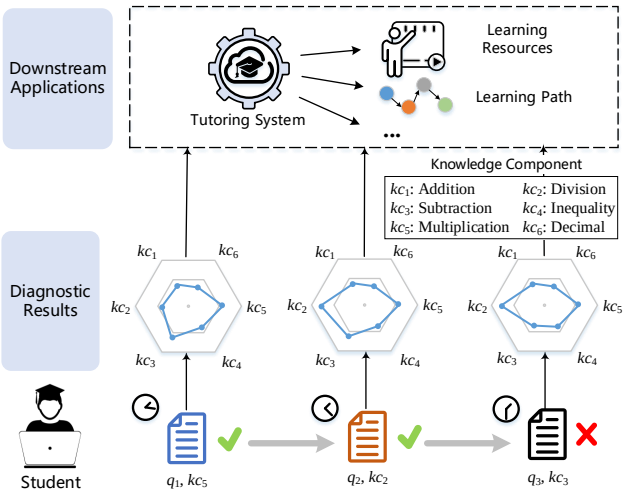


Fig. 1. A toy example of dynamic cognitive diagnosis. A student successively answers some questions (e.g., q_1) at different times, and each question is associated with certain knowledge components (e.g., question q_1 contains knowledge concept kc_5, which denotes *Multiplication*). According to the responses (right or wrong), we evaluate the student's level of mastery over each knowledge component. The diagnostic results can be used for downstream applications such as learning resource recommendation.

- With extensive experiments and analysis, we qualitatively and quantitatively measure the effects that educational priors bring about, including better interpretability and higher prediction accuracy than deep knowledge tracing.

The remainder of this paper is organized as follows. Section II reviews the background of this work. Section III presents discussions and examples of the two research questions. Subsequently, we verify our proposal in Section IV with experiments and analysis, and make discussions in Section V. Finally, we present our conclusions and talk about future works in Section VI. For easier checking, we summarize the notations that are important and frequently used in this paper in Table I.

II. Background

Knowledge tracing and cognitive diagnosis for students are the foundations of this research. A brief review of related works is provided below.

A. Knowledge tracing

Knowledge tracing is a type of task that caters to the demand of modeling students' knowledge states. With the learning systems, students are free to choose learning sources and do exercises by themselves. Their degrees of mastery over knowledge components can change frequently, which is reflected by their learning activities in the systems. The goal of knowledge tracing is to track the knowledge states of students and predict their future performance based on their historical learning activities (e.g., answering questions). Knowledge tracing models the sequential characteristics of students' learning activities and makes dynamic predictions.

The most popular knowledge tracing approach among earlier works was Bayesian knowledge tracing (BKT) [7], which modeled student's mastery of each knowledge component

TABLE I
NOTATIONS USED IN THIS PAPER

Variable	Description
s_n	student n
q_m	question m
kc_k	knowledge component k
$\mathcal{R}_n$	the response sequence of student n
$\mathcal{R}_n^t$	the t-th element of $\mathcal{R}_n$, the response of s_n at time t
q_n^t	the question that s_n answers at time t
$\boldsymbol{q}_n^t$	the embedding of question q_n^t
T_n	the length of student n's response log
$\hat{y}_{n,m}^t$	the m-th element of $\hat{\boldsymbol{y}}_n^t$, predicted (0,1) probability of s_n giving correct response to q_m after time t
y_n^t	{0,1} response ({incorrect, correct}) of s_n at time t
$\hat{\theta}_{n,k}^t$	the k-th element of $\hat{\boldsymbol{\theta}}_n^t$, predicted (0,1) probability that s_n could give correct responses to questions containing KC kc_k after time t
θ_n^t	predicted (0,1) overall ability of student s_n after time t
$\tilde{\theta}_{n,k}^t$	the k-th element of $\tilde{\boldsymbol{\theta}}_n^t$, predicted (0,1) proficiency of s_n on kc_k after time t
$Q_{m,k}$	the element of the m-th row and the k-th column of Q-matrix $\boldsymbol{Q}$, {0,1} value of whether q_m contains kc_k
γ_n^t	(0,1) value of the difficulty of q_n^t
a_n^t	(0,1) value of the discrimination of q_n^t
$\beta_{n,k}^t$	(0,1) value of the difficulty of kc_k in q_n^t

using a hidden Markov process. Subsequent research improved BKT by considering the various factors that influence the evolution of students' knowledge states, such as individualized student parameters [17], forgetting [18], knowledge topologies [19], and intervention types [20]. Some works proposed to integrate BKT with latent factor models to enhance the fitting capability of BKT [21]. Another representative example is performance factor analysis (PFA) [22], in which a student's performance is modeled as the accumulation of learning from both successful attempts and unsuccessful attempts. However, PFA did not provide explicit knowledge proficiencies as BKT-based models do.

Deep learning was first introduced into knowledge tracing by DKT [8], in which a recurrent neural network (RNN) was used to model the evolution of students' knowledge states. This approach yielded significant improvement in student performance prediction. Since then, deep learning-based methods reached state of the art on knowledge tracing. Compared to earlier works, deep knowledge tracing methods can achieve more advanced sequential modeling through their use of deep learning, such as RNN in the DKT model [8] and key-value memory network in the DKVMN model [9]. Due to the personalized nature of online education platforms, the response data of students are usually sparse; this was addressed in [10] with the prerequisite constraint between knowledge components. A knowledge query network was proposed in [23] in which knowledge components were encoded with positive and unit-length restrictions.

There have been attempts to integrate educational factors for improving accuracy. For example, Yang et al. [24] considered various statistical features, such as question ID, question type, number of attempts and hint. Sonkar et al. [25] considered that questions containing the same knowledge components should have close probabilities of being answered correctly. Nagatani et al. [26] considered forgetting behaviors by embedding practice space and number of attempts into the model input. However, these factors were integrated mainly through feature embedding, and the models were still black boxes. Although there were a few post-hoc analyses about the effects of educational factors on model predictions, the interpretability of student knowledge states was still limited due to the model structures and was rarely quantitatively analyzed. It has been broadly accepted that some fields require high level of accountability and thus transparency, such as education and medical science [27]. Lack of interpretability of the models would painfully impede their practical applications [28].

B. Cognitive diagnosis

Cognitive diagnosis is an important research branch of test theories, which studies the relation between students' knowledge states and their test performances. Different from knowledge tracing, test theories are mostly designed for tests during which the knowledge states of students are assumed to be static. Moreover, test theory methods are mostly designed based on educational or psychometric theories and assumptions; thus, most of them, especially cognitive diagnosis, can provide explainable diagnostic reports. Based on diagnostic level, existing studies can be classified to ability level paradigm and cognition level paradigm [29].

Approaches belonging to the ability level paradigm diagnose the students at the macro level. Representative works include classical test theory (CTT) [30] and item response theory (IRT) [31], [32]. CTT assumed that the observed test score is the sum of the true score (which characterizes the student's ability) and error. Unlike CTT, IRT outputs the probability of correctly answering a question through a logistic-like function with unidimensional student ability and question parameters as input. Question parameters could include question difficulty [33], discrimination [15] and guess probability [34].

Approaches belonging to the cognition level paradigm diagnose the students at the micro level, which typically analyze the students' knowledge proficiencies on each knowledge component. Representative works include the rule space model [35], deterministic input, noisy-and model (DINA) [36], noisy inputs, deterministic-and model (NIDA) [37], and reparameterized unified model (RUM) [14]. With the use of Q-matrix (a

binary matrix that indicates the knowledge components of each question) [38], these models are able to diagnose students' abilities at the knowledge component level. Besides, multidimensional IRT was proposed to improve the fitting ability of IRT [11]. However, the multidimensional student ability vector was usually not explainable. Neural network-based cognitive diagnosis framework (NeuralCD) was recently proposed [16], which used neural network to learn the interaction function between students and questions, and simultaneously ensured the interpretability of students' knowledge proficiency vectors.

Although cognitive diagnosis focuses on the cognition level paradigm, works from both paradigms can be the source of educational priors.

III. Dynamic cognitive diagnosis

In this section, we first present preliminaries of our problem. Then we review normal structure of deep knowledge tracing models. After that, through in-depth discussions of the two research questions (what educational priors can be brought and what they can bring to deep knowledge tracing) in detail, we show examples of dynamic cognitive diagnosis models.

A. Preliminary

1) Problem Definition: Suppose there are N students, M questions, and K relevant knowledge components. In the dataset, the response history of a student s_n is a sequence $\mathcal{R}_n = \{(q_n^t, y_n^t)|t = 1, 2, \ldots, T_n\}$, where T_n is the sequence length, q_n^t is the question that student s_n answered at time t and y_n^t is the response (i.e., correct or incorrect).

Definition 1. (*Dynamic Cognitive Diagnosis*) Given each student's response history $\{\mathcal{R}_1, \mathcal{R}_2, \ldots\}$, the goal of dynamic cognitive diagnosis is to train a model such that when a student's (e.g., s_n; usually not in the training data) response history $\{(q_n^\tau, \mathcal{R}_n^\tau)|\tau = 1, 2, \ldots, t\}$ is input, the model outputs the student's ability $\theta_n^t \in (0, 1)$ or proficiency $\tilde{\boldsymbol{\theta}}_n^t \in (0, 1)^K$ that denotes s_n's levels of mastery of the knowledge components $kc_1, \ldots, kc_K$ after time t.

In the definition, we do not place limits on the information about the questions (e.g., question ID, question knowledge components) given to the models. We will discuss the question attributes in detail later in subsection III-C.

2) Interpretability: There is currently no consensus on the definition of "interpretability" in machine learning. In this study, we first give a definition of interpretability based on previous attempts [27].

Definition 2. (*Interpretability*) The interpretability of a model is the ability to explain the reasoning behind model decisions in terms understandable to a human.

In our case, it can be further divided into two aspects: 1) the model decisions (i.e., parameters such as students' knowledge states) should be consistent with domain knowledge or human experience; 2) there is a clear relation between model inputs (i.e., students' response histories) and the model's decisions.

B. Deep learning-based knowledge tracing

The normal structure of deep knowledge tracing models is composed of two modules: sequential modeling and performance prediction (Fig. 2). The sequential modeling module takes the student's response sequence as input, and fits the evolving pattern of the student's hidden state with sequential models, such as recurrent neural network (used in DKT) and memory network (used in DKVMN). To facilitate fair comparison and without loss of generality, we use Gate Recurrent Unit (GRU) as the sequential module for all models in this work before and after integrating educational priors. GRU is a recurrent neural network architecture first proposed in [39], and has achieved good performance in various sequential modeling tasks [40], [41]. In GRU, a hidden state vector $\boldsymbol{h}_n^t$ is preserved and updated after each input $\boldsymbol{x}_n^t$, and the update is performed by "forgetting" part of the information from $\boldsymbol{h}_n^t$ and "learning" information from $\boldsymbol{x}_n^t$. Thus, the sequential modeling process is formulated as follows:

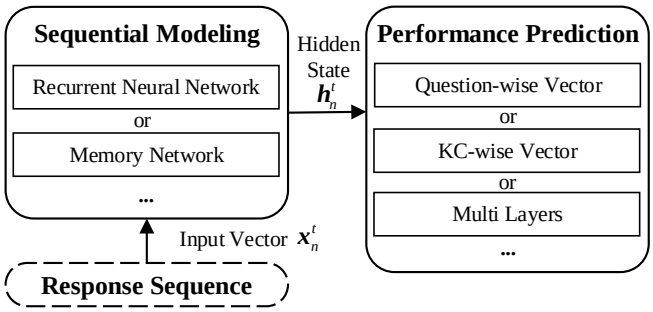


Fig. 2. Normal structure of deep knowledge tracing models.

$$\boldsymbol{r}_n^t = \sigma(\boldsymbol{W}_{ir}\boldsymbol{x}_n^t + \boldsymbol{W}_{hr}\boldsymbol{h}_n^{t-1} + \boldsymbol{b}_r), \quad (1)$$

$$\boldsymbol{z}_n^t = \sigma(\boldsymbol{W}_{iz}\boldsymbol{x}_n^t + \boldsymbol{W}_{hz}\boldsymbol{h}_n^{t-1} + \boldsymbol{b}_z), \quad (2)$$

$$\tilde{\boldsymbol{h}}_n^t = \tanh(\boldsymbol{W}_{ih}\boldsymbol{x}_n^t + \boldsymbol{r}_n^t \circ (\boldsymbol{W}_h\boldsymbol{h}_n^{t-1}) + \boldsymbol{b}_h), \quad (3)$$

$$\boldsymbol{h}_n^t = (1 - \boldsymbol{z}_n^t) \circ \tilde{\boldsymbol{h}}_n^t + \boldsymbol{z}_n^t \circ \boldsymbol{h}_n^{t-1}, \quad (4)$$

where $\boldsymbol{r}_n^t$ is the reset gate vector that controls what information should be forgotten from $\boldsymbol{h}_n^{t-1}$. $\boldsymbol{z}_n^t$ is the update gate vector that controls what information should be updated to the student's hidden state. $\tilde{\boldsymbol{h}}_n^t$ is the candidate activation gate vector that is the result of forgetting part of the information from $\boldsymbol{h}_n^{t-1}$, and $\boldsymbol{h}_n^t$ is the hidden state of s_n at time t, which is finally updated with Eq. (4). $\boldsymbol{x}_n^t$ is the input vector of student s_n at time t, σ is the Sigmoid function that $\sigma(x) = 1/(1+\exp(-x))$, $\circ$ is the element-wise product. $\boldsymbol{W}_{**}$, and $\boldsymbol{b}_*$ are parameters that will be learned after model training.

The the hidden state $\boldsymbol{h}_n^t$ is then used to predict the student's outcome of answering questions after time t. The performance prediction can be performed through a question-wise vector (the dimension of the vector is the number of questions), a KC-wise vector (the dimension of the vector is the number of KCs), or multi layers that output a probability. Prediction with a question-wise vector is the original method described in DKT, which is implemented with a normal full connection layer that transforms $\boldsymbol{h}_n^t$ into an M-dimensional vector:

$$\hat{\boldsymbol{y}}_n^t = \sigma(\boldsymbol{W}_y\boldsymbol{h}_n^t + \boldsymbol{b}_y), \quad (5)$$

where $\boldsymbol{W}_y$ and $\boldsymbol{b}_y$ are parameters learned after model training. The m-th element in $\hat{\boldsymbol{y}}_n^t$ denotes the probability of student s_n correctly answering question q_m.

However, in experiments, question-wise prediction usually suffers from the sparsity problem [42]. Thus, KC-wise prediction is more frequently adopted. In KC-wise prediction, each question is represented with its KC tag and the prediction is on the granularity of KC. It is formulated as follows:

$$\hat{\boldsymbol{\theta}}_n^t = \sigma(\boldsymbol{W}_\theta \boldsymbol{h}_n^t + \boldsymbol{b}_\theta), \tag{6}$$

where $\boldsymbol{W}_\theta$ and $\boldsymbol{b}_\theta$ are learnable parameters. The $\hat{\boldsymbol{\theta}}_n^t$ is a K-dimensional vector transformed from $\boldsymbol{h}_n^t$, where the k-th element denotes the probability that student s_n correctly answers questions containing knowledge component kc_k.

Furthermore, with multi layers, more complicated interactions between students and questions can be modeled. Taking $\boldsymbol{h}_n^t$ and question embedding $\boldsymbol{q}_m$ as input, the probability is:

$$\hat{y}_{n,m}^t = \mathcal{F}(\boldsymbol{h}_n^t, \boldsymbol{q}_m; \vartheta), \tag{7}$$

where $\mathcal{F}(\cdot;\vartheta)$ denotes the multi-layer function with learnable parameters ϑ. The function takes $\boldsymbol{q}_m$ ($\boldsymbol{q}_n^{t+1}$ during training, denoting the question that s_n answered at time $t+1$ in the data) and $\boldsymbol{h}_n^t$ as input, and outputs the probability that student s_n will correctly answer question q_m after time t. $\mathcal{F}(\cdot|\vartheta)$ can be implemented with different multi-layer structures. We will provide an example in the experiments (Eq. (19)˜ (21)).

In conclusion, the hidden states of the students in these deep knowledge tracing models have limited interpretability, thus the models are not competent for dynamic cognitive diagnosis. Besides, in KC-wise prediction, the question information is lost, and questions with multiple KCs are not handled properly.

C. Research Question 1: What educational priors can be brought to deep knowledge tracing?

In this paper, we define educational priors as "the factors from educational theories or technology research, that are thought of or discovered related to students' learning process". This is a relatively broad definition. Various factors can be considered as educational priors, such as the learning and forgetting curves from memory theory [43], [44], the question types [24] and difficulties [7], [11] from student modeling methods (e.g., cognitive diagnosis and knowledge tracing). It is not possible to study all priors in one study. In this paper, we focus on educational priors from cognitive diagnosis.

In more detail, two types of educational priors from cognitive diagnosis are considered herein: question attribute and interaction function (also called item response function in IRT). The former helps distinguish the factors that influence the result of a response, while the latter models how the response is given by the student to the given question. After discussing the educational priors, we go on to show where they can be integrated into deep knowledge tracing models.

Question attribute. Here, we discuss three types of question attributes frequently considered in cognitive diagnosis: knowledge component, difficulty, and discrimination.

1) Knowledge Component: The knowledge components required by the question is the necessary information for diagnosing knowledge proficiency. In cognitive diagnosis models (e.g., DINA [36], NeuralCDM [16]), KC is obtained with Q-matrix $\boldsymbol{Q} \in \{0,1\}^{M\times K}$, where $Q_{m,k} = 1$ if question q_m contains the KC kc_k, and $Q_{mk} = 0$ otherwise. The Q-matrix can be either human-labeled or learned from data [45].

2) Difficulty: A more difficult question requires higher proficiency to answer. In IRT models[31], [32], a scalar parameter γ is used to characterize the overall difficulty of a question. In NeuralCDM, a K-dimensional-vector parameter $\boldsymbol{\beta}$ is used to indicate the difficulty of each KC required by the question. The difficulty parameter is typically estimated through model training.

3) Discrimination: The discrimination indicates the ability of the question to discriminate between students with different mastery levels [11]. For example, when answering a question with discrimination $a = 0.9$, a student with a proficiency 0.8 would much more likely to get a higher score than another with a proficiency 0.4. By contrast, these students are likely to get similar scores if $a = 0.1$. In IRT models and NeuralCDM, the discrimination a is a scalar parameter and is usually learned through training.

Notably, although we discuss these question attributes from the perspective of cognitive diagnosis, the knowledge component and difficulty are also considered by BKT models.

Interaction function. Based on the question attributes and students' knowledge states, the interaction function aims to capture the relation between students and questions and outputs the final score (or the probability of answering correctly). In this paper, we choose two interaction functions in cognitive diagnosis, IRT and NeuralCDM, which fall under the ability level paradigm and cognition level paradigm respectively.

1) IRT: We consider the IRT model with question difficulty and discrimination factors. When calculating the probability of student s_n correctly answering question q_m, the interaction function is formulated as follows:

$$\hat{y}_{n,m} = \sigma(a_m(\theta_n - \gamma_m)) = \frac{1}{1 + e^{-a_m(\theta_n - \gamma_m)}}, \tag{8}$$

where a_m and γ_m denote the discrimination and difficulty of question q_m respectively; these are the parameters estimated through training. θ_n is the ability of student s_n and is estimated with reference to the student's response records.

2) NeuralCDM: The interaction function of NeuralCDM considers KC, KC difficulty, and question discrimination, and is learned via neural network. The probability of s_n correctly answering q_m is calculated as follows:

$$\boldsymbol{x}_{in} = \boldsymbol{Q}_m \circ (\tilde{\boldsymbol{\theta}}_n - \boldsymbol{\beta}_m) \times a_m, \tag{9}$$

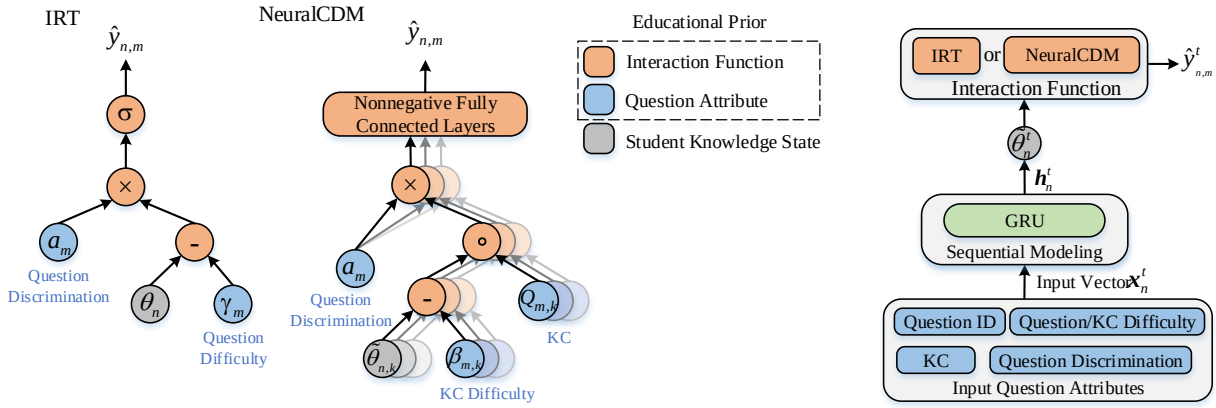
$$\boldsymbol{f}_1 = \phi(\boldsymbol{W}_1 \boldsymbol{x}_{in}^T + \boldsymbol{b}_1), \tag{10}$$

$$\boldsymbol{f}_2 = \phi(\boldsymbol{W}_2 \boldsymbol{f}_1 + \boldsymbol{b}_2), \tag{11}$$

$$\hat{y}_{n,m} = \sigma(\boldsymbol{W}_3 \boldsymbol{f}_2 + b_3), \tag{12}$$

where $\boldsymbol{Q}_m$ is the m-th row of Q-matrix $\boldsymbol{Q}$, and ϕ is the activation function (here we use $\tanh$). $\boldsymbol{\beta}_m$ and a_m are learnable parameters denoting the KC difficulties and discrimination of question q_m. $\tilde{\boldsymbol{\theta}}_n$ denotes student s_n's mastery of each KC, and is estimated with reference to the student's response records. $\boldsymbol{W}_*, \boldsymbol{b}_*$ are learnable parameters, and all elements in $\boldsymbol{W}_*$ are constrained to be nonnegative in accordance with the monotonicity assumption in [16].

Other priors. Some other educational priors have been considered in previous research and can be integrated into



(a) IRT and NeuralCDM models.

(b) Dynamic cognitive diagnosis.

Fig. 3. (a) Educational priors (i.e., question attributes and interaction functions) from cognitive diagnostic models; and (b) our dynamic cognitive diagnosis framework. The question attributes and interaction functions are colored blue and orange respectively. The operations in IRT/NeuralCDM (e.g., $-, \times, \sigma$ in IRT) form the interaction function (e.g., $\hat{y}_{n,m} = \sigma(a_m(\theta_n - \gamma_m))$ for IRT).

dynamic cognitive diagnosis. For example, guess and slip factors, which respectively indicate the probability of making correct guesses and making mistakes, are considered in some cognitive diagnosis models [13], [46] and BKT [7]. In [26], the space of practice and historical practice frequency are considered and show impacts on student knowledge states. KC relations, such as KC topologies [19], prerequisite constraints [10], and graph structure [47], have been proven beneficial for improving prediction accuracy in both BKT and deep knowledge tracing. Some researchers have also investigated the relationship between students' detailed learning activities and their knowledge states. For example, multiple attempts at a question, question type, use of hints, and interaction types are all considered relevant to the knowledge states of a student [20], [24]. We leave these priors for future exploration.

Dynamic cognitive diagnosis framework. The question attributes and interaction functions of IRT and NeuralCDM are presented in Fig. 3(a), while Fig. 3(b) illustrates the structure of the dynamic cognitive diagnosis model that integrates the educational priors into knowledge tracing. The overall procedure has four steps.

Step 1: Construct the input of the sequential modeling module. First, choose the educational priors of the question that s_n answered at time t and get the embedding $\boldsymbol{q}_n^t$. In this paper, we use only question attributes as examples. Depending on the attribute features we select, the embedding of q_n^t takes different forms. Here are some example combinations.

- Question ID: When using only question ID as the input feature, the $\boldsymbol{q}_n^t$ in Eq. (13) is the one-hot vector of question ID ($\boldsymbol{I}_n^{t,q}$). A one-hot vector of question ID is an M-dimensional indicator vector, of which only the element corresponding to the question ID is 1, and all other elements are 0. For example, if the input question is the third question (question ID=3) in the dataset, then $\boldsymbol{q}_n^t = \boldsymbol{I}_n^{t,q} = [0, 0, 1, 0, \ldots, 0]$.
- KC: When using KC ID as the input feature, the $\boldsymbol{q}_n^t$ in Eq. (13) is the multi-hot vector of the KCs contained by question ($\boldsymbol{I}_n^{t,kc}$). A multi-hot vector of the KCs is a K-dimensional indicator vector, of which only the elements corresponding to the KCs (one or more) are set to 1, while all other elements are 0. For example, if the input question contains the first and third KCs (KC IDs=1,3), then $\boldsymbol{q}_n^t = \boldsymbol{I}_n^{t,kc} = [1, 0, 1, 0. \ldots, 0]$.
- KC + question difficulty: When both KC IDs and the question difficulty in IRT are considered, $\boldsymbol{q}_n^t = \boldsymbol{I}_n^{t,kc} \times \gamma_n^t$.
- KC + KC difficulty: When both KC IDS and the KC difficulty in NeuralCDM are considered, $\boldsymbol{q}_n^t = \boldsymbol{I}_n^{t,kc} \circ \boldsymbol{\beta}_n^t$.
- KC + difficulty + discrimination. When discrimination is also considered, then $\boldsymbol{q}_n^t = \boldsymbol{I}_n^{t,kc} \times \gamma_n^t \times a_n^t$ or $\boldsymbol{q}_n^t = \boldsymbol{I}_n^{t,kc} \circ \boldsymbol{\beta}_n^t \times a_n^t$.

Subsequently, the input of the sequential modeling module (GRU in this paper) is calculated as follows:

$$\tilde{\boldsymbol{x}}_n^t = \begin{cases} \boldsymbol{q}_n^t \oplus \boldsymbol{0}, & \text{if } y_n^t = 0; \\ \boldsymbol{0} \oplus \boldsymbol{q}_n^t, & \text{if } y_n^t = 1. \end{cases} \tag{13}$$

$$\boldsymbol{x}_n^t = \tanh(\boldsymbol{W}_{qx} \tilde{\boldsymbol{x}}_n^t + \boldsymbol{b}_{qx}), \tag{14}$$

where $\boldsymbol{W}_{qx}$ and $\boldsymbol{b}_{qx}$ are learnable parameters. Here, two different operations to construct $\tilde{\boldsymbol{x}}_n^t$ are used for $y_n^t = 0$ and $y_n^t = 1$. This activates half of $\boldsymbol{W}_{qx}$ to transform $\boldsymbol{q}_n^t$ when $y_n^t = 0$ and activates the other half to transform $\boldsymbol{q}_n^t$ when $y_n^t = 1$, which both increases the fitting ability and benefits the model training. As a result, the calculation of Eq. (13) has different effects depending on the attribute features chosen.

- Question ID: The transformation of Eq. (14) is equivalent to learning a correct-answer embedding and an incorrect-answer embedding for each question. However, this might suffer from the data sparsity which is a common problem for on-line education platforms.
- KC: The transformation of Eq. (14) is equivalent to calculating the sum of the correct-answer (incorrect-answer) embeddings of the contained KCs.
- KC + question difficulty: The transformation of Eq. (14) is equivalent to calculating the sum of correct-answer (incorrect-answer) embeddings of the contained KCs with a unified weight.

- KC + KC difficulty: The transformation of Eq. (14) is equivalent to calculating the sum of the correct-answer (incorrect-answer) embeddings of the contained KCs with different weights for each KC.
- KC + difficulty + discrimination: The transformation of Eq. (14) is equivalent to calculating the weighted sum of the correct-answer (incorrect-answer) embeddings of the contained KCs. The weights depend on both the KCs and question discrimination.

Step 2: Get the student's latent state $\boldsymbol{h}_n^t$ through sequential modeling with Eq. (1) ˜ (4).

Step 3: Transform the latent state $\boldsymbol{h}_n^t$ to the student's explicit state θ_n^t or $\tilde{\boldsymbol{\theta}}_n^t$. Specifically, overall ability $\theta_n^t = \sigma(\boldsymbol{W}_{h\theta}\boldsymbol{h}_n^t + b_{h\theta})$, and knowledge proficiency $\tilde{\boldsymbol{\theta}}_n^t = \sigma(\boldsymbol{W}_{h\tilde{\theta}}\boldsymbol{h}_n^t + \boldsymbol{b}_{h\tilde{\theta}})$, where $\boldsymbol{W}_{**}, \boldsymbol{b}_{**}$ and b_{**} are learnable parameters.

Step 4: Predict the correctness of the student's response to the input question q_m (in experiments, the question is q_n^{t+1}) with the interaction function of IRT (Eq. (8)) or NeuralCDM (Eq. (9) ˜ (12)), in which θ_n and $\tilde{\boldsymbol{\theta}}_n$ are replaced with θ_n^t and $\tilde{\boldsymbol{\theta}}_n^t$ respectively. As a result, Eq. (8) changes to $\hat{y}_n^t = \sigma(a_n^{t+1}(\theta_n^t - \gamma_n^{t+1}))$, Eq. (9) changes to $\boldsymbol{x}_{in} = \boldsymbol{Q}_n^{t+1} \circ (\tilde{\boldsymbol{\theta}}_n^t - \boldsymbol{\beta}_n^{t+1}) \times a_n^{t+1}$, and Eq. (12) changes to $\hat{y}_n^t = \sigma(\boldsymbol{W}_3\boldsymbol{f}_2 + b_3)$.

Table II lists the combinations of question attributes and interaction functions; each of these is a variant of the dynamic cognitive diagnosis model and will be evaluated in our experiments (section IV).

TABLE II
COMBINATIONS OF QUESTION ATTRIBUTES AND INTERACTION FUNCTIONS

	Question Attribute	Interaction Function
DIRT_1	Question ID	IRT
DIRT_2	KC ID	IRT
DIRT_3	KC ID, Difficulty	IRT
DIRT_4	KC ID, Difficulty, Discrimination	IRT
DNeuralCDM_1	Question ID	NeuralCDM
DNeuralCDM_2	KC ID	NeuralCDM
DNeuralCDM_3	KC ID, Difficulty	NeuralCDM
DNeuralCDM_4	KC ID, Difficulty, Discrimination	NeuralCDM

Model Training. It should be noted that the question ID is previously given in the data as the identification of each question, and the KCs contained by each question are provided by experts. While question attribute parameters (i.e., question difficulty, KC difficulty, and question discrimination) in the models are learned from data rather than being provided directly[2]. When the input ($\boldsymbol{x}_n^t$) needs to be constructed using the latter type of question attributes, the values of these attribute vectors are unknown at first. Therefore, we design a two-stage training strategy when considering these attributes.

Stage 1: Use KC IDs only to construct question embedding $\boldsymbol{q}_n^t$ and train the model. All parameters, including neural network parameters ($\boldsymbol{W}_*, \boldsymbol{b}_*$) and question attribute parameters in the performance prediction module (i.e., IRT or NeuralCDM), are learned during this stage.

[2]In our experiments, we find that the models perform poorly if the difficulty and discrimination parameters are directly calculated with their statistical definitions. Therefore, we choose to learn the parameters with training.

Stage 2: Use the question attribute parameters learned in stage 1 to form the complete input vectors, then train the model again with the parameters in IRT or NeuralCDM fixed.

The training objective is to find the optimal parameters that maximize the predicted probability of the responses:

$$\underset{\Theta}{\text{maximize}} \hat{P}(\hat{y}_n^t = y_n^{t+1}; \Theta), \tag{15}$$

where Θ is the set of learnable parameters. As $y_n^{t+1} \in \{0, 1\}$, the task is equivalent to a binary classification problem. Therefore, the objective is equivalent to minimizing the binary cross-entropy loss:

$$\underset{\Theta}{\text{minimize}} \ -[y_n^{t+1}\log(\hat{y}_n^t) + (1 - y_n^{t+1})\log(1 - \hat{y}_n^t)]. \tag{16}$$

Through averaging the losses on all data samples, the final loss function can be obtained:

$$\mathcal{L} = -\frac{1}{N}\sum_{n=1}^{N}\frac{1}{T_n}\sum_{t=2}^{T_n}[y_n^{t+1}\log(\hat{y}_n^t) + (1 - y_n^{t+1})\log(1 - \hat{y}_n^t)]. \tag{17}$$

The model training (parameter estimation) can be conducted with normal gradient descent methods; this is currently the most widely used approach for training deep learning models. For example, stochastic gradient descent [48] and Adam optimizer [49] are two widely adopted gradient descent methods, implemented by several deep learning frameworks (e.g., PyTorch, TensorFlow).

D. Research Question 2: What effects do educational priors bring to deep knowledge tracing?

The effects of educational priors from cognitive diagnosis include giving interpretability to the student knowledge state vector (which changes the model from deep knowledge tracing to dynamic cognitive diagnosis) and improving the accuracy of prediction (e.g., predicting students' performances).

Interpretability. In this paper, we focus on the first aspect of interpretability, as interpretable student knowledge states are important for downstream applications. In essence, the interpretability of the parameters—including question attributes and the student's explicit state vector (θ_n^t or $\tilde{\boldsymbol{\theta}}_n^t$)—varies depending on the interaction functions used. There is a background assumption in IRT and NeuralCDM called the *Monotonicity Assumption*, which claims that when a student's ability (knowledge proficiency) improves, the probability of correctly answering a question would not drop (i.e., it will rise or at least remain unchanged) [11], [16]. The assumption is mathematically satisfied as follows.

1) In IRT, the partial gradient of the output $\hat{y}_{n,m}$ on θ_n is:

$$\frac{\partial \hat{y}_{n,m}}{\partial \theta_n} = \hat{y}_{n,m}(1 - \hat{y}_{n,m})a_m. \tag{18}$$

As $\hat{y}_{n,m} \in (0, 1)$, if a_m is constrained to be positive, then $\partial\hat{y}_{n,m}/\partial\theta_n > 0$; in other words, the optimization direction of θ_n during training remains the same with the changing direction of $\hat{y}_{n,m}$. For example, suppose $y_{n,m} = 1$ while the model output $\hat{y}_{n,m} = 0.2$. The optimizer should increase $\hat{y}_{n,m}$ to get closer to $y_{n,m}$ in order to decrease the loss (i.e., $\partial L/\partial\hat{y}_{n,m} > 0$). Thus $\partial L/\partial\theta_n > 0$, causing θ_n to increase.

On the other hand, $\partial\hat{y}_{n,m}/\partial\gamma_m = -\partial\hat{y}_{n,m}/\partial\theta_n$ and $\partial\hat{y}_{n,m}/\partial a_m = \hat{y}_{n,m}(1-\hat{y}_{n,m})(\theta_n - \gamma_m)$. Thus, the updating direction of b_m is opposite to the optimization direction of $\hat{y}_{n,m}$; the updating direction of a_m is the same as the optimization direction of $\hat{y}_{n,m}$ if θ_n and γ_m have correct partial order, and opposite to the optimization direction of $\hat{y}_{n,m}$ otherwise. Therefore, the values of the question attribute parameters are also learned in an explainable way that is in accordance with their definitions.

2) In NeuralCDM, the monotonicity assumption is satisfied by constraining all elements of the weights in multi-layers to be nonnegative. Similarly, we have $\partial\hat{y}_{n,m}/\partial\tilde{\theta}_{n,k} \geq 0$ (detailed gradient deviation is provided in Appendix A). Therefore, the optimization direction of $\tilde{\theta}_{n,k}$ during training is the same as the changing direction of $\hat{y}_{n,m}$. Moreover, $\partial\hat{y}_{n,m}/\partial\tilde{\theta}_{n,k}$ can be nonzero only when $Q_{m,k} = 1$; this means that for samples answering question q_m, only the dimensions corresponding to the knowledge concepts that are relevant to q_m will be updated during training. With the constraint of interaction function, the sequence modeling module is forced to learn the evolving pattern of students' explainable explicit states instead of unexplainable latent states. As for $\partial\hat{y}_{n,m}/\partial\beta_m$ and $\partial\hat{y}_{n,m}/\partial a_m$, it is easy to obtain a similar conclusion with IRT, which proves that the question attribute parameters are also learned in an explainable way.

Although the other aspect of interpretation in our definition (i.e., a clear relation between model input and students' knowledge states) is not the focus of this paper, there have been explorations in traditional research. For example, models based on BKT use a hidden Markov process to model the transformation of knowledge mastery, along with its probabilistic relations with factors such as learning rate, difficulty, and forgetting [7], [18], [24]. Most deep knowledge tracing models fit data better than BKT models, and by considering factors such as forgetting and hints [24], [26], they also reveal that these factors have something to do with student knowledge states. However, it is usually difficult to explain how these factors influence the outputs in deep learning models, as the mechanism of deep learning is insufficiently transparent. We leave this aspect of interpretation for future research.

Accuracy. If appropriate interaction functions and question attribute features are chosen, they can improve the accuracy of the diagnosed knowledge states, and consequently benefit the student performance prediction.

First, the sequential modeling module builds the relationship between students' practice history and their current knowledge states. The obtained relationship would be more precise if the module could get more information about the practice history. Moreover, the module structure also influences the accuracy of sequence modeling. Deep learning-based approaches (e.g., GRU) often outperform traditional approaches (e.g., Markov process in BKT) despite being less interpretable. We use GRU in all deep learning models in this paper, and focus on the influence of educational priors.

Second, compared to deep knowledge tracing (Eq. (5) $\sim$ (7)), the interaction functions leverage more attribute information about questions and capture more reasonable interactions between students and questions. This not only renders interpretation to student knowledge states, but also leads to better prediction of student performances.

IV. Evaluation

To illustrate the effectiveness of our methods, we conduct experiments on three real-world datasets. We first demonstrate the influence of educational priors in dynamic cognitive diagnosis models on the accuracy through the student performance prediction task and compare them with baseline knowledge tracing models. Then we conduct statistical analyses of the interpretation of the estimated parameters, including student knowledge states, question attributes, and the relationship between their estimated values.

A. Dataset description

In the experiments, we used three public real-world datasets: ASSIST2009, ASSIST2012 and KDDCup. ASSIST2009[3] and ASSIST2012[4] are datasets collected by the ASSISTments on-line tutoring system [50]. For the ASSIST2009 dataset, we chose the corrected version of the skill-builder subset, which repaired the duplication problem reported by [51]. KDDCup[5] is a dataset released by PSLC DataShop collected from Carnegie Learning's Cognitive Tutor in Algebra, and was provided as one of the development datasets in the KDD Cup 2010 competition (labeled as Bridge to Algebra 2006-2007). Notably, the "question" referred to in this paper is referred to as the "problem" in ASSIST2009 and ASSIST2012 (identified in the data with problem_id). While in KDDCup, we regard the concatenation of "problem name" and "step name" as a "question", because there may be multiple steps in a problem. For example, a problem with 4 steps would be separated into 4 different questions. Consequently, a "record" here refers to a response log in the datasets. Specifically, in ASSIST2009 and ASSIST2012, each record contains the results of a student's answer to a problem; in KDDCup, each record contains the results of a student's answer to a step of a problem.

For each dataset, we first deleted questions without a KC label and those with less than 15 responses. Next, we divided students with more than 200 responses into multiple dummy students, each with no more than 200 responses. For example, a student with 455 responses would be divided into three dummy students with 200, 200, and 55 responses respectively. This was done to avoid extremely long strings of response data, as these can be harmful to the training speed of recurrent neural networks. After that, we deleted students with less than 15 response logs. Table III presents some basic statistics about these datasets after preprocessing.[6] Finally, we opted to select 80% of the preprocessed response sequences for training, and the remaining 20% of sequences for testing.

[3]https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data/skill-builder-data-2009-2010 (last access: 2019/11/5)

[4]https://sites.google.com/site/assistmentsdata/datasets/2012-13-school-data-with-affect (last access: 2020/04/13)

[5]https://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp (last access: 2020/04/11)

[6]We also provide the statistics of the original datasets before preprocessing, and show some data samples in Appendix B.

TABLE III
DATA STATISTICS (AFTER PREPROCESSING)

Statistics	ASSIST2009	ASSIST2012	KDDCup
#Students	3,079	24,768	9,571
#Questions	17,671	32,111	129,240
#Records	271,468	2,460,433	1,816,860
#KCs	123	236	565
#KC-combinations	149	236	564
#KCs per question	1.20	1	1.31
#Records per question	15.36	76.62	14.05
#Records per KC-combination	1,821.93	10,425.56	3,221.38

B. Experimental setup

In the student performance prediction experiments, we compared the proposed dynamic cognitive diagnosis models with knowledge tracing baselines. BKT was chosen as the representative of traditional knowledge tracing models, while DKT and DKVMN are the representatives of deep-learning-based knowledge tracing. For fairness, models using extra information (e.g., knowledge graph [47], text [52]) were not compared, whereas the educational priors discussed in this work can be naturally integrated into these models.

To show the effects of educational priors, we focused on conducting comparisons with DKT. For a fair comparison, we used GRU as the sequence modeling module in DKT and our dynamic cognitive diagnosis models. All three types of prediction module (Eq. (5) $\sim$ (7)) were tested:

- Question-wise (DKT_Q): Only the question IDs were used as the input feature in Eq. (13).
- KC-wise (DKT_KC): Following the experiments in [8], [9], each question was represented by its KC ID instead of question ID (i.e., the question IDs in question-wise were replaced with KC IDs). It is notable that, in ASSIST2009, when a question contains more than one KC, its response is split into multiple logs, each containing one KC. We regarded the duplicate logs as one response and used the combination of the KCs as its new joint KC tag.
- MLP (DKT_MLP): The input of GRU was the same as the KC-wise prediction, while the output probability was produced with two full connection layers. Specifically, Eq. (7) was implemented as follows:

$$\boldsymbol{f}_3 = \boldsymbol{W}_{\vartheta 3}\boldsymbol{h}_n^t, \tag{19}$$

$$\boldsymbol{f}_4 = \boldsymbol{W}_{\vartheta 4}\boldsymbol{q}_n^{t+1}, \tag{20}$$

$$\hat{y}_{n,m}^t = \sigma(\boldsymbol{W}_{\vartheta 6}\tanh(\boldsymbol{W}_{\vartheta 5}[\boldsymbol{f}_3 \oplus \boldsymbol{f}_4] + \boldsymbol{b}_{\vartheta 5}) + b_{\vartheta 6}), \tag{21}$$

  where $\boldsymbol{W}_{\vartheta *}$ and $\boldsymbol{b}_{\vartheta *}$ are learnables parameters. $\boldsymbol{f}_3 \oplus \boldsymbol{f}_4$ means concatenating the vector $\boldsymbol{f}_3$ and $\boldsymbol{f}_4$, which is a common practice in deep learning when combining information from different vectors.

The experiments for dynamic cognitive diagnosis models were conducted with both IRT and NeuralCDM interaction functions and using different question attributes (labeled as DIRT_1 $\sim$ DIRT_4 and DNeuralCDM_1 $\sim$ DNeuralCDM_4 respectively). Table IV shows the differences of the input to the sequence modeling modular. All the models were implemented by PyTorch v1.5.0 (except BKT[7]) using Python, and Adam optimizer [49] was adopted to train the models. Experiments were run on a Linux server with four 2.0GHz Intel Xeon E5-2620 CPUs and a Tesla K20m GPU.

C. Improvements for student performance prediction

The proposed dynamic cognitive diagnosis models have the functionality of deep knowledge tracing, which is predicting students' performance. We conducted experiments on the three datasets to show that the prediction accuracy can be improved if proper educational priors are integrated. We selected area under curve (AUC) [53] and accuracy as the metrics, which are frequently adopted for classification tasks. The experimental results are shown in Table V. From the table, we can observe that BKT is outperformed by deep learning-based methods, which is in accordance with previous studies. Among the deep knowledge tracing models, when there is a non-negligible sparsity problem with the questions, models having knowledge component information as input (DKVMN, DKT_KC, and DKT_MLP) have better performance than DKT_Q, which is unaware of the knowledge components. This can be observed on the ASSIST2009 and KDDCup datasets, of which #Records per question is small (Table X). The reason is that the only information DKT_Q gets about the questions is question ID, and it is difficult to learn appropriate representations for those questions with sparse response logs. By contrast, in ASSIST2012, the sparsity problem is much less severe, causing DKT_Q and DKT_MLP to perform better than DKT_KC.

Through further observation, we can draw the following conclusions:

- The impact of question attributes. From DIRT_1 to DIRT_4 (also from DNeuralCDM_1 to DNeuralCDM_4), the number of question attributes that are input into the sequential modeling module gradually increases. As a result, the prediction performances continually improve from DIRT_1 to DIRT_4 (from DNeuralCDM_1 to DNeuralCDM_4). This indicates that all the educational priors that have been integrated (i.e., KC ID, difficulty, and discrimination) have a positive impact on the sequential modeling of students' knowledge state evolution.
- The impact of interaction functions. Compared to DIRT_x, DNeuralCDM_x (x=1,2,3,4) performs much better. This is because, in DNeuralCDM, a student's knowledge state is represented with a KC-wise proficiency vector rather than an overall ability (as in DIRT), and the interaction between the student knowledge state and question attributes is also better fitted with the neural network. A similar conclusion can be drawn if we compare DNeuralCDM_2 with DKT_MLP or DKT_KC. These models get the same input for the sequential modeling module; the difference lies in the performance prediction module. It is apparent that an appropriate interaction function from cognitive diagnosis is superior to simply applying multi-layer for prediction.
- Overall, the DNeuralCDM models (DNeuralCDM_x, x=2,3,4) perform better than the baselines, indicating

[7]https://iedms.github.io/standard-bkt/

TABLE IV
QUESTION ATTRIBUTES FOR SEQUENTIAL MODELING MODULAR

Model	Question ID	KC ID	Difficulty	Discrimination
BKT	✗	✓	✗	✗
DKVMN	✗	✓	✗	✗
DKT_Q	✓	✗	✗	✗
DKT_KC	✗	✓	✗	✗
DKT_MLP	✗	✓	✗	✗
DIRT_1	✓	✗	✗	✗
DIRT_2	✗	✓	✗	✗
DIRT_3	✗	✓	✓ (Question)	✗
DIRT_4	✗	✓	✓ (Question)	✓
DNeuralCDM_1	✓	✗	✗	✗
DNeuralCDM_2	✗	✓	✗	✗
DNeuralCDM_3	✗	✓	✓ (KC)	✗
DNeuralCDM_4	✗	✓	✓ (KC)	✓

TABLE V
EXPERIMENTAL RESULTS OF STUDENT PERFORMANCE PREDICTION

	ASSIST2009		ASSIST2012		KDDCup	
Model	AUC	Accuracy	AUC	Accuracy	AUC	Accuracy
BKT	0.7061	0.7045	0.6781	0.7212	0.7353	0.8351
DKVMN	0.7483	0.7293	0.7231	0.7354	0.7913	0.8440
DKT_Q	0.6960	0.6807	0.7425	0.7387	0.7484	0.8249
DKT_KC	0.7648	0.7401	0.7336	0.7408	0.7918	0.8478
DKT_MLP	0.7658	0.7366	0.7662	0.7527	0.7852	0.8435
DIRT_1	0.7067	0.6941	0.7206	0.7251	0.7406	0.8334
DIRT_2	0.7427	0.7223	0.7501	0.7421	0.7538	0.8353
DIRT_3	0.7430	0.7225	0.7509	0.7444	0.7533	0.8373
DIRT_4	0.7442	0.7238	0.7510	0.7444	0.7540	0.8375
DNeuralCDM_1	0.7006	0.6915	0.7578	0.7490	0.7710	0.8421
DNeuralCDM_2	0.7780	0.7436	0.7771	0.7590	0.8050	0.8504
DNeuralCDM_3	0.7813	0.7460	**0.7780**	0.7591	0.8100	0.8523
DNeuralCDM_4	**0.7838**	**0.7471**	0.7778	**0.7597**	**0.8114**	**0.8525**

the positive effects of properly integrated educational priors. Among the priors investigated herein, the KC assignment and interaction function (or more specifically, NeuralCDM) bring the most performance gain.

D. Analysis of interpretation

We evaluate the interpretability of dynamic cognitive diagnosis models from three aspects: interpretation of students' knowledge states (i.e, θ_n^t or $\tilde{\boldsymbol{\theta}}_n^t$), interpretation of question attribute parameters (i.e., $a, \gamma, \boldsymbol{\beta}$), and the relation between students' knowledge states and question attribute parameters.

Interpretation of students' knowledge states. Following [16], we check whether or not the results of our models are consistent with the empirical observations that students with better performance have higher diagnosed abilities/proficiencies. For example, suppose that students n_1 and n_2 answered the question q_m, and the response results are correct and incorrect respectively. Under this circumstance, the diagnosed abilities should have the relation $\theta_{n_1} > \theta_{n_2}$ (in DIRT) or $\tilde{\theta}_{n_1,k} > \tilde{\theta}_{n_2,k}, \forall k \in \mathrm{KC}(q_m)$ (in DNeuralCDM), where $\mathrm{KC}(q_m)$ is the set of KCs that are contained in q_m. When this relation is satisfied, we say that this pair of samples ((n_1, n_2) and (n_2, q_m)) aligns with the empirical observation.[8] To numerically capture how well the models align with the empirical observations, an intuitive way is to calculate the percentage of sample pairs that align with the empirical observations. Therefore, we adopt an adapted measure of Degree of Agreement (DOA). DOA was originally proposed in [54] to evaluate the scoring algorithm in recommender systems, and is defined as the percentage of item pairs aligning with the empirical observations. An item pair aligning with empirical observation means that the item preferred by the user is ranked higher by the scoring algorithm than another item that is not preferred by the user. In this study, DOA is defined as the percentage of sample pairs aligning with empirical observations. As discussed above, a sample pair that aligns with empirical observation means that the student (n_1) who performs better on question q_m has a higher diagnosed ability (θ_{n_1}) or proficiency ($\tilde{\theta}_{n_1,k}, \forall k \in \mathrm{KC}(q_m)$) than another student ($n_2$) with poorer performance on q_m. It should be noted that when the question contains multiple KCs (i.e., $\mathrm{KC}(q_m) > 1$), all proficiencies should satisfy this relation (i.e., $\tilde{\theta}_{n_1,1} > \tilde{\theta}_{n_2,1}, \tilde{\theta}_{n_1,2} > \tilde{\theta}_{n_2,2}$, ...). For detailed formulas, please refer to Appendix C.

We take DKT_KC, which uses the same sequential modeling module as DIRT and DNeuralCDM models but without priors, as the baseline for comparison. The DOAs of the models are listed in Table VI. Generally, the high DOAs in the table reveal that the models have a strong tendency to assign a higher diagnostic value (ability, proficiency or probability) after receiving a correct response record and a lower value after receiving an incorrect response record.

[8]It should be noted that this relation is consistent with the monotonicity assumption but we still need to evaluate it as there is a fitting process of student performance and model output. The fitting abilities of the models impact a lot to what degree this relation is actually satisfied.

TABLE VI
DOA OF STUDENTS' KNOWLEDGE STATES

Model	ASSIST2009	ASSIST2012	KDDCup
DKT_KC	0.8413	0.9078	0.9140
DIRT_1	0.6736	0.8077	0.5750
DIRT_2	0.8654	0.8929	0.7758
DIRT_3	0.8564	0.8876	0.7264
DIRT_4	0.8570	0.9033	0.7363
DNeuralCDM_1	0.7302	0.8763	0.7093
DNeuralCDM_2	0.8473	0.9285	0.8763
DNeuralCDM_3	0.7938	0.9294	0.8655
DNeuralCDM_4	0.8347	0.9235	0.8698

The DOA of DNeuralCDM_1 is significantly lower than that of the other DNeuralCDM models because of the lack of knowledge component information in the inputs. Moreover, the DOAs of the DNeuralCDM models are lower than DKT_KC because the questions contain multiple KCs (except in dataset ASSIST2012, where every question contains one KC). To prove this, we separately calculate the DOAs of questions that contain only one KC and denote it as $\mathrm{DOA}^{\mathrm{single}}$. As shown in Table VII, we can see an apparent increase of $\mathrm{DOA}^{\mathrm{single}}$ compared with DOA, which suggests low DOAs of questions with multiple KCs. The reason is that the assumption behind Eq. (29a) is strict. When a question contains multiple KCs, $\prod_{k\in \mathrm{KC}(q_m)} \delta(\tilde{\theta}^{t_{n_1,m}}_{n_1,k}, \tilde{\theta}^{t_{n_2,m}}_{n_2,k})$ in Eq. (29a) equals 1 only if $\tilde{\theta}^{t_{n_1,m}}_{n_1,k} > \tilde{\theta}^{t_{n_2,m}}_{n_2,k}, \forall k \in \mathrm{KC}(q_m)$. That is, student s_{n_1} (who answered correctly) needs to master all relevant KCs better than another student s_{n_2} (who answered incorrectly), which is in fact unnecessary. For example, suppose question q_m contains KCs kc_1 and kc_2 with difficulty 0.4 and 0.7 respectively. s_{n_1} masters kc_1 and kc_2 with proficiency 0.5 and 0.8, which satisfies all requirements of the question. On the other hand, s_{n2} masters kc_1 and kc_2 with proficiency 0.55 and 0.6 (<0.7) and fails to answer the question. Although s_{n_1} received a higher score than s_{n_2}, s_{n_1}'s proficiency on kc_1 is lower than s_{n_2}'s. Instead, we can assume that there should be at least one KC among $\mathrm{KC}(q_m)$ that s_1 has mastered better than s_2. Therefore, we loosen the calculation of DOA and call it Partial DOA (PDOA). Compared to DOA, the only difference is that, when the question contains multiple KCs, only one proficiency needs to satisfy the relation (i.e., $\tilde{\theta}_{n_1,k} > \tilde{\theta}_{n_2,k}, \exists k \in \mathrm{KC}(q_m)$). The formula thus changes to:

$$\mathrm{PDOA}_{\tilde{\theta}} = \frac{1}{Z}\sum_{m=1}^{M}\sum_{n_1=1}^{N}\sum_{n_2=1}^{N} J(q_m, s_{n_1}, s_{n_2})\cdot \delta(y_{n_1,m}, y_{n_2,m})[1 - \prod_{k\in \mathrm{KC}(q_m)} \delta(\tilde{\theta}^{t_{n_2,m}}_{n_2,k}, \tilde{\theta}^{t_{n_1,m}}_{n_1,k})], \quad (22a)$$

$$Z = \sum_{m=1}^{M}\sum_{n_1=1}^{N}\sum_{n_2=1}^{N} J(q_m, s_{n_1}, s_{n_2})\delta(y_{n_1,m}, y_{n_2,m}). \quad (22b)$$

The PDOAs calculated upon questions with multiple KCs (denoted as $\mathrm{PDOA}^{\mathrm{multi}}$) are presented in Table VII, and we can observe an increase compared to DOA.

Fig. 4 illustrates the tracking of student knowledge states in an intuitive way. We randomly chose a student in ASSIST2009 and select the answer records related to 5 KCs that appear frequently. The upper part (5×30 grid) of the figure shows the changing knowledge proficiencies diagnosed by DNeuralCDM_2. We can observe that when the student gives a correct (incorrect) answer, DNeuralCDM_2 tends to increase (decrease) its diagnosed proficiency on a related KC. The lower part (1×30 grid) of the figure shows the changing ability diagnosed by DIRT_2. We can further observe that when the student gives a correct (incorrect) answer, DIRT_2 increases (decreases) its diagnosed overall ability.

From all the results above, we can conclude that although the student's knowledge state vector is separated from the performance prediction vector in dynamic cognitive diagnosis models (unlike DKT_KC), the interpretability of the state vector remains high. Moreover, the advantage of DNeuralCDM models is that they are able to handle the change of proficiencies for each KC, even when the question has multiple KCs. By contrast, deep knowledge tracing models cannot get the proficiencies on KCs (e.g., DKT_Q and DKT_MLP) or can only handle questions with a single KC (e.g., in DKVMN and DKT_KC, multiple KCs in a question are combined and regarded as a dummy KC).

Interpretation of question attribute parameters. The knowledge components required by the questions are provided by experts, and both discrimination and difficulty have been studied in traditional research. In classical test theory, the common measurement of question discrimination is the point biserial correlation or biserial correlation between the students' score (0 or 1) on the question and their total score on the test [11]. As for question difficulty, it has been determined by many existing works (e.g., [11], [30], [31]) that the more difficult the question is, the lower correct rate the question has. In other words, question difficulty should have a negative correlation with the question's correct rate (or a positive correlation with the incorrect rate). Therefore, the correlation between the estimated difficulty parameters and the correct rates of questions is a reasonable measurement for the interpretability of difficulty parameters.

Unfortunately, due to the sparsity problem of questions in the datasets, it is difficult to select enough questions that have plenty of response logs to calculate reliable biserial correlations or point biserial correlations (with p-value $\ll$ 0.05). Thus, in this paper, we only compare the estimated values of difficulty parameters (i.e., the values of γ in DIRT_3 and DIRT_4, or $\boldsymbol{\beta}$ in DNeuralCDM_3 and DNeuralCDM_4) with the incorrect rates of the corresponding questions. We choose Pearson correlation coefficient (PCC) [55] to measure their correlation. For questions with multiple KCs in DNeuralCDM models, we need to transform the KC difficulties to the questions so as to calculate PCC. While combining all the difficulties of the KCs contained in a question could be a reasonable approach, how exactly this combination should be performed is underexplored. Instead, we consider the highest difficulty of the contained KCs as the question difficulty, because it is most representative of how difficult the question is, and the #KCs per question is close to 1.0 on all datasets (Table X). The results are shown in Table VIII, where p-value $\ll$ 0.05. In the table, we can observe strong correlations between the estimated question difficulties and incorrect rates.

TABLE VII
DOA (PDOA) FOR SINGLE AND MULTIPLE KC STATES

Model	ASSIST2009		ASSIST2012		KDDCup	
	$\text{DOA}^{\text{single}}$	$\text{PDOA}^{\text{multi}}$	$\text{DOA}^{\text{single}}$	$\text{PDOA}^{\text{multi}}$	$\text{DOA}^{\text{single}}$	$\text{PDOA}^{\text{multi}}$
DNeuralCDM_1	0.7648	0.8831	0.8763	-	0.7663	0.8023
DNeuralCDM_2	0.8749	0.9522	0.9285	-	0.9241	0.8782
DNeuralCDM_3	0.8435	0.9185	0.9294	-	0.9230	0.8866
DNeuralCDM_4	0.8689	0.9464	0.9235	-	0.9253	0.8823

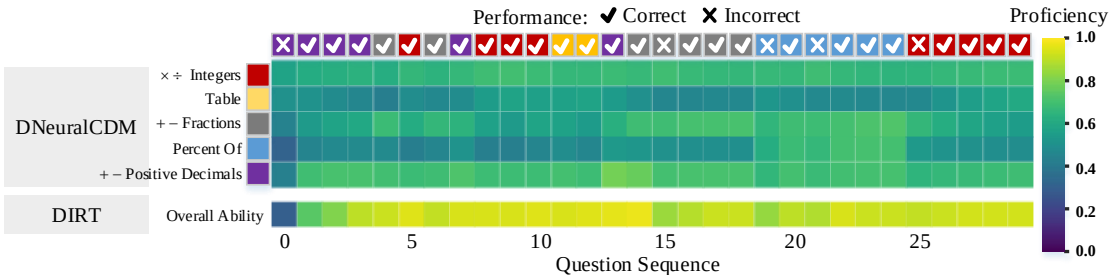


Fig. 4. A case of tracking a student's knowledge state. The leftmost column shows the KCs (with colored squares) or overall ability. The top line records performance on 30 questions, the colors of which denotes related KCs. The upper 1×30 is the diagnosed results of DNeuralCDM, and the lower 1×30 grid is the diagnosed results of DIRT.

TABLE VIII
PCC OF QUESTION DIFFICULTY PARAMETER

Model	ASSIST2009	ASSIST2012	KDDCup
DIRT_1	0.8350	0.9136	0.8903
DIRT_2	0.8152	0.9281	0.8912
DNeuralCDM_1	0.7329	0.8316	0.6049
DNeuralCDM_2	0.7958	0.8358	0.7033

p-value $\ll 0.05$

Even on ASSIST2009 and KDDCup, where the results are influenced by questions with multiple KCs, the PCCs are still high enough to show strong correlations.

Relation between student knowledge states and question attributes. Different from the interpretation of student knowledge states and question attributes, which need to be evaluated by calculating the metrics among estimated parameters and students' response data, the relation between student knowledge states and question attributes can be analytically derived once the model has been trained. For ease of understanding, we show this relation by presenting some representative examples that illustrate how the output probability of a correct response changes with student knowledge states and question attributes (difficulty and discrimination). As this relation is apparent in DIRT (a logistic-like function), we only discuss DNeuralCDM models below. Without losing generality, the following experiments are all conducted with DNeuralCDM_2 on the ASSIST2009 dataset.

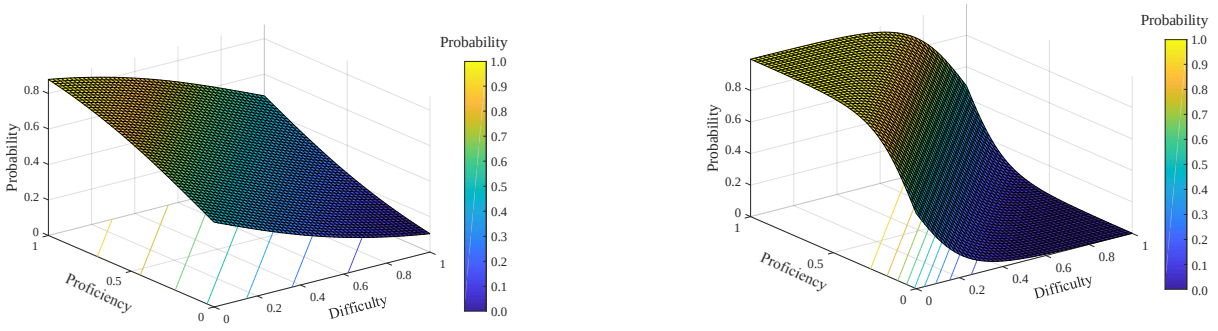
First, we show the relation among probability, proficiency, and KC difficulty in Fig. 5. Specifically, we simulate a question with one KC kc_k (randomly selected) and uniformly sample the proficiency and difficulty of kc_k in the range [0,1] at 0.02 intervals. We first set the discrimination to a fixed value, and then feed the student proficiencies and question attributes into the performance prediction module (i.e., NeuralCDM) to get the probability. From Fig. 5, we can observe that the probability increases with proficiency and decreases with difficulty, which is in line with expectations. Furthermore, if we change the discrimination (i.e., 0.2 and 0.8), we can observe from Fig. 5(a) and Fig. 5(b) that the effect of discrimination is to control the slope of the curved plane: a higher value of discrimination makes the probability more sensitive to the difference between proficiency and difficulty when the values of proficiency and difficulty are close to each other.

Next, we demonstrate how knowledge proficiency influences the probabilities for questions with single or multiple KCs in Fig. 6. In Fig. 6(a), we fix both discrimination and difficulty to 0.5, then randomly choose 10 KCs from data. By feeding proficiencies sampled from the range [0,1] at intervals of 0.02 into NeuralCDM, we get the probabilities of giving correct responses. As illustrated in the figure, the changing patterns of different KCs are not the same. In Fig. 6(b), we fix both discrimination and difficulty to 0.5, and randomly choose two KCs that appear simultaneously in one question in the data (*Conversion of Fraction Decimals Percents* and *Subtraction Whole Numbers*). The sampling of the two proficiencies is done in the range [0,1] at 0.02 intervals. We can observe that the weights of the KCs are different in the question.

Compared to deep knowledge tracing, the relation between student knowledge states and question attributes in dynamic cognitive diagnosis is unique, which separates student performance prediction and knowledge state tracing. By contrast, in deep knowledge tracing, DKT_Q and DKT_MLP are unable to trace the evolution of knowledge proficiency. Although DKT_KC offers some interpretability, the proficiency tracing and performance prediction are mixed, and questions with multiple KCs are not handled well.

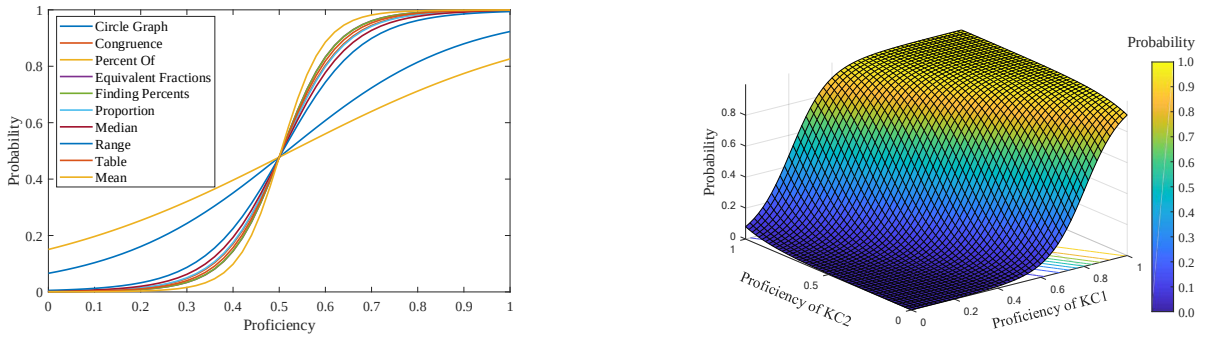
E. Training Cost

In addition to the model accuracy and interpretation, training cost is also a non-negligible factor that affects practicability. Table IX shows the training time of DKT models and our dynamic cognitive diagnosis models; here each training time



(a) Discrimination=0.2 (b) Discrimination=0.8

Fig. 5. Output probability changes with proficiency and difficulty.



(a) Single KC (b) Two KCs simutaneouly

Fig. 6. Output probability changes differently with proficiencies on single or multiple KCs.

TABLE IX
AVERAGE TRAINING TIME (SEC.) OF EACH EPOCH

Model	ASSIST2009	ASSIST2012	KDDCup
DKT_Q	83	826	629
DKT_KC	92	747	477
DKT_MLP	83	642	417
DIRT_1	75	549	603
DIRT_2	61	407	349
DIRT_3	57	490	404
DIRT_4	61	464	394
DNeuralCDM_1	57	694	920
DNeuralCDM_2	73	496	341
DNeuralCDM_3	46	518	338
DNeuralCDM_3	64	558	342

is the average over the first 5 epochs (i.e., 5 iterations of the training data). For models trained with the two-stage strategy (i.e., DIRT_3∼DIRT_4, DNeuralCDM_3∼DNeuralCDM_4), we only list the training time of the second stage, as their first stage is exactly the same as DIRT_2 and DNeuralCDM_2 respectively. From the table, we can observe that although larger data size obviously results in a longer training time, the training time of models within the same datasets does not vary significantly. Considering that no model exhibited the need for a large number of training epochs, we can conclude that the integrated educational priors do not introduce a considerable training burden.

V. DISCUSSION

Although combining deep learning-based knowledge tracing models with cognitive diagnosis methods is an intuitive thought, few investigations have been made by researchers. This paper accordingly has important implications for researchers and practitioners who are developing AI technologies for education. In this paper, we first present a summary of existing deep knowledge tracing models. Due to their use of deep learning, deep knowledge tracing models have made great progress in sequence modeling with students' learning data and predicting students' future performance (scores on questions). However, one of the weaknesses of existing deep knowledge tracing models is their insufficient ability to provide students' explicit knowledge states. Most works do not provide students' overall abilities or their mastery levels on each knowledge component. Although the predicted probability of correctly answering a question could be seen as the knowledge proficiency if we were to equate the question with the knowledge component it contains, this trick ignores the difference between questions and knowledge components, and is only applicable to questions with a single knowledge component. Moreover, while several studies have provided interpretations about their student knowledge states, these were merely based on their designing and lack support from educational theory. Accordingly, we propose to introduce educational priors from cognitive diagnosis to deep knowledge tracing in order to address their weaknesses, along with full discussions, experiments, and analyses. We review our answers to the two research questions in the following paragraphs.

Research Question 1: What educational priors can be brought to deep knowledge tracing? In this paper, we mainly study two types of educational priors: question attribute and interaction function. These two types of priors are not independent, as question attributes are typically components of inter-

action functions (e.g., difficulty and discrimination in IRT). We conduct analyses and experiments on two interaction functions (IRT and NeuralCDM) and four educational priors (question ID, knowledge component, difficulty, and discrimination). IRT and NeuralCDM are selected because their parameters can be easily tuned by gradient descent algorithms together with deep learning-based sequence modeling modules; we use them as the representation of models that fall under the ability level paradigm and cognition level paradigm respectively. It should be noted that some question attributes, such as difficulty and discrimination, are not always accessible by human labeling. Although there might be statistical definitions, these definitions might not be unified (for example, both biserial correlation and point biserial correlation can be used to calculate question discrimination [11]), and accurate statistical values rely on reasonable data distribution, which is not always accessible. Therefore, in this paper, we use a two-stage method to get these attribute values from pre-training (III-C).

The main limitation of this work is that we only study the educational priors from two cognitive diagnosis methods. In reality, there are various educational priors from educational research or traditional knowledge tracing research. Factors about questions (e.g., question type), knowledge components (e.g., knowledge topology), and student behaviors (e.g., multiple attempts, practice space, and forgetting) have been proven to be relevant to students' knowledge states. It would be excessive to investigate all priors in a single paper. However, most priors can be integrated into our dynamic cognitive diagnosis framework in a similar way to the question attributes.

Research Question 2: What effects do educational priors bring to deep knowledge tracing? The most important effect that educational priors bring is that they change deep knowledge tracing models into dynamic cognitive diagnosis models that can provide explainable student knowledge states. In cognitive diagnosis models, the properties of interaction functions enforce the interpretability of student and question parameters. For instance, the monotonicity assumption declares that (when keeping other parameters fixed) the higher a student's proficiency, the higher the probability that the student will answer correctly, and vice versa. In our proposed dynamic cognitive diagnosis models, the output of the sequential modeling module is part of the interaction function (i.e., student knowledge state), and is therefore constrained by the interaction function. Then, by integrating question attributes into the model inputs, we achieve further improvements in the accuracy of future student performance prediction. Finally, we can observe from our experimental results that different educational priors affect the model to different extents. Among the question attributes we studied, knowledge components play the most important role, especially when the questions are impacted by a data sparsity problem. As for the interaction function, NeuralCDM makes greater improvements compared to IRT, and also provides a more fine-grained diagnosis of students' knowledge states.

It should be noted that although the interpretability of machine learning models has attracted broad attention in recent years, there is currently no commonly accepted definition. The evaluation of interpretability is also a tricky problem. Basically, the analysis of model interpretability is either based on predefined model structures or post-hoc analyses. Some models are designed based on existing knowledge or empirical evidence; the model structures are usually simple and considered obviously interpretable. Examples of such models include linear regression, Bayesian network [56], IRT, and DINA. When it comes to deep learning models, post-hoc analysis is preferred due to the complex model structures. One popular type of post-host analysis is to visualize important parts of the model. For example, Liu et al. [57] visualized the results of the attention mechanism that compares similar parts of two exercise texts. Piech et al. [8] and Zhang et al. [9] visualized the clustering of questions based on their model outputs. Another popular post-hoc analysis method is to mathematically calculate or observe the influence of input features or the stimulation of network neurons. For example, Lu et al. [28] applied a layer-wise relevance propagation method to interpret RNN-based knowledge tracing models. In our experiments, we evaluate the interpretability from different aspects. Visualization is used to demonstrate the relation between student knowledge states and question attributes, as well as present a case of the evolving of a student's knowledge states. Moreover, considering the educational background of this work, we design novel metrics that measure whether the estimated values of student and question parameters are consistent with domain knowledge or human experience. The limitation of this work is that we do not pay attention to the interpretation of the sequential modeling module. In other words, we ignore what has been learned by the neural network about the relation between input response histories and model decisions (e.g., student knowledge states). We opt to use GRU because of its better fitting ability. However, some traditional approaches, such as the Markov process in BKT, are more interpretable and can sometimes outperform neural networks [58], [59]. There is usually a trade-off between accuracy and interpretability, which merits further exploration.

VI. Conclusion and Future Work

In this paper, we studied the task of dynamic cognitive diagnosis and proposed approaches to solve it by integrating educational priors into deep learning-based knowledge tracing models. Specifically, we first discussed the difference between knowledge tracing and dynamic cognitive diagnosis, along with the disadvantages of current knowledge tracing models in tracing students' explicit knowledge states. Next, we introduced educational priors, including question attributes and interaction functions from cognitive diagnosis, into deep knowledge tracing. Through extensive experiments and analyses, we showed that these priors bring high interpretability to the model parameters, thereby changing deep knowledge into dynamic cognitive diagnosis; the prediction accuracy can also be improved if proper priors are chosen.

Future research can be conducted to address the limitations of this work. First, additional educational priors (other than the question attributes and interaction functions from IRT and NeuralCDM) can be taken into consideration. Various factors have been found to be relevant to students' learning process,

such as slip, guessing, forgetting, and knowledge topology. Second, the sequential modeling module needs further improvement. Although there is usually a trade-off between accuracy and interpretability, it is valuable to explore the possibility of improving both by combining domain theories with deep learning technologies.

APPENDIX A
GRADIENT DEVIATION IN NEURALCDM

According to Eq. (9) $\sim$ (12), the partial gradient of the output $\hat{y}_{n,m}$ on the k-th dimension of θ_n (i.e., $\theta_{n,k}$) is:

$$\frac{\partial \hat{y}_{n,m}}{\partial \tilde{\theta}_{n,k}} = \frac{\partial \boldsymbol{x}_{in}}{\partial \tilde{\theta}_{n,k}} \times \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{x}_{in}} \times \frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{f}_1} \times \frac{\partial \hat{y}_{n,m}}{\partial \boldsymbol{f}_2}, \tag{23}$$

$$\frac{\partial \boldsymbol{x}_{in}}{\partial \tilde{\boldsymbol{\theta}}_{n,k}} = (0, \ldots, 0, Q_{m,k}, 0, \ldots, 0), \tag{24}$$

$$\frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{x}_{in}} = \boldsymbol{W}_1^T \times diag(\boldsymbol{f}_1 \circ (1 - \boldsymbol{f}_1)), \tag{25}$$

$$\frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{f}_1} = \boldsymbol{W}_2^T \times diag(\boldsymbol{f}_2 \circ (1 - \boldsymbol{f}_2)), \tag{26}$$

$$\frac{\partial \hat{y}_{n,m}}{\partial \boldsymbol{f}_2} = \boldsymbol{W}_3^T \times \hat{y}_{n,m}(1 - \hat{y}_{n,m}). \tag{27}$$

$Q_{m,k} \in \{0,1\}$, $\hat{y}_{n,m} \in (0,1)$, all elements in $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ lie in (0,1) and all elements in $\boldsymbol{W}_1$, $\boldsymbol{W}_2$ and $\boldsymbol{W}_3$ are nonnegative. We could easily get that all elements in $\partial \boldsymbol{x}_{in}/\partial \tilde{\theta}_{n,k}, \partial \boldsymbol{f}_1/\partial \boldsymbol{x}_{in}, \partial \boldsymbol{f}_2/\partial \boldsymbol{f}_1$ and $\partial \hat{y}_{n,m}/\partial \boldsymbol{f}_2$ are nonnegative, therefore $\partial \hat{y}_{n,m}/\partial \tilde{\theta}_{n,k} \geq 0$.

APPENDIX B
DETAILS ABOUT THE DATASETS

Table X lists some basic statistics about the original datasets used in our experiments (before preprocessing).

TABLE X
DATA STATISTICS (BEFORE PREPROCESSING)

Statistics	ASSIST2009	ASSIST2012	KDDCup
#Students	4,217	46,674	1,146
#Questions	26,688	179,999	207,856
#Records	401,756	6,123,270	3,679,199
#KCs	123	245	565
#KC-combinations	149	245	564
#KCs per question	1.13	1	1.31
#Records per question	13.00	34.02	17.70
#Records per KC-combination	1,900.03	10,735.02	3,222.48

Table XI shows the headings from three datasets, and presents one record sample for each dataset. The headings of each dataset correspond to "Record ID", "Student ID", "Question ID", "KC ID", and "Response" (which are the terms adopted in this paper) respectively (e.g., ASSIST2009 uses "user_id" to indicate Student ID).

APPENDIX C
DOA FORMULAS

As the output format of DIRT, DNeuralCDM and DKT_KC are different, the formulas of DOA are slightly different.

TABLE XI
DATA EXAMPLES

Dataset	Headings — sample
ASSIST2009	order_id, user_id, problem_id, skill_id, correct
	33022537, 64525, 51424, 1, 1
ASSIST2012	problem_log_id, user_id, problem_id, skill_id, correct
	137689528, 100009, 89710, 277, 1
KDDCup	First Transaction Time, Anon Student ID, Problem Name - Step Name, KC, Correct First Attempt
	2006-10-16 14:24:19.0, 0A63H9, LCM-C-15-18-CommonMultiple1, Identify LCM, 0

The formula for DIRT is as follows:

$$\mathrm{DOA}_\theta = \frac{1}{Z} \sum_{m=1}^{M} \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} J(q_m, s_{n_1}, s_{n_2}) \cdot \delta(y_{n_1,m}, y_{n_2,m}) \cdot \delta(\theta_{n_1}^{t_{n_1,m}}, \theta_{n_2}^{t_{n_2,m}}), \tag{28a}$$

$$Z = \sum_{m=1}^{M} \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} J(q_m, s_{n_1}, s_{n_2}) \delta(y_{n_1,m}, y_{n_2,m}); \tag{28b}$$

and for DNeuralCDM the formula is:

$$\mathrm{DOA}_{\tilde{\theta}} = \frac{1}{Z} \sum_{m=1}^{M} \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} J(q_m, s_{n_1}, s_{n_2}) \cdot \delta(y_{n_1,m}, y_{n_2,m}) \cdot \prod_{k \in \mathrm{KC}(q_m)} \delta(\tilde{\theta}_{n_1,k}^{t_{n_1,m}}, \tilde{\theta}_{n_2,k}^{t_{n_2,m}}), \tag{29a}$$

$$Z = \sum_{m=1}^{M} \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} J(q_m, s_{n_1}, s_{n_2}) \delta(y_{n_1,m}, y_{n_2,m}), \tag{29b}$$

where $J(q_m, s_{n_1}, s_{n_2}) = 1$ if both s_{n_1} and s_{n_2} answered q_m, and $J(q_m, s_{n_1}, s_{n_2}) = 0$ otherwise; $\delta(x_1, x_2) = 1$ if $x_1 > x_2$, and $\delta(x_1, x_2) = 0$ otherwise; $y_{n_1,m}$ is the correctness of s_{n_1} on q_m; $t_{n_1,m}$ is the time that s_{n_1} answered q_m.

For DKT_KC, if we regard the output probabilities of DKT_KC (i.e. $\hat{\boldsymbol{\theta}}$) as the proficiencies of corresponding knowledge components, we can calculate the DOA as follows:

$$\mathrm{DOA}_{\hat{\theta}} = \frac{1}{Z} \sum_{k=1}^{K} \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} \sum_{t_1 \in \mathrm{T}(n_1,k)} \sum_{t_2 \in \mathrm{T}(n_2,k)} \delta(y_{n_1}^{t_1}, y_{n_2}^{t_2}) \cdot \delta(\hat{\theta}_{n_1,k}^{t_1}, \hat{\theta}_{n_2,k}^{t_2}), \tag{30a}$$

$$Z = \sum_{k=1}^{K} \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} \sum_{t_1 \in \mathrm{T}(n_1,k)} \sum_{t_2 \in \mathrm{T}(n_2,k)} \delta(y_{n_1}^{t_1}, y_{n_2}^{t_2}), \tag{30b}$$

where $\mathrm{T}(n_1, k)$ is the time indexes that s_{n_1} answered questions containing KC kc_k; $y_{n_1}^{t_1}$ is the correctness of s_{n_1}'s response at time t_1; $\hat{\theta}_{n_1,k}^{t_1}$ is the value of k-th dimension of $\hat{\boldsymbol{\theta}}_{n_1}^{t_1}$ which is the output probability vector for s_{n_1} at time t_1.

APPENDIX D
ACRONYMS

Here we list the acronyms frequently used in this paper, along with their meanings.

- BKT: Bayesian knowledge tracing.
- DKT: the deep knowledge tracing model proposed in [8].
- IRT: item response theory.
- MIRT: multidimensional item response theory.
- DINA: deterministic input, noisy-and model [13].
- NeuralCD (NeuralCDM): the neural cognitive diagnosis framework (model) proposed in [16].
- DKVMN: the deep learning-based knowledge tracing model proposed in [9].
- KC: knowledge component.
- GRU: gate recurrent unit (a type of recurrent neural network).
- MLP: multi-layer perceptron.
- DOA: degree of agreement (an evaluation metric).
- PDOA: partial degree of agreement (a variant evaluation metric of DOA).

REFERENCES

[1] R. Deng, P. Benckendorff, and D. Gannaway, "Progress and new directions for teaching and learning in moocs," *Computers & Education*, vol. 129, pp. 48–60, 2019.

[2] J. Han, W. Zhao, Q. Jiang, M. Oubibi, and X. Hu, "Intelligent tutoring system trends 2006-2018: A literature review," in *2019 Eighth International Conference on Educational Innovation through Technology (EITT)*. IEEE, 2019, pp. 153–159.

[3] J. Cárdenas-Cobo, A. Puris, P. Novoa-Hernández, J. A. Galindo, and D. Benavides, "Recommender systems and scratch: An integrated approach for enhancing computer programming learning," *IEEE Transactions on Learning Technologies*, vol. 13, no. 2, pp. 387–403, 2019.

[4] A. Cano and J. D. Leonard, "Interpretable multiview early warning system adapted to underrepresented student populations," *IEEE Transactions on Learning Technologies*, vol. 12, no. 2, pp. 198–211, 2019.

[5] Q. Liu, S. Tong, C. Liu, H. Zhao, E. Chen, H. Ma, and S. Wang, "Exploiting cognitive structure for adaptive learning," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 627–635.

[6] A. Al-Hmouz, J. Shen, R. Al-Hmouz, and J. Yan, "Modeling and simulation of an adaptive neuro-fuzzy inference system (anfis) for mobile learning," *IEEE Transactions on Learning Technologies*, vol. 5, no. 3, pp. 226–237, 2012.

[7] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User modeling and user-adapted interaction*, vol. 4, no. 4, pp. 253–278, 1994.

[8] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein, "Deep knowledge tracing," in *Advances in Neural Information Processing Systems*, 2015, pp. 505–513.

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

[10] P. Chen, Y. Lu, V. W. Zheng, and Y. Pian, "Prerequisite-driven deep knowledge tracing," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 39–48.

[11] M. D. Reckase, "Multidimensional item response theory models," in *Multidimensional Item Response Theory*. Springer, 2009, pp. 79–112.

[12] S. E. Embretson and S. P. Reise, *Item response theory*. Psychology Press, 2013.

[13] J. De La Torre, "Dina model and parameter estimation: A didactic," *Journal of educational and behavioral statistics*, vol. 34, no. 1, pp. 115–130, 2009.

[14] S. M. Hartz, "A bayesian framework for the unified model for assessing cognitive abilities: Blending theory with practicality." Ph.D. dissertation, ProQuest Information & Learning, 2002.

[15] A. L. Birnbaum, "Some latent trait models and their use in inferring an examinee's ability," *Statistical theories of mental test scores*, 1968.

[16] F. Wang, Q. Liu, E. Chen, Z. Huang, Y. Chen, Y. Yin, Z. Huang, and S. Wang, "Neural cognitive diagnosis for intelligent education systems," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 6153–6161.

[17] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon, "Individualized bayesian knowledge tracing models," in *International conference on artificial intelligence in education*. Springer, 2013, pp. 171–180.

[18] M. Khajah, R. V. Lindsey, and M. C. Mozer, "How deep is knowledge tracing?" pp. 94–101, 2016.

[19] T. Käser, S. Klingler, A. G. Schwing, and M. Gross, "Beyond knowledge tracing: Modeling skill topologies with bayesian networks," in *International conference on intelligent tutoring systems*. Springer, 2014, pp. 188–198.

[20] Y. Mao, "Deep learning vs. bayesian knowledge tracing: Student models for interventions." *Journal of educational data mining*, vol. 10, no. 2, 2018.

[21] M. Khajah, R. Wing, R. Lindsey, and M. Mozer, "Integrating latent-factor and knowledge-tracing models to predict individual differences in learning," in *Educational Data Mining 2014*. Citeseer, 2014.

[22] P. I. Pavlik Jr, H. Cen, and K. R. Koedinger, "Performance factors analysis–a new alternative to knowledge tracing." *Online Submission*, 2009.

[23] J. Lee and D.-Y. Yeung, "Knowledge query network for knowledge tracing: How knowledge interacts with skills," in *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, 2019, pp. 491–500.

[24] H. Yang and L. P. Cheung, "Implicit heterogeneous features embedding in deep knowledge tracing," *Cognitive Computation*, vol. 10, no. 1, pp. 3–14, 2018.

[25] S. Sonkar, A. E. Waters, A. S. Lan, P. J. Grimaldi, and R. G. Baraniuk, "qdkt: Question-centric deep knowledge tracing," pp. 677–681, 2020.

[26] K. Nagatani, Q. Zhang, M. Sato, Y.-Y. Chen, F. Chen, and T. Ohkuma, "Augmenting knowledge tracing by considering forgetting behavior," in *The world wide web conference*, 2019, pp. 3101–3107.

[27] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (xai): Toward medical xai," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 11, pp. 4793–4813, 2020.

[28] Y. Lu, D. Wang, Q. Meng, and P. Chen, "Towards interpretable deep learning models for knowledge tracing," in *International Conference on Artificial Intelligence in Education*. Springer, 2020, pp. 185–190.

[29] R. J. Mislevy, "Foundations of a new test theory," *ETS Research Report Series*, vol. 1982, no. 2, pp. i–32, 1982.

[30] L. Crocker and J. Algina, *Introduction to classical and modern test theory*. ERIC, 1986.

[31] G. Rasch, *Probabilistic models for some intelligence and attainment tests.* ERIC, 1993.

[32] K. Pliakos, S.-H. Joo, J. Y. Park, F. Cornillie, C. Vens, and W. Van den Noortgate, "Integrating machine learning into item response theory for addressing the cold start problem in adaptive learning systems," *Computers & Education*, vol. 137, pp. 91–103, 2019.

[33] G. H. Fischer, "Derivations of the rasch model," in *Rasch models*. Springer, 1995, pp. 15–38.

[34] F. M. Lord, *Applications of item response theory to practical testing problems*. Routledge, 1980.

[35] K. K. Tatsuoka, "Rule space: An approach for dealing with misconceptions based on item response theory," *Journal of educational measurement*, vol. 20, no. 4, pp. 345–354, 1983.

[36] B. W. Junker and K. Sijtsma, "Cognitive assessment models with few assumptions, and connections with nonparametric item response theory," *Applied Psychological Measurement*, vol. 25, no. 3, pp. 258–272, 2001.

[37] E. Maris, "Estimating multiple classification latent class models," *Psychometrika*, vol. 64, no. 2, pp. 187–212, 1999.

[38] K. K. Tatsuoka, "Architecture of knowledge structures and cognitive diagnosis: A statistical pattern recognition and classification approach," *Cognitively diagnostic assessment*, pp. 327–359, 1995.

[39] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[40] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[41] H. Zhu, Q. Liu, N. J. Yuan, C. Qin, J. Li, K. Zhang, G. Zhou, F. Wei, Y. Xu, and E. Chen, "Xiaoice band: A melody and arrangement generation framework for pop music," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2837–2846.

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

[43] M. J. Anzanello and F. S. Fogliatto, "Learning curve models and applications: Literature review and research directions," *International Journal of Industrial Ergonomics*, vol. 41, no. 5, pp. 573–583, 2011.

[44] L. Averell and A. Heathcote, "The form of the forgetting curve and the fate of memories," *Journal of mathematical psychology*, vol. 55, no. 1, pp. 25–35, 2011.
[45] J. Liu, G. Xu, and Z. Ying, "Data-driven learning of q-matrix," *Applied psychological measurement*, vol. 36, no. 7, pp. 548–564, 2012.
[46] J. De La Torre, "The generalized dina model framework," *Psychometrika*, vol. 76, no. 2, pp. 179–199, 2011.
[47] H. Nakagawa, Y. Iwasawa, and Y. Matsuo, "Graph-based knowledge tracing: Modeling student proficiency using graph neural network," in *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE, 2019, pp. 156–163.
[48] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.
[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.
[50] M. Feng, N. Heffernan, and K. Koedinger, "Addressing the assessment challenge with an online system that tutors as it assesses," *User Modeling and User-Adapted Interaction*, vol. 19, no. 3, pp. 243–266, 2009.
[51] X. Xiong, S. Zhao, E. G. Van Inwegen, and J. E. Beck, "Going deeper with deep knowledge tracing." *International Educational Data Mining Society*, 2016.
[52] Q. Liu, Z. Huang, Y. Yin, E. Chen, H. Xiong, Y. Su, and G. Hu, "Ekt: Exercise-aware knowledge tracing for student performance prediction," *IEEE Transactions on Knowledge and Data Engineering*, 2019.
[53] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.
[54] A. Pirotte, J.-M. Renders, M. Saerens *et al.*, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Transactions on Knowledge & Data Engineering*, vol. 19, no. 3, pp. 355–369, 2007.
[55] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise reduction in speech processing*. Springer, 2009, pp. 1–4.
[56] S. J. Russell and P. Norvig, "Artificial intelligence: A modern approach," 2010.
[57] Q. Liu, Z. Huang, Z. Huang, C. Liu, E. Chen, Y. Su, and G. Hu, "Finding similar exercises in online education systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1821–1830.
[58] K. H. Wilson, X. Xiong, M. Khajah, R. V. Lindsey, S. Zhao, Y. Karklin, E. G. Van Inwegen, B. Han, C. Ekanadham, J. E. Beck *et al.*, "Estimating student proficiency: Deep learning is not the panacea," in *In Neural Information Processing Systems, Workshop on Machine Learning for Education*, vol. 3, 2016.
[59] T. Gervet, K. Koedinger, J. Schneider, T. Mitchell *et al.*, "When is deep learning the best approach to knowledge tracing?" *Journal of Educational Data Mining*, vol. 12, no. 3, pp. 31–54, 2020.

Fei Wang received the BE degree in computer science and technology from the University of Science and Technology of China, Hefei, China. He is currently working toward the Ph.D. degree majoring in Applied Computer Technology with the School of Computer Science and Technology, University of Science and Technology of China. His research interests include data mining and intelligent education systems.

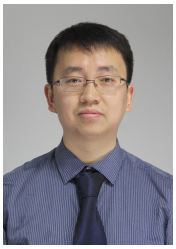
Zhenya Huang received the B.E. degree from Shandong University, in 2014 and the Ph.D. degree from USTC, in 2020. He is currently an associate researcher of the School of Computer Science and Technology, University of Science and Technology of China (USTC). His main research interests include data mining, knowledge discovery, representation learning and intelligent tutoring systems. He has published more than 30 papers in refereed journals and conference proceedings including TKDE, TOIS, KDD, AAAI. He has served regularly in the program committees of a number of conferences, and is reviewers for the leading academic journals.

Qi Liu received the Ph.D. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China. He is current a Professor with USTC. He has published prolifically in refereed journals and conference proceedings, such as, the TKDE, TOIS, TKDD, TIST, KDD, IJCAI, AAAI, ICDM, SDM and CIKM. His current research interests include data mining, machine learning, recommender systems, intelligent education systems and social network analysis. Dr. Liu was a recipient of the ICDM-2011 Best Research Paper Award, the Special Prize of President Scholarship for Postgraduate Students, Chinese Academy of Sciences (CAS), and the Distinguished Doctoral Dissertation Award of CAS. He has served regularly in the program committees of a number of conferences, and is a Reviewer for the leading academic journals in his fields. He is a member of IEEE and ACM.

Enhong Chen (SM'07) received the B.S. degree from Anhui University, Hefei, China, the M.S. degree from the Hefei University of Technology, Hefei, and the Ph.D. degree in computer science from the University of Science and Technology of China (USTC), Hefei, in 1989, 1992 and 1996 respectively. He is currently a Professor and the Executive Dean of the School of Data Science, the Vice Director of the National Engineering Laboratory for Speech and Language Information Processing, USTC. He has published lots of papers on refereed journals and conferences, such as TKDE, TIST, TMC, KDD, ICDM, NIPS and CIKM. His current research interests include data mining and machine learning, social network analysis, recommender systems and intelligent education systems. Dr. Chen was a recipient of the National Science Fund for Distinguished Young Scholars of China, the Best Application Paper Award on KDD-2008, the Best Student Paper Award on KDD-2018 (Research), and the Best Research Paper Award on ICDM-2011 and Best of SDM-2015. He is a senior member of the IEEE.

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

Jianhui Ma received the BE degree in computer science from University of Science and Technology of China (USTC), Hefei, China, in 1997 and Ph.D. degree from USTC in 2004. He is currently a lecturer of the School of Computer Science and Technology in USTC. His research interests include intelligent computing, artificial immune system, computer security and data mining.



Shijin Wang received the BE degree in electronic science and Technology from University of Science and Technology of China, Hefei, China, in 2003 and Ph.D. degree in pattern recognition & intelligent system from Institute of Automation, Chinese Academy of Science, Beijing, China. He is currently the president of IFLYTEK Beijing Research, vice president of IFLYTEK AI Research and vice leader of technology and industry working group of AIIA, Young Expert of Internet Association of China. He has published more than 30 papers in refereed conferences such as ICASSP, ACL, KDD, SIGIR and AAAI. His research interests include speech processing, natural language processing and intelligent education. He led the team that won more than ten championships in international technical evaluation such as Blizzard Challenge and CHiME.