Information Sciences 596 (2022) 567–587

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Ability boosted knowledge tracing

Sannyuya Liu [a,b,c], Jianwei Yu [b,c], Qing Li [a,c], Ruxia Liang [a,c], Yunhan Zhang [a,c], Xiaoxuan Shen [a,c,*], Jianwen Sun [a,c,*]

[a] National Engineering Laboratory for Educational Big Data, Central China Normal University, Wuhan 430079, China
[b] National Engineering Research Center for E-Learning, Central China Normal University, Wuhan 430079, China
[c] Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan 430079, China

ARTICLE INFO

Article history:
Received 11 June 2021
Received in revised form 23 February 2022
Accepted 24 February 2022
Available online 11 March 2022

Keywords:
Knowledge tracing
Learner modeling
Artificial intelligence in education
Ensemble learning
Matrix factorization
Graph neural networks

ABSTRACT

Knowledge tracing (KT) has become an increasingly relevant problem in intelligent education services, which estimates and traces the degree of learner's mastery of concepts based on students' responses to learning resources. The existing mainstream KT models, only attribute learners' feedback to the degree of knowledge mastery and leave the influence of mental ability factors out of consideration. Although ability is an essential component of the problem-solving process, these knowledge-centered models cause a contradiction between data fitting and rationalization of the model decision-making process, making it difficult to achieve high precision and readability simultaneously.

In this paper, an innovative KT model, **a**bility **b**oosted **k**nowledge **t**racing (ABKT)[1] is proposed, which introduces the ability factor into learning feedback attribution to enable the model to analyze the learning process from two perspectives, knowledge and ability, simultaneously. Based on constructive learning theory, continuous matrix factorization (CMF) model is proposed to simulate the knowledge internalization process, following the initiative growth and stationarity principles. In addition, the linear graph latent ability (LGLA) model is proposed to construct learner and item latent ability features, from graph-structured learner interaction data. Then, the knowledge and ability dual-tracing framework is constructed to integrate the knowledge and ability modules. Experimental results on four public databases indicate that the proposed methods perform better than state-of-the-art knowledge tracing algorithms in terms of prediction accuracy in quantitative assessments, displaying some advantages in model interpretability and intelligibility.

1. Introduction

With the rapid growth of big data and artificial intelligence technology, data-driven intelligent education services, such as intelligent tutoring systems (ITS), offer new opportunities to achieve personalized learning, which was once thought to be a difficult project to carry out. Accurate and comprehensive learner modeling is at the core of personalized learning technology, and knowledge tracing (KT) is a vital part of learner modeling techniques. The goal of KT is to estimate students' degree of mastery of a specific knowledge concept based on students' responses to items while tracing the development of learners' knowledge concepts. KT has become a hot research topic in intelligent education and data mining communities [16,44].

* Corresponding authors.
E-mail addresses: shenxiaoxuan@ccnu.edu.cn (X. Shen), sunjw@ccnu.edu.cn (J. Sun).

[1] Our implementations are available inhttps://github.com/ccnu-mathits/ABKT.

https://doi.org/10.1016/j.ins.2022.02.044
0020-0255/

In a general intelligent e-learning system, learners' knowledge states are estimated by analyzing the history of the feedback on questions (exercises). Furthermore, Fig. 1 presents a toy example of the exercise processes of three typical students. KT has two primary goals: state estimation and performance prediction. In state estimation, the mastery of the knowledge concepts ("line segment" and "rectangle" in Fig. 1) of each learner (students 1, 2, and 3 in Fig. 1) is estimated. In performance prediction, the goal is to predict each learner's feedback on new questions (exercises) (Q8 in Fig. 1) accurately. Therefore, KT presents two important applications in intelligent e-learning systems: first, estimating and visualizing learners' knowledge states, which can assist learners in adjusting their learning strategies, and second, recommending exercises and resources at the appropriate level for learners to improve learning efficiency.

In recent decades, the knowledge tracing algorithm, which directly determines the quality of the KT module and student modeling, has received extensive attention from researchers and achieved excellent progress. Numerous KT algorithms have been developed, of which item response theory (IRT) based KT, Bayesian knowledge tracing (BKT) and deep learning-based KT (DLKT) are the three most quintessential branches. However, in real exercise scenarios, there are huge discrepancies among different learners in terms of the learning path, initial state, inherent characteristics, and so forth, which pose enormous challenges for KT models.

In knowledge tracing, mainstream models are built based on machine-learning paradigms, by mining patterns of learners' knowledge acquisition from historical data. However, this method often fails to produce a readable and convincing result, as shown in Fig. 2. DKT [33] is one of the most advanced and widely used knowledge tracing models. However learner knowledge acquisition patterns seem irrational, as they are not consistent with general cognition and pedagogical theories. In general, it is believed that the learners' knowledge mastery process is relatively stable, will not fluctuate dramatically within a short period of time, and has certain directionality. From this perspective, many researchers have introduced purpose-built

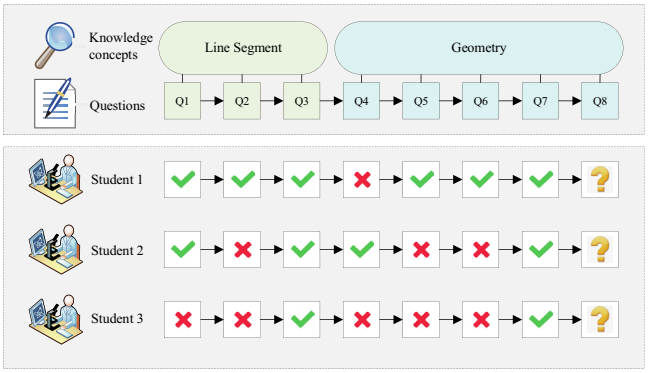

Fig. 1. A toy example of learning process.

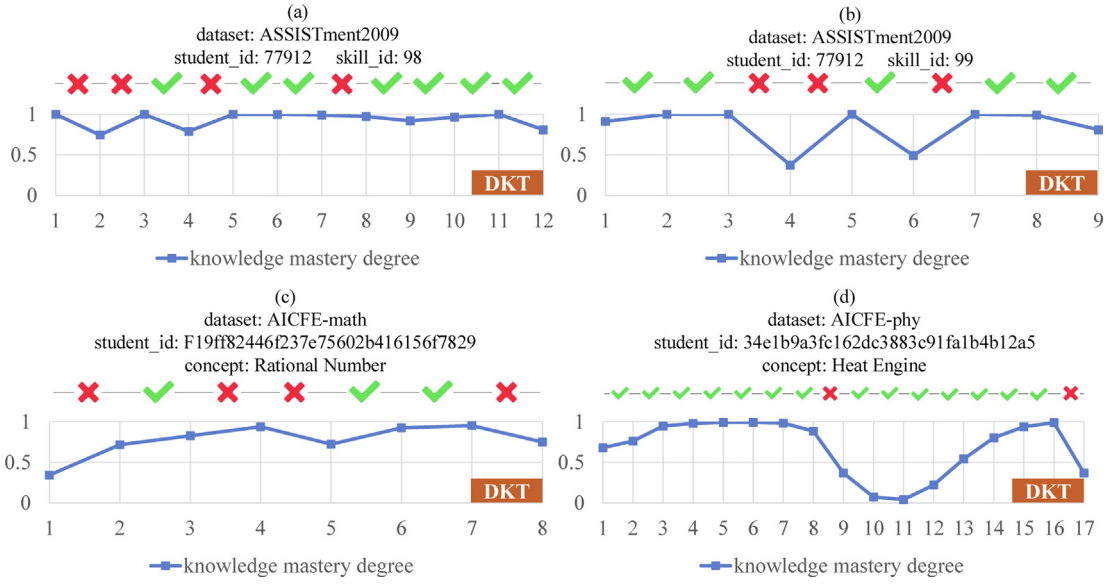

Fig. 2. Knowledge acquisition curve of specific learners generated by DKT model.

constraints [45,35,31] into KT models to alleviate this irrationality. However these measures can only alleviate the problem, not solve it, and may possibly reduce the model accuracy.

In this paper, a classical educational psychology theory of problem solving, is introduced to analyze the KT problem. Problem-solving is a complex cognitive process in which people identify problems and develop methods to resolve them. Performing exercises is a typical problem-solving process. In problem solving theory, there are two important factors that affect students' problem-solving outcomes, namely, knowledge mastery and mental ability [39]. Based on this analytical framework, the existing mainstream knowledge tracing frameworks attribute learners' feedback only to knowledge mastery, ignoring the influence of learners' ability. These knowledge-centered models cause a contradiction between data fitting and the rationalization learners' knowledge mastery curves, which makes it difficult to achieve high precision and readability simultaneously.

From the resource analysis point of view, questions linked to the same knowledge point may require different types of abilities for the solution. An example of this is presented in Fig. 3. To solve the questions (in Fig. 3), students not only need to master the knowledge concept of addition, but also need to have numeracy and comprehension ability. The existing KT models are difficult to identify and process the non-knowledge information effectively, which results in a performance bottleneck to a certain extent.

Based on the above observation and analyses, ability factors are introduced into a knowledge-centered knowledge tracing framework, to explore a possible solutions for student modeling from a new perspective, which attributes learners' feedback to two dimensions (knowledge and ability) to boost the performance and interpretability of the KT model concurrently. In this paper, the ability boosted knowledge tracing (ABKT) model is proposed. ABKT constructs a new KT framework, namely, knowledge and ability dual-tracing framework, based on the traditional knowledge-centered model, by introducing the ability factor as a supplement to learning feedback attribution. Combined with the aforementioned analysis, the major contributions of our study can be summarized as follows:

- A novel knowledge tracing model, the ability boosted knowledge tracing, namely, ABKT, is proposed. In ABKT, the ability factor is introduced as a supplement to traditional knowledge-centered models in terms of learning feedback attribution. ABKT consists of knowledge and ability modules, as well as a knowledge and ability dual-tracing framework, whereby these two modules are integrated using the ensemble learning method. This treatment allows ABKT to analyze the learning process from two perspectives (knowledge and ability) simultaneously.
- In the knowledge module, a matrix factorization-based knowledge evolution model called continuous matrix factorization (CMF) is proposed. CMF regards the changes in student's mastery of knowledge as an outcome of knowledge internalization, following the constructive learning theory. Hence, CMF simulates the knowledge internalization process according to the initiative growth and stationarity principles, which ensure the readability and rationality of the model.
- In the ability module, an efficient graph-based representation learning model, the linear graph latent ability (LGLA) is proposed. The LGLA model utilizes graph-structured interactions of the learners to construct learner and item latent ability factors. In addition, LGLA simplifies vanilla graph neural networks by linearizing the feature aggregation layer, while improving model training efficiency and usability.
- An empirical study was conducted using real-world data. The proposed ABKT method was evaluated on four real-world datasets. The results indicate that ABKT outperforms the state-of-the-art KT models. Furthermore, the proposed ABKT exhibits advantages in terms of model intelligibility and interpretability.

The remainder of this paper is organized as follows. In Section 2, research related to this study is systematically reviewed. In Section 3, the proposed ABKT model is elaborated upon. In Section 4 the optimization methods and parameter determination of the proposed method are described. In Section 5, the experimental analysis of the four public databases is presented, and the conclusions of this study are provided in Section 6.

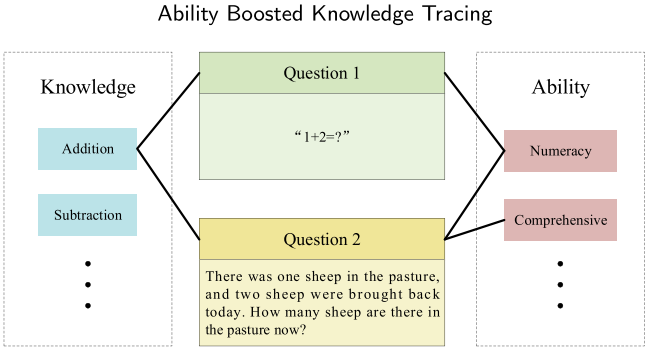

Fig. 3. Two key factors in problem solving.

2. Related work

2.1. Item response theory

Item response theory [9] is one of the most classic and widely studied KT models. In IRT, the model ascribes the elements affecting learners' feedback to subjective and objective factors, where the subjective factors denote the degree of learners' knowledge mastery [22], and the objective factors include item difficulty [13], item discrimination [7], guessing rate, slip rate [8], and the like. In the vanilla IRT model, the degree of knowledge mastery is treated as a set of static parameters. Thus, the IRT-based KT model is suitable for scenarios in which learners' knowledge remains unchanged, such as assessment. However, the process of learning is long-term and dynamic, whereby learners' knowledge levels are best modeled by time-series. Based on this fact, many researchers introduce time sequence superposition of students' learning logs to obtain time series information and propose improved models, for example, AFM [3] and PFA [32]. However, these improved models fail to capture the abundant sequence information available. Thus, numerous researchers have explored new frameworks to model the learning process dynamically.

2.2. Bayesian knowledge tracing

Bayesian knowledge tracing, which is another branch of KT modeling, is a dynamic probabilistic model constructed to simulate the learning process of learners. BKT [6] models the learners' potential knowledge state (discrete finite state) based on dynamic user interaction data, and further maintains and updates the changes in the learners' knowledge states over the learning procedure via a hidden Markov model. Based on the classical BKT model, many researchers have proposed modified models by extending the knowledge states [48], introducing forgetting factors [15], studying time features [28], and performing differential modeling of learning parameters [2,29]. In general, BKT models can effectively fit the learning process and provide good interpretability. However, the learners' knowledge states in BKT models are assumed to be discrete and finite; thus, the number of model parameters is limited, which hurts the expressive ability and capacity of the model. Therefore, it is difficult to fit learners' complex learning sequence data, using BKT.

2.3. Deep learning based knowledge tracing

Deep learning (DL) models have contributed enormously to many artificial intelligence fields [46,37], and they are becoming universal machine learning models. The DL model is found to be better at modeling complex source data because it can easily scale parameters, and readily boost model capacity. Therefore, deep learning models seem to be promising for modeling complex learning sequences, and DLKT models are considered as a possible way to solve KT problems. Deep knowledge tracing (DKT) [33] is an inchoate and representative DLKT model, which applies LSTM to fit learners' sequential feedback. DKT improves the prediction accuracy significantly compared with the IRT and BKT models by incorporating neural networks. To further improve the model performance, many variants have been proposed by introducing more advanced DL sequential models [47,36,20,18], modeling the forgetting process of learners [24], and optimizing knowledge or exercise encoding [19,4,42,21,12]. Recently, some researchers introduced graph representation learning [27,40,49] and transformer encoder [26,11,34,5,38,50] techniques to KT and further improved the model performance. Compared with traditional knowledge tracing models (IRT and BKT), DLKT models have a great advantage in terms of prediction accuracy. However, the DLKT model suffers from poor interpretability and controllability, which hinders the practical application of the model.

3. Proposed method

3.1. Problem formulation

In the formulation of the knowledge tracing task, the training data consisted of a student set ($\mathscr{U}$), question set ($\mathscr{I}$) and knowledge concept set ($\mathscr{KC}$). The student set can be denoted by $\mathscr{U} = \{u_1, u_2, \ldots, u_m\}$, where $m$represents the number of students in the e-learning system, and $\mathscr{I}$ is the question set, $\mathscr{I} = \{q_1, q_2, \ldots, q_s\}$, where $s$denotes the number of questions. The learning sequence and corresponding feedback on the questions from a specific student are in the form of time-series data. $\mathscr{S}_{u_i} = \left[\mathscr{S}^1_{u_i}, \mathscr{S}^2_{u_i}, \ldots, \mathscr{S}^l_{u_i}\right]$ represents the learning sequence of u_i; $\mathscr{S}^l_{u_i}$ refers to the l-th question u_i performed in the learning sequence and $\mathscr{S}^l_{u_i} \in \mathscr{I}$. The corresponding feedback for learning sequence $\mathscr{S}_{u_i}$ can be denoted as $\mathscr{C}_{u_i} = \left[\mathscr{C}^1_{u_i}, \mathscr{C}^2_{u_i}, .., \mathscr{C}^l_{u_i}\right]$, where $\mathscr{C} \in \{0, 1\}$. $\mathscr{C} = 1$ when the student answer the question correctly, and $\mathscr{C} = 0$ otherwise. The knowledge concept set in the system is represented by $\mathscr{KC} = \{kc_1, kc_2, \ldots, kc_n\}$, and the index matrix indicating the relationship between the questions and knowledge concepts is usually called the Q-matrix where $\mathbf{Q} \in \mathbb{R}^{s \times n}$. Further, $\mathbf{Q}_{\mathscr{S}^t_{u_i}} \in \mathbb{R}^n$ denotes the knowledge concept index of question $\mathscr{S}^t_{u_i}$. The goals of the typical KT problem are to estimate the knowledge state of each student on $\mathscr{KC}$ over the entire learning process $\mathscr{S}_{u_i}$ to predict the feedback of a given student u_i on question q_j, based on the student's historical learning sequence and feedback ($\mathscr{S}_{u_i}$ and $\mathscr{C}_{u_i}$). The accompanying notations are listed Table 1.

Table 1
Notations in the paper.

Symbol	Description
$\mathscr{U} = \{u_1, u_2, \ldots, u_m\}$	Set of student(s), m is the number of students
$\mathscr{I} = \{q_1, q_2, \ldots, q_s\}$	Set of question(s), s is the number of questions
$\mathscr{KC} = \{kc_1, kc_2, \ldots, kc_n\}$	Set of knowledge concept(s), n is the number of concepts
$\mathscr{S}_{u_i} = \left[\mathscr{S}^1_{u_i}, \mathscr{S}^2_{u_i}, .., \mathscr{S}^l_{u_i}\right]$	Learning sequence (questions) of u_i, l is the length of sequence
$\mathscr{C}_{u_i} = \left[\mathscr{C}^1_{u_i}, \mathscr{C}^2_{u_i}, .., \mathscr{C}^l_{u_i}\right]$	Correctness sequence of $\mathscr{S}_{u_i}$
$\mathbf{Q} \in \mathbb{R}^{s\times n}$	Index matrix between questions and knowledge concepts
$\mathbf{Q}_{\mathscr{S}^t_{u_i}} \in \mathbb{R}^n$	Knowledge concept index of question $\mathscr{S}^t_{u_i}$
$\mathbf{K}^t_{u_i} \in \mathbb{R}^n$	Knowledge mastery degree for each knowledge concept of student u_i in time t
$\mathbf{I}^K_{u_i}, \mathbf{I}^K_{\mathscr{S}^t_{u_i}} \in \mathbb{R}^{n\times k_K}$	Knowledge internalization preference vectors of student u_i and question $\mathscr{S}^t_{u_i}$ for each knowledge concept
$\mathbf{A} \in \mathbb{R}^{(m+s)\times(m+s)}$	Adjacency matrix of student-question interaction graph
$\mathbf{I}^A_{u_i}, \mathbf{I}^A_{\mathscr{S}^t_{u_i}} \in \mathbb{R}^{k_A}$	Ability preference vectors of student u_i and question $\mathscr{S}^t_{u_i}$
$g^A_{u_i}, g^A_{\mathscr{S}^t_{u_i}}$	Global feature of student u_i and question $\mathscr{S}^t_{u_i}$ in ability model
$\mathrm{P}\left(u_i, \mathscr{S}^t_{u_i}\right), \mathrm{P_K}\left(u_i, \mathscr{S}^t_{u_i}\right), \mathrm{P_A}\left(u_i, \mathscr{S}^t_{u_i}\right)$	Probability that student u_i can solve question $\mathscr{S}^t_{u_i}$ on perspectives of entirety, knowledge and ability, respectively

3.2. Outline of ABKT

The proposed model, ABKT, has three primary modules, as shown in Fig. 4, consisting of the knowledge module (continuous matrix factorization model, CMF), ability module (linear graph latent ability model, LGLA), and the knowledge and ability dual tracing framework. First, the CMF and LGLA models are constructed to model the knowledge and ability states during the learning process, respectively, where CMF and LGLA are two independent and parallel models. Then, the knowledge and ability dual-tracing framework predict the feedback based on the states of students' knowledge and ability, where the structure of the knowledge and ability dual tracing framework is inspired by the action mechanism of knowledge and ability in problem-solving theory. In the subsequent sections, the details and connections among these three modules are elaborated upon, in Sections 3.3 (knowledge module), 3.4 (ability module) and 3.5 (knowledge and ability dual tracing framework).

3.3. Knowledge module: continuous matrix factorization model

In ABKT, a continuous matrix factorization model is proposed to analyze the learning process from the perspective of knowledge. To build a more interpretable knowledge module, In this study, the CMF model was built based on the classic learning theory of constructivism [10], which is one of the most prestigious pedagogical theories. According to the constructivist learning theory, variations in the degree of knowledge mastery are the result of knowledge internalization. Based on this assumption, the core of the knowledge module is to model the way the students internalize specific learning tasks.

In CMF, $\mathbf{K}^t_{u_i}$ denotes the knowledge mastery degree of student u_i at time t, where $K^t_{u_i} \in \mathbb{R}^n$. The knowledge internalization (increment of the knowledge mastery degree) of student u_i in the learning procedure at time t can be recognized as $\Delta K^t_{u_i}$. Based on the assumption of constructivism learning theory, the recursion function of $K^t_{u_i}$ can be written as

$$K^t_{u_i} = K^{t-1}_{u_i} + \Delta K^t_{u_i} \tag{1}$$

Specially, $K^0_{u_i}$ denotes the initial knowledge mastery degree of student u_i.

Knowledge internalization $\Delta K^t_{u_i}$ is a consequence of student u_i interacting with question $\mathscr{S}^{t-1}_{u_i}$. The latent factor model was introduced to estimate $\Delta K^t_{u_i}$. $\mathbf{I}^K_{u_i}$ is a matrix composed of the knowledge internalization preference vectors of student u_i for all knowledge concepts, where $\mathbf{I}^K_{u_i} \in \mathbb{R}^{n\times k_K}$. $\mathbf{I}^K_{\mathscr{S}^t_{u_i}}$ denotes the corresponding knowledge internalization feature matrix of question $\mathscr{S}^t_{u_i}$, where $\mathbf{I}^K_{\mathscr{S}^t_{u_i}} \in \mathbb{R}^{n\times k_K}$. $\mathbf{I}^K_{u_i}$ and $\mathbf{I}^K_{\mathscr{S}^t_{u_i}}$ are a pair of feature-aligned matrices, and k_K denotes the feature dimension, which is set artificially. The feature vectors in $\mathbf{I}^K_{u_i}$ and $\mathbf{I}^K_{\mathscr{S}^t_{u_i}}$ are independent of the different knowledge concepts. $\Delta K^t_{u_i}$ can be represented as

$$\Delta K^t_{u_i} = \mathrm{SUM}\left(\mathbf{I}^K_{u_i} \circ \mathbf{I}^K_{\mathscr{S}^{t-1}_{u_i}}\right) \circ \mathbf{Q}_{\mathscr{S}^{t-1}_{u_i}} \tag{2}$$

where $\circ$ denotes the Hadamard product between matrixes.

Furthermore, knowledge internalization is usually considered to have a positive influence on learners. Thus, a non-negativity restriction is applied to $\Delta K^t_{u_i}$, where the restricted knowledge internalization can be expressed as,

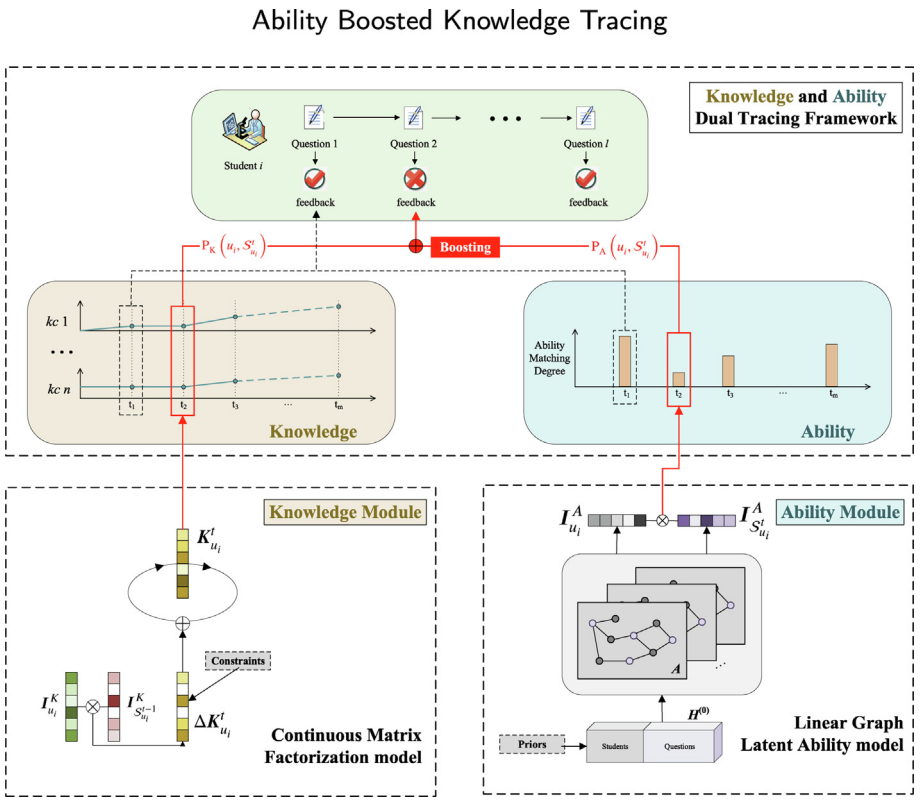

Fig. 4. Outline of ABKT.

$$\Delta K_{u_i}^{t} = \mathrm{ReLU}\left(\mathrm{SUM}\left(\mathbf{I}_{u_i}^{K} \circ \mathbf{I}_{\mathscr{S}_{u_i}^{t-1}}^{K}\right) \circ \mathbf{Q}_{\mathscr{S}_{u_i}^{t-1}}\right) \tag{3}$$

ReLU denotes a rectified linear unit [25], and ReLU(x)=max(0, x) is one of the most widely used activation functions in modern neural networks. In ABKT, the ReLU is introduced as a non-negative constrained operator.

The output of CMF, $\mathrm{P_K}\left(u_i, S_{u_i}^{t}\right)$ denotes the probability that student u_i can solve question $\mathscr{S}_{u_i}^{t}$ from the perspective of knowledge, as modeled by the parameterized Rasch model, where

$$\mathrm{P_K}\left(u_i, \mathscr{S}_{u_i}^{t}\right) = \gamma + \frac{1-\gamma}{1+\exp\left(-d \cdot \frac{\mathrm{SUM}\left(\mathbf{K}_{u_i}^{t} - \mathbf{D}_{\mathscr{S}_{u_i}^{t-1}} \circ \mathbf{Q}_{\mathscr{S}_{u_i}^{t-1}}\right)}{\mathrm{SUM}\left(\mathbf{Q}_{\mathscr{S}_{u_i}^{t-1}}\right)}\right)} \tag{4}$$

where γ means surmise, which is a constant associated with a given dataset, usually set to 0 or 0.25. d is a constant in the Rasch model, equal to 1.702. The coefficient of difficulty of $\mathscr{S}_{u_i}^{t}$ over *KC* can be indicated as $\mathbf{D}_{\mathscr{S}_{u_i}^{t-1}}$, where $\mathbf{D}_{\mathscr{S}_{u_i}^{t-1}} \in \mathbb{R}^{n}$ is the set of trainable parameters. Considering that some questions correspond to multiple knowledge concepts, the average degree of these knowledge concepts is taken as the degree of knowledge mastery of learners on the question.

In this section, the knowledge model is described in detail by introducing the computing procedure $\mathrm{P_K}\left(u_i, \mathscr{S}_{u_i}^{t}\right)$. This calculation process can be extended to all training data. In the next section, these ideas are illustrated prior to constructing the ability model. Fig. 5.

3.4. Ability module: linear graph latent ability model

In the traditional knowledge tracing model, students' degree of knowledge mastery is treated as the only factor that affects the exercises. However, ability is also a determining factor which has been ignored by existing knowledge tracing models. Many abilities, such as spatial imagination, abstract thinking and so forth, greatly affect learners' performance in problem solving.

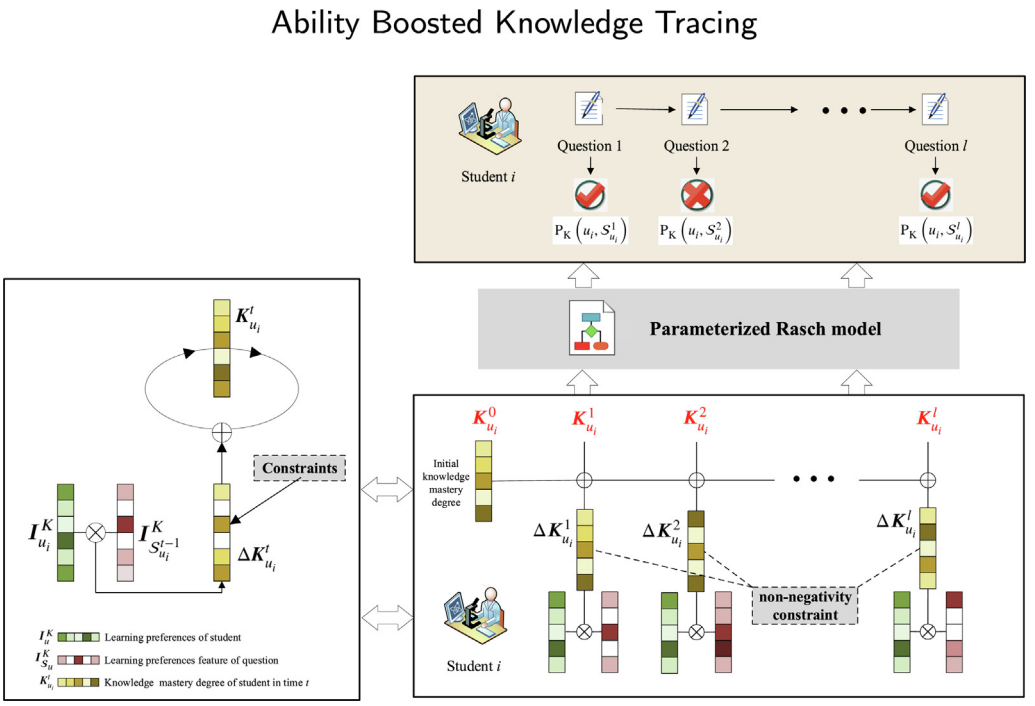

Fig. 5. Sketch of continuous matrix factorization model (knowledge module).

Ability, as an attribute of learners, displays three main characteristics: tacitness, compressibility, and stability. Furthermore, we believe that the learners with similar behaviors are more likely to possess similar abilities. On the basis of these analyses and hypotheses, the latent factor and graph neural network models were introduced to depict learner ability.

In the ability module, the learner-question interaction data can be represented as an undirected and unweighted graph. Its adjacency matrix can be recognized as $\mathbf{A}$, where $\mathbf{A} \in \mathbb{R}^{(m+s)\times(m+s)}$. $\mathbf{H}^{(0)}$ denotes the initial feature of each node (students and questions), where $\mathbf{H}^{(0)} \in \mathbb{R}^{(m+s)\times k_A}$, k_A is a hyper-parameter set manually.

Graph convolutional network (GCN) [17] is an emblematic graph neural network model and a powerful tool for extracting features from graph-structured data. GCN is contructed by stacking feature aggregation layers, to generate analogical features for nodes sharing similar relations. Thus, GCN is an ideal model for learner ability modeling. The fundamental feature aggregation formula of GCN is as follows.

$$\begin{aligned} H^{(l+1)} &= \sigma\left(\tilde{\mathbf{A}}H^{(l)}W^{(l+1)}\right) \\ \tilde{\mathbf{A}} &= \hat{D}^{-1/2}\hat{\mathbf{A}}\hat{D}^{-1/2} \end{aligned} \tag{5}$$

where $\tilde{\mathbf{A}}$ denotes the symmetric normalized Laplacian matrix, $\hat{\mathbf{A}}$ is the Laplacian matrix, where $\hat{\mathbf{A}} = \mathbf{A} + I$, and I is the identity matrix. $\hat{D}$ represents the degree matrix of the interaction graph.

The output of each layer in the GCN model is related to the output of its previous layer, for example, in Eq. 4, $H^{(l+1)}$ is related to $H^{(l)}$. However, this structure does not allow GCN to be optimized by the stochastic or mini-batch iterative methods, such as SGD. Thus, training a GCN model can be time consuming, especially when processing large-scale data. To alleviate this problem, inspired by [43], in this study, the feature aggregation layer of GCN was linearized, and expressed as follows.

$$\begin{aligned} H^{(l+1)} &= \tilde{\mathbf{A}}H^{(l)}W^{(l+1)} \\ &= \tilde{\mathbf{A}}\tilde{\mathbf{A}}H^{(l-1)}W^{(l)}W^{(l+1)} \\ &\cdots \\ &= \underbrace{\tilde{\mathbf{A}}\tilde{\mathbf{A}}\cdots\tilde{\mathbf{A}}}_{l+1}H^{(0)}W^{(1)}W^{(2)}\cdots W^{(l+1)} \end{aligned} \tag{6}$$

where l is the depth of feature aggregation model, $H^{(0)}, W^{(1)}, W^{(2)}\cdots W^{(l+1)}$ are all trainable parameters, such that, their product is equivalent to the trainable parameter matrix $H^{(0)}$. Thus, Eq. (6) is equivalent to

$$H^{(l)} = \tilde{\mathbf{A}}^l H^{(0)} \tag{7}$$

With this treatment, the feature update process is dramatically simplified. The relations between adjacent layers is broken, as shown in Eq. 7 such that $H^{(l)}$ can be computed directly from $H^{(0)}$. This new structure enables the model to be easily optimized using stochastic or mini-batch iterative methods. In general, deeper GCNs have larger feature receptive fields, but the scale of the computations scale proportionally increases. Using this linearization treatment (Eq. 7), $\tilde{\mathbf{A}}^l$ can be considered a constant and computed before the iterative optimization process. Furthermore, when the depth of the GNN model (l) increases, only the value of $\tilde{\mathbf{A}}^l$ changes, without changing the scale of $\tilde{\mathbf{A}}^l$ or the update operator, whereby the computational complexity of the feature update process remains constant.

An adaptive matrix, $\mathbf{A}^a$, was introduced to adjust the feature propagation matrix ($\tilde{\mathbf{A}}^l$) adaptively. $\mathbf{A}^a$ has the same shape as $\tilde{\mathbf{A}}^l$ and can be indicated as a sparse matrix in the training process. The feature aggregation formula of the LGLA model is formulated as follows.

$$H^{(l)} = \tilde{\mathbf{A}}^l \circ \mathbf{A}^a \cdot H^{(0)} \tag{8}$$

The aggregated feature matrix, $H^{(l)}$, represents the latent ability feature of all nodes (students and questions). The latent ability feature of student u_i can be represented by $\mathbf{I}^A_{u_i}$, which is a specific row in $H^{(l)}$ for node student u_i. Similarly, $\mathbf{I}^A_{\mathscr{S}^t_{u_i}}$ denotes the latent ability feature of question $\mathscr{S}^t_{u_i}$ of the form $H^{(l)}$. $\mathrm{P_A}\left(u_i, \mathscr{S}^t_{u_i}\right)$ denotes the probability that student u_i can solve question $\mathscr{S}^t_{u_i}$ from the perspective of ability, which can be expressed as follows.

$$\mathrm{P_A}\left(u_i, \mathscr{S}^t_{u_i}\right) = \mathbf{I}^A_{u_i} \cdot {\mathbf{I}^A_{\mathscr{S}^t_{u_i}}}^{\mathrm{T}} + g^A_{u_i} + g^A_{\mathscr{S}^t_{u_i}} \tag{9}$$

In Eq. 9, $g^A_{u_i}$ and $g^A_{\mathscr{S}^t_{u_i}}$, respectively represent the global ability factors of student u_i and question $\mathscr{S}^t_{u_i}$, which are trainable parameters.

In Sections 3.3 and 3.4, the knowledge and ability modules are elaborated upon, The prediction functions of the corresponding modules, $\mathrm{P_K}\left(u_i, \mathscr{S}^t_{u_i}\right)$ and $\mathrm{P_A}\left(u_i, \mathscr{S}^t_{u_i}\right)$, are presented. In the next section, where the idea of dual module integration is discussed, and the knowledge and ability dual-tracing framework is proposed along with the objective functions of ABKT. Fig. 6.

3.5. Knowledge and ability dual tracing framework

3.5.1. Joint model

In the joint model, $\mathrm{P}\left(u_i, \mathscr{S}^t_{u_i}\right)$ represents the probability that student u_i can solve question $\mathscr{S}^t_{u_i}$. Based on this idea, $\mathrm{P}\left(u_i, \mathscr{S}^t_{u_i}\right)$ is composed of knowledge and ability factors, namely $\mathrm{P_K}\left(u_i, \mathscr{S}^t_{u_i}\right)$ and $\mathrm{P_A}\left(u_i, \mathscr{S}^t_{u_i}\right)$ respectively. Two simple hypotheses are explored with respect to the structure of the joint model: *additive joint model* and *multiplicative joint model*. The two joint models can be expressed as follows.

- *additive joint model*: $\mathrm{P^{add}}\left(u_i, \mathscr{S}^t_{u_i}\right) = \mathrm{P_K}\left(u_i, \mathscr{S}^t_{u_i}\right) + \mathrm{P_A}\left(u_i, \mathscr{S}^t_{u_i}\right)$
- *Multiplicative joint model*: $\mathrm{P^{mul}}\left(u_i, \mathscr{S}^t_{u_i}\right) = \mathrm{P_K}\left(u_i, \mathscr{S}^t_{u_i}\right) \cdot \mathrm{P_A}\left(u_i, \mathscr{S}^t_{u_i}\right)$

Additive and multiplicative joint models are two representative models frequently used for data-aggregation with different assumptions. The additive joint model assumes that each component is interchangeable. In contrast, the multiplicative joint model assumes that each component is concurrent. Let us consider the research problem in this study as an example. If the probability of learners solving problems is low from the perspective of knowledge ($\mathrm{P_K}\left(u_i, \mathscr{S}^t_{u_i}\right) = 0.1$), and if the learner's ability is strong ($\mathrm{P_A}\left(u_i, \mathscr{S}^t_{u_i}\right) = 0.8$), the additive joint model assumes the learner has a high probability ($\mathrm{P^{add}}\left(u_i, \mathscr{S}^t_{u_i}\right) = 0.9$) to solve the overall problem. However, in the multiplicative joint model ($\mathrm{P^{mul}}\left(u_i, \mathscr{S}^t_{u_i}\right) = 0.08$). In other words, the multiplicative joint model assumes that learners can solve the problem only when they meet the requirements of both knowledge and ability. However, the additive joint model claims that the two aspects of knowledge and ability are complementary.

In the next section, the objective function of the ABKT model is introduced. The objective function of ABKT is related to the joint model. Therefore, in the next section, the objective functions of the additive and multiplicative models are separately demonstrated.

3.5.2. Objective function

To integrate the knowledge and ability models, a boosting method, which is a typical ensemble learning method is introduced. In the ensemble learning framework, a single model (knowledge or ability) is regarded as a weak classifier, which will make unreliable predictions. ABKT introduces the boosting method to integrate the knowledge and ability modules, thereby

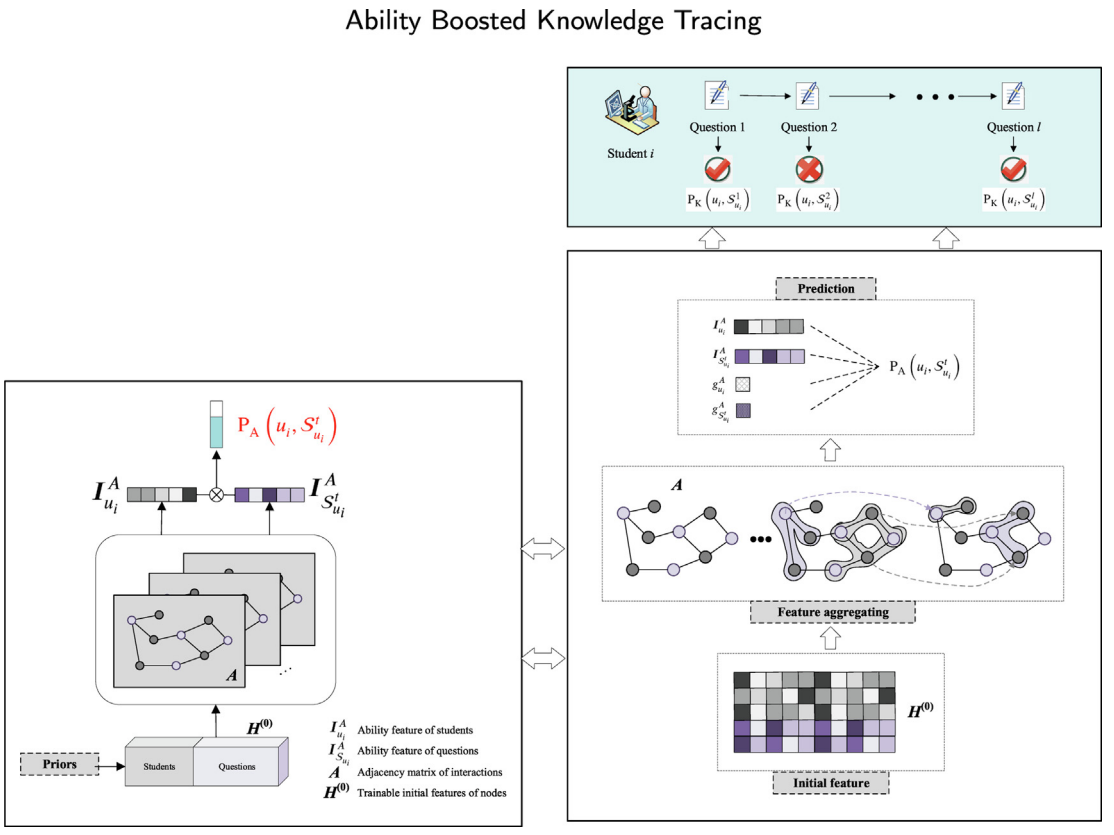

Fig. 6. Sketch of linear graph latent ability model (ability module).

improving model performance. Based on the boosting method protocol, the CMF model (in the knowledge module) is set as the foregoing model. Then, the LGLA model (in the ability module) can be treated as a complementary model and, trained based on the results of the CMF model. The likelihood function of CMF model for the entire dataset can be expressed as

$$\mathrm{P_K}(\mathbf{U},S,C|\Theta_K)=\prod_{i=1}^{m}\prod_{t=1}^{l}\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^t\right)^{\mathscr{C}_{u_i}^t}\cdot\left[1-\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^t\right)\right]^{1-\mathscr{C}_{u_i}^t} \tag{10}$$

where Θ_K denotes all parameters in the CMF model, and, the log-likelihood function is given as,

$$\begin{aligned}\mathscr{L}_K &= \log\ \mathrm{P_K}(\mathbf{U},S,C|\Theta_K)\\ &=\sum_{i=1}^{m}\sum_{t=1}^{l}\mathscr{C}_{u_i}^t\ln\left[\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^t\right)\right]+\left(1-\mathscr{C}_{u_i}^t\right)\ln\left[1-\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^t\right)\right]\end{aligned} \tag{11}$$

Based on the maximum likelihood estimation (MLE) rule, the learnable parameters in the CMF model can be optimized by maximizing the log-likelihood function, namely $\mathscr{L}_K$, presented in Eq. (11).

The objective function of the LGLA model depends on the mold of the joint model. Thus, the objective function for the two joint models (additive and multiplicative) is discussed separately.

Additive joint model:

The log-likelihood function of the *additive joint model* can be expressed as:

$$\begin{aligned}\mathscr{L}^{add} &=\sum_{i=1}^{m}\sum_{t=1}^{l}\mathscr{C}_{u_i}^t\ln\left[\mathrm{P^{add}}\left(u_i,\mathscr{S}_{u_i}^t\right)\right]+\left(1-\mathscr{C}_{u_i}^t\right)\ln\left[1-\mathrm{P^{add}}\left(u_i,\mathscr{S}_{u_i}^t\right)\right]\\ &=\sum_{i=1}^{m}\sum_{t=1}^{l}\mathscr{C}_{u_i}^t\ln\left[\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^t\right)+\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^t\right)\right]+\left(1-\mathscr{C}_{u_i}^t\right)\ln\left[1-\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^t\right)-\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^t\right)\right]\end{aligned} \tag{12}$$

the Taylor series expansion is applied to $\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^t\right)=0$ to simplify $\mathscr{L}^{add}$.

$$\begin{aligned}\mathscr{L}^{add} &\approx \sum_{i=1}^{m}\sum_{t=1}^{l}\mathscr{C}_{u_i}^{t}\ln\left[\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)\right]+\left(1-\mathscr{C}_{u_i}^{t}\right)\ln\left[1-\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)\right]+g_{it}\cdot\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)+\frac{w_{it}\left[\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)\right]^2}{2}\\ &=\mathscr{L}_K+\frac{1}{2}\sum_{i=1}^{m}\sum_{t=1}^{l}w_{u_i}^{t}\left[\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)+\frac{g_{u_i}^{t}}{w_{u_i}^{t}}\right]^2-\frac{1}{2}\sum_{i=1}^{m}\sum_{t=1}^{l}\frac{g_{u_i}^{t\ 2}}{w_{u_i}^{t}}\end{aligned}\tag{13}$$

In the simplified calculations course only retain the first- and second-order terms are retained in the Taylor expansion formula, where g_{it} and w_{it} denote the first and second derivatives of $\mathscr{L}^{add}$ in $\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)=0$ respectively. g_{it} and w_{it} can be written as follows:

$$\begin{cases}\frac{\partial\mathscr{L}^{add}}{\partial\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)}=\mathscr{C}_{u_i}^{t}\cdot\frac{1}{\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)+\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)}-\left(1-\mathscr{C}_{u_i}^{t}\right)\cdot\frac{1}{1-\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)-\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)}\\ \frac{\partial^2\mathscr{L}^{add}}{\partial\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)^2}=\mathscr{C}_{u_i}^{t}\cdot\frac{-1}{\left[\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)+\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)\right]^2}-\left(1-\mathscr{C}_{u_i}^{t}\right)\cdot\frac{-1}{\left[1-\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)-\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)\right]^2}\end{cases}\tag{14}$$

where $\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)=0$, then,

$$\begin{cases}g_{u_i}^{t}=\frac{\mathscr{C}_{u_i}^{t}}{\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)}-\frac{1-\mathscr{C}_{u_i}^{t}}{1-\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)}\\ w_{u_i}^{t}=-\frac{\mathscr{C}_{u_i}^{t}}{\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)^2}-\frac{1-\mathscr{C}_{u_i}^{t}}{\left[1-\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)\right]^2}\end{cases}\tag{15}$$

For the *additive joint model*, the overall objective is to maximize the log-likelihood function in Eq. (13), based on the MLE rule. The first term in Eq. (13) can be optimized by (maximizing the log-likelihood function of the CMF model, as shown in Eq. (11)), Then the only part that needs to optimized in Eq. (13) is the LGLA model (second term), which can be expressed as follows:

$$\mathscr{L}_A^{add}=\frac{1}{2}\sum_{i=1}^{m}\sum_{t=1}^{l}w_{u_i}^{t}\left[\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)+\frac{g_{u_i}^{t}}{w_{u_i}^{t}}\right]^2\tag{16}$$

Eq. (16) represents the objective function of the LGLA model based on the hypothetical *additive joint model*. The parameters in the LGLA model can be optimized by maximizing Eq. (16).

Multiplicative joint model:

Similarly, the log-likelihood function of the *multiplicative joint model* is expressed as follows:

$$\begin{aligned}\mathscr{L}^{mul} &=\sum_{i=1}^{m}\sum_{t=1}^{l}\mathscr{C}_{u_i}^{t}\ln\left[\mathrm{P^{mul}}\left(u_i,\mathscr{S}_{u_i}^{t}\right)\right]+\left(1-\mathscr{C}_{u_i}^{t}\right)\ln\left[1-\mathrm{P^{mul}}\left(u_i,\mathscr{S}_{u_i}^{t}\right)\right]\\ &=\sum_{i=1}^{m}\sum_{t=1}^{l}\mathscr{C}_{u_i}^{t}\ln\left[\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)\cdot\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)\right]+\left(1-\mathscr{C}_{u_i}^{t}\right)\ln\left[1-\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)\cdot\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)\right]\end{aligned}\tag{17}$$

In order to simplify $\mathscr{L}^{mul}$, the Taylor series expansion is applied to $\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)=1$. Similar to the additive joint model, only the first- and second-order terms are retained in Taylor expansion formula to simplify the log-likelihood function in Eq. (17). To make the expression more concise, the same sign system as the additive joint model is adopted in the following, where the first and second derivative functions of $\mathscr{L}^{mul}$ about $\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)$ are expressed as

$$\begin{cases}\frac{\partial\mathscr{L}^{mul}}{\partial\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)}=\mathscr{C}_{u_i}^{t}\cdot\frac{\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)}{\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)\cdot\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)}-\left(1-\mathscr{C}_{u_i}^{t}\right)\cdot\frac{\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)}{1-\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)\cdot\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)}\\ \frac{\partial^2\mathscr{L}^{mul}}{\partial\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)^2}=\mathscr{C}_{u_i}^{t}\cdot\frac{-\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)^2}{\left[\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)\cdot\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)\right]^2}-\left(1-\mathscr{C}_{u_i}^{t}\right)\cdot\frac{-\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)^2}{\left[1-\mathrm{P_K}\left(u_i,\mathscr{S}_{u_i}^{t}\right)\cdot\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)\right]^2}\end{cases}\tag{18}$$

according to the assumption $\mathrm{P_A}\left(u_i,\mathscr{S}_{u_i}^{t}\right)=1$, then,

$$\begin{cases} \mathscr{L}^{mul'}\left(\mathrm{P_A}\left(u_i,\mathscr{S}^t_{u_i}\right)=1\right) & = \mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)\cdot\left(\frac{\mathscr{C}^t_{u_i}}{\mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)}-\frac{1-\mathscr{C}^t_{u_i}}{1-\mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)}\right) \\ & = \mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)\cdot g^t_{u_i} \\ \mathscr{L}^{mul''}\left(\mathrm{P_A}\left(u_i,\mathscr{S}^t_{u_i}\right)=1\right) & = \mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)\cdot\left(\frac{-\mathscr{C}^t_{u_i}}{\mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)^2}-\frac{1-\mathscr{C}^t_{u_i}}{\left[1-\mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)\right]^2}\right) \\ & = \mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)\cdot w^t_{u_i} \end{cases} \tag{19}$$

where $g^t_{u_i}$ and $w^t_{u_i}$ share the same form, as in Eq. (15), and $\mathscr{L}^{mul}$ (Eq. 17) can be written as,

$$\begin{aligned} \mathscr{L}^{mul} \approx & \sum_{i=1}^{m}\sum_{t=1}^{l}\mathscr{C}^t_{u_i}\ln\left[\mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)\right]+\left(1-\mathscr{C}^t_{u_i}\right)\ln\left[1-\mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)\right] \\ & +\mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)\cdot\left(\mathrm{P_A}\left(u_i,\mathscr{S}^t_{u_i}\right)-1\right)\cdot g^t_{u_i}+\frac{1}{2}\mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)^2\left(\mathrm{P_A}\left(u_i,\mathscr{S}^t_{u_i}\right)-1\right)^2\cdot w^t_{u_i} \\ = & \mathscr{L}_K+\frac{1}{2}\sum_{i=1}^{m}\sum_{t=1}^{l}w^t_{u_i}\cdot\mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)\left[\mathrm{P_A}\left(u_i,\mathscr{S}^t_{u_i}\right)-\left(1-\frac{g^t_{u_i}}{w^t_{u_i}\cdot\mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)}\right)\right]^2-\frac{1}{2}\sum_{i=1}^{m}\sum_{t=1}^{l}\frac{{g^t_{u_i}}^2}{w^t_{u_i}} \end{aligned} \tag{20}$$

Similar to the additive joint model, a two-step optimization strategy is applied. The first term in Eq. (20) is optimized as in step one by maximizing the log-likelihood function of the CMF model, as shown in Eq. (11). The only part that needs to be optimized in Eq. (20) is the second term (LGLA model), which can be summarized as follows:

$$\mathscr{L}^{mul}_A=\frac{1}{2}\sum_{i=1}^{m}\sum_{t=1}^{l}w^t_{u_i}\cdot\mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)\left[\mathrm{P_A}\left(u_i,\mathscr{S}^t_{u_i}\right)-\left(1-\frac{g^t_{u_i}}{w^t_{u_i}\cdot\mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)}\right)\right]^2 \tag{21}$$

The objective function of the LGLA model is given by Eq. (21). The parameters in the ability model can be optimized by maximizing Eq. (21).

3.5.3. Regularization

The objective function of the knowledge (Eq. (11)) and ability (Eqs. (16) and (21)) modules are presented in Section 3.5.2. There are two steps in optimizing the ABKT model: (1) optimizing the CMF model and (2) optimizing the LGLA model. It can be seen from Eq. (16) or (21) that the objective function of the LGLA model is related to the output of the CMF model, i.e. $\mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)$. The value range of $\mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)$ is [0,1]. However, in the objective function of the LGLA model, $\mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)$ is applied to the denominator, which makes the value range of the regression target of $\mathrm{P_A}\left(u_i,\mathscr{S}^t_{u_i}\right)$[1,+∞] in Eq. (16) or (21). This will make the training process of the LGLA model difficult to control and may even cause the model to lose efficacy, because of the extreme numerical differences on the regression target. To alleviate this issue, a value clip is introduced to regulate the output of the CMF model.

$$\mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)=\mathrm{CLIP}\left(\mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right),[\mu,1-\mu]\right) \tag{22}$$

where CLIP is the clip operation and $[\mu,1-\mu]$ denotes the boundary range. With the clip treatment, the value range of the regression target of the LGLA model can be controlled to improve the model stationarity.

In addition, in the CMF model, a non-negativity restriction is applied to knowledge internalization, $\Delta\mathbf{K}^t_{u_i}$, in Eq. (3). For the LGLA model, the Frobenius norm was introduced to regularize the initial feature matrix $\mathbf{H}^{(0)}$, similar to the classic latent factor models [23], which equals to apply the Gaussian prior to parameters. Subsequently, the objective functions of the LGLA model with the additive or multiplicative joint model are presented.

$$\begin{aligned} \mathscr{L}^{add}_A &= \frac{1}{2}\sum_{i=1}^{m}\sum_{t=1}^{l}w^t_{u_i}\left[\mathrm{P_A}\left(u_i,\mathscr{S}^t_{u_i}\right)+\frac{g^t_{u_i}}{w^t_{u_i}}\right]^2-\lambda\|H^{(0)}\|^2 \\ \mathscr{L}^{mul}_A &= \frac{1}{2}\sum_{i=1}^{m}\sum_{t=1}^{l}w^t_{u_i}\cdot\mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)\left[\mathrm{P_A}\left(u_i,\mathscr{S}^t_{u_i}\right)-\left(1-\frac{g^t_{u_i}}{w^t_{u_i}\cdot\mathrm{P_K}\left(u_i,\mathscr{S}^t_{u_i}\right)}\right)\right]^2-\lambda\|H^{(0)}\|^2 \end{aligned} \tag{23}$$

λ is a regularization parameter, that balances the fidelity and regularization terms in the objective function.

4. Optimization and parameter determination

4.1. Optimization of ABKT

The ABKT model is proposed under a typical boosting ensemble learning framework, which can be optimized by a sectionalized and sequential protocol as follows:

1. Optimizing the parameters in the CMF model (knowledge module), by maximizing $\mathscr{L}_K$, in Eq. (11).
2. Computing and regularizing $\mathrm{P_K}\left(u_i, \mathscr{S}^t_{u_i}\right)$.
3. Optimizing the parameters in the LGLA model (ability module), by maximizing $\mathscr{L}_A^{add}$ or $\mathscr{L}_A^{mul}$, in Eq. (23).

More precisely, the parameters in the CMF model can be optimized using gradient-based iterative methods, such as the standard gradient ascent algorithm, where

$$\begin{aligned} &K_u^0, \mathbf{I}_u^K, \mathbf{I}_{\mathscr{S}_u}^K = \arg\max_{K_u^0, \mathbf{I}_u^K, \mathbf{I}_{\mathscr{S}_u}^K} \mathscr{L}_K \\ &\Rightarrow \begin{cases} K_u^0 \leftarrow & K_u^0 + \alpha \cdot \frac{\partial \mathscr{L}_K}{\partial K_u^0} \\ \mathbf{I}_u^K \leftarrow & \mathbf{I}_u^K + \alpha \cdot \frac{\partial \mathscr{L}_K}{\partial \mathbf{I}_u^K} \\ \mathbf{I}_{\mathscr{S}_u}^K \leftarrow & \mathbf{I}_{\mathscr{S}_u}^K + \alpha \cdot \frac{\partial \mathscr{L}_K}{\partial \mathbf{I}_{\mathscr{S}_u}^K} \end{cases} \end{aligned} \tag{24}$$

LGLA model can be optimized in a similar manner,

$$\begin{aligned} &H^{(0)}, \mathbf{g}_u^A, \mathbf{g}_{\mathscr{S}_u}^A, GQ = \arg\max_{H^{(0)}, \mathbf{g}_u^A, \mathbf{g}_{\mathscr{S}_u}^A} \mathscr{L}_A^{add} \text{or} \mathscr{L}_A^{mul} \\ &\Rightarrow \begin{cases} H^0 \leftarrow & H^0 + \alpha \cdot \frac{\partial \mathscr{L}_A^{add} \text{or} \mathscr{L}_A^{mul}}{\partial H^0} \\ \mathbf{g}_u^A \leftarrow & \mathbf{g}_u^A + \alpha \cdot \frac{\partial \mathscr{L}_A^{add} \text{or} \mathscr{L}_A^{mul}}{\partial \mathbf{g}_u^A} \\ \mathbf{g}_{\mathscr{S}_u}^A \leftarrow & \mathbf{g}_{\mathscr{S}_u}^A + \alpha \cdot \frac{\partial \mathscr{L}_A^{add} \text{or} \mathscr{L}_A^{mul}}{\partial \mathbf{g}_{\mathscr{S}_u}^A} \end{cases} \end{aligned} \tag{25}$$

where α denotes the learning rate of gradient-based iterative optimization methods. The partial derivatives of $\mathscr{L}_K$ and $\mathscr{L}_A$ with respect to parameters can be computed using the chain rule. To better illustrate this framework the major steps of the entire process are summarized in **Algorithm 1**.

Algorithm 1: Ability boosted knowledge tracing

Input : student set $\boldsymbol{\mathcal{U}}$, learning sequence and its feedback $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{C}$, Q-matrix $\boldsymbol{Q}$
Set : dimensionality in knowledge and ability model namely k_K and k_A, surmise in knowledge model γ, clip range μ, regularization factor in ability model λ, learning rate α
Output: ABKT model

Initialize $\boldsymbol{K}_u^0$, $\boldsymbol{I}_u^K$, $\boldsymbol{I}_{\boldsymbol{S}_u}^K$ in knowledge model randomly
while $\mathcal{L}_K$ *is not converged* **do**
  Update $\boldsymbol{K}_u^0$, $\boldsymbol{I}_u^K$, $\boldsymbol{I}_{\boldsymbol{S}_u}^K$ via Eq. (24)
foreach *element in* $\boldsymbol{\mathcal{U}}$ *and* $\boldsymbol{\mathcal{S}}$ **do**
  Compute and regularize $\mathrm{P_K}\left(u_i, S^t_{u_i}\right)$ via Eq. (4) and (22)
Initialize $\boldsymbol{H}^{(0)}$, $\boldsymbol{g}_u^A$, $\boldsymbol{g}_{\boldsymbol{S}_u}^A$ in ability model randomly
while $\mathcal{L}_A^{add}$ *or* $\mathcal{L}_A^{mul}$ *is not converged* **do**
  Update $\boldsymbol{H}^{(0)}$, $\boldsymbol{g}_u^A$, $\boldsymbol{g}_{\boldsymbol{S}_u}^A$ via Eq. (25)

4.2. Parameter determination

In CMF model, shown in Eq. (4), γ is the surmise, which is a dataset-related hyperparameter usually set to 0 or 0.25 depending on the type of question in the dataset; k_K and k_A are the dimensionality of the knowledge and ability modules, respectively, indicating the number of latent factors. They control the capacity of the knowledge and ability modules, and the larger the dimensionality set, the larger is the capacity of the model; l denotes the depth of feature aggregation in the LGLA model. In the objective function of the LGLA model, shown in Eq. (19), λ is the control factor, which balances the weights of the first and second terms in the objective function. μ denotes the clip range, as shown in Eq. (18).

In this study, the hyperparameters were determined heuristically. A large range of hyperparameters were validated using the generalized cross-validation [1] technique. Depending on the different scales of datasets, small changes are introduced to the optimal hyperparameter combination. In most cases, γ is set to 0.25 and $\mu = 0.4$. For λ, k_K and k_A, the greedy search strategy was used to determine the values. The details are presented in the next section. Excellent performance was achieved with the parameters of $\gamma = 0.25$, $\mu = 0.4$, $k_K = 5$, $k_A = 32$ or 64, $l = 1$ or 2, $\lambda = 0.1$.

5. Experiments and discussion

5.1. Datasets

Four datasets were included in the experiment. The ASSISTment dataset[1] was collected from the ASSISTment tutoring system and shared by [14]. It is the most classic and widely used dataset in KT research. In this study, two datasets are built with the data with regard to the two modes in ASSISTment09 dataset: random iterate section, abbreviated as RIS, (D1) and random child order section, abbreviated as RCOS, (D2). In addition, the AICFE dataset [2] represents the data collected from the smart learner platform, which was developed by the Advanced Innovation Center for Future Education (AICFE) center at the Beijing Normal University. Two subsets from the AICFE dataset were also introduced into our experiments, namely Math (D3) and Physics (D4). The statistical information on the four datasets is summarized in Table 2.

The feedback space on all four datasets is {0, 1}, where 0 means that the student completed the exercise incorrectly, and 1 represents correct completion. To obtain objective and unbiased results, the latest feedback prediction task was employed as an evaluation protocol, which primarily fits the definition of knowledge tracing. The latest feedback prediction task was set to the last feedback of all students and constituted the test set, and the remaining learning sequences were set as the training set. The KT model predicts the most recent performance for each student according to their historical interactive information. The latest feedback prediction task demonstrates the personalized modeling ability of the KT models.

5.2. Evaluation metrics

Knowledge tracing is a complex task with many aspects. Thus there are many applications targeting the different aspects, such as feedback prediction and knowledge state estimation. In this study, feedback prediction was introduced as the main evaluation task. Prediction accuracy, the error between predictions and the actual feedback, is introduced as our main evaluation metric because it demonstrates the most significant value and advancements of KT models. In our experiments, two popular error metrics were used to measure the prediction accuracy: Prediction Accuracy (ACC) and Area Under ROC Curve (AUC). The higher the ACC or AUC, the higher is the accuracy.

5.3. Tested models

Nine models were included in our experiments.

a. PFA: Performance factor analysis was presented by Pavlik et al. in [32], who introduced the cumulative feedback information to predict learning feedback. The PFA is a representative and widely used KT model.
b. IRT: Item response theory-based knowledge tracing model, which is a popular cognitive diagnostic model modeling student exercise records by subjective and objective factors with a logistic-like function, proposed by Masters in [22].
c. DKT: The deep knowledge tracing model is the earliest deep learning-based KT model that employs a recurrent neural network to model students' exercise routines, exhibiting better performance compared to classical KT models. The DKT [33] was constructed by Piech et al.
d. DHKT: Deep hierarchical knowledge tracing is one of the most advanced deep learning-based KT models, proposed by Wang et al. in [42]. The hierarchical relations between concepts and items are modeled by hinge loss, which improves item representation quality.
e. KTM: Knowledge tracing machines [41] is a state-of-the-art knowledge model, developed by Vie et al. KTM employs factorization machines to model student knowledge with high processing efficiency.

[1] https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data
[2] https://aic-fe.bnu.edu.cn/cgzs/kfsj/index.html

Table 2
Statistics of experimental datasets.

Datasets	D1	D2	D3	D4
	ASSISTment09		AICFE	
category	RIS	RCOS	math	physics
# of records	206 516	45 206	52 301	36 354
# of students	2 791	1 621	1 080	488
# of questions	1 209	957	609	1 432
# of concepts	102	132	32	50
Avg. exercising records per student	73.99	27.88	48.42	74.49
Avg. exercises per knowledge concept	11.85	7.25	19.03	28.64

f. SAINT: Separated self-attentive neural knowledge tracing builds an encoder-decoder structure to separately represent exercise and response with stacked self-attention layers. SAINT is a state-of-the-art KT model that was proposed by Choi et al. [5].
g. AKT: Attentive knowledge tracing [11] couples flexible attention-based neural network models with monotonic attention mechanisms as well as Rasch model-based concepts and question embedding regularizations. The experimental results indicate that AKT is one of the best preforming KT models.
h. ABKT-A: Ability boosted knowledge tracing with additive assumption is the model proposed in this study, which applies *additive joint model*.
i. ABKT-M: is the proposed model with *multiplicative joint model*.

5.4. Experimental settings

Framework Setting. For ABKT-A and ABKT-M, the same hyperparameter combinations were set, with slightly different settings for the four involved datasets. The γ was set to 0.25 and $\mu = 0.4$ for all datasets. The dimensionality in the knowledge model, k_K, was set to 5 in all datasets. l was set to 1 for ABKT-A model and to 2 for ABKT-M. k_A and λ were equal to 64 and 0.1 respectively in the ASSIST09 datasets (D1 and D2), and were similarly set to 32 and 0.1, respectively for the AICFE datasets (D3 and D4).

Training Setting. During the training process, all eight comparison models were trained under the same settings. The Adam optimizer was employed in our experiment, where *lr* = 0.0005, *beta1* = 0.9, *beta2* = 0.999 and the batch size was set to 512. The experiment was performed on a PC server equipped with an Intel(R) Core(TM) i7-7700 K CPU@4.20 GHz, NVIDIA GeForce GTX 1080 Ti GPU, and 32 GB RAM. ABKT was implemented using the Pytorch software library [30].

5.5. Results and discussion

5.5.1. Accuracy analysis of ABKT models

For the accuracy comparison of PFA, IRT, DKT, DHKT, KTM, SAINT, AKT, ABKT-A and ABKT-M on all four datasets, the highest ACC and AUC of each method and their ranks are summarized in Table 3. Some interesting facts can be observed in Table 3. First, in general, deep learning-based KT models outperform the traditional methods. DKT, DHKT, SAINT and AKT exhibit significant improvements over the baseline methods, PFA and IRT, in all four datasets. However, the KTM and ABKT models are competitive in performance compared to the most advanced DL models, in diverse frameworks. This phenomenon also explains, to some extent, that the bottleneck of the current KT method is not the expressive ability of the learning algorithm, but the understanding and modeling framework of the KT problem. Second, the proposed methods,

Table 3
Performance comparison in terms of ACC, AUC on D1, D2, D3 and D4.The best results are highlighted in bold and the number in parenthesis is the rank of each algorithm.

	D1		D2		D3		D4	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
PFA	0.6804 (9)	0.7493 (9)	0.6421 (9)	0.6939 (9)	0.5685 (9)	0.6703 (9)	0.6639 (9)	0.6824 (9)
IRT	0.7022 (8)	0.7664 (8)	0.6631 (8)	0.7085 (8)	0.6509 (8)	0.7064 (8)	0.6946 (7)	0.7222 (7)
DKT	0.7311 (7)	0.8051 (7)	0.7001 (7)	0.7371 (7)	0.6666 (7)	0.7370 (7)	0.6885 (8)	0.7211 (8)
DHKT	0.7488 (5)	0.8214 (6)	0.7088 (5)	0.7561 (5)	0.6916 (6)	0.7865 (6)	0.7418 (4)	0.8026 (5)
KTM	0.7477 (6)	0.8234 (5)	0.7149 (4)	0.7616 (4)	0.7212 (3)	0.7923 (3)	0.7397 (5)	0.8133 (4)
SAINT	0.7491 (4)	0.8263 (4)	0.7075 (6)	0.7531 (6)	0.6972 (5)	0.7892 (5)	0.7382 (6)	0.8014 (6)
AKT	0.7521 (3)	0.8297 (3)	0.7162 (3)	0.7652 (3)	0.7021 (4)	0.7902 (4)	0.7452 (3)	0.8142 (3)
ABKT-A	**0.7631 (1)**	**0.8403 (1)**	0.7186 (2)	0.7740 (2)	**0.7490 (1)**	**0.8239 (1)**	0.7725 (2)	**0.8290 (1)**
ABKT-M	0.7595 (2)	0.8370 (2)	**0.7186 (1)**	**0.7771 (1)**	0.7259 (2)	0.8038 (2)	**0.7766 (1)**	0.8229 (2)

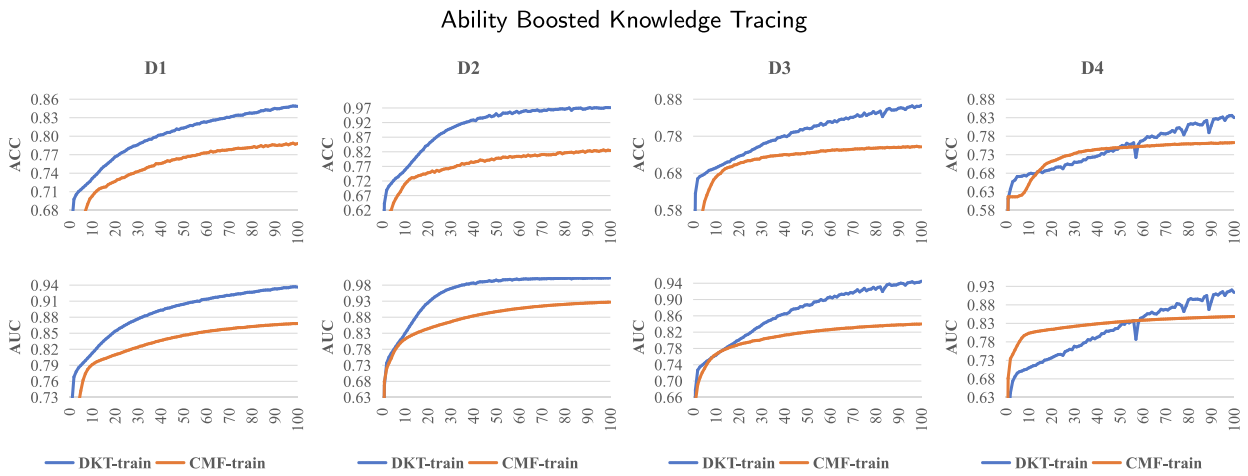

Fig. 7. Training processes of DKT and CMF compared with ACC and AUC measuring the fitting error on D1-D4 in training set.

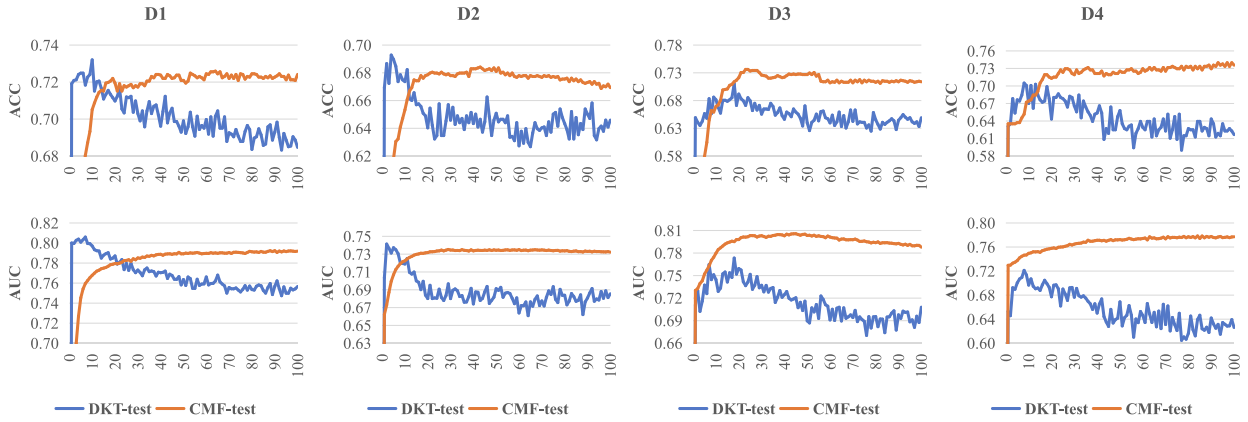

Fig. 8. Training processes of DKT and CMF compared with ACC and AUC measuring the generalized error on D1-D4 in test set.

ABKT-A and ABKT-M, achieved higher accuracy on both the ACC and AUC metrics in all experiments. However, ABKT models are built on simple linear models, and they outperform methods constructed based on complex learning models (GRU in DKT and DHKT, factorizing machine in KTM, transformer encoder in SAINT and AKT). This further demonstrates that ABKT is advanced in its algorithmic mechanism, deconstructing the KT problem more efficaciously.

5.5.2. Effectiveness analysis of regularization in knowledge model

ABKT is a boosting-based ensemble learning model, consisting two single modules: the knowledge and ability modules. To obtain good performance with the overall model, the sub-models in the ensemble learning framework need to pursue generalization ability rather than fitting ability. Thus, in the CMF model, a non-negativity restriction is applied to knowledge internalization estimation. To analyze the generalization ability and hypothesis reasonableness of the proposed CMF model, a few experiments were carried out on the four datasets with DKT as the control method. The fitting and generalization errors in the training process are displayed in Figs. 7 and 8 respectively. Fig. 7 shows the fitting errors, measured by ACC and AUC, in the training set. It is observed that the DKT model has smaller fitting error compared to CMF. In addition, DKT achieved nearly 100% fitting degree in all four datasets. CMF, on the other hand, achieved 75 to 80% fitting accuracy. These results demonstrate that compared to the CMF model, the DKT model has stronger fitting ability and larger model capacity. Fig. 8 demonstrates the model generalization errors in the test set. It is observed that both CMF and DKT reach the optimal generalization point in approximately 10–20 epochs. Subsequently, the generalization error of the DKT increases, while that of CMF remains stable. This indicates that the CMF model has better generalization ability than the DKT model, and provides an evidence that the proposed regularization method in knowledge model is effective. Based on the above comparative experimental results and analyses, CMF is an appropriate model for the ensemble learning framework as a sub-model. CMF, as a single model, is less accurate than DKT in terms of prediction, in certain cases.

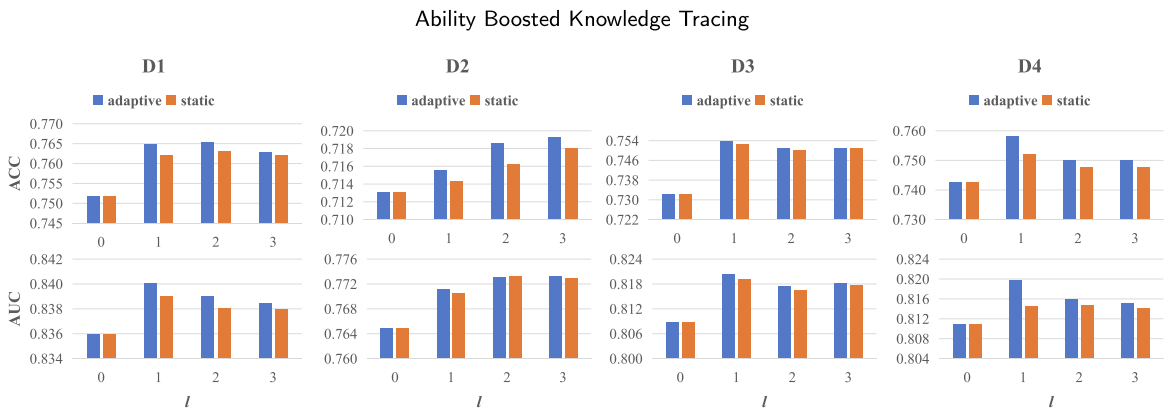

Fig. 9. Performance of ABKT-A with different depth and propagation mode measured by ACC and AUC on D1-D4.

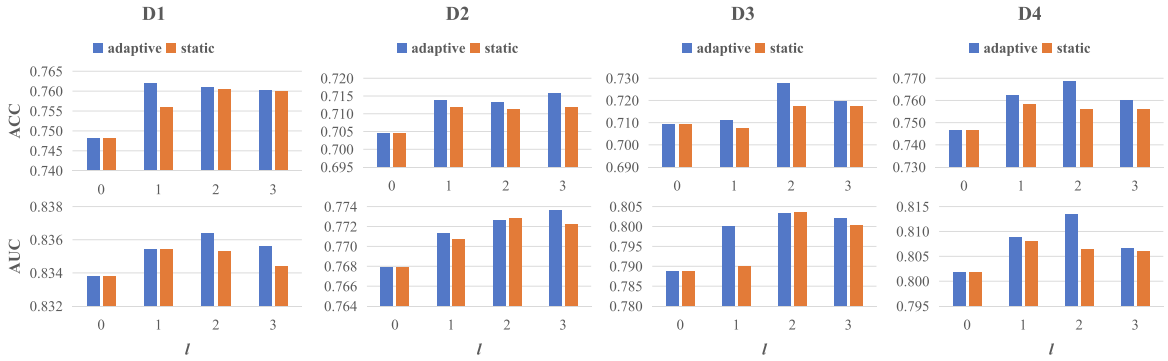

Fig. 10. Performance of ABKT-M with different depth and propagation mode measured by ACC and AUC on D1-D4.

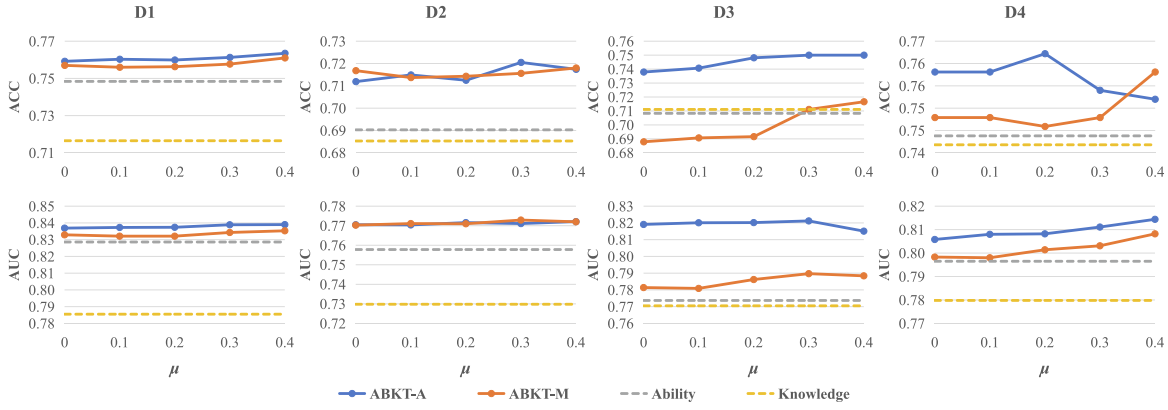

Fig. 11. Performance of ABKT-A, ABKT-M, ability model and knowledge model measured by AUC and ACC on D1-D4.

5.5.3. Effectiveness analysis of the graph-based ability model

Two key subassemblies exist in the proposed LGLA model: linear feature aggregation and adaptive propagation. To verify the effectiveness of the LGLA model, contrast experiments were conducted.The ABKT-A and ABKT-M models were implemented on all four datasets with different depths of LGLA (0–3). Some variants of ABKT-A and ABKT-M invalidated adaptive propagation. The results of all supplementary experiments are presented as bar graphs in Figs. 9 and 10. From the results in Figs. 9 and 10. First, comparing the results of the models with and without adaptive propagation (adaptive and static in Figs. 9 and 10, respectively). It is observed that the adaptive model has some advantages in terms of prediction accuracy in all four datasets. This demonstrates the effectiveness of the adaptive model in feature propagation (Eq. 8). Second, the model with

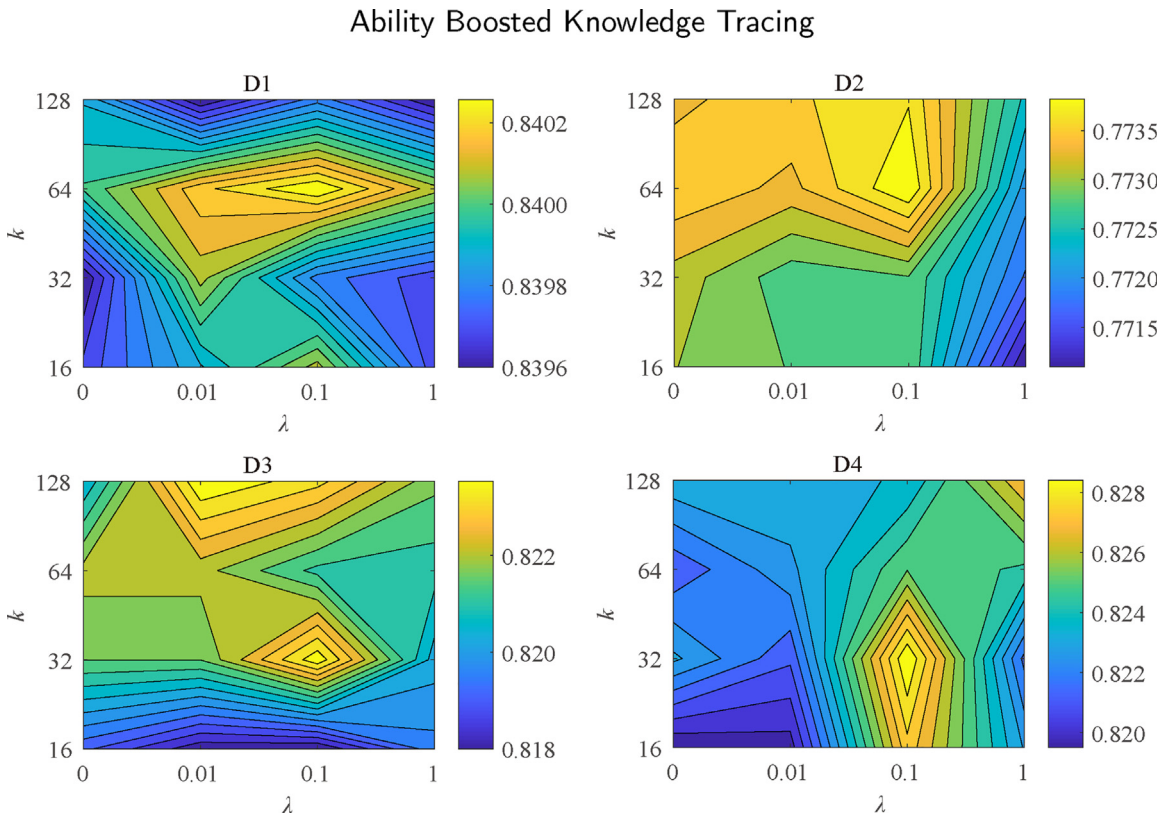

Fig. 12. Performance of ABKT-A with different embedding size (k_A) and regularization factor (λ) measured by AUC on D1-D4.

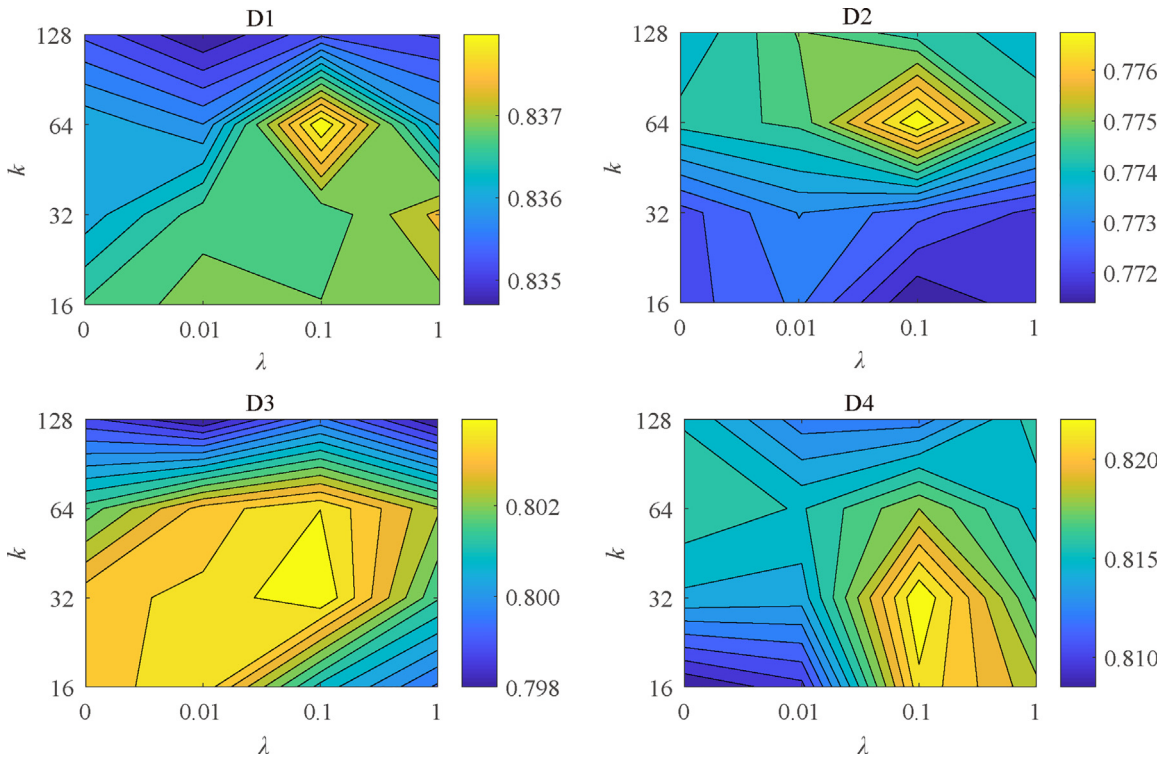

Fig. 13. Performance of ABKT-M with different embedding size (k_A) and regularization factor (λ) measured by AUC on D1-D4.

feature aggregation ($l > 0$) significantly outperformed the non-feature aggregation model ($l = 0$). This result indicates that the graph information contributes to the generation of reasonable latent representations, that the LGLA model is capable of capturing it.

5.5.4. Effectiveness analysis of boosting model

To explore the effectiveness of the knowledge and ability dual-tracing framework, ABKT-A and ABKT-M models (with different μ values ranging from 0 to 0.4) and single models (CMF and LGLA models) were constructed on all four datasets. All the experimental results are plotted in Fig. 11. First, compared with the single model (CMF and LGLA), the dual-tracing models (ABKT-A and ABKT-M) achieved better performance in all datasets. This result indicates that the key idea of this study, which is integrating knowledge and ability, is achievable. Furthermore, boosting is a valid method for fusing these factors. Second, the experimental results under different μ values show that the models with clip operation (μ>0) perform better than those with non-clip duplicates ($\mu = 0$). This demonstrates that the clip operation can facilitate effective model fusion.

5.5.5. Capacity analysis of ABKT

The model capacity has to match the scale of the training data to avoid underfitting or overfitting. In ABKT, the embedding size (k_A) in the ability module, illustrated in Section 3.4, is a hyperparameter that affects the model capacity proportionally. Correspondingly, a Gaussian-prior-based regular term is introduced, as shown in Eq. (23), where λ is the regularization parameter that balance the fidelity and regularization terms. In a set of comparative experiments, the influence of different combinations of k_A and λ on the model performance in D1-D4 was explored. The experimental results are summarized in Figs. 12 and 13. First,the different combinations do not have a discernible effect on model performance. These results indicate that the proposed ABKT model is robust against changes in k_A and λ. Furthermore, for the ABKT-A and ABKT-M models, the range of optimal values is approximately the same. In ASSISTment datasets (D1 and D2), 64 seems to be a promising

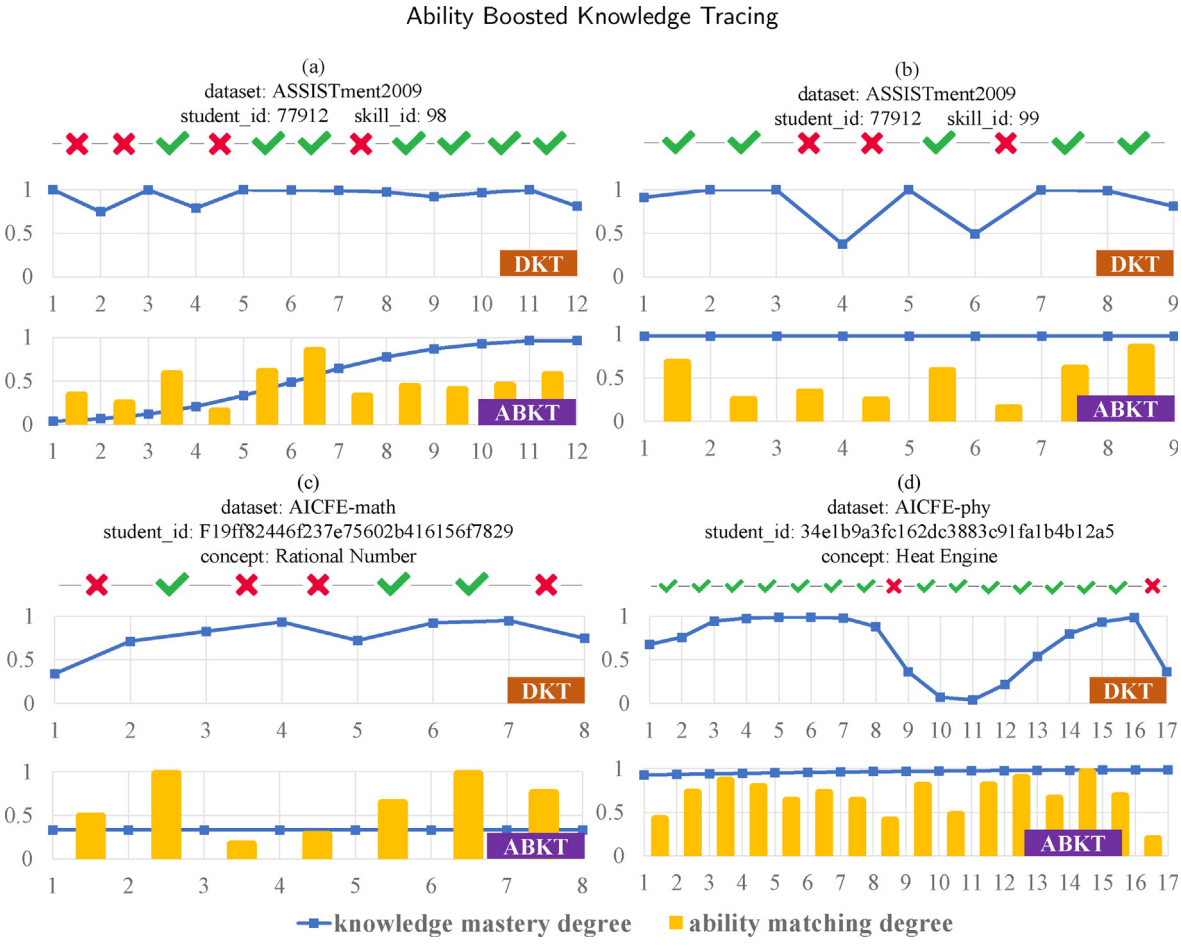

Fig. 14. Visualization of decision-making process of DKT and ABKT models on selected users and concepts..

value for hyper-parameter k_A. This value was 32 for the AICFE datasets (D3 and D4). Regarding the value of the regularization factor (λ), the best prediction accuracy appears is obtained when λ is approximately 0.1, in all four datasets.

5.5.6. Visualization of knowledge and ability dual tracing

In order to analyze the decision-making and model-fusing processes in ABKT, some experiments on feedback analysis of the learning process were conducted. First, four learning sequences based on the specific concept of random learners were selected as analysis objects. The decision-making process in the feedback prediction of the ABKT-A and DKT models is visualized in Fig. 14. The results in Fig. 14 show that, in ABKT, the feedback fits from both the knowledge and ability perspectives. In addition, the knowledge mastering degree curve of DKT is irregular because the fickle learner feedback sequences need to be fitted. In contrast, in the ABKT model, the simulation curves of the students' degree of knowledge mastery are more intuitive and readable. ABKT curves are more directional, intuitively revealing the changing trend of the learners' knowledge mastery degree over time. ABKT can analyze the reasons why learners make mistakes on specific test questions (knowledge deficit or ability shortage). Taking the student in Fig. 14 (d) as an example, from the results of ABKT, it is obvious that the degree of student's mastery of knowledge on the concept "Heat engine" is high, and the reason he/she answered two questions incorrectly is the deficiency of relevant ability. This demonstrates that ABKT has some advantages in model interpretability, and offers a different view on analyzing learner feedback in a knowledge-tracing task.

6. Conclusion

In this paper, we proposed an ability-boosted knowledge tracing algorithm, ABKT, which achieves high-quality performance providing advanced model interpretability and intelligibility. The main idea of ABKT is to introduce the ability factor as a supplement to traditional knowledge-centered KT models, to analyze the learning process of learners from two perspectives (knowledge and ability) simultaneously. ABKT constructs two parallel models for knowledge and ability, and the knowledge and ability dual-tracing framework is constructed based on ensemble learning and boosting techniques. A matrix factorization-based knowledge evolution model, CMF, is proposed to simulate the degree of knowledge internalization during the learning process based on constructive learning theory. In addition, the LGLA model is proposed, which utilizes graph-structured learner interaction data to construct learner and item latent ability features. LGLA simplifies the graph neural networks by linearizing the feature aggregation layer, allowing the model to be optimized by stochastic or mini-batch methods, further improving model training efficiency and usability. Experimental results on four manifold real-world datasets demonstrate that ABKT has an advantage in terms of prediction accuracy compared to state-of-the-art KT models. Furthermore, the CMF model exhibits impressive generalization ability. The proposed LGLA model and the knowledge and ability dual-tracing framework were further validated.

Future work will explore preferable frameworks to integrate knowledge and ability factors, such as the causal inference framework. KT models will be constructed conforming to pedagogical and psychological theories, such as the Ebbinghaus forgetting curve, to make the model more reasonable and explicable. In the long term, exploring methods or mechanisms to explicitly define and measure learners' abilities is essential. This technique will enable intelligent e-learning systems to diagnose learner states more accurately, and provide more precise and effective personalized services for learners. It is expected that this will have an enormous positive impact on the field.

CRediT authorship contribution statement

Sannyuya Liu: Supervision, Writing - original draft, Writing - review & editing. **Jianwei Yu:** Investigation, Methodology, Writing - review & editing. **Qing Li:** Visualization, Data curation. **Ruxia Liang:** Validation, Writing - review & editing. **Yunhan Zhang:** Formal analysis. **Xiaoxuan Shen:** Conceptualization, Methodology, Software, Writing - original draft. **Jianwen Sun:** Conceptualization, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research is supported by the National Key R&D Program of China (2020AAA0108804); National Natural Science Foundation of China (62107017, 62077021, 61977030, 61937001); China Postdoctoral Science Foundation (2020M682454); MOE and China Mobile Joint Research Fund (MCM20200406); Teaching Reform Research Project for Postgraduates of CCNU (2020JG14); and Teaching Research Project for Undergraduates of CCNU (202009).

References

[1] S. Arlot, A. Celisse, et al, A survey of cross-validation procedures for model selection, Stat. Surveys 4 (2010) 40–79.
[2] R.S. d Baker, A.T. Corbett, V. Aleven, More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing, in: International conference on intelligent tutoring systems, Springer, 2008. pp. 406–415..
[3] H. Cen, K. Koedinger, B. Junker, Learning factors analysis–a general method for cognitive model evaluation and improvement, International Conference on Intelligent Tutoring Systems, Springer (2006) 164–175.
[4] P. Chen, Y. Lu, V.W. Zheng, Y. Pian, Prerequisite-driven deep knowledge tracing, in: 2018 IEEE International Conference on Data Mining (ICDM), IEEE, 2018, pp. 39–48.
[5] Y. Choi, Y. Lee, J. Cho, J. Baek, B. Kim, Y. Cha, D. Shin, C. Bae, J. Heo, Towards an appropriate query, key, and value computation for knowledge tracing, in: Proceedings of the Seventh ACM Conference on Learning@ Scale, 2020, pp. 341–344.
[6] A.T. Corbett, J.R. Anderson, Knowledge tracing: Modeling the acquisition of procedural knowledge, User modeling and user-adapted interaction 4 (1994) 253–278.
[7] B. Deonovic, M. Yudelson, M. Bolsinova, M. Attali, G. Maris, Learning meets assessment, Behaviormetrika 45 (2018) 457–474.
[8] Desmarais, M.C., d Baker, R.S., 2012. A review of recent advances in learner and skill modeling in intelligent learning environments. User Modeling and User-Adapted Interaction 22, 9–38..
[9] S.E. Embretson, S.P. Reise, Item response theory, Psychology Press, 2013.
[10] C.T. Fosnot, Constructivism: Theory, perspectives, and practice, Teachers College Press, 2013.
[11] A. Ghosh, N. Heffernan, A.S. Lan, Context-aware attentive knowledge tracing, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 2330–2339.
[12] A. Ghosh, J. Raspat, A. Lan, Option tracing: Beyond correctness analysis in knowledge tracing, International Conference on Artificial Intelligence in Education, Springer. (2021) 137–149.
[13] R.K. Hambleton, H. Swaminathan, Item response theory: Principles and applications, Springer Science & Business Media, 2013.
[14] N.T. Heffernan, C.L. Heffernan, The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching, Int. J. Artif. Intell. Educ. 24 (2014) 470–497.
[15] Z. Huang, Q. Liu, Y. Chen, L. Wu, K. Xiao, E. Chen, H. Ma, G. Hu, Learning or forgetting? a dynamic approach for tracking the knowledge proficiency of students, ACM Trans. Inform. Syst. (TOIS) 38 (2020) 1–33.
[16] Y. Huo, D.F. Wong, L.M. Ni, L.S. Chao, J. Zhang, Knowledge modeling via contextualized representations for lstm-based personalized exercise recommendation, Inf. Sci. 523 (2020) 266–278.
[17] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016. arXiv preprint arXiv:1609.02907..
[18] C. Liu, X. Li, Multi-factor memory attentive model for knowledge tracing, Asian Conference on Machine Learning, PMLR (2021) 856–869.
[19] Q. Liu, Z. Huang, Y. Yin, E. Chen, H. Xiong, Y. Su, G. Hu, Ekt: Exercise-aware knowledge tracing for student performance prediction, IEEE Trans. Knowl. Data Eng. 33 (2019) 100–115.
[20] S. Liu, R. Zou, J. Sun, K. Zhang, L. Jiang, D. Zhou, J. Yang, A hierarchical memory network for knowledge tracing, Expert Syst. Appl. 177 (114935) (2021) 787.
[21] Liu, Y., Yang, Y., Chen, X., Shen, J., Zhang, H., Yu, Y., 2020. Improving knowledge tracing via pre-training question embeddings, pp. 1556–1562. doi:10.24963/ijcai.2020/216..
[22] G.N. Masters, A rasch model for partial credit scoring, Psychometrika 47 (1982) 149–174.
[23] A. Mnih, R.R. Salakhutdinov, Probabilistic matrix factorization, Adv. Neural Inform. Process. Syst. 20 (2007) 1257–1264.
[24] K. Nagatani, Q. Zhang, M. Sato, Y.Y. Chen, F. Chen, T. Ohkuma, Augmenting knowledge tracing by considering forgetting behavior, The world wide web conference (2019) 3101–3107.
[25] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, ICML'10 (2010) 807–814.
[26] S. Pandey, G. Karypis, A self-attentive model for knowledge tracing, International Educational Data Mining Society, 2019.
[27] S. Pandey, J. Srivastava, Rkt: Relation-aware self-attention for knowledge tracing, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 1205–1214.
[28] Z.A. Pardos, N. Heffernan, C. Ruiz, J. Beck, Effective skill assessment using expectation maximization in a multi network temporal bayesian network, in: Proceedings of the Young Researchers Track at the 9th International Conference on Intelligent Tutoring Systems, Citeseer, 2008.
[29] Z.A. Pardos, N.T. Heffernan, Modeling individualization in a bayesian networks implementation of knowledge tracing, International Conference on User Modeling, Adaptation, and Personalization, Springer (2010) 255–266.
[30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al, Pytorch: An imperative style, high-performance deep learning library, Adv. Neural Inform. Process. Syst. 32 (2019) 8026–8037.
[31] P.I. Pavlik, L.G. Eglington, L.M. Harrell-Williams, Logistic knowledge tracing: A constrained framework for learner modeling, IEEE Trans. Learn. Technol. (2021).
[32] P.I. Pavlik Jr, H. Cen, K.R. Koedinger, Performance factors analysis–a new alternative to knowledge tracing. Online Submission, 2009..
[33] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. Guibas, J. Sohl-Dickstein, Deep knowledge tracing, in: Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1, 2015, pp. 505–513..
[34] S. Pu, M. Yudelson, L. Ou, Y. Huang, Deep knowledge tracing with transformers, International Conference on Artificial Intelligence in Education, Springer (2020) 252–256.
[35] S. Shen, Q. Liu, E. Chen, Z. Huang, W. Huang, Y. Yin, Y. Su, S. Wang, Learning process-consistent knowledge tracing, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 1452–1460.
[36] S. Shen, Q. Liu, E. Chen, H. Wu, Z. Huang, W. Zhao, Y. Su, H. Ma, S. Wang, Convolutional knowledge tracing: Modeling individualization in student learning process, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 1857–1860.
[37] X. Shen, B. Yi, H. Liu, W. Zhang, Z. Zhang, S. Liu, N. Xiong, Deep variational matrix factorization with knowledge embedding for recommendation system, IEEE Trans. Knowl. Data Eng. 33 (2021) 1906–1918.
[38] D. Shin, Y. Shim, H. Yu, S. Lee, B. Kim, Y. Choi, Saint+: Integrating temporal features for ednet correctness prediction, in: LAK21: 11th International Learning Analytics and Knowledge Conference, 2021, pp. 490–496.
[39] R.E. Slavin, Educational psychology: Theory and practice, 2019.
[40] X. Song, J. Li, Y. Tang, T. Zhao, Y. Chen, Z. Guan, Jkt: A joint graph convolutional network based deep knowledge tracing, Inf. Sci. 580 (2021) 510–523.
[41] J.J. Vie, H. Kashima, Knowledge tracing machines: Factorization machines for knowledge tracing, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 750–757.
[42] T. Wang, F. Ma, J. Gao, Deep hierarchical knowledge tracing, in: Proceedings of the 12th International Conference on Educational Data Mining, 2019.
[43] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, K. Weinberger, Simplifying graph convolutional networks, International conference on machine learning, PMLR (2019) 6861–6871.
[44] Z. Wu, T. He, C. Mao, C. Huang, Exam paper generation based on performance prediction of student group, Inf. Sci. 532 (2020) 72–90.
[45] C.K. Yeung, 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.
[46] B. Yi, X. Shen, H. Liu, Z. Zhang, W. Zhang, S. Liu, N. Xiong, Deep matrix factorization with implicit feedback embedding for recommendation system, IEEE Trans. Industr. Inf. 15 (2019) 4591–4601.

[47] J. Zhang, X. Shi, I. King, 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.
[48] K. Zhang, Y. Yao, A three learning states bayesian knowledge tracing model, Knowl.-Based Syst. 148 (2018) 189–201.
[49] M. Zhang, X. Zhu, C. Zhang, Y. Ji, F. Pan, C. Yin, Multi-factors aware dual-attentional knowledge tracing, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 2588–2597.
[50] X. Zhang, J. Zhang, N. Lin, X. Yang, Sequential self-attentive model for knowledge tracing, International Conference on Artificial Neural Networks, Springer (2021) 318–330.