Expert Systems With Applications 237 (2024) 121637

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Global and local neural cognitive modeling for student performance prediction

Yu Su [a,b], Shuanghong Shen [d,*], Linbo Zhu [c], Le Wu [e], Zhenya Huang [d], Zeyu Cheng [f], Qi Liu [c,d], Shijin Wang [f]

[a] School of Computer Science and Artificial Intelligence, Hefei Normal University, China
[b] Key Laboratory of Philosophy and Social Science of Anhui Province on Adolescent Mental Health and Crisis Intelligence Intervention, China
[c] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, China
[d] Anhui Province Key Lab. of Big Data Analysis and Application, School of Data Science, University of Science and Technology of China, China
[e] Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology, China
[f] State Key Laboratory of Cognitive Intelligence, iFLYTEK Co., Ltd, China

ARTICLE INFO

Keywords:
Knowledge tracing
Cognitive diagnosis
Student performance prediction
Student modeling

ABSTRACT

Student performance prediction is a critical task in supporting decision-making for Intelligent Tutoring Systems (ITS). Correct predictions of student performance are prerequisites for ITS to supply intelligent services and optimize learning efficiency, e.g., recommending the most appropriate learning resources for each student. Existing methods mainly include cognitive diagnosis and knowledge tracing, both of which focus on students' cognitive modeling based on their interactions on a sequence of items and give predictions by assessing if their cognitive states can meet the item requirements. Specifically, cognitive diagnosis only considers students' global static cognitive states, while knowledge tracing focuses on students' local dynamics. However, both global and local features are critical for predicting student performance. Therefore, in this paper, we propose a novel **G**lobal and **L**ocal **N**eural **C**ognitive (GLNC) model to capture both global and local features in student-item interactions for more accurate predictions. Specifically, we first learn students' global cognitive level according to all student-item interactions. Then, we propose a self-attentive encoder based on the scaled dot-product attention mechanism to extract the local cognitive dynamics and the dependencies between students' recent interactions. Finally, to obtain better predictions, we design a fused gate based on the similarity between students' recently responded items and the item to be predicted to adaptively combine the global and local features. To evaluate the effectiveness of GLNC, we compare it with both cognitive diagnosis and knowledge tracing methods. All experiments are conducted on three public datasets that contain real student-item interactions on mathematics or language courses from various ITS. The experimental results demonstrate that GLNC achieves an average score of 0.7810 on the AUC metric, 0.7627 on the ACC metric, 0.3987 on the RMSE metric, 0.2023 on the r^2 metric, respectively achieving an average improvement of 1.84%, 1.07%, 1.87%, and 11.38% in contrast to existing state-of-the-art methods (i.e., NCD and LPKT). Moreover, we further analyze the performance of GLNC under different probabilities of guessing and slipping, the results indicate that GLNC is more robust against the influence of noisy data while considering both global and local features. Benefiting from the superior accuracy and stability, our proposed GLNC has a wide range of potential implications for ITS, which can be easily applied to improve students' learning efficiency and experience.

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail addresses: yusu@hfnu.edu.cn (Y. Su), closer@mail.ustc.edu.cn (S. Shen), lbzhu@iai.ustc.edu.cn (L. Zhu), lewu.ustc@gmail.com (L. Wu), huangzhy@ustc.edu.cn (Z. Huang), zycheng2@iflytek.com (Z. Cheng), qiliuql@ustc.edu.cn (Q. Liu), sjwang3@ifytek.com (S. Wang).

[1] https://eedi.com/
[2] https://new.assistments.org/
[3] https://www.zhixue.com/login.html

https://doi.org/10.1016/j.eswa.2023.121637
Received 26 March 2023; Received in revised form 12 September 2023; Accepted 13 September 2023
Available online 22 September 2023
0957-4174/

1. Introduction

With the rapid development of artificial intelligence, numerous Intelligent Tutoring Systems (ITS) (Chrysafiadi & Virvou, 2013) have emerged that are designed to realize high-quality online education, such as Eedi,[1] Assistments,[2] ZhiXue.[3] There is empirical evidence that indicates ITS are nearly as effective as human tutoring (VanLehn, 2011), because they can offer necessary instructions as human tutors. Student performance prediction, which aims to predict students' performance on future items according to their previous item interactions (Deeva et al., 2022; Wang, Mei et al., 2021), is one of the fundamental tasks for ITS to supply intelligent services. Specifically, ITS have recorded a lot of daily data about student-system interactions, which can be further utilized to assess students' future performance. Actually, Intelligent Tutoring Systems (ITS) are multi-module expert systems (Yazdani, 1986), where student performance prediction has been an absolutely necessary module for ITS. By predicting student performance in advance, ITS can make adaptive feedback to various students and customize the most suitable learning schemes for them, thereby improving students' learning efficiency and experience by a large margin (Ding & Larson, 2020; Liu et al., 2022; Xie et al., 2016). Moreover, based on precise and reliable prediction results, ITS can provide timely and necessary assistance for students struggling with specific problems to avoid potential course failures (Deeva et al., 2022; Kotsiantis et al., 2010).

To better illustrate the task of student performance prediction and its critical role in ITS, we show a toy example in Fig. 1. Specifically, there are four students and their corresponding item interactions, the ITS predicts their performance on potential future items and recommends the most suitable one for each student based on the predictions (Liang et al., 2023; Liu et al., 2019). In the literature, most existing methods conduct student performance prediction through students' cognitive modeling (i.e., measuring their knowledge proficiencies on knowledge concepts, such as *Adding and Subtracting Frac*) (Wang, Ma, Zhao, Li et al., 2022), which can be mainly divided into cognitive diagnosis (Wang et al., 2020) and knowledge tracing (Corbett & Anderson, 1994). Specifically, cognitive diagnosis models consider both the student factor and the item factor, which use either manual-designed or learnable interaction functions (e.g., the logistic function, and the neural network) to measure students' global static cognitive states based on their previous interactions with different items (Liu, 2021). In contrast, knowledge tracing models focus on students' historical interaction sequences and monitor their local dynamic cognitive states (Sun et al., 2022). After cognitive modeling, both cognitive diagnosis and knowledge tracing methods calculate if the student's cognitive states can meet the item's requirements for making predictions.

Generally, the learning process is ongoing (Ghosh et al., 2020), and students' cognitive states are not stable, which will be increased by learning and decreased due to forgetting (Wang, Ma et al., 2021). However, since cognitive diagnosis models assume that students' cognitive states are static, they fail to measure the above dynamics of learning. Besides, cognitive diagnosis models consider each student-item interaction independently, thus they cannot capture the inner relations of different interactions. In contrast, knowledge tracing models consider the dependencies between different interactions by the manner of sequential modeling (Gan et al., 2022), the dynamic evolution process of the cognitive state is also captured in this manner. Unfortunately, knowledge tracing is heavily sensitive to the interaction sequence, and therefore not stable, minor changes in the interaction sequence can cause significant differences in the predictions of knowledge tracing methods. Therefore, there is still room for improvement in both cognitive diagnosis and knowledge tracing.

Actually, both the global static features of students and the local dynamics of the learning process are of great significance for student performance prediction. Specifically, from the perspective of global cognitive modeling, we can infer that Alice in Fig. 1 has a higher probability of passing the item i_8 than Jack, even though neither of them has responded to an item similar to i_8 (i.e., relates to the same knowledge concept). Because Alice has got right on all the historical items, while Jack has only mastered about half of them. From the perspective of the local dynamic cognitive modeling, we argue that Jasper should also do well on i_8 but Lucy may fail. Although Jasper and Lucy have similar global performance, Jasper's most recent interaction on the item i_7 that is similar to i_8 was correct, while Lucy was the opposite. Therefore, we tend to conclude that Jasper can correctly answer i_8 and Lucy has a high risk of failure. In summary, students' global characteristics indicate their general cognitive level, on the basis of which we can make predictions on the majority of items. Meanwhile, the local dynamics of learning provide a more fine-grained inference of students' performance on specific items. Therefore, to achieve more precise and reliable prediction results, we should consider both global and local features in student-item interactions together.

In this paper, combining the complementary advantage of cognitive diagnosis and knowledge tracing, we explore capturing global and local features together in student-item interactions and making a trade-off between them to improve the validity of student performance prediction. To achieve this goal, we propose a novel **G**lobal and **L**ocal **N**eural **C**ognitive (GLNC) model. Specifically, to conduct global cognitive modeling, following existing cognitive diagnosis models, we first present neural networks to model each student-item interaction and learn a global student vector from all interactions. Then, to summarize the local dynamics, we design a self-attentive encoder based on the scaled dot-product attention mechanism to capture the dependencies between items. Finally, we propose the fused gate to combine the global and local features based on the correlation between students' recently responded items and the item for predicting. In this way, we can predict student performance based on the difference between the students' combined cognitive states and the requirements of the given items to be answered.

Our code has been published in Code Ocean.[4] The main contributions of this paper are as follows:

- We analyze the complementary advantage of cognitive diagnosis and knowledge tracing on the student performance prediction task, where the former focuses on students' global static cognitive states and the latter captures students' local dynamics.
- We propose a novel **G**lobal and **L**ocal **N**eural **C**ognitive (GLNC) model to extract both global and local features in student-item interactions and make a trade-off between them to improve the accuracy and reliability of student performance prediction.
- We conduct extensive experiments on three real-world datasets to evaluate the effectiveness of GLNC, and the results illustrate that GLNC outperforms existing state-of-the-art methods with superior robustness.
- Our proposed method can be applied in existing ITS and improve students learning efficiencies, which also has the potential to support more personalized learning services.

2. Related works

In this section, we will briefly introduce existing related works of the student performance prediction task from two categories: (1) cognitive diagnosis that learns students' global cognitive states by specific student-item interaction functions. and (2) knowledge tracing that captures the local dynamics and dependencies in learning by sequential modeling. Besides, we also describe how our work differs from existing related works, i.e., it explicitly considers both the global features of students and the local dynamics of learning in parallel.

[4] https://codeocean.com/capsule/9537102/tree/v1

(a) The student performance prediction task based on students' historical item interactions.

Item	Item Content	Knowledge Concept
i_1	5/7 - 1/4 = ?	Adding and Subtracting Fractions
i_2	(1/6) × 6 = ?	Multiplying Fractions
i_3	(-0.2) ÷ (-1/2) = ?	Dividing Fractions
i_4	0.7 + 6/5 −5% = ?	Adding and Subtracting with Fractions, Decimals, and Percentage Equivalence
i_5	4 × 2 = ?	Multiplying and Dividing with Intergers
i_6	8 × 3 = ?	Multiplying and Dividing with Intergers
i_7	0.4 = 2/5 = ? %	Fractions, Decimals and Percentage Equivalence
i_8	0.6 = 3/5 = ? %	Fractions, Decimals and Percentage Equivalence

(b) The item contents and their related knowledge concepts.

Fig. 1. A toy example of the student performance prediction task which aims to predict students' responses on future items according to their previous item interactions.

2.1. Cognitive diagnosis

Cognitive diagnosis aims to measure students' cognitive abilities based on student-item interactions (Liu, 2021). Existing cognitive diagnosis models (CDMs) include traditional CDMs and deep learning-based CDMs, the difference is whether they used neural networks to model student-item interactions. Concretely, traditional CDMs can be classified into discrete CDMs that consider discrete students' cognitive states and continuous CDMs that treat students' cognitive states as continuous values. In these models, the student-item interactions are modeled by manually-designed functions, such as the probability function in the Deterministic Inputs, Noisy And gate model (DINA) (De La Torre, 2011) and the logistic function in the item response theory (IRT) model (Hambleton et al., 1991). However, either the probability function or the logistic function is too simple to capture the complex student-item interactions. Therefore, Wang et al. (2020) proposed the Neural Cognitive Diagnosis (NCD), which utilized non-linear neural networks to model student-item interactions more sufficiently. As neural networks can approximate any continuous functions (Hornik et al., 1989), NCD better captured the student-item interaction and achieved better performance than traditional CDMs. NCD's general framework can be formulated as:

$$p_i(r = 1|\boldsymbol{s}) = \varphi_n(\dots\varphi_1(\boldsymbol{s}, i)), \tag{1}$$

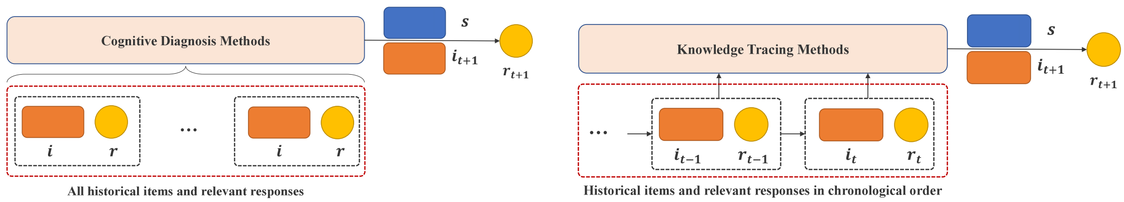
where φ denotes the neural network used to model the complex student-item interactions. $\boldsymbol{s}$ is the cognitive state of students, i means the feature of items. Some latest works extended NCD from different aspects. For example, Zhou et al. (2021) considered the influence of educational contexts (such as the highest education degree of students' parents, and students' science activities experience out of school), they presented a hierarchical attentive network to measure the impact of educational context. Su et al. (2022) focused on the high-order relations between student-item interactions. They designed a heterogeneous cognitive graph to discover the high-order relations among students, items, and knowledge concepts. Qi et al. (2023) quantified the interaction between knowledge concepts, as well as the correlations between items and knowledge concepts. They achieved more interpretable diagnosis results for students. Song et al. (2023) extended MIRT under the framework of deep neural networks, which effectively captured cross-modal semantic information for predicting student performance. In summary, as shown in Fig. 2(a), CDMs first learn a global static student vector $\boldsymbol{s}_g$ that represents students' cognitive states from all student-item interactions. They, CDMs predict students' performance by measuring if their cognitive levels can meet the requirement of the item.

2.2. Knowledge tracing

In contrast to cognitive diagnosis, knowledge tracing utilizes the sequence of students' interactions for their cognitive modeling (Song et al., 2022). Existing knowledge tracing models (KTMs) include BKT-based models and deep learning-based models, the difference is whether they used neural networks to complete sequence modeling. Bayesian Knowledge Tracing (BKT) utilized the Hidden Markov Model (HMM) to capture students' cognitive states (Corbett & Anderson, 1994). Piech, Bassen et al. (2015) presented Deep Knowledge Tracing (DKT), which utilized Recurrent Neural Networks (RNNs) (Williams & Zipser, 1989) to model the sequence of student-item interactions. DKT model can be formulated as:

$$\begin{aligned} \boldsymbol{s}_t &= tanh(\boldsymbol{W}_{hs}\boldsymbol{x}_t + \boldsymbol{W}_{hs}\boldsymbol{s}_{t-1} + \boldsymbol{b}_h), \\ p_i(r = 1|\boldsymbol{s}_t) &= \sigma(\boldsymbol{W}_{ys}\boldsymbol{s}_t + \boldsymbol{b}_y), \end{aligned} \tag{2}$$

where $\boldsymbol{W}_{hs}$, $\boldsymbol{b}_h$, $\boldsymbol{W}_{ys}$, and $\boldsymbol{b}_y$ are learnable parameters, $\boldsymbol{s}$ is the cognitive state of students. Ding and Larson (2020) further studied the uncertainty of DKT, which provided the confidence level for DKT's predictions and improved DKT's robustness. Then, more deep learning-based approaches have been applied in students' sequential modeling for KT, including graph-based models (Nakagawa et al., 2019; Wu et al., 2022), Attention-based models (Pandey & Karypis, 2019; Wang, Ma, Zhao, Yang et al., 2022), and learning process-based models (Shen et al., 2021a; Wang, Ma, Zhao, Li et al., 2022). Specifically, Nakagawa et al. (2019) presented graph-based knowledge tracing (GKT), which utilized the potential graph structure of knowledge concepts to measure item relations. Pandey and Karypis (2019) introduced the self-attention mechanism in Transformer (Vaswani et al., 2017). Ghosh et al. (2020) proposed utilizing the IRT model to construct item embeddings and designed an encoder–decoder architecture to realize KT. Shen et al. (2021a) proposed to model students' learning gains and forgetting for calculating their dynamic ability. Liu, Zou et al. (2021) explored the human memory mechanism for knowledge tracing. Wang, Ma, Zhao, Li et al. (2022) designed some calibration methods to alleviate the subjective bias introduced by experts. In summary, as shown in Fig. 2(b), knowledge tracing methods learn a local dynamic student vector s that represented students' cognitive states from their historical interaction sequence through recurrent neural networks. Analogously, students' cognitive states and item requirements are compared for performance prediction.



(a) Diagram of Cognitive Diagnosis Methods. (b) Diagram of Knowledge Tracing Methods.

Fig. 2. The illustration of cognitive diagnosis and knowledge tracing for the student performance prediction task. The predicting process contains two stages: **(1)** modeling students' cognitive states from their historical item interactions and **(2)** making predictions based on students' cognitive states and the item to be predicted. As marked by the red box, the left figure shows the non-serialized global input, as cognitive diagnosis methods learned the global cognitive state from students' historical item interactions for all future items. The right figure shows the serialized local input, as knowledge tracing methods summarized the specific local cognitive state by sequence modeling for each item to be predicted.

Table 1
Mathematical notations and descriptions.

Notations	Descriptions
$\mathbb{S}, \mathbb{I}, \mathbb{K}$	The set of students, items, and knowledge concepts.
L, J, M	The number of students, items, and knowledge concepts.
$\boldsymbol{I} \in \mathbb{R}^{J\times d_i}$	The embedding matrix of all items.
$\boldsymbol{K} \in \mathbb{R}^{M\times d_k}$	The embedding matrix of all knowledge concepts.
$\boldsymbol{Q}$	The Q-matrix.
r	The binary response.
$i, \boldsymbol{i} \in \mathbb{R}^{d_i}$	The item and its corresponding embedding.
$k_i, \boldsymbol{k}_i \in \mathbb{R}^{d_k}$	The knowledge concept and its embedding of item i .
$\boldsymbol{x} \in \mathbb{R}^{d_x}$	The enhanced item vector.
$\boldsymbol{x}^r \in \mathbb{R}^{d_i}$	The item-response pair.
$\boldsymbol{s}^g \in \mathbb{R}^{d_s}$	The global cognitive vector.
$\boldsymbol{s}^l \in \mathbb{R}^{d_s}$	The local cognitive vector.
$\boldsymbol{att}$	The item similarity.
$\boldsymbol{\Gamma}$	The fused gate.

2.3. Differences with existing works

As we claimed above, CDMs learn students' global cognitive states through the student-item interaction function. They assumed that students' cognitive states are static and considered every interaction independently. KTMs capture the local dynamics and dependencies in learning by sequential modeling. They assumed that students' cognitive states are dynamic and considered dependent interactions in a sequence. It is worth noting that some existing works have tried to combine cognitive diagnosis and knowledge tracing with the aim of improving interpretability. For example, Gan et al. (2022) leveraged the interpretability of IRT in the cognitive modeling process to promote the robustness of KTMs. Similarly, Su et al. (2021) introduced the item parameters learned by the MIRT model into the DKT model to enhance its performance. However, they are limited in implicit local parameter sharing.

Our proposed method differs from existing methods in that we explicitly consider both the global features of students and the local dynamics of learning in parallel. Specifically, given students' previous item interactions, we first learn their global cognitive states. Then, to predict students' responses to a specific item, we also model their local dynamics by the most recent interactions. Finally, we will make a trade-off between the global and local features to give the predictions. Therefore, our proposed model is superior in both accuracy and robustness.

3. Problem definition

In the online learning system, students are assigned to answer different items to gain knowledge. We denote the sequence of item-response interactions of the student s as $\mathbb{X} = \{(s, i_1, r_1), (s, i_2, r_2), \dots, (s, i_t, r_t), \dots, (s, i_T, r_T)\}$, where i_t is the tth item and the binary variable r_t is the corresponding response of i_t:

$$r_t = \begin{cases} 1, & \text{if} \quad \text{student } s \text{ correctly answered item } i_t, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Supposing there are L students in the learning system and the student set is $\mathbb{S} = \{s_1, s_2, \dots, s_L\}$. The item set is $\mathbb{I} = \{i_1, i_2, \dots, i_J\}$ with J unique items, while The knowledge concept set is denoted as $\mathbb{K} = \{k_1, k_2, \dots, k_M\}$ with M knowledge concepts. Moreover, the Q-matrix $\boldsymbol{Q} \in \mathbf{R}^{J\times M}$ is defined by educational experts for each ITS (de la Torre & Chiu, 2016), which indicates the priori relations between items and knowledge concepts. $\boldsymbol{Q}$ is usually utilized as instructions for assigning items to students. For example, if the student needs to practice a specific knowledge concept, we can choose the related items according to the Q-matrix (Wang, Ma, Zhao, Li et al., 2022). Specifically, $\boldsymbol{Q}$ is made up of zeros and ones, where $\boldsymbol{Q}(j, m) = 1$ if item i_j contains the knowledge concept k_m, otherwise $\boldsymbol{Q}(j, m) = 0$.

Following existing studies (Shen et al., 2022; Wang et al., 2020), we use a cognitive vector $\boldsymbol{s} \in \mathbb{R}^{d_s}$ (here d_s is the dimension of $\boldsymbol{s}$, which is defined in Section 5.4) to represent the cognitive ability of the specific student s. The embedding matrix $\boldsymbol{I} \in \mathbb{R}^{J\times d_i}$ is used to represent all items. Then, for item i_j, we can directly obtain its embedding $\boldsymbol{i}_j$ from $\boldsymbol{I}$. Similarly, we use an embedding matrix $\boldsymbol{K} \in \mathbb{R}^{M\times d_k}$ to represent all knowledge concepts and we can directly obtain the embedding $\boldsymbol{k}_m$ of the knowledge concept k_m from $\boldsymbol{K}$. The mathematical notations utilized throughout our paper are summarized in Table 1.

Then, we can formulate the task of student performance prediction as predicting the probability that the student will correctly respond to a new item i_{new}, formally as follows:

Problem definition. Given the interaction sequence of the student s, i.e., $\mathbb{X} = \{(s, i_1, r_1), (s, i_2, r_2), \dots, (s, i_t, r_t), \dots, (s, i_T, r_T)\}$ and the Q-matrix $\boldsymbol{Q}$, where i_t and r_t respectively mean the item and s's response at timestep t, the student performance prediction task aims to measure the probability that s can get a correct response on a new item i_{new}, i.e., $P(r_{new} = 1 | i_{new}, \mathbb{X}, \boldsymbol{Q})$.

4. The GLNC model

In this section, we will present the GLNC model in detail. The model architecture is shown in Fig. 3 and Algorithm 1. Specifically, we will start by introducing how to model students' global cognitive states based on their previous item interactions. Then, we propose the self-attention encoder to learn the inner relation between recent student-item interactions and extract the local dynamics in the interaction sequence. Finally, we present how to adaptively combine the global cognitive states and local cognitive dynamics together for making predictions on future items.

4.1. Global cognitive modeling

To predict student performance, we should first understand the student's cognitive states. Thereby, existing cognitive diagnosis methods devote great efforts to cognitive modeling (Liu, 2021; Wang et al., 2020). As our primary goal is to combine both global and local features in student-item interactions for better predictions, global cognitive modeling is still the basis of GLNC. Specifically, in this global cognitive

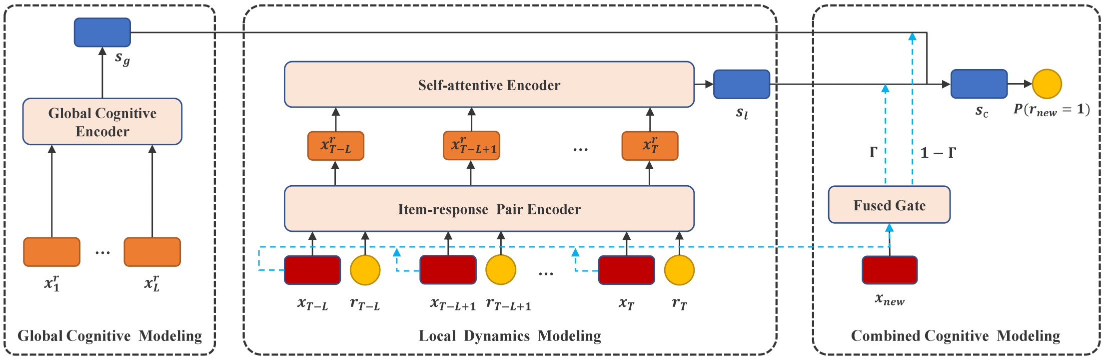


Fig. 3. The framework of the GLNC model, which includes three modules: (1) Global Cognitive Modeling that captures students' global static cognitive features, (2) Local Dynamics Modeling that measures students' local dynamic cognitive features in the interaction sequence, and (3) Combined Cognitive Modeling that adaptively combines the global and local cognitive features together for a more comprehensive understanding of students' cognitive states.

modeling module, similar to existing works, we will learn the global cognitive vector for each student based on the interactions on different items.

Generally, a specific student-item interaction contains four factors: (1) the student s, (2) the item i_j, (3) the knowledge concept k_{i_j} of i_j, and (4) the student's response a_{i_j} to i_j. For the student factor s, we randomly initialize a global cognitive vector $\boldsymbol{s}_g$ to characterize the students' global cognitive states, which will be learned from all the student's item interactions in the training process. For the item and the knowledge concept factor, we combine the item embedding $\boldsymbol{i}_j$ and the knowledge concept embedding $\boldsymbol{k}_{i_j}$ by a multi-layer perceptron (MLP) to enhance the item representation:

$$\boldsymbol{x}_j = \boldsymbol{W}_1^T(\boldsymbol{i}_j \oplus \boldsymbol{k}_{i_j}) + \boldsymbol{b}_1, \tag{4}$$

where $\boldsymbol{x}_j \in \mathbb{R}^{d_x}$ denotes the enhanced item vector, $\oplus$ means vector concatenation, $\boldsymbol{W}_1 \in \mathbb{R}^{(d_i+d_k)\times d_x}$ and $\boldsymbol{b}_1 \in \mathbb{R}^{d_x}$ are trainable parameters. In the experiment, we will show that the combination of $\boldsymbol{i}_j$ and $\boldsymbol{k}_{i_j}$ helps to enhance the predictions, as considering them together can more completely reflect the characteristics of the item. In addition, for the response factor, we respectively design the right layer and the wrong layer to distinguish the different effects of two binary responses and represent the item-response pair:

$$\boldsymbol{x}_j^r = \begin{cases} \boldsymbol{W}_r^T(\boldsymbol{x}_j) + \boldsymbol{b}_r, & \text{if } r_{i_j} = 1, \\ \boldsymbol{W}_w^T(\boldsymbol{x}_j) + \boldsymbol{b}_w, & \text{if } r_{i_j} = 0, \end{cases} \tag{5}$$

where $\boldsymbol{x}_j^r \in \mathbb{R}^{d_i}$ denotes the interaction vector. $\boldsymbol{W}_r \in \mathbb{R}^{d_x\times d_i}$ and $\boldsymbol{b}_r \in \mathbb{R}^{d_i}$ are trainable parameters for the right layer, $\boldsymbol{W}_w \in \mathbb{R}^{d_x\times d_i}$ and $\boldsymbol{b}_w \in \mathbb{R}^{d_i}$ are trainable parameters for the wrong layer.

According to the student's global cognitive vector $\boldsymbol{s}_g$ and the item interaction $\boldsymbol{x}_j^r$, the next step is to model their interactions, i.e., measuring if the student's cognitive state can meet the requirement of the item, which is just as existing cognitive diagnosis methods have done. Through the interaction modeling, we can gradually optimize the global cognitive vector and the enhanced item vector $\boldsymbol{x}_j$. Once after training on all student-item interactions, we can get the static global cognitive vector $\boldsymbol{s}_g$ and utilize it to predict the student's performance on future items. However, as we have claimed, there will be unenviable performance loss if equally predicting the student's performance on all future items according to the same global cognitive vector $\boldsymbol{s}_g$. The local dynamics of the learning process are also of great significance for student performance prediction. Therefore, before conducting the interaction modeling, we measure the local dynamic feature in our proposed GLNC methods, which will be presented in the following.

4.2. Local dynamics modeling

Intuitively, learning is a dynamic process that is influenced by complex factors. For example, students' cognitive states may decline due to forgetting (Averell & Heathcote, 2011). Previous interacted items will impact students' performance on the following items (Pettijohn II & Sacco, 2007). In general, students know little things at the beginning, and their knowledge continues to grow as they learn (Ayre & Nettle, 2015). Therefore, the measurement of students' cognitive states should be dynamic as well. We summarize the local dynamics of learning are reflected by two aspects: (1) students' cognitive states are dynamic (e.g., increasing by practicing and decreasing due to forgetting), and (2) the relation between different interactions (e.g., students tend to have similar performance on similar items). Therefore, we design the self-attentive encoder to capture students' local dynamic cognitive states and the dependencies between different student-item interactions.

Specifically, among the student's whole interaction sequence $\mathbb{X} = \{(s, i_1, r_1), (s, i_2, r_2), \ldots, (s, i_t, r_t), \ldots, (s, i_T, r_T)\}$, we choose the most recent L interactions $\mathbb{X}_L = \{(s, i_{T-L}, r_{T-L}), (s, i_{T-L+1}, r_{T-L+1}), \ldots, (s, i_T, r_T)\}$ to extract the local feature in the student's cognitive states for the next item i_{T+1}. Here $\mathbb{X}_L$ was chosen by the timestamp information recorded by ITS. Similar to the global cognitive modeling, students' item-response pairs in the interaction sequence are transformed to the interaction vector $\boldsymbol{x}_j^r$ based on Eqs. (4) and (5). $\mathbb{X}_L$ is the input of the self-attentive encoder, and the corresponding output is the local cognitive vector s_l. The structure of the proposed self-attentive encoder contains six identical basic modules, which is shown in Fig. 4. In the basic module, there are three layers: the self-attentive attention layer, the normalization layer, and the feed-forward layer. Noting that we

---

Algorithm 1 The GLNC Model.

Require: The student's historical interactions $\mathbb{X} = \{(s, i_1, r_1), (s, i_2, r_2), ..., (s, i_t, r_t), ..., (s, i_T, r_T)\}$ The embedding matrix of items, $\boldsymbol{I} \in \mathbb{R}^{J\times d_i}$; The embedding matrix of knowledge concept, $\boldsymbol{K} \in \mathbb{R}^{M\times d_k}$; The Q-matrix $\boldsymbol{Q} \in \mathbf{R}^{J\times M}$; The new item to be answered, $\boldsymbol{i}_{new} \in \mathbb{R}^{d_i}$.

Ensure: The probability that the student can get a correct response on the new item, i.e., $P(r_{new} = 1|i_{new}, \mathbb{X}, \boldsymbol{Q})$.

compute the enhanced item vector by Eq. (4);
compute the item-response vector from Eq. (5);
realize the basic module of the self-attentive encoder by Eq. (6), Eq. (7), Eq. (8), and Eq. (9);
repeat the basic module of the self-attentive encoder six times;
obtain the local cognitive vector from Eq. (10);
obtain the representation of the item to be predicted from Eq. (11);
compute the fused gate by Eq. (12) and Eq. (13);
combine the global cognitive vector and the local cognitive vector by Eq. (14);
predict the student's performance by Eq. (15);
return $P(r_{new} = 1)$;

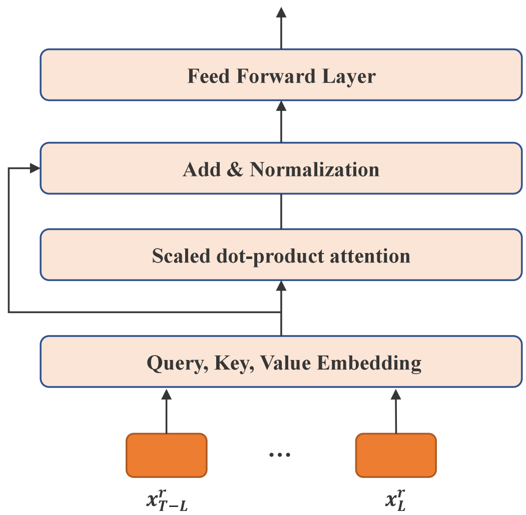


Fig. 4. The basic module of the self-attentive encoder.

stack the basic modules six times to make the modeling of students' local dynamic cognitive states more adequate.

The attention mechanism has been widely applied in existing KTMs (Ghosh et al., 2020; Vaswani et al., 2017) and broader user profiling (Liu et al., 2023) to capture the similarity between different items/interactions. The scaled dot-product attention is a specific implementation of the attention mechanism, which has wide applications in computer vision and natural language processing (Shen, Zhang et al., 2021), and its effectiveness in capturing long-term dependencies of various elements has been widely demonstrated. In the self-attentive layer, we also apply the scaled dot-product attention mechanism to measure the dependencies between different student-item interactions. Specifically, for every interaction vector in $\mathbb{X}_L$, we first utilize three embedding layers to project $\boldsymbol{x}^r_j$ into the query vector $\boldsymbol{q}^r_j \in \mathbb{R}^{d_x \times 1}$, the key vector $\boldsymbol{k}^r_j \in \mathbb{R}^{d_x \times 1}$ and the value vector $\boldsymbol{v}^r_j \in \mathbb{R}^{d_x \times 1}$. Then, the scaled dot-product attention values $\boldsymbol{w}_{ll'}$ between $\boldsymbol{x}^r_{l,j}$ and $\boldsymbol{x}^r_{l',j}$ are calculated as:

$$\boldsymbol{w}_{ll'} = \mathrm{softmax}(\frac{\boldsymbol{q}^{rT}_{l,j}\boldsymbol{k}^r_{l',j}}{\sqrt{d_x}}) = \frac{\exp(\frac{q^{rT}_{l,j}\boldsymbol{k}^r_{l',j}}{\sqrt{d_x}})}{\sum_{l'=T-L}^{T}\frac{q^{rT}_{l,j}\boldsymbol{k}^r_{l',j}}{\sqrt{d_x}}}, \tag{6}$$

where $\boldsymbol{q}^r_{l,j} \in \mathbb{R}^{d_x \times 1}$ and $\boldsymbol{k}^r_{l,j} \in \mathbb{R}^{d_x \times 1}$ respectively denote the query and the key for the lth interaction. The scaling factor is $\frac{1}{\sqrt{d_x}}$. After calculating, $\boldsymbol{w}_{ll'}$ is now a weight vector with values in ranges of 0 to 1, and its sum is 1. The value of $\boldsymbol{w}_{ll'}$ directly represents the similarity between different interaction vectors, therefore we multiply it to $\boldsymbol{v}_{l',j}$ to get the output of the scaled dot-product attention mechanism as a weighted sum of the values:

$$\boldsymbol{a}^r_j = \sum_{l'=T-L}^{T} \boldsymbol{w}_{ll'}\boldsymbol{v}^r_{l',j}, \tag{7}$$

where we have $\boldsymbol{v}^r_{l,j} \in \mathbb{R}^{d_x \times 1}$ denotes the value for the lth interaction.

In the normalization layer, we aim to enhance the generalization ability of GLNC by applying layer normalization (Xu et al., 2019), which has also been verified to enable smoother gradients and faster training. Concretely, we first add a short-cut connection (He et al., 2016) between the input interaction vector $\boldsymbol{x}^r_j$ and the output attentive vector $\boldsymbol{a}^r_j$, which directly propagates the original feature in the bottom item interaction to higher layers. Then, we leverage the operation of layer normalization to normalize across the features and accelerate the training process, as follows:

$$\hat{\boldsymbol{a}}^r_j = \mathrm{LayerNormalization}(\boldsymbol{x}^r_j + \boldsymbol{a}^r_j). \tag{8}$$

In the feed-forward layer, we conduct a multi-layer perceptron with the Relu activation on $\hat{\boldsymbol{a}}^r_j$ to increase the power of non-linearity in the self-attentive encoder:

$$\tilde{\boldsymbol{a}}^r_j = \mathrm{Relu}(\boldsymbol{W}^T_2\hat{\boldsymbol{a}}^r_j + \boldsymbol{b}_2) + \hat{\boldsymbol{a}}^r_j, \tag{9}$$

where $\boldsymbol{W}_2 \in \mathbb{R}^{d_x \times d_x}$ and $\boldsymbol{b}_2 \in \mathbb{R}^{d_x}$ are trainable parameters.

Finally, we add the operation of global average pooling followed by a multi-layer perceptron to the output of the self-attentive encoder and obtain the local cognitive vector $\boldsymbol{s}_l$ as follows:

$$\boldsymbol{s}_l = \boldsymbol{W}^T_3\frac{\sum\tilde{\boldsymbol{a}}^r_j}{L} + \boldsymbol{b}_3, \tag{10}$$

where $\boldsymbol{W}_3 \in \mathbb{R}^{d_x \times d_x}$ and $\boldsymbol{b}_3 \in \mathbb{R}^{d_x}$ are trainable parameters. After the above self-attentive encoder, we extract the local dynamics of learning from students' most recent L student-item interactions. The output local cognitive vector $\boldsymbol{s}_l$ contains complex dependencies in interactions, which can reflect students' dynamic cognitive states in the evolving learning process. In Section 5.6, the experimental results have demonstrated the effectiveness of this local dynamics modeling module.

4.3. Combined cognitive modeling and performance predicting

So far, we have obtained both the global cognitive vector $\boldsymbol{s}_g$ and the local cognitive vector $\boldsymbol{s}_l$, which respectively include the static and dynamic features in students' cognitive states. As we have mentioned in Section 4.1, to predict student performance on a specific item, we need to measure if the student's cognitive state can meet the item's requirement. Therefore, we should first integrate $\boldsymbol{s}_g$ and $\boldsymbol{s}_l$ together to represent students' cognitive states.

One natural idea of combining $\boldsymbol{s}_g$ and $\boldsymbol{s}_l$ is directly adding them together. However, a simple addition gives the same weight to the global and local cognitive features, which differs from the fact that the significance of global and local features varies depending on the item to be predicted. Considering such differences, we present the fused gate based on the similarity between students' recently responded items and the item to be predicted. Then, $\boldsymbol{s}_g$ and $\boldsymbol{s}_l$ will be adaptively combined by the fused gate and return students' cognitive states for specific items.

Specifically, we first combine the item embedding $\boldsymbol{i}_{new}$ and the knowledge concept embedding $\boldsymbol{k}_{i_{new}}$ to represent the item i_{new} to be predicted as Eq. (4):

$$\boldsymbol{x}_{new} = \boldsymbol{W}^T_1(\boldsymbol{i}_{new} \oplus \boldsymbol{k}_{i_{new}}) + \boldsymbol{b}_1, \tag{11}$$

where we use the same parameters in Eq. (4). Subsequently, we calculate the overall similarity $\boldsymbol{att}$ between students' recently responded items and $\boldsymbol{x}_{new} \in \mathbb{R}^{d_x \times 1}$, as follows:

$$\boldsymbol{att} = \sigma(\frac{\sum_{l=T-L}^{T}\boldsymbol{x}^T_l\boldsymbol{x}_{new}}{L}), \tag{12}$$

where σ is the sigmoid activation function. Then, we can design the fused gate based on the overall similarity:

$$\boldsymbol{\Gamma} = \sigma(\boldsymbol{W}^T_4(\boldsymbol{att}) + \boldsymbol{b}_4), \tag{13}$$

where $\boldsymbol{W}_4 \in \mathbb{R}^{1 \times d_s}$ and $\boldsymbol{b}_4 \in \mathbb{R}^{d_s}$ are trainable parameters.

Finally, students' cognitive states can be combined by the fused gate $\boldsymbol{\Gamma}$ as:

$$\boldsymbol{s}_c = \boldsymbol{\Gamma} \cdot \boldsymbol{s}_l + (1 - \boldsymbol{\Gamma}) \cdot \boldsymbol{s}_g. \tag{14}$$

In this way, we realize the control of the importance of $\boldsymbol{s}_g$ and $\boldsymbol{s}_l$ for different future items based on their similarity to students' recently responded items.

After getting $\boldsymbol{s}_c$, To make predictions, we utilize the inner product of the cognitive vector $\boldsymbol{s}_c$ and the item vector $\boldsymbol{x}_{new}$ to simulate the student's decision-making process while answering the item:

$$P(r_{new} = 1) = \sigma(\sum(\boldsymbol{s}_c \cdot \boldsymbol{x}_{new})), \tag{15}$$

Table 2
Statistics of all datasets.

Statistics	Datasets		
	TAL2023	EdNet	Eedi2020
# of Students	18,066	78,430	118,971
# of Items	7,652	12,372	27,613
# of KCs	1,175	141	282
# of Interactions	5,549,635	9,058,114	11,001,297
Avg.interactions per student	307	115	92

In order to make the above formula computable, we set the dimension d_i equal to d_s in our implementation.

Noting that the above method for combining global and local features does not consider the knowledge correlations of items and the individual characteristics of students. Actually, items have correlations of varying strengths, therefore it is necessary to explore the fine-graded knowledge correlations between various items and their influence on students' performance. Besides, students' individual characteristics impact their distinct dynamic learning processes, while our method cannot capture these differences. However, as the focus of this paper is on validating the effectiveness of combining global and local features, we leave the exploration of more sufficient combination ways as future works.

4.4. Model learning

To learn all randomly initialized embeddings (e.g., the embedding matrix $\boldsymbol{I}$ and $\boldsymbol{K}$) and learnable parameters (e.g., $\boldsymbol{W}_*$, $\boldsymbol{b}_*$) in GLNC. We utilize the cross entropy between the predicted probability of correct response p and actual binary response r as the objective function:

$$\mathbb{L} = -\sum(p\log r + (1-p)\log(1-r)) + \lambda_{\Theta}\|\Theta\|^2, \tag{16}$$

where Θ denotes all parameters of GLNC and λ_{Θ} is the regularization hyperparameter. The objective function will be minimized using Adam optimizer (Kingma & Ba, 2014) on mini-batches. More details of settings will be specified in the experiments.

5. Experiments

In this section, we conduct experiments on three real-world datasets to evaluate our proposed GLNC model. We aim to answer the following research questions through the experiments:

- **RQ1**: How does GLNC perform on the student performance task as compared with state-of-the-art cognitive diagnosis and knowledge tracing methods?
- **RQ2**: How do different elements (e.g. the global cognitive modeling, the local dynamics modeling, the adaptive combination, and the knowledge concept) in GLNC affect its performance?
- **RQ3**: How is the robustness of GLNC in case of sparse and noisy data?

5.1. Experimental dataset

We evaluate our method and all comparison methods on three public real-world datasets: (1) TAL2023,[5] (2) EdNet,[6] and (3) Eedi2020.[7] Table 2 shows their statistics.

- **TAL2023** is published in the AAAI 2023 Global Knowledge Tracing Challenge. The data is collected from students' item interactions in grade 3 math classes at Xueersi from TAL Education Group. Specifically, there are 18,066 students, 7652 questions, 1175 KCs, and 5,549,635 interactions in this dataset. The average number of interactions for each student is 307. 79.47% of responses in this dataset have positive labels, i.e., students correctly answer the questions.
- **EdNet** is collected from Santa, an intelligent tutoring platform in Korea (Choi et al., 2020). The full dataset contains all student-system interactions collected over 2 years with more than 780 thousand students and 130 million interactions. Considering that the full dataset requires many computation resources, according to the suggestion of Long et al. (2021), we randomly sample 78,430 students (one-tenth of the full dataset) and their 9,058,114 interactions on 12,372 items in our experiments.
- **Eedi2020** is collected from about two school years (September 2018 to May 2020) of students who interact daily around the globe on the learning platform Eedi. The data was published in the NeurIPS 2020 Education Challenge. There were 118,971 students, 27,613 items, and 11,001,297 interactions in this dataset. This dataset has hierarchical KCs, where the KC in the lower level is contained in the upper level. For convenience sake, we utilize only the most fine-grained KC for each question in our experiments.

5.2. Baseline methods

To evaluate the effectiveness of our GLNC, we consider both cognitive diagnosis models and knowledge tracing models as baselines. All models are implemented by open-source code, and all experiments were run on a Linux server with the NVIDIA RTX 3090 GPU.

- **IRT** is one of the most classical CDMs, which models students' cognitive ability as a continuous variable (Hambleton et al., 1991). IRT uses a logistic regression model by estimating student ability and item difficulty. IRT has been widely applied in ITS due to its superior interpretability. However, as a classical method, it is less accurate than the latest models. In our experiments, we used the popular 3-parameter IRT model for comparison:

$$p_i(r=1|\theta_s) = c + \frac{1-c}{1+e^{-(\theta_s-\beta_i)}}, \tag{17}$$

  where the 3-parameter respectively stands for the random guessing probability c, the student's cognitive state θ_s, and the item difficulty β_i.
- **PMF** is the probabilistic matrix factorization model, which projects students and items into latent factors for performance predicting (Thai-Nghe et al., 2012). PMF is simple and has comparable performance with IRT. In our experiments, we used the public complementation of PMF for cognitive diagnosis according to Wang et al. (2020)
- **NCD** is the first deep learning-based CDM (Wang et al., 2020), which captures the complex student-item interaction by neural networks. NCD is the present state-of-the-art cognitive diagnosis method. However, its interpretability and robustness still to be improved. In our experiments, we reproduced NCD based on its open-sourced code.[8]
- **EKT** takes students' interaction sequence as the input of the RNN layer, then applies a linear mapping and the sigmoid activation function to the output hidden states to get students' cognitive states (Liu, Huang et al., 2021). It is worth noting that EKT can be seen as an extension of DKT, which introduces the rich information in the text content of the item. Therefore, EKT should have at least the same performance as DKT, so we do not compare our model with DKT anymore. In our experiments, we used

[5] http://ai4ed.cc/competitions/aaai2023competition

[6] http://ednet-leaderboard.s3-website-ap-northeast-1.amazonaws.com/

[7] https://eedi.com/projects/neurips-education-challenge

[8] https://github.com/bigdata-ustc/Neural_Cognitive_Diagnosis-NeuralCD

Table 3
Results of student performance prediction on two datasets with all metrics. We use underline to highlight the best result among all baselines in each column, while bold number shows the best result of the whole column. The error bars after ± are the standard deviations of 5 evaluation runs for each method. The gain shows our model improvement over the best performance of all baselines.

(a) Results of student performance prediction on TAL2023.

Methods	TAL2023			
	AUC	ACC	RMSE	r^2
IRT (Hambleton et al., 1991)	0.796±.002	0.818±.002	0.359±.002	0.188±.002
PMF (Thai-Nghe et al., 2012)	0.790±.002	0.821±.002	0.360±.002	0.181±.003
NCD (Wang et al., 2020)	0.793±.002	0.823±.002	0.358±.001	0.189±.001
EKT (Liu, Huang et al., 2021)	0.791±.002	0.816±.002	0.363±.001	0.196±.004
SAKT (Pandey & Karypis, 2019)	0.787±.002	0.814±.001	0.364±.001	0.191±.004
LPKT (Shen et al., 2021a)	0.792±.002	0.816±.001	0.363±.001	0.196±.004
GLNC	**0.808±.002**	**0.827±.001**	**0.352±.003**	**0.218±.001**
Gain	1.51%	0.49%	1.68%	11.22%

(b) Results of student performance prediction on EdNet.

Methods	EdNet			
	AUC	ACC	RMSE	r^2
IRT (Hambleton et al., 1991)	0.736±.002	0.711±.003	0.433±.002	0.143±.003
PMF (Thai-Nghe et al., 2012)	0.740±.002	0.717±.003	0.434±.002	0.139±.003
NCD (Wang et al., 2020)	0.743±.002	0.723±.003	0.430±.002	0.156±.003
EKT (Liu, Huang et al., 2021)	0.726±.002	0.686±.001	0.451±.001	0.149±.003
SAKT (Pandey & Karypis, 2019)	0.726±.002	0.685±.002	0.452±.001	0.146±.002
LPKT (Shen et al., 2021a)	0.731±.002	0.690±.002	0.449±.001	0.155±.003
GLNC	**0.762±.002**	**0.731±.002**	**0.422±.001**	**0.187±.003**
Gain	2.56%	1.11%	1.86%	19.87%

(c) Results of student performance prediction on Eedi2020.

Methods	Eedi2020			
	AUC	ACC	RMSE	r^2
IRT (Hambleton et al., 1991)	0.745±.001	0.703±.002	0.435±.001	0.151±.001
PMF (Thai-Nghe et al., 2012)	0.751±.002	0.714±.001	0.435±.002	0.148±.003
NCD (Wang et al., 2020)	0.749±.001	0.718±.002	0.434±.001	0.158±.005
EKT (Liu, Huang et al., 2021)	0.760±.001	0.715±.001	0.432±.001	0.186±.002
SAKT (Pandey & Karypis, 2019)	0.758±.001	0.714±.001	0.433±.001	0.182±.002
LPKT (Shen et al., 2021a)	0.762±.001	0.716±.001	0.431±.001	0.196±.004
GLNC	**0.773±.001**	**0.730±.001**	**0.422±.001**	**0.202±.002**
Gain	1.44%	1.63%	2.07%	3.06%

a more efficient variant of RNN to implement EKT, i.e., GRU. Besides, as most datasets (including the three dataset we used in our experiments) do not supply the text content of the item, we rewrote the original EKT code and randomly initialized the item representation for EKT as the same as GLNC.

- **SAKT** applies the Transformer framework (Vaswani et al., 2017) to the knowledge tracing task (Pandey & Karypis, 2019). It proposes a self-attentive model to capture long-term dependencies between student-item interactions. As the Transformer framework has a powerful ability of extraction and representation, the most improvement of SAKT is from the Transformer framework. In our experiments, we reproduced SAKT based on its open-sourced code.[9]
- **LPKT** is the learning process-consistent knowledge tracing model (Shen et al., 2021a). It designs a specific architecture to model students' learning process and calculates students' learning gains and forgetting to assess their cognitive states. LPKT is the present state-of-the-art knowledge tracing method. In our experiments, we reproduced LPKT based on its open-sourced code.[10]

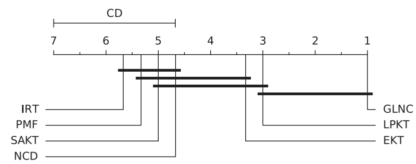
5.3. Evaluation metrics

We use multiple metrics to evaluate the performance of GLNC and all comparison methods. we use multiple metrics from both regression and classification perspectives. Specifically, from the perspective of classification, a record with a score of 1(0) indicates a positive (negative) instance, we adopt Area Under ROC Curve (AUC) and Prediction Accuracy (ACC) to measure the effectiveness, and larger values equal to better results. Here we set a threshold of 0.5 when calculating the result of accuracy. Then, from the perspective of regression, we quantify the distance between the predicted and actual response with Root Mean Square Error (RMSE) and the square of Pearson correlation (r^2). For the RMSE, smaller values mean better results. The r^2 is the opposite, where larger values are better results.

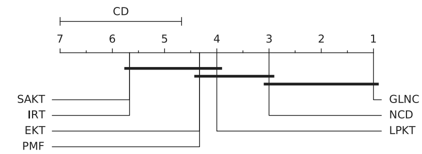
5.4. Experimental settings

In our experiments, we filter out students with less than 30 interactions in both datasets to guarantee that each student has enough interactions for conducting precise cognitive modeling. All students' interaction logs were first sorted by the timestamp (i.e., the order of item interactions). For all students in the dataset, we randomly selected 80% of them as the training set and validation set (the ratio of training and validation is 8:2), the rest 20% were used as the testing set. Experiments on all datasets have been 5-fold cross-validated

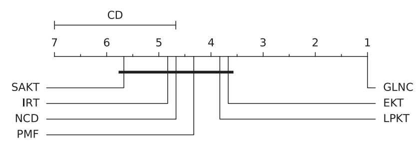
[9] https://github.com/arshadshk/SAKT-pytorch
[10] https://github.com/shshen-closer/LPKT_tensorflow_version



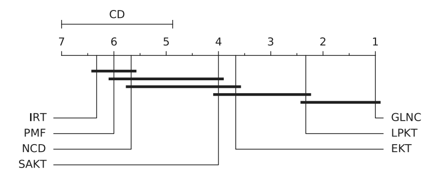
(a) Nemenyi's critical difference diagram on the AUC metric.



(b) Nemenyi's critical difference diagram on the ACC metric.



(c) Nemenyi's critical difference diagram on the RMSE metric.



(d) Nemenyi's critical difference diagram on the r^2 metric.

Fig. 5. Comparison of all baselines and GLNC on three public datasets with the Nemenyi test.

Table 4
Results of ablation experiments on TAL2023.

Methods	Global feature	Local feature	Adaptive combination	Knowledge concept	AUC	ACC	RMSE	r^2
GLNC w/o g	✗	✓	✓	✓	0.793	0.823	0.358	0.195
GLNC w/o l	✓	✗	✓	✓	0.800	0.824	0.355	0.207
GLNC w/o c	✓	✓	✗	✓	0.806	0.826	0.353	0.215
GLNC w/o KC	✓	✓	✓	✗	0.802	0.825	0.354	0.209
GLNC	✓	✓	✓	✓	**0.808**	**0.827**	**0.352**	**0.218**

on students. All the hyperparameters are learned on the training set, and the model that performed best on the validation set was used to evaluate the testing set. We initialize all parameters and embeddings with Xavier initialization (Glorot & Bengio, 2010). All the learnable parameters and embedding are learned on the training set, and the model that performed best on the validation set was used to evaluate the testing set. Empirically, we set all dimensions with a uniform value of 128, including the dimension d_k of the knowledge concept matrix, the dimension d_s of the cognitive vector, the dimension d_i of the interaction vector, and the dimension d_x of the enhanced item vector. The learning rate was 2e-3 during the training process, and the mini-batch size was 512. We set a dropout rate of 0.5 for mitigating overfitting. It is worth noting that we empirically chose the above values, without carefully choosing specific values of the dimension, the learning rate, the mini-batch size, and the dropout rate for better experimental results. Utilizing other values may bring better results, We leave the exploration of optimal settings of hyperparameters as future work.

5.5. Student performance prediction (RQ1)

To evaluate the performance of the GLNC model compared with baselines, we report the results of all metrics on all datasets in Table 3. In this implementation of GLNC, the parameter L in local dynamic modeling was 30, i.e., we extracted the local dynamics of learning from students' 30 most recent interactions for each prediction (see Table 4).

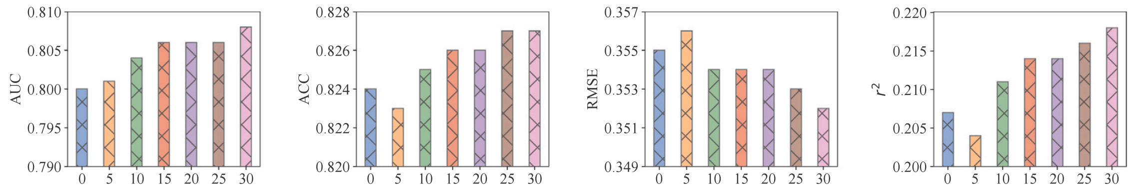
According to Table 3, we can see that our GLNC model outperforms all baselines and achieves the best performance. For the dataset TAL2023, our model outperforms the existing state-of-the-art method by 1.51% in AUC, 0.49% in ACC, 1.68% in RMSE, and 11.22% in r^2. For the dataset EdNet, our model outperforms the existing state-of-the-art method by 2.56% in AUC, 1.11% in ACC, 1.86% in RMSE, and 19.87% in r^2. For the dataset Eedi2020, the corresponding gains compared with the existing state-of-the-art method are 1.44% in AUC, 1.63% in ACC, 2.07% in RMSE, and 3.06% in r^2. We have noticed that knowledge tracing methods and cognitive diagnosis methods have different performances on different datasets, i.e., knowledge tracing methods are not superior to the cognitive diagnosis methods on EdNet while EKT has the best result among all baselines on TAL2023 and LPKT achieves the best result among all baselines on Eedi2020. We guess the reason is that the global features and the local features have different importance in different datasets. Our proposed GLNC can get the best results on all datasets because it captures both the global cognitive states of students and the local dynamics in learning.

To evaluate if GLNC significantly outperforms existing baselines, we have applied Friedman and post-hoc Nemenyi tests (Pereira et al., 2015). Specifically, we first conducted the Friedman hypothesis test with $p = 0.05$ considering the 5-fold cross-validated results of 6 baselines and GLNC over the three datasets we used. The Friedman test rejected the null hypothesis that all comparison methods were equally comparable. We further executed a post-hoc Nemenyi test. The results of the Nemenyi test on all metrics are presented as a critical distance graphic in Fig. 5, where the value indicates the average rank of each method. Lower ranks stand for better performance, the connected crossline means that there is no significant difference among the models in the line. We can see that GLNC achieves the first ranking for all metrics from Fig. 5. For the AUC and r^2 metrics, the Nemenyi test shows significant differences between GLNC and all baselines except LPKT. For the RMSE metric, GLNC significantly outperforms all baselines. For the ACC metric, GLNC significantly outperforms all baselines except NCD, which requires more experiments to indicate the significant difference as the Nemenyi performs multiple comparisons.

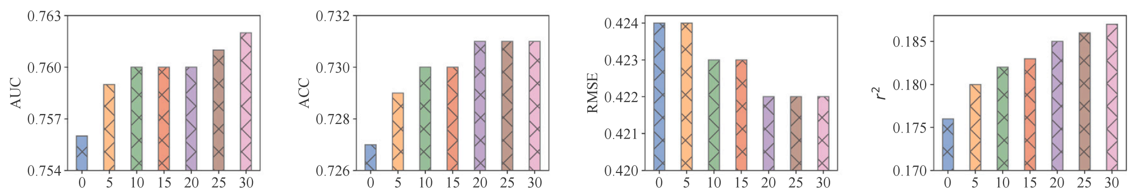
5.6. Ablation study (RQ2)

In this section, to highlight the effectiveness of each part in our GLNC model including the global cognitive modeling, the local dynamics modeling, the adaptive combination, and the knowledge concept, we introduce and compare the performance of the following variants of GLNC on the dataset TAL2023:

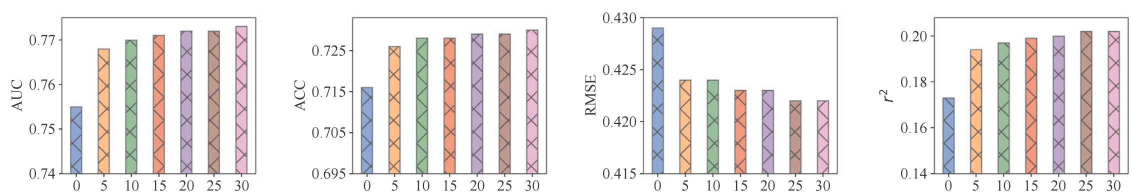
- **GLNC w/o g** is the variant of GLNC without measuring students' global cognitive features.



(a) The effect of the length of students' historical interactions on GLNC's performance for the dataset TAL2023.



(b) The effect of the length of students' historical interactions on GLNC's performance for the dataset EdNet.



(c) The effect of the length of students' historical interactions on GLNC's performance for the dataset Eedi2020.

Fig. 6. The effect of the length of students' historical interactions on GLNC's performance. We have compared 7 different lengths: 0, 5, 10, 15, 20, 25, 30. The results indicate a positive relationship between the performance of GLNC and the length of the utilized interactions.

- **GLNC w/o l** is the variant of GLNC without measuring students' local dynamic cognitive features.
- **GLNC w/o c** is the variant of GLNC that equally combines global and local cognitive features.
- **GLNC w/o kc** is the variant of GLNC without considering items' knowledge concepts.

The corresponding results are reported in Section 5.6, where we can find some interesting conclusions. First, both the global cognitive modeling and the local dynamics modeling are critical for more accurately predicting students' responses, which damages GLNC's performance if we drop one of them. Second, it is necessary to combine global and local cognitive features with various weights according to the similarity between students' recently responded items and the item to be predicted. Third, the information of items' knowledge concepts is beneficial as expected, which brings more complete characteristics of items.

5.7. Robustness analysis (RQ3)

In this part, we will evaluate the robustness of GLNC from two aspects: (1) the robustness under sparse interactions, and (2) the robustness under the impact of noisy student-item interaction data.

- **The effect of sparse data.** In Table 3, students' local dynamic modeling was completed based on their most recent 30 interactions. However, in real ITS, it takes large resources to store more than 30 real-time interactions for each student. To evaluate the robustness of GLNC under sparse recent interaction, we compared the performance of GLNC under 7 different lengths of students' historical interactions: 0, 5, 10, 15, 20, 25, and 30. Intuitively, the fewer the students' historical interactions we use, the worse the performance of GLNC should be. The results are reported in Fig. 6, where we can observe that the performance of GLNC and the length of the utilized interactions have a positive relation as expected. It is natural as more historical interactions help GLNC to conduct more complete summarizations. What is more, it is worth noting that GLNC has good stability and robustness, which outperforms the best baseline even when considering only the most recent 5 interactions. Therefore, GLNC can freely adapt to different ITS with consistent competitive performance.
- **The effect of noisy data.** In this part, we will further evaluate if GLNC has robustness under the impact of noisy student-item interaction data. Specifically, the noise in student-item interaction mainly comes from students' guessing and slipping (Chen et al., 2022). To evaluate the robustness of GLNC with noisy data, we randomly add the guessing and slipping noise with a probability of 0.2, 0.1, 0.05, 0.02, and 0.01 respectively in the training set of TAL2023. The performance variations of GLNC, GLNC w/o g, and GLNC w/o l on the testing set are shown in Fig. 7. We can observe that adding guessing and slipping causes uniform performance degradation for both GLNC and its two variants. However, although GLNC has considered both students' global cognitive ability and local dynamics (i.e., measuring the noisy data twice), its performance suffered no more declines in contrast to its two variants that measure the noisy data only once. On the contrary, GLNC is more robust against the influence of noisy data with considering both global and local features, especially when the probability of guessing and slipping is high.

6. Conclusions

In this paper, we proposed a novel Global and Local Neural Cognitive (GLNC) model, which is more precise and robust than existing methods. By combining both global and local features in student-item interactions, GLNC realized better cognitive modeling for students. When making predictions, GLNC presented the fused gate to adaptively integrate global and local features, based on the similarity between students' recently responded items and the item to be predicted. We utilized three real-world public datasets to validate the performance of GLNC on the student performance prediction task, which indicated more than 1.5% average improvement on the AUC metric and more than 10% average improvement on the r^2 metric. We also conducted

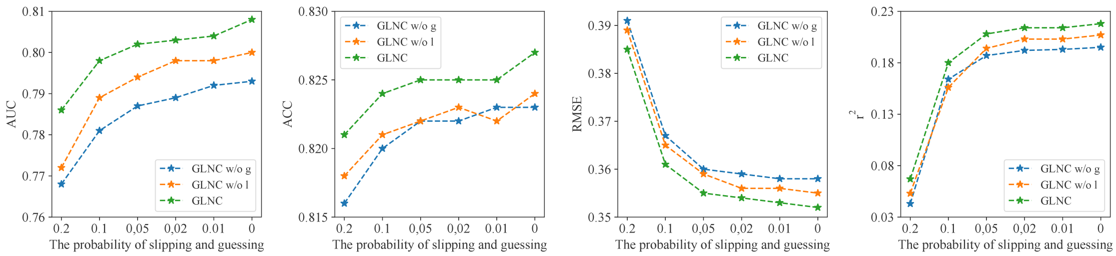


Fig. 7. The performance of GLNC and its two variants under different probabilities of guessing and slipping.

further experiments to evaluate the behavior of GLNC under the influence of sparse and noisy data, and the results indicated GLNC had good robustness. Due to the superior accuracy and robustness of GLNC, it can be freely applied in existing ITS to improve students' learning efficiency and experience.

The value of combining both global and local features in student-item interactions has been evaluated in this paper. It is promising to enhance student performance prediction and cognitive modeling by combining the complementary advantage of cognitive diagnosis and knowledge tracing. However, our proposed method to combine global and local features is insufficient, as we omitted the knowledge correlations of items and the individual characteristics of students. In the future, we will explore the fine-graded knowledge correlations between various items and their influence on students' performance. Besides, although students' cognitive states are dynamic, the distinct dynamic learning process of each student could be quite different, which is also worth studying.

CRediT authorship contribution statement

Yu Su: Conceptualization, Methodology, Project administration, Validation, Funding acquisition. **Shuanghong Shen:** Methodology, Investigation, Writing – original draft , Writing – review & editing. **Linbo Zhu:** Data curation, Software, Formal analysis. **Le Wu:** Supervision, Writing – review & editing. **Zhenya Huang:** Writing – review & editing, Resources. **Zeyu Cheng:** Investigation, Resources. **Qi Liu:** Project administration, Investigation. **Shijin Wang:** Funding acquisition, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Shuanghong Shen reports financial support was provided by University of Science and Technology of China. Yu Su reports financial support was provided by Hefei Normal University.

Data availability

I have shared the data address in the manuscript.

Acknowledgments

This research was partially supported by grants from the University Synergy Innovation Program of Anhui Province (No. GXXT-2022-042, No. GXXT-2022-101), Open Research Fund of the State Key Laboratory of Cognitive Intelligence (iED2022-002), Open Research Fund of Key Laboratory of Philosophy and Social Science of Anhui Province (SYS2023A06), the National Natural Science Foundation of China (No. sU20A20229).

References

Averell, L., & Heathcote, A. (2011). The form of the forgetting curve and the fate of memories. *Journal of Mathematical Psychology*, *55*(1), 25–35.

Ayre, M., & Nettle, R. (2015). Doing integration in catchment management research: Insights into a dynamic learning process. *Environmental Science & Policy*, *47*, 18–31.

Chen, P., Lu, Y., Pian, Y., Li, Y., & Cao, Y. (2022). Introducing response time into guessing and slipping for cognitive diagnosis. In *AIED* (pp. 320–324). Springer.

Choi, Y., Lee, Y., Shin, D., Cho, J., Park, S., Lee, S., Baek, J., Bae, C., Kim, B., & Heo, J. (2020). Ednet: A large-scale hierarchical dataset in education. In *AIED* (pp. 69–73). Springer.

Chrysafiadi, K., & Virvou, M. (2013). Student modeling approaches: A literature review for the last decade. *Expert Systems with Applications*, *40*(11), 4715–4729.

Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, *4*(4), 253–278.

De La Torre, J. (2011). The generalized DINA model framework. *Psychometrika*, *76*(2), 179–199.

de la Torre, J., & Chiu, C.-Y. (2016). A general method of empirical Q-matrix validation. *Psychometrika*, *81*, 253–273.

Deeva, G., De Smedt, J., Saint-Pierre, C., Weber, R., & De Weerdt, J. (2022). Predicting student performance using sequence classification with time-based windows. *Expert Systems with Applications*, *209*, Article 118182.

Ding, X., & Larson, E. C. (2020). Incorporating uncertainties in student response modeling by loss function regularization. *Neurocomputing*, *409*, 74–82.

Gan, W., Sun, Y., & Sun, Y. (2022). Knowledge interaction enhanced sequential modeling for interpretable learner knowledge diagnosis in intelligent tutoring systems. *Neurocomputing*, *488*, 36–53.

Ghosh, A., Heffernan, N., & Lan, A. S. (2020). Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2330–2339).

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *AISTATS* (pp. 249–256).

Hambleton, R. K., Swaminathan, H., & Rogers, H. J. (1991). *Fundamentals of item response theory*. Sage.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *CVPR* (pp. 770–778).

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kotsiantis, S., Patriarcheas, K., & Xenos, M. (2010). A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education. *Knowledge-Based Systems*, *23*(6), 529–535.

Liang, Y., Song, Q., Zhao, Z., Zhou, H., & Gong, M. (2023). BA-GNN: Behavior-aware graph neural network for session-based recommendation. *Frontiers of Computer Science*, *17*(6), Article 176613.

Liu, Q. (2021). Towards a new generation of cognitive diagnosis. In *IJCAI* (pp. 4961–4964).

Liu, Q., Huang, Z., Yin, Y., Chen, E., Xiong, H., Su, Y., & Hu, G. (2021). EKT: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering*, *33*(1), 100–115.

Liu, Q., Tong, S., Liu, C., Zhao, H., Chen, E., Ma, H., & Wang, S. (2019). Exploiting cognitive structure for adaptive learning. In *ACM SIGKDD* (pp. 627–635).

Liu, S., Yu, J., Li, Q., Liang, R., Zhang, Y., Shen, X., & Sun, J. (2022). Ability boosted knowledge tracing. *Information Sciences*, *596*, 567–587.

Liu, X., Zhu, Y., & Wu, X. (2023). Joint user profiling with hierarchical attention networks. *Frontiers of Computer Science*, *17*(3), Article 173608.

Liu, S., Zou, R., Sun, J., Zhang, K., Jiang, L., Zhou, D., & Yang, J. (2021). A hierarchical memory network for knowledge tracing. *Expert Systems with Applications*, *177*, Article 114935.

Long, T., Liu, Y., Shen, J., Zhang, W., & Yu, Y. (2021). Tracing knowledge state with individual cognition and acquisition estimation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval* (pp. 173–182).

Nakagawa, H., Iwasawa, Y., & Matsuo, Y. (2019). Graph-based knowledge tracing: Modeling student proficiency using graph neural network. In *International conference on web intelligence* (pp. 156–163). IEEE.

Pandey, S., & Karypis, G. (2019). A self-attentive model for knowledge tracing. arXiv preprint arXiv:1907.06837.

Pereira, D. G., Afonso, A., & Medeiros, F. M. (2015). Overview of Friedman's test and post-hoc analysis. *Communications in Statistics. Simulation and Computation*, *44*(10), 2636–2653.

Pettijohn II, T. F., & Sacco, M. F. (2007). Multiple-choice exam question order influences on student performance, completion time, and perceptions. *Journal of Instructional Psychology*, *34*(3).

Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., & Sohl-Dickstein, J. (2015). Deep knowledge tracing. In *Advances in neural information processing systems* (pp. 505–513).

Qi, T., Ren, M., Guo, L., Li, X., Li, J., & Zhang, L. (2023). ICD: A new interpretable cognitive diagnosis model for intelligent tutor systems. *Expert Systems with Applications*, *215*, Article 119309.

Shen, S., Huang, Z., Liu, Q., Su, Y., Wang, S., & Chen, E. (2022). Assessing student's dynamic knowledge state by exploring the question difficulty effect. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval* (pp. 427–437).

Shen, S., Liu, Q., Chen, E., Huang, Z., Huang, W., Yin, Y., Su, Y., & Wang, S. (2021). Learning process-consistent knowledge tracing. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 1452–1460).

Shen, Z., Zhang, M., Zhao, H., Yi, S., & Li, H. (2021). Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 3531–3539).

Song, L., He, M., Shang, X., Yang, C., Liu, J., Yu, M., & Lu, Y. (2023). A deep cross-modal neural cognitive diagnosis framework for modeling student performance. *Expert Systems with Applications*, Article 120675.

Song, X., Li, J., Cai, T., Yang, S., Yang, T., & Liu, C. (2022). A survey on deep learning based knowledge tracing. *Knowledge-Based Systems*, *258*, Article 110036.

Su, Y., Cheng, Z., Luo, P., Wu, J., Zhang, L., Liu, Q., & Wang, S. (2021). Time-and-concept enhanced deep multidimensional item response theory for interpretable knowledge tracing. *Knowledge-Based Systems*, *218*, Article 106819.

Su, Y., Cheng, Z., Wu, J., Dong, Y., Huang, Z., Wu, L., Chen, E., Wang, S., & Xie, F. (2022). Graph-based cognitive diagnosis for intelligent tutoring systems. *Knowledge-Based Systems*, *253*, Article 109547.

Sun, J., Zou, R., Liang, R., Gao, L., Liu, S., Li, Q., Zhang, K., & Jiang, L. (2022). Ensemble knowledge tracing: Modeling interactions in learning process. *Expert Systems with Applications*, *207*, Article 117680.

Thai-Nghe, N., Drumond, L., Horváth, T., Krohn-Grimberghe, A., Nanopoulos, A., & Schmidt-Thieme, L. (2012). Factorization techniques for predicting student performance. In *Educational recommender systems and technologies* (pp. 129–153). IGI Global.

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, *46*(4), 197–221.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Wang, F., Liu, Q., Chen, E., Huang, Z., Chen, Y., Yin, Y., Huang, Z., & Wang, S. (2020). Neural cognitive diagnosis for intelligent education systems. In *AAAI 2020*.

Wang, C., Ma, W., Zhang, M., Lv, C., Wan, F., Lin, H., Tang, T., Liu, Y., & Ma, S. (2021). Temporal cross-effects in knowledge tracing. In *The international conference on web search and data mining* (pp. 517–525). New York, NY, USA: Association for Computing Machinery.

Wang, W., Ma, H., Zhao, Y., Li, Z., & He, X. (2022). Tracking knowledge proficiency of students with calibrated Q-matrix. *Expert Systems with Applications*, *192*, Article 116454.

Wang, W., Ma, H., Zhao, Y., Yang, F., & Chang, L. (2022). SEEP: Semantic-enhanced question embeddings pre-training for improving knowledge tracing. *Information Sciences*, *614*, 153–169.

Wang, X., Mei, X., Huang, Q., Han, Z., & Huang, C. (2021). Fine-grained learning performance prediction via adaptive sparse self-attention networks. *Information Sciences*, *545*, 223–240.

Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, *1*(2), 270–280.

Wu, Z., Huang, L., Huang, Q., Huang, C., & Tang, Y. (2022). SGKT: Session graph-based knowledge tracing for student performance prediction. *Expert Systems with Applications*, *206*, Article 117681.

Xie, T., Zheng, Q., & Zhang, W. (2016). A behavioral sequence analyzing framework for grouping students in an e-learning system. *Knowledge-Based Systems*, *111*, 36–50.

Xu, J., Sun, X., Zhang, Z., Zhao, G., & Lin, J. (2019). Understanding and improving layer normalization. In *Advances in neural information processing systems, Vol. 32*. Curran Associates, Inc..

Yazdani, M. (1986). Intelligent tutoring systems: An overview. *Expert Systems*, *3*(3), 154–163.

Zhou, Y., Liu, Q., Wu, J., Wang, F., Huang, Z., Tong, W., Xiong, H., Chen, E., & Ma, J. (2021). Modeling context-aware features for cognitive diagnosis in student learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 2420–2428).