Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 1552745, 12 pages
https://doi.org/10.1155/2022/1552745

Research Article

Knowledge Tracing via Attention Enhanced Encoder-Decoder

Kai Zhang,[1] **Zhengchu Qin**,[1] **and Ying Kuang**[2]

[1] *School of Computer Science, Yangtze University, Jingzhou 434000, China*
[2] *School of Foreign Languages, Yangtze University College of Arts and Sciences, Jingzhou 434020, China*

Correspondence should be addressed to Ying Kuang; 2021710632@yangtzeu.edu.cn

Received 17 October 2022; Revised 28 November 2022; Accepted 5 December 2022; Published 17 December 2022

Academic Editor: Kuruva Lakshmanna

The knowledge tracing model takes students' learning behaviours data as input to determine their current knowledge status and predict their future answers. The learning behaviours data describes three main types of learning behaviours: learning process, learning end, and learning interval. The classical knowledge tracing models only use the data of the learning end, which contains limited information and the models cannot accurately describe constraint in the same learning behaviour in the time series. Subsequent models add other types of learning behaviours data but do not integrate different types of learning behaviours, and the models cannot accurately describe collaboration in different learning behaviours. To address these issues, knowledge tracing via attention-enhanced encoder-decoder is proposed to synthesize and analyse the three types of learning behaviours mentioned above and firstly adopts the multiheaded attention mechanism to describe constraint in the same learning behaviours; secondly adopts the channel attention mechanism modelling collaboration in the three types of learning behaviours. In the experiments, various comparisons are made with related models on several real data sets, and the results show that our model achieves certain advantages in terms of performance and knowledge state representation. In terms of practical application, an intelligent learning platform based on the model has been implemented, which predicts the future answer of students in the teaching process of two offline courses: computer and English and has achieved better performance than other knowledge tracing models.

1. Introduction

Influenced by the COVID-19 epidemic, the public gradually accepts smart education platforms such as Intelligent Tutoring System (ITS) and Massive Open Online Course (MOOC). However, the initial endowment attributes of smart education do not include functions such as determining the state of students' knowledge and predicting their future learning performance.

For these reasons, Knowledge Tracing (KT) has become an important research element in the field of smart education, which analyses the learning behaviours data collected by the platform to determine the students' knowledge status and predicts their future performance in answering exercises based on the knowledge status. Knowledge tracing is now widely used in various online education platforms, such as Academy Online, Khan Academy, edX, Coursera, and so on. The main meaning and function of current knowledge tracing is to provide fine-grained educational strategies for smart education platforms by grasping students' knowledge status and predicting future performance of answering questions and to provide personalized educational services for each student.

Learning sequences consist of students' learning records, mainly those of students' learning behaviours. Learning behaviours data can be generally divided into three categories [1], namely learning process data, learning end data, and learning interval data, which are used to describe the corresponding learning behaviours in the learning records. The learning process data describe the learning process behaviour, mainly including the number of attempts to answer and the number of requests for hints. The learning end data describe the learning end behaviour, mainly including the exercises students answer and the results of their answers. The learning interval data describe the learning interval behaviour, mainly including the time interval between two adjacent learning sessions and the number of times students learn a concept. Figure 1 shows the learning

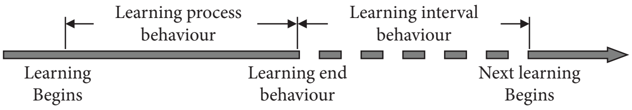

Figure 1: Learning behaviours and their sequential relationship.

process behaviour, learning end behaviour, and learning interval behaviour and their sequential relationships.

The classical knowledge tracing models [2–4] only use the learning end data. These models are generally able to determine the basic knowledge state of students by analysing their learning end behaviour, but since the learning end data only contain information about students' correct or incorrect answers to a certain exercise, they cannot trace students' knowledge state more accurately. For example, when learning the third-person singular concept in English, students A and B have the same learning end data but different learning process data, the different knowledge states of students A and B on the third-person singular concept cannot be represented in such classical knowledge tracing models.

Students' learning records also include learning process behaviour and learning interval behaviour, which also map changes in students' knowledge states. Some researchers have used learning process data and learning end data to trace students' knowledge states [5] and learning interval data to model students' forgetting behaviours [6, 7], but none of them have considered collaboration in different behaviours, i.e., the interaction of multiple types of learning behaviours in a learning sequence.

In order to more accurately trace the state of students' knowledge, the main tasks of this paper are as follows:

(1) Describing constraint in the same behaviours. First, the set of three types of learning behaviour data is selected as the input; second, the attention weights of the input data are obtained using the multiheaded attention mechanism to represent the constraint relationship of a single type of learning behaviour on the time series, which is used to describe constraint in same behaviours.

(2) Describe the collaboration in different behaviours. First, the set of three types of learning behaviours data is stitched as input; second, the global information of the three types of learning behaviours is obtained using the channel attention mechanism; finally, the global information is mapped into attention weights among learning behaviours, which represent the interaction of multiple types of learning behaviours and is used to describe collaboration in different behaviours.

(3) Knowledge tracing via attention-enhanced encoder-decoder is proposed. First, the encoder is used to fuse the constraint in the same behaviours and the collaboration in different behaviours; second, the decoder is used to obtain students' learning vectors and forgetting vectors by inputting different query vectors; finally, the purpose of tracing students' knowledge states more accurately is achieved.

2. Related Work

2.1. Knowledge Tracing

2.1.1. Knowledge Tracing Models Based on Learning End Behaviour. Bayesian Knowledge Tracing (BKT) [2] first introduced the concept of knowledge tracing and used a probabilistic calculation method to solve the task of knowledge tracing. BKT takes the learning end data as input and defines the probability of initially learning a concept $P(L_0)$, the probability of transferring an unlearned state to a learned state $P(T)$, the probability of not mastering a concept but guessing correctly $P(G)$, the probability of the probability of mastering a concept but answering incorrectly $P(S)$, etc., and the Hidden Markov Model (HMM) [8] is used to model the relationship between the above four probabilities to predict students' future learning performance.

Deep Knowledge Tracing (DKT) [3] first used deep sequential models to solve the task of knowledge tracing. Similar to BKT, DKT still uses learning end data as input to represent students' knowledge states with the hidden states of Recurrent Neural Network (RNN) [9] or Long Short-Term Memory (LSTM) [10], and finally a fully connected layer to predict students' future learning performance.

Dynamic Key-Value Memory Networks (DKVMN) [4], inspired by standard memory enhancement networks [11], proposes a memory matrix approach to solve the task of knowledge tracing. DKVMN still uses learning end data as input, a key matrix to store concepts, and a value matrix to store the student's mastery state of the concept; the model uses these two matrices to determine the student's mastery state of each concept at each learning session and finally outputs the probability of the student's future learning performance in a fully connected layer.

In subsequent studies, researchers still modelled students' knowledge states using only learning end data as the input to the model: Kaser et al. [12] proposed a dynamic Bayesian knowledge tracing model based on BKT to model the dependencies between different concepts; Su et al. [13] added exercise information to the input of the model based on DKT; Abdelrahman et al. [14] used a Hop-LSTM network structure on top of DKVMN, enabling the model to capture long-term boundedness in student learning records. Other variants of such models include TLS-BKT [15], Multigrained-BKT [16], PDKT-C [17], HMN [18], and other models.

BKT, DKT, and DKVMN are classical knowledge tracing models, and these models have laid a solid foundation for the subsequent studies. Their shortcomings are that only learning end data are used to trace students' knowledge states, modelling constraint in the same behaviours. Learning process data and learning interval data are not used, not modelling collaboration in different behaviours, so they cannot provide more adequate support for representing students' knowledge states.

2.1.2. Knowledge Tracing Models Based on Learning Interval Behaviour. Some of the studies used learning interval data: Nagatani et al. [6] were inspired by the Ebbinghaus forgetting curve [19] and added learning interval data as input to the DKT model. They considered learning interval data as a factor affecting forgetting behaviours and were able to model forgetting behaviours by adding learning interval data as input to the model. Inspired by the memory traces of decline said [20], the study by Li et al. [7] proposed the LFKT model, which considered not only the above learning interval data but also the effect of students' conceptual mastery status on forgetting.

Although these two models add learning interval data to the use of learning end data and achieve better results, they still model only constraint in same behaviours and neglect to model collaboration in different behaviours.

2.1.3. Knowledge Tracing Models Based on Learning Process Behaviour. Some of the studies used learning process data: Cheung and Yang [5] input the learning process data to a Classification And Regression Tree (CART) to predict whether students could answer the exercises correctly, then combined the predicted results with the real results, and finally input the combined data and the learning end data to DKT The combined results are then combined with the real results, and finally the combined data and the learning end data are fed into the DKT model to predict their future answers. This method uses the learning process data as a complement to the learning end data to improve the method of modelling constraint in the same behaviours but does not yet model collaboration in different behaviours.

In general, most studies use only learning end data as input or introduce multiple types of learning behaviours data as input when tracing students' knowledge states, but none of them model collaboration in different behaviours. To address the above problems, this paper proposes a model: knowledge tracing via attention enhanced encoder-decoder, which models collaboration in different behaviours while modelling constraint in the same behaviours to provide a more adequate support for representing students' knowledge states.

2.2. Attentional Mechanisms. A biological perspective on attention mechanisms is based on the principle that humans selectively direct the focus of their attention based on nonvolitional cue and volitional cue [21]. Nonvolitional cue refers to the fact that a person is not cognitively and consciously driven to access information, and volitional cue refers to the fact that a person is cognitively and consciously driven to access information. In attention mechanisms, queries refer to volitional cues, keys, and values refer to nonvolitional cues. The benefit of adding volitional cues is to bias the output of the attention mechanism towards certain input data, rather than taking in the input data wholesale.

For example, in determining the state of students' knowledge, student S answered correctly the exercise about the concept of third-person singular in learning English. If there is no cognitive and consciousness drive and only the learning end data is used as the criterion, the teacher's attention is guided by the non-volitional cue and judges the mastery status of student S on the concept of third-person singular; however, if there is a cognitive and consciousness drive, on top of the learning end data, the teacher will also notice the learning process data and learning interval data of the student. The teacher's attention is guided by the volitional cue to judge the state of student S's mastery of the third-person singular concept.

Ghosh et al. [22] proposed the AKT model to solve the knowledge tracing task by constructing context-aware representations of exercises and outcomes and summarizing students' past performance using attention mechanisms. The inputs of the attention mechanisms are query, key, and value, and the output is a weighted sum of values, and the attention weights are obtained by calculating the similarity of query and key. The self-attention mechanism is a variant of the attention mechanism, which has inputs from the same data and is better at capturing the similarity within the data and reduces the dependence on external data because there is no input from external data. Pandey et al. [23] proposed the SAKT model, which first applied the Transformer model [24] to the domain of knowledge tracing by describing the inputs in terms of temporal constraint relations to solve knowledge tracing tasks. The main structure of the Transformer model is a multiheaded attention mechanism, consisting of multiple attention mechanisms or self-attention mechanisms in parallel, where a fully connected layer maps the input data to different subspaces and is able to learn different weights based on the same mechanism, which is used to describe constraint in same behaviours.

The disadvantage of the multiheaded attention mechanism using learning process data, learning end data, and learning interval data as volitional cues is that different learning behaviours are treated as having the same weight when tracing knowledge states. The channel attention mechanism solves this problem [25–28] by using three types of learning behaviours data as input to the channel attention mechanism, the "squeeze" operation collects global information about the three types of learning behaviours data, and the "stimulate" operation converts the global information into attention weights, which represent the interaction of multiple types of learning behaviours, are used to describe collaboration in different behaviours.

3. Materials and Methods

3.1. The Idea Proposed by the Model. As a whole, the learning sequence includes several different types of learning behaviours, such as the learning process behaviour, the learning end behaviour, and the learning interval behaviour. In this paper, we use learning process data b^{I}, learning end data b^{II}, and learning interval data b^{III} to describe the above three types of learning behaviours, in which b^{I} mainly includes data such as the number of attempts of students to answer and the number of requests for hints; b^{II} mainly includes data such as the exercises students answer and the results of their answers; b^{III} mainly includes data such as the time interval between two adjacent learning sessions and the number of times students learn a concept. This paper finds that the learning behaviours possess the constraint in the same behaviours and collaboration in different behaviours. The specific descriptions are as follows:

According to the literature [29], changes in students' knowledge states are bounded by their pre-existing knowledge states and are manifested in the constraint in the same behaviours, i.e., changes in knowledge state are reflected in a learning behaviour is gradual. Specifically, the constraint in the learning process data b^I may be manifested by the fact that the change in the number of attempted responses for a given exercise is smooth at adjacent time steps; the constraint in the learning end data b^{II} may be manifested by the fact that the change in the result of a student's response to a given exercise is also smooth; the constraint in the learning interval data b^{III} may be manifested by the fact that the change in a number of adjacent learning intervals is also flat. From a modelling perspective, the characterization of the three types of learning behaviours data should take into account their respective similarity constraints to reflect the objective changes in students' knowledge states, which are neglected in the current study.

According to the literature [30], the interaction of multiple types of learning behaviours in a learning sequence is manifested by the collaboration in different behaviours. Specifically, the collaboration in learning process data b^I and learning end data b^{II} may be manifested in that the probability of correct answers for a given exercise is lower when students have more attempts and higher when they have fewer attempts; the collaboration in learning interval data b^{III} and learning end data b^{II} may be manifested in that the probability of correct answers for a given exercise is lower when students have a longer learning interval and higher when they have a shorter learning interval. From the modelling point of view, the collaboration in different learning behaviours data should be considered in order to reflect the objective changes of students' knowledge state, which is neglected in the current study.

BKT uses learning end data b^{II} to trace the student's knowledge state. However, because b^{II} only contains information about students answering a certain exercise correctly or incorrectly, and the model does not express constraint in learning end behaviour on the time series. Although subsequent studies [12–15] still used only the learning end data b^{II}, they mostly used deep models, so there was some progress in modelling the boundedness of the learning end behavior. Subsequently, some researchers added learning process data b^I [5] and learning interval data b^{III} [6, 7] to the input of the model to improve the performance of the model. Although these studies validated the validity of other learning behaviours, they did not model collaboration in different behaviours in a learning sequence.

In summary, it is advantageous to integrate multiple types of learning behaviours data when tracing students' knowledge states, which enables knowledge tracing models to more accurately predict students' future performance. However, when modelling learning behaviours, the constraint in the same behaviours and collaboration in different behaviours should be considered in an integrated manner.

In this paper, we use the multiheaded attention mechanism to adaptively assign the weights of each type of learning behaviours data itself, so as to model constraint in the same behaviours; and the channel attention mechanism to adaptively assign the weights between different types of learning behaviours data, so as to model collaboration in different behaviours.

3.2. Definition of Learning Behaviours Data. In this paper, we define three types of learning behaviours data as follows: learning process data $b_t^I = (\mathrm{AN}, \mathrm{RN}, \mathrm{FA})$ describes the learning process behaviour of the student's $t, t \geq 1$ th learning record, where $\mathrm{AN} \in \mathbb{N}$ indicates the number of times the student attempted to answer; $\mathrm{RN} \in \mathbb{N}$ indicates the number of times the student requested a hint; $\mathrm{FA} = \{0, 1\}$ indicates the first action of the student when answering the exercise, where 1 indicates that the student first attempted to answer and 0 indicates that the student first requested a hint. $B^I = (b_1^I, b_2^I, \cdots b_n^I)$ is the set of learning process data b^I, i.e., it is composed of learning process data $b_n^I, n \geq 1$.

The learning end data $b_t^{II} = (q_{t,} r_t)$ describes the learning end behaviour of the student's $t, t \geq 1$ th learning record, where $q_t \in \mathbb{N}$ indicates the exercise that the student answered; $r_t = \{0, 1\}$ indicates the result of the student's answer, where 1 indicates that the student answered the exercise correctly and 0 indicates that the student answered the exercise incorrectly. $B^{II} = (b_1^{II}, b_2^{II}, \cdots b_n^{II})$ is the set of the learning end data b^{II}, which is composed of the learning end data $b_n^{II}, n \geq 1$.

The learning interval data $b_t^{III} = (\mathrm{RT}, \mathrm{ST}, \mathrm{LT})$ describes the learning interval behaviour of the student's $t, t \geq 1$ th learning record, where $\mathrm{ST} \in \mathbb{N}$ indicates the time interval between the student's $t - 1$ th learning and t th learning; $\mathrm{RT} \in \mathbb{N}$ indicates the time interval between the student's learning of the current concept; $\mathrm{LT} \in \mathbb{N}$ indicates the number of repetitions of the current concept. $B^{III} = (b_1^{III}, b_2^{III}, \ldots, b_n^{III})$ is the set of the learning end data b^{III}, i.e., it consists of the learning interval data $b_n^{III}, n \geq 1$. Figure 2 shows the learning behaviours described by the learning behaviours data in the learning sequence.

3.3. Knowledge Tracing via Attention Enhanced Encoder-Decoder. In this paper, we propose Knowledge Tracing via Attention Enhanced Encoder-Decoder (AED-KT), whose overall architecture is shown in Figure 3.

The model consists of five components: an input module, an encoder, a decoder, a conceptual attention module, and a prediction module. The input module embedding represents a number of continuous learning behaviours data. The encoder models constraint in the same behaviours and collaboration in different behaviours. The decoder generates students' learning and forgetting vectors and updates the state matrix M_{t-1}^v. The conceptual attention module is used to capture the similarity between concepts. The prediction module predicts students' answers at moment t based on the state matrix M_{t-1}^v, the concept matrix M_{t-1}^k, and the exercise $q_t, t \geq 1$. Concept matrix M^k represents the concept and state matrix M^v represents the student's concept mastery state, and these two matrices are dynamically updated with the learning sequence.

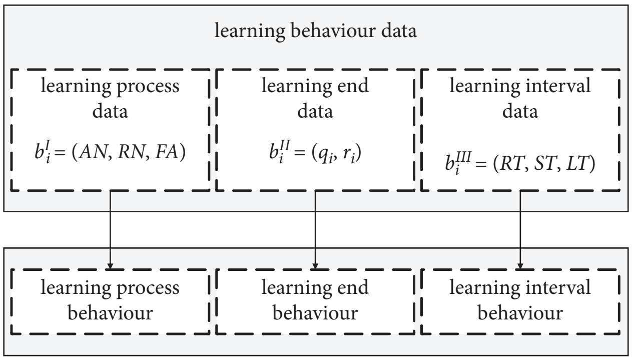

Figure 2: Correspondence between learning behaviours and their data.

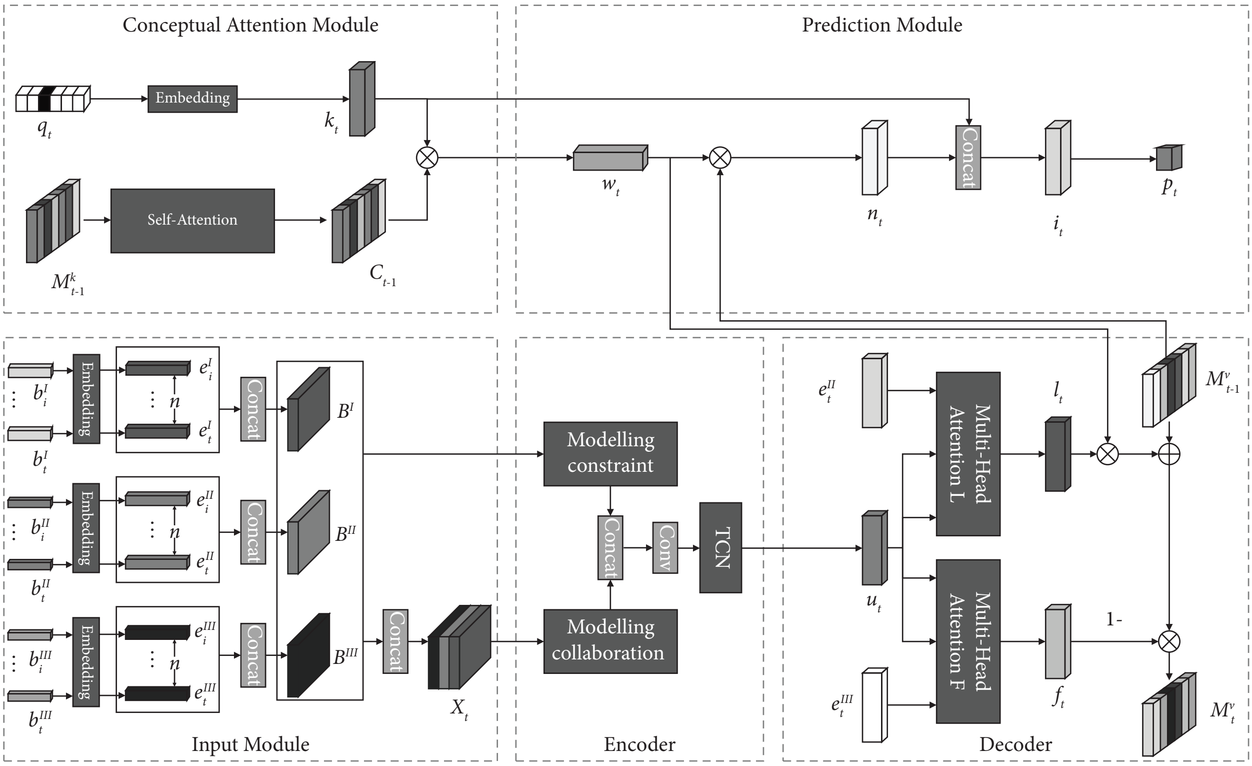

Figure 3: Knowledge tracing via attention enhanced encoder-decoder.

3.3.1. *Input Module.* The learning process data $b_i^I = (\mathrm{AN}, \mathrm{RN}, \mathrm{FA}), i \geq 1$ is represented as a row vector: $b_i^I \epsilon \mathbb{R}^{1\times 3}$, and multiplied with the embedding matrix $C^I \epsilon \mathbb{R}^{1\times 3}$ to obtain vector $e_i^I \epsilon \mathbb{R}^{1\times d_v}$. The learning end data $b_i^{II} = (q_i, r_i), i \geq 1$ is transformed into one-hot encoding: $b_i^{II} \epsilon \mathbb{R}^{1\times 2N}$, and multiplied with the embedding matrix $C^{II} \epsilon \mathbb{R}^{2N\times d_v}$ to obtain vector $e_i^{II} \epsilon \mathbb{R}^{1\times d_v}$ in order to solve the problem of b_i^{II} sparsity. The learning interval data $b_i^{III} = (\mathrm{RT}, \mathrm{ST}, \mathrm{LT}), i \geq 1$ is represented as a row vector: $b_i^{III} \epsilon \mathbb{R}^{1\times 3}$, and multiplied with the embedding matrix $C^{III} \epsilon \mathbb{R}^{1\times 3}$ to obtain vector $e_i^{III} \epsilon \mathbb{R}^{1\times d_v}$.

The learning behaviours data with n consecutive embedding representations are taken, and then three matrices B^I, B^{II} , and B^{III} of size $n \times d_v$ are combined according to the learning behaviour types as the input of the multi-headed attention mechanism; these three matrices are stitched into a three-dimensional array X_t of size $3 \times n \times d_v$ as the input of the channel attention mechanism, where 3 indicates that the array X_t contains three types of learning behaviours, and n indicates that the array X_t contains n consecutive learning behaviours, and d_v is the dimension of the vector representation of the learning behavior data.

3.3.2. *Encoder.* The array X_t consists of matrices B^I, B^{II} , and B^{III} stitched together, and each of these three matrices includes n consecutive learning behaviours data, which represent three different types of learning behaviours: learning process, learning end, and learning interval behaviour.

(1) Modelling Constraint in Same Behaviours. It is important to note that each type of learning behaviour has a constraint on subsequent similar behaviours on the learning sequence,

i.e., the constraint between the same type of learning behaviour in the learning sequence on the time sequence. Because the multiheaded attention mechanism can locate similar information on the learning sequence and translate it into the relative weights of the learning records in the sequence, the multiheaded attention mechanism is used to model constraint in the same behaviours, and the specific process is shown in Figure 4.

Firstly, using the parameter $v \epsilon \mathbb{R}^{1\times n}$ as the position code, which represents the relative position of the data of n consecutive learning behaviours in time sequence, it is added to the input matrices B^{I}, B^{II} , and B^{III} to form a learning behavior matrix containing relative position information in time sequence:

$$B^{j*}(i) = B^{j}(i) + v(i), j \in \{I, II, II\}, i \in \{1, \ldots, n\}. \tag{1}$$

Secondly, the learning behaviour matrices B^{I*}, B^{II*} and B^{III*} are input into the multi-headed attention mechanism, and the attention weights are obtained by calculate the similarity between each learning behaviour, which is used to model constraint in the same behaviours, and the magnitude of the attention weights indicates the strength of the learning behaviour constraint relationship. The output matrices X_B^{I}, X_B^{II} and X_B^{III}, which represent the constraint of learning process behaviour, learning end behaviour, and learning interval behaviour, respectively:

$$X_B^{j} = \text{MultiHead}\left(B^{j*}, B^{j*}, B^{j*}\right), j \in \{I, II, II\}. \tag{2}$$

Finally, these three output matrices are stitched together into a three-dimensional array $X_B \in \mathbb{R}^{3\times n\times d_v}$, which represents the constraint in the same behaviours.

$$X_B = \text{Concat}\left(X_B^{I}, X_B^{II}, X_B^{III}\right). \tag{3}$$

(2) Modelling Collaboration in Different Behaviours. It is also important to note that there is a mutual collaboration between multiple types of learning behaviours, i.e., the interaction of multiple types of learning behaviours in a learning sequence. Because the channel attention mechanism is able to capture the global information of multiple types of learning behaviours and translate it into the relative weights of each learning behaviour, the collaboration in different behaviours is modelled using the channel attention mechanism, as shown in Figure 5.

Using the array X_t as the input of the channel attention mechanism, the attention weights are obtained by collecting the global information of three types of learning behaviours to model collaboration in different behaviours, and the magnitude of the attention weights indicates the degree of collaboration of learning behaviours. The squeeze operation collects the global information of learning behaviours, and the excitation operation translates the above global information into attention weights s among different learning behaviours through a fully connected layer:

$$s = \text{Sigmoid}\left(W \bullet \text{RC}\left(\text{Cov}\left(X_t\right)\right)\right), \tag{4}$$

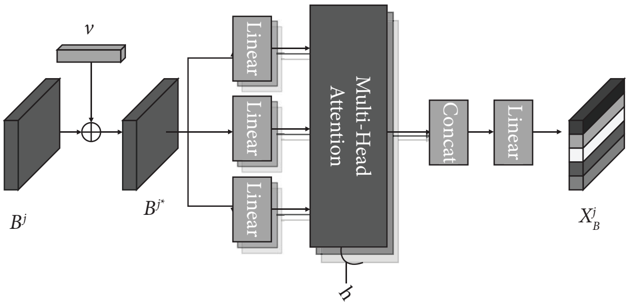

Figure 4: Modelling constraint in same behaviours.

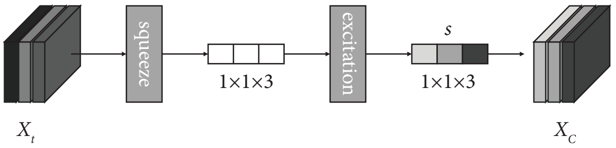

Figure 5: Modelling collaboration in different behaviours.

where Sigmoid $= 1/\left(1 + e^{-x_i}\right)$, the weight matrix of the fully connected layer is W, RC denotes the rowwise convolution, and Cov$(\bullet)$ denotes the calculation of the covariance matrix, which is used to characterize the degree of correlation between the three types of learning behaviours.

The output attention weight s represents the collaboration in different behaviours, which is multiplied with the array X_t by the channel, changing the expression of the array X_t eigenvalues to obtain the array X_C, representing the collaboration in different behaviours assigned to the array X_t:

$$X_C = s \bullet X_t. \tag{5}$$

The array $X' \in \mathbb{R}^{6\times n\times d_v}$ is obtained by summing the array X_B, which represents the constraint in the same behaviours, and the array X_C, which represents the collaboration in different behaviours. By using the convolution kernel of $6 \times 1 \times d_v$, the dimension of array X' is reduced by row-wise convolution to obtain the output matrix $X_E \in \mathbb{R}^{\text{n}\times d_v}$:

$$X_E = \text{RC}\left(\left[X_B, X_C\right]\right). \tag{6}$$

The temporal convolutional networks (TCNs) model uses a 1D Fully Convolutional Networks (1D FCNs) structure to ensure that the input sequence and output of each hidden layer have the same length so that no matter which layer of the network, the input at each time has a corresponding output. In addition, TCN uses causal convolution to satisfy the feature that sequence data does not use future information, that is, when the model outputs the results at time t, it can only input data before time t.

Furthermore, the TCN model uses the derived calcium transformations to obtain longer historical information, avoiding the construction of deeper neural networks. For one-dimensional input sequence $X = \{x_1, x_2, x_3, \ldots, x_t\}$, convolution kernel $f = \{0, 1, 2, \ldots, k-1\}$, the expansion convolution operation can be expressed as follows:

$$\mathrm{TCN}\left(x_t\right) = \sum_{i=0}^{k-1} f\left(i\right)x_{t-ic}, \tag{7}$$

where c is the expansion coefficient; k is the size of convolution kernel; x_{t-ic} represents the past data.

3.3.3. Decoder. The decoder consists of two multi-headed attention mechanisms, which generate the learning vector and forgetting vector, respectively, through matrix X_E. The structure is shown in Figure 6. Firstly, the t th learning end data e_t^{II} is used as query input to represent the learning vector l_t in different dimensions; secondly, the t th learning interval data e_t^{III} is used as query input to represent the forgetting vector f_t in different dimensions; finally, the state matrix M^v is updated according to the vectors l_t and f_t.

The decoding vector $u_t \in \mathbb{R}^{1\times d_v}$ is obtained by feeding the matrix X_E into the TCN to obtain. u_t represents the integration of constraints in the same behaviour and collaboration in different behaviours:

$$u_t = \mathrm{TCN}\left(X_E\right). \tag{8}$$

The decoding vector u_t contains the constraint in the same behaviours and collaboration in different behaviours, which is used as the input to the key and value in the multiheaded attention mechanism L and F in Figure 6.

In the multiheaded attention mechanism L, the learning vector l_t is obtained using the vector e_t^{II} as the query input:

$$l_t = \mathrm{Tanh}\left(W_l^T \mathrm{Softmax}\left(e_t^{II} u_t^T\right) u_t + b_l\right), \tag{9}$$

where $\mathrm{Softmax}\left(x_i\right) = x_i / \sum_{n=1}^{N}\left(\left[ee\right]^{x_n}\right)$, vector e_t^{II} is the transformed learning end data, the vector describes the information of students' answer situation, and using it as the query input of decoding process can get the change of students' knowledge state due to the t th learning.

In the multiheaded attention mechanism F, the forgetting vector f_t is obtained with the vector e_t^{III} as the query input:

$$f_t = \mathrm{Sigmoid}\left(W_f^T \mathrm{Softmax}\left(e_t^{III} u_t^T\right) u_t + b_f\right), \tag{10}$$

where vector e_t^{III} is the processed learning interval data, which describes the learning behaviour such as the time interval between two adjacent learning sessions and the number of times a student learns a concept and using it as the query input of the decoding process can obtain the changes of the students' concept mastery status due to forgetting.

The learning vector l_t and the forgetting vector f_t and the associated weights w_t are used to update the concept state matrix at the current moment, the association weights w_t will be described in the prediction module:

$$M_t^v\left(i\right) = M_{t-1}^v\left(i\right)\left(1 - f_t\right) + l_t w_t\left(i\right). \tag{11}$$

3.3.4. Conceptual Attention Module. The conceptual attention module uses the self-attention mechanism to strengthen the relationship between concepts according to the similarity between concept vector representations. The more similar the concept vector representations are, the stronger the relationship is.

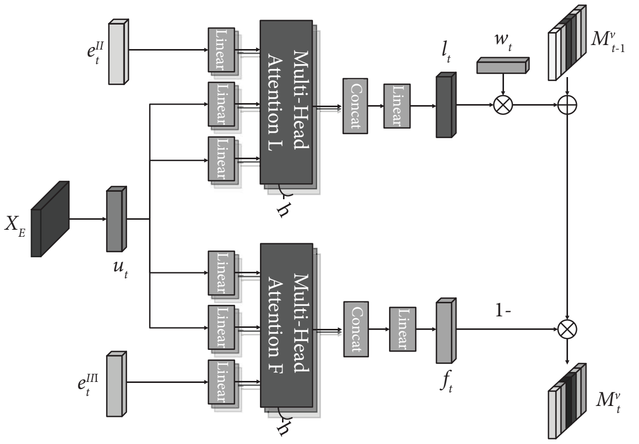

Figure 6: Decoder.

First, the exercise q_t is converted into one-hot encoding and multiplied with the embedding matrix $A \in \mathbb{R}^{d_v\times \mathrm{N}}$ to obtain the exercise embedding vector k_t with dimension d_k, which describes the information related to the exercise q_t.

Second, the self-attention mechanism is used to strengthen the connection between concepts with high similarity, and the output matrix C is obtained:

$$C_{t-1} = \mathrm{Attention}\left(M_{t-1}^k T \times M_{t-1}^k\right) M_{t-1}^k. \tag{12}$$

Finally, vector k_t is multiplied with the concept matrix $C \in \mathbb{R}^{d_v\times \mathrm{N}}$ that stores the concepts and transformed into the associated weights w_t by the Softmax function, which is used to describe the concepts contained in the exercise q_t:

$$w_t = \mathrm{Softmax}\left(k_t \times C_{t-1}\right). \tag{13}$$

3.3.5. Prediction Module. The prediction module is used to predict students' future answers. First, the association weights w_t are multiplied with the state matrix M_{t-1}^v to obtain the vector n_t, which represents the student's mastery status of the concepts contained in exercise q_t:

$$n_t = w_t M_{t-1}^v. \tag{14}$$

Second, considering that there are certain differences between the exercises, such as different difficulty coefficients, the vector n_t is spliced with the vector k_t and fed into the fully connected layer with Tanh activation function to obtain the vector i_t. The vector i_t contains both the student's mastery state of the concept and the information of the exercise:

$$i_t = \mathrm{Tanh}\left(W_2^T\left[n_t, k_t\right] + b_2\right). \tag{15}$$

Finally, an output layer with a Sigmoid activation function, using vector i_t as input, is used to predict student performance on the exercise q_t:

$$P_t = \text{Sigmoid}\left(W_3^T i_t + b_3\right). \tag{16}$$

3.4. Loss Function. In this paper, the cross-entropy loss function is chosen to minimize the variability between the predicted value P_t and the true value r_t:

$$\text{Loss} = -\sum_t \left(r_t \log P_t + (1 - r_t)\log\left(1 - P_t\right)\right). \tag{17}$$

4. Experiments and Analysis

4.1. Data Set and Experimental Environment. The experiments related to this paper are conducted on three real datasets: ASSISTments2012 (Assist12), ASSISTments2017 (Assist17), and Junyi Academy (Junyi). In each dataset, 70% of the data were used as a training set and 30% of the data were used as a test set. The basic information of the above datasets is shown in Table 1, including the number of students, the number of learning records, and the number of concepts.

The experiments in this paper are implemented under the Windows system with GeForce graphics acceleration units, based on python and PyTorch platforms, with the hardware and software configurations shown in Table 2.

Table 1: Datasets information.

Data set	Learners	Records	Concepts
Assist12	46674	5818868	266
Assist17	1709	942816	102
Junyi	238120	26666117	684

Table 2: Experiment environment.

Experimental configuration	Parameter value
Operating system	Windows 11
CPU	Inter(R) core(TM) i9-9900K CPU@ 3.60 GHz
GPU	NVIDIA GeForce RTX 3080 Ti
Python	3.10
PyTorch	1.10.2
RAM	64 GB

4.2. Implementation Details. In each dataset, 80% of the data is divided into a training set and 20% of the data was divided into a test set. Twenty percent of the data in the training set was divided into the validation set, which was used to select the hyperparameters of the best model. Considering that the data sets differ in the number of learners, the number of exercise interactions, and the number of concepts, the learning rate was initialized to 0.001 and reduced by 10 every 10 epochs. Adam was chosen as the optimizer with the batch-size set to 32. The initialization of the parameters was chosen to be randomly initialized with a Gaussian distribution with zero mean and standard deviation.

The AED-KT model also focuses on the input set size n, the dimension d_v of the state matrix, the dimension d_k of the state matrix, and the expansion coefficient of TCN c. To facilitate the calculation, set $d = d_v = d_k$. In the dataset ASSISTments2012, the input set size n was set to 32, the dimension d was set to 64, and the expansion coefficient $c = \{1, 2, 4, 8, 16\}$. In the dataset ASSISTments2017, the input set size n was set to 32, the dimension d was set to 64, and the expansion coefficient $c = \{1, 2, 4, 8, 16\}$. In the dataset Junyi, the input set size n was set to 32, the dimension d was set to 64, and the expansion coefficient $c = \{1, 2, 4, 8, 16\}$.

4.3. Evaluating Indicator and Baseline

4.3.1. Evaluating Indicator. The performance of the AED-KT model proposed in this paper is analysed and evaluated using the metric Area Under Curve (AUC), which is the area of the graph enclosed by the ROC curve and the horizontal axis, and the value of this area is between 0.5 and 1. If the value of AUC is 0.5, it means that the model is a stochastic prediction model; the larger the value of AUC, the better the prediction performance of the model.

4.3.2. Baseline. The core of the AED-KT model is to use three types of learning behaviours as input and model the constraint in the same behaviours and collaboration in different behaviours using the multihead and channel attention mechanisms, respectively. Based on the above-mentioned theory, we mainly consider the following three conditions when selecting the comparison model: first, the comparison model belongs to the widely accepted model with the best-in-class performance, second, the type of input learning behaviours data of the comparison model, and third, the comparison methods model constraint in same behaviours and collaboration in different behaviours. Based on the above-mentioned three conditions, we choose the following model as the comparison model:

DKT [3]: DKT is the first time to use the deep learning method in the field of knowledge tracing, using high-dimensional vectors in RNN or LSTM to represent the students' knowledge state. But there are problems of long sequence dependency and gradient explosion, and students' specific mastery of each concept cannot be obtained according to the vector.

DKVMN [4]: DKVMN uses a static matrix to store concepts and a dynamic matrix to store knowledge states. Thanks to this design, DKVMN is able to know the student's knowledge state for each concept, solving the problem that DKT uses a hidden state to represent the student's overall knowledge state. But without the long-term dependence of modelling sequence data.

SAKT [24]: SAKT is based on the Transformer model to accomplish the knowledge tracing task. Thanks to the Transformer architecture, the SAKT model can be trained in parallel, solving the problem that recurrent neural networks cannot be trained in parallel.

DKT-F [6]: DKT-F models forgetting behaviour based on the DKT model by introducing learning interval data as input, but the forgetting mechanism is not interpretable.

DKT-DT [5]: DKT-DT introduces learning process data as input on the basis of DKT, while using the decision tree method to analyse learning process data and feature selection on learning process data, the model can analyse richer feature energy and judge students' knowledge status more effectively.

TCN-KT [31]: TCN-KT used LSTM to model the students' prior basis, combined with temporal convolution neural network to complete the knowledge tracing task, and solved the problem that DKT could not obtain long-term dependence.

The main reason is that all of these models take as input some or all of the learning behaviours data and model constraint in the same behaviours or collaboration in different behaviours. Specifically, DKT, DKVMN, and TCN-KT use learning end data as input and model constraint in the same behaviours using a sequence model; SAKT also uses learning end data as input and also models constraint in the same behaviours using a multiheaded attention mechanism; DKT-F and DKT-DT add learning interval and learning process data, respectively, as input, in addition to using learning end data. The constraint in the same behaviours is modelled using the sequence model, and collaboration in different behaviours is not modelled.

4.4. Model Performance Comparison. The results of the performance comparison experiments are shown in Table 3.

The above-mentioned models can be divided into two categories: single learning behaviour models, which refer to models that use only learning end data as input and multiple learning behaviours models, which refer to models that use learning end data as input and introduce other learning behaviours data.

The single learning behaviour models are mainly DKT, DKVMN, and SAKT. Among them, the AUC values of SAKT reached 0.734 and 0.853 on the Assist17 and Junyi datasets, which are the highest of their kind and have good overall performance. Although all three models use learning end data as input, the differences in modelling constraint in the same behaviours lead to differences in model performance.

The main multilearning behaviours models are DKT-F, DKT-DT, and AED-KT. The first two models introduce other learning behaviours data as inputs, which do not improve the method of modelling collaboration in different behaviours but improve the inputs of the models, and all perform better compared with single learning behaviour models. The AUC values of AED-KT outperform the other models on all three datasets. This illustrates the effectiveness of modelling collaboration in different behaviours based on modelling constraint in the same behaviours.

4.5. Training Process Comparison. We use the early stop strategy to compare the number of training rounds required for the model to reach the same performance. Using the early stop strategy can avoid overfitting the model during training. To verify the impact of using the early stopping strategy, we trained 200 epochs for AED-KT. The experimental results are shown in Table 4.

Table 3: Model performance comparison.

Model	Date set		
	Assist12	Assist17	Junyi
DKT	0.717 [32]	0.726 [33]	0.814 [34]
DKVMN	0.732 [34]	0.707 [33]	0.822 [34]
SAKT	0.691	0.734 [24]	0.853
DKT-F	0.722 [34]	0.729	0.840 [34]
DKT-DT	0.749	0.721	0.741 [5]
TCN-KT	0.743	0.732	0.758
AED-KT	0.768 ± 0.003	0.815 ± 0.005	0.864 ± 0.004

Table 4 shows that with the early stop strategy, the deviation between the AUC of the Asisstment2012 training set and the ASSISTments2012 test set is not large, and there is no fitting phenomenon. However, after 200 epochs of training the model directly, there is a large deviation between the AUC of the Asisstment2012 training set and the Asisstment2012 test set, and there is a problem of overfitting. In addition, the AUC of the training model with an early stop strategy is 0.768 on the Asisstment2012 training set, and the AUC of the model with 200 training cycles is 0.757. The difference is in a reasonable range, and there is no obvious underfitting phenomenon.

Furthermore, we explored the time cost of AED-KT and the comparison models when training the same epochs, and the experimental results are shown in Figure 7.

As can be seen from Figure 7, the time cost of AED-KT and TCN-KT training is lower than that of DKVMN, DKT-F, and SAKT. This is because AED-KT and TCN-KT use TCN to build models. TCN does not need to process data in time sequence to the recurrent neural network, which reduces the time cost. Compared with TCN-KT, AED-KT costs less time. This is because TCN-KT first uses LSTM to model students' prior basis, and then models their knowledge level.

4.6. Comparison of Learning Behaviours. The model performance comparison results show that the introduction of other learning behaviours as inputs can lead to improved model performance. To further compare and analyse the importance of the three types of learning behaviours in the model, we adjust the inputs of the default AED-KT model: AED-e indicates that the model uses only learning end data b^{II} as input; AED-pe indicates that the model uses learning process data b^{I} and learning end data b^{II} as input; AED-ei indicates that the model uses learning end data b^{II} and learning interval data b^{III} as input. The above-given models and their AUC values on the three datasets are given in Table 5.

The experimental results show that the AUC values of the AED-e model are lower than those of the other models on the three real data sets, indicating that analysing only the constraint in learning end behaviour can basically determine

Table 4: Experimental results of different training strategies.

Method	Assist12 training set	Assist12 testing set
Early stop	0.768	0.757
200 epochs	0.757	0.729

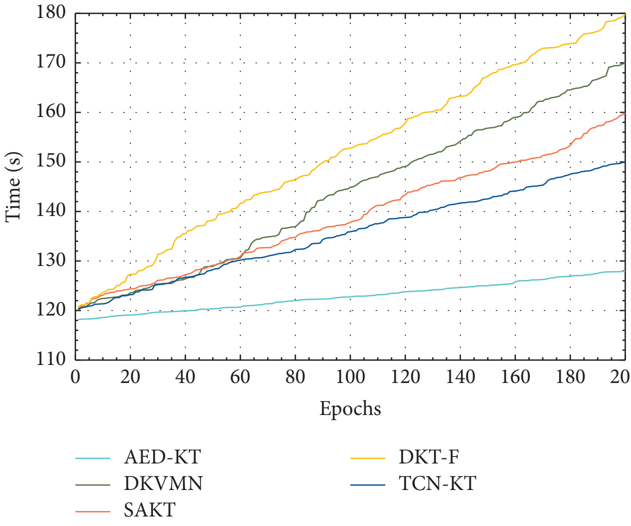

Figure 7: Time cost.

Table 5: The impact of different data on model performance.

Model	Data set		
	Assist12	Assist17	Junyi
AED-e	0.724 ± 0.004	0.778 ± 0.007	0.829 ± 0.005
AED-pe	0.763 ± 0.005	0.805 ± 0.006	0.856 ± 0.004
AED-ei	0.761 ± 0.004	0.799 ± 0.006	0.844 ± 0.006
AED-KT	0.768 ± 0.003	0.815 ± 0.005	0.864 ± 0.004

the students' knowledge status, but since b^{II} only contains learning end data of students' correct or incorrect answers, it contains limited information and cannot model the constraint in the same behaviours more accurately. The AED-pe and AED-ei models have higher AUC values than AED-e, indicating that introducing other learning behaviours data as input and modelling both constraint in the same behaviours and collaboration in different behaviours can improve the performance of the model on the basis of using learning end data as input; however, the AUC values of these two models are lower than AED-KT, indicating that on the basis of modelling both constraint in same behaviours and collaboration in different behaviours. However, the AUC values of these two models are lower than those of AED-KT, indicating that the more comprehensive the learning behaviours analysed by the model, the higher the performance of the model.

4.7. Encoder Ablation Experiment. To analyse modelling constraint in the same behaviours and collaboration in different behaviours impact on model performance, we ablation the attention mechanisms in the encoders that model the constraint in the same behaviours and collaboration in different behaviours: the model in the AED-B representation models only the constraint in same behaviours, and the model in the AED-C representation models only collaboration in different behaviours. The above-given models and their AUC values on the three datasets are given in Table 6.

The experimental results show that the AUC values of the AED-KT model are better than the other two models on all three data sets, indicating that it is effective to analyse the constraint in the same behaviours and collaboration in different behaviours when considering the three types of learning behaviours together. The AUC values of the AED-C model are lower than those of the AED-B model on the three datasets, indicating that modelling only the collaboration in different behaviours while ignoring the similarity constraint will result in the model losing the constraint relationship of learning behaviours on the time series and will not improve the performance of the model. The AUC values of the AED-B model on the three real data sets is lower than those of the AED-KT model, indicating that modelling only the constraint in the same behaviours and ignoring the collaboration in different behaviours leads to the loss of the interactions of multiple types of learning behaviours, which also fails to improve the performance of the model.

4.8. Comparison of Model Representation Quality. The representation quality of a model refers to the overall difference between the predicted and actual results of the model in a real application. For example, the knowledge tracing model KT is trained from a real data set and has good predictive performance. When the model KT is applied to a real teaching environment if the model predicts that 40% of the students will answer the questions about concept C incorrectly, but the actual results show that only 10% of the students answer the questions about concept C incorrectly, this indicates that the overall difference between the prediction results and the actual results of the model KT in practice is large and the representation quality of the model needs to be improved.

The model KT in the above example has good predictive performance but performs poorly in practice and cannot be applied in a real teaching environment, indicating that both high predictive performance and the quality of the representation of the model are critical. The consistency between predicted and observed probabilities is generally measured using calibration curves [35], and the representation quality of each model is measured using the baseline alignment line $x = y$. A calibration curve that is closer to the baseline alignment line indicates that the model prediction probability is closer to the observed probability, i.e., the model has better representation quality. Table 7 shows the position of the calibration curve of each model in relation to the baseline (Assist12 is used as an example).

According to Table 7, first, the AED-KT model calibration curve value to the baseline at both lower and higher Prediction probability, indicating that the AED-KT model has better representation quality in comparison with the comparison model. Second, although the calibration curve

Table 6: Encoder ablation experiment.

Model	Data set		
	Assist12	Assist17	Junyi
AED-B	0.760 ± 0.007	0.806 ± 0.008	0.859 ± 0.007
AED-C	0.757 ± 0.005	0.788 ± 0.009	0.843 ± 0.006
AED-KT	0.768 ± 0.003	0.815 ± 0.005	0.864 ± 0.004

Table 7: Calibration curve value.

Model	Prediction probability								
Baseline	0.15	0.18	0.23	0.39	0.45	0.58	0.65	0.75	0.85
AED-KT	0.154	0.185	0.233	0.396	0.455	0.584	0.652	0.754	0.855
DKT	0.113	0.152	0.201	0.353	0.425	0.554	0.622	0.728	0.829
DKVMN	0.123	0.164	0.215	0.361	0.438	0.567	0.634	0.739	0.837
SAKT	0.130	0.165	0.222	0.373	0.436	0.573	0.642	0.741	0.844
DKT-F	0.161	0.194	0.242	0.400	0.459	0.594	0.664	0.764	0.863
DKT-DT	0.173	0.198	0.255	0.405	0.467	0.600	0.671	0.771	0.871
TCN-KT	0.165	0.181	0.251	0.412	0.468	0.597	0.673	0.780	0.864

value of the SAKT model is also close to the baseline, the calibration curve value of this model shows more dramatic fluctuations than that of the AED-KT model, i.e., the representation quality of the SAKT model is unstable on the whole. The results in Table 7 show that AED-KT is effective in considering multiple learning behaviours and modelling the constraint in the same behaviours and collaboration in different behaviours, which enables the model to not show severe bias and to obtain better representation.

5. Conclusions

In this paper, we propose AED-KT, a knowledge tracing model with multiple learning behaviours, to solve the problem that the existing knowledge tracing models cannot accurately describe the boundedness of a single type of learning behaviour in time series; or cannot accurately describe the interaction of multiple types of learning behaviours. AED-KT model uses a multiheaded attention mechanism to represent the constraint in the same behaviours and uses a channel attention mechanism to represent the collaboration in different behaviours. Fusing the constraint in the same behaviours and the collaboration in different behaviours to complete the synergistic representation of different types of learning behaviours. The experimental results of the proposed knowledge tracing model and five comparison models on three real datasets show that the proposed AED-KT model performs better and validates the effectiveness of the constraint in the same behaviours and collaboration in different behaviours. In the future, we will continue to investigate the impact of the constraint in the same behaviours and collaboration in different behaviours on the knowledge tracing model in depth.

Data Availability

The data used to support the findings of this study can be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (Grant no. 62077018); the Ministry of Science and Technology high-end foreign experts introduction plan (Grant no. G2022027006L); Natural Science Foundation Program of Hubei, China (Grant no. 2022CFB132); Hubei Provincial Department of Education Science Research Program (Grant no. B2022038).

References

[1] Q. Liu, S. Shen, Z. Huang, E. Chen, and Y. A. Zheng, "A survey of knowledge tracing," 2021, https://arxiv.org/abs/2105.15106.

[2] A. T. Corbett and J. R. Anderson, "Knowledge tracing: modeling the acquisition of procedural knowledge," *User Modeling and User-Adapted Interaction*, vol. 4, no. 4, pp. 253–278, 1995.

[3] C. Piech, J. Bassen, J. Huang et al., "Deep knowledge tracing," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

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

[5] L. P. Cheung and H. Yang, "Heterogeneous features integration in deep knowledge tracing," in *Proceedings of the International Conference on Neural Information Processing*, pp. 653–662, Springer, Cham, Guangzhou, China, November 2017.

[6] K. Nagatani, Q. Zhang, M. Sato, Y. Y. Chen, F. Chen, and T. Ohkuma, "Au0gmenting knowledge tracing by considering forgetting behavior," in *Proceedings of the The world wide web conference*, pp. 3101–3107, San Francisco, CA, USA, May 2019.

[7] X. G. Li, S. Q. Wei, X. Zhang, Y. F. Du, and G. Yu, "LFKT: deep knowledge tracing model with learning and forgetting behavior merging," *Journal of Software*, vol. 32, no. 3, pp. 818–830, 2021.

[8] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *Ieee Assp Magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[9] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *Proceedings of the International conference on machine learning*, pp. 1842–1850, PMLR, Honolulu, Hawai'i, June 2016.

[12] T. Kaser, S. Klingler, A. G. Schwing, and M. Gross, "Dynamic bayesian networks for student modeling," *IEEE Transactions on Learning Technologies*, vol. 10, no. 4, pp. 450–462, 2017.

[13] Y. Su, Q. Liu, Q. Liu et al., "Exercise-enhanced sequential modeling for student performance prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[14] G. Abdelrahman and Q. Wang, "Knowledge tracing with sequential key-value memory networks," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 175–184, Paris, France, July 2019.

[15] K. Zhang and Y. Yao, "A three learning states Bayesian knowledge tracing model," *Knowledge-Based Systems*, vol. 148, pp. 189–201, 2018.

[16] K. Zhang, "A three-way c-means algorithm," *Applied Soft Computing*, vol. 82, Article ID 105536, 2019.

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

[18] S. Liu, R. Zou, J. Sun et al., "A hierarchical memory network for knowledge tracing," *Expert Systems with Applications*, vol. 177, Article ID 114935, 2021.

[19] H. Ebbinghaus, "Memory: a contribution to experimental psychology," *Annals of Neurosciences*, vol. 20, no. 4, pp. 155-156, 2013.

[20] C. D. Bailey, "Forgetting and the learning curve: a laboratory study," *Management Science*, vol. 35, no. 3, pp. 340–352, 1989.

[21] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into deep learning," 2021, https://arxiv.org/abs/2106.11342.

[22] A. Ghosh, N. Heffernan, and A. S. Lan, "Context-aware attentive knowledge tracing," in *Proceedings of the 26th ACM SIGKDD international*, California, CA, USA, August 2020.

[23] S. Pandey and G. Karypis, "A self-attentive model for knowledge tracing," 2019, https://arxiv.org/abs/1907.06837.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, and L. Jones, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[25] M. H. Guo, T. X. Xu, J. J. Liu et al., "Attention mechanisms in computer vision: a survey," *Computational Visual Media*, vol. 8, no. 3, pp. 331–368, 2022.

[26] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, Salt Lake City, UT, USA, June 2018.

[27] Z. Gao, J. Xie, Q. Wang, and P. Li, "Global second-order pooling convolutional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3024–3033, Long Beach, CA, USA, January 2019.

[28] H. J. Lee, H. E. Kim, and H. Nam, "Srm: a style-based recalibration module for convolutional neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1854–1862, Montreal, Canada, October 2019.

[29] W. T. Fitch, M. D. Hauser, and N. Chomsky, "The evolution of the language faculty: clarifications and implications," *Cognition*, vol. 97, no. 2, pp. 179–210, 2005.

[30] J. D. Bransford, A. L. Brown, and R. R. Cocking, *How People Learn*, pp. 10–19, National academy press, Washington, DC, USA, 2000.

[31] C. Wang, C. H. Liu, B. Wang, Z. Y. Zhao, and K. Tang, "TCN-KT: temporal convolutional knowledge tracing model based on fusion of personal basis and forgetting," *Application Research of Computers*, vol. 39, no. 05, pp. 1496–1500, 2022.

[32] J. Zeng, Q. Zhang, N. Xie, and B. Yang, "Application of deep self-attention in knowledge tracing," 2021, https://arxiv.org/abs/2105.07909.

[33] A. Ghosh, N. Heffernan, and A. S. Lan, "Context-aware attentive knowledge tracing," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2330–2339, California, CA, USA, August 2020.

[34] S. Pandey and J. Srivastava, "RKT: relation-aware self-attention for knowledge tracing," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 1205–1214, New York, NY, USA, October 2020.

[35] W. Lee, J. Chun, Y. Lee, K. Park, and S. Park, "Contrastive learning for knowledge tracing," in *Proceedings of the ACM Web Conference 2022*, pp. 2330–2338, Lyon, France, April 2022.