Time-Aware Attentional Knowledge Tracing based on Pre-training and Feature Extraction

Yuxin Tian
School of Information Science and Engineering
East China University of Science and Technology
Shanghai, China
y80210031@mail.ecust.edu.cn

Zhanquan Wang✉
School of Information Science and Engineering
East China University of Science and Technology
Shanghai, China
zhqwang@ecust.edu.cn

***Abstract*—Knowledge tracing (KT) has attracted increasing attention as the level of education informatization has increased. KT models students' changing knowledge states over time based on their historical question responding and then forecasts students' success in question answering. Many knowledge tracing models have been proposed to support the smart education system, but these models frequently fail to provide good interpretability, are insufficiently extracted for question information and frequently ignore the influence of the time factor on prediction. To address these concerns, we propose the Time-Aware Attentional Knowledge Tracing based on Pre-training and Feature Extraction (PFTKT), which is implemented in three steps: First, pre-train the question inputs to enrich the question representation. Second, extract student attributes to guide prediction. Finally, add time distance parameters to the attention mechanism to model students' forgetting behavior. We conduct comprehensive tests on three real-world datasets to validate the model's effectiveness, and the results reveal that PFTKT surpasses previous knowledge tracing models in terms of AUC scores, and we also validate the effectiveness of each important component of PFTKT.**

***Index Terms*—deep learning, online learning, knowledge state, knowledge tracing model**

I. Introduction

With the rapid growth of information technology and the rising updating of knowledge, offline education is increasingly unable to meet people's new expectations for lifelong learning, which online education may answer. Online education not only makes learning more convenient for students, but it also opens up new potential and challenges for personalized learning and tailored teaching. Students will face difficulties such as selecting learning resources, fragmented learning, and difficulty managing learning progress. Teachers are also confronted with new challenges: it is difficult to understand students' needs and measure students' learning effects quickly [1]. To address a number of issues raised by online learning, many researchers have proposed developing the knowledge tracing models based on student interaction data. These models analyze learners' learning behaviors to determine their unique characteristics and then provide them with individualized interventions. Furthermore, knowledge tracing models can be utilized to dynamically monitor learning behaviors, forecast learning trends in real-time, and properly evaluate learning results to provide adaptive guidance to learners [2].

In particular, the researcher first obtains the sequences of question-answering behaviors developed by students during the learning process, and then uses a knowledge-tracking algorithm to model the learner and the sequence of learning behaviors, ultimately reasoning about the learner's skill and cognitive level [3] [4].Finally, based on the learner's learning state as determined by the knowledge tracing model analysis, multiple learning paths and individualized guidance are given to the students, hence improving learning efficiency. Many various types of KT models have been proposed thus far, and the present KT models are primarily classified into three categories. The first type of model is a probability distribution model, such as the Hidden Markov Model (HMM) or the Bayesian Knowledge Tracing Model (BKT) [5]. The second group includes factor analysis methods such as Item Response Theory (IRT) [6], Additive Factor Model (AFM) [7], and others, which extract factors connected to students and issues to demonstrate students' learning behaviors. The third model group is the most used KT, which is centered on deep sequence models, such as the Deep Knowledge Tracing Model (DKT) . Long Short-Term Memory Network (LSTM) [8] and Self-Attention Knowledge Tracing Model (SAKT) [9] are attention-based memory networks.

Since then, additional researchers have proposed models based on the attention mechanism, and the model's prediction accuracy is increasing steadily. The SAKT model introduces the attention mechanism to KT and achieves better prediction results. Its proposal also enables researchers to see the potential of the attention mechanism in the field of knowledge tracing. Poor interpretability is a question that arises when employing the attention mechanism, so in order to address it, we present the Time-Aware Attentional Knowledge Tracing based on Pre-training and Feature Extraction (PFTKT). This model adds a time factor to the attention mechanism to imitate students' forgetting behavior while enriching the question representation and extracting student attributes to aid prediction. PFTKT improves the model's interpretability and prediction accuracy, and our experimental results on three real-world datasets demonstrate that PFTKT outperforms the existing KT model. We further test the approach's effectiveness using ablation experiments for each important component. Each major component's validity.

2833-8898/23/$31.00 ©2023 IEEE
DOI 10.1109/CyberC58899.2023.00029

The contributions of our paper can be summarized as follows:

- Unlike other models that take the question information directly as input, PFTKT extracts the question features by pre-training the question information, enriching the question representation and allowing the question matrix to hold more auxiliary information.
- Added a student-related feature extraction module, which extracts student personalized information such as learning ability and then introduces the personalized features into the decoder to drive prediction generation.
- Modified the multi-head attention mechanism by adding the temporal distance metric parameter to the multi-head attention, which allows the model to imitate this behavior of students forgetting knowledge points, boosting the model's interpretability.

II. Related Work

Assessing students' knowledge status based on answer records is an important aspect of KT research, with the basic idea being to track the change in students' knowledge status level over time based on their learning behaviors [10]. Early knowledge transfer depended mostly on probabilistic models that saw knowledge mastery prediction as a probability distribution inference question of "mastery/non-mastery," such as the Hidden Markov Model (HMM) and Bayesian Knowledge Tracing Model (BKT). Hidden Markov models may forecast the probability distribution of hidden variables based on learners' previous learning behaviors and depict the transfer between states [11], allowing learners' learning states to be predicted.

PIECH proposed the classical Deep Knowledge Tracing (DKT) model in 2015 [3] , which recognizes that the learner practice sequence is a typical time series data, so it introduces Recurrent Neural Networks (RNNs) into the KT to be able to capture the historical correlation of the time series, and the variants that continue to be developed based on the DKT [9] [12] [13] [14] [15] achieved prediction accuracies far superior to other KTs. The Dynamic Key-Value Memory Model [16] (DKVMN) is a DKT model variation that uses a key-value memory network to uncover the relationship between the question and the underlying talent. However, both the DKT and the DKVMN use notions as a replacement for the question, failing to capture the individual distinctions inherent in the situation.To address this issue, numerous academics have advocated using graphs to represent the relationship between concepts and issues, leading to the development of Graph-based Knowledge Tracing [17] (GKT), PEBG [18], and Graph-based Interaction Modeling [19] (GIKT). GKT initially randomly initializes a conceptual graph and then trains and predicts it constantly to optimize it, which is computationally hard and data-heavy. The size of the set will influence the prediction outcomes.Based on their notions, PEBG and GIKT define the relationship between difficulties.

Another widely used method in the field of KT is factor analysis, which incorporates knowledge from psychology and is designed from a cognitive diagnostic perspective, such as Item Response Theory (IRT) [6], Attentional Factorization Machine (AFM) [7], Performance Factors Analysis (PFA) [20], and other models, which focus on student and question-related factors, such as students' learning ability and question characteristics. Furthermore, Multidimensional Item Response Theory (MIRT) [21] broadens the dimensions based on IRT in order to extract more information from student interaction data. DIRT [22] and NeuralCD [23], on the other hand, extend IRT by incorporating deep neural networks into the factor analysis approach, allowing the model to mine more complex information from interaction data while retaining the interpretability of the factor analysis approach.

Currently, the attention mechanism is widely used in the field of KT, and models that use it have a significant gain in prediction accuracy. The attention mechanism is more adaptable than recurrent neural networks and memory-based neural networks, and it performs well in natural language processing tasks. The Self-Attentive Knowledge Tracing [9] (SAKT) model is the first approach to use attentional mechanisms in KT. The SAKT model's basic design is quite similar to the Transformer [4] model, which is a useful model for many sequence prediction questions. The SAINT+ [15] model extends SAKT by exploiting the whole Transformer to discover the hidden patterns of student interaction sequences, and it successfully applied SOTA on the EdNet [24] dataset. Multifactor-aware Dual Attention Knowledge Tracing (MF-DAKT) was proposed by Zhang et al [25], which improves question representations and uses multifactor to explain students' learning progress based on the dual attention mechanism. Zhou [26] proposed personalized deep knowledge tracing through distinguishable interaction sequences (LANA), in which an interpretable Rasch model [27] was used to cluster students for hierarchical learning and personalized DKT. Lee [28] proposed Monotonic Attention-Based Knowledge Tracing (MonaCoBERT), a BERT architecture capable of representing monotonic convolutional multi-head attention and modeling for student forgetting, as well as an effective embedding strategy based on classical test theory to represent difficulty. The context-aware attention model (AKT) [12] and the relationally-aware self-attention model (RKT) [13] also use an attention mechanism and introduce a decay function to model students' forgetting behavior.

III. Question Setup

Knowledge tracing is a key method for achieving individualized learning [29], and many scholars have performed extensive research, practice, and investigation in order to model knowledge tracing. The information in a learner's learning record primarily consists of questions, answers, correctness or incorrectness, and time of answer in a time series. Given a triad of learners' time series, the knowledge tracing issue can be characterized as predicting whether the learner answered the question correctly or incorrectly at the $i+1$ th moment. At moment i, learners' learning state can be characterized as a triad of questions, concepts, and responses (q_i^s, c_i^s, r_i^s),

where $q_i^s \in N^+$ signifies the questions encountered during the learning process and N^+ denotes the number of questions. $c_i^s \in N^+$ stands for the knowledge concepts to which the questions are corresponding. $r_i^s \in \{0, 1\}$ indicates whether the learner answered the questions properly or incorrectly. Usually, $r_i^s = 0$ indicates a bad answer, while $r_i^s = 1$ indicates a correct answer. The learner's learning behavior at the time i can then be represented by a series of triples: $\{(q_1^s, c_1^s, r_1^s), (q_2^s, c_2^s, r_2^s), \ldots, (q_i^s, c_i^s, r_i^s)\}$. The flowchart of knowledge tracing is shown in Fig. 1.

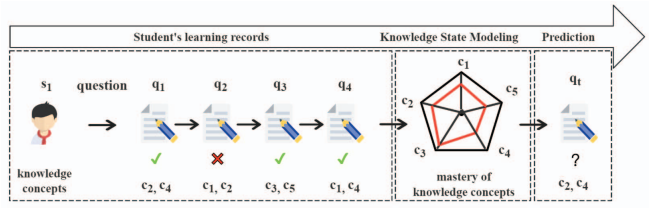

Fig. 1. Knowledge tracing flowchart

IV. Methodology

In this section, we introduce the PFTKT model's major components. Fig. 2 depicts the PFTKT model framework diagram. The pre-training component, the student feature extractor, and the attention mechanism based on the temporal distance metric are the three fundamental components of the PFTKT model. First, we pre-train the question correlations and complexity, increasing the auxiliary information in the question input. The student feature extractor is then used to extract the students' personalized features, and the personalized feature parameters are supplied into the decoder to help guide prediction generation. Finally, we describe the behavior of students losing knowledge points using an attention mechanism based on a temporal distance measure.

A. Question Pre-training

We provide three question-related information: question-related relations, concept-related relations, and question difficulty, which enable to design a question information graph, to enrich the representation of questions so that they can carry more information. Given that students' answers frequently contain multiple concepts, and that a concept frequently appears in multiple questions, we can represent the questions as a relationship graph in which there are not only relationships between questions and concepts, but also correlations between concepts and implicit correlations between questions, which have frequently been neglected in previous studies.

Unlike previous studies, we propose in this study a method of generating Question Embedding by pre-training the information of the questions, which allows the low-dimensional Embedding of the questions to be better learned, and auxiliary information, which has been neglected in previous studies, is also included in these low-dimensional Embedding. The auxiliary information contains the difficulty of the question and three kinds of correlations: question-similarity information, concept-similarity information, and question-concept correlations. We use the product layer [30] to fuse the question features, concept features, and attribute features to generate the final Question Embedding, and the Question Embedding generated in this manner can provide more information as input, which includes not only the question difficulty information and the correlation between questions and concepts, but also auxiliary information about the question and concepts. The correlation diagram between questions and concepts is shown in Fig. 3.

1) Question-Concept Relationship Computation: In the question-concept correlation graph, the edges between the question vertices and concept vertices explicitly indicate the explicit relationship between the question and the concept. As a result, we model the explicit link by the concept's and question's local proximity, namely by the inner product of the concept and the question.

$$\hat{y}_{ij} = \sigma(q_i^T c_j), i \in [1, \ldots, |Q|], j \in [1, \ldots, |C|] \quad (1)$$

where Q is the identity matrix of the question and C is the identity matrix of the concept. A question q_i^T is the i th row of matrix Q and a concept c_j is the j th row of matrix C. $\hat{\mathrm{y}}_{ij}$ indicates whether there is an edge connecting the question and the concept, and is 1 if there is one, and 0 otherwise. $\sigma(x)$ is a sigmoid function that converts the value of the question-concept relationship into a probability. The cross-entropy loss function is then used to train the question-concept relationship.

$$L_1(Q, C) = \sum_{i=1}^{|Q|} \sum_{j=1}^{|C|} -(y_{ij} \log \hat{y}_{ij} + (1 - y_{ij}) \log(1 - \hat{y}_{ij})) \quad (2)$$

2) Calculating Similarity: The question-concept correlation graph has two types of similarity relations: similarity between concepts and similarity between questions. We define the neighbor set of question q_i as $\Gamma_Q(i) = c_j | r_{ij} = 1$ and the neighbor set of concepts c_j as $\Gamma_C(j) = q_i | r_{ij} = 1$. Then the question similarity can be defined as:

$$y_{ij}^q = \begin{cases} 1, & \Gamma_Q(i) \cap \Gamma_Q(j) \neq \emptyset, i, j \in [1, \ldots, |Q|]; \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Similarly, the concept similarity can be defined as:

$$y_{ij}^c = \begin{cases} 1, & \Gamma_C(i) \cap \Gamma_C(j) \neq \emptyset, i, j \in [1, \ldots, |C|]; \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

In the vertex feature space, we employ the inner product to estimate the implicit relationship between questions and concepts.

$$\hat{y}_{ij}^q = \sigma(q_i^T q_j), i, j \in [1, \ldots, |Q|] \quad (5)$$

$$\hat{y}_{ij}^c = \sigma(c_i^T c_j), i, j \in [1, \ldots, |C|] \quad (6)$$

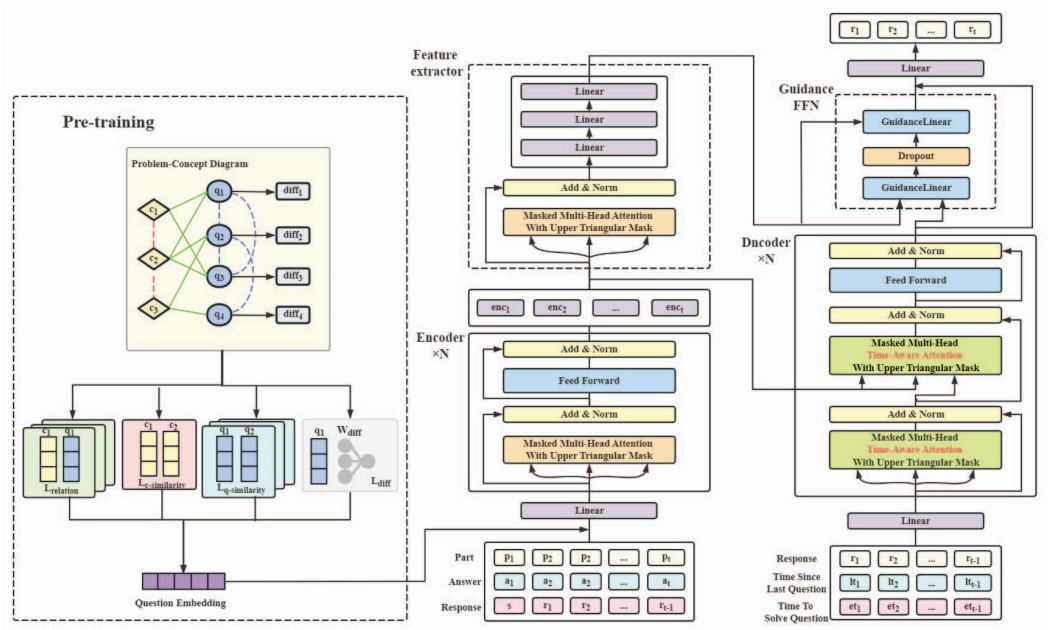

Fig. 2. The overall model architecture of PFTKT.

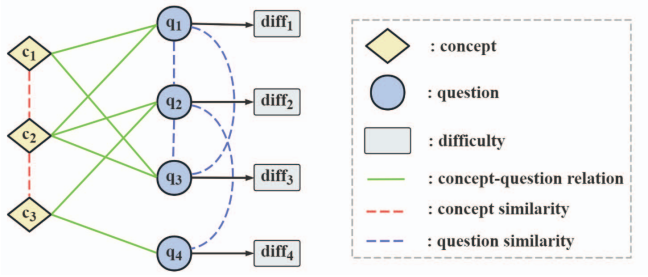

Fig. 3. Question-concept relationship diagram.

Finally, we train the implicit relationships between questions and concepts in the vertex feature space by minimizing the cross entropy.

$$L_2(Q) = \sum_{i=1}^{|Q|}\sum_{j=1}^{|Q|} -(y_{ij}^q \log \hat{y}_{ij}^q + (1-y_{ij}^q)\log(1-\hat{y}_{ij}^q)) \quad (7)$$

$$L_3(C) = \sum_{i=1}^{|C|}\sum_{j=1}^{|C|} -(y_{ij}^c \log \hat{y}_{ij}^c + (1-y_{ij}^c)\log(1-\hat{y}_{ij}^c)) \quad (8)$$

3) Question Difficulty: Another crucial piece of auxiliary information in knowledge tracing is the difficulty of the questions. It is possible to discriminate between questions that are based on the same concept, which is important for predicting the degree of knowledge mastery. Therefore, we introduce the attribute of question difficulty as auxiliary information as well. A question is typically thought to be simple to solve if many people successfully answer it, and vice versa the question is difficult to solve. For the question i, we use the percentage of correct responses to indicate the difficulty of the question (d_i). Since the difficulty of the question is a scalar, we convert the question representation vector to a scalar value through a fully connected layer:

$$\hat{d}_i = p_i W_d iff \quad (9)$$

where p_i is the projection vector of question i and $W_d iff$ is the weight matrix of the fully connected layer. The error is measured by the squared loss function:

$$L_4(Q,C,\theta) = \sum_{i=1}^{|Q|} -(d_i - \hat{d}_i)^2 \quad (10)$$

where θ denotes all the parameters in the network layer.

4) Joint Optimization: For the generated Question Embedding to retain both explicit relations, implicit relations, and question difficulty information, we combine the above four loss functions to form a joint training framework:

$$\min_{Q,C,\theta} \lambda(L_1(Q,C) + L_2(Q) + L_3(C)) + (1-\lambda)L_4(Q,C,\theta) \quad (11)$$

Where λ denotes the parameter of the trade-off between the question-concept-related information and the question difficulty information. Once the above joint training framework is optimized, question embedding can be obtained. The computational flow of the whole joint optimization framework is shown in Fig. 4.

B. Student Feature Extractor

Previous knowledge tracing models based on the self-attention mechanism handled student interaction data as ordinary time-series data and used a uniform training technique,

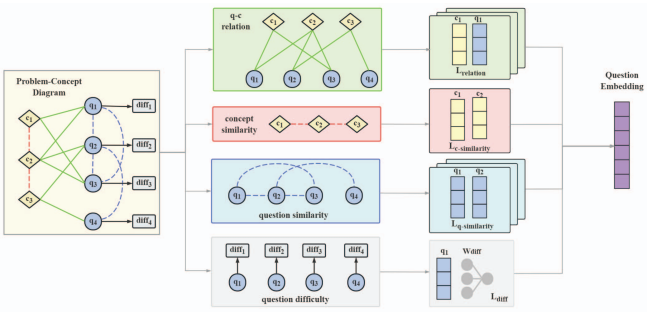

Fig. 4. Pre-training joint optimization framework

which ignores the learning capacity gap across students and reduces prediction effectiveness. In this paper, we offer a student feature extractor that summarizes students' inherent features from their interaction data with questions, hence assisting the decoder in training individualized parameters. Our student feature extractor consists of an attention layer, a normalization layer, and several linear layers. The attention layer extracts student features, while the normalization and linear layers recreate and refine the extracted student features. The structure of the student feature extractor is shown in Fig. 5.

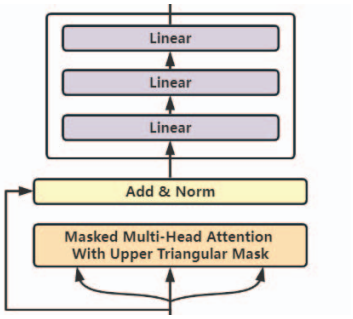

Fig. 5. Student feature extractor

The student feature extractor obtains the student interaction data from the encoder and then generates feature parameters that describe the student's learning capacity, which in turn can guide the decoder in predicting student responses. To make the student feature parameters better direct the decoder's prediction, we construct a guidance module. The inputs to the guidance module are the decoder output x, and the student feature parameters θ, respectively, and the output is y. Previously, researchers would use feedforward neural networks to project x to y, however in our study, the projection matrix of x will be dynamically mapped based on the student feature parameters θ. The formula is as follows:

$$y = W^x x + b^x, W^x = W_1^\theta \theta + b_1^\theta, b^x = W_2^\theta \theta + b_2^\theta \quad (12)$$

The above equation can be written more simply as follows:

$$y = (W\theta)x + b = GuidanceLinear(x, \theta) \quad (13)$$

Following the decoder, the guidance module is added to the feedforward neural network, resulting in model prediction guided by feature parameters. Fig. 6 depicts the internal structure of the feedforward neural network based on the guidance module.

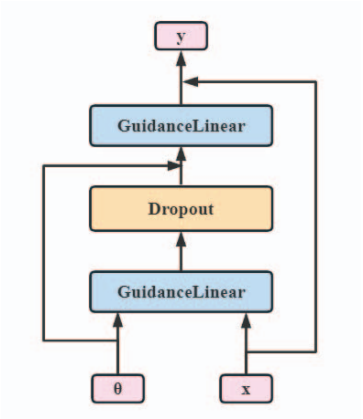

Fig. 6. Internal structure of feedforward neural network based on guidance module.

C. Time-Aware Attention

Since the self-attention mechanism was applied to knowledge tracing, there has been a significant improvement in prediction accuracy on knowledge tracing. The core of the self-attention mechanism is the scaling dot product attention mechanism [31]. In the scaled dot product attention mechanism, each encoder and knowledge retriever has a key, query, and value embedding layer, which maps the inputs to the output query, key, and value with dimensions D_k, D_k, and D_v, respectively, and processes the query, key, and value into matrices Q, K, and V, respectively, and then applies the Softmax function to calculate the weights, and the computed output matrix is:

$$Attention(Q, K, V) = Softmax(\frac{(Q^T K)}{\sqrt{D_k}})V \quad (14)$$

However, because learning is a dynamic process, students' memory of knowledge will gradually decline over time, and students constantly go through the process of memorization and forgetting during the learning process, simply using the scaled dot product attention mechanism cannot train a model that fits the real situation. Furthermore, a student's recent performance is more indicative of his current mastery of knowledge than his past performance, and when a learner is faced with a new question, past experiences about irrelevant concepts and long-ago experiences often do not help the student solve the question. Based on the above ideas, we add the following penalty exponential decay term to the attention score of the scaled dot product attention mechanism:

$$\alpha_{t,\tau} = Softmax(\frac{e^{-\theta \cdot d(t,\tau)} Q_t^T K_\tau}{\sqrt{D_k}})V_\tau \quad (15)$$

TABLE I
DATASET STATISTICS

	ASSIST2009	ASSIST2012	EdNet
Students	3840	27403	5000
Questions	15913	47109	13169
Concepts	126	263	188
Records	190335	1867158	1048576
Records per student	49.57	68.17	209.72
Concepts per question	1.21	1.00	2.28
Records per question	11.96	39.63	79.62

where $Q_t \in \mathbb{R}^{D_k \times 1}$ denotes the query corresponding to the question answered by the student at time t, while $K_\tau \in \mathbb{R}^{D_k \times 1}$ and $V_\tau \in \mathbb{R}^{D_v \times 1}$ denote the key and value of the question at time step τ, θ is a parameter of the learnable decay rate and $d(t, \tau)$ is the temporal distance between time steps t and τ. When calculating the attention weights, the above algorithm takes into account not only the similarity between the relevant query and key, but also the temporal distance between the current question and the previous questions. In summary, when a current question is substantially similar to a previous question, we will lessen the similarity in a way that decays with time, simulating students' memorizing and forgetting.

V. EXPERIMENTS

In this section, we analyze the efficacy of PFTKT and alternative baseline models using three real-world datasets, as well as many sets of ablation experiments aimed to test the usefulness of each important component in the PFTKT model.

A. Datasets

We use three real-world datasets to evaluate the predictive performance of the PFTKT model and design multiple sets of comparative tests to compare the PFTKT with numerous state-of-the-art KT models to validate its usefulness for response prediction. The datasets we used are ASSIST2009, ASSIST2012, and EdNet, and the statistics of these three datasets are shown in Table I.

- ASSISTment 2009 and ASSISTment 2012: Both datasets were collected from the ASSISTments online tutoring platform. For both datasets, we did some preprocessing; we first removed records for questions that were not labeled with concepts, and then we also removed users who had too few answer records. After preprocessing, the ASSIST09 dataset contains 126 concepts, 15,913 questions, and 3,840 students, for a total of 190,335 student interaction records. In contrast, the ASSIST12 dataset has 263 concepts, 47,109 questions, and 27,403 students, for a total of 1,867,158 student interaction records.
- EdNet: This dataset was collected by [24]. EdNet is divided into four subsets, each of which contains a particular form of student activity. Due to the vast amount of data in the entire dataset, in this study, we randomly choose 1048576 data from 5000 students from the EdNet-KT1 dataset, which contains 188 concepts and 13169 questions, with a maximum of 6 concepts contained in one question.

B. Baselines

To validate the effectiveness of PFTKT for knowledge tracing, we compare the model with the state-of-the-art KT models.

- DKT: DKT is the first model to apply deep learning to knowledge tracing, it models students' knowledge state and predicts future behaviors using LSTM, and students' knowledge state in DKT is represented by hidden vectors of LSTM.
- DKVMN: DKVMN extends DKT, which utilizes two novel matrices, a key matrix, and a value matrix, to store the relationship between different concepts and each student's mastery of the corresponding concept, respectively.
- SAKT: SAKT is the first model to incorporate an attention mechanism into knowledge tracing, it replaces the LSTM module with a self-attention module and outperforms the RNN-based model on several knowledge tracing datasets.
- DSAKT: DSAKT is a knowledge tracing model that improves on SAKT and further increases the prediction effect of the self-attention based model on knowledge tracing datasets.
- SAINT: The architecture of SAKT only includes one self-attention module, but SAINT employs the complete transformer, which considerably improves prediction accuracy.
- SAINT+: SAINT+ is the successor to SAINT, which adds two more features to the model input. Experiments have demonstrated the effectiveness of SAINT+ compared to SAINT.
- AKT: AKT monitors students' knowledge level by employing a monotonic attention mechanism that accounts for the human brain's "forgetting" habit.
- MF-DAKT: This model enriches the question representation and uses multifactorial based dual attention mechanisms to model students' learning progress.
- GIKT: It uses graph convolutional networks to capture relationships between questions and employs a recall module to capture long-term dependencies.

We followed the source code and work settings for all baselines. Our model's hyperparameter settings are as follows: the AdamW optimizer's learning rate is set to 5e-4, the length of the input sequence is set to 100, and the batch size is set to 256. In our research, we use 5-fold cross-validation, with each fold recording 20% of the interactions as the test set and the remaining 80% of the data as the training set. In this research, two generally used KT measures are employed for evaluation: the AUC (area under the ROC curve) and the ACC (accuracy). Since AUC is insensitive to the question of category imbalance, utilizing AUC as the evaluation foundation is a more

TABLE II
THE RESULTS OVER THREE DATASETS

Dataset	ASSIST2009		ASSIST2012		EdNet	
Metric	ACC	AUC	ACC	AUC	ACC	AUC
DKT	0.721	0.744	0.713	0.690	0.656	0.692
DKVMN	0.728	0.751	0.712	0.682	0.651	0.687
SAKT	0.767	0.731	0.722	0.716	0.687	0.702
DSAKT	**0.798**	0.749	0.731	0.737	0.664	0.685
SAINT	0.723	0.759	0.740	0.742	0.656	0.749
SAINT+	0.733	0.752	0.736	0.744	0.694	0.756
AKT	0.754	0.802	0.735	0.738	0.705	0.731
MF-DAKT	0.793	0.807	0.733	0.750	0.705	0.761
GIKT	0.741	0.781	0.715	0.724	0.709	0.748
PFTKT	0.782	**0.812**	**0.745**	**0.763**	**0.712**	**0.772**

effective method when the dataset is extremely imbalanced in classification.

C. Results and Analysis

The experimental results of various baseline KT models as well as PFTKT on different datasets are shown in Table II. As shown in the table, PFTKT achieves very excellent results on the three datasets, except for not obtaining the highest prediction accuracy in ASSIST2009, it achieves better results than SOTA. Since AUC is more indicative of prediction performance when the datasets are classified as imbalanced, we take AUC to compare PFTKT and SOTA. PFTKT obtained 81.2%, 76.3%, and 77.2% AUC on the three datasets, whereas SOTA obtained 80.7%, 75.0%, and 76.1% AUC, respectively. PFTKT outperforms SOTA in AUC by 0.62%, 1.73%, and 1.45%, respectively.

D. Ablation Study

In this section, we design ablation experiments to test the effectiveness of each essential component of the PFTKT model. Three sets of ablation experiments, PFTKT-PRE (remove the pre-training component), PFTKT-FEATURE (remove the feature extractor), and PFTKT-TIME (remove the temporal distance metric), are designed to determine whether or not to use pre-training to generate Question Embedding, whether or not to use the feature extractor, and whether or not to use the temporal distance metric. The results of the ablation experiments are compared with the original model for comparison and the results of the ablation experiments are shown in Table III. The table shows that PFTKT performs best across all datasets, demonstrating that all three components of our proposed model play a positive effect. Furthermore, we discovered that the model's performance degraded the most after removing the temporal distance metric, while the model without pre-training achieved an AUC that was second only to the original model, indicating that the introduction of the temporal distance metric is very effective in improving the model's performance, as well as that pre-training has a limited effect on the model, and that there is still room for continued improvement.

TABLE III
ABLATION STUDY

Dataset	ASSIST2009		ASSIST2012		EdNet	
Metric	ACC	AUC	ACC	AUC	ACC	AUC
PFTKT	**0.782**	**0.812**	**0.745**	**0.763**	**0.712**	**0.772**
PFTKT-PRE	0.767	0.801	0.744	0.759	0.706	0.765
PFTKT-FEATURE	0.770	0.792	0.729	0.755	0.698	0.761
PFTKT-TIME	0.763	0.788	0.732	0.753	0.695	0.759

VI. CONCLUSION

In this work, we proposed the Time-Aware Attentional Knowledge Tracing based on Pre-training and Feature Extraction, which improves on three previously proposed KT models based on attentional mechanisms. First, we constructed a question-concept correlation graph, and then used pre-training to build question embedding, thus enriching the representation of the question. Second, we extracted students' intrinsical feature, which summarizes students' feature parameters from their interaction data with questions, thus helping to train personalized models. Finally, we added a temporal distance metric into the attention mechanism to model the behavior of students forgetting knowledge points, boosting the model's interpretability as well as the model's prediction accuracy. We employed three real-world datasets in our tests to validate that PFTKT outperforms the state-of-the-art, and we also completed ablation experiments to validate the effectiveness of each component of PFTKT. However, PFTKT currently has certain flaws: First, in the student feature extractor, the student's features are not classified, which undoubtedly weakens the guiding role of feature parameters in training. Second, our research on incorporating time into the attention process is insufficient, and we must continue to investigate the effect of temporal elements on prediction, such as time distance and answer time. We will continue to address these two concerns and train better models in the future.

REFERENCES

[1] Y. Huo, D. F. Wong, L. M. Ni, L. S. Chao, and J. Zhang, "Knowledge modeling via contextualized representations for lstm-based personalized exercise recommendation," *Information Sciences*, vol. 523, pp. 266–278, 2020.

[2] Q. Jiang, W. Zhao, S. Li, and P. Wang, "Research on the mining of precise personalized learning path in age of big data: Analysis of group learning behaviors based on aprioriall. e-educ," *Res*, vol. 39, pp. 45–52, 2018.

[3] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein, "Deep knowledge tracing," *Advances in neural information processing systems*, vol. 28, 2015.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[5] Z. A. Pardos and N. T. Heffernan, "Modeling individualization in a bayesian networks implementation of knowledge tracing," in *User Modeling, Adaptation, and Personalization: 18th International Conference, UMAP 2010, Big Island, HI, USA, June 20-24, 2010. Proceedings 18*, pp. 255–266, Springer, 2010.

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

[7] H. Cen, K. Koedinger, and B. Junker, "Learning factors analysis–a general method for cognitive model evaluation and improvement," in *International conference on intelligent tutoring systems*, pp. 164–175, Springer, 2006.

[8] A. Graves and A. Graves, "Long short-term memory," *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.

[9] S. Pandey and G. Karypis, "A self-attentive model for knowledge tracing," *arXiv preprint arXiv:1907.06837*, 2019.

[10] G. Abdelrahman, Q. Wang, and B. Nunes, "Knowledge tracing: A survey," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–37, 2023.

[11] J. Zhao, S. Bhatt, C. Thille, D. Zimmaro, and N. Gattani, "Interpretable personalized knowledge tracing and next learning activity recommendation," in *Proceedings of the Seventh ACM Conference on Learning@ Scale*, pp. 325–328, 2020.

[12] 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, 2020.

[13] 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, 2020.

[14] Y. Choi, Y. Lee, J. Cho, J. Baek, B. Kim, Y. Cha, D. Shin, C. Bae, and J. Heo, "Towards an appropriate query, key, and value computation for knowledge tracing," in *Proceedings of the seventh ACM conference on learning@ scale*, pp. 341–344, 2020.

[15] D. Shin, Y. Shim, H. Yu, S. Lee, B. Kim, and Y. Choi, "Saint+: Integrating temporal features for ednet correctness prediction," in *LAK21: 11th International Learning Analytics and Knowledge Conference*, pp. 490–496, 2021.

[16] 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, 2017.

[17] H. Nakagawa, Y. Iwasawa, and Y. Matsuo, "Graph-based knowledge tracing: modeling student proficiency using graph neural network," in *IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 156–163, 2019.

[18] Y. Liu, Y. Yang, X. Chen, J. Shen, H. Zhang, and Y. Yu, "Improving knowledge tracing via pre-training question embeddings," *arXiv preprint arXiv:2012.05031*, 2020.

[19] Y. Yang, J. Shen, Y. Qu, Y. Liu, K. Wang, Y. Zhu, W. Zhang, and Y. Yu, "Gikt: a graph-based interaction model for knowledge tracing," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I*, pp. 299–315, Springer, 2021.

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

[21] R. P. Chalmers, "mirt: A multidimensional item response theory package for the r environment," *Journal of statistical Software*, vol. 48, pp. 1–29, 2012.

[22] S. Cheng, Q. Liu, E. Chen, Z. Huang, Z. Huang, Y. Chen, H. Ma, and G. Hu, "Dirt: Deep learning enhanced item response theory for cognitive diagnosis," in *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 2397–2400, 2019.

[23] F. Wang, Q. Liu, E. Chen, Z. Huang, Y. Chen, Y. Yin, Z. Huang, and S. Wang, "Neural cognitive diagnosis for intelligent education systems," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 6153–6161, 2020.

[24] Y. Choi, Y. Lee, D. Shin, J. Cho, S. Park, S. Lee, J. Baek, C. Bae, B. Kim, and J. Heo, "Ednet: A large-scale hierarchical dataset in education," in *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II 21*, pp. 69–73, Springer, 2020.

[25] M. Zhang, X. Zhu, C. Zhang, Y. Ji, F. Pan, and C. Yin, "Multi-factors aware dual-attentional knowledge tracing," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2588–2597, 2021.

[26] Y. Zhou, X. Li, Y. Cao, X. Zhao, Q. Ye, and J. Lv, "Lana: towards personalized deep knowledge tracing through distinguishable interactive sequences," *arXiv preprint arXiv:2105.06266*, 2021.

[27] T. Eckes, "Introduction to many-facet rasch measurement," *Franfurt am Main: Peter Lang*, 2011.

[28] U. Lee, Y. Park, Y. Kim, S. Choi, and H. Kim, "Monacobert: Monotonic attention based convbert for knowledge tracing," *arXiv preprint arXiv:2208.12615*, 2022.

[29] W. Kong, S. Han, and Z. Zhang, "Construction of adaptive learning path supported by artificial intelligence," *Modern Distance Education Research*, vol. 32, no. 3, pp. 94–103, 2020.

[30] Y. Qu, H. Cai, K. Ren, W. Zhang, Y. Yu, Y. Wen, and J. Wang, "Product-based neural networks for user response prediction," in *2016 IEEE 16th international conference on data mining (ICDM)*, pp. 1149–1154, IEEE, 2016.

[31] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.